

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»

СИСТЕМИ І МЕТОДИ ПІДТРИМКИ ПРИЙНЯТТЯ РІШЕНЬ

Підручник

*Затверджено Вченою радою КПІ ім. Ігоря Сікорського як підручник
для здобувачів ступеня магістра за спеціальністю 124 Системний аналіз*

Електронне мережне навчальне видання

Київ
КПІ ім. Ігоря Сікорського
2022

Автори: *Бідюк Петро Іванович*, д-р техн. наук, проф.
Тимощук Оксана Леонідівна, канд. техн. наук, доц.
Коваленко Анатолій Єніфанович, канд. техн. наук, доц.
Коршевніук Лев Олександрович, канд. техн. наук, доц.

Рецензенти: *Цюцюра С. В.*, д-р техн. наук, проф., завідувач кафедри інформаційних технологій Київського національного університету будівництва і архітектури
Положаєнко С. А., д-р техн. наук, проф., завідувач кафедри комп'ютеризованих систем та програмних технологій Національного університету «Одеська політехніка»

Відповідальний редактор *Данилов В. Я.*, д-р. техн. наук., проф., професор кафедри математичних методів системного аналізу КПІ ім. Ігоря Сікорського

Гриф надано Вченою радою КПІ ім. Ігоря Сікорського (протокол № 3 від 14.02.2022 р.)

Системно викладено методика проектування інформаційних систем підтримки прийняття рішень (СППР), яка ґрунтується на сучасних принципах системного аналізу.

Розглянуто функціональну структуру та альтернативні підходи до реалізації СППР із використанням моделей проектування різних типів, а також на основі створення прототипу. Подано всі етапи проектування інформаційної СППР, які охоплюють написання технічного завдання, створення повного проекту системи, програмування та тестування модулів, а також супроводження системи на етапі її експлуатації замовником. Розглянуто основні типи архітектур СППР залежно від обмежень, які накладаються методами оброблення даних, типом інформації, необхідної для прийняття рішень та альтернативними процедурами оцінювання варіантів. Описано типи інтерфейсів інформаційних систем та процедуру їх проектування. Значну увагу приділено практичним прикладам побудови конкретних типів СППР. Зокрема розглянуто СППР для прогнозування часових рядів, СППР на основі мереж Байєса і системи, яка ґрунтується на використанні експертних оцінок.

Для студентів, аспірантів та викладачів, а також інженерів, що спеціалізуються в галузі проектування та розроблення систем підтримання прийняття рішень у сферах виробництва, бізнесу, науки та техніки.

Реєстр. № П 21/22-007. Обсяг 29,6 авт. арк.

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
проспект Перемоги, 37, м. Київ, 03056

<https://kpi.ua>

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

© П. І. Бідюк, О. Л. Тимощук, А. Є. Коваленко, Л. О. Коршевніук, 2022
© КПІ ім. Ігоря Сікорського, 2022

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ	12
ВСТУП.....	15
РОЗДІЛ 1 АРХІТЕКТУРА СИСТЕМ ПІДТРИМКИ ПРИЙНЯТТЯ РІШЕНЬ	20
1.1 Прийняття рішень у системах підтримки прийняття рішень	20
1.1.1 Визначення поняття систем підтримки прийняття рішень	20
1.1.2 Інтелектуальні системи підтримки прийняття рішень	22
1.1.3 Узагальнена будова системи підтримки прийняття рішень	23
1.1.4 Характеристика процесів прийняття рішень	24
1.1.4.1 Основні ознаки рішення.....	24
1.1.4.2 Класифікація рішень	25
1.1.4.3 Моделі прийняття рішень	25
1.1.5 Характеристика етапів прийняття рішень.....	26
1.1.5.1 Основні етапи прийняття рішення.....	26
1.1.5.2 Послідовність прийняття рішення.....	27
1.1.5.3 Помилки прийняття рішення	28
1.1.6 Ефективність і фактори прийняття рішень	28
1.1.7 Умови прийняття рішень	30
1.2 Невизначеність прийняття рішень	31
1.2.1 Рівні невизначеності	31
1.2.2 Типи невизначеностей	32
1.2.2.1 Характеристика основних типів невизначеностей.....	32
1.2.2.2 Додаткові типи невизначеностей.....	33
1.3 Людино-машинні системи підтримки прийняття рішень.....	34
1.3.1 Основні напрями створення людино-машинних систем.....	35
1.3.2 Людино-машинні системи прийняття рішень з передбаченням	36
1.4 Автономні системи прийняття рішень	36
1.5 Вимоги до сучасних систем підтримки прийняття рішень	38
1.6 Особливості застосування сучасних СППР.....	39
1.6.1 Сфери застосування СППР у мікро і макроекономіці	39
1.6.2 Системи СППР розподілу інвестиційних і фінансових ресурсів	40
1.6.3 Особливості інтелектуальних засобів підтримки прийняття рішень.....	41
1.7 Класифікація СППР	43
1.7.1 Підходи до класифікації СППР	43
1.7.2 Оперативні і стратегічні СППР	44
1.8 Історія виникнення та розвитку інформаційних СППР	45
1.8.1 Напрями застосування засобів обчислювальної техніки.....	45
1.8.2 Зовнішня пам'ять	47
1.8.3 Інформаційні системи і бази даних	48
1.8.4 Хронологія розвитку інформаційних СППР.....	49
1.9 Системи OLAP	51
1.10 Покоління систем підтримки прийняття рішень.....	52
1.10.1 Покоління СППР	52
1.10.2 СППР на основі GRID-систем	53

1.10.3 СППР на основі інфраструктур хмарних обчислень	55
Запитання та завдання для самоконтролю	56
РОЗДІЛ 2 ОСНОВИ ПОБУДОВИ СИСТЕМ ПІДТРИМКИ ПРИЙНЯТТЯ	
РІШЕНЬ	59
2.1 Аналіз процесу прийняття рішення	59
2.1.1 Атрибути прийняття рішень	59
2.1.2 Ухвалення рішень особою прийняття рішень	61
2.2 Стратегії прийняття рішення	64
2.2.1 Оптимізаційна стратегія	65
2.2.2 Стратегія «вибір першої прийнятної стратегії»	66
2.2.3 Стратегія аспектного виключення	67
2.2.4 Інкрементна стратегія	67
2.2.5 Стратегія змішаного сканування	68
2.2.6 Аналітико-ієрархічна стратегія	68
2.3 Прийняття особистого рішення відносно власної кар'єри	69
2.3.1 Фаза аналізу шляху	69
2.3.2 Фаза проектування альтернатив	70
2.3.3 Фаза вибору кращої (прийнятної) альтернативи	72
2.4 Етапи проектування СППР	73
2.4.1 Склад етапів проектування СППР	73
2.4.2 Когнітивний процес проектування	73
2.4.3 Визначення і декомпозиція задачі прийняття рішень	77
2.4.3.1 Визначення задачі прийняття рішень	77
2.4.3.2 Декомпозиція задачі прийняття рішень	78
2.4.3.3 Опис ситуації з прийняття рішення	80
2.4.3.4 Формат опису ситуації з ухвалення рішення	81
2.4.4 Аналіз ситуації з прийняття рішення	85
2.4.4.1 Визначення обмежень процесу прийняття	85
2.4.4.2 Характеристика типових проблем прийняття рішень	87
2.4.4.3 Категорії опису ситуацій	90
2.4.5 Визначення функцій системи підтримки прийняття рішень	91
2.4.5.1 Функції для підтримки прийняття рішень з боку ОПР	91
2.4.5.2 Вибір конкретних обчислювальних процедур	92
2.4.6 Вибір технології для реалізації СППР	92
2.5 Засоби підтримки прийняття рішень	93
2.5.1 Види математичного інструментарію	93
2.5.2 Правила реалізації функцій СППР	95
Запитання та завдання для самоконтролю	96
РОЗДІЛ 3 ЗАСТОСУВАННЯ МОДЕЛЕЙ І ДАНИХ У СППР	100
3.1 Вибір моделей і критеріїв для побудови СППР	100
3.2 Вибір моделі для оцінювання наслідків прийняття рішень із використанням СППР	103
3.3 Вибір інструментарію для інформаційного менеджменту	107
Запитання та завдання для самоконтролю	111
РОЗДІЛ 4 ПРОЕКТУВАННЯ АРХІТЕКТУРИ СППР	112

4.1 Основні підходи до проектування СППР	112
4.1.1 Інформаційний підхід	112
4.1.2 Підхід, заснований на знаннях	114
4.1.3 Інструментальний підхід	117
4.2 Типи архітектур спеціалізованих СППР	119
4.2.1 Текстово-орієнтовані СППР	119
4.2.2 СППР з використанням бази даних	121
4.2.3 СППР на основі використання електронних таблиць	122
4.2.4 СППР на основі алгоритмічних процедур для розв'язання задач	123
4.2.5 СППР на основі правил	126
4.2.6 Гібридні СППР	127
4.3 Функції системи оброблення даних та генерування результатів	129
4.4 Вибір та описання алгоритмів, на яких базується СОДГР	131
4.5 Дані і знання систем підтримки прийняття рішень	133
4.5.1 Характеристики даних і знань	133
4.5.2 Моделі подання знань	134
4.5.2.1 Продукційна модель	135
4.5.2.2 Семантичні мережі	136
4.5.2.3 Фрейми	138
4.5.2.4 Формальні логічні моделі	141
4.5.2.5 Процедурні моделі	141
4.6 Функції системи подання результатів	141
4.6.1 Форма подання результатів	141
4.6.2 Використання кольорів, мигання і клавіатури	142
4.6.3 Використання клавіатури	143
Запитання та завдання для самоконтролю	144
РОЗДІЛ 5 ПРОЕКТУВАННЯ ІНТЕРФЕЙСУ КОРИСТУВАЧА	147
5.1 Вимоги до інтерфейсів інформаційних систем	147
5.2 Характеристики адаптивного інтерфейсу користувача	149
5.2.1 Адаптивні інтерфейси	149
5.2.2 Природно-мовний інтерфейс	151
5.2.3 Можливості подання даних інтерфейсу	155
5.2.4 Загальні принципи побудови адаптивних інтерфейсів	156
5.3 Проектування інтерфейсу з урахуванням людського фактора	158
5.3.1 Роль людського фактора	158
5.3.2 Загальні принципи врахування людського фактора	160
5.4 Тональність діалогу та термінологія	166
Запитання та завдання для самоконтролю	168
РОЗДІЛ 6 МЕТОДОЛОГІЯ РОЗРОБКИ І РЕАЛІЗАЦІЇ СИСТЕМ	
ПІДТРИМКИ ПРИЙНЯТТЯ РІШЕНЬ	170
6.1 Умови успішної реалізації СППР	170
6.1.1 Визначення потреб і вимог замовника	170
6.1.2 Правильний вибір інструментальної платформи для реалізації проекту	172
6.1.3 Вимоги до знань та досвіду проектування	172
6.1.4 Вимоги до розуміння проблеми прикладної області	172
6.1.5 Вимоги доступу до інформації	173

6.1.6	Вимога фінансової підтримки	173
6.1.7	Створення групи із розроблення проекту	174
6.2	Фактори ризиків при проектуванні та реалізації СППР	174
6.3	Менеджмент ризиків проектів з розроблення СППР	177
6.4	Послідовність розробки та реалізації СППР	179
6.4.1	Основні етапи розробки та реалізації	179
6.4.2	Формулювання постановки задачі	180
6.4.3	Аналіз можливості реалізації СППР	181
6.4.4	Формулювання вимог до СППР	183
6.4.5	Створення проекту СППР	185
6.4.6	Опис програмного забезпечення системи	186
6.4.7	Послідовність програмування модулів системи	187
6.4.8	Послідовність тестування системи	188
6.4.9	Підготовка інсталяційної версії і експлуатаційної документації	188
6.4.10	Упровадження системи на технічних засобах замовника	189
6.4.11	Супроводження СППР	189
6.5	Створення прототипу СППР	190
6.6	Сучасні СППР ринку інформаційних послуг	191
6.6.1	Типи сучасних СППР	192
6.6.2	Система «Симплан»	193
6.6.3	Система PIMS	195
6.6.4	Система ISDS	196
6.6.5	Система IFPS	197
6.6.6	Система MAUD	198
6.6.7	Пакети і інтегровані системи	199
6.6.8	Розроблення російських виробників	200
6.7	Проектування банківських СППР	201
6.7.1	Аналітичний програмний комплекс	201
6.7.2	Підсистема збору і збереження даних	202
6.7.3	Підсистема доступу до даних, аналізу і репортингу	204
6.7.4	Інструмент опису семантичного шару	204
6.8	Основні можливості звітів у АПК	205
6.8.1	Інструментарій АПК	205
6.8.2	Рейтинг-аналіз	207
6.8.3	Довідники	208
6.8.4	Підсистема адміністрування	209
6.9	Перелік аналітичних і управлінських задач, які розв'язує АПК ...	210
6.10	Етапи впровадження АПК та її налагодження	212
6.10.1	Вартість впровадження АПК	212
6.10.2	Розробка АПК	213
6.10.3	Упровадження і налагодження системи	214
6.10.4	Супровід і подальший розвиток системи	215
	Запитання та завдання для самоконтролю	215
РОЗДІЛ 7 ІНФОРМАЦІЙНІ СИСТЕМИ ПІДТРИМКИ ПРИЙНЯТТЯ		
РІШЕНЬ НА ОСНОВІ РОЗПОДІЛЕНИХ СИСТЕМ		218
7.1	Архітектура розподілених інформаційних систем	218

7.1.1 Узагальнена структура розподіленої інформаційної системи.....	218
7.1.2 Основні характеристики розподілених систем.....	221
7.1.2.1 Прозорість.....	221
7.1.2.2 Відкритість.....	223
7.1.2.3 Масштабованість.....	224
7.2 Концепція апаратних і програмних рішень.....	225
7.2.1 Класифікації РІС.....	225
7.2.2 Концепції програмних рішень.....	231
7.3 Програмне забезпечення проміжного рівня.....	235
7.4 Архітектури розподілених систем.....	238
7.4.1 Модель взаємодії клієнт-сервер розподілених систем.....	238
7.4.2 Архітектура клієнт-сервер корпоративних інформаційних систем.....	240
7.4.3 Модель розподіленої системи з проміжним рівнем.....	243
7.4.4 Механізми реалізації взаємодій проміжного рівня.....	246
7.5 Зв'язок на основі повідомлень.....	253
7.5.1 Застосування сокетів.....	253
7.5.2 Стандарт MPI.....	254
7.5.3 Проміжний рівень на основі черг повідомлень.....	255
7.5.4 Зв'язок на основі потоків даних.....	257
7.6 Реалізації розподілених систем.....	258
7.6.1 Промисловий стандарт CORBA.....	258
7.6.2 Система GLOBE.....	269
7.6.3 Система CODA.....	277
7.6.4 Система World Wide Web.....	283
7.6.5 Система Lotus Notes.....	291
7.6.6 Система Jini.....	295
7.7 Програмні агенти.....	300
Запитання та завдання для самоконтролю.....	302
РОЗДІЛ 8 МАТЕМАТИЧНІ МЕТОДИ ПОБУДОВА СПІР ПРИ	
ПРОГНОЗУВАННІ ДИНАМІКИ ЧАСОВИХ РЯДІВ.....	305
8.1 Методи прогнозування на основі часових рядів.....	305
8.2 Основні складові прогнозування процесу.....	306
8.2.1 Детермінований тренд.....	307
8.2.2 Стохастичний тренд.....	308
8.2.3 Прогнозування коливань, що накладаються на тренд.....	308
8.2.4 Прогнозування дисперсії процесу.....	309
8.3 Умовні та безумовні статистичні характеристики.....	310
8.4 Побудова функцій прогнозування.....	314
8.4.1 Прогнозування без знаходження розв'язку рівнянь.....	314
8.4.1.1 Прогнозування для рівняння $AR(1)$	314
8.4.1.2 Узагальнення функції прогнозування (на процес $ARCS(p,q)$).....	317
8.4.2 Побудова функції прогнозування на основі різницевого рівняння.....	320
8.4.3 Прогнозування з мінімальною дисперсією.....	328
8.4.4 Альтернативна форма функції прогнозування на один крок.....	332
8.4.5 Функція прогнозування на два кроки.....	335
8.5 Загальна постановка задачі прогнозування з мінімумом дисперсії.....	338

8.6	Приклад побудови функцій прогнозування	346
8.7	Схема створення СППР для прогнозування часових рядів	349
8.8	Алгоритм процесу аналізу та прогнозування	349
8.8.1	Попереднє оброблення та аналіз даних.....	350
8.8.2	Перевірка наявності нелінійностей	352
8.8.3	Визначення порядку нелінійності	355
8.8.4	Перевірка процесу на стаціонарність.....	355
8.8.5	Перевірки наявності гетероскедастичності	360
8.8.5.1	Тест Уайта	361
8.8.5.2	Тест Бройша-Пагана/Годфрі	362
8.8.5.3	Тест Голдфельда-Квандта	364
8.8.6	Визначення типу моделі для опису гетероскедастичності.....	366
8.8.7	Визначення способу вилучення або моделювання тренду.....	369
8.8.8	Побудова моделі часового ряду.....	369
8.8.8.1	t - статистика Стьюдента.....	370
8.8.8.2	Коефіцієнт детермінації R^2	371
8.8.8.3	Сума квадратів похибок моделі.....	372
8.8.8.4	Інформаційний критерій Акайке	372
8.8.8.5	Статистика Дарбіна-Уотсона.....	373
8.8.8.6	Статистика Фішера.....	373
8.8.8.7	Коефіцієнт Тейла.....	374
8.8.9	Побудова функції прогнозування.....	376
8.8.10	Уточнення вхідних даних	378
	Запитання та завдання для самоконтролю	378
РОЗДІЛ 9 ОБ'ЄКТНО-ОРІЄНТОВАНІ СППР НА ОСНОВІ		
ПРОГНОЗУВАННЯ ЧАСОВИХ РЯДІВ.....		
9.1	Приклад реалізації СППР на основі часових рядів	380
9.1.1	Основні етапи побудови СППР прогнозування часових рядів	380
9.1.2	Архітектура і структурна схема СППР	382
9.2	Класи компонентів СППР	383
9.2.1	Діаграма основних класів	383
9.2.2	Опис основних класів	384
9.2.2.1	Клас DataClass	384
9.2.2.2	Клас BasicForm.....	386
9.2.2.3	Клас ShowRowsForm.....	386
9.2.2.4	Клас GraphRowForm.....	387
9.2.2.5	Клас DataConversionClass.....	387
9.2.2.6	Клас StatisticalAnalysisClass.....	387
9.2.2.7	Клас ARMAForm.....	388
9.2.2.8	Клас MethodClass.....	389
9.2.2.9	Клас PrognosisForm	389
9.2.2.10	Клас ARMAModelClass.....	389
9.2.2.11	Клас MatrixClass	390
9.2.2.12	Клас Leksem.....	391
9.2.2.13	Клас Expression.....	391
9.2.2.14	Клас GlobalDataTables.....	392
9.2.2.15	Клас InnerCLIFunctions.....	393

9.3 Функціональні можливості СППР	393
9.3.1 Функціональна схема СППР	393
9.3.2 Редагування та попереднє оброблення даних	395
9.3.3 Графічне подання даних	395
9.3.4 Аналіз даних	396
9.3.5 Побудова та вибір кращої моделі АРКС	396
9.3.6 Прогнозування	397
9.3.7 Командний блок СППР	398
9.4 Приклад застосування СППР для моделювання та прогнозування	
ВВП України	402
Запитання та завдання для самоконтролю	407
РОЗДІЛ 10 ПОБУДОВА СППР НА ОСНОВІ БМ.....	409
10.1 Основи застосування байєсових мереж у прийнятті рішень	409
10.1.1 Сфери використання	409
10.1.2 Побудова БМ.....	410
10.1.2.1 Постановка задачі.....	410
10.1.2.2 Теорема Байєса і формування висновку на її основі.....	411
10.1.2.3 Застосування БМ у розпізнаванні кіл	414
10.1.2.4 Правдоподібність прийняття рішення	415
10.1.3 Проста БМ	416
10.1.4 Подання і оброблення БМ.....	420
10.1.5 Байєсове дерево прийняття рішень	422
10.2 Евристичний метод побудови БМ.....	426
10.2.1 Складність побудови БМ	426
10.2.2 Спрощений евристичний метод.....	427
10.2.3 Принцип формування опису БМ мінімальної довжини	428
10.2.4 Простий алгоритм навчання БМ з використанням опису мінімальної довжини.....	431
10.2.5 Евристичний алгоритм побудови БМ.....	435
10.3 Оцінки якості навчання БМ.....	437
10.4 Практичне застосування БМ	438
10.4.1 Прогнозування Нью-Йоркського біржового композитного індексу.....	438
10.4.2 Застосування БМ для постановки діагнозу.....	441
10.5 Інформаційні СППР на основі БМ.....	444
10.5.1 Передумови застосування сучасних технологій обробки даних.....	444
10.5.2 Технологія розробки інформаційної системи на основі БМ	447
10.5.3 Системи на основі БМ.....	449
10.6 Розробка експертної СППР для підприємства на основі БМ	457
10.6.1 Виділення класу задач експертної системи.....	457
10.6.2 Побудова БМ у експертній системі	459
10.6.3 Модель виробничої діяльності підприємства	461
10.6.4 Моделювання управлінських рішень	464
Запитання та завдання для самоконтролю	469
РОЗДІЛ 11 ПОБУДОВА СППР НА ОСНОВІ МЕТОДІВ ЕКСПЕРТНОГО ОЦІНЮВАННЯ.....	472

11.1	Методи експертних оцінок для розв'язання задач прийняття рішень розподілу ресурсів	472
11.1.1	Змістовне формулювання задачі.....	472
11.1.2	Задачі розподілу ресурсів в умовах невизначеності.....	474
11.1.3	Методологія евристичного розв'язання задач розподілу ресурсів	476
11.1.4	Застосування методу експертних оцінок Дельфі.....	478
11.1.5	Методологія розподілу ресурсів в умовах невизначеності	479
11.1.5.1	Основні етапи розв'язання задачі.....	479
11.1.5.2	Постановка і формалізація задачі	481
11.1.5.3	Оцінювання варіантів.....	483
11.1.5.4	Узагальнення оцінок	484
11.1.5.5	Пошук рішення.....	488
11.2	Підходи до аналізу невизначеності.....	489
11.3	Нечітка логіка Л. Заде.....	491
11.4	Система нечіткого логічного висновку.....	493
11.4.1	Схема системи нечіткого логічного висновку	493
11.4.2	Методи дефазифікації	495
11.4.2.1	Центроїдний метод.....	495
11.4.2.2	Методи максимуму.....	496
11.4.2.3	Метод центра максимумів.....	498
11.4.3	Модифікації процесу формування нечіткого логічного висновку.....	499
11.4.3.1	Нечіткий логічний вивід Мамдані	499
11.4.3.2	Нечіткий логічний висновок Ларсена.....	501
11.4.3.3	Нечіткий логічний висновок Цукамото.....	502
11.4.3.4	Нечіткий логічний вивід Сугено.....	503
11.4.3.5	Спрощений нечіткий логічний висновок	504
11.5	Експертне оцінювання проектів.....	505
11.5.1	Підходи до оцінювання проектів.....	505
11.5.2	Застосування лінгвістичних змінних.....	506
11.5.3	Геометричний спосіб визначення оцінок.....	508
11.6	Вибір функції належності нечітких понять	510
11.6.1	Обґрунтування функції належності.....	510
11.6.2	Методи побудови функцій належності	511
11.6.3	Когнітивні функції належності та метод їх побудови	512
11.6.4	Евристичний підхід до побудови функції належності нечітких оцінок	513
11.6.5	Методи визначення властивостей у мисленні ОПР	514
11.6.6	Наметоподібні функції належності та метод α , β -рівнів їх побудови.....	517
11.7	Побудова набору лінгвістичних змінних на основі кластерів.....	519
11.8	Процедури агрегування та аналізу погодженості експертних оцінок поектів.....	528
11.9	Аналіз погодженості оцінок ОПР	529
11.9.1	Коефіцієнт конкордації Кендала	529
11.9.2	Коефіцієнт рангової кореляції Спірмена.....	531
11.9.3	Жорсткі і м'які методи погодження оцінок	531
11.9.4	Загальний підхід до аналізу і формування погоджених оцінок	533
11.9.5	Метод α , β - коаліцій	534

11.10 Агрегування оцінок всіх ОПР	540
11.10.1 Методи розрахунку агрегованих оцінок	540
11.10.2 Алгоритм агрегування оцінок для проекту	545
11.11 Процедури вибору варіантів.....	547
11.11.1 Виділення ресурсів для проекту	547
11.11.2 Виділення ресурсів проектам з ресурсним обмеженням.....	549
11.11.3 Застосування процедури вибору проектів.....	550
11.12 Побудова СППР для розподілу обмежених ресурсів.....	552
11.12.1 Структура інформаційної системи підтримки прийняття рішень.....	552
11.12.2 Застосування СППР для розподілу ресурсів	555
Запитання та завдання для самоконтролю	561
РОЗДІЛ 12 ПРОЕКТУВАННЯ СППР	565
12.1 Визначення типу СППР	565
12.2 Визначення інших типів СППР.....	567
12.3 Послідовність побудови СППР управління процесом приватизації	570
Контрольні запитання.....	574
РЕКОМЕНДОВАНА ЛІТЕРАТУРА	575
Додаток А Перелік технічної документації на розроблення інформаційних систем	583
Додаток Б Технічна пропозиція	584
Додаток В Технічне завдання.....	585
В.1 Форма титульного аркуша технічного завдання на систему	590
В.2 Форма останнього аркуша технічного завдання	591
Додаток Г Технічний проект	592
Г.1 Форма титульного аркуша технічного проекту	597
Г.2 Форма останнього аркуша технічного проекту	598
Додаток Д Експлуатаційна документація.....	599
Д.1 Інструкція стосовно експлуатації комплексу технічних засобів	600
Д.2 Посібник користувача	601
Д.3 Формуляр	603
Додаток Е Організаційно-розпорядча документація	605
Е.1 Оформлення та зберігання документів	609
Е.2 Заявка на виконання робіт	610

ПЕРЕЛІК СКОРОЧЕНЬ

AI	–	адаптивний інтерфейс
АКФ	–	автокореляційна функція
АМАВ	–	аспектний мультиатрибутний вибір
АПК	–	аналітичний програмний комплекс
АР	–	авотрегресія
АРІКС	–	авторегресія з інтегрованим ковзним середнім
АРКС	–	авторегресія з ковзним середнім
АРУГ	–	авторегресійна умовно гетероскедастична (модель)
АСУ	–	автоматизована система управління
БД	–	база даних
БЗ	–	база знань
БЗД	–	база знань і даних
БМ	–	байєсівська мережа
ВВП	–	валовий вироблений продукт
ДЕС	–	дорадча економічна система
ЕС	–	експертна система
ЕТ	–	електронна таблиця
ІС	–	інформаційна система
ІТ	–	інформаційні технології
КІС	–	комп'ютерна інформаційна система
КС	–	ковзне середнє
ЛОС	–	локальна операційна система
МАП	–	максимальна абсолютна похибка
МГВА	–	метод групового врахування аргументів
МіАП	–	мінімальна абсолютна похибка

ММП	–	метод максимальної правдоподібності
МНК	–	метод найменших квадратів
МС	–	мовна система
НЛ	–	нечітка логіка
НМНК	–	нелінійний метод найменших квадратів
ОМД	–	опис мінімальної довжини
ОПР	–	особа, яка приймає рішення
ОС	–	операційна система
ПЗ	–	програмне забезпечення
ПЗ	–	програмне забезпечення
ПМ	–	природно-мовний (інтерфейс)
ПС	–	програмна система
РІС	–	розподілена інформаційна система
РІСППР	–	розподілена інформаційна система підтримки прийняття рішень
РМНК	–	рекурсивний метод найменших квадратів
РП	–	робоча пам'ять
СБДЗ	–	система бази даних і знань
СКП	–	сума квадратів похибок
СОДГР	–	система оброблення даних та генерування результатів
СП	–	середня похибка
СППР	–	система підтримки прийняття рішень
СПР	–	система подання результатів
СУБД	–	система управління базою даних
СУБЗ	–	система управління базою знань
СУБМ	–	система управління базою моделей
ТБ	–	техніка безпеки

ТЗ	–	технічне завдання
ТНМ	–	теорія нечітких множин
ТП	–	технічний проект
ТПР	–	теорія прийняття рішень
ТУІ	–	таблиця умовних імовірностей
УАРУГ	–	узагальнена авторегресійна умовно гетероскедастична модель
УМАВ	–	утилітарний мультиатрибутний вибір
УММ	–	узагальнений метод моментів
ФН	–	функція належності
ЧАКФ	–	частково автокореляційна функція

ВСТУП

Застосування сучасних теорій інформаційних технологій, менеджменту у бізнесі, концепцій отримання знань, принципи побудови інтелектуальних систем повинні ґрунтуватись на усвідомленій цілеспрямованій діяльності людини. Важливу роль у цьому відіграє створення інформаційних систем підтримки прийняття рішень (СППР) адаптованих до навколишнього середовища. У СППР застосовують експертні оцінки, нейронні мережі, алгоритми м'яких обчислень, методи оптимізації, регресійний аналіз, байєсівські моделі та методи тощо.

Досягнення усвідомлених цілей потребує створення інструментальних засобів для скорочення зростаючих витрат за обмеженими наявними ресурсами. Потреба в таких засобах викликала до життя, у межах методів та систем штучного інтелекту, цілий спектр інформаційних технологій, покликаних допомогти у справі управління суспільством, виробництвом, торгівлею, кредитною і фінансовою сферами. Найпоширенішими застосуваннями цих технологій є: експертні системи, дорадчі системи, інтелектуальні системи, інформаційні системи підтримки прийняття рішень. Спільною їх рисою є використання знань людини-експерта. Серед таких систем можна виділити системи економічної сфери, які отримали назву дорадчих економічних систем (ДЕС).

Усі ДЕС умовно можна поділити на два класи: системи, що відтворюють усвідомлені розумові зусилля людини (статичні детерміновані або стохастичні системи); системи, що відтворюють підсвідомі розумові дії людини (еволюційні системи на основі нейротехнологій і генетичних алгоритмів). До першого класу ДЕС входять три підкласи: розрахунково-діагностичні системи; системи підтримки прийняття рішень; експертні системи на основі наближених міркувань, які

можна розглядати як один із типів СППР. До другого класу входять системи: нейромережових обчислень; орієнтовані на природно-мовні запити.

Дорадчі системи призначені для допомоги у прийнятті рішень у випадках пошуку альтернатив і вибору одного правильного рішення. Зазвичай існуючі методики проектування ДЕС спираються на визначеність варіантів рішень і наслідків їх прийняття (зокрема, матриць рішень). Однак практика свідчить, що пошук альтернативних варіантів і побудова функцій, спроможних оцінити наслідки прийняття варіанта, є особливо складною проблемою у фінансово-економічній сфері.

Метою створення ДЕС є: надання управлінському персоналу знань, яких їм бракує у процесі виконання своїх фахових обов'язків; навчання управлінського персоналу діям, які необхідні для виконання рекомендацій, наданих СППР із подальшим контролем такого виконання.

Основним підходом до організації ДЕС є технологія створення СППР, яка передбачає центральною ланкою особу, яка приймає рішення (ОПР). Тому структуру і склад ДЕС визначають з: інформаційних та інструментальних потреб ОПР. Ці потреби виникають у ОПР у процесі реалізації поставлених цілей та виконання ним службових обов'язків.

До інформаційного поля прийняття рішень може входити зовнішня інформація (зовнішнього середовища) і внутрішня, що виникає у процесі використання ДЕС. Інструментарій ОПР зазвичай має функції, що забезпечують реалізацію функціональних інформаційних технологій (ІТ).

Серед ІТ можна виділити [14]: ІТ, що констатують (забезпечують користувача необхідною інформацією для розпізнавання існуючої економічної ситуації); ІТ моделюючого типу на основі застосування математичної моделі з оцінки можливих результатів прийнятого рішення,

відповідаючи на запитання «що робити, якщо?»; ІТ на основі експертних систем, що дозволяє відповісти на запитання «як зробити, щоб?»

Підручник присвячено задачам проектування, розроблення та реалізації інформаційних СППР (ІСППР). Прийнята методологія проектування ІСППР ґрунтується на системному підході, який передбачає ієрархічність структури процесів і об'єктів, стосовно яких приймається рішення, та врахуванням невизначеностей опису процесів (зокрема, невизначеності структурного, параметричного, статистичного характеру).

У першому розділі наведено поняття СППР, досліджено поняття рішення, етапи процесу прийняття рішень, проведено аналіз видів рішень, умов та середовищ прийняття рішень, визначено доцільність та можливість підтримки різних стадій прийняття рішень, розглянуто передумови виникнення, етапи розвитку та класифікації СППР.

У другому розділі розглянуто методику побудови СППР. Аналіз процесу прийняття рішень охоплює ключові атрибути та обмеження процесу ухвалення рішення, що виконує ОПР без допоміжних засобів. Досліджено стратегії прийняття рішень, їх переваги та недоліки. Наведено приклад процесу прийняття особистого рішення. Розглянуто поетапну методику проектування СППР, до якої входить вивчення і декомпозиція задачі прийняття рішень, аналіз ситуації з прийняття рішення, визначення функцій системи та вибір технології реалізації СППР.

Третій розділ охоплює проблему застосування моделей і даних у СППР, дослідження вибору моделі під час проектування СППР та моделі для оцінювання результатів роботи СППР, вибір методу керування даними. Для кожного етапу оброблення даних запропоновано застосування множини критеріїв якості для забезпечення якості використання СППР.

У четвертому розділі наведено методику проектування архітектури СППР, розглянуто інформаційний та інструментальний підходи до

проектування. Досліджено основні типи архітектур ІСППР залежно від обмежень, які обумовлені методами оброблення даних, типом інформації, та альтернативними процедурами оцінювання. Описано основні структурні підсистеми СППР, зокрема, мовну систему, базу знань та даних, систему оброблення даних та генерації результатів, систему подання результатів.

П'ятий розділ присвячено проектуванню інтерфейсу користувача. Описано типи інтерфейсів ІСППР, досліджено вимоги до інтерфейсу та принципи його формування.

У шостому розділі визначено умови успішного розроблення і реалізації СППР з урахуванням ризиків проектування. Наведено повний цикл розроблення та реалізації СППР, який охоплює написання технічного завдання, створення повного проекту системи, програмування та тестування модулів, а також супроводження СППР на етапі її експлуатації замовником. Розглянуто альтернативні підходи до реалізації інформаційної системи із використанням моделей різних типів і на основі створення прототипу. Виконано аналіз переваг та недоліків кожного з підходів із наведенням прикладів.

Сьомий розділ охоплює аналіз реалізацій розподілених інформаційних систем, які набувають дедалі більшого значення у створенні інформаційних систем оброблення даних. Розподілені ІСППР передбачають багаторівневий підхід до створення середовища оброблення даних ІСППР.

Восьмий розділ містить теоретичні та прикладні аспекти побудови СППР при прогнозуванні динаміки часових рядів різної природи. Розглянуто можливості застосування різницевого рівняння до опису динаміки процесів на основі часових рядів, а також задачі побудови функцій їх прогнозування. Запропоновано методику оцінювання багатокрокових прогнозів без використання проміжних оцінок. Отримано

функції прогнозування для багатьох моделей, розглянуто аналіз точності прогнозів за допомогою множини статистичних характеристик з наведенням прикладу при прогнозуванні часових рядів.

У дев'ятому та десятому розділах висвітлено питання побудову ІСППР на основі мереж Байєса, що дозволяє враховувати структурні і статистичні невизначеності досліджуваних процесів. Запропонована послідовність побудови байєсівських мереж довіри може бути використана, зокрема, у моделюванні соціально-економічних процесів та для описання динаміки і статички технічних систем. Розглянуто процес створення експертної системи для оцінювання і прогнозування стану підприємства. Наведено приклад застосування розробленої СППР.

Одинадцятий розділ охоплює застосування методів оброблення та використання експертних оцінок у ІСППР. Методи дозволяють розв'язати проблеми домінування якісних, погано визначених факторів, які виявляються у нечітких, неточних, розпливчастих властивостях процесів та явищ, а також урахувати у математичних моделях невизначеності реальних процесів. Запропоновано методологію розв'язання задач на основі нечіткої логіки, яка передбачає формування множини критеріїв, експертне оцінювання альтернатив, аналіз погодженості думок експертів, агрегування оцінок та вибір кращих варіантів. Розглянуто приклад застосування СППР для розв'язання задач розподілу ресурсів і вибору варіантів.

Дванадцятий розділ містить завдання для самостійної роботи, які можна використати під час проектування інформаційних систем різного призначення.

РОЗДІЛ 1 АРХІТЕКТУРА СИСТЕМ ПІДТРИМКИ ПРИЙНЯТТЯ РІШЕНЬ

Автоматизація процесів підтримки прийняття рішень зумовило появу кількох підходів до побудови СППР. Системи СППР ґрунтуються на застосуванні сучасних математичних методів прийняття рішень, різноманітних математичних моделях, технічних і програмних засобах. Тому визначення поняття систем підтримки прийняття рішень часто пов'язують з цими методами, моделями, засобами.

1.1 Прийняття рішень у системах підтримки прийняття рішень

1.1.1 Визначення поняття систем підтримки прийняття рішень

Єдиного визначення для СППР немає. Існує кілька поширених визначень СППР, які відображають основні особливості будови, використання та ефективності застосування цих систем.

До таких визначень можна віднести такі [<https://studfile.net/preview/3564600/page:50>].

СППР – інтерактивна комп'ютерна система для підтримки різних видів діяльності під час прийняття рішень стосовно слабоструктурованих і неструктурованих проблем. Такі системи дають змогу особам ОПР відшукувати релевантні дані, згенеровані системами оброблення транзакцій та інших внутрішніх інформаційних джерел, а також надають доступ до зовнішньої, по відношенню до організації, інформації. СППР дає змогу користувачам моделювати й аналізувати інформацію у такий спосіб, який буде найефективнішим для вироблення певного специфічного рішення і буде забезпечувати підтримку в інтерактивному режимі [40].

СППР – інтерактивна прикладна система, яка забезпечує користувачам, котрі приймають рішення, легкий і зручний доступ до даних і моделей з метою прийняття рішень у слабоструктурованих і неструктурованих ситуаціях з різних галузей людської діяльності.

СППР – інтерактивна автоматизована система, яка допомагає особі, яка приймає рішення, використовувати дані і моделі для розв’язання неструктурованих і слабоструктурованих проблем.

СППР – комп’ютерна інформаційна система, яка використовується для підтримки різних видів діяльності під час прийняття рішень у ситуаціях, коли неможливо або небажано мати автоматичну систему, яка повністю виконує весь процес створення рішень.

В англomовних виданнях поширеним є застосування терміну decision support systems (DSS) [54]. Зокрема, DSS визначають так: «Sometimes a manager makes decisions individually. In other cases decision making may be distributed, involving the combined and coordinated efforts of many knowledge workers. Both individual and distributed decision making are susceptible to support by systems that facilitate, expand, or enhance a manager’s ability to work with one or more kinds of knowledge. Such knowledge-based systems are called decision support systems» [54].

Тому надалі під СППР або DSS будемо розуміти інтерактивну комп’ютерну автоматизовану систему (програмний комплекс), яка призначена для допомоги та підтримки різних видів діяльності людини при прийнятті рішень стосовно розв’язання слабоструктурованих або неструктурованих проблем. Застосування СППР забезпечує виконання ґрунтовного та об’єктивного аналізу предметної області при прийнятті рішень у складних умовах.

1.1.2 Інтелектуальні системи підтримки прийняття рішень

Сучасні інтелектуальні СППР (ІСППР) характеризують широким використанням інтелектуальних методів оброблення даних та прийняття рішень, а також проектуванням інтерфейсу на принципах інтелектуалізації процесу взаємодії користувач-система. Зокрема, інтелектуалізація стосується адаптації системи до реакції користувача та його уподобань стосовно способів подання результатів оброблення даних, вибору зручного для конкретного користувача способу введення, редагування та поповнення бази знань і даних.

Системи ІСППР – це широкий і надзвичайно зручний клас інформаційних обчислювальних систем оброблення даних, які дозволяють інтегрувати різноманітні дані.

До інтелектуальних методів і алгоритмів (оскільки алгоритм – це реалізація методу) пошуку та оброблення даних відносять такі: алгоритми пошуку даних із заданими характеристиками, алгоритми приведення даних до заданої форми, ігрові методи пошуку та оброблення даних, способи подання знань у базі знань. До даного класу методів також відносять: різні види планування, методи оброблення невизначеностей, методи прийняття рішень на основі нейромереж, імовірнісних моделей і суджень, чіткої та нечіткої логіки, методи навчання на основі попереднього досвіду та спостережень, ймовірнісні методи оброблення лінгвістичної інформації, мережі Байєса, методи сприйняття та розпізнавання образів, методи керування у робототехніці тощо.

1.1.3 Узагальнена будова системи підтримки прийняття рішень

У процесі проектування СППР необхідно виконати аналіз відомих архітектур (або структур) та обґрунтовано вибрати ту, яка найкраще відповідає поставленій задачі. Важливим є обґрунтування вибору закритої або відкритої архітектури системи. Обов'язково треба визначити визначення СППР, необхідне для розуміння головної мети проекту СППР.

Розмаїття СППР зумовлено наявністю різних типів та форм підтримки прийняття рішень, а також архітектур СППР. Необхідно розуміти, що СППР будь-якого типу повинна містити:

- підсистему введення та аналізу запитів користувача (ПВАЗ);
- підсистему оброблення запитів користувача та генерації результатів (ПОЗГР);
- базу знань і даних (БЗД);
- підсистему подання результатів (ППР) в зручній для користувача формі.

Функції цих підсистем повинні бути присутні як у комерційних, так і у некомерційних варіантах СППР.

Підсистема введення та аналізу запитів користувача повинна включати опис всіх припустимих запитів користувача та їх формалізоване подання (синтаксис). Необхідно вказати який саме тип введення використовується (командний рядок, стандартні форми тощо) та обґрунтувати його вибір.

Підсистема оброблення даних та генерації результатів є ядро СППР, яке виконує оброблення коректних запитів користувача; звертається до БДі знань за необхідними алгоритмами, правилами, критеріями та даними, виконує оброблення даних і передає результат в систему подання разом із інформацією щодо форми подання результатів.

База БЗД передбачає певну структурну організацію даних та можливі алгоритми оброблення цих даних.

Підсистема ППР враховує режими запитів і форму подання оброблених даних.

1.1.4 Характеристика процесів прийняття рішень

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

Рішенням вважають обґрунтований набір дій з боку особи, що приймає рішення (ОПР). Ці дії спрямовані на об'єкт чи систему управління, а набір дій надає можливість привести даний об'єкт чи систему до бажаного стану або досягнути поставленої мети [34, 65]. Рішення є одним із видів розумової діяльності і проявом волі людини.

1.1.4.1 Основні ознаки рішення

Характерними ознаками рішення є:

- можливість вибору з набору альтернативних варіантів, оскільки за відсутності альтернатив, відсутній і вибір, а тому відсутнє й рішення;
- наявність мети, оскільки вибір без цілі не розглядають як рішення;
- необхідність вольового акту ОПР при виборі рішення, оскільки ОПР формує рішення в умовах боротьби мотивів і думок.

1.1.4.2 Класифікація рішень

У проектуванні СППР важливою є задача класифікації рішень. За існуючими підходами можна скористатись класифікацією рішень, яка наведена у табл. 1.1 [25].

Прийняття рішення – це процес вибору найбільш преференційного рішення з множини припустимих рішень або упорядкування множини рішень [34]. Прийняття рішень можливо 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. Тому необхідні адекватна модель об'єкту і модель прийняття та оцінювання прийнятого рішення.

Таблиця 1.1 – Класифікація видів рішень

Ознака	Вид рішення		
	Гарно структуроване	Погано структуроване	Не структуроване
Ступінь структуризації проблеми			
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

1.1.4.3 Моделі прийняття рішень

Під моделлю прийняття рішень розуміють формальне подання поставленої задачі та процесу прийняття рішень.

Питання про формальну основу вибору, зокрема, про походження критерію оптимальності складає одну з фундаментальних проблем теорії

прийняття рішень (ТПР), що зародилася ще у XVIII ст. [34]. У ТПР були досліджені задачі опису і аналізу типів вибору, та таких теоретичних конструкцій, якими є корисність, перевага тощо.

Наукові засади ТПР були закладено у період Другої світової війни. Фундаментальні основи ТПР закладені у роботі з теорії ігор [34] Дж. Фон Неймана і О. Моргенштерна. До середини двадцятого століття оптимізаційний вибір за одним або кількома критеріями базувався на бінарних відношеннях переваг [34, 65].

В основу сучасних моделей покладено припущення стосовно того, яким чином здійснює вибір варіантів індивідуум, і яким чином здійснюється вибір рішення колективом [19, 34].

1.1.5 Характеристика етапів прийняття рішень

1.1.5.1 Основні етапи прийняття рішення

Процес прийняття рішення здійснюється у кілька основних етапів [34, 25]: постановки задачі, формування рішень, вибору рішення.

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 цьому етапі відбувається визначення обмежень, що дозволяють відокремити прийнятні варіанти рішень від неприйнятних. Формуються критерії, які сприяють вибору кращих з придатних до розгляду варіантів рішень. 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. Зазвичай, цінність альтернативних варіантів не однакова, але за умов неявної переваги одного варіанту перед іншим можуть виникати певні складності.

1.1.5.2 Послідовність прийняття рішення

Процес прийняття рішення складається з таких кроків:

- 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.

1.1.5.3 Помилки прийняття рішення

Часто в процесі прийняття рішень ОПР припускають помилки. До найбільш поширених типових належать такі помилки [19, 40, 47]:

- прийняття так званих односторонніх рішень;
- відсутнім є системний підхід у прийнятті рішення;
- під час вибору варіантів перевагу надають «звичній» альтернативі;
- розглядають лише позитивні варіанти, а можливий ризик не враховують;
- приймають рішення, керуючись емоціями;
- рішення приймають імпульсивно;
- рішення приймають поспіхом;
- у прийнятті рішення керуються припущеннями, прихованими бажаннями і хибними передумовами, а не достовірною об'єктивною інформацією;
- у рішенні неправильно тлумачать наявні факти;
- приймають неактуальне рішення (рішення є неправильним, невчасна або несвоєчасна реалізація прийнятого рішення, що, на жаль, притаманне сучасній українській економіці тощо) [47].

1.1.6 Ефективність і фактори прийняття рішень

Необхідно зазначити, що будь-яке рішення має сенс лише тоді, коли воно є ефективним. Виділяють два основних фактори, що впливають на ефективність рішень E : фактор якості рішення Q та фактор прийняття рішення людиною A .

Фактори Q і A можуть мати чисельні значення і пов'язані між собою аналітичними відношеннями (наприклад, найпростішими адитивними або мультиплікативними). Часто ефективність рішення E може бути формалізовано за формулою мультиплікативного об'єднання Q і A

$$E = Q \cdot A.$$

За умови, коли один із зазначених факторів зменшується (зокрема, прямує до мінімуму), ефективність рішення спадає.

Фактор якості рішення Q пов'язують із вибором кращої альтернативи з тих, які зумовлює проблемна ситуація. Проблемна ситуація може враховувати умови прийняття рішень та можливості виконавців рішення. Фактор прийняття рішення людиною A може мати кілька рівнів (градацій), які є прийнятними для окремих альтернатив.

Для підвищення ефективності рішення основні зусилля варто спрямовувати на покращення фактора якості, а саме, на вірний добір обмежень і критеріїв рішення, правильне формування множини допустимих альтернатив та на коректний вибір найкращого, за умов задачі, варіанта.

Наприклад, ефективність розв'язання задачі розподілу ресурсів характеризує ступінь співвимірності досягнутих цілей із витратами ресурсів на їх досягнення та визначається фактором якості рішення Q . Фактор Q значною мірою обумовлюється якістю і глибиною виконання етапу постановки задачі та вибором методів і моделей для розв'язання задачі.

1.1.7 Умови прийняття рішень

Суттєвий вплив на розв'язання задач прийняття рішень зумовлюють умови та середовище, в яких відбувається прийняття рішень. В сучасній теорії ТПР класифікують такі умови прийняття рішень [34, 44, 45]: визначеність, ризик, невизначеність, умови, зумовлені середовищем прийняття рішень.

Рішення приймається в умовах визначеності, якщо точно відомо про результат кожного з альтернативних варіантів вибору рішень. Таких рішень при управлінні бізнес-процесами відносно небагато. Такі умови прийняття рішень характерні для ситуацій, які зустрічались у минулому.

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, якщо 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.

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 для обставин, що швидко змінюються.

Середовище, в якому відбувається прийняття рішення 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.

Більшість рішень у сучасних складних задачах приймаються людиною одноособово або колегіально в умовах наявності невизначеностей різної природи та типів.

1.2 Невизначеність прийняття рішень

1.2.1 Рівні невизначеності

Невизначеність припускає наявність факторів, за якими результати дій не є детермінованими, а ступінь можливого впливу цих факторів на результати є невідомим.

Аналіз умов наявності невизначеностей може виконуватись на абстрактному теоретичному рівні або на практичному рівні, виходячи з певної точки зору прийняття рішень для конкретної ситуації. Наприклад, на абстрактному теоретичному рівні застосовують різноманітні математичні моделі, а на практичному рівні проводять оцінку кількості інформації вибору рішення. Вибір таких моделей враховує можливість їх побудови за конкретних умов. Оцінку кількості інформації вибору рішення можна проводити на основі інформаційної ентропії для характеристики невизначеності ситуації вибору.

Категорія невизначеності характеризується деякими змінними параметрами, які описують різні види невизначеностей: глобальну невизначеність, ситуативну, політичну, соціальну тощо. Для розв'язання задач прийняття рішень в умовах наявності невизначеностей необхідно встановити рівень аналізу і типи невизначеностей, що розглядаються.

Необхідно зазначити, що часто невизначеність ототожнюють лише з відсутністю повної інформації про той чи інший об'єкт. Насправді недостатні знання станів об'єкту – це не єдина невизначеність. Поряд з цим іноді можна розглядати невизначеність цілей та невизначеність критеріїв вибору рішень.

У багатьох реальних задачах складність прийняття рішення визначається насамперед кількістю альтернативних варіантів та кількістю і різноманітністю критеріїв оцінювання цих варіантів.

1.2.2 Типи невизначеностей

При розв'язанні задач системного аналізу, прийняття рішень та дослідження операцій виділяють такі основні типи невизначеностей [22]:

- невизначеність цілей;
- ситуаційна невизначеність;
- стратегічна невизначеність;
- інформаційна невизначеність.

1.2.2.1 Характеристика основних типів невизначеностей

Розглянемо характеристики основних типів невизначеностей у прийнятті рішень.

Невизначеність цілей – це невизначеність вибору цілей в багатокритеріальних задачах.

Ситуаційна невизначеність – це невизначеність впливу неконтрольованих факторів, що позначаються на процесах практичної діяльності. Зокрема, невизначеність може бути зумовлена:

- невизначеністю природи за відсутності достатніх знань про оточення та зовнішні фактори;
- ненадійністю очікувань, наприклад, як невизначеність розвитку певних подій у майбутньому.

Стратегічна невизначеність – це невизначеність цілей і дій активного або пасивного партнера чи противника (так звана невизначеність конфліктів).

Інформаційна невизначеність – це нечіткість та розпливчастість процесів і явищ та інформації про досліджувану систему, відсутність відомостей про достовірність інформації.

1.2.2.2 Додаткові типи невизначеностей

Додатково виділяють такі типи невизначеностей:

- структурна невизначеність – невизначеність структури моделі досліджуваної системи;
- параметрична невизначеність – це апіорна невизначеність параметрів моделі системи, складність оцінювання і аналізу якості параметрів моделі;

- статистична невизначеність – невизначеність статистичних даних, що переважно впливає з об'єму даних, наявності пропусків, імпульсних викидів тощо. Сюди можна віднести, також, невизначеності, зумовлені наявністю збурюючих впливів та похибок (шуму) вимірів;
- методична невизначеність – невизначеність (неоднозначність), притаманна методу оброблення даних чи методу розв'язання задачі;
- комбінаторна невизначеність – неможливість знання всіх можливих варіантів. Комбінаторна невизначеність пов'язана із усіма іншими типами невизначеностей і найчастіше впливає з них.

Необхідно зазначити, що в реальних практичних задачах прийняття рішень і системного аналізу часто наявними є різноманітні види невизначеностей, які разом утворюють деякий комплекс невизначеностей, і називають системною невизначеністю [22].

1.3 Людино-машинні системи підтримки прийняття рішень

Слід враховувати, що процес прийняття рішень людиною має певні обмеження стосовно можливості аналізу, оброблення 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 [22, 41].

Людина має можливість одночасно оперувати лише обмеженим числом операндів і понять, щонайбільше 7 ± 2 [22, 41]. Крім того, при аналізі і розв'язанні багатокритеріальних задач особи ОПР досить часто проявляють мінливість, невпевненість, нелогічність, намагання суттєво

спростити задачу. У таких випадках можливості обчислювальних машин значно перевищують можливості людини. Така ситуація призвела до створення напряму розробки систем та методологій, які мають можливість об'єднати переваги людини і комп'ютера та компенсувати їх недоліки, тобто людино-машинних систем [17, 44].

1.3.1 Основні напрями створення людино-машинних систем

Серед сучасних напрямів розробки людино-машинних систем – системи автоматичного керування, експертні системи та СППР. Найбільш придатними для розв'язання багатьох задач, зокрема задачі розподілу ресурсів, виявляються СППР [3, 15, 35]. Саме за допомогою СППР ОПР має можливість безпосередньо за допомогою обчислювальних засобів проектувати, порівнювати та обирати альтернативні варіанти рішень у різноманітні способи.

У сучасних умовах застосування нових альтернативних підходів до формування і прийняття високоякісних рішень неможливо уявити без використання електронних інформаційних систем, інформаційних технологій [3, 24, 43]. Причинами широкого впровадження інформаційних технологій є їх зростаюча роль практично у кожній галузі діяльності суспільства та зростаючі потенційні можливості цих технологій. Застосування сучасних підходів на основі інформаційних технологій надає можливість використовувати обчислювальні потужності комп'ютерів для виконання розрахунків, оброблення, аналізу і прогнозування даних в режимі реального часу, для допомоги у прийнятті рішень.

За цих умов відбувається автоматизація не стільки ручної праці, скільки інтелектуальної, тобто людино-машинні системи дедалі стають

інтелектуальними системами підтримки прийняття рішень. Такі системи для багатьох практичних задач виявляють себе як більш ефективні у порівнянні з іншими.

1.3.2 Людино-машинні системи прийняття рішень з передбаченням

У багатьох випадках автоматизації прийняття рішень доцільно використовувати СППР, які дозволяють приймати важливі рішення, керуючись подіями, які ще не здійснилися. Такі СППР уможливають розробку декількох можливих сценаріїв типу «що було б, якби», визначати оптимальні дії тощо.

СППР інтегрують в собі такі якості, які роблять їх не тільки дуже корисними для системних задач управління і прийняття рішень, але й по суті незамінними інструментами аналізу даних в сучасних умовах економічного розвитку [35, 40, 43].

Системи СППР створюються для підтримки прийняття рішень ОПР в складних та слабоструктурованих ситуаціях. Існує ряд напрямків діяльності стосовно процесу прийняття рішень. Тому розрізняють такі види СППР: аналіз рішень (Decision Analysis); обчислення (визначення) рішень (Decision Calculus); дослідження рішень (Decision Research) та процес впровадження (Implementation Process). Кожний із зазначених напрямів постає самостійною перспективою розвитку СППР, проте в «чистому вигляді» вони практично не зустрічаються [35].

1.4 Автономні системи прийняття рішень

Слід зазначити, що для процесу прийняття рішень, який здійснює особа ОПР без допоміжних засобів, суттєве значення мають такі важливі

чинники, які повинні бути досліджені і враховані при розробці структури СППР і реалізації системи [35, 54]. Розглянемо основні з них.

Робоча пам'ять.

Особа ОПР спроможна для оброблення інформації оперувати з даними до восьми понять. Для цього ОПР застосовує деяку робочу пам'ять, як проміжну між короткостроковою і довгостроковою пам'яттю. Ці поняття варто утримувати без відновлення 7-13 секунд.

Джерела інформації.

ОПР має можливість отримувати інформацію від органів чуття та із довгострокової пам'яті. Проте інформація з довгострокової пам'яті вважається менш надійною. По-перше, з часом вона може стиратись, а по-друге, людина схильна застосовувати ту інформацію, яка повторюється, та є семантично наближеною до інформації, що міститься у робочій пам'яті.

Оброблення числових даних.

Прийняття рішень часто вимагає виконання оброблення великих масивів числових даних та здійснення різноманітних обчислень. ОПР намагається уникати методологій із значними числовими обчисленнями, а за основу алгоритмів прийняття рішень вибирає операції, що ґрунтуються на якісних та евристичних механізмах мислення.

Виконання операцій.

Реалізація складних процесів мислення і оброблення інформаційних елементів може вимагати значного часу від ОПР. Це відіграє важливу роль для систем, які функціонують у масштабі реального часу. ОПР не завжди здатна сформулювати прийнятне рішення за короткий проміжок часу.

Зазначені аспекти прийняття рішень постають певними загальними обмеженнями у прийнятті рішень особи ОПР без допоміжних засобів таких, зокрема без СППР. Саме такі обмеження сприяли появі і розвитку наряду створення людино-машинних систем.

1.5 Вимоги до сучасних систем підтримки прийняття рішень

За сучасним станом розвитку система СППР має відповідати таким вимогам [32, 54]:

- використовує слабоструктуровані та нечіткі дані;
- оперує зі слабоструктурованими рішеннями;
- підтримує як взаємозалежні, так і послідовні рішення;
- може застосовувати знання;
- підтримує моделювання та прогнозування;
- може бути легко побудована, якщо може бути сформульована логіка конструкції СППР;
- повинна бути простою у застосуванні та модифікації;
- повинна підтримувати три фази процесу прийняття рішень: інтелектуальну частину, проектування та вибір;
- система призначена для ОПР різного рівня;
- система може бути адаптована до індивідуального та групового застосування;
- СППР підтримує різні стилі та методи рішень, що можуть бути корисними при застосуванні групою ОПР;
- система виявляє гнучкість і адаптується до змін в організації та в її оточенні;
- СППР дозволяє людині керувати процесом прийняття рішення за допомогою комп'ютера, а не навпаки;
- система підтримує еволюційне застосування та легко адаптується до мінливих вимог;
- застосування СППР підвищує ефективність процесу прийняття рішень.

1.6 Особливості застосування сучасних СППР

Інтерес до створення і застосування СППР, як до перспективної галузі використання обчислювальної техніки та інструментарію підвищення ефективності праці у сфері управління економікою, постійно зростає. У багатьох країнах розробка та реалізація СППР перетворилася на бізнес, що швидко розвивається. СППР набувають широкого застосування в економіках передових країн світу, з одночасним постійним збільшенням їх кількості. Суттєвого значення набувають розширення функціональних можливостей і удосконалення самих СППР.

1.6.1 Сфери застосування СППР у мікро і макроекономіці

За сферою застосування СППР виділяють такі рівні керування: стратегічного керування, орієнтовані на операційне керування СППР, СППР підтримки інвестиційної діяльності і фінансування,

На рівні стратегічного керування використовують ряд СППР, окремо для довго-, середнє- і короткострокового керування. Системи СППР також застосовують для фінансового планування, включно із системою розподілу капіталовкладень та ресурсів.

Орієнтовані на операційне керування СППР застосовують у галузях маркетингу (прогнозування й аналіз збуту, дослідження ринку і цін), у науково-дослідних і конструкторських роботах, у керуванні кадрами. Операційно-інформаційне застосування СППР пов'язано з виробництвом, придбанням і обліком товарно-матеріальних запасів, їхнім фізичним розподілом і бухгалтерським обліком.

У країнах з розвинутою ринковою економікою значну увагу приділяють розвитку систем підтримки інвестиційної діяльності [35, 43], а також розподілу інвестиційних і фінансових ресурсів.

1.6.2 Системи СППР розподілу інвестиційних і фінансових ресурсів

Прикладом такої СППР може бути розроблена у США ще на початку 1980х років система ISDS (Investment Strategy Decision System – система для підтримки рішень з інвестиційних стратегій) [35].

Система ISDS призначена для формування портфеля замовлень і забезпечує виконання таких операцій:

- попередній добір пропозицій;
- порівняльна оцінка нових пропозицій між собою, а також із вже розпочатими роботами;
- об'єднання відібраних пропозицій і виконуваних робіт в інвестиційні групи, кожна з яких формується згідно з програмними цілями, політикою й бюджетними обмеженнями;
- порівняльний аналіз розподілу довгострокових капіталовкладень;
- подання підсумкових даних з різних трендів зміни капіталовкладень;
- видача статистичної інформації, необхідної для звітності.

Крім цього, у зазначеній СППР передбачено накопичення досвіду практичного використання системи, що допомагає враховувати колишні результати при формуванні варіантів планів довгострокових капітальних вкладень. Це дозволяє перевіряти правильність рішень в історичній перспективі, порівнюючи їх з аналогічними ситуаціями у минулому.

Застосування СППР в інвестиційних процесах для розв'язання задач розподілу інвестиційних ресурсів надає можливість враховувати вплив значної кількості різноманітних факторів, достатньо швидко отримувати

результат та дозволяє виконувати імітаційне моделювання інвестиційних рішень при різних значеннях параметрів та для різних умов задачі. Всі ці операції дуже складно та, найчастіше, неможливо виконувати без застосування спеціальних засобів, таких як СППР. Роль СППР в інвестиційній діяльності не в тому, щоб замінити ОПР, а в тому щоб значно підвищити їх ефективність.

1.6.3 Особливості інтелектуальних засобів підтримки прийняття рішень

Аналіз практичних аспектів реалізації СППР свідчить, що останнім часом актуальною постає розробка інтелектуальних засобів підтримки прийняття рішень, що полягає у створенні СППР на базі методів математичного моделювання і сучасних методів штучного інтелекту спільно із апаратом експертних систем.

Дружні до людини системи СППР надають можливість вести рівноправний діалог із ЕОМ за допомогою звичайних мов спілкування. Системи можна підлаштовувати під стиль мислення користувача, з урахуванням рівня його знань і фахової підготовки, а також під зручні для нього засоби роботи.

Для сучасних СППР характерними є такі особливості застосування.

1. СППР надає керівнику допомогу у процесі прийняття рішень і забезпечує підтримку у всьому діапазоні контекстів задач. Думка людини та інформація, що генерує ЕОМ, становлять єдине ціле для прийняття рішень.

2. СППР підтримує і посилює (але не змінює і не відміняє) міркування та оцінку керівника. Контроль залишається за людиною.

Для користувача СППР створює умови «почувати себе комфортно» і «як удома».

3. СППР підвищує ефективність прийняття рішень. На відміну від адміністративних систем, акцентованих на аналітичному процесі, у випадку СППР найважливішою характеристикою є ефективність процесу прийняття рішень.

4. СППР виконує інтеграцію моделей і аналітичних методів із стандартним доступом до даних та вибіркою з них. Для надання допомоги при прийнятті рішень активується одна або декілька моделей. Вміст БД охоплює історію поточних і попередніх операцій, а також інформацію зовнішнього характеру та інформацію про середовище.

5. Система СППР є простою в роботі для осіб, що мають досвід роботи з комп'ютерами. Системи є дружніми для користувачів і не потребують глибоких знань про обчислювальну техніку, забезпечують просте керування обчислювальними і інформаційними ресурсами системи.

6. Сучасні системи СППР побудовані за принципом інтерактивного розв'язання задач. Користувач має можливість підтримувати діалог із СППР у неперервному режимі.

7. Система СППР є орієнтованою на гнучкість і адаптивність для пристосування до змін середовища або модифікації підходів до розв'язання задач, які обирає користувач. Керівник повинен пристосовуватися до змінюваних умов сам і відповідно підготувати систему.

8. Зазвичай система СППР не повинна нав'язувати користувачу визначеного процесу прийняття рішень, а надає інтелектуальну підтримку у прийнятті правильних рішень.

1.7 Класифікація СППР

1.7.1 Підходи до класифікації СППР

Єдиної класифікації СППР зараз не існує. Відома для свого часу класифікація Альтера [48], яка поділяла СППР на сім класів є дещо застарілою. Розглянемо деякі основні підходи до поділу 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;
- кооперативна – дозволяє ОПР змінювати, поповнювати або поліпшувати рішення, пропоновані системою, надсилаючи потім ці зміни в систему для перевірки.

На технічному рівні розрізняють такі СППР [35, 63]:

- 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 [63]:

- 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 оброблення даних та відповідають властивостям декількох видів СППР.

Залежно від даних, з якими ці системи працюють, СППР умовно можна поділити на оперативні та стратегічні.

1.7.2 Оперативні і стратегічні СППР

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

Такі СППР називають Виконавчі Інформаційні Системи (Executive Information Systems). Зазвичай системи надають кінцеві множини звітів, побудовані за даними із транзакційної інформаційної системи оперативного обліку підприємства. Вони забезпечують адекватне відображення в режимі реального часу основних аспектів виробничої і фінансової діяльності підприємства.

Для оперативних СППР характерними є такі риси:

- звіти ґрунтуються на стандартних, для організації, запитах, кількість яких відносно невелика;

- СППР подає звіти у максимально зручному вигляді, що включає надання, окрім таблиць, ділової графіки, мультимедійної інформації тощо;
- СППР зазвичай орієнтовані на конкретну сферу, наприклад, фінанси, маркетинг, керування ресурсами.

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

Такі СППР припускають достатньо глибоке спеціальне перетворення даних для їх зручного використання у процесі прийняття рішень. Невід'ємним компонентом цих СППР є правила прийняття рішень, які на основі агрегованих даних надають можливість менеджерам компанії обґрунтовувати свої рішення, використовувати фактори стійкого росту бізнесу компанії і знижувати ризики. Стратегічні СППР будують на принципах багатовимірного подання та аналізу даних у системах OLAP (Online Analytical Processing).

1.8 Історія виникнення та розвитку інформаційних СППР

1.8.1 Напрями застосування засобів обчислювальної техніки

Інформаційні системи СППР на кожному етапі свого розвитку були тісно пов'язані з досягненнями обчислювальної техніки та інформатики. З самого початку розвитку обчислювальної техніки утворилися два основних напрями її використання у СППР: для виконання досить складних

чисельних розрахунків з обробки даних, для створення автоматичних та автоматизованих інформаційних систем (ІС) даних великого об'єму.

На практиці іноді дуже важко провести між ними грань, оскільки для обчислювальної техніки та інформатики характерними є прискорені темпи розвитку (за продуктивністю, енергоспоживанням, об'ємами оперативної та зовнішньої пам'яті, можливостями програмного забезпечення тощо).

Перший напрям застосування обчислювальної техніки для виконання чисельних розрахунків СППР дозволяє виконувати обробку даних, яку занадто довго або взагалі неможливо робити вручну. Становлення цього напрямку сприяло інтенсифікації методів чисельного розв'язання складних математичних задач, розвитку класу мов програмування, орієнтованих на зручний запис чисельних алгоритмів, встановленню зворотного зв'язку із розробниками нових архітектур комп'ютерів.

Другий напрям пов'язано з використанням засобів обчислювальної техніки для створення автоматичних та автоматизованих інформаційних систем. У самому широкому розумінні ІС є апаратно-програмний комплекс, функції якого полягають у підтримці надійного збереження інформації в пам'яті комп'ютера, виконанні специфічних для даного додатка перетворень інформації та/або проведенні обчислень, наданні користувачам зручного і легкого для засвоєння інтерфейсу. Зазвичай обсяги інформації в таких системах досить великі, а сама інформація має досить складну структуру. Класичними прикладами інформаційних систем є банківські системи, системи резервування авіаційних або залізничних квитків, місць у готелях, бортові системи літальних апаратів тощо.

Слід зазначити, що другий напрям виник пізніше, ніж перший. Це зв'язано з тим, що у початковий період розвитку обчислювальної техніки комп'ютери мали обмежені можливості стосовно об'єму пам'яті (оперативної та зовнішньої), швидкодії, доступу до ресурсів, надійного та довгострокового зберігання даних у запам'ятовувальних пристроях, що зберігають дані після вимкнення електричного живлення.

1.8.2 Зовнішня пам'ять

Спочатку застосовувались два основних види пристроїв зовнішньої пам'яті: магнітні стрічки і барабани. Ємність магнітних стрічок була порівняно великою, але за своєю фізичною природою вони забезпечували лише послідовний доступ до даних. Магнітні ж барабани (вони більше всього схожі на сучасні магнітні диски з фіксованими голівками) давали можливість довільного доступу до даних, але мали обмежену ємність.

Зазначені обмеження не дуже суттєві для простих чисельних розрахунків. Навіть якщо програма повинна обробити (або створити) великий обсяг даних, під час програмування можна продумати розташування цієї інформації в зовнішній пам'яті таким чином, щоб програма працювала якнайшвидше.

Для інформаційних систем, у яких необхідність в оперативних даних визначається користувачем, наявність тільки магнітних стрічок і барабанів не задовольняла потребам. Однією із природних вимог до таких систем є середня швидкість виконання операцій.

Функціональні вимоги до обчислювальної техніки з боку прикладних задач зумовили появу накопичувачів на магнітних дисках з рухомими голівками. Ці пристрої зовнішньої пам'яті мали істотно більшу ємність ніж магнітні барабани, забезпечували прийнятну швидкість доступу до даних у

режимі довільної вибірки, а можливість зміни дискового пакета на пристрої дозволяла мати практично необмежений архів даних.

З появою магнітних дисків розпочалася історія систем керування даними в зовнішній пам'яті. До цього кожна прикладна програма, якій було потрібно зберігати дані в зовнішній пам'яті, сама визначала розташування кожної порції даних на магнітній стрічці або барабані і виконувала обміни між оперативною і зовнішньою пам'яттю за допомогою програмно-апаратних засобів низького рівня (машинних команд або викликів відповідних програм операційної системи). Такий режим роботи не дозволяє або дуже ускладнює підтримку на одному зовнішньому носії кількох архівів довгочасного зберігання інформації. Окрім цього, прикладні програми вирішували проблеми іменування частин даних і структурування даних у зовнішній пам'яті.

Історичним кроком став перехід до використання централізованих систем керування файлами, файлових систем. З погляду прикладної програми файл – це іменована область зовнішньої пам'яті, у яку можна записувати і з якої можна зчитувати дані. Система керування файлами бере на себе розподіл зовнішньої пам'яті, відображення імен файлів у відповідні адреси в зовнішній пам'яті і забезпечує доступ до даних.

1.8.3 Інформаційні системи і бази даних

Інформаційні системи головним чином орієнтовані на збереження, вибір і модифікацію постійно існуючої інформації. Структура інформації найчастіше дуже складна і хоча структури даних різні у різних інформаційних системах, між ними часто буває багато спільного. На початковому етапі використання обчислювальної техніки для керування інформацією проблеми структурування даних вирішувалися

індивідуально для кожної інформаційної системи. Згодом почали створювати необхідні надбудови над файловими системами.

Але оскільки інформаційні системи вимагають складних структур даних, ці додаткові індивідуальні засоби керування даними були суттєвою частиною інформаційних систем і практично повторювалися для різних системи.

Прагнення виділити й узагальнити загальну частину інформаційних систем, відповідальну за керування складними структурованими даними, спонукало до створення систем управління базами даних (СУБД). Дуже скоро стало зрозуміло, що неможливо обійтися загальною бібліотекою програм, що реалізує над стандартною базовою файловою системою більш складні методи збереження даних.

Поява СУБД призвела до виникнення і розвитку великих інформаційних систем. А поява великих інформаційних систем та стрімкий розвиток обчислювальних можливостей комп'ютерів, значний розвиток теорії прийняття рішень після Другої світової війни та нові практичні задачі щодо прийняття рішень в складних умовах спричинили появу нових систем – систем підтримки прийняття рішень

1.8.4 Хронологія розвитку інформаційних СППР

До середини 60-х років ХХ століття створення інформаційних систем було надзвичайно дорогим, тому перші ІС менеджменту (так звані Management Information Systems — MIS) були створені в ці роки лише в досить великих компаніях. MIS призначалися для підготовки періодичних структурованих звітів для менеджерів.

Наприкінці 60-х років з'являється новий тип ІС -- модельно-орієнтовані СППР (Model-oriented Decision Support

Systems — Model-oriented DSS) або системи управлінських рішень (Management Decision Systems — MDS).

На думку розробників СППР Кіна та Скота Мортона [63] (1978), концепція підтримки рішень була розвинена на основі «теоретичних досліджень в області прийняття рішень...і технічних робіт із створення інтерактивних комп'ютерних систем».

У 1971 р. опублікована книга Скота Мортона [64], у якій були уперше описані результати впровадження СППР, заснованої на використанні математичних моделей.

У 1974 р. дано визначення ІС менеджменту — MIS (Management Information System): «MIS – це інтегрована людино-машинна система забезпечення інформацією, що підтримує функції операцій, менеджменту і прийняття рішень в організації. Системи використовують комп'ютерну техніку і програмне забезпечення, моделі керування і прийняття рішень, а також базу даних» [63].

У 1975 р. Літл у роботі [59] запропонував критерії проектування СППР у менеджменті.

У 1978 р. опубліковано підручник з СППР [55], у якому вичерпно описано аспекти створення СППР: аналіз, проектування, впровадження, оцінювання характеристик і розробка.

У 1980 р. опублікована дисертація Альтера (S. Alter) [48], у якій він дав основи класифікації СППР.

У 1981 р. Бонзек, Холсепл та Уінстон у книзі [50] створили теоретичні основи проектування СППР. Вони виділили 4 необхідних компоненти, властивих усім СППР:

1. Мовна система (Language System — LS). СППР може приймати всі повідомлення.

2. Система подання (Presentation System (PS)). СППР може видавати свої повідомлення.
3. Система знань (Knowledge System — KS). Усі знання СППР зберігає.
4. Система розв'язання задач (Problem-Processing System (PPS)). Програмний «механізм», що намагається розпізнати і розв'язати задачу під час функціонування СППР.

У 1981 р. у книзі [66] Спраг та Карлсон описали, яким чином на практиці можна побудувати СППР. Тоді ж було розроблено інформаційну систему керівника (Executive Information System (EIS)). Система EIS є комп'ютерна система, призначена для забезпечення поточною адекватною інформацією менеджера з метою підтримки прийняття управлінських рішень. Починаючи з 1990-х років розробляються так звані Data Warehouses, тобто сховища даних великого об'єму.

1.9 Системи OLAP

У 1993 р. Е. Кодд запропонував термін OLAP (On-Line Analytical Processing) – оперативний аналіз даних, аналітичне оброблення даних в реальному часі для систем підтримки прийняття важливих рішень. Вихідні дані для аналізу подають у вигляді багатовимірної куба, за яким можна одержувати потрібні розрізи — звіти. Виконання операцій над даними здійснюється OLAP-машиною.

За способом збереження даних розрізняють моделі оброблення MOLAP, ROLAP і HOLAP. За місцем розміщення OLAP-машини розрізняють OLAP-клієнти і OLAP-сервери.

OLAP-клієнт робить побудову багатовимірного куба й обчислення на клієнтському комп'ютері, а OLAP-сервер отримує запит, обчислює і зберігає агрегатні дані на сервері, видає результати.

На початку XXI сторіччя було створено СППР на основі Web-технологій.

1.10 Покоління систем підтримки прийняття рішень

1.10.1 Покоління СППР

Аналіз еволюції систем СППР дає можливість виділити два покоління СППР:

- СППР першого покоління, які розроблялись у період з 1970 по 1980 р;
- СППР другого покоління, які започатковано у 1980 р. і існують дотепер.

Ці покоління можуть суттєво розрізнятись за складом компонентів, засобами керування тощо.

СППР першого покоління майже цілком повторювали функції звичайних управлінських систем по відношенню до комп'ютеризованої допомоги у прийнятті рішень. Основні компоненти СППР мали такі ознаки:

- керування даними – велика кількість інформації, внутрішні і зовнішні банки даних, оброблення та оцінювання даних;
- керування обчисленням і моделюванням – моделі, розроблені спеціалістами в галузі інформатики для спеціальних проблем;
- інтерфейси користувача (мова спілкування) – мови програмування, розроблені для великих комп'ютерів, що використовуються винятково програмістами.

СППР другого покоління вже мають принципово нові ознаки:

- керування даними – необхідна і достатня кількість інформації про факти згідно з прийняттям рішень, що охоплюють приховані припущення, інтереси і якісні оцінки;
- керування обчисленням і моделюванням – гнучкі моделі, що відображають засіб мислення особи, приймаючої рішення, у процесі прийняття рішень;
- інтерфейс користувача – програмні засоби, дружні користувачу; звична мова, безпосередня робота кінцевого користувача із системою.

Цілі та призначення СППР другого покоління можна визначити таким чином:

- допомога у розумінні розв'язуваної проблеми: структуризація проблеми, генерування постановок задач, визначення переваг, формування критеріїв;
- допомога у розв'язанні задач: генерування і вибір моделей і методів, збір і підготовка даних, виконання обчислень, оформлення і видача результатів;
- допомога у проведенні аналізу типу «що, якщо» тощо, пояснення ходу рішення; пошук і видача аналогічних рішень у минулому і їхні результати.

1.10.2 СППР на основі GRID-систем

СППР на основі GRID-систем та систем хмарних обчислень є сучасним підходом до створення розподілених систем.

Світова GRID-архітектура включає ряд національних GRID-архітектур, які орієнтовані на концентрацію інформаційно-обчислювальних ресурсів у комп'ютерних кластерах, суперкомп'ютерах і інших глобальних середовищах ресурсів у межах країни (NGI — National Grid Initiative) і забезпечують ефективну віддалену взаємодію цих GRID-архітектур між собою для розв'язання складних задач.

Сучасне програмне забезпечення GRID-архітектур (ПЗГА) розробляють з використанням взаємоузгоджених наборів засобів обслуговування Grid-інфраструктур – платформ. Залежно від конкретної Grid-архітектури, використовують різні платформи: ARC(NorduGrid), Alien, LCG, DataGrid (www.datagrid.org), Unicore (www.unicore.org), gLite (www.glite.org), які ґрунтуються на одній базі — системі Globus Toolkit (www.globus.org).

Платформа gLite дозволяє підтримувати такі сервіси: аутентифікації і авторизації; брокера; моніторингу і діагностики; реплікації даних і програм; пошуку програмного забезпечення; каталогів даних і додатків; менеджменту ресурсів і завдань; розв'язання задачі; обліку і оплати отриманих послуг. Доступ користувача до сервісів відбувається через Grid – портал і служби доступу, які дозволяють вирішити основні проблеми ідентифікації, аутентифікації і авторизації до ресурсів Grid-архітектури.

Основними класами задач, для яких Grid-технології найбільш ефективні, є: масове оброблення потоків даних великого об'єму; багатопараметричний аналіз даних; моделювання на віддалених суперкомп'ютерах; реалістична візуалізація великих наборів даних; складні бізнес-додатки з великими об'ємами обчислень.

Існує п'ять основних груп Grid-додатків: розподілені супер-обчислення, обчислення з високою пропускнуою спроможністю та запитом, з інтенсивним обміном даними і сумісні додатки.

Найбільшою в світі Grid-інфраструктурою є європейська EGEE (Enabling Grids for E-science), яка включала у 2008 році 240 вузлів у 45 країнах, в яких задіяні 41000 процесорів і 5 Пб (PetaBytes) пам'яті. Мережа обслуговує більше 10000 споживачів і 150 віртуальних організацій з продуктивністю більше 100000 обчислювальних завдань за день. Основними є додатки з археології, астрономії, астрофізики, захисту довкілля, комп'ютерної хімії, науки про Землю, фінансів, фізики плазми, геофізики, фізики високих енергій, науки про життя, мультимедіа, матеріалознавства.

Розвитком EGEE стали EGEE-II, EGEE-III. Зокрема EGEE-II складається з мережних напрямів робіт NA (Networking Activities), служб SA (Service Activities) і напрямів сумісних досліджень JRA (Joint Research Activities). Відповідно до сучасного стану Grid-технологій в EGEE-II частка фінансування SA і NA підвищиться, а JRA -- знизиться. Це дозволить включити до інфраструктури проекту нові країни, додатки і сайти, а також у більшому масштабі поширювати інформацію, вести навчання і підтримувати додатки. Скоротиться розробка програмного забезпечення, оскільки його можна буде інтегрувати з інших проектів і джерел.

1.10.3 СППР на основі інфраструктур хмарних обчислень

Більшість інфраструктур хмарних обчислень приховують міжмережну взаємодію і включають такі рівні: додаток, клієнт, інфраструктура, платформа, служба, сховище даних.

Складовими компонентами є IaaS (Interface as a Service), PaaS (Platform as a Service), SaaS (Software as a Service). Клієнт звертається до цих сервісів і отримує їх із середовища Інтернет на час проведення обчислень. Як наслідок проблеми, пов'язані з інсталяцією, модифікацією адаптацією програмного забезпечення значно спрощуються, уніфікуються сам програмний продукт.

Інфраструктура хмарних обчислень включає, зокрема, комп'ютерні кластери, суперкомп'ютери, локальні мережі. Тому системи на основі хмарних обчислень мають фізичне апаратно-програмне забезпечення від найпростіших обчислювальних пристроїв (смартфонів, планшетів, сенсорів, GPS, лаптопів тощо) до суперкомп'ютерів, кластерів, GRID-архітектур.

У хмарних обчисленнях загальною концепцією є використання програмного забезпечення як служби (SaaS). Іноді це може призводити до відмінностей у порівнянні з Grid-обчисленнями для кластерних структур, обчислень на основі використання утилітів (зокрема, використання програмних пакетів обробки дани) і автономних обчислень для систем з самоменеджментом. Хмарні обчислення можуть бути основою соціальних обчислювальних мереж (BitTorrent, Skipe, Facebook тощо).

Запитання та завдання для самоконтролю

1. Які особливості будови і застосування владають у визначення поняття систем підтримки прийняття рішень?
2. Спробуйте окреслити вимоги до інтерактивності заданого типу СППР.
3. У чому полягають відмінності Інтелектуальної системи підтримки прийняття рішень?

4. Що необхідно врахувати для створення Інтелектуальної систем підтримки прийняття рішень?
5. Описати склад і функції основних підсистем узагальненої структури системи СППР.
6. Перерахувати характерні ознаки рішення в СППР.
7. Як класифікують рішення за ознаками і видами рішення?
8. Описати основні етапи прийняття рішення.
9. Надати власну інтерпретацію кроків послідовності прийняття рішення.
10. Дати характеристику основних помилок у прийнятті рішень та їх роль у якості прийнятого рішення.
11. Як визначають основні фактори і поняття ефективності рішення?
12. Описати основні умови прийняття рішень.
13. У чому полягає визначення категорії невизначеності прийняття рішення?
14. Дати характеристику основних типів невизначеностей прийняття рішень.
15. Як визначають додаткові типи невизначеностей у прийнятті рішень?
16. Дати оцінку основних напрямів створення людино-машинних систем для побудови СППР.
17. Що являють собою людино-машинні системи прийняття рішень з передбаченням?
18. Дати оцінку обмежень автономних систем прийняття рішень.
19. Основні вимоги до сучасних СППР.
20. Що надають сучасні СППР і якими є характерні особливості їх застосування?

21. Як класифікують СППР на рівні користувача?
22. Як класифікують СППР на технічному рівні?
23. Які типи виділяють на концептуальному рівні?
24. Які існують системи OLAP?
25. Дати оцінку переваг і недоліків оперативних і стратегічних СППР.
26. Описати покоління СППР та роль сучасних апаратно-технічних засобів.
27. Дайте визначення поняттю: система підтримки прийняття рішень.
28. Дайте визначення інтелектуальній СППР.
29. Опишіть процес прийняття рішень та розкрийте поетапну процедуру його здійснення.
30. Наведіть основні існуючі класифікації видів рішень.
31. Поясніть особливості прийняття рішень в умовах невизначеностей.
32. Які існують причини виникнення невизначеностей?
33. Охарактеризуйте важливі аспекти прийняття рішень людиною без допоміжних засобів.
34. Сформулюйте основні цілі створення та застосування СППР.
35. Наведіть основні актуальні класифікації СППР.
36. Сформулюйте та поясніть основні причини зародження та виникнення СППР.
37. Які види підтримки може надавати СППР користувачу?
38. Опишіть, які покоління стосовно створення та використання СППР виділяють та розглядають на практиці.

РОЗДІЛ 2 ОСНОВИ ПОБУДОВИ СИСТЕМ ПІДТРИМКИ ПРИЙНЯТТЯ РІШЕНЬ

Основою створення СППР є аналіз процесу прийняття рішення. Такий аналіз є підґрунтям для вибору стратегії прийняття рішення.

Реальні задачі прийняття рішень приводять до значної кількості можливих реалізацій СППР. Тому важливим є врахування досвіду аналізу реальних процесів прийняття рішень.

Результатом аналізу і вибору стратегії є формулювання основних вимог до СППР і подальша побудова СППР за існуючими підходами. Ці підходи узагальнюють як етапи проектування СППР.

2.1 Аналіз процесу прийняття рішення

2.1.1 Атрибути прийняття рішень

Аналіз процесу прийняття рішення полягає у визначенні атрибутів прийняття рішень і дослідженні процесів ухвалення рішень особою, яка приймає рішення.

Для розуміння процесу прийняття рішень необхідно знати три ключових атрибути [3, 32]: використання внутрішнього (розумового) подання проблеми (задачі), досягнення мети; сприйняття інформації.

1. Використання внутрішнього (розумового) подання проблеми (задачі). Коли людина приймає рішення, вона у дійсності не покладається або ж мало покладається на той досвід, який був накопичений раніше. У більшості випадків прийняття рішення використовують підхід, заснований на поданні або розумінні поточних даних. Особа (експерт), що приймає рішення, досить глибоко розуміє ситуацію, має ширше і більш повне уявлення про проблему, ніж інші люди.

Експерт намагається створити з наявних даних загальну картину конкретної ситуації, а тому знає, що йому необхідно одержати і як інтерпретувати нові дані. Внутрішнє подання рішень може діяти в деяких ситуаціях як своєрідний захист проти даних, що не входять у створений образ проблеми. В результаті виникнення такої ситуації можна втратити або неправильно інтерпретувати важливі дані.

2. Досягнення мети. Ухвалення рішення людиною починається, зазвичай, з аналізу бажаного результату. Процес ухвалення рішення структурується таким чином, що зв'язується з бажаним результатом, тобто створюється опис або розумове подання бажаних ситуацій або умов, які повинні мати місце в результаті ухвалення рішення. Оскільки головним елементом ухвалення рішення є поставлена кінцева мета, то процес ухвалення рішення рідко узгоджується з детермінованим комп'ютерним алгоритмом.

Процес ухвалення рішення може оперативнo модифікуватися у відповідності до надходження нової інформації або з появою альтернативних шляхів досягнення мети. Таким чином, процес ухвалення рішення людиною можна описати досить чітко і строго з погляду поставленої мети, але не з погляду процедури, яка використовується для її досягнення.

3. Сприйняття інформації. Інформація надходить до індивідуума через візуальну й аудіосистеми, але для цих каналів не існує фіксованих смуг пропускання. Швидше за все, здатність людини сприймати інформацію залежить від того, що він «уже знає». Індивідуум інтерпретує світ з погляду осмислених понять, а не з погляду інформаційних одиниць, визначених теорією інформації. Однак, смислове значення може бути різним для різних людей, тому що в кожного свій рівень освіти, досвіду, підготовки до розв'язання конкретної проблеми.

Наприклад, для неосвіченої людини результати хімічного спектрального аналізу можуть являти собою лише абстрактний малюнок із прямих ліній, а для фахівця – це чітка вказівка до визначення типу речовини, що аналізується. Принцип розпізнавання (сприйняття) взаємодіє з принципом внутрішнього подання інформації. ОПР швидше і легше сприймає інформацію, що збігається із звичним для неї внутрішнім поданням.

2.1.2 Ухвалення рішень особою прийняття рішень

На процес ухвалення рішення індивідумом, який виконує роль ОПР без допоміжних засобів, у загальному випадку впливають такі основні обмеження: особливості використання робочої пам'яті, швидкість виконання осмислених операцій, шляхи одержання інформації, методи і засоби обробки числових даних, зв'язок виконання операцій у часі і просторі. Ці обмеження мають істотне значення при проектуванні системи підтримки прийняття рішень.

Розглянемо основні особливості впливу цих обмежень на прийняття рішень:

1. Використання робочої пам'яті.

Людина виконує оброблення інформації в пам'яті, яка концептуально знаходиться між короткостроковою і довгостроковою зонами запам'ятовування людини. Цю проміжну пам'ять називають робочою пам'яттю (РП). Тобто, у прийнятті рішень можна вести оброблення тільки інформації з РП, використовувати РП для прийняття рішень, що знаходиться в РП. Обмеження РП полягають у першу чергу у кількості елементів РП та швидкості доступу до них.

Зокрема, РП людини має досить обмежені характеристики. Дослідження показують, що РП може містити тільки від трьох до восьми інформаційних елементів. Це обмеження треба враховувати при поданні інформації (на лекції, практичному занятті, у проведенні управлінських рішень тощо). При цьому ОПР не може оперувати всіма елементами інформації одночасно без використання допоміжних засобів.

Інформація, що утримується у РП, також досить швидко «стирається», якщо вона не використовується або не «відновлюється». У відповідності до результатів виконаних досліджень дані утримується в РП від 7 до 13 секунд.

2. Швидкість виконання осмислених операцій.

Оброблення інформаційних елементів виконується в пам'яті із скінченою швидкістю, тобто цей процес характеризується своєю конкретною швидкодією. Кожна елементарна операція мислення, тобто порівняння даних, створення асоціацій з минулими діями (подіями), генерування висновку на основі отриманих даних і перехід до робочої гіпотези, вимагає для реалізації деякого фіксованого відрізка часу, який можна оцінити приблизно в 0,1 секунди.

Для реалізації складних процесів мислення необхідно набагато більше часу, оскільки вони складаються з багатьох згаданих елементарних операцій. Час, необхідний для ухвалення рішення, відіграє важливу роль для систем, що функціонують у масштабі реального часу.

Очевидно, що у багатьох випадках ОПР далеко не завжди здатна прийняти правильне рішення за короткий проміжок часу. Можливо, що ОПР взагалі не може прийняти рішення, якщо цей проміжок занадто короткий.

3. Шляхи одержання інформації.

Зазвичай ОПР має можливість одержувати інформацію з двох джерел: від органів відчуттів і з довгострокової пам'яті. Необхідно зазначити, що інформація, яка «читається» з довгострокової пам'яті, не завжди є надійною. Це є наслідком того, що згодом вона частково або цілком «стирається».

Часто люди схильні до використання тієї інформації, яка частіше необхідна в повсякденному житті, або часто повторюється за різних причин. Слід враховувати також, людина схильна до використання інформації, яка семантично наближена до даних, які перебувають у РП.

4. Методи і засоби обробки числових даних.

Однією з операцій мислення, які людина часто виконує в процесі прийняття рішень, є оброблення числових даних. Але оброблення таких даних пов'язано з низкою обмежень.

Наприклад, навіть «арифметично» добре тренована ОПР здатна робити помилки в обчисленнях і забувати проміжні результати. Зазвичай, кожна елементарна операція займає набагато більше часу ніж 0,1 с. А забування числових даних вимагає повторних обчислень, що значно сповільнює оброблення даних.

Звичайно ОПР знає про ці обмеження, і, як наслідок, намагається уникнути операцій, пов'язаних із складними арифметичними обчисленнями.

Тому в основу методів і засобів прийняття рішень ОПР часто вибирає алгоритми прийняття рішень, у яких часто беруть операції, які ґрунтуються на якісних і евристичних операціях мислення. Подібні обмеження можуть обумовлювати досить серйозні перешкоди у створенні СППР, що ґрунтуються на складних комп'ютерних обчисленнях. Основна причина тут полягає у тому, що в ОПР часто виникає бажання не чекати

завершення виконання складних обчислювальних операцій від комп'ютера, а більше покладатися на звичне для себе якісне та евристичне мислення.

5. Зв'язок виконання операцій у часі і просторі.

Особи ОПР і люди загалом призвичаєні до візуального подання результатів своєї роботи. Зокрема, це стосується і подання результатів прийняття рішень.

Але візуальне подання не завжди надає можливість досягти хороших результатів. Наприклад, ми можемо спостерігати траєкторії польоту двох літаків у вигляді кривих на площині, але ми не можемо точно спрогнозувати точку перехоплення.

Аналогічні обмеження стосуються прогнозування розвитку в часі різноманітних фізичних процесів. Ми можемо спостерігати рух світлової крапки на екрані, але не можемо точно вказати координати її перебування через 7 с.

Таким чином, як і у випадку оброблення числової інформації, прийняття рішень у реальному часі потребує збільшення у багато разів швидкості виконання операцій у часі і просторі.

Зазначені обмеження носять загальний характер і відносяться до усіх випадків, коли прийняття рішень виконується без допоміжних засобів (наприклад, комп'ютера). Вони призводять до виникнення кількох специфічних проблем, які пов'язані з прийняттям рішень у реальному часі, впливають з декомпозиції й аналізу у СППР.

2.2 Стратегії прийняття рішення

Поширеним є виділення таких загальних стратегій, які використовує ОПР для прийняття рішень [3]:

- 1) оптимізаційна;

- 2) перша прийнятна;
- 3) стратегія аспектного виключення;
- 4) інкрементна;
- 5) змішане сканування (перегляд);
- 6) аналітико-ієрархічний підхід.

2.2.1 Оптимізаційна стратегія

Оптимізаційна стратегія передбачає використання математичних моделей і критеріїв оптимізації в явному вигляді.

Особа ОПР обирає ту альтернативу, яка є найкращою (неминучою) за деяким критерієм або критеріями.

Прикладами критеріїв є :

- а) оптимізація функції вигідності для різних варіантів розподілу ресурсів на споживання та розвиток виробництва;
- б) оптимальне управління інфляцією шляхом регулювання грошової маси в обігу (оптимум грошової маси);
- в) оптимальний вибір місця для розміщення складів постачання мережі магазинів (мінімізація витрат на перевезення);
- г) оптимальне управління процесом трансформування власності.

Оптимізаційний підхід підвищує якість рішення за рахунок таких факторів:

- підхід дозволяє знаходити варіанти розв'язку задачі при різних значеннях реальних обмежень на змінні та різних початкових умовах;
- підхід дозволяє спростити процедуру вибору кращого рішення завдяки використанню аналітичних критеріїв; при цьому можна одночасно використати декілька критеріїв;

- за наявності множини методів розв’язання задач динамічної оптимізації підхід дає можливість вибрати кращу альтернативу.

Недоліками оптимізаційного підходу є:

- критерій може мати якісний, а не кількісний характер (наприклад якість життя), що ускладнює застосування аналітичних процедур;

- існують складності з оцінюванням вартості реалізації оптимальних стратегій (оскільки неможливо оптимізувати все);

- об’єм необхідної інформації для побудови моделі може виявитись надзвичайно великим;

- існує принципова неможливість знаходження розв’язку деяких багатокритеріальних задач оптимізації;

- складності обчислень за побудованою моделлю можуть звести задачу до неоптимального рішення.

2.2.2 Стратегія «вибір першої прийнятної стратегії»

За стратегією вибору першої прийнятної стратегії за прийнятну приймають першу стратегію, яка забезпечує суттєве покращення у порівнянні з існуючою ситуацією або часткове покращення за деяким нескладним критерієм. Ідея стратегії полягає у тому, щоб знайти «будь-яку голку в копиці сіна», а не в тому щоб знайти «найгострішу голку».

Альтернативи порівнюють (приймають або відхиляють) у відповідності із визначеним правилом. Наприклад, якщо приймається рішення стосовно зміни місця роботи, то можуть бути такі прості критерії:

- неприйнятна відстань до місця роботи;

- неприйнятний час, необхідний для того щоб дістатися до місця.

2.2.3 Стратегія аспектного виключення

За стратегією аспектного виключення кількість альтернативних варіантів рішень скорочують за рахунок їх виключення за деяким одним аспектом, потім виключають за другим аспектом тощо.

Наприклад, при розміщенні нових підприємств на території України необхідно враховувати:

- наявність трудових ресурсів (кваліфікованих та некваліфікованих);
- потреби в сировині та водопостачанні;
- вплив підприємства на навколишнє середовище;
- розв'язання проблеми транспортування і збуту готової продукції.

Так, в США перше питання, яке розглядають при реєстрації нового підприємства, є рівень податку у конкретному штаті. Ці рівні податку є різними для різних територій. Тому вигідним є реєстрація у штаті з нижчим рівнем оподаткування.

2.2.4 Інкрементна стратегія

Ця стратегія полягає у тому, що ОПР послідовно порівнює альтернативні шляхи розв'язання задачі по відношенню до поточної ситуації.

Задача прийняття рішення полягає у тому, щоб виключити знайдені поточні недоліки функціонування підприємства, установи тощо. При цьому кожне нове рішення в більшій мірі є реакцією на попереднє.

Наприклад, якщо приймається рішення стосовно розширення об'ємів виробництва, то наступне рішення може бути спрямоване на збільшення площ складських приміщень. Після цього виникає задача автоматизації перевезення продукції із цеху на склад та автоматизації транспортно-складської системи підприємства в цілому.

2.2.5 Стратегія змішаного сканування

Стратегія змішаного сканування полягає у перегляді інформації. Перегляд стосується збирання, оброблення, оцінювання та порівняльного аналізу інформації, яка відноситься до поставленої задачі.

Спочатку збирають список можливих альтернатив і в результаті їх прискореного аналізу виключають ті, які явно не підходять. Альтернативи, які залишились, розглядають докладно та знову відхиляють (виключають) неприйнятні за простими зрозумілими критеріями. Процедури відхилення продовжують доти, доки не залишиться одна альтернатива.

2.2.6 Аналітико-ієрархічна стратегія

Аналітико-ієрархічна стратегія полягає у декомпозиції загальної цілі рішення, що приймається, в ієрархічну структуру критеріїв, підкритеріїв та альтернатив. Потім ОПР порівнює критерії попарно з метою знаходження кращого критерія, який задовольняє глобальній меті. Кожному судженню попарного порівняння присвоюють ваговий коефіцієнт у діапазоні $1 \div 9$. Результатом є отримання матриці результатів порівняння підкритеріїв.

Для кожної матриці попарних порівнянь з використанням математичних методів отримують шкалу відносних значень, які оцінюють через пріоритетні одиниці.

Аналітико-ієрархічна стратегія надає можливість включити в критерії, що розглядаються, якісні критерії. Недоліком стратегії є те, що не завжди можна побудувати матрицю попарних порівнянь. До недоліків можна віднести також громіздкість із збільшенням кількості рівнянь, критеріїв та альтернатив.

2.3 Прийняття особистого рішення відносно власної кар'єри

2.3.1 Фаза аналізу шляху

Кожний окремий випадок прийняття рішення зазвичай потребує виконання послідовності етапів (фаз) прийняття рішення. Прикладом є послідовність фаз прийняття особистого рішення стосовно власної кар'єри.

Вибір (планування) власної кар'єри складається з таких етапів або фаз: аналізу шляху, проектування альтернатив, вибору кращої (прийнятної) альтернативи.

Після закінчення середнього навчального закладу необхідно обрати подальший шлях. Очевидно, що прийняття рішення стосовно напрямку подальшого руху залежить від:

- рівня і спрямованості нахилів особи (здібностей);
- скільки грошей можна витратити (знайти) на освіту;
- який результат очікується після вибору того чи іншого шляху.

Основна мета (ціль) прийняття рішень – обрати такий шлях, щоб забезпечити собі (щонайменше) середній рівень матеріального забезпечення і щоб робота не викликала відрази.

Допоміжні цілі:

- вища освіта;
- власна квартира (дім);
- пристойна зарплата;

- зберегти здоров'я;
- дружна сім'я (створити таку сім'ю, щоб завжди хотілось повертатись до дому).

2.3.2 Фаза проектування альтернатив

Фаза проектування або побудови альтернатив передбачає виконання відповідних дій визначення альтернатив.

Прийнятним у цій фазі є виконання таких дій:

- вибрати (спроєктувати) альтернативи;
- дослідити можливості їх реалізації.

Основними можливостями вибору (альтернативи) у цьому випадку є:

- досягти всього самому;
- частково досягти самому, а частково нехай допоможуть батьки;
- вдало одружитись або вийти заміж.

Зупинимось на першій альтернативі і виконаємо аналіз можливих шляхів досягнення основної мети (тобто, всього досягаю сам).

Напрями в межах першої альтернативи можуть бути:

- а) технічний або гуманітарний заклад вищої освіти (ЗВО);
- б) військовий навчальний заклад;
- в) міліцейська академія;
- г) навчання за кордоном;
- д) служба в армії з перспективою залишитись прапорщиком, або продовжити навчання у військовому училищі;
- е) професійно-технічне училище з перспективою стати робітником на заводі.

Розглянемо перший варіант.

Після закінчення технічного вузу можливі такі варіанти працевлаштування:

а) знайти фірму де хороша зарплата і намагатися зробити кар'єру (якщо є такі здібності) менеджера. Припустимо, що в 35 років ви стаєте одним із директорів із заробітною платою 2500\$ – 3000\$ на місяць. Це дозволить купити в кредит квартиру і поселитися там із своєю сім'єю;

б) аспірантура: захистити кандидатську дисертацію через 3 роки (якщо умови сприятливі) і отримувати в 25 років зарплату \approx 200-300\$. Через 5 – 6 років захистити докторську дисертацію і в 33 –34 роки отримувати зарплату \approx 500-600 \$. Немає можливості купити квартиру в кредит, але ще є час, щоб знайти до 40 років кваліфіковану та високооплачувану роботу за кордоном;

в) після закінчення вузу можна знайти роботу, або продовжити навчання за кордоном:

– знайти роботу програмістом і відразу отримувати пристойну зарплату; це дозволить купити квартиру в кредит і мати загалом нормальний рівень життя;

– поступити в аспірантуру і отримати ступінь доктора філософії за обраним напрямом (PhD) за 3 – 4 роки.

Далі робота викладачем і в 40 років можна стати повним професором. Зарплата дозволяє купити квартиру в кредит і жити відносно пристойно;

г) якщо після закінчення ВНЗ є досить грошей на життя, то можна отримати другу освіту, яка, можливо, приведе до успіху;

д) влаштуватись програмістом у великому місті. Можливо, що через декілька років можна буде купити квартиру в кредит.

2.3.3 Фаза вибору кращої (прийнятної) альтернативи

Які критерії (аспекти) використаємо для вибору кращої альтернативи:

- чи є здібності до керування (до організаційної роботи);
- скільки часу необхідно для того, щоб отримати пристойну зарплату;
- рівень зарплати;
- скільки зусиль необхідно витратити;
- чи влаштовує життя в умовах відірваності від родичів, друзів, батьківщини у випадку виїзду за кордон.

Для вибору альтернативи можна застосувати стратегію аспектного виключення. Всього п'ять альтернатив:

а´) якщо не влаштовує відірваність від родичів і друзів, то в) відпадає;

б´) якщо не влаштовує аспірантура в Києві чи іншому місті, то б) відпадає;

в´) якщо немає здібностей до керування, то а) відпадає;

г´) якщо не хочеться бути програмістом, то д) відпадає.

Залишається одна альтернатива г), тобто отримати другу освіту і сподіватись, що вона допоможе знайти те, чого хочеться досягти в житті.

2.4 Етапи проектування СППР

2.4.1 Склад етапів проектування СППР

Процес реалізації етапів проектування СППР тісно пов'язан з визначенням складу етапів та когнітивними процесами виконання окремих підетапів (кроків) проектування.

З формулювань, наведених вище, очевидно, що основною проблемою при проектуванні СППР є аналіз і з'ясування процесу ухвалення рішення ОПР, визначення обмежень, що накладаються на процес ухвалення рішення, а також вибір методів і обчислювальних процедур, що дозволять зняти подібні обмеження. У загальному випадку проектування СППР складається з трьох етапів.

1. Декомпозиція процесу ухвалення рішення на елементарні операції і опис виконання цього процесу особою, що приймає рішення.

2. Аналіз конкретної задачі стосовно ухвалення рішення і проектування СППР на функціональному рівні.

3. Докладна специфікація функцій системи, її реалізація, верифікація (тестування) і супроводження.

2.4.2 Когнітивний процес проектування

Процес проектування представлено графічно на рис. 2.1. Центральний ряд прямокутників показує які конкретно етапи або дії необхідно виконати для того щоб спроектувати систему. Для виконання конкретних етапів необхідно мати в розпорядженні спеціальний інструментарій, а також опис результату виконання кожного етапу. Види інструментарію і специфікації результатів наведені відповідно ліворуч і праворуч від центрального ряду.

Процес проектування починається з того, що обирається задача, для розв'язання якої необхідно створити систему підтримки прийняття рішень або ця задача ставиться ОПР, яка уже знайома з проблемою прийняття конкретних рішень на підставі власного досвіду.

У відповідності з рис. 2.1, на першому етапі виконується декомпозиція задачі на елементарні операції і описується виконання цього процесу особою, що приймає рішення. Основна мета цього етапу полягає в наступному:

- визначити перешкоди («вузькі місця»), які необхідно перебороти при прийнятті рішень за допомогою СППР, що проектується;

- визначити набір комп'ютерних алгоритмів, які необхідно використати для подолання вузьких місць, зв'язаних з оперативним прийняттям правильних рішень.

Для того щоб правильно розв'язати задачу проектування СППР, необхідно максимально структурувати (зробити прозорим) опис процесу ухвалення рішення. Така структура представляється докладним, але чітко визначеним протоколом, у якому вказується, які дані необхідно зібрати, і описати всі часткові (окремі) рішення, які повинні бути прийняті при проектуванні СППР. Результатом виконання цього етапу є структурована таблиця, у яку зводяться всі результати, що відносяться до декомпозиції проблеми проектування СППР.

Три наступних етапи, представлених на рис. 2.1, являють собою фазу аналізу і функціонального проектування СППР.

Спочатку необхідно проаналізувати саму задачу, стосовно якої необхідно приймати рішення, і підхід ОПР до її розв'язання. У процесі цього аналізу визначають фактори, які створюють перешкоди до розв'язання задачі, якщо це робиться без комп'ютера. Простим інструментом для виконання цієї задачі є написання чіткого переліку

загальної інформації, що необхідна для підтримки ухвалення рішення. Цей перелік можна порівняти з таблицею вимог, що була створена на першому етапі. Табулювання ситуацій, зв'язаних із прийняттям рішень, дозволяє конкретизувати зв'язані з ними конкретні обмеження. Таким чином, результатом виконання цього етапу проектування СППР є перелік конкретних труднощів, що виникають при прийнятті рішень щодо конкретної проблеми.

Наступним кроком процесу проектування є ідентифікація (визначення) тих труднощів (перешкод) при прийнятті рішень, що можуть бути цілком або частково усунуті завдяки застосуванню обчислювальних процедур. На цьому етапі необхідно скласти перелік функцій стосовно прийняття рішень, які можна перенести на комп'ютер. Цей перелік у деякій мірі буде схожий на список перешкод, складений на попередньому етапі. Тобто тут необхідно вказати конкретно, які з перешкод можуть бути усунуті при прийнятті рішень і за рахунок використання яких методів.

На останньому етапі фази функціонального проектування встановлюється відповідність між бажаними функціями СППР і одним або більше методами, що підтримують реалізацію цих функцій і є сумісними з процесом прийняття рішень користувачем системи. Це досить складне завдання, що вимагає визначення сумісності між спеціальними функціями системи із загальними принципами побудови СППР. Необхідна сумісність забезпечується функціонально організованою класифікацією методів створення СППР, а також множиною правил узгодження індивідуальних методів СППР конкретними задачами на основі характеристик задач і ОПР.

Подібні правила застосовують до переліку бажаних функцій з метою визначення конкретних обчислювальних процедур, що будуть реалізовані в рамках проектованої системи.



Рис. 2.1 Когнітивний процес проектування СППР

Сполучення функцій системи з обчислювальними процедурами дає можливість створити так звану функціональну архітектуру СППР.

Інші етапи проектування і створення СППР деталізують процес реалізації створеної на попередніх етапах функціональної архітектури. Конкретні обчислювальні процедури зв'язують з відповідними задачами системи за допомогою аналізу даних, інженерії знань та інших методів. Після сполучення функцій і процедур реалізується етап програмування окремих модулів системи. На цьому етапі також розробляється і програмується інтерфейс між користувачем і машиною.

Розглянемо докладніше основні етапи проектування.

2.4.3 Визначення і декомпозиція задачі прийняття рішень

2.4.3.1 Визначення задачі прийняття рішень

Визначення і декомпозиція задачі прийняття рішень є першим етапом проектування СППР. Створення опису й аналіз задачі прийняття рішень є фактично основним етапом у проектуванні системи. Якщо не встановити із самого початку обмеження на функції і дані системи, то всі подальші зусилля можуть виявитися марними, оскільки вони можуть виходити за рамки можливостей проектованої СППР.

Експерти, що приймають рішення і звикли бачити весь процес «у цілому», можуть мати істотні труднощі з використанням системи, що змусить «звузити» 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 ормату.

2.4.3.2 Декомпозиція задачі прийняття рішень

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 подання 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. Зазвичай, це фізичні події, такі як підвищення доходу до заданого рівня, досягнення бажаних характеристик системи тощо.

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.

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.

2.4.3.3 Опис ситуації з прийняття рішення

Опис ситуації з прийняття рішення. Пропонується підхід до підтримки рішень, що ґрунтується на когнітивному описі цього процесу. З іншого боку, прагматична сторона проектування СППР на основі ЕОМ вимагає глибокого розуміння фізичної, інформаційної і контекстної поведінки, у якій повинна працювати система. Таким чином, існує дві сторони опису ситуації з ухвалення рішення – більш традиційними є біхевіористично-фізичний опис, а сучасним – когнітивний опис. Використовуючи такий подвійний опис процесу, можна знайти ті «вузькі місця» архітектури прийняття рішень ОПР, що потребують допомоги з боку ЕОМ.

Для того щоб застосувати цей підхід на практиці, необхідно докладно структурувати і описати весь процес прийняття рішень. Структура опису створюється за допомогою двох інструментів. Першим інструментом є формат опису, що ґрунтується на широких категоріях аналізу, загальних для більшості рішень. Іншим інструментом є протокол збору даних і опитування (збір інтерв'ю), складені таким чином, щоб зібрати конкретну докладну інформацію з кожної широкої категорії, необхідної для аналізу труднощів прийняття рішень і визначення необхідності підтримки їхнього прийняття.

2.4.3.4 Формат опису ситуації з ухвалення рішення

Формат опису ситуації з ухвалення рішення складається з таких восьми категорій:

1. Ситуативні цілі. У цій частині документуються результати декомпозиції головної мети. Описується головна мета верхнього рівня, на якій зосереджується процес прийняття рішень і когнітивне оброблення 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. 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 загального рішення.

Розглянутий опис необхідний для можливої розробки і створення БД 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. Подання рішення. Досвідчений експерт з прийняття рішень має свій «алгоритм» мислення і подання 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 подання 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 подання проблеми виявляють своє втілення у візуальних образах і картинах. Визначення цього розумового подання 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. Нарешті, корисно створити скорочений опис, який займав би одну сторінку у вигляді таблиці. Формат такої таблиці представлений на рис. 2.2.

2.4.4 Аналіз ситуації з прийняття рішення

2.4.4.1 Визначення обмежень процесу прийняття

Аналіз ситуації з прийняття рішення є другим етапом проектування СППР. Він потребує визначення обмежень процесу прийняття, аналіз типових проблем прийняття рішень.

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 вмикання в комп'ютерну систему.

Ситуація з ухвалення рішення: (назва ситуації або проблеми).		
Динаміка задачі: (тип динаміки: ітерації в замкнутому циклі, послідовність дій або одноразова дія).		
Ситуативні цілі: (ціль найвищого рівня, що визначає ситуацію і яка виражена через події – фізичні або інші, котрі можна спостерігати).		
Критерії оцінки: (перелік індивідуальних критеріїв, за допомогою яких буде оцінюватися можливе рішення).		
Фізичний процес: (тут дається короткий опис (одна пропозиція) основного процесу, стосовно якого приймається рішення).		
Інформаційна підтримка:		
Входи	Виходи	Параметри
Перелік видів інформації, яка може бути використана при прийнятті рішення і яка може змінити своє значення в процесі роботи.	Перелік видів інформації, яка створюється в процесі прийняття рішення; наприклад, відносно різних аспектів рішення.	Перелік видів інформації, яка може бути використана при прийнятті рішення і не змінюється на протязі однієї ситуації, але може змінюватися у подальшому.
Проміжний аналіз: (перелік етапів процесу ухвалення рішення, що виконує ОПР, приймаючи рішення без комп'ютера).		
Подання ситуації: (короткий опис ухвалення рішення ОПР і які засоби (лінгвістичні, візуальні і т. ін.) використовуються).		
Необхідні судження: (перелік евристичних суджень, що повинна виконати ОПР при ухваленні рішення).		

Рис. 2.2 Формат таблиці для опису процесу прийняття рішень

Визначення обмежень (труднощів) процесу прийняття рішень пов'язано з особливостями способів прийняття рішень. В літературі визначено, що існує шість загальних моментів, зв'язаних з необхідністю використання способів підтримки прийняття рішень. Вони є загальними для більшості ситуацій із прийняття рішень і мають відомі форми обчислювальної підтримки.

Ці шість моментів можна розглядати як «вузькі місця» у процесі прийняття рішень ОПР (які працюють без засобів підтримки прийняття рішень). Такий підхід збігається з вищеописаною ідеологією.

Як тільки вдається визначити слабкі місця процесу прийняття рішень, стає можливим вибрати необхідну технологію для посилення цих місць за допомогою відповідних логічних і обчислювальних процедур. Розглянемо шість зазначених моментів.

2.4.4.2 Характеристика типових проблем прийняття рішень

Можна виділити кілька типових проблем прийняття рішень:

1. Неможливість прогнозувати процес. Численні дослідження показують, що більшість людей мають труднощі з прогнозуванням подій у майбутньому. Це особливо відноситься до тих випадків, коли у протіканні реальних процесів спостерігається невизначеність. Труднощі пов'язані з такими основними обмеженнями когнітивної архітектури людини: інформаційні потреби при розв'язанні задач перевищують можливості робочої пам'яті; існує велика потреба у вивченні і запам'ятовуванні дуже багатьох подробиць; крім того, дуже часто виникає необхідність у виконанні досить складних об'ємних обчислень.

Багато ОПР мають труднощі з прогнозуванням подій, а тому покладаються на свій досвід і загальну евристику. З іншого боку, комп'ютер цілком розв'язує задачу прогнозування за допомогою ряду процедур, якщо є відповідна інформація стосовно спостережень. Тому цю задачу необхідно перекладати на ЕОМ.

2. Труднощі з об'єднанням атрибутів і цілей. У більшості ситуацій із прийняття рішень ОПР має у своєму розпорядженні декілька атрибутів або

критеріїв, що використовуються для опису очікуваного результату після ухвалення рішення.

Такі критерії і їхні комбінації часто бувають нечітко визначеними або субоптимальними. У таких випадках ОПР може скористатися знаннями більш досвідчених експертів із прийняття рішень. Доступ до їхніх знань можна організувати за допомогою бази знань. Навіть якщо ОПР має у своєму розпорядженні чіткі правила для порівняння альтернатив, знайти найкращу буває досить складно, що зумовлено необхідністю оброблення числових атрибутів. Такі обчислення легко і швидко може виконати комп'ютер.

3. Труднощі з обробкою даних, що необхідні для прийняття рішень. Зазвичай у ОПР виникають труднощі з використанням усієї наявної інформації при ухваленні рішення. Мається на увазі, що вся інформація повинна пройти через обмежену робочу пам'ять ОПР. Необхідно враховувати, що процес «пошуку і читання» інформації з довгострокової пам'яті не характеризується високою надійністю.

У зв'язку з цими особливостями ОПР може просто застосувати правило «вгадування» замість того щоб скористатися істинною інформацією. У цьому випадку необхідно скористатися допомогою комп'ютера, що забезпечує швидкий і надійний пошук інформації у великих обсягах пам'яті.

4. Проблеми з аналізом і формуванням логічного висновку (прийняття рішення). ОПР знають як необхідно обмірковувати задачу, але часто не можуть виконати цю задачу внаслідок обмежених ресурсів пам'яті і часу. ОПР може знати, наприклад, що можна знайти оптимальне рішення шляхом порівняння багатьох альтернатив, але знає і те, що це неможливо зробити за відведений для цього проміжок часу.

Подібні труднощі виникають внаслідок загальних обмежень людини з погляду можливості оброблення великих обсягів інформації, таких як обмеження робочої пам'яті, труднощі оброблення великих масивів чисел і складних обчислень і прогнозування значень параметрів. Необхідно також відзначити, що тип аналізу і необхідність формування логічного висновку істотно відрізняються для різних практичних задач. Але в більшості випадків існує можливість створення комп'ютерної підтримки у виді алгоритму, що буде виконувати функції формування логічного висновку.

5. Труднощі з візуалізацією результатів. Для людини природно використовувати звичне для неї візуальне подання процесу розв'язку тієї або іншої задачі. Але це досить складний процес, особливо якщо в процесі рішення присутні масиви числових даних. Це ще один приклад того, що людина не може розумово виконувати складні обчислення. Задача полегшується, якщо для візуалізації застосувати комп'ютер. Візуалізація не тільки дає загальне подання про хід процесу мислення, але і допомагає його виконувати. Наприклад, вона набагато полегшує прогнозування наслідків прийняття того або іншого рішення (психологи стверджують, що хороший інформативний рисунок може замінити до двох тисяч слів).

6. Неточність процесу евристичних суджень. Можна привести багато прикладів, коли ОПР змушена робити логічні висновки, які можна пояснити тільки з погляду суджень, тобто без застосування числових критеріїв.

Досвідчені ОПР можуть робити такі судження з високою якістю і точністю, але коли зустрічаються кількісні моделі й аспекти, то спостерігається систематичний зсув оцінок або «шум». Це зумовлено загальною обмеженістю людини виконувати складні кількісні операції і труднощами «читання» даних з довгострокової пам'яті. У даному випадку комп'ютер не може конкурувати із людиною з погляду реалізації суджень,

але його можна використовувати для усунення «зсуву» або фільтрації шуму. Комп'ютер можна також використати для порівняння поточного рішення з минулими варіантами розв'язків і результатами.

2.4.4.3 Категорії опису ситуацій

Розглянуті шість загальних проблем із прийняттям рішень досить тісно пов'язані з категоріями, що були використані вище для опису ситуацій із прийняття рішень, тобто:

- конкретний фізичний процес – проблема прогнозування;
- критерії вибору – труднощі з об'єднанням атрибутів (якщо необхідно об'єднати декілька критеріїв, або порівняти кілька альтернатив, то ОПР не може це зробити з необхідною якістю без допомоги ЕОМ);
- інформаційні вимоги – труднощі з «пошуком і читанням» інформації;
- проміжний аналіз – проблеми з аналізом або логічним висновком;
- подання рішення – труднощі з візуалізацією або постановкою у відповідність даних і розумової моделі;
- необхідність у судженнях – кількісні неточності евристичних суджень.

Наведене подання допомагає структурувати (уточнити) процес підтримки прийняття рішень. Наведена вище таблиця (рис. 2.2) і декомпозиція ситуації з ухвалення рішення розглядаються етап за етапом з метою виявлення труднощів ОПР при прийнятті рішень без допоміжних засобів. В результаті одержують перелік проблемно-орієнтованих труднощів, які є прикладами шести розглянутих вище загальних моментів.

Очевидно, що інформаційне забезпечення і процес ухвалення рішення можуть відрізнитися для конкретних задач, тому потреби в

допомозі при ухваленні рішення можуть носити різний характер. У більшості випадків необхідно забезпечити декілька видів комп'ютерної підтримки, але дуже рідко буває, що підтримка необхідна за всіма шістьма пунктами.

2.4.5 Визначення функцій системи підтримки прийняття рішень

2.4.5.1 Функції для підтримки прийняття рішень з боку ОПР

Визначення функцій системи підтримки прийняття є третім етапом проєктування СППР. Сучасні технології проєктування СППР надають ряд можливостей проєктування.

Загальноприйнятим є використання 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 оброблення інформації, даних, знань використовують сучасні комп'ютерні технології. Завдяки цьому значно розширюються можливості ОПР із прийняття рішень і оброблення 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 подання результатів. Для того щоб реалізувати функції доступу до інших СППР, баз даних і знань, необхідно застосовувати засоби комп'ютерної графіки й інструментарій для оброблення 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.

2.4.5.2 Вибір конкретних обчислювальних процедур

Вибрані функції СППР для підтримки прийняття рішень з боку ОПР потребують певної реалізації. Тому 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.

2.4.6 Вибір технології для реалізації СППР

Вибір технології (методів) для реалізації СППР є четвертим етапом створення системи. Цей вибір виконують у межах біхевіористичних і когнітивних обмежень.

Функціональне проектування СППР є 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.

2.5 Засоби підтримки прийняття рішень

2.5.1 Види математичного інструментарію

Основним інструментарієм, який необхідно мати на цьому етапі, є чітке визначення і категоризація технологічної бази СППР. Таке визначення і категоризація ґрунтується на функціях і потребах, розглянутих вище.

Можна виділити шість видів математичного інструментарію, що використовується при проектуванні і реалізації СППР (рис. 2.3). Такими видами є [23]:

- 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 подання 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. 2.3, 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 категорії рис. 2.3 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.



Рис. 2.3 Класифікація засобів підтримки прийняття рішень

2.5.2 Правила реалізації функцій СППР

Правила реалізації повинні ґрунтуватись 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. 2.3. 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. 2.3, з конкретними потребами підтримки прийняття рішень будуть розглянуті нижче.

Запитання та завдання для самоконтролю

1. Пояснити сутність атрибута використання внутрішнього (розумового) подання проблеми (задачі) прийняття рішень. Оцінити переваги і недоліки.
2. Навести приклад використання внутрішнього (розумового) подання проблеми (задачі) та його наслідки з власного досвіду або для заданої ситуації.
3. Пояснити сутність атрибута досягнення мети і прийнятті рішення. Яку роль відіграють структурування і розумове подання даних?
4. У чому полягає атрибут сприйняття інформації і які він має особливості для особи ОНР?
5. Як атрибути комплексно сприймаються ОНР і впливають на прийняття рішення?
6. Що впливає на процес ухвалення рішення особою ОНР?
7. Якими є особливості обробки числових даних особою ОНР?
8. У чому полягає особливості зв'язку виконання операцій у часі і просторі для особи ОНР?

9. Перерахувати основні стратегії прийняття рішення.
10. Пояснити особливості стратегії прийняття рішення. Пояснити особливості оптимізаційної стратегії прийняття рішення
11. Пояснити особливості вибору першої прийнятної стратегії прийняття рішення.
12. У чому полягає стратегія аспектного виключення у прийнятті рішень?
13. Що є інкрементна стратегія прийняття рішення?
14. Як виконують змішане сканування у прийнятті рішення?
15. Пояснити особливості аналітико-ієрархічного підходу прийняття рішення.
16. Від чого залежить і які цілі у фазі аналізу шляху прийняття особистого рішення відносно власної кар'єри?
17. Які дії і альтернативи характерні для фази проектування альтернатив прийняття особистого рішення відносно власної кар'єри?
18. Пояснити вибір критеріїв і стратегії у фазі вибору кращої (прийнятної) альтернативи прийняття особистого рішення відносно власної кар'єри.
19. Дати загальну характеристику складу етапів проектування СППР.
20. Перерахувати основні кроки процесу проектування СППР.
21. Якими є особливості декомпозиції процесу ухвалення рішення?
22. Як описують виконання процесу проектування особою, що приймає рішення?
23. Дати характеристику трьох підетапів аналізу конкретної задачі стосовно ухвалення рішення і проектування СППР на функціональному рівні.

24. Як виконують специфікацію функцій системи СППР, її реалізацію, верифікація (тестування) і супроводження?
25. Як виконують декомпозицію задачі прийняття рішень?
26. Як визначають задачі прийняття рішень?
27. Пояснити ієрархічність цілей і декомпозиції задачі прийняття рішень.
28. Як описують ситуації з прийняття рішення?
29. Описати формат опису ситуації з ухвалення рішення.
30. Пояснити основні категорії формат опису ситуації з ухвалення рішення.
31. У чому полягає динаміка виконання задачі?
32. Як формують критерії вибору рішення?
33. Описати види інформації, з яким має справу ОПР для інформаційної підтримка рішень.
34. Як впливає проміжний аналіз експертів на рішення ОПР?
35. Описати особливості мислення і подання задачі при поданні рішення.
36. Як використовують судження, які неможливо точно описати?
37. Як виконують аналіз ситуації з прийняття рішення?
38. Як визначають обмеження (труднощі) процесу прийняття рішень?
39. Описати сутність основних проблем, які визначають обмеження процесу прийняття рішень.
40. У чому полягає зв'язок обмежень процесу прийняття рішень з категоріями опису ситуацій?
41. Як визначають засоби підтримки прийняття рішень?
42. Дати характеристику основних видів математичного інструментарію засобів підтримки прийняття рішень.

43. Охарактеризуйте атрибути та обмеження процесу прийняття рішення ОПР.

44. Наведіть стратегії, які використовують ОПР для прийняття рішень.

45. Які недоліки і переваги має оптимізаційна стратегія?

46. Наведіть перелік елементів постановки оптимізаційної задачі.

47. Які ви знаєте методи статичної та динамічної оптимізації?

48. Наведіть приклад процесу прийняття рішення ОПР.

49. Сформулюйте етапи процесу проектування інформаційних СППР.

50. Розкрийте процедуру визначення і декомпозиції задачі прийняття рішень.

51. Наведіть категорії, з яких складається формат опису ситуації з ухвалення рішення; наведіть приклад опису ситуації.

52. Опишіть існуючі обмеження процесу прийняття рішень, що вимагають застосування допоміжних засобів.

53. Сформулюйте функції, які виконує ОПР у процесі прийняття рішень.

54. Наведіть класифікацію засобів підтримки прийняття рішень.

55. Поясніть зміст етапу вибору технології для реалізації СППР.

РОЗДІЛ 3 ЗАСТОСУВАННЯ МОДЕЛЕЙ І ДАНИХ У СППР

3.1 Вибір моделей і критеріїв для побудови СППР

Створення і використання СППР тісно пов'язано із застосуванням моделей та інструментаріїв обслуговування. Можна виділити моделі і критерії побудови самої СППР, моделі для оцінювання наслідків прийняття рішень із використанням даної СППР та інструментарій обслуговування або менеджменту СППР із урахуванням різноманітних режимів та ситуацій прийняття рішень.

Існує кілька умов, які визначають необхідність використання моделі процесу прийняття рішень. Очевидною умовою є те, що рішення повинне безпосередньо ґрунтуватись на первинному процесі, що аналізується. Також слід аналізувати поведження процесу з прогнозуванням у сильному або слабкому сенсі.

Процес розглядають як прогнозований у сильному сенсі, 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» сенсі, оскільки прогнозоване значення є достовірним лише для тих умов, які відповідають конкретному наборові даних.

Наприклад, «слабко прогнозованими» є процеси на товарному ринку. Економетричний аналіз свідчить, що агреговане поведження великої кількості 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, які характеризують його поведінку 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 за їх наявності. Правила вибору конкретної моделі процесу подано на рис. 3.1.

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, у динаміці або в даних на виході). Якщо невизначеності мають місце, то необхідно вибирати стохастичну модель. Якщо ж невизначеності не грають істотної ролі, то процес можна розглядати як детермінований. Прикладами стохастичних процесів можуть бути процеси, зв'язані з навколишнім середовищем (зокрема, забруднення, погода), і з економікою.

Важливою характеристикою є бачення реального процесу з погляду ОПР. ОПР має своє уявлення щодо того, 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.

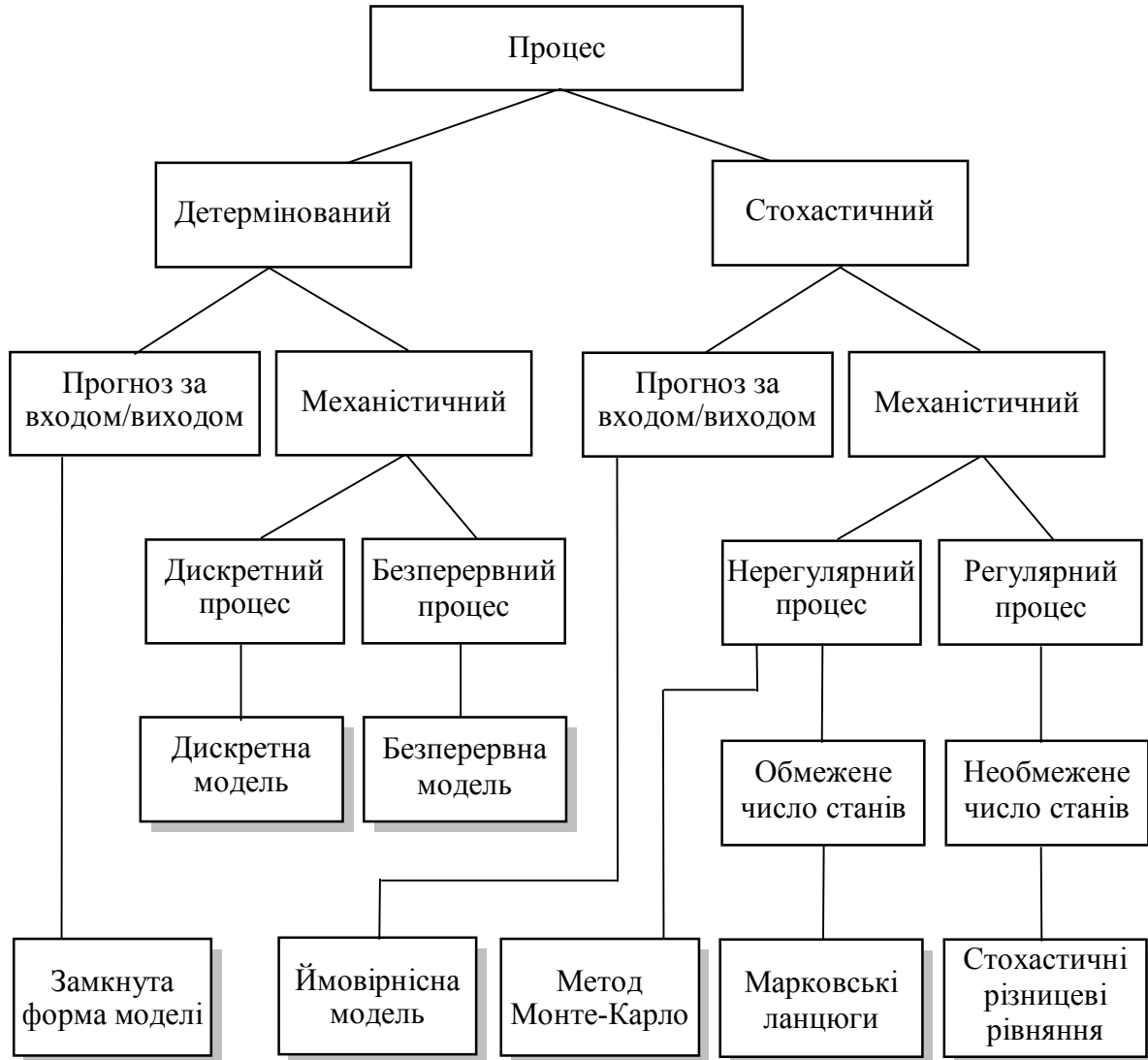


Рис. 3.1 Вибір моделі процесу при проектуванні СППР

Якщо ж процес стохастичний і використовується прогноз на основі відношення вихід/вхід, то при проектуванні СППР необхідно скористатися ймовірнісною моделлю процесу. У цьому випадку зв'язок між вхідними і вихідними змінними процесу встановлюється на основі результатів теорії ймовірностей і статистики.

Крім розглянутих характеристик необхідно визначити яку модель (дискретну або неперервну) слід використовувати з погляду врахування протікання часу.

Очевидно, що найбільше труднощів пов'язано з аналізом стохастичних процесів. Якщо процес адекватно описується за допомогою двадцяти або меншої кількості станів, то його вважають процесом з обмеженою кількістю станів.

Істотною характеристикою процесу є його регулярність. Процес вважають 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 ощо. Підхід до моделювання на основі методу Монте-Карло розглядають як «останню» можливість. 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.

3.2 Вибір моделі для оцінювання наслідків прийняття рішень із використанням СППР

У більшості випадків прийняття рішень ОПР зацікавлена у порівнянні наслідків прийняття альтернативних рішень. Порівняння альтернатив пов'язане, зазвичай, із складними кількісними розрахунками, необхідністю використання спеціальних знань і специфічних моделей для оцінювання альтернатив.

Правила вибору конкретної моделі з числа наведених на рис. 3.1, для конкретної ситуації з прийняття рішень, подано на рис. 3.2.

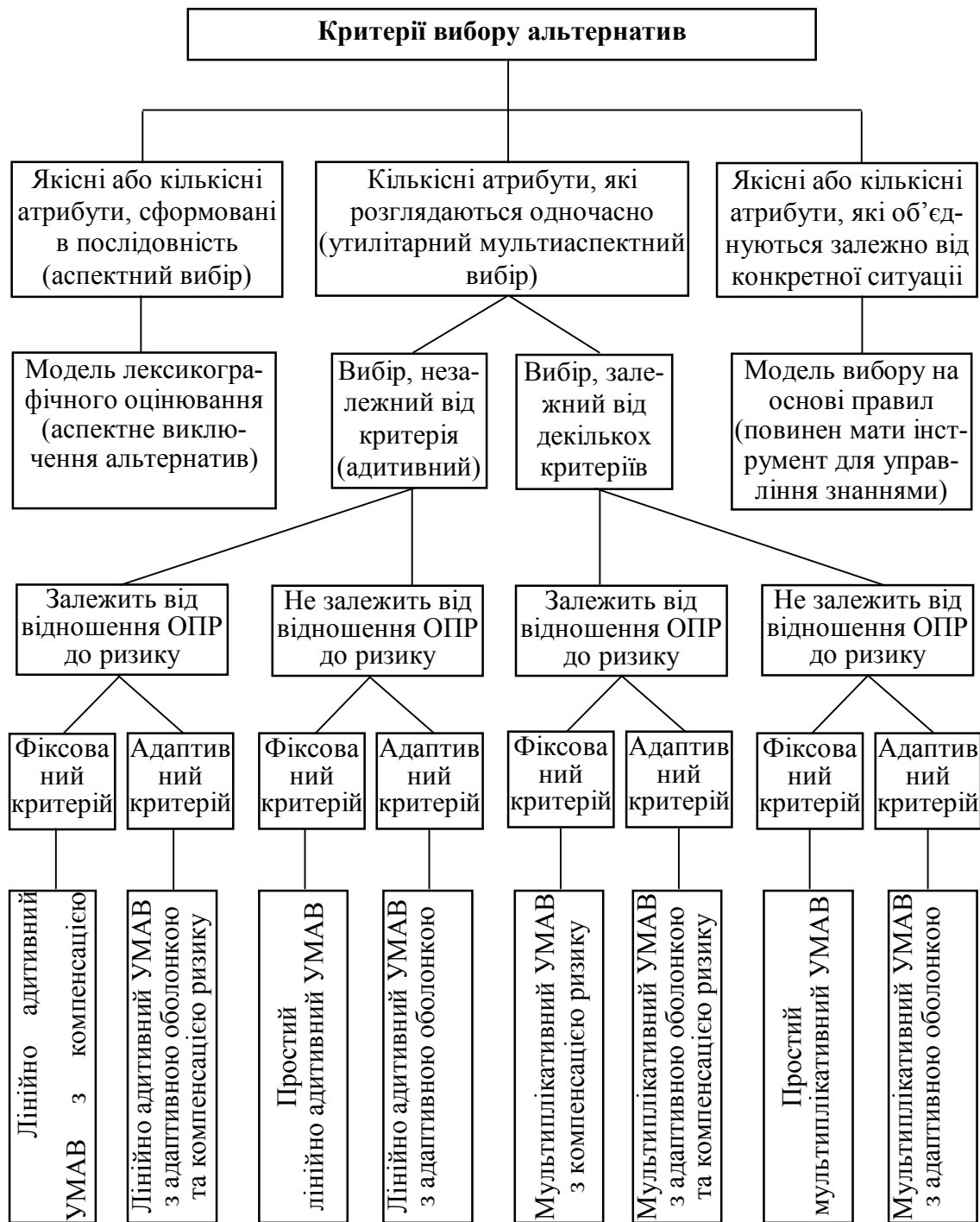


Рис. 3.2 Вибір моделі для оцінювання результату роботи СППР

Якщо у процесі прийняття рішень без засобів підтримки аналізують кілька альтернатив, кожна з яких описують більше ніж одним атрибутом, то такий процес називають мультиатрибутним процесом прийняття рішень.

Якщо всі атрибути розглядають при оцінюванні альтернативи одночасно, то такий вибір називають утилітарним мультиатрибутним вибором (УМАВ). Якщо ж при оцінюванні альтернативи атрибути розглядаються в деякій визначеній послідовності, то такий вибір називають аспектним мультиатрибутним вибором (АМАВ). Якщо атрибути мають якісний характер і розглядаються у визначеній послідовності, то вибір називають також аспектним.

Класичним прикладом УМАВ є прийняття рішення про прийом до вищого навчального закладу, коли одночасно розглядаються декілька атрибутів, що характеризують рівень знань і розвитку абітурієнта.

Прикладом аспектного вибору рішення є покупка будинку для проживання. У цьому випадку спочатку, зазвичай, розглядається ціна, а потім інші атрибути, такі як район його розміщення, відстань до метро (якщо воно є в місті), відстань до дитячого саду і школи.

У деяких випадках прийняття рішень процес вибору альтернатив ґрунтується на комплексних кількісних та/або якісних критеріях і асоціаціях, що змінюються в залежності від характеру проблеми. Якщо вибір носить аспектний характер, то при проектуванні СППР необхідно розглянути можливість «виключення альтернативи за аспектом», тобто використовувати модель «лексикографічного» вибору. За допомогою цієї моделі атрибути оцінюються шляхом їхнього зваженого послідовного порівняння із граничними значеннями. Ця частина процесу проектування носить прямолінійний характер і може бути виконана шляхом спілкування з досвідченим експертом.

Якщо вибір ґрунтується на правилах, то при проектуванні необхідно передбачити створення моделі, що заснована на правилах, або експертній системі. Це модель, що генерує одну альтернативу або перелік можливих альтернатив в інтерактивному режимі роботи з ОПР. У процесі діалогу аналізуються значення атрибутів і додаткові дані, що задасть ОПР. Включення в систему моделі на основі продукційних правил вимагає використання інструментарію для керування базою знань.

Якщо вибір альтернатив ґрунтується на використанні УМАВ, то виконується додатковий аналіз. При цьому, якщо кожен атрибут вносить деякий постійний внесок у загальну якість альтернативи, то такі атрибути називають лінійними. Якщо внесок деякого атрибута не залежить від внеску інших атрибутів, то він називається незалежним. У випадку, коли всі атрибути лінійні і незалежні, то УМАВ-вибір називають лінійно-адитивним. Якщо в межах деякої комбінації атрибутів існує взаємодія, вони називаються взаємозалежними. Коли всі атрибути лінійні, але між деякими з них існує взаємодія, то вибір типу УМАВ називають мультиплікативним-лінійно-мультиплікативним.

При використанні у всіх ситуаціях всіма експертами (зокрема, з осіб ОПР) тільки одного правила для об'єднання атрибутів базис УМАВ називають фіксованим. Якщо ж правило об'єднання атрибутів може змінюватися тим самим ОПР або змінюватися в залежності від ситуації, то базис УМАВ називають адаптивним.

Необхідно зазначити, що різні індивідууми (у ролі ОПР) мають різну схильність до ризику. Одні виявляють схильність до прийняття рішень з ризиком, інші намагаються уникати ситуацій з підвищеним ризиком, а треті не мають чутливості до ризику, тобто, нейтральні. Базис УМАВ називають заснованим на ризику, якщо ОПР ризикує стосовно себе або стосовно ресурсів, за які вона буде нести відповідальність в результаті

ухвалення рішення, і цей ризик можна оцінити за допомогою деяких атрибутів оцінювання результатів. Навпаки, ризик вважається відсутнім, якщо ОПР не вносить елементів ризику в прийняття рішень або в процес оцінювання результатів.

Тепер можна сформулювати правила вибору моделі оцінювання для ситуацій УМАВ. Якщо вибір характеризується як УМАВ і є лінійно-адитивним, то при проектуванні СППР необхідно синтезувати «адитивну утилітарну функцію корисності». Якщо ж ситуація характеризується як УМАВ і лінійно-мультиплікативна, то необхідно синтезувати «мультиплікативну мультиатрибутну функцію корисності» для аналізу альтернатив. У випадку коли вибір ґрунтується на УМАВ при відсутності ризику, але процес адаптивний, то необхідно створити оболонку для адаптивного оцінювання параметрів для мультиплікативної мультиатрибутної функції корисності. Така оболонка представляє собою, фактично, алгоритм для оцінювання параметрів моделі, що використовується для порівняльного аналізу альтернатив. Специфіка інших варіантів, наведених на рис. 3.2, очевидна з вищенаведеного аналізу попередніх випадків.

3.3 Вибір інструментарію для інформаційного менеджменту

Інформація надходить до СППР не менше ніж в двох видах: дані, що характеризують значення конкретних атрибутів або факти стосовно реальної ситуації, і знання, що структурно і семантично описують попередній досвід і дають можливість екстраполювати нові ситуації. Якщо деякі дані характеризують ту частину проблеми, про яку ОПР має визначені знання, то говорять, що ці дані конкретизують знання.

Очевидно, що самі по собі дані без знання того, як їх інтерпретувати, змісту не мають. З іншого боку, знання саме по собі без конкретних даних можна характеризувати як «цікаве», але застосувати його неможливо. За допомогою цього визначення можна сформулювати дві різні проблеми інформаційного менеджменту, що зустрічаються при прийнятті рішень. Тобто ОПР може мати:

- потенційно корисні дані, але не мати досить знань для їхньої інтерпретації;
- потенційно корисні знання, але не мати конкретних даних, щоб застосувати ці знання.

Для багатьох ситуацій, зв'язаних із прийняттям рішень, характерна наявність однієї або обох цих проблем. Перша проблема є особливо характерною для систем реального часу, оскільки можливості ОПР щодо оброблення даних обмежені. При обмеженому обсязі робочої пам'яті і порівняно довгому циклі роботи когнітивного процесора для ОПР може знадобитися досить довгий відрізок часу, щоб виконати аналіз даних і одержати корисну для подальшого розгляду інформацію. У таких випадках розумову роботу ОПР необхідно підтримати методикою (методом) менеджменту даних [12].

Проблема іншого типу зв'язана з відсутністю необхідного навчання і досвіду ОПР. Тут відіграє також роль специфічна архітектура процесора оброблення даних людського організму. Знання, отримані людиною, зберігаються в довгостроковій пам'яті і «читаються» (вибираються) за допомогою семантичних методів, тобто за змістом. Такий тип доступу до пам'яті характеризується високою швидкістю, але низькою надійністю. Люди часто не можуть згадати ту частину знання або даних, яка терміново потрібна у конкретний момент часу. Ця проблема особливо гостра при роботі в режимі реального часу. Її вирішують за допомогою методів

менеджменту знань. Правила вибору методу менеджменту даних наведені на рис. 3.3.

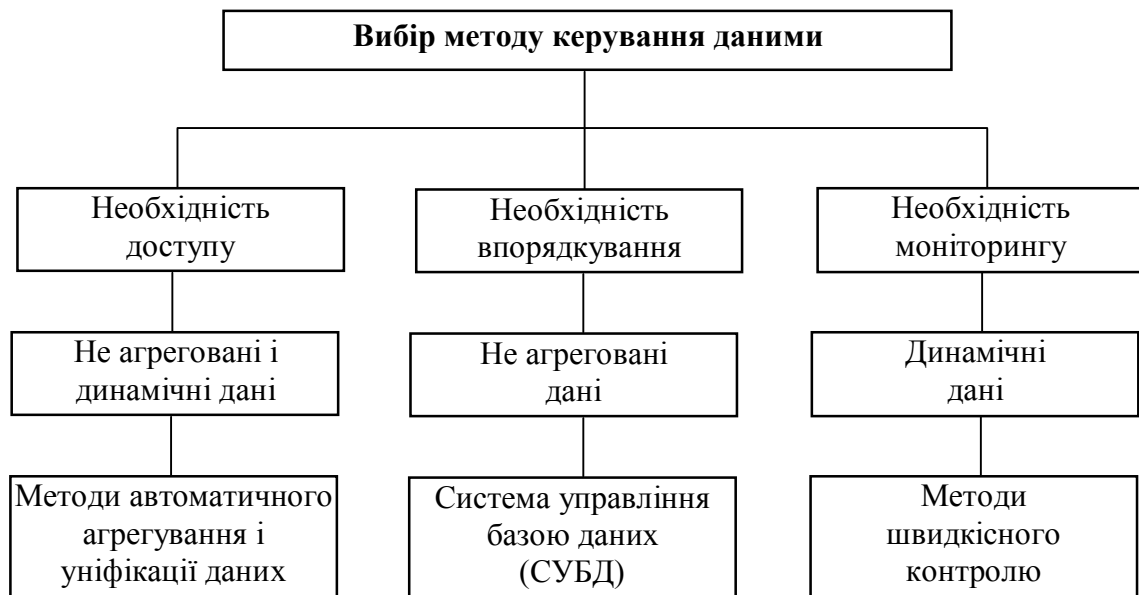


Рис. 3.3 Вибір методу керування даними

Оскільки необхідність керування структурами даних визначається зовнішніми умовами, то правила ґрунтуються на характеристиках інформаційного середовища ситуації з прийняття рішень. Якщо дані змінюються перед кожним сеансом прийняття рішень, то інформаційне середовище називають динамічним [49]. У протилежному випадку його називають статичним.

Окремий випадок менеджменту даних виникає у динамічному інформаційному середовищі, коли ОПР ставить вимогу висвітлити зміни значень множини даних, тобто ставиться вимога моніторингу. У таких випадках застосовують методи швидкісного контролю даних. Вони можуть бути застосовані безпосередньо до агрегованих баз даних або просто до потоків (масивів) даних, що передаються по шинах або мережевих лініях зв'язку.

Інший тип моніторингу даних стосується інтеграції різнотипних масивів даних. Якщо дані надходять до ОПР у вигляді окремих незв'язаних фрагментів з різних масивів у різних формах, то такий формат називають не агрегованим, що призводить до значних труднощів з їх аналізом. Тому необхідно розробляти і впроваджувати в систему спеціальні форми агрегування (об'єднання) різнотипових даних.

Як приклад, можна навести інтегрування даних у системах визначення координат рухомих об'єктів (або нерухомих об'єктів) за допомогою декількох джерел вимірювальної інформації. Такі системи складаються з висотоміра, локаційної станції для визначення азимута і кута місця і, можливо, додаткових супутникових способів визначення координат. Усі дані вимірів надходять у центральну станцію інтегрування й оброблення даних, де приймається остаточне рішення стосовно визначення місця розташування об'єкта, що знаходиться під спостереженням.

Третя проблема моніторингу пов'язана із забезпеченням швидкого і гнучкого доступу до даних. Якщо дані знаходяться в неагрегованому форматі, а існує необхідність організації швидкого доступу до них, то необхідно скористатися системою управління базою даних (СУБД). Існує три загальних типи СУБД. Якщо дані необхідно організувати в ієрархічній формі, то СУБД повинна мати ієрархічну або реляційну форму. Коли виникає необхідність у створенні мережевої форми або асоціативної структури, то СУБД повинна ґрунтуватися на мережевому підході до її побудови. У загальному випадку мережеві структури можна реалізувати на основі реляційного підходу.

Запитання та завдання для самоконтролю

1. Чим зумовлено виділення груп моделей при побудові і експлуатації СППР?
2. Як класифікують моделі процесу прийняття рішень?
3. Що таке слабко і сильно прогнозовані процеси?
4. У чому полягає регулярність і нерегулярність процесів прийняття рішень?
5. Чим відрізняються процеси, прогнозовані за входом/виходом і механістичні детерміновані і стохастичні процеси?
6. Як класифікують моделі за критеріями вибору альтернатив при оцінюванні результату роботи СППР?
7. В чому полягають відмінності аспектного вибору від утилітарного мультиаспектного?
8. Що являють собою фіксовані і адаптивні критерії? Навести приклади.
9. Чим зумовлено поділ критеріїв вибору альтернатив на послідовні і одночасні?
10. Як класифікують правила вибору методу менеджменту?
11. Опишіть процедуру вибору (побудови) моделі досліджуваного процесу при проектуванні СППР.
12. Наведіть можливості вибору моделі (критеріїв) для оцінювання результату роботи СППР.
13. Сформулюйте правила вибору моделі оцінювання стану процесу (об'єкта) для ситуацій утилітарного мультиатрибутного вибору.
14. Вкажіть варіанти методів керування даними та розкрийте особливості їх вибору.

РОЗДІЛ 4 ПРОЕКТУВАННЯ АРХІТЕКТУРИ СППР

Існують різні підходи до проектування СППР. Системний аналіз цих підходів дозволяє узгодити існуючі вимоги до СППР з вибраною архітектурою СППР.

Проектування СППР передбачає визначення типу архітектури і подальшу деталізацію основних функцій системи оброблення даних та генерування результатів (СОДГР) СППР. Розробку системи СОДГР починають з вибору та опису алгоритмів відповідно до методів оброблення даних, форм і засобів подання даних і знань, які можуть використовуватись в СППР.

На завершальній стадії проектування важливими є функції системи подання результатів, форми подання даних і результатів обробки даних тощо. Розгляд цих питань і складає основу проектування архітектури СППР.

4.1 Основні підходи до проектування СППР

4.1.1 Інформаційний підхід

За категорією класифікації – концептуальна модель (схема) виділяють: інформаційний підхід до проектування; підхід, що ґрунтується на знаннях; інструментальний підхід.

З позицій інформаційного підходу СППР належать до класу інформаційних систем, основне призначення яких полягає у поліпшенні характеру діяльності управлінського персоналу підприємства (саме покращення характеру, а не у наданні потрібної інформації у певний час) за рахунок застосування засобів інформаційних технологій. У межах цього

підходу запропоновано дві моделі СППР: «Спрага» та еволюціонуючи модель.

Основними компонентами моделі СППР «Спрага» є: інтерфейс «користувач–система», БД і база моделей. Інтерфейс «користувач-система» забезпечує зв'язок з кожною БД. Він включає програмні засоби: систему управління БД (СУБД), систему управління базою моделей (СУБМ), управління і генерування діалогу. Інтерфейс повинен забезпечити виконання таких функцій: керування різними стилями ведення діалогу; змінювання стилю діалогу за бажанням користувача; подання даних у різних форматах і видах; надання гнучкої підтримки користувачеві.

Структурну схему СППР «Спрага» подано на рис. 4.1.

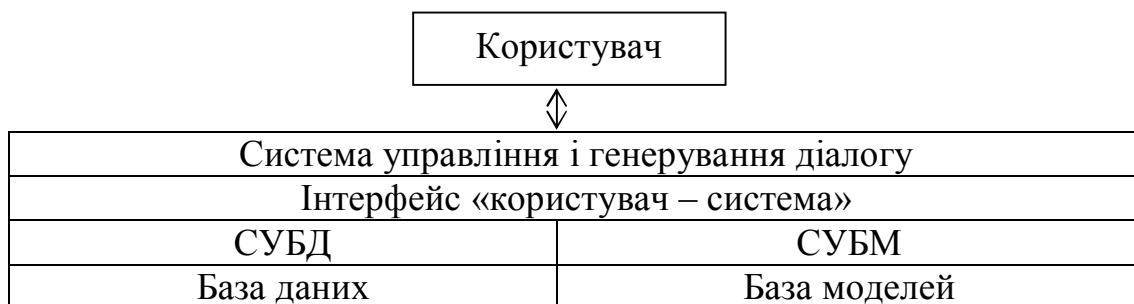


Рис. 4.1 Структурна схема СППР «Спрага»

Бази даних СППР включають як кількісну, так і якісну інформацію, що надходить із різних джерел. Засоби створення і ведення БД повинні надавати такі можливості: об'єднувати різні джерела інформації, використовуючи процедуру їх «добування» даних; представляти логічну структуру у термінах користувача; надавати повний набір функцій управління даними.

База моделей повинна забезпечувати гнучкість моделювання, зокрема, за рахунок використання готових блоків моделей і підпрограм.

Управління моделями дає такі можливості: каталогізувати та обслуговувати широкий спектр моделей, які підтримують всі рівні управління; легко і видно створювати нові моделі; пов'язувати моделі з відповідними БД.

Подальшим розвитком СППР «Спрага» є еволюціонуюча СППР. Крім інтерфейсу користувача, бази даних, бази моделей ця система включає базу текстів і базу правил. Завдяки цьому розширюється їх функціональні можливості. Інформаційна база СППР дає змогу використовувати як менш структуровані види інформації (наприклад, тексти звичайною мовою), так і більш структуровану інформацію (наприклад, правила подання знань, евристичні процедури).

Структурна схема еволюціонуючої СППР зображена на рис. 4.2.

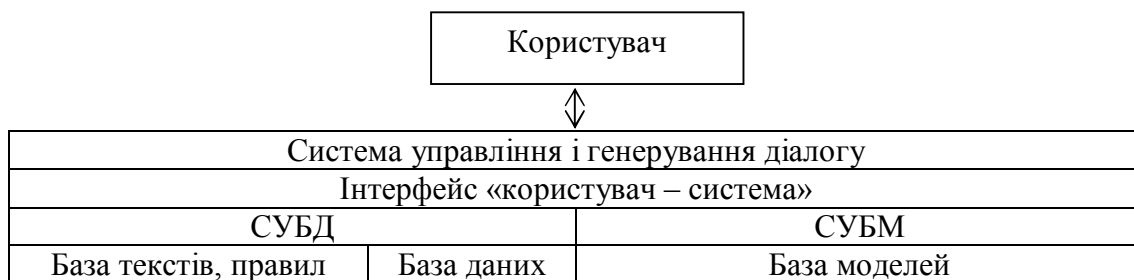


Рис. 4.2 Структурна схема еволюціонуючої СППР

4.1.2 Підхід, заснований на знаннях

Одним із перспективних напрямів розвитку систем підтримки прийняття рішень є об'єднання технологій підтримки рішень і технології штучного інтелекту [12, 13, 39]. Проте у контексті класифікації СППР доцільно розглянути модель СППР, яка ґрунтується на знаннях.

Елементи штучного інтелекту, зокрема використання звичайної мови для спілкування з системою, методологія експертних систем, інженерія

знань і комп'ютерних мов штучного інтелекту знайшла застосування у трьох базових компонентах СППР: БД і СУБД, база моделей і система управління базою моделей, інтерфейсі користувача. Але є концепції створення СППР, в яких система знань в СППР виступає як один з визначальних чинників. Відмінною особливістю СППР, що ґрунтуються на знаннях, є явне виділення нового аспекту підтримки рішень - спроможність «розуміти» проблему, тобто здатність прийняти запит користувача, зібрати відповідну інформацію і підготувати звіт.

Структурну схему СППР, яка ґрунтується на знаннях, зображено на рис. 4.3. Ця система складається з трьох взаємодіючих частин: мовна система (МС); система знань (БД, СУБД, база знань (БЗ) і система управління базою знань (СУБЗ)), і системи оброблення (розв'язання) проблеми (проблемний процесор).

Мовна система забезпечує зв'язок між користувачем і всіма компонентами комп'ютерної системи. За її допомогою користувач формулює проблему і керує процесом її рішення, використовуючи синтаксичні та семантичні засоби, запропоновані мовною системою. Система знань вміщує інформацію стосовно предметної області.

Типи цих систем відрізняються за характером подання в них даних і використаними моделями формалізації знань (ієрархічні структури, графи, семантичні мережі, фрейми, обчислення предикатів тощо).

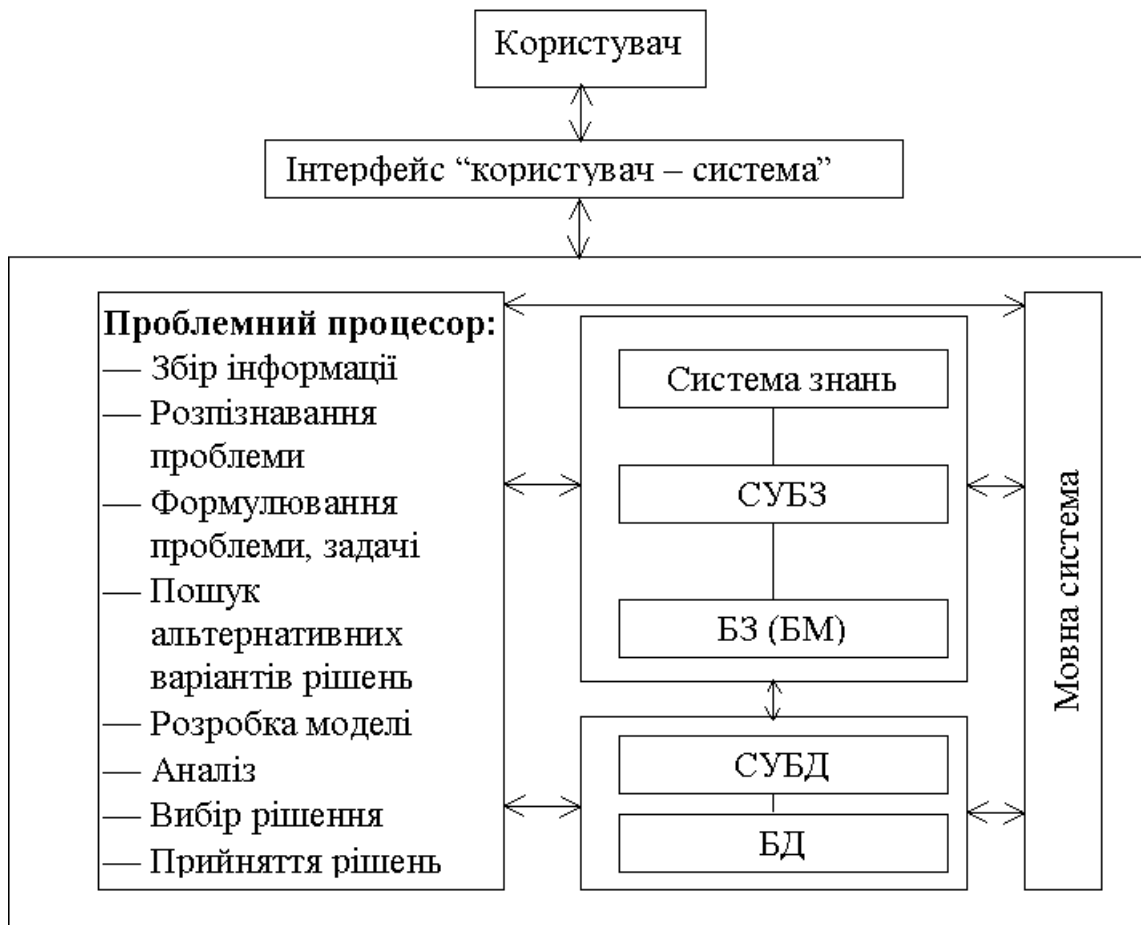


Рис. 4.3 Структурна схема СППР, яка ґрунтується на знаннях

Система оброблення задач – це механізм, який пов’язує мовну систему і систему знань. Цей проблемний процесор забезпечує збір інформації, формулювання моделі, її аналіз тощо. Він сприймає опис проблеми, виконаний відповідно до синтаксису мовної системи, і використовує знання згідно з прийнятими у системі знань правилами з метою створення інформації, необхідної для підтримки рішень.

Проблемний процесор – це динамічна компонента СППР, що відображає (моделює) поведінку особи, яка вирішує проблему. Тому він повинен мати як мінімум, можливість інтегрувати інформацію, що надходить від користувача через мовну систему і систему знань.

Проблемний процесор з використанням математичних моделей може перетворювати формулювання проблеми у докладні процедури, виконання яких дає відповідь (розв'язок задачі). У складніших випадках проблемний процесор повинен вміти формулювати моделі, необхідні для вирішення поставленої проблеми.

4.1.3 Інструментальний підхід

Підвищена увага представників інформатики та економічної практики до методів розробки і впровадження СППР зумовила необхідність розробки програмних інструментів для створення СППР. Це вплинуло на появу нової концепції класифікації СППР - інструментального підходу. В залежності від специфіки розв'язуваних задач і використовуваних технологічних засобів процесу створення систем можна виділити три рівні СППР:

- спеціалізовані (прикладні) СППР;
- генератори СППР (СППР-генератори);
- інструментарій СППР (СППР-інструментарій).

Спеціалізовані СППР призначені для використання окремим користувачем або групами користувачів. Вони дають змогу індивідуальному ОПР чи колективу ОПР вирішувати специфічні проблеми у конкретних ситуаціях.

СППР-генератор – це пакет взаємопов'язаних програмних засобів (пошуку, переробки й видачі даних, моделювання тощо), який дає змогу легко і швидко створювати спеціалізовану СППР. Прикладом може бути інформаційна керуюча система, яка складається з різноманітних елементів: пошуку інформації, підготовки звітів, мови моделювання, а також множини засобів для виконання фінансових та статистичних аналізів.

Оскільки генератори СППР можуть використовуватись і не програмістами, для створення систем підтримки прийняття рішень в області планування й управління розроблено чимало СППР-генераторів: CUFFS88, EXPRESS, FAME тощо.

Концептуальна структура СППР-генератора, яка відображає точку зору користувача, включає п'ять компонент: управління інтерфейсом користувача, управління поданнями даних і результатів, управління аналізом, системне управління, управління даними.

Управління інтерфейсом користувача повинне забезпечувати реалізацію трьох основних типів інтерфейсу: меню, мова команд, звичайна мова запитань і відповідей.

Управління поданнями повинне підтримувати різноаспектні образи користувача стосовно своєї проблеми, яку потрібно вирішувати. Ці подання можуть відображатись у вигляді таблиць, графіків або командних процедур.

Управління аналізом даних зводиться до ведення бази моделей. У разі маніпулювання даними при математичному моделюванні множини інструкцій можна подати як підпрограму аналізу. СУБД повинна забезпечувати доповнення бази моделей за рахунок додаткових засобів аналізу. Системний адміністратор забезпечує координацію дій користувачів, а також системного тренажера, що використовується для підготовки користувачів.

Управління даними реалізується за допомогою СУБД, яка повинна містити засоби ведення словника даних, що дасть змогу створити на цій основі інші словники, наприклад, графічний словник чи словник моделей.

До прототипу описаного генератора можна зарахувати систему REGIMES, орієнтовану на персональні комп'ютери. Цей генератор складається з таких компонент: командний процесор, діалоговий процесор,

процесор подання результатів, підсистема управління регресійним аналізом, а також три словники.

СППР-інструментарій надає в розпорядження проєктантів СППР потужні засоби, в тому числі нові мови спеціалізованої спрямованості, удосконалені операційні системи, засоби обміну інформацією, проєкції кольорових графічних образів та ін. Тому вони можуть використовуватись для створення як спеціалізованих СППР, так і для генераторів СППР.

4.2 Типи архітектур спеціалізованих СППР

4.2.1 Текстово-орієнтовані СППР

Виділяють так основні архітектури спеціалізованих СППР: текстово-орієнтовані СППР, СППР, які орієнтовані на використання бази даних, СППР, які орієнтовані на використання електронних таблиць (ЕТ), наприклад, Excel.

Такі системи створювались на початку 70-х років і містили в собі:

- мовну систему (МС) для взаємодії користувача з системою;
- базу знань та даних (БЗД), яка складається з текстових файлів, що являють собою інформацію для особи, яка приймає рішення (ОПР); це так звана електронна документація;
- систему оброблення даних та генерування результатів (СОДГР), яка виконує різноманітні маніпуляції над текстовою документацією і включає програмне забезпечення (ПЗ), що полегшує користувачеві складання запитів;
- систему подання результатів (СПР), яка складає всі можливі формати подання текстової, табличної і графічної інформації та повідомлення, які полегшують користувачеві спілкування з СППР.

Приклад. Текстова СППР (рис. 4.4) для інженера - технолога. Якщо необхідно розпочати виробництво технічно складної деталі або вузла, то при цьому виникає необхідність прийняти рішень стосовно запитань:

- які параметри повинен мати новий продукт?
- скільки буде коштувати його виробництво?
- які інструменти, станки і т.д. використовувати?
- яка площа необхідна для обладнання?
- яку, звідкіля і в якій кількості замовити сировину?
- які характеристики має подібний продукт конкурента?
- які проблеми необхідно врахувати стосовно охорони праці, набору додаткових робітників, обслуговування нової техніки?
- як організувати збут продукції?
- на який об'єм продажу і прибутку можна розраховувати?



Рис. 4.4 Структура текстової СППР

Ще одна 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.

4.2.2 СППР з використанням бази даних

Ще один окремий випадок СППР. Найбільш поширені – це реляційні бази даних, тобто в цьому випадку йде оброблення строго структурованих знань у вигляді числових і описових даних (рис. 4.5).



Рис. 4.5 Структура СППР на основі БД

В такій системі СОДГР включає три типи програмного забезпечення (ПЗ):

- ПЗ для СУБД;
- інтерактивне ПЗ для оброблення запитів;
- спеціальне ПЗ, яке створюється для задоволення потреб користувача (включає, зазвичай, деяку логіку стосовно аналізу даних і формування відповідей на запити, а також необхідні обчислення: статистичні розрахунки, оцінювання параметрів моделей і прогнозів, порівняння отриманих результатів).

4.2.3 СППР на основі використання електронних таблиць

При використанні технології на основі електронних таблиць (наприклад, типу Excel) для управління знаннями користувач СППР не тільки може створити, проглянути і модифікувати процедурні знання в БЗД, але може дати запит СОДГР виконати команди, які там містяться.

Такі СППР мають такі характеристики:

- БЗД містить файли з таблицями, які наповнені описовими і процедурними знаннями;
- СОДГР може виконувати алгоритмічні процедури, тобто аналіз типу «що буде, якщо ...».

Архітектура СППР на основі ЕТ зображена на рис. 4.6. Формули, які містяться в таблиці, являють собою основу процедурних знань, які і є командами для СОДГР. Такі СППР, зазвичай, призначені для виконання аналізу «що буде, якщо ...» для того, щоб побачити і порівняти результати дій над вмістом таблиць. Це засіб для прискореного порівняння альтернатив.

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

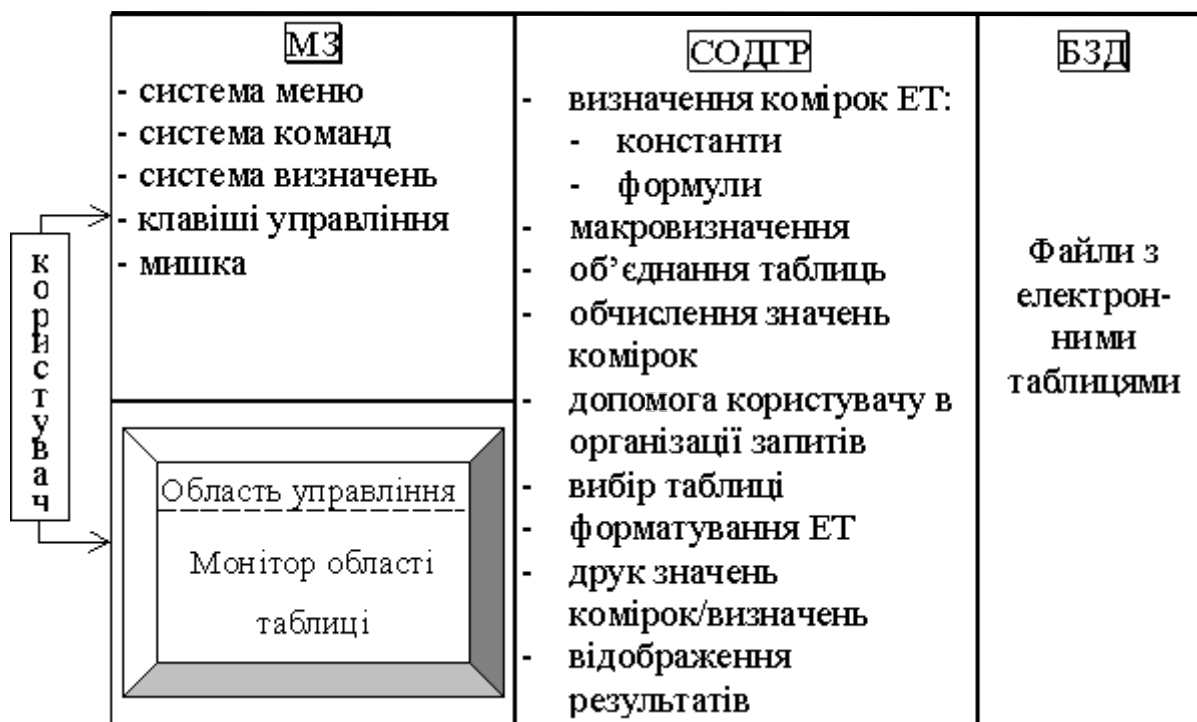


Рис. 4.6 Архітектура СППР на основі ЕТ

4.2.4 СППР на основі алгоритмічних процедур для розв'язання задач

Це може бути, наприклад, СППР для розв'язання задачі оптимізації інвестицій, або максимізації прибутку для конкретного виду виробництва, оптимального розміщення централізованих складів для торгової мережі тощо. Зазвичай, СППР такого типу містить множину алгоритмів для розв'язання вибраного класу задач. Набори алгоритмів для розв'язання конкретних задач фокусуються на задачах таких класів:

- аналіз фінансово-економічних процесів;
- прогнозування процесів довільної природи;
- планування (оперативне, тактичне і стратегічне);
- статистичний аналіз даних в різних галузях;

– розв’язування статистичних і динамічних оптимізаційних задач.

Існує два основних підходи до використання обчислювальних алгоритмів в СППР: (1) фіксований; (2) гнучкий.



Рис. 4.7 Структура СППР з фіксованим використанням алгоритмів

При фіксованому підході алгоритми є частиною СОДГР, а це означає, що до СППР не можна легко додати (видалити) нові алгоритми або модифікувати її в цілому. При гнучкому підході можна легко додавати, видалити, модифікувати і об’єднувати множину алгоритмів у процесі експлуатації СППР завдяки модульній структурі системи.

При цьому БЗ може містити дані, постановки задач, формати звітів, правила прийняття рішень з метою координації виконання алгоритмічних модулів. Розглянемо більш докладно систему з фіксованим використанням алгоритмів. Структура такої системи наведена вище на рис. 4.7.

Структура гнучкої СППР на основі алгоритмічних процедур наведена на рис. 4.8.



Рис. 4.8 Структура гнучкої СППР на основі алгоритмічних процедур

Відмінність гнучкої системи від жорсткої полягає у тому, що БЗ містить алгоритмічні модулі, які можна комбінувати в необхідній послідовності та запускати на виконання. Крім того, гнучка система передбачає можливість формування нових та модифікації існуючих критеріїв аналізу якості розв'язку задачі.

4.2.5 СППР на основі правил

В СППР на основі правил (рис. 4.9) в БЗД зберігаються правила та описи станів процесу, а СОДГР містить правила формування висновку для формування логічного висновку на основі правил типу [13]:

If < описання ситуації >;
 Then < які дії виконати >;
 Because < вказівка причин дій >.

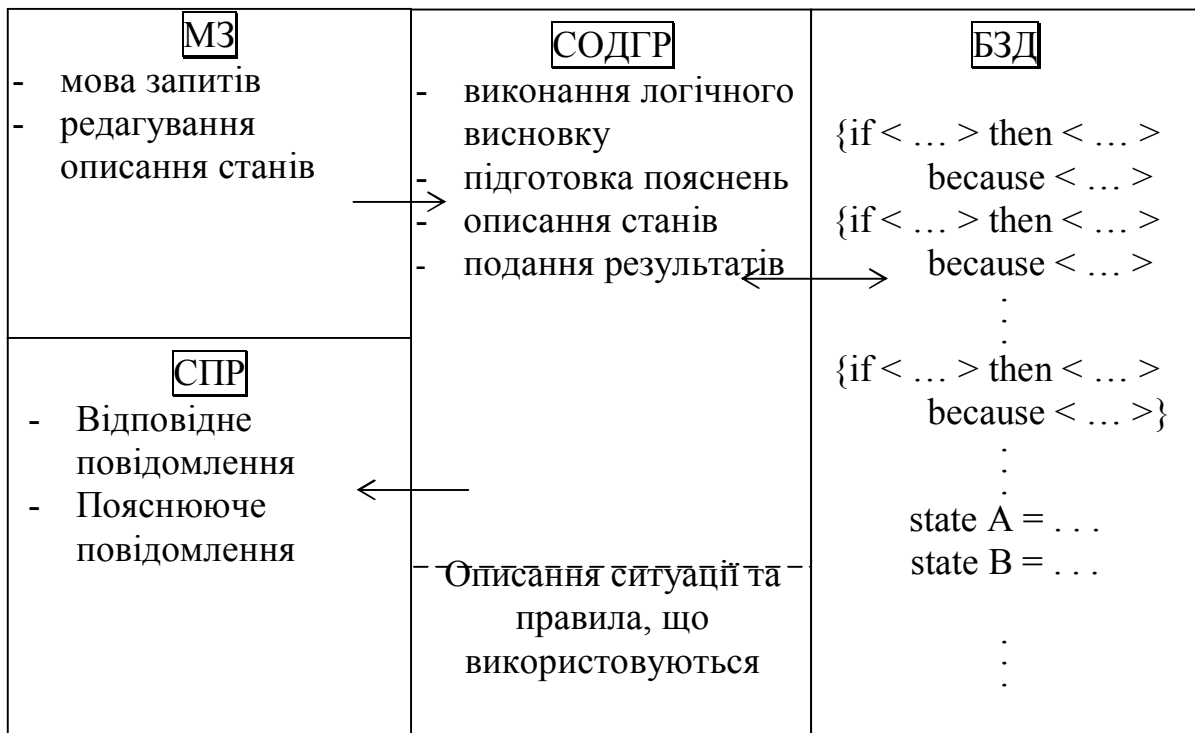


Рис. 4.9 Структура СППР на основі правил

Кожний набір правил в БЗ створюється для того, щоб отримати рекомендації стосовно розв'язання конкретної проблеми. Наприклад, один набір правил може стосуватися того, як уникнути випуску продукції з

дефектами, а інший – надає пораду як краще організувати продаж готової продукції.

Описання поточного стану може стосуватися:

- поточного рівня технології на підприємстві;
- розміщення складів для розповсюдження товару на території України;
- кількості та кваліфікації спеціалістів на підприємстві.

Окрім запитів про допомогу (Help) та запитів щодо редагування опису станів, користувач СППР на основі правил може давати два типи запитів стосовно підтримки прийняття рішень:

- запити щодо рекомендацій (порад):
- запити щодо пояснення фактів.

Наприклад, користувач може дати запит відносно можливих причин відмови механічного вузла, а потім дати запит щодо докладного пояснення старіння або виробленості металевих деталей.

4.2.6 Гібридні СППР

Вище ми розглянули окремо такі СППР: текстові, на основі БД, на основі електронних таблиць, на основі алгоритмічних процедур та на основі правил. Якщо об'єднати кілька типів СППР в одну, то отримаємо гібридну систему. Часто об'єднують системи на основі БД та алгоритмічних процедур. Структура такої СППР наведена на рис. 4.10.

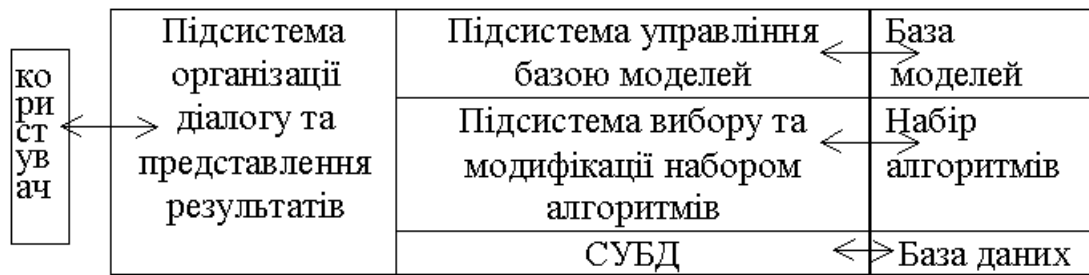


Рис. 4.10 Структура гібридної СППР

Очевидно, що такі системи є функціонально універсальнішими і дають більше можливостей з точки зору запитів і отримуваних результатів. При цьому СППР може сама приймати рішення про те, який метод оброблення знань вибирати.

Можливий запит: “Показати середній приріст промислового виробництва для південних районів України».

Розглянемо загальну архітектуру гібридної СППР (рис. 4.11).

За таким типом побудована СППР при управлінні повітряним рухом у великих аеропортах. Вона має такі функції:

- чисто довідкові;
- розрахунок оптимальної траєкторії польоту в аварійних ситуаціях;
- автоматизований гнучкий розклад посадок і зльотів;
- СППР як радник у нештатних ситуаціях;
- стан парку літаків, приписаних до аеропорту;
- довідки стосовно кадрів (персоналу);
- довідки щодо матеріально-енергетичних ресурсів;
- інформація про нештатні ситуації в минулому;
- інформація щодо прогнозу погоди.

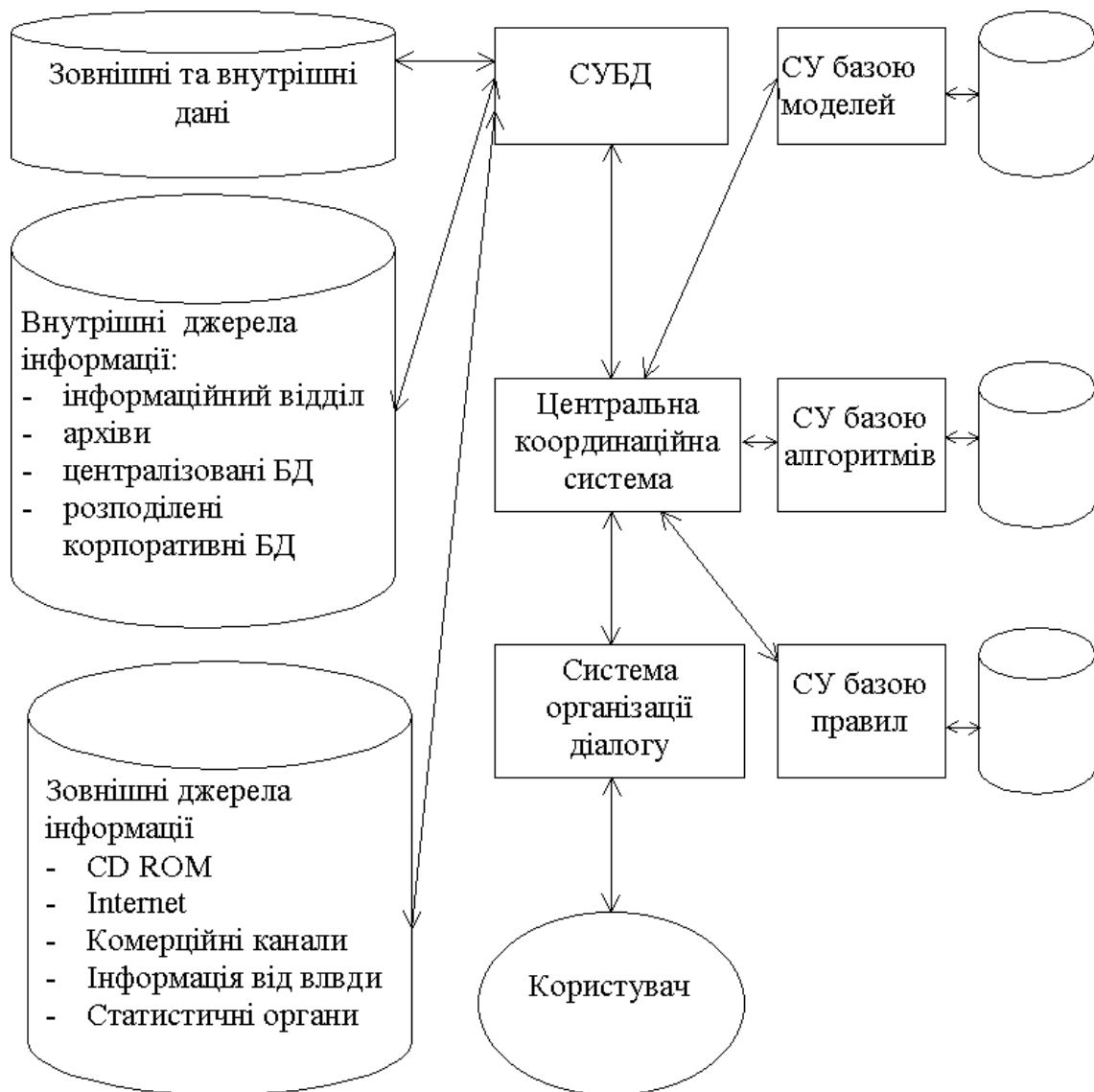


Рис. 4.11 Узагальнена архітектура гібридної СППР

4.3 Функції системи оброблення даних та генерування результатів

Розробка системи оброблення 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.

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

- поповнює (при необхідності) базу даних і знань;
- вибирає алгоритм (алгоритми) оброблення даних з БДі 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 тощо) 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 подання 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.

СППР, яка проектується для підтримки прийняття рішень в іншій прикладній області, буде мати інші функції, хоча досить часто СППР мають ряд подібних функцій. Наприклад, попереднє оброблення даних, побудова математичних та статистичних моделей, визначення їх адекватності, перетворення до стандартизованих форм (простір станів), знаходження розв'язків відповідних рівнянь, отримання функцій прогнозування тощо.

4.4 Вибір та описання алгоритмів, на яких базується СОДГР

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 оброблення даних (логарифмування, нормування, фільтрація, заповнення пропусків, оброблення 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 МГУА; за допомогою фільтра Калмана; за методом подібних траєкторій; методами експоненціального згладжування; нечітких множин; нейронних мереж тощо);
- 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 БДі знань.

Якщо СППР призначена для обчислення (прогнозування) оптимальних траєкторій розвитку того чи іншого процесу (тобто, оптимального керування процесом), то вона повинна містити такі алгоритми:

- алгоритми попередньої оброблення даних (логарифмування, нормування, фільтрація, заповнення пропусків, оброблення великих імпульсних значень);
- оцінювання параметрів (коефіцієнтів) математичних та/або статистичних моделей;
- алгоритми обчислення статистичних параметрів, які характеризують якість (адекватність) побудованої моделі;
- алгоритм перетворення побудованої моделі в стандартизовану форму простору станів;
- алгоритми обчислення оптимальних траєкторій розвитку процесу, тобто траєкторій керуючих дій (сигналів) і траєкторій для вихідних змінних (змінних стану) процесу; при цьому постановка задачі оптимального керування вимагає таких знань: математичну модель процесу, початкові умови, обмеження на змінні та критерій оптимальності;
- алгоритми обчислення показників якості процесу керування (прогнозування оптимальних траєкторій);

- алгоритм(и) (правила) вибору кращої траєкторії з множини знайдених.

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 тощо.

4.5 Дані і знання систем підтримки прийняття рішень

4.5.1 Характеристики даних і знань

При вивченні інформаційних СППР актуальним є питання: що таке дані і знання та в чому полягає різниця між ними? Існують різні визначення даних, які принципово не відрізняються між собою. Зокрема, у якості одного з таких визначень даних можна вважати [12, 41]: дані – це окремі факти, які характеризують об'єкти, процеси і явища предметної області та їх властивості.

У процесі комп'ютерної оброблення дані умовно перебувають у певних формах і проходять наступні етапи отримання, зберігання і перетворення:

D1 – дані як результат вимірів та спостережень;

D2 – дані на матеріальних носіях інформації (таблиці, протоколи, довідники);

D3 – моделі (структури) даних у вигляді діаграм, графіків, функцій;

D4 – дані у комп'ютері, подані мовою опису даних;

D5 – БД на машинних носіях інформації.

Знання можна розділити на теоретичні і практичні (емпіричні). Емпіричні знання ґрунтуються на даних, отриманих емпіричним шляхом. Вони являють собою результат розумової діяльності людини, спрямованої на узагальнення його досвіду, набутого в результаті практичної діяльності [12].

Знання – це закономірності предметної області (принципи, зв'язки, закони тощо), отримані у результаті теоретичної і практичної діяльності, а також професійного досвіду, які дозволяють робити постановки задач і розв'язувати їх в даній конкретній області.

В процесі створення та оброблення знання трансформуються аналогічно до даних:

Z1 – знання в пам'яті людини як результат осмислення фактів, явищ, процесів та їх взаємодії;

Z2 – зберігання знань на матеріальних носіях (підручники, посібники, монографії, довідники);

Z3 – поле знань, як умовний опис основних об'єктів предметної області, їх атрибутів та закономірностей функціонування, що зв'язують їх;

Z4 – кодування і зберігання знань на мовах подання знань (продукційні мови, семантичні мережі, фрейми);

Z5 – база знань на машинних носіях інформації.

Часто використовують ще таке визначення знань: знання – це добре структуровані дані, або дані щодо даних, тобто метадані.

4.5.2 Моделі подання знань

Існує багато моделей (або мов) подання знань для різних предметних галузей. Однак більшість з них можна звести до таких класів:

– продукційні моделі;

- семантичні мережі;
- фрейми;
- формальні логічні моделі;
- процедурні моделі.

4.5.2.1 Продукційна модель

Продукційна модель – це модель, яка ґрунтується на правилах і дозволяє подати знання у вигляді правил типу:

Якщо < умова > то < дія > .

«Умова» – це деяке речення-зразок, за допомогою якого виконується пошук в базі знань, а «дія» – це той наслідок або дія, яку необхідно виконати в результаті успішного результату пошуку (дії можуть бути проміжними і виступати надалі як умови), або ж дії можуть бути термінальними чи цільовими, що завершують роботу продукційної системи.

Висновок на основі такої бази знань може бути прямим (від даних до пошуку мети) або зворотним (від мети до її підтвердження – до даних).

Дані – це початкові факти, що зберігаються в базі фактів (даних), і використовуються машиною формування висновку в процесі перебору правил з продукційної бази знань.

Продукційна модель даних характеризується наочністю, високою модульністю, простотою внесення доповнень та змін, а також чіткістю механізму формування логічного висновку.

4.5.2.2 Семантичні мережі

У семантичних моделях застосовують термін семантичний, який означає смисловий. Семантика – це наука, яка встановлює відношення між символами і об'єктами, які вони позначають; тобто, це наука, яка визначає смисл знаків (символів).

У продукційних моделях застосовують семантичні мережі. Семантична мережа – це орієнтований граф, вершинами якого є поняття, а дуги являють собою відношення між ними.

Поняття зазвичай подають як абстрактні або конкретні об'єкти, а відношення, як зв'язки таких типів: «це» («a kind of» = АКО, «is»), «складає частину» («has a part»), «належить», «любить».

Характерною особливістю семантичних мереж є обов'язкова наявність таких трьох типів відношень:

- клас – елемент класу (дерево – береза);
- властивість – значення (колір – синій);
- приклад елемента класу (роза – чайна).

Можна назвати декілька напрямів класифікації семантичних мереж, зв'язаних з типами відношень між поняттями.

За числом типів відношень:

- однорідні (з одним типом відношень);
- неоднорідні (з різними типами відношень);

За типами відношень:

- бінарні (в яких відношення зв'язують два об'єкти);
- *N*-арні (в яких є спеціальні відношення, що зв'язують більше двох понять);

Найчастіше у семантичних мережах використовують такі відношення:

- зв’язок типу «частина – ціле» («клас – підклас», «елемент – множина» тощо.);
- функціональні зв’язки (визначаються, зазвичай, дієсловами «впливає», «створює», ...);
- кількісні зв’язки (більше, менше, дорівнює, ...);
- просторові зв’язки (далеко, близько, за, під, над, ...);
- часові зв’язки (раніше, пізніше, на протязі, ...);
- атрибутивні зв’язки (мати властивість, мати значення);
- логічні зв’язки (І, АБО, НІ);
- лінгвістичні зв’язки та інші.

Проблема пошуку розв’язку у базі знань типу семантичної мережі зводиться до задачі пошуку фрагмента мережі, який відповідає деякій підмережі, що відображає поставлений запит до бази. Приклад семантичної мережі показано на рис. 4.12.

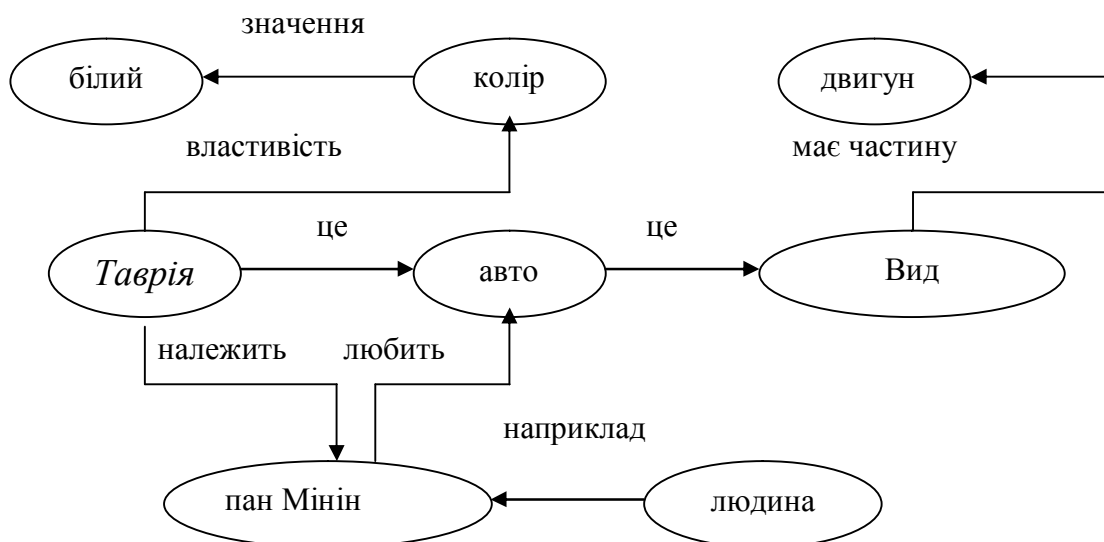


Рис. 4.12 Приклад семантичної мережі

Вершинами тут виступають поняття «людина», «пан Мінін», «Таврія», «автомобіль», «вид транспорту» і «двигун». Така модель запропонована американським психологом Квінлліаном. Перевагою даної моделі є те, що вона краще інших відповідає сучасним уявленням про організацію довгострокової пам'яті людини. Недолік моделі полягає у складності організації процедури пошуку висновку на семантичній мережі. Для реалізації семантичних мереж створено спеціальні мови: NET, SIMER+MIR та інші.

Широко відомі експертні системи, які використовують семантичні мережі як мову подання знань – PROSPECTOR, CASNET, TORUS.

4.5.2.3 Фрейми

Термін фрейм (від англійського frame – каркас, рамка) запропонований в 70-х роках 20-го століття для позначення структури знань, пов'язаних із просторовими сценами. Так само як і семантична мережа, ця модель має психологічне обґрунтування. Фрейм – це абстрактний образ для подання деякого стереотипу мислення.

У психології і філософії відоме поняття абстрактного образу. Наприклад, коли промовляють вголос слово «кімната», воно породжує образ кімнати – «приміщення з чотирьох стінами, підлогою, стелею, вікнами та дверями, площею 12-20 квадратних метрів». З цього опису не можна нічого викинути. Наприклад, якщо викинути вікна, то вже буде не кімната, а комора. Однак, в цьому опису є «отвори» або слоти (або «слоти» – slots). Слот – це незаповнені значення деяких атрибутів, наприклад, кількість вікон, колір стін, висота стелі, покриття підлоги та інші.

В теорії фреймів такий образ кімнати називають фреймом кімнати. Фреймом називають, також, формалізовану модель для відображення образу.

Розрізняють фрейми-зразки або прототипи, які зберігаються у базі знань, і фрейми-екземпляри, які створюють для відображення реальних ситуацій на основі фактичних даних.

Модель фрейма є досить універсальною, оскільки вона дає можливість відобразити все розмаїття знань про світ через такі елементи:

- фрейми-структури, що використовуються для позначення об'єктів та понять (вексель, позика, застава);
- фрейми-ролі (менеджер, касир, клієнт);
- фрейми-сценарії (банкротство, збори акціонерів, святкування іменин);
- фрейми-ситуації (тривога, аварія, робочий режим машини) та інші.

Традиційно структуру фрейма можна подати у вигляді переліку його властивостей таким чином:

(ІМ'Я ФРЕЙМА:

(ім'я 1-го слота: значення 1-го слота),

(ім'я 2-го слота: значення 2-го слота),

...

(ім'я N -го слота: значення N -го слота)).

Таку ж структуру можна подати у вигляді таблиці 4.1 з двома додатковими стовпчиками:

Таблиця 4.1 – Структура фрейма

Ім'я фрейма			
Ім'я слота	Значення слота	Спосіб отримання значення	Приєднана процедура

Додаткові стовпчики таблиці призначені для опису способу отримання слотом його значення та можливого приєднання до слота спеціальних процедур, що допускається теорією фреймів. Значенням слота може бути ім'ям іншого фрейма. В результаті можна утворити мережі фреймів.

Є кілька способів отримання слотом значень у фреймі-екземплярі:

- стандартизоване значення (default);
- через успадкування властивостей від фрейма, вказаного в слоті ЦЕ (a kind of = АКО = ЦЕ);
- за формулою, вказаною у слоті;
- через приєднану процедуру;
- явно через діалог з користувачем;
- з бази даних.

Важливою властивістю фреймів є запозичення з теорії семантичних мереж так званого успадкування властивостей. В фреймах і у семантичних мережах успадкування реалізується через зв'язки типу АКО. Слот АКО вказує на фрейм більш високого рівня ієрархії, звідки неявно успадковуються, тобто переносяться, значення аналогічних слотів.

4.5.2.4 Формальні логічні моделі

Традиційно у представленні знань виділяють формальні логічні моделі, які ґрунтуються на класичному численні предикатів 1-го порядку, коли предметна область або задача описується у вигляді аксіом. Однак числення предикатів 1-го порядку на практиці застосовується надзвичайно рідко внаслідок того, що воно висуває занадто високі вимоги до предметної області і накладає на неї сильні обмеження.

4.5.2.5 Процедурні моделі

Процедурні моделі знань – це алгоритми, функції, формули та алгебраїчні вирази, які використовують у процесі побудови математичних моделей, визначення адекватності, обчислення прогнозів та критеріїв, задавання обмежень різного типу тощо. Наприклад, база знань може містити десятки різних алгоритмів обчислення параметрів моделей, які використовують в залежності від конкретного запиту користувача.

4.6 Функції системи подання результатів

4.6.1 Форма подання результатів

Система подання результатів (СПР) приймає результати оброблення даних та знань від системи оброблення даних та генерації результатів і надає їх користувачу у зручній для сприйняття формі. Зручність застосування СПР може суттєво від від технічних засобів, форми подання і виконання ергономічних вимо обробки і подання дани.

Можливі форми подання результатів роботи СОДГР залежать від конкретної прикладної галузі та характеру задач, що розв'язуються за

допомогою СППР. Однак можна виділити кілька відносно універсальних форм подання результатів, до яких відносяться такі:

- гістограми, графічне зображення функцій розподілу ймовірностей отриманих величин;
- графічне подання результатів у вигляді двовимірних та тривимірних графіків;
- застосування ліній різних типів для різних змінних на графіках;
- одночасне зображення поточних результатів роботи СППР та результатів, отриманих раніше – ретроспективне порівняння результатів;
- подання результатів у вигляді таблиць зручного формату;
- поєднання тексту з цифровим матеріалом, таблицями і графіками;
- кругові та стовпчикові діаграми;
- тривимірні стовпчикові діаграми;
- використання можливостей гортання сторінок (paging = пейджинг) та переміщення змісту екрану на один рядок за допомогою функції скролінг (scrolling).

В проекті СППР необхідно описати окремо спеціальні способи подання інформації, які сприяють прискоренню та поглибленню її сприйняття. Деякі з них розглянемо нижче.

4.6.2 Використання кольорів, мигання і клавіатури

Мультимедійне подання даних засовують у багатьох СППР. Зорова інформація може суттєво впливати на якість прийняття рішень. Зокрема, кольори використовують для таких цілей:

- «піднімання» конкретних повідомлень;
- для зображення кластерів даних;

- виділення цифрових даних, графіків або областей екрану.

Сучасні монітори забезпечують можливості використання досить широкого набору відео- та аудіо-атрибутів, наприклад таких:

- подвійна яскравість вибраних полів екрану або окремих повідомлень;
- мигання вибраних полів або повідомлень;
- приховування відображення вибраних полів (наприклад, пароллю);
- інверсне відображення вибраних полів, повідомлень або областей екрану.

4.6.3 Використання клавіатури

Існує значна кількість базових і спеціалізованих клавіатур, які спрощують інтерактивну взаємною ОПР з СППР. Клавіатура містить досить широкий набір функціональних клавіш (F1, ..., F12), Alt (alternative), Esc (escape) та інші. Ці клавіші можна використати для ініціалізації виконання деяких загальних операцій, які часто повторюються в процесі взаємодії з СППР. Наприклад, до таких клавіш належать START, HELP, PAGE UP, PAGE DOWN, EXIT.

У деяких стандартах обов'язковим є використання клавіші F1 для реалізації функції отримання допомоги (HELP). Очевидно, що одні й ті ж функціональні клавіші повинні завжди використовуватись для однієї мети.

Вміле поєднання вказаних атрибутів дозволяє суттєво підвищити швидкість сприйняття та глибину розуміння результатів роботи СППР, прискорити процеси введення/виведення даних.

Запитання та завдання для самоконтролю

1. Пояснити особливості інформаційного підходу до проектування СППР.
2. Дати характеристику основних компонентів СППР при інформаційному підході.
3. Пояснити особливості проектування СППР при підході, заснованому на знаннях.
4. Які основні функції компонентів структурної схеми СППР, яка ґрунтується на знаннях?
5. Які типи СППР виділяють при інструментальному підході?
6. Дати характеристику типів архітектур спеціалізованих СППР.
7. Які особливості мають текстово-орієнтовані СППР?
8. Які особливості мають СППР, орієнтовані на використання бази даних?
9. Дати характеристику СППР, орієнтованих на використання електронних таблиць.
10. Що являють собою СППР на основі алгоритмічних процедур для розв'язання задач?
11. Дати характеристику структури СППР на основі алгоритмічних процедур для розв'язання задач.
12. Що являють собою СППР на основі правил?
13. Які особливості мають гібридні СППР?
14. Дати характеристику узагальненої архітектура гібридної СППР.
15. Які алгоритми використовують у СППР, призначених для прогнозування?

16. Які алгоритми використовують у СППР, призначених для обчислення (прогнозування) оптимальних траєкторій розвитку?
17. Дати характеристику даних і знань СППР.
18. Які існують моделі подання знань в СППР?
19. Що являють собою продукційні моделі СППР?
20. Що являють собою семантичні мережі?
21. Які типи відношень застосовують у семантичних мережах?
22. Навести основні характеристики відношень у семантичних мережах.
23. Що являють собою фрейми?
24. Як розрізняють фрейми за елементами?
25. Описати стандартну структуру фрейма.
26. Які існують способи отримання слотом значень у фреймі-екземплярі?
27. Що являють собою формальні логічні моделі?
28. Описати основні особливості процедурних моделей.
29. Якими є функції системи подання результатів СППР?
30. Опишіть існуючі підходи до проектування концептуальної моделі СППР.
31. Поясніть особливості інформаційного підходу та підходу ґрунованого на знаннях до проектування СППР.
32. Надайте класифікацію типів архітектури спеціалізованих СППР.
33. Побудуйте узагальнену архітектуру гібридної СППР; наведіть приклад такої системи.
34. Сформулюйте функції системи обробки даних та генерування результатів.

35. Вкажіть варіанти вибору обчислювальних процедур для різних видів СППР.

36. Дайте визначення поняттям дані і знання. Поясніть, у чому полягають відмінності між ними; наведіть приклади знань та способи їх зберігання у базі даних і знань.

37. Наведіть типи моделей подання знань.

38. Сформулюйте функції системи подання результатів.

39. Надайте опис основних можливих форм подання результатів функціонування інформаційної СППР.

РОЗДІЛ 5 ПРОЕКТУВАННЯ ІНТЕРФЕЙСУ КОРИСТУВАЧА

5.1 Вимоги до інтерфейсів інформаційних систем

Під інтерфейсом взаємодії розуміють сукупність домовленостей про форми, способи, процеси, правила взаємодії користувача з ПЕОМ [17, 46].

Інтерфейс визначає основні можливості застосування СППР з точки зору користувача. Тому значної уваги у процесі проектування СППР приділяють коректному формулюванню вимог до інтерфейсу СППР як інформаційної системи та апаратно-програмних особливостей реалізації.

Оскільки якість процесу інтерактивної взаємодії користувача із системою (швидкість, зручність, низький рівень втоми) пов'язана з такими психологічними характеристиками людини як короткострокова та середньострокова пам'ять, час реакції, можливості сприйняття візуальної інформації, то при розробці інтерфейсу необхідно пам'ятати, що [17, 46]:

- інтерфейс – сама важлива частина СППР з точки зору її реклами з метою продажу і з точки зору безпосереднього користувача системи, який може працювати з нею по декілька годин поспіль;
- інтерфейс впливає на характер рішень, які приймає ОПР, він може прискорювати час прийняття рішення та покращувати або погіршувати їх якість;
- який саме конкретний тип інтерфейсу можна створити за допомогою вибраних інструментальних засобів і які принципові можливості може надати інструментальна система.

Основними вимогами до інтерфейсу [46] є: адаптованість, достатність, дружність, гнучкість інтерфейсу.

Адаптованість означає, що інтерфейс повинен бути:

- сумісним з потребами та можливостями користувача;

- забезпечувати простоту переходу від виконання однієї функції до іншої;
- забезпечувати користувача на високому рівні вказівками стосовно його можливих дій, а також генерувати належний зворотний зв'язок на його запити;
- надавати користувачу можливість відчувати себе повноправним керівником ситуації при розв'язанні всіх типів задач, тобто, забезпечувати його всією необхідною інформацією; користувач повинен бути впевненим, що він сам розв'язує поставлену задачу;
- забезпечувати користувача різними, взаємно доповнюючими формами подання результатів в залежності від типу запиту або від характеру отриманого рішення;
- враховувати особливості користувачів різних рівнів; наприклад, для керівника підприємства зручнішим є узагальнене графічне подання результатів роботи СППР у вигляді діаграм та графіків, а інженеру-економісту потрібні конкретні цифри у їх часовій послідовності.

Достатність інтерфейса означає таке:

- допустимі запити користувача повинні бути чіткими і однозначними для користувачів всіх рівнів, а також для прикладних задач всіх типів;
- реакція системи на всі типи запитів також повинна бути однозначною і зрозумілою і, по можливості, простою.

Дружність інтерфейсу полягає у максимальній простоті його використання і готовність в повній мірі задовольнити запитам користувача при розв'язанні визначеного класу задач.

Гнучкість інтерфейсу визначає можливість його адаптування до розв'язання конкретної задачі. Якщо розв'язувана задача дуже складна, то інтерфейс повинен полегшувати формулювання запитів і видавати

результати у формі, яка легко і швидко сприймається користувачем. Тобто інтерфейс повинен буди максимально простим навіть у випадку, коли розв'язується дуже складна задача.

Простота означає що:

- інтерфейс не повинен бути перевантажений деталями щодо подання розв'язку поставленої задачі – користувач може не охопити всіх подробиць (і в цьому, зазвичай, немає потреби) – тобто нічого зайвого, крім того, що необхідно для розуміння результату;
- інтерфейс не повинен містити зайвих декоративних деталей, які відволікають від головної задачі;
- інтерфейс повинен бути консистентним, тобто, ґрунтуватись на використанні відомих, загальноприйнятих методів і засобів подання інформації;
- в ідеалі процес взаємодії користувача з системою не повинен представляти ніяких труднощів.

5.2 Характеристики адаптивного інтерфейсу користувача

5.2.1 Адаптивні інтерфейси

Одним із перспективних напрямів створення інтерфейсів користувача вважають адаптивні інтерфейси. Адаптивним інтерфейсом користувача (AI) називають сукупність програмних та технічних засобів, які дозволяють користувачу ефективно використовувати всі можливості, які надає система, та задаються конкретні налагодження для кожного користувача.

Одним з головних критеріїв ефективності функціонування програмних продуктів є максимальне використання для роботи з ними людських ресурсів. Це пов'язано перш за все з тим, що вартість людських

ресурсів для роботи з програмним забезпеченням вже досягла вартості системних компонент, а в багатьох випадках суттєво їх перевищує.

Для того, щоб адаптивний інтерфейс міг бути коректно вбудованим в процес діалогу для користувача кожного рівня, він повинен враховувати апріорну інформацію про психофізичні, професійні, особисті характеристики користувача. Ця інформація здобувається шляхом попереднього початкового тестування користувача. Але цієї інформації для коректної роботи адаптивного інтерфейсу явно недостатньо. Додаткова інформація про користувача, який взаємодіє в даний момент з системою, повинна бути отримана з аналізу дій користувача безпосередньо в процесі роботи. Цей спосіб отримання знань про користувача і є основою динамічної адаптації.

Адаптивний інтерфейс повинен забезпечувати користувачу полегшений режим взаємодії. Адаптивна система повинна комбінувати в собі особливості адаптивних і адаптованих компонентів. Це буде давати користувачам засоби, які надають можливість проводити власні зміни в налагодженнях інтерфейсу, виходячи з їх потреб.

Системи, які дозволяють користувачу змінювати певні системні параметри і відповідно змінювати їх поведінку, називають адаптованими. Системи, які адаптуються до користувача автоматично, ґрунтуючись на припущеннях системи, називаються адаптивними.

Незважаючи на те, що на сьогодні вже досягнуто певний прогрес у побудові інтерфейсів користувача, питання взаємної адаптації функціонування програмних систем (ПС) і користувача ПС є досі відкритими. Більш того, актуальність їх зростає у зв'язку з розвитком мережі Internet, появою повністю графічних інтерфейсів, розвитком засобів мультимедії, широким застосуванням гіпертекстових документів, розширенням сфер використання та масовістю застосування ПЕОМ.

5.2.2 Природно-мовний інтерфейс

Розробка природно-мовного (ПМ) інтерфейсу в складі СППР дозволяє забезпечити виконання запиту ОПР, або експерта-аналітика до бази експертних знань. Можливість адаптації системи до рівня професіоналізму користувача здійснюється за рахунок здатності сприймати і виконувати запити на внутрішній формальній мові, що забезпечує більш швидкий доступ до інформації.

Природно-мовний інтерфейс є посередником між людиною і базою даних. Він переводить запити, що надходять природною мовою, у формальне подання, звертається з ним до бази даних, і представляє результат, використовуючи алгоритми і технології реалізовані в СППР, у вигляді, придатному для перегляду й аналізу (рис. 5.1).

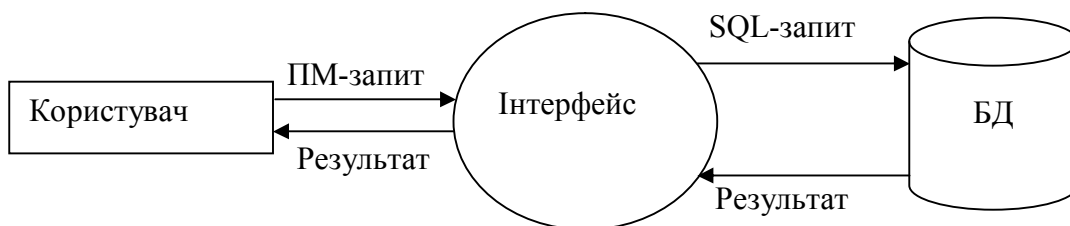


Рис. 5.1 Загальна схема потоків даних ПМ-інтерфейсу

Користувачами ПМ-інтерфейсів можуть бути як співробітники підприємства – менеджери різних рівнів, так і клієнти. Усім, кому потрібна можливість швидкого і прямого доступу до актуальної інформації. Наведемо кілька областей основних застосувань ПМ-інтерфейсів у діяльності організацій:

- каталог товарів і електронний магазин;
- календарне планування;
- керування кадрами і бухгалтерія;

- склад і планування закупівель;
- аналіз продаж;
- банківська діяльність;
- біржова діяльність.

Використання ПМ-інтерфейсу найвигідніше у випадку масового користувача (наприклад, відвідувача сайту), а також як інтегровану з інформаційною системою організації можливість прийняття рішення, при якому доступ до даних є штатною можливістю інформаційної системи організації і, таким чином, є універсальним і, можливо, основним способом пошуку й одержання інформації.

У ситуації, коли успіх організації визначає його здатність адекватно реагувати на зміну зовнішніх і внутрішніх умов, прямий і швидкий доступ до достовірної актуальної інформації, використання ПМ-інтерфейсу надає можливість значно скоротити витрати організації на різних рівнях:

- можливість формулювати потреби в інформації найбільш простим і разом з тим самим доступним чином – як для співробітників усередині компанії, так і для доступу клієнтів до публічної інформації (наприклад, до каталогу товарів);
- розвантажує фахівців від рутини створення форм і звітів по кожному розрізі інформаційного простору в базі даних. Крім того, оскільки формальне подання ПМ-запиту доступно, ПМ-інтерфейс може бути корисний для розробки традиційних додатків на основі баз даних, оскільки рятує від ручного етапу побудови складних SQL-запитів;
- при розробці ПМ-інтерфейсів використовуються найсучасніші методи інженерії знань, що дозволяє використовувати технологію розуміння на більш високому рівні абстракції даних, ніж реляційна модель зі збереженням записів у зв'язаних таблицях; це серйозний

крок на шляху до керування не просто інформацією, але й корпоративними знаннями.

Найбільшої ефективності можна досягти у випадку використання комбінації традиційних засобів доступу до даних і доступу природною мовою. Якщо на етапі проектування системи в неї буде вбудовано ПМ-інтерфейс, це одночасно підвищить надійність і корисність ПМ-інтерфейсу, а також різко збільшить інтелектуальність і дружність інформаційної системи в цілому.

Розглянемо конкретний випадок використання ПМ-інтерфейсу. Нехай фірма використовує БД для збереження внутрішньої бізнес-інформації. На основі баз даних працюють різноманітні додатки – складська програма, бухгалтерія, кадри, планування, прогнозування стану ринку, закупівлі, які часто об'єднані в єдину комплексну систему. Усі ці додатки, зазвичай, являють інформацію у виді таблиць і звітів у визначених розрізах з можливістю фільтрації.

Але часто буває так: для прийняття рішення "тут і зараз" необхідно мати такий розріз інформації, таку комбінацію умов, які неможливо задати стандартними засобами, особливо якщо ця інформація знаходиться на стику різних підсистем, наприклад така:

- витрати, зумовлені дефіцитом товару А на складі з початку місяця;
- коли очікується партія товарів фірми Х, і що в ній буде;
- кількість проданих холодильників Bosh за кольорами;
- прибуток від продажі відеомагнітофонів у залежності від їхньої ціни;
- наявність на складі 10 самих ходових товарів за минулий місяць.

Зазвичай, уся необхідна інформація для одержання таких цифр є у корпоративній базі даних. Але ні конструкцію бази, ні мови запитів до неї ви не знаєте – цим володіють розробники БД та інформаційної системи, побудованої на її основі. Можна, звичайно, попросити їх обчислити ці

цифри вручну (з використанням мови SQL, якщо база реляційна), або додати в систему нові функціональні можливості.

Це робиться просто, якщо розробники системи і БД працюють на даному підприємстві. Ситуація складніша, якщо розроблювачі – це стороння організація. Але у будь-якому випадку наступного разу вам буде потрібна яка-небудь інша інформація і все почнеться спочатку.

В наявності проблема – і є потреба в оперативному одержанні інформації в різноманітних розрізах для аналізу і прийняття рішень, є сама вихідна інформація, але доступ до неї вимагає знання особливостей побудови БДі досвіду розроблювача. Це рішення полягає у побудові природно-мовного інтерфейсу до БДі (у більш загальному випадку) – до інформаційної системи.

Це надає можливість:

- мати прямий доступ до будь-яких аналітичних розрізів інформації, що зберігається в базі;
- швидко отримувати необхідну інформацію, що потрібна у поточний момент;
- зосереджуватися на тому, що треба одержати з бази, а не на тому, як це зробити;
- вчасно контролювати правильність занесення інформації в базу;
- доповнити інформаційну систему інтелектуальною технологією, що дозволить в остаточному підсумку скоротити витрати на її експлуатацію і підвищити ефективність роботи з нею в цілому.

У загальному вигляді вимоги до ПМ-інтерфейсів повинні визначати зміст і форми інформації, що надається, а також регламент взаємодії користувача і системи. Ці вимоги повинні впливати з функцій користувачів, що знаходяться на різних рівнях ієрархії, а також стандартних вимог законодавчих і виконавчих органів до форм обліку і

звітності, регламентам діяльності об'єкта, що визначає в сукупності зовнішній і внутрішній документообіг.

Базовою формою опису технологічних і бізнесів-процесів повинна бути сама схема процесу, представлена на різних рівнях ієрархії з різним ступенем деталізації, на якій повинні бути відображені взаємозалежні технологічні і функціональні компоненти процесу разом з показниками функціонування, що дозволяють судити про факти і якість виконання функцій.

5.2.3 Можливості подання даних інтерфейсу

Сервісними формами подання інформації можуть бути тексти, таблиці, графіки, гістограми, а також абстрактні образи, що у сполученні з базовими формами являють повний спектр типів інформування користувача про стан об'єкта діяльності.

Таким чином, програмне забезпечення інтерфейсного модуля повинне забезпечити такі можливості:

- введення запитів до бази знань на внутрішній формальній мові системи;
- редагування запитів користувача;
- виконання запитів до бази експертних знань і вивід результатів на екран комп'ютера в зручному для користувача вигляді;
- перегляд проміжних результатів роботи;
- поповнення і коректування бази експертних знань у режимі діалогу;
- верифікацію відповідей на запити.

Основними етапами діяльності користувача в системі, як особи, що приймає рішення, є такі:

1. сприйняття інформації: виявлення об'єкту сприйняття, виділення в об'єкті окремих ознак, ознайомлення з виділеними ознаками і розпізнавання об'єкта сприйняття;
2. оцінювання інформації, її аналіз і узагальнення на основі раніше сформованих критеріїв оцінки; оцінка виробляється на основі співставлення і аналізу сформованої у ОПП інформаційної моделі з внутрішньою образно-концептуальною моделлю об'єкта управління (предметної області);
3. прийняття управлінських рішень.

5.2.4 Загальні принципи побудови адаптивних інтерфейсів

На основі загальних принципів побудови інформаційних систем і врахування особливостей адаптивного інтерфейсу можна сформулювати загальні принципи побудови адаптивних інтерфейсів:

1. Принцип відповідності призначення і структури інтерфейсу поставленим цілям і задачам. Неможливо побудувати універсальний багатофункціональний інтерфейс, який міг би бути використаний і в складі АСУ реального часу, і в складі системи управління фізичним експериментом. Типізація і універсальність можлива тільки в рамках визначених класів систем.

2. Принцип мінімізації витрат ресурсів користувача:

- користувач повинен виконувати тільки ту роботу, яка необхідна і не може бути виконана системою, не повинна бути повторенням уже виконаної роботи;
- користувач повинен запам'ятовувати якомога меншу кількість інформації, оскільки це збільшує його можливості приймати оперативні рішення в екстремальних випадках.

3. Принцип максимального взаєморозуміння та непротириччя. Робота з системою повинна бути простою, не викликати у користувача додаткових ускладнень в пошуках необхідних елементів інтерфейсу. Інформація, що отримана за допомогою інтерфейсу, не повинна вимагати перекодування або додаткової інтерпретації користувачем. Якщо в процесі роботи були використані деякі прийоми, то і в інших випадках прийоми роботи повинні бути ідентичні.

4. Принцип незбитковості. Користувач повинен введенняти тільки мінімальну інформацію та недопустимий ввід уже введеної раніше і отриманої системою інформації.

5. Принцип безпосереднього доступу до системи підказок. Система повинна забезпечувати користувача необхідними інструкціями. Система підказок повинна задовольняти трьом критеріям:

- стислість, якість і повнота інформації;
- вичерпний характер повідомлень про помилки;
- наявність повідомлень про стан системи.

6. Принцип гнучкості. Адаптивний інтерфейс повинен забезпечити спілкування з системою користувачам різного рівня підготовки. Наприклад, для недосвідчених користувачів інтерфейс може бути організовано як ієрархічна структура меню, а для досвідчених – з команд, комбінацій натиснень клавіш та параметрів командного рядка.

7. Принцип максимальної концентрації користувача на задачі, що розв'язується і локалізації повідомлень про помилки. Адаптивний інтерфейс повинен допомагати концентрувати увагу користувача на конкретній задачі, що розв'язується.

8. Принцип врахування професійних навичок конкретного користувача. При роботі з системою на основі збору деяких даних, які отримуються з моделі користувача, проектується «людський фактор», який

тісно вплітається в особливості функціонування всієї системи. Формування концептуальної системи означає усвідомлення і оволодіння алгоритмами функціонування інтерфейсу користувача.

9. Принцип легкості користування і простоти навчання. Адаптивний інтерфейс не повинен приводити користувача до роздратування, примушувати до необдуманих дій. Адаптивний інтерфейс враховує, що користувач отримує досвід роботи з системою, цілі користувача можуть змінюватись у процесі роботи з системою, Адаптивний інтерфейс характеризується простотою виправлення помилок. Керуванню роботи з системою можна легко навчитись, система забезпечує навчання в процесі роботи.

10. Принцип надійності. Система повинна бути надійною з точки зору роботи користувача. Вона готова до роботи завжди, коли в цьому виникає необхідність, відмови трапляються рідко, час відповідей системи не повинен перевищувати встановлених границь. В системі реалізуються можливості захисту інформації та забезпечення необхідного ступеня секретності та різних рівнів доступу до наявної інформації.

5.3 Проектування інтерфейсу з урахуванням людського фактора

5.3.1 Роль людського фактора

Предметом вивчення «людського фактору» є дослідження факторів, явищ, подій, які впливають на продуктивність праці, ефективність та якість життя всіх тих, хто працює, а також проектування машин та обладнання, що використовується в процесі виконання роботи [46].

Власне наука про людський фактор займається розробкою інженерних принципів проектування (дизайну) в усіх технічних галузях, використанням антропометричних таблиць вимірів людського тіла у

процесі дизайну, психологічними принципами навчання, аналізом процесів запам'ятовування візуальної та аудіоінформації, аналізом впливу на людину таких соціальних факторів, як шум навколишнього середовища, скупчення великої кількості людей, засобів зв'язку, а також проектуванням житлових приміщень, офісів та місць відпочинку (парків, технопарків, майданчиків тощо). Фактично будь-яка продукція, яка з'являється на ринку сьогодні і буде з'являтися у майбутньому, стає виграною і конкурентоспроможною завдяки застосуванню при її створенні принципів людського фактору.

Фахівці з людського фактора використовують всю наявну інформацію із психології, математики та технічних наук з метою удосконалення дизайну машин, механізмів, обчислювальних систем і підвищення, таким чином, ефективності роботи людини-оператора, а також поліпшення умов її життя та відпочинку. Набагато ефективніше та дешевше розробити і впровадити елементи «людського фактора» в процес проектування систем, ніж переробляти готову продукцію і відповідності до рекламаций та скарг споживачів.

Можна сказати, що людський фактор – це метод проектування, дослідження та пояснювання інтерфейсу в широкому смислі (тобто, проміжного середовища) між людиною та машиною або знаряддями, які створюються людиною. З точки зору психології людини ми розглядаємо характеристики оператора і машини або знаряддя праці з метою визначення кращих методів чи способів конструювання цих машин та знаряддя, які підвищують ефективність праці та полегшують користування ними. В результаті такого аналізу проекту може з'явитись необхідність змінити конструкцію, щоб вона була доступною та зручною (комфортною) для оператора.

Вимоги та рекомендації науки про людський фактор є надзвичайно важливими з точки зору організації взаємодії користувач-комп'ютер. Всіх користувачів комп'ютерних систем можна розділити на дві основні групи: фахівці (постійно користуються комп'ютером) та тимчасові користувачі (користуються комп'ютером час від часу – керівники різних рівнів, бухгалтери, користувачі-початківці). Фахівці проводять, зазвичай, за комп'ютером значний час, користуючись програмними системами для розв'язання своїх задач. Такі користувачі не мають проблем з користуванням клавіатурою та монітором незалежно від типу програмного продукту.

Тимчасові користувачі можуть час від часу використовувати конкретну програмну систему для знаходження розв'язку спеціалізованих задач, наприклад, редактор текстів для набору текстових матеріалів або Excel для підготовки бухгалтерських звітів. Такий користувач може ніколи не почуватись повністю комфортно з терміналом та програмою. Зважаючи на те, що на сьогодні кількість тимчасових користувачів надзвичайно велике, інтерфейс програмної системи повинен бути максимально дружнім, щоб забезпечити комфорт та якість результатів взаємодії.

5.3.2 Загальні принципи врахування людського фактора

При проектуванні інтерфейсу програмна система – користувач необхідно враховувати такі фактори, які є важливими з точки зору інженерії людини. Основними з них є такі:

1. Користувач повинен завжди знати, що робити далі.

Тобто, система повинна давати йому інструкції стосовно того, як продовжити роботу, створити резервний файл результатів, вийти із

системи тощо. Існують декілька ситуацій, які вимагають зворотного зв'язку від системи (пояснення для користувача):

- користувач повинен знати, що система чекає від нього. Це може бути зроблено у вигляді простого повідомлення, такого як ГОТОВО (READY), ВВЕДІТЬ КОМАНДУ (ENTER COMMAND), ВВЕДІТЬ ВИБІР (ENTER CHOICE) або ВВЕДІТЬ ДАНІ (ENTER DATA);
- користувач повинен знати, що дані введено коректно. Наприклад, це можна зробити переміщенням курсора у поле, де написано ДАНІ КОРЕКТНІ або ВВЕДЕННЯ ЗАВЕРШЕНО (INPUT OK);
- користувач повинен знати, що дані не були введені коректно. Система повинна дати просте коротке повідомлення щодо необхідного формату даних. Більш докладні рекомендації щодо можливих форматів даних можна надати за допомогою функції ДОПОМОГА (HELP);
- якщо має місце затримка з обчисленнями, то користувач повинен про це знати. Для виконання деяких видів обчислень (наприклад, при виконанні оптимізаційних розрахунків) необхідно витратити кілька десятків секунд або навіть хвилин. У такому випадку система повинна давати короткі повідомлення, наприклад: ПОШУК ОПТИМУМУ – БУДЬ-ЛАСКА ЗАЧЕКАЙТЕ або СОРТУВАННЯ – ЦЯ ОПЕРАЦІЯ МОЖЕ ТРИВАТИ ДЕКІЛЬКА ХВИЛИН, БУДЬ-ЛАСКА ЗАЧЕКАЙТЕ. Користувач буде впевнений, що система продовжує працювати;
- користувач повинен знати, що система завершила (не завершила) виконання завдання. Це важливо завжди, але особливо важливо у тих випадках, коли обчислення можуть тривати відносно довго. Повідомлення може мати такий характер: ДРУК ЗАВЕРШЕНО або

ПРИНТЕР НЕ ГОТОВИЙ – ПЕРЕВІРТЕ НАЯВНІСТЬ КАРТРИДЖУ, або ОПТИМАЛЬНА ТРАЄКТОРІЯ ЗНАЙДЕНА.

2. Форматування екрану монітора необхідно робити таким чином, щоб різні типи інформації, команди, повідомлення завжди з'являлись в одній і тій же області.

При такому підході до форматування екрану користувач завжди буде хоча б приблизно знати, де шукати необхідну інформацію. Наприклад, екран можна розділити на такі зони (області):

- вікно з назвою (титолом) сторінки на екрані (title window). Титул (назва) необхідні для ідентифікації вікна конкретного типу з точки зору користувача;
- вікно з прапорцями (flag window). Це вікно містить вказівник (вказівники) на інші конкретні вікна екрану з метою показати, що виникла помилка чи проблема з виконанням завдання. Наприклад, якщо користувач зробив помилку при введенні команди чи даних, то у вікні з прапорцями повинно з'явитись повідомлення «...», яке показує в якому рядку чи полі має місце помилка. Для того щоб взнати конкретну причину чи суть помилки, користувач повинен подивитись у вікно повідомлень (розглядається нижче). Замість вікна з прапорцями можна скористатись такими атрибутами як мигання або інверсним зображенням;
- вікно повідомлень (message window). В це вікно система посилає повідомлення для користувача. Наприклад, повідомлення про помилки або пропозиції щодо продовження обчислень, ведення діалогу. Це вікно може бути порожнім на протязі більшої частини часу роботи з системою;
- вікно виходу (escape window). Воно призначене для того щоб користувач міг завершити поточний сеанс роботи з системою або

підсистемою. Наприклад, вікно виходу може містити інформацію стосовно того як повернутись до головного меню системи або до попереднього меню опцій (вибору);

- операційне вікно (body window). Операційне вікно – найбільше вікно діалогу. Ця область використовується для введення даних, виведення результатів роботи, виведення пояснюючих повідомлень за допомогою ведення діалогу з системою, висвітлення опцій меню тощо.

Зони не обов'язково повинні мати візуальні границі, але, якщо вони є, то покращують сприйняття інформації. Приклад можливого розділення екрану на зони наведено на рис. 5.2. В іншому варіанті розподілу екрану повідомлення, допомога та групування клавіш можуть розміщуватись в нижній або верхній частині екрану.

ЗАГОЛОВОК	
КОМАНДИ ТА ПОВІДОМЛЕННЯ	
ОСНОВНЕ ПОЛЕ ПОДАННЯ РЕЗУЛЬТАТІВ	П Р А П О Р Ц І
ВИХІД	

Рис. 5.2 Приклад розподілу екрану терміналу на зони

3. При проектуванні функцій пейджингу чи скролінгу у межах операційного вікна діалог повинен обмежуватись однією ідеєю на фрейм.

Наприклад, зона повинна висвітлювати одне меню, один результат виконання завдання або відповідь на один запит.

Вибір між пейджингом та скролінгом в зоні операційного вікна залежить від типу інформації, яку необхідно висвітлити. Якщо інформація є неперервною за своєю природою (наприклад, текстова інформація щодо отриманих результатів), то використовують скролінг. Перегляд рядків тексту виконують за допомогою курсора. Пейджинг використовують у випадках, коли інформація висвітлюється у вигляді окремих записів за один раз.

4. Повідомлення, команди або інформація щодо результатів повинні утримуватись на екрані достатньо довго, для того щоб користувач міг прочитати і сприйняти їх.

Наприклад, результати обчислень не повинні швидко зникати з екрану, тобто, до того, поки користувач їх прочитає. Одним із підходів до висвітлення інформації в зоні є висвітлення такого об'єму інформації, який може вмістити зона, а потім «заморозити» висвітлення до наступного запиту користувача (наприклад, до натискування клавіші із стрілкою).

Для економії місця на екрані деякі зони можна тимчасово анулювати і замінювати їх іншими, які необхідні в конкретній ситуації.

5. Економне використання ресурсів монітора

Такі атрибути монітора як мигання, збільшення яскравості та інверсне відео необхідно використовувати економно, оскільки їх надмірне використання може призводити до зворотного ефекту, тобто ускладнення або погіршення характеристик діалогу (надмірне витрачання часу, відволікання уваги від головного). Коректним є використання додаткових ресурсів монітора для притягнення уваги до важливих моментів,

наприклад, до поля для введення даних, екстрених повідомлень або до команд.

6. Спростуйте складні функції і скорочуйте об'єм введення даних з клавіатури за рахунок використання функціональних клавіш.

Функціональним клавішам можуть бути присвоєні, наприклад, такі функції:

- ЗАПУСК (START) програми або функції;
- ДОПОМОГА (HELP): відображення на екрані пояснюючого тексту;
- переміщення курсора на екрані;
- ВИХІД (EXIT) або ЗАВЕРШЕННЯ (TERMINATE) поточного сеансу діалогу;
- переривання поточної операції (ESCAPE), наприклад, введення або виведення даних; така операція може означати «повторний запуск», якщо була зроблена серйозна помилка оператором;
- використання комбінацій клавіш; комп'ютер може мати спеціальне програмне забезпечення для розширення функцій клавіатури; воно дозволяє присвоювати одній клавіші (або парі клавіш) функції виконання визначеної послідовності команд.

7. Створюйте специфікації значень або повідомлень «за замовчуванням», які спрощують введення даних користувач системи в процесі діалогу.

Наприклад, якщо необхідно ввести дату деякого замовлення, то інформаційний рядок може мати наступну форму:

ДАТА ЗАМОВЛЕННЯ? <Сьогодні?>

Для введення поточної дати користувач натискає клавішу Ввести <Enter>.

8. Передбачайте помилки, які можуть мати місце з боку користувача системи.

Зазвичай, користувачі роблять помилки навіть при виконанні самих простих і очевидних операцій. Якщо дії користувача можуть спричинити потенційно небезпечну ситуацію, то він повинен знати про це. Наприклад, при спробі створити таку ситуацію, він отримує повідомлення:

ВИ ВПЕВНЕНІ, ЩО ТАКА ДІЯ НЕОБХІДНА?

Навіть такі прості попередження відіграють значну роль у попередженні потенційно небезпечних дій користувача.

5.4 Тональність діалогу та термінологія

Ще однією важливою характеристикою діалогу реального часу з точки зору людського фактору є загальна тональність діалогу та термінологія, яка використовується при цьому.

Діалог повинен бути дружнім по відношенню до користувача, а тональність діалогу повинна ґрунтуватись на таких загальних рекомендаціях:

- використовувати прості, граматично правильні речення. При цьому краще використовувати розмовні фрази, але не говірки (сленг) або загальнонеприйнятні вирази.
- не бажаним є вживання «гострих фраз». Якщо хтось буде користуватись вашою системою 50 разів на день, то гумор, наявний в діалозі, буде сприйнятись так само як один і той же жарт 50 разів підряд.

– не вживати фраз, які звучать як повчання для користувача. Наприклад, не потрібно хвалити користувача або говорити, що він ще не все знає з того, що необхідно (це фрази такого типу: О, ВИ РОБИТЕ ВСЕ ПРАВИЛЬНО або ВИ ПОВИННІ ЗНАТИ ЦЕ КРАЩЕ). Стосовно термінології фахівці з проектування комп'ютерних систем рекомендують притримуватись таких правил:

- не використовувати комп'ютерний жаргон;
- уникати аббревіатур, наприклад, не використовувати ВД замість ВВЕДІТЬ ДАНІ.
- уникати символів, які можуть бути невідомими для користувача; наприклад, математичних позначень там, де в цьому немає крайньої необхідності;
- використовувати просту і зрозумілу термінологію. Наприклад, краще використати фразу ПОМИЛКА ВВЕДЕННЯ ДАНИХ З ДИСКА ніж ПОМИЛКА ЗОВНІШНЬОГО ПРИСТРОЮ;
- використовувати одні й ті ж терміни для виконання однакових операцій. Наприклад, не потрібно використовувати терміни РЕДАГУВАННЯ ДАНИХ і МОДИФІКАЦІЯ ДАНИХ в одному й тому ж значенні;
- команди і пояснення користувачу повинні бути продумані і коректно сформульовані з правильним вживанням дієслів. Краще сказати ВИБЕРІТЬ МЕТОД РОЗРАХУНКУ ПАРАМЕТРІВ замість ВКАЖІТЬ НА МЕТОД. Що стосується дій з клавіатурою, то необхідно говорити НАТИСНІТЬ КЛАВІШУ <F6>, а не СКОРИСТАЙТЕСЬ КЛАВІШЕЮ F6>.

Запитання та завдання для самоконтролю

1. Якими є вимоги до інтерфейсів інформаційних систем?
2. Описати характеристики адаптивного інтерфейсу користувача.
3. Якими є особливості природно-мовного інтерфейсу?
4. Загальна схема потоків даних природно-мовного –інтерфейсу.
5. Основні можливості для організації природно-мовного інтерфейсу.
6. В чому полягають особливості використання баз даних у природно-мовних інтерфейсах?
7. Якими є можливості подання даних інтерфейсу?
8. Загальні принципи побудови адаптивних інтерфейсів.
9. Що є предметом вивчення людського фактору в інтерфейсах?
10. Роль людського фактора в інтерфейсах.
11. Якими є загальні принципи врахування людського фактору в інтерфейсі СППР?
12. У чому полягає коректність і правильний вибір тональності діалогу та термінології інтерфейсу СППР?
13. Сформулюйте вимоги до інтерфейсів інформаційних систем.
14. Поясніть, що таке адаптованість інтерфейсу?
15. До яких характеристик користувача може адаптуватись інтерфейс?
16. Наведіть основні характеристики інтерфейсів користувача СППР.
17. Вкажіть, які можливості повинен забезпечувати інтерфейсний модуль інформаційної системи.
18. Сформулюйте принципи формування інтерфейсу користувача.

19. Поясніть такі принципи побудови інтерфейсів, як принцип мінімізації затрат ресурсів користувача та принцип незбитковості.
20. Які існують напрями розвитку науки про людський фактор?
21. Розкрийте принципи проектування інтерфейсу користувача на засадах людського фактору.
22. Наведіть ситуації, в яких система повинна надавати пояснення для користувача.
23. Зазначте особливості форматування екрану монітора при розробці інтерфейсів користувача.
24. Опишіть аспекти вибору тональності та термінології при реалізації інтерфейсів користувача.
25. Які недоліки і переваги має інтерфейс на основі натуральних мовних конструкцій?
26. Наведіть приклад застосування інтерфейсу на основі безпосереднього маніпулювання графічними об'єктами.
27. Де застосовуються інтерфейси на основі заповнення стандартизованих форм?

РОЗДІЛ 6 МЕТОДОЛОГІЯ РОЗРОБКИ І РЕАЛІЗАЦІЇ СИСТЕМ ПІДТРИМКИ ПРИЙНЯТТЯ РІШЕНЬ

6.1 Умови успішної реалізації СППР

Умовами успішної реалізації СППР і взагалі будь-якої комп'ютерної інформаційної системи (КІС), є: якомога повне врахування потреб та вимог замовника, правильний вибір інструментарію для реалізації системи, наявність практичного досвіду і знань виконавця, належна фінансова підтримка, а також доступ до джерел необхідної інформації. Розглянемо докладніше ці умови.

Основними вимогами для успішної реалізації СППР є: визначення потреб і вимог замовника; правильний вибір інструментальної платформи для реалізації проекту; достатні знання та досвід розробника у проектуванні і реалізації комп'ютерних інформаційних технологій; глибоке розуміння розробником проблеми прикладної області; доступ до джерел необхідної інформації для розробника (виконавця); належна фінансова підтримка проекту; формування замовником належним чином проектної групи.

6.1.1 Визначення потреб і вимог замовника

Зазвичай визначення потреб і вимог замовника є непростою задачею, оскільки замовник рідко має необхідне чітке уявлення про майбутню систему. Вимоги до інтерфейсу та функцій системи замовник формулює досить наближено, а тому задачею виконавця є максимально точно визначення потреб користувача (замовника).

Вміння коректно працювати із замовником – одна з головних умов успішної реалізації проекту.

Вимоги замовника включають:

- вимоги до інтерфейсу. Необхідно визначити, який конкретно тип інтерфейсу необхідно спроектувати та якими повинні бути його можливості стосовно взаємодії з користувачем та визначити детальні характеристики цього інтерфейсу;
- докладний перелік функцій системи. Виконавець надає замовнику максимальну допомогу у складанні переліку функцій системи, оскільки це впливає на її вартість і якість в цілому. Перелік складається у письмовому вигляді і підписується замовником та виконавцем;
- конкретизація методів попереднього та основного оброблення даних у рамках системи. Зазвичай, методи оброблення даних пропонує виконавець, але замовник повинен дати згоду на їх використання. Необхідно визначити ступінь складності реалізації кожного методу та довести цю інформацію до відома замовника;
- замовник повинен надати свої вимоги до способів подання результатів, а виконавець може доповнити їх своїми пропозиціями. В результаті формулюється перелік вимог до системи подання результатів;
- необхідно встановити вимоги до точності обчислень та максимального часу виконання обчислювальних алгоритмів, що використовує система генерування результатів.

6.1.2 Правильний вибір інструментальної платформи для реалізації проекту

Інструментальна система повинна мати такі характеристики:

- доступна за ціною;
- не дуже складною у використанні;
- забезпечувати сучасний рівень реалізації інформаційної системи.

6.1.3 Вимоги до знань та досвіду проектування

Розробник повинен мати достатні знання та досвід проектування і реалізації комп'ютерних інформаційних технологій.

Як недооцінка, так і переоцінка своїх можливостей можуть призвести до суттєвих проблем у процесі реалізації проекту. Зокрема, недооцінювання своїх можливостей призводить до заниження вартості і збільшення термінів реалізації системи. Переоцінювання можливостей може призвести до зриву строків реалізації системи і зниження її якості.

6.1.4 Вимоги до розуміння проблеми прикладної області

Розробник повинен глибоко розуміти проблеми прикладної області, до якої відноситься поставлена задача проектування і реалізації інформаційної системи. Якщо проектна група не включає такого фахівця, то його обов'язково необхідно знайти і найняти на час виконання проекту.

Виконавець повинен пам'ятати, що:

- для вивчення предметної області необхідно витратити щонайменше 1,0 – 1,5 роки (в нормальному режимі роботи);

- якщо немає фахівця з предметної області у своїй фірмі, то його краще знайти в іншій щоб не витратити час на вивчення цієї області;
- можливе створення об'єднаної групи виконавців, яка включає фахівців замовника та виконавця, особливо якщо предметна область труднодоступна (або секретна) – військова галузь, спеціальні наукові дослідження в області ядерної енергетики і таке інше.

6.1.5 Вимоги доступу до інформації

Розробник (виконавець) повинен мати доступ до джерел необхідної інформації. Якщо це забезпечити неможливо, то інформацію надає замовник.

Всі випадки забезпечення виконавця спеціальними (не загальнодоступними) даними обов'язково оговорюються у технічному завданні. Відсутність необхідної інформації може суттєво вплинути на строки виконання проекту.

6.1.6 Вимога фінансової підтримки

Належна фінансова підтримка проекту передбачає попередню роботу з боку виконавця. Виконавець повинен:

- впевнитись у фінансовій спроможності замовника;
- вимагати аванс, який забезпечить виконання робіт на 50 %; зазвичай, такий аванс дає можливість створити перший варіант системи, який можна продемонструвати замовнику;
- чітко сформулювати свої дії у випадку відмови замовника вчасно оплачувати роботу.

6.1.7 Створення групи із розроблення проекту

Замовник повинен належним чином сформувати проектну групу (колектив) для розробки та реалізації проекту. Створення такої групи повинно гарантувати успішність реалізацію проекту СППР, запуск і підтримку впровадження системи.

Наведені сім умов успішної реалізації проекту комп'ютерної інформаційної системи, зокрема, СППР, стосуються як професійного розробника системи, так і експерта, який глибоко знає предметну область, але не має достатнього досвіду реалізації інформаційних технологій. Методологія реалізації проекту повинна зменшувати ризики проектування СППР.

6.2 Фактори ризиків при проектуванні та реалізації СППР

При проектуванні та реалізації СППР і взагалі комп'ютерних інформаційних систем, розробники таких систем стикаються з певними факторами ризику проекту. Серед них можна виділити такі основні фактори ризику:

1. Непорозуміння із замовником.

Це один із основних факторів ризиків. Він виникає у випадку недостатньої взаємодії виконавця із замовником і може стосуватись таких елементів проекту:

- недостатньо глибоко проаналізовані питання стосовно вчасного і повного фінансування робіт;
- непорозуміння стосовно вимог користувача, вони можуть бути сформульовані недостатньо чітко з відповідними наслідками -- користувач отримує не те, що йому хотілося б;

– непорозуміння стосовно функцій системи.

2. Неперервні зміни стосовно вимог до системи.

Замовник занадто часто змінює вимоги до проекту системи, що унеможлиблює його своєчасну і високоякісну реалізацію.

Метод боротьби – тісна співпраця із замовником на етапі створення технічного завдання і проекту. Дотримання всіх вимог документації.

3. Нечітко сформульовані цілі проекту.

Ризик незапланованих витрат часу на переформулювання цілей. Він зустрічається у випадках, коли виконавець ще не має достатнього досвіду розробки систем конкретного типу.

Метод боротьби – коректне формування групи виконавців, забезпечення їх високої кваліфікації.

4. Непорозуміння стосовно функціональних вимог серед членів групи виконавців проекту.

Метод боротьби – покращення менеджменту проекту, підвищення кваліфікації членів групи. Фірми, які працюють в галузі ІТ, повинні постійно працювати над підвищенням кваліфікації своїх працівників.

5. Неправильне використання людських ресурсів замовником.

Неякісний менеджмент проекту.

Метод боротьби – потрібно замінити керівництво або виконавцям знайти іншу роботу.

6. Нереальний календарний план.

Причина – неякісний менеджмент проекту.

Методи боротьби полягають у залученні висококваліфікованих виконавців з досвідом та застосування сучасних комп'ютерних засобів для оптимізації календарного плану.

7. Неможливість задовольнити вимоги замовника-користувача.

Причина ризику полягає у прорахунки з формулюванням вимог до системи на етапі написання технічного завдання, недостатня увага щодо складності функцій та можливостей їх реалізації в конкретних умовах.

8. Ризик, пов'язаний із впровадженням нових технологій.

Тобто неможливість освоїти нову технологію реалізації системи в межах проекту.

Методом боротьби з ризиком є підвищення кваліфікації виконавців. Перед прийняттям рішення стосовно впровадження нових технологій необхідно переконатись у тому, що виконавці готові до їх використання.

9. Незадовільний контакт із замовником в процесі реалізації проекту та недостатньо висока якість проміжних демонстрацій (презентацій) досягнутих результатів.

В результаті можуть виникнути непорозуміння із фінансуванням і виконанням вимог користувача.

Методом боротьби з ризиком є постійний контакт із замовником і робота над покращенням стосунків з ним, продумування демонстрацій.

10. Недастатньо глибокі знання і досвід роботи виконавця.

В такому випадку існує ризик повністю провалити проект.

Методи боротьби з ризиком: підбір кадрів для виконання проекту, попереднє підвищення кваліфікації. Для забезпечення успіху необхідно, щоб не менше 60% членів групи виконавців мали достатній досвід реалізації проектів.

11. Відсутність належної методики виконання проекту.

Причиною ризику є недостатній досвід виконавця.

Методом боротьби є використати досвіду провідних фірм-проектувальників інформаційних систем, розробити власну методику

створення системи на основі попереднього досвіду. Все корисне має документуватись.

12. Надмірна увага частини виконавців проекту до підвищення якості виконання окремих функцій (за принципом: краще – це ворог хорошого).

Це характерний недолік виконавців, які ще не мають достатнього досвіду реалізації проектів.

Метод боротьби полягає у підтримці (моніторингу роботи) початківців з боку висококваліфікованих працівників.

13. Неузгодженість програмних модулів між собою.

Ризик пов'язано з недостатньо високим рівнем безпосереднього менеджменту проекту. Тобто це ризик прорахунків менеджера проекту.

Ризик особливо стосується використання аутсорсингу (outsourcing), коли немає безпосереднього контакту із замовником, а основне керівництво приїздить нечасто.

6.3 Менеджмент ризиків проектів з розроблення СППР

Процес ідентифікації та менеджменту ризиків в компаніях, що працюють в галузі інформаційних технологій та виконують проекти з розробки комп'ютерних інформаційних систем, наприклад, таких як СППР, спрямований на виявлення наявних у проекті ризиків, їх оцінювання, планування та реалізацію заходів стосовно зниження впливу та ліквідацію факторів ризиків. Схема процесу ідентифікації та менеджменту ризиків ІТ-проектів представлена на рис. 6.1.

Для мінімізації та ліквідації ризиків при проектуванні та реалізації СППР вдаються до таких заходів:

1. Уникнення ризиків.

Виконується аналіз можливих факторів ризику та створення умов, несприятливих для їх реалізації (прояву). Це ідеальний варіант, але уникнути ризиків на 100% – нереально.

2. Перенесення ризику на третю сторону.

Вживаються такі заходи:

- страхування проекту;
- перенесення ризику на сторону замовника;
- створення власного хедж-фонду (фонд, який гарантує виживання компанії-виконавця в умовах настання та реалізації ризиків).

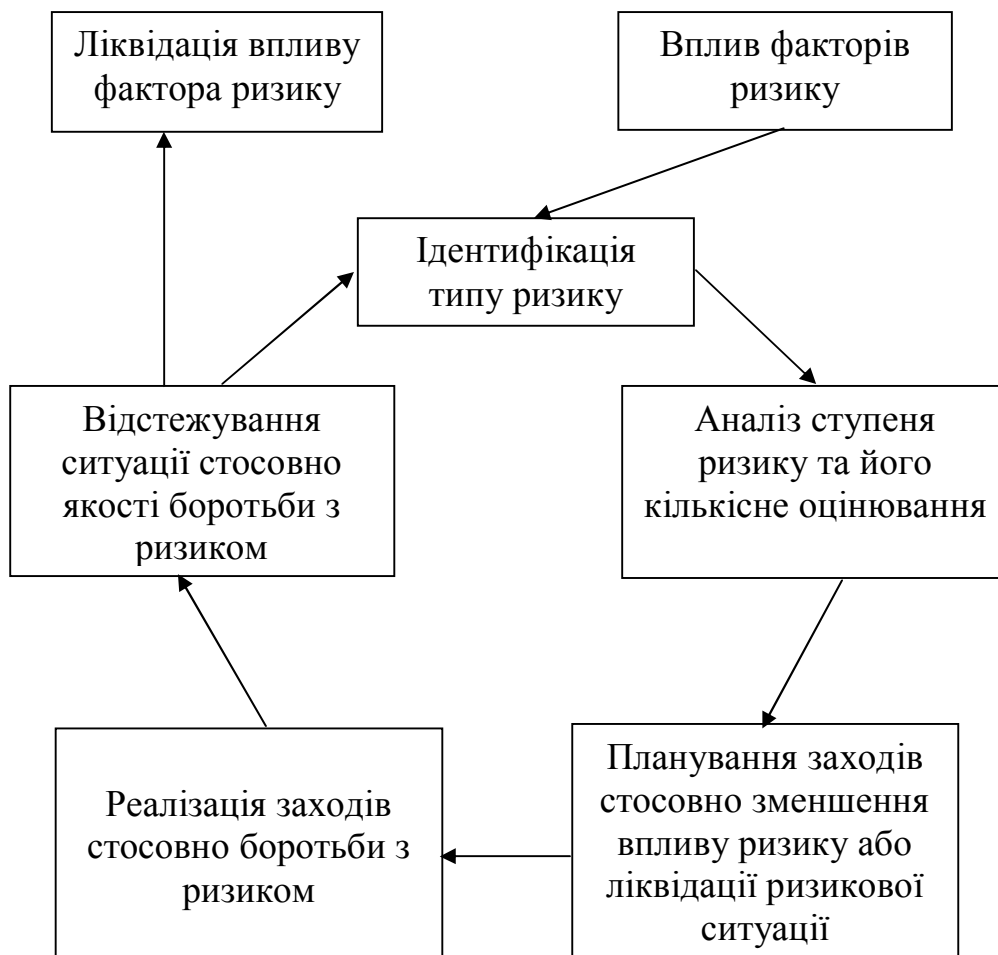


Рис. 6.1 Процес ідентифікації та менеджменту ризиків в компанії

3. Зменшення впливу ризику на функціонування компанії.

Планування і виконання дій, спрямованих на зменшення впливу факторів ризику, зменшення ймовірності його виникнення шляхом використання власного досвіду виконання проектів і досвіду інших компаній.

Вживаються заходи: підвищення кваліфікації працівників, прийняття реалістичних строків і бюджет проекту.

4. Прийняття ситуації з виникненням ризику як вона є і робота над зменшенням наслідків реалізації ризику.

Можливо, що витрати на проект перевищать прибуток, але в майбутньому можлива компенсація.

Необхідно прийняти рішення стосовно того, в якому випадку втрати компанії будуть меншими:

- у випадку завершення проекту з фінансовими втратами;
- у випадку відмови від його виконання.

Найчастіше проект завершують, щоб не втратити імідж компанії.

6.4 Послідовність розробки та реалізації СППР

6.4.1 Основні етапи розробки та реалізації

Процес розробки та реалізації СППР складається з таких послідовних етапів:

- формулювання постановки задачі;
- аналіз можливості реалізації СППР;
- формулювання вимог до СППР;
- створення проекту СППР;
- опис програмного забезпечення системи;
- послідовність програмування модулів системи;

- послідовність тестування системи;
- підготовка інсталяційної версії і експлуатаційної документації;
- впровадження системи на технічних засобах замовника;
- супроводження СППР.

6.4.2 Формулювання постановки задачі

Постановка задачі оформляється як частина технічного завдання і містить такі елементи:

а) повну назву системи і її аббревіатуру, якщо передбачається її використання. Наприклад, повною назвою може бути «Комп'ютерна інформаційна система підтримки прийняття рішень при виконанні операцій на біржі», а скорочена – Стоктрейдер (StockTrader). Інші приклади: Econometric Views (Eviews); Statistical Analysis of Systems (SAS).

б) опис проблеми, стосовно якої необхідно приймати рішення. При цьому необхідно підкреслити особливості цієї проблеми, це можуть бути:

- багатовимірність і стохастичність процесу;
- необхідність застосування оптимізаційних процедур для формування альтернатив;
- використання спеціальних чисельних методів оцінювання параметрів математичних моделей (наприклад, нелінійні процедури та метод Монте Карло);
- застосування методів аналізу нестационарних та нелінійних процесів,
- використання експертних оцінок, ймовірнісних методів формування висновку тощо;

в) попередній аналіз можливості розв'язання задачі за допомогою наявних (відомих) методів комп'ютерної підтримки, даних і знань. При цьому необхідно обов'язково врахувати свій попередній досвід проектування і реалізації систем, подібних до тієї, що проектується;

г) основний метод (методи) прийняття рішень та проектування альтернатив, який буде використано в СППР, наприклад: (1) на основі байєсівського підходу; (2) нечіткої логіки; (3) методу аналізу ієрархій; (4) послідовного перебору варіантів на основі деякого чисельного критерію;

д) типи моделей, критеріїв вибору моделей і рішень, які будуть використані в СППР.

6.4.3 Аналіз можливості реалізації СППР

Результати аналізу включають такі елементи:

а) аналіз можливості технічної реалізації системи на наявних засобах обчислювальної техніки;

б) наявність технологій розробки систем даного класу і досвіду створення та використання готового продукту. Можливо необхідно передбачити спеціальні засоби для вимірювання та введення даних та організації зв'язку з іншими системами;

в) аналіз економічних можливостей, тобто, аналіз достатньої фінансової підтримки на всіх етапах розробки і реалізації системи.

Основне запитання: чи достатньо коштів, які може запропонувати замовник для проектування і реалізації системи? Враховуючи при цьому, що виконавець і замовник мають протилежні інтереси, розрахункову вартість необхідно збільшити в 1,5-2,0 рази;

г) аналіз біхевіористичного ефекту, тобто в чому буде проявлятися позитивний ефект від впровадження СППР в практику. Ефект може проявлятися у такому:

- можливості використання відносно складних моделей процесів та сучасних методів оптимізації процесів управління на різних рівнях ієрархії;
- підвищення якості рішень та скорочення часу на їх прийняття;
- підвищення продуктивності праці персоналу;
- підвищення якості продукції і зменшення витрат на виробництво;
- зменшення навантаження на працівників компанії тощо;

д) аналіз можливостей реалізації системи включає також розробку календарного плану, який може корегуватись в подальшому і є частиною технічного завдання. Календарний план визначає послідовність виконання робіт у часі і дає можливість:

- виявити роботи, які можуть виконуватись паралельно;
- оптимізувати строки виконання проекту методами дослідження операцій;
- корегувати навантаження на виконавців;
- порівнювати плани робіт з планами раніше виконаних проектів.

Фірми з розроблення інформаційних систем ведуть каталоги виконаних успішних та неуспішних проектів з метою виявлення причин повного або часткового провалу проекту. Тобто виявляються можливості, які не були використані у процесі реалізації проекту.

6.4.4 . Формулювання вимог до СППР

Вимоги до системи діляться на дві групи: вимоги користувача і функціональні вимоги. Обидві множини вимог можуть перетинатись, але краще записати вимогу два рази ніж пропустити її зовсім.

Вимоги користувача – формулюються виконавцем і узгоджуються із замовником.

Розглянемо формулювання типових вимог користувача на прикладі СППР при прогнозуванні динаміки часових рядів. Вимоги користувача, зазвичай, стосуються таких елементів:

- тип комп'ютера і операційна система, для якої створюється СППР;
- можливість використання двох або більше форматів зображення інтерфейсу на екрані: розширеного (на весь екран) і зменшеного (приблизно на $\frac{1}{2}$ екрану);
- максимальний час реакції на запит (наприклад, не більше 2-5-и секунд); при перевищенні цього часу видавати відповідне повідомлення;
- генерування, при необхідності, звукових сигналів визначеної тональності, які свідчать про успішність або неможливість виконання запиту;
- введення даних з клавіатури, жорсткого диска, гнучкого диска та компакта, через Інтернет та локальну мережу;
- перегляд введених даних у цифровому та графічному вигляді або у вигляді діаграми з використанням визначеної гама кольорів;
- одночасний перегляд кількох рядів на одному графіку;

- можливість використання різних форматів даних: ASCII, XLS, db та інші;
- автоматизація аналізу альтернатив за допомогою множини правил;
- задавати точність отримуваних результатів (прогнозів);
- вибирати тип подання результатів;
- можливість доповнювати та редагувати дані;
- можливість доповнювати систему алгоритмами оцінювання параметрів моделей і критеріями оцінювання якості результату (прогнозу).

Функціональні вимоги до системи – формулюються виконавцем на основі вимог та побажань замовника.

Функціональні вимоги стосуються безпосередньо тих функцій, які буде виконувати система. Наприклад, функціональні вимоги до СППР при прогнозуванні часових рядів можна сформулювати так:

- методи попередньої оброблення даних (логарифмування, нормування, фільтрація, заповнення пропусків);
- методи оцінювання параметрів лінійних, псевдолінійних та нелінійних математичних моделей;
- методи автоматичного тестування часових рядів на нестационарність стосовно тренду та дисперсії за допомогою відповідних дерев рішень з видачею повідомлень користувачу;
- критерії оцінювання якості моделі;
- правила вибору кращої моделі з множини кандидатів;
- критерії оцінювання якості прогнозу;
- методика автоматичного оцінювання (за умовчанням) якості моделі та прогнозу;

- генерування рекомендацій стосовно вибору методу оцінювання параметрів моделі і методу прогнозування;
- можливості модифікації, розширення та тестування бази знань і даних;
- функція ретроспективного аналізу результатів.

Вимоги до системи також представляються в технічному завданні.

6.4.5 Створення проекту СППР

На основі сформульованих вимог до системи створюється проект, основними елементами якого є такі:

А) остаточний вибір інструментальної системи для реалізації СППР. Необхідно пам'ятати, що вартість різних інструментальних систем може відрізнятися одна від одної в декілька разів. Вибір ґрунтується на досвіді виконавця і замовника, наявних коштах і можливостях конкретного інструментарію. Деякі приклади вибору будуть розглянуті нижче;

Б) проект мовної системи (МС) містить такі елементи:

- всі види запитів до функцій системи в їх конкретному формулюванні та структурі;
- всі види запитів стосовно допомоги користувачу;
- можливі незначні відхилення від специфікацій, якщо дозволяють засоби реалізації системи; тобто, надання більше свободи користувачу при введенні запитів;

В) проект системи подання результатів (СПР) повинен містити:

- всі види реакції системи на запити про допомогу користувачу;

- всі види реакції системи на запити щодо розв’язку задачі та подання результатів із зазначенням типу подання – текст, графік, таблиця, числа та комбіноване подання;

- можливі незначні відхилення від специфікацій в залежності від конкретних інструментальних засобів, які використовуються для створення системи;

Г) проект системи оброблення даних та генерування результатів (СОДГР) повинен містити:

- опис всіх алгоритмів і правил оброблення даних і знань;
- опис критеріїв вибору альтернатив;
- в якій формі та на яких носіях зберігаються процедури;
- можливі незначні відхилення від специфікацій, які дозволяють спростити застосування процедур і скоротити час на їх програмування;

Д) проект системи бази даних і знань (СБДЗ) повинен містити:

- типи даних і знань, способи їх зберігання та доступу до них;
- специфікацію форматів даних і знань та їх узгодження з процедурами (алгоритмами), які їх використовують;
- можливі модифікації та розширення форматів;
- можливі відхилення від специфікацій.

6.4.6 Опис програмного забезпечення системи

Складається з опису програмних модулів, з яких буде компонуватись система і створюватись цілісний програмний продукт.

Опис модуля повинен обов’язково містити:

- ім’я модуля та його функціональне призначення;
- типи вхідних та вихідних даних (числові, лінгвістичні, змішані);

- допустимі формати даних;
- метод оброблення даних, який реалізується модулем;
- вимоги до точності обчислень (якщо це необхідно);
- вимоги до швидкодії;
- метод (методику) забезпечення обчислювальної стійкості модуля;
- повідомлення, які може генерувати модуль в процесі виконання запрограмованої функції (наприклад, про неможливість завершити виконання операції, несумісність форматів даних).

6.4.7 Послідовність програмування модулів системи

Рекомендується така послідовність програмування модулів системи:

- програмування інтерфейсу (для того щоб врахувати вимоги замовника і продемонструвати йому зручність взаємодії з майбутньою системою, її можливості);
- програмування бази знань і даних;
- програмування функцій попередньої оброблення даних;
- програмування процедур подання результатів;
- програмування методів основної оброблення даних з метою створення математичних моделей і генерування альтернатив;
- програмування (при необхідності) спеціальних драйверів введення/виведення даних для нестандартних пристроїв.

6.4.8 Послідовність тестування системи

Для тестування використовують всі можливі запити, дані і функції. Тестування рекомендується розділяти на три фази: попереднє, загальне і остаточне:

- попереднє тестування стосується окремих модулів, якщо це можливо, і встановлюється принципова можливість виконання основних функцій системою в цілому;
- загальне тестування стосується узгодження функціонування всіх модулів у межах однієї системи;
- остаточне тестування призначене для встановлення правильності і точності реалізації запрограмованих функцій в системі.

6.4.9 Підготовка інсталяційної версії і експлуатаційної документації

Головним документом цього етапу є керівництво для користувача стосовно інсталяції системи та послідовності виконання окремих функцій.

Керівництво для користувача передбачає визначення засобів і носіїв інформації програмного забезпечення, вимоги до апаратури, особливості режимів застосування та версійність інстальованої системи тощо.

Бажаним для користувача є вибір прийнятної конфігурації системи виходячи із особливостей її застосування.

6.4.10 Упровадження системи на технічних засобах замовника

Впровадження системи на технічних засобах замовника може передбачати додаткові умови подальшого використання системи з урахуванням рівня підготовки користувачів системи.

Тому, зазвичай, описується послідовність дій стосовно інсталяції на технічних засобах замовника та початку експлуатації системи:

- інсталяція;
- навчання замовника;
- врахування особливостей (рівня) підготовки замовника, його готовності до експлуатації системи;
- врахування можливого організаційного опору – не всі члени команди замовника позитивно відносяться до новинок, особливо до тих, які можуть позбавити їх роботи.

6.4.11 Супроводження СППР

Процес супроводження впровадженої системи включає в себе такі дії:

- остаточне виправлення помилок в системі на етапі її експлуатації;
- розширення та модифікація функцій;
- врахування додаткових побажань користувача (замовника);
- формулювання доповнень до керівництва користувача.

6.5 Створення прототипу СППР

Альтернативним шляхом проектування та реалізації СППР є створення спочатку її прототипу, який є модульною демонстраційною версією системи з частково реалізованими функціями. Мета створення прототипу:

- підвищення рівня зацікавленості замовника в даній розробці;
- апробації основних ідей, які закладаються в систему (наприклад, тип інтерфейсу, методів генерування альтернатив та способів подання результатів);
- прискорення в цілому процесу проектування та реалізації системи.

Особливостями прототипу системи є таке:

- в прототипі реалізують тільки ті функції, які не потребують значних зусиль і часу, тобто, деяку підмножину функцій реально діючої системи;
- зазвичай, для прототипу не створюється або створюється мінімальна документація;
- тестування максимально спрощується;
- прототип створюється в максимально стислі строки з метою демонстрації замовнику можливостей системи.

Створення прототипу дозволяє досягти такого ефекту:

1) показати потенційному замовнику які принципові можливості та зручність надає СППР, а також в чому полягають її переваги перед подібними програмними продуктами;

2) розробник може оцінити в принципі можливість реалізації його проекту за допомогою вибраних засобів;

3) у випадку, коли виникають труднощі із встановленням істинних побажань замовника (тобто, замовник не може точно сформулювати свої побажання), то прототип дозволяє розв'язати і цю проблему;

4) вдалий прототип можна поступово трансформувати у справжню СППР;

5) виявлення інших потенційних замовників.

Існує два види прототипів: одноразового використання і еволюційного типу. Одноразовий прототип, зазвичай, ліквідується після демонстрації замовнику, а еволюційний після демонстрації розвивається далі, корегується і покращується з метою його доведення до остаточного продукту та зацікавлення інших замовників. Нарешті, еволюційний прототип може трансформуватись у систему, придатну для розв'язання практичних задач.

Для прискорення створення прототипів існують спеціальні інструментальні системи, які отримали назву систем прискореного створення прототипів (наприклад, RPS – rapid prototyping system).

6.6 Сучасні СППР ринку інформаційних послуг

Системи підтримки прийняття рішень набули широкого застосування в економіках передових країн світу, причому їхня кількість постійно зростає.

Сучасні СППР ринку інформаційних послуг охоплюють широке коло застосувань. Тому важно провести систематизацію різноманіття цих систем і виділити найбільш представницькі СППР, які задовольняють широкому спектру СППР ринку інформаційних послуг. Тому будемо розглядати ці системи за типами і за галузями використання.

6.6.1 Типи сучасних СППР

На рівні стратегічного управління використовується ряд СППР, зокрема для довго-, середньо- і короткострокового прогнозування, а також для фінансового планування, включаючи систему для розподілу капіталовкладень.

Орієнтовані на оперативне управління СППР застосовують в галузях маркетингу (прогнозування та аналіз збуту, дослідження ринку і цін), науково-дослідних та конструкторських робіт, в управлінні кадрами. Оперативно-інформаційні застосування пов'язані з виробництвом, придбанням та обліком товарно-матеріальних запасів, їх фізичним розподілом та бухгалтерським обліком.

Узагальнені СППР можуть поєднувати дві чи більше з перелічених функцій. У США в 1984 році було проаналізовано 131 типів СППР. Завдяки аналізу виявлено пріоритетні галузі використання систем. До них належать такі: виробничий сектор, гірничорудна справа, будівництво, транспорт, фінанси та урядова діяльність.

Комп'ютерна підтримка різних функцій за допомогою СППР має такий розподіл:

- оперативне керування — 30 %;
- довгострокове планування — 40 %;
- розподіл ресурсів — 15 %;
- розрахунки річного бюджету — 12 %.

Перелік найвідоміших комерційних СППР містить сотні назв. Наведемо ряд найбільш типових СППР, які стосуються проблем мікро- та макроекономіки:

«Симплан» — призначена для корпоративного планування;

«Прожектор» — фінансового планування;

«Джі-план» — загального планування;
«Експрес» — маркетингу, фінансів;
PMS — керування цінними паперами;
CIS — планування виробів;
PIMS — маркетингу;
BIS — керування бюджетом;
IFPS — інтерактивного фінансового планування;
FOCUS — фінансового моделювання;
ISDS — формування «портфеля замовлень»;
MAUD — індивідуального вибору ;

З ілюстративною метою нижче подано короткий опис деяких із систем.

6.6.2 Система «Симплан»

СППР Симплан (SIMPLAN) було створено в середині 70-х років з метою надання допомоги керівникам подолання невизначеності, притаманної корпоративному плануванню. Її призначення полягає у вивченні складних взаємозалежностей, що існують між діяльністю корпорації в галузях фінансів, маркетингу й виробництва та сукупністю математичних і логічних співвідношень, уведених до комп'ютера.

Ця система містить три центральні компоненти – фінансові моделі, моделі маркетингу та моделі виробництва. Призначення фінансових моделей полягає у тому, щоб показати ефективність різних варіантів фінансового стану фірми; моделі маркетингу використовуються для оцінювання майбутнього обсягу ринку в тій частині, якою має намір заволодіти компанія. Моделі виробництва застосовують для визначення питань, пов'язаних із витратами і плануванням, політикою в галузі щодо

товарно-матеріальних запасів, вимогами до робочої сили, вартістю та наявністю сировини, змінами в потужності обладнання і підприємства в цілому.

Система «Симплан» складається з таких підсистем:

- керування даними, яка забезпечує ефективне зберігання і вибір великих обсягів даних і має засоби управління ними;
- моделювання, яка дає змогу відбивати будь-які види зв'язків у галузі фінансів, маркетингу й виробництва в належній формі;
- одержання звітів для забезпечення генерації звітів для користувачів;
- контроль безпеки – це багаторівнева система контролю безпеки з метою обмеження доступу до даних та інформації;
- графічне відображення – містить множину форматів графічного відображення для візуального сприйняття діаграм і графіків;
- прогнозування з реалізованими методами лінійного прогнозування, експоненціального згладжування, адаптивного прогнозування;
- економетричний і статистичний аналіз, яка дає користувачеві змогу вирізняти значущу інформацію про взаємозв'язки, що характеризують розглядувані планові періоди.

За допомогою системи «Симплан» користувач може створювати нові функції і вводити їх до СППР. Моделі (разом з переліченими і пов'язаними з ними функціями) є організаційними складовими системи. Спочатку користувач задає режим керування, тобто позицію, з якої можна увійти в будь-який інший режим. Режим даних об'єднує засоби системи з управлінням даними. Режим аналізу містить набір релевантних економетричних і статистичних методів аналізу, прогнозування та мову моделювання системи «Симплан»; режим звітування — основа

генерування звітів; режим редагування має на меті подальше спрощення створення і використання моделей та звітів; графічний режим дає змогу ідентифікувати закономірності даних, використовуваних як база для прогнозування, розглядати розбіжності між практичними даними і прогнозами або бюджетами, а також забезпечує візуальне порівняння результатів реалізацій моделей, що ґрунтуються на різних системних припущеннях.

6.6.3 Система PIMS

При розробці системи PIMS (Profit Impact of Marketing Strategy) був узагальнений досвід торговельних операцій і ринкової діяльності сотень фірм, а також враховані різні чинники (поділ ринків збуту, розподіл капіталовкладень, структура управління, тощо).

Фірми – члени клубу PIMS регулярно дістають звіти, які стосуються кожного з аналізованих ринкових продуктів. Ці звіти відбивають стан справ з конкретним продуктом на фірмі у порівнянні із середнім світовим рівнем або з фірмами-конкурентами, пропонують для розгляду декілька стратегій короткострокового планування з оцінкою імовірностей тих чи інших наслідків використання стратегій. Крім того, у звітах наводиться оптимальна стратегія, яка вибирається з урахуванням можливих змін кон'юнктури на ринку і досвіду успішної діяльності у подібних ситуаціях. Системою можуть користуватися також представники фірм, які не входять до клубу PIMS. Будь-який користувач за певну плату може підключитися до БДРІMS за допомогою віддаленого терміналу через телефонну лінію.

Сеанс інтерактивної роботи з PIMS, зазвичай, розпочинається оглядом стану ринку певного продукту і триває у вигляді діалогу типу «Що, ... якщо». Запитання можуть стосуватися практично будь-якої

частини моделі планування і формулюватися у такому, наприклад, вигляді: «Який має бути рівень конкретного виду продукції, що випускається фірмою, і як він повинен змінюватися щороку, якщо потрібно отримати прибуток за перший рік у сумі не менш як 700 000 доларів, а наступний приріст прибутку має становити не менше 15 % за рік».

Використовуючи засоби оброблення запитів типу «Що, ... якщо», керівник має змогу звертатися до БД і до бази моделей мовою фінансового планування і фактично конструювати свої власні моделі, відтворюючи за їх допомогою уявні ситуації в інтерактивному режимі.

Запитання типу «Що відбудеться, коли внести певні зміни до даних або моделі?» потребує залучення математичних моделей (ідеться про статистичні моделі, імітаційне моделювання, методи математичного програмування, аналіз дерев рішень, тощо).

6.6.4 Система ISDS

Система ISDS (Investment Strategy Decision System) призначена для керівників, які відповідають за формування «портфеля замовлень» на наукові дослідження, розробки, випробовування та оцінювання дослідних зразків у великих організаціях. Головною особливістю цієї задачі є високий ступінь невизначеності кінцевих результатів планування, через що у довгострокових планах доводиться щорічно змінювати майже половину показників.

Система забезпечує виконання таких процедур:

- попередній добір пропозицій стосовно досліджень, розробок і випробувань;
- порівняльна оцінка нових пропозицій між собою та з розпочатими роботами;

- об'єднання дібраних пропозицій і виконуваних робіт в інвестиційні групи, кожна з яких формується згідно з програмними цілями, політикою і бюджетними обмеженнями;
- порівняльний аналіз розподілу довгострокових капіталовкладень;
- подання підсумкових даних з різних трендів зміни капіталовкладень;
- видача статистичної інформації, необхідної для звітності.

Поряд з основними підсистемами, орієнтованими на дані та моделі, до складу ISDS введено підсистему «історичних аналогій», в якій накопичується досвід практичного використання системи. Підсистема допомагає керівникові враховувати колишні успіхи і невдачі при формуванні варіантів планів довгострокових капітальних вкладень, дозволяє перевіряти правильність рішень в історичній перспективі, порівнюючи їх з аналогічними ситуаціями у минулому.

6.6.5 Система IFPS

Система IFPS (Interactive Financial Planning System) підтримує процеси розв'язування проблем, будуючи зрозумілі ділові ситуації. Основні моделі IFPS, завдяки яким система стала корисним інструментом для керівників, містять мову моделювання і структуру команд, які дають змогу описувати проблеми звичною для людини мовою і діставати результативні рішення (розв'язки) у табличному вигляді. IFPS здатна виражати співвідношення між клітками таблиці, інтерпретація значень яких цілком перебуває в розпорядженні користувачів.

Робота із системою починається з опису потрібної моделі мовою моделювання, який супроводжується введенням послідовності положень,

що визначають джерела даних для рядків і стовпців, а також співвідношень для обчислення значень розв'язків. При цьому користувач може викликати різні програми, вносити коментарі, визначати логічні умови, обмеження та сфери використання даних, виконувати процедури, пов'язані з аналізом ризику, і ряд інших функцій. Система дає змогу розв'язувати досить широкий спектр задач: добір балансових підсумків, розподіл прибутку за статтями доходів, передбачення змін валютних курсів, прогнозування, аналіз ризику розробки стратегії збуту продукції, збір науково-дослідних проектів, стратегічне планування, планування прибутку і бюджету, вибір між стратегіями закупівлі або виготовлення власними силами продукції, тощо.

6.6.6 Система MAUD

Система індивідуального призначення MAUD (Multi Attribute Utility Decomposition) поширена у Великобританії в «центрах зайнятості» для надання допомоги у виборі можливого місця роботи на підставі особистих уявлень клієнтів про бажаний характер майбутньої (можливої) діяльності. Підтримка прийняття рішень у цій специфічній ситуації (ситуація індивідуального вибору) забезпечується інтерактивністю та гнучкістю СППР: система працює разом з користувачем, розвиваючи і змінюючи його уявлення про проблему, структура і зміст якої мають бути описані в термінах багатокритеріальної оцінки альтернативних варіантів. Робота із системою розпочинається з короткого опису альтернатив (об'єктів), між якими буде проводитись вибір. Користувачеві, засобами звичної йому мови, пропонується дати багатокритеріальну оцінку кожного з можливих варіантів. Далі система перевіряє узгодженість інформації, поданої людиною, виявляє суперечності і визначає цінність інформації, що

надходить. Після цього інформація введенняється до системи і на основі концепції багатокритеріальної теорії корисності видаються пріоритети користувача, що дає змогу ранжувати об'єкти вибору. Упорядкований таким чином список варіантів супроводжується даними про важливість кожного з критеріїв оцінки. Під час роботи із системою можна вносити корективи, включати і виключати об'єкти і критерії, змінювати власні оцінки та пріоритети. Якщо система виявить суперечності в дії користувача, то вона відсилає його до тієї процедури, де ця суперечність виникла. Важливою характерною особливістю системи MAUD є те, що вона дає змогу переривати роботу із системою в будь-якому місці, причому подальше поновлення перерваної роботи можливе в зручний для користувача час без проведення додаткового налагодження.

6.6.7 Пакети і інтегровані системи

Система «Darwin», розроблена компанією Thinking Machines (Бедфорд, шт. Массачусетс), дозволяє будувати моделі на основі нейромереж і дерев рішень, а також використовувати методи візуалізації та класифікації даних.

Пакет «PowerPlay 5.0» фірми Cognos виконує багатовимірний аналіз даних, що включають до двох і більше мільйонів записів, у масштабі корпоративного підприємства (понад 2000 користувачів). Система дозволяє: будувати тривимірні графіки і діаграми, ранжувати дані, негайно повертатись до верхнього рівня ієрархії даних і систем меню, повністю обумовлених ОНР.

ROLAP-система «DSS Agent» компанії MicroStrategy (Вієнна, шт. Вірджінія) надає для побудови інформаційних сховищ інтегрований набір інструментів і методів об'єднання даних з різномірних джерел. Проект

«Pablo for Windows» фірми Andyne Computing (Кінгстон, Канада) пропонує СППР, що дозволяє переглядати узагальнені вибірки на основі даних з реляційних баз даних і маніпулювати ними.

Програмний пакет «Integrity Data Re-engineering Tool» виробництва компанії Vality Technology надає середовище програмування, яке можна використати для дослідження, стандартизації та інтегрування даних з різних джерел. «Integrity» виявляє нову інформацію і набори правил з оперативних даних, що дозволяє розроблювачам інформаційних сховищ планувати та визначати моделі даних, які правильно відображають складності реального світу.

Продукти зберігання даних фірми Red Brick Systems дозволяють: швидко розробляти і встановлювати додатки для керування; будувати запити до БД будь-якого розміру, які містять інформацію, зібрану з різнорідних джерел; забезпечувати найкращий доступ до узагальнюючої та докладної інформації в єдиній базі даних.

6.6.8 Розроблення російських виробників

Серед російських розробок слід відзначити такі.

Нейронно-мережевий пакет «STATISTICA Neural Networks» компанії StatSoft-Росія, який надає можливість автоматично одержувати ефективні рішення слабоструктурованих завдань, у яких нераціонально використовувати традиційні статистичні методи.

У системі реалізовано повний набір архітектур нейромереж, алгоритмів навчання (методи зворотного поширення, квазі-ньютонівський, Левенберга-Марквардта, Кохонена, квантування навчального вектора тощо), потужні засоби візуалізації даних, що допомагають оцінювати якість роботи мережі та оцінювати прогнози. Крім того, у систему

закладені генетичні алгоритми відбору вхідних даних, а також повний інтерфейс прикладного програмування (API), який дозволяє включати нейронні мережі в інші додатки. На основі методів штучного інтелекту реалізовано підсистему «Майстер розв'язання задач», який дозволяє автоматизувати вибір найкращої архітектури і побудови мережі.

Система «PolyAnalyst» представлена російською компанією Megaruter Intelligence як інструментарій для автоматичного витягування з даних вирішуючих правил, залежностей та інших знань, на основі яких можуть прийматися керуючі рішення.

В основі системи «PolyAnalyst» лежить набір методик і алгоритмів аналізу даних – як традиційних, так і сучасних – метод автоматичного виявлення розмитих нелінійних залежностей та інструментарій побудови довільних нелінійних регресійних моделей методами еволюційного програмування.

6.7 Проектування банківських СППР

У даному випадку СППР ринку інформаційних послуг у якості об'єктів виступають засоби доступу до даних і методи їх аналізу, системи збору, збереження, оброблення і передачі інформації, комунікаційні системи тощо., а у якості суб'єктів виступають аналітики, експерти і керівники.

6.7.1 Аналітичний програмний комплекс

Взаємини між суб'єктами СППР, які регламентують від постановки проблеми до ухвалення рішення, розрізняють для кожної окремої організації. Тому такий опис потребує окремого обговорення.

Основна увагу тут приділимо структурі і можливостям аналітичного програмного комплексу (АПК).

Комплекс АПК є універсальним інструментальним засобом збору, консолідації, оброблення і аналізу великих обсягів інформації. Він здатний, на основі наявних даних, наприклад про клієнтів банку, філіях, зроблених банківських операціях, послідовності зовнішніх подій тощо, виконувати такі функції:

- аналізувати поточний стан головного банку і його філій, а також клієнтів банку;
- оцінювати стан банків-кореспондентів;
- прогнозувати поведження різних показників;
- автоматично відслідковувати, що відбувається і чи насуваються критичні події;
- аналізувати взаємозв'язок подій і процесів, що відбуваються одночасно, а також подій, зміщених у часі;
- на основі виявлених взаємозв'язків прогнозувати поведження одних показників у залежності від значень інших.

АПК складається з трьох підсистем: підсистеми збору і збереження даних, підсистеми доступу до даних, аналізу і репортингу та підсистеми адміністрування.

6.7.2 Підсистема збору і збереження даних

Фактично під цим терміном розуміють проект стосовно створення банківського сховища даних. До його функцій відносяться :

- перетворення різнорідних даних з різних джерел до єдиного формату;

- перевірка логічної коректності і повноти інформації, що вводиться;
- контроль за виконанням регламенту надходження і повноти інформації, що вводиться;
- збереження поточних та історичних даних;
- забезпечення зручного доступу до збереженої інформації.

Першоджерелами інформації є основна АБС банку, інші інформаційні системи, що реєструються, платна інформація таких агентств, як Reuters, і зведена інформація з мережі Інтернет. В АПК передбачено також введення даних вручну, що дозволяє експертам вводити задані коефіцієнти.

На сьогодні найбільш ефективним рішенням є організація сховища даних на основі реляційної СУБД. Сучасні СУБД забезпечують високу продуктивність на аналітичних запитах і мають цілий ряд переваг перед MOLAP-серверами, у тому числі підтримують можливість використання існуючої в банку СУБД, збереження та оброблення набагато більших обсягів даних, а також забезпечують відкритість платформи. Для реалізації невеликого проекту процедури вибору даних із джерел, очищення і перетворення їх до єдиного формату, а завантаження у сховище можна реалізувати без використання спеціального ПЗ. Однак у великих банках це сильно ускладнить розвиток і підтримку системи у майбутньому. Найбільш ефективним шляхом є використання Sybase Adaptive Server IQ для організації сховища і Ascential DataStage для керування ним.

Інформацію, що утримується в сховищі, необхідно також розділити для потреб різних департаментів по вітринах даних. Крім даних, що надходять з центрального сховища АПК, вітрина може містити інформацію, отриману з інших джерел.

6.7.3 Підсистема доступу до даних, аналізу і репортингу

Для безпосередньої оброблення даних у сховищі пропонують використовувати програмне забезпечення Business Objects, який є одним з лідерів на ринку засобів доступу і репортингу, що максимально відповідає сьогоденішньому рівневі вимог до аналітичних інформаційних систем.

Ядром підсистеми є семантичний процесор, у якому визначені правила перетворення фізичних об'єктів БД у терміни предметної області. Такий підхід, який реалізовано на рівні ядра, дозволить користувачам самостійно брати інформацію із сховища і вітрин даних, виконувати OLAP-аналіз і створювати професійно оформлені звіти.

6.7.4 Інструмент опису семантичного шару

Семантичний процесор створюється за допомогою BusinessObjects Designer. Програміст або адміністратор, який знає як інформація «лежить» у базі, створює каталог термінів кінцевого користувача і визначає для кожного терміну метод одержання даних (фрагменти запиту SQL). Тут також задаються вихідні ієрархії вимірів і вихідні формати відображення об'єктів у звітах. У створюваних словниках реалізована можливість коректно працювати на довільних структурах відносин і зв'язків між таблицями, не обмежуючись найпростішою «зіркою» або «сніжинкою».

Уся принадність словників полягає ще й у тому, що в них дані та їх сукупності вибудовуються у виді багатовимірного куба. У такий спосіб, при їх подальшому аналізі та відображенні у звітах, можна з легкістю вибирати осі подання даних із усієї доступної множини осей, надаючи повну інформацію для візуального аналізу.

Деякі характеристики об'єкта можна визначити складною функцією, що задається надалі звичайною змінною. Шляхом групування об'єктів у словнику можна створити будь-яку ієрархію даних.

6.8 Основні можливості звітів у АПК

6.8.1 Інструментарій АПК

При побудові різних звітів АПК має широкі можливості масштабування, «розгортання» формул (drill-down) і побудови похідних графіків, а також спеціальні аналітичні функції. Усе це можливо за рахунок великого інструментарію.

Першим інструментом є комплекс продуктів BusinessObjects, BusinessMiner і Set Analyzer. З їхньою допомогою реалізуються такі функції:

- прогнозування; на підставі наявних даних про значення показників система може прогнозувати їх поведження у майбутньому; результат прогнозу відображається на графіку;

- аналіз «що буде, якщо»; використовуючи отриманий прогноз, користувач може виконувати аналіз «що буде, якщо». У процесі його виконання система знаходить закономірності, що враховують взаємозв'язки між показниками. Користувач може задати конкретні значення одного або кількох показників для різних моментів часу і у результаті виконання аналізу одержати картину поведження інших значень. Спосіб врахування взаємозв'язку можна добудовувати. При зміні значення показника в якій-небудь точці може змінюватися поведження як на його власній кривій, так і інших кривих. Користувач може задати черговість, з якою система буде враховувати зазначені закономірності;

– моніторинг; існує багато методик, які дозволяють аналізувати стан справ у тій чи іншій галузі і сигналізувати про несприятливу ситуацію (або, навпаки, про можливість здійснення вигідних операцій). У той же час задача підрахунку і перевірки великої кількості необхідних показників для різних галузей завжди була надзвичайно трудомісткою;

– аналіз клієнтської бази; за допомогою Set Analyzer фахівці банку можуть здійснювати сегментацію й оцінювати рух клієнтів між сегментами. Цей вид аналізу розрахований на маркетологів банку і є одним з компонентів стратегії банку по керуванню взаєминами з клієнтами.

За допомогою інструментів моніторингу, реалізованих в АПК, у системі автоматично ведуться регулярний розрахунок і перевірка значень великого числа показників. У випадку порушення заданих обмежень система інформує про це банківського аналітика.

Сутність моніторингу можна показати на прикладі класичної панелі керування технічним об'єктом, на яку виведена велика кількість різних приладів і індикаторів (стрілок, лампочок тощо.). У нормальному робочому режимі на більшість цих приладів персонал не звертає уваги, але якщо раптом виникла екстренна (нештатна) ситуація, то на панелі керування висвітлюється відповідна лампочка, що вказує на яку ділянку варто звернути увагу. Так само у процесі моніторингу набір своєрідних «червоних лампочок» сигналізує про позаштатні ситуації.

У дійсній версії АПК контролю підлягають абсолютне значення показника, його відносні зміни, абсолютне і відносне відхилення від моделі, а також монотонність.

Визначаючи параметри моніторингу можна використовувати граничні умови таких типів:

– абсолютна верхня границя. Вона спрацьовує, якщо значення показника виявилось вище встановленого числа;

- абсолютна нижня границя; спрацьовує, якщо значення показника виявилось нижчим встановленого числа;
- відносне відхилення. Вказують величину припустимого відхилення у відсотках. Спрацьовує, якщо значення показника змінилося більш ніж на задану величину у порівнянні з попереднім днем;
- абсолютне відхилення від моделі. За модель вибирають інший показник, і вказується величина припустимого відхилення від нього. Спрацьовує, якщо модуль різниці між значеннями показника і моделі перевищив величину заданого відхилення;
- відносне відхилення від моделі. Цей випадок аналогічний попередньому, однак тут величина відхилення задається у відсотках від значення моделі;
- монотонність. Вказують кількість днів до поточної дати, за які перевіряється монотонне зниження (зростання) значення показника. Спрацьовує, якщо на цьому інтервалі показник не монотонний.

6.8.2 Рейтинг-аналіз

До складу АПК входить спеціальний набір програмних засобів і інструментів, який дозволяє аналізувати дані про довільний набір об'єктів і формувати на підставі їхніх показників різні рейтинги. Рейтинг-аналіз забезпечує оцінювати як поточний стан сукупності об'єктів, так і їх стан у минулому. При цьому здійснюється порівняння отриманого результату із станом інших аналогічних сукупностей або із заданими попередньо середніми характерними значеннями. В АПК реалізовано широкий спектр можливостей перегляду різних діаграм і складання рейтингів-звітів.

Ще одним інструментом аналізу і подання даних є WebIntelligence, що має могутні засоби побудови звітів через веб-браузер. Користувач може будувати і переглядати звіти, задаючи довільні запити до баз даних знову ж у термінах свого бізнесу. WebIntelligence використовує ті ж словники, що і «товстий клієнт» BusinessObjects.

Підсистема збору, аналізу і репортинга може бути розширена за рахунок відкритості ПЗ Business Objects. Банківські розроблювачі можуть самостійно створити якесь налагодження, орієнтоване на розв'язання конкретних аналітичних задач. Можна також скористатися ПЗ таких виробників, як SPSS.

Окремо необхідно виділити те, що аналітичні звіти зберігаються в одному місці. За бажанням можна завантажити будь-який звіт з домена документів (якщо це право регламентоване адміністратором системи).

6.8.3 Довідники

Механізм роботи з довідниками дозволяє користувачеві переглядати інформацію з окремих і балансових рахунків, філіям, валютам і клієнтам. При перегляді користувач може робити сортування, пошук потрібного елемента, переходити між довідниками за посиланнями і здійснювати фільтрацію (наприклад, переглядати тільки клієнтські особові рахунки, тільки рахунки великих клієнтів або тільки особові рахунки, відкриті на даному балансовому рахунку).

Користувачеві надається можливість створювати нестандартні довідники, відбираючи інформацію для перегляду, а також добудовувати перехресні посилання на інші довідники.

Довідники можуть бути зв'язані між собою: за елементом одного можна викликати інший, зв'язаний з ним довідник або повністю, або у

відфільтрованому за значенням елемента вигляді. Наприклад, логічно назвати зв'язаними рівні плану рахунків: розкриваючи розділ плану рахунків, користувач звичайно хоче бачити також балансові рахунки 1-го порядку даного розділу, розкриваючи кожний з них - балансові рахунки 2-го порядку тощо.

Реалізована можливість перегляду попередньо введеної додаткової інформації з довідника. Зміст будь-якого довідника залежить від дати його перегляду. У випадку якщо після останнього перегляду довідника в системі здійснювалось наочування або редагування даних і існує імовірність, що зміст довідника змінився, АПК видасть відповідне повідомлення і запропонує оновити інформацію. Користувач може зробити оновлення або відмовитися від нього.

Реалізовано можливість швидкого переключення між різними довідниками. Наприклад, один з них може мати невелику кількість полів і швидко завантажуватися, а другий, навпаки, мати велику кількість полів; відповідно, такий довідник завантажується повільніше. Якщо при перегляді першого довідника з'являється необхідність у більш докладній інформації, то можна швидко переключитися на другий.

Зміни конфігурації довідника зберігаються у пам'яті системи для того самого користувача до наступного сеансу роботи з ним. Інший користувач, відповідно, може налаштувати конфігурацію для себе.

6.8.4 Підсистема адміністрування

Інструменти адміністрування розділені на дві частин – засоби адміністрування сховища даних (Sybase і Ascential) і засоби керування системою аналізу і звітності (Business Objects), представлені в BusinessObjects Supervisor. Перші дозволяють обслуговувати сховище

даних, побудувати фізичні форми доступу до об'єктів бази, ETL-процедури, другі – регламентувати роботу кінцевих користувачів.

Застосування системи BusinessObjects в АПК для банку обґрунтовано і з погляду масштабу системи. ПО Business Objects традиційно проектувалося для використання в корпоративних системах. Організація єдиного репозиторія дозволяє істотно підвищити інформаційну безпеку підприємства. Фізично репозиторій розташовується в реляційній СУБД і містить повну інформацію про користувачів системи та їхніх прав доступу до даних і функцій, а також семантичний процесор і документи. Наявність єдиного репозиторію спрощує адміністрування системи.

6.9 Перелік аналітичних і управлінських задач, які розв'язуює АПК

У загальному випадку, для визначених структури і можливостей аналітичного комплексу, можна окреслити повний перелік задач, які можна вирішувати банку за допомогою АПК. Завдяки цьому можуть прийматись зважені рішення. : До таких задач можна віднести:

- аналіз кредитного портфеля (у тому числі стан кредитного портфеля, структура кредитного портфеля, прибутковість кредитного портфеля);

- аналіз кредитного ризику (у тому числі розподіл по групах ризику, оцінка заборгованості, зваженої із урахуванням ризику, аналіз структури портфеля цінних паперів);

- аналіз доходів і витрат банку (у тому числі аналіз прибутковості активів, аналіз витрат за пасивними операціями, розрахунок структури доходів і витрат, аналіз динаміки доходів і витрат банку);

– аналіз власних засобів і капіталу банку (у тому числі аналіз забезпеченості власними засобами);

– аналіз стану клієнтської бази (у тому числі аналіз прибутковості клієнтів, структурний аналіз клієнтської бази, якісний аналіз клієнтської бази, аналіз клієнтських платежів, зміни структури клієнтської бази, зміни затребуваності продуктів за групами клієнтів, виявлення прихованої афільованості клієнтів, аналіз фінансового становища кредиторів/позичальників, фундаментальний аналіз підприємств і потенційних клієнтів);

– аналіз фінансового стану банку (у тому числі аналіз фінансової діяльності і ранжирування філій, аналіз економічних нормативів діяльності банку, методика аналізу фінансової стійкості, експрес-аналіз, комплексний і порівняльний аналіз банків, дослідження положення свого банку щодо конкурентів, розрахунок лімітів за видами операцій, за інструментами і за контрагентами, аналіз контрагентів на ринку МБК, розрахунок лімітів по МБК);

– аналіз структури балансу (у тому числі аналіз активів-пасивів за термінами розміщення, управління активами-пасивами, аналіз динаміки балансових статей, аналіз оборотного балансу, аналіз, моніторинг і прогноз дотримання встановлених нормативів, складання і аналіз загальної фінансової звітності (ЗФЗ), аналіз балансу за термінами, аналіз позабалансових рахунків, аналіз рентабельності банківських операцій і банківських продуктів, аналіз рентабельності підрозділів, трансферне ціноутворення);

– бюджетування, планування і контроль (у тому числі структура планування – за центрами звітності (ЦЗ), за центрами фінансової відповідальності (ЦФВ), за центрами прибутку (ЦП), за центрами витрат

(ЦВ), за центрами ціноутворення (ЦЦ), види планування – від досягнутого, за вимогами/зобов'язаннями, за платежами, контроль виконання - план/факт, розбіжність абсолютна і процентна, тренди);

– аналіз кадрових ресурсів (у тому числі аналіз структури трудових ресурсів за рівнем освіти, спеціальностях і інших групах, аналіз структури філій і підрозділів за кількістю і якістю персоналу, аналіз ефективності діяльності філій, порівняння витрат на зарплату за різними групами у різних філіях в залежності від прибутку філій);

– інформаційна база новин і аналіз зовнішньої економічної інформації.

Деякі з перерахованих задач мають універсальний характер і не відносяться винятково до банківської сфери. Частина з цих задач можна цілком віднести до банківської сфери для розв'язання банківських задач.

6.10 Етапи впровадження АПК та її налагодження

6.10.1 Вартість впровадження АПК

Побудова і впровадження системи підтримки прийняття рішень – це якісний позитивний стрибок у подальшому розвитку організації. Однак для здійснення подібного стрибка необхідно вкласти деяку критичну масу грошей і часу, після чого можлива віддача.

Бар'єром для впровадження багатьох готових комплексів стає вартість. Більшість керівників ще не готові викладати значні кошти за розпливчасті можливості та обіцянки. Адже замовникові надається в основному інструментарій, що включає засоби розробки і набір готових методів.

Найчастіше для налагодження СППР під конкретну систему ведення бізнесу потрібно витратити стільки ж зусиль, скільки вистачило б на побудову своєї власної системи, що задовольняє сучасному розвитку технологій. Досвід засвідчує, що термін введення готової системи в експлуатацію складає від шести місяців до одного року.

Зауважимо, що створювати систему краще самому, використовуючи передові розробки в області програмного забезпечення – це найбільш дешевий варіант системи. При цьому принципово необхідною задачею буде формування інформаційного поля, орієнтованого на аналітичну оброблення – як за структурою, так і за змістом.

Розвиток системи представляє собою введення нових модулів, що відповідають за окремі задачі і нерозривно пов'язані з іншими компонентами системи. У цьому випадку термін впровадження системи явно збільшиться у два рази, однак система буде включати більшу розмаїтість у засобах аналітичної оброблення.

Розробка і впровадження системи підтримки прийняття рішень, подібної описаної вище, відбуваються у відповідності до наступного плану.

6.10.2 Розробка АПК

Розробка АПК передбачає виконання ряду етапів. Основними є такі:

1. Визначити потреби і описати основних користувачів системи.
2. Визначити бажані результати функціонування системи.
3. Визначитися з джерелами даних.
4. Розробити інформаційну модель системи.
5. Вибрати спосіб збереження даних і тип сховища.
6. Вибрати систему візуалізації і аналізу даних.

7. Розробити технологічну модель системи.
8. Підготувати графік впровадження системи і

відповідальних виконавців.

Повна реалізація цих етапів забезпечує передумови створення ефективного АПК.

6.10.3 Упровадження і налагодження системи

Упровадження АПК передбачає виконання низки заходів. Зокрема, можна виділити такі важливі моменти:

1. Закупівля і встановлення програмного забезпечення.
2. Розмежування користувацьких прав доступу до системи.
3. Підготовка структури бази для заповнення даними і встановлення взаємозв'язків.
4. Наповнення сховища даними.
5. Пробна реалізація однієї із передбачених проектом під задач СППР.
6. Демонстрація й оцінювання реалізованих можливостей системи.
7. Корегування подальших планів стосовно подальшого розвитку системи.
8. Реалізація підзадач, що залишилися.

6.10.4 Супровід і подальший розвиток системи

Подальша експлуатація створеного АПК передбачає регулярну роботу з підтримання життєдіяльності системи. Основними етапами цієї роботи є такі:

1. Поточне адміністрування системи.
2. Навчання користувачів.
3. Подальша робота стосовно модернізації системи.

Запитання та завдання для самоконтролю

1. Що входить до вимог замовника СППР?
2. Дати характеристику основним вимогам при проектуванні СППР.
3. Які існують фактори ризиків під час проектування та реалізації СППР?
4. У чому полягає менеджмент ризиків проектів з розробки СППР?
5. Описати основні етапи розроблення та реалізації СППР.
6. Які елементи містить формулювання постановки задачі проектування СППР?
7. Які елементи містить аналіз можливості реалізації СППР?
8. Описати формулювання типових вимог користувача СППР при прогнозуванні динаміки часових рядів.
9. Які елементи застосовують при створенні проекту СППР?
10. Що передбачає опис модуль програмного забезпечення системи СППР?

11. Яка рекомендована послідовність програмування модулів системи?
12. Що містить послідовність тестування системи?
13. Як виконують упровадження та супроводження системи СППР на технічних засобах замовника?
14. Описати послідовність створення прототипу СППР.
15. Дати характеристику сучасних СППР на ринку інформаційних послуг.
16. У чому полягає проектування банківських СППР?
17. Описати функції підсистеми збору і збереження даних.
18. Які особливості має підсистема доступу до даних, аналізу і репортингу?
19. Що описує інструмент опису семантичного шару?
20. Описати особливості інструментарію АПК.
21. Що таке рейтинг-аналіз?
22. Описати особливості підсистеми адміністрування.
23. Дати характеристику переліку аналітичних і управлінських задач, які розв'язує АПК.
24. Описати основні етапи впровадження АПК та її налагодження.
25. Дати характеристику етапів розроблення АПК.
26. Що треба враховувати при впровадженні і налагодженні системи?
27. Сформулюйте умови успішної реалізації СППР.
28. Опишіть структуру вимог замовника інформаційної системи. Наведіть приклад таких вимог.
29. Наведіть основні фактори ризику, які зустрічаються під час проектування і реалізації СППР.

30. Надайте описання процесу ідентифікації та менеджменту ризиків під час розроблення інформаційних систем.
31. Які заходи виконавця проекту гарантовано зменшують ризик невиконання проекту?
32. Укажіть етапи процесу розроблення та реалізації СППР.
33. Які елементи має технічне завдання на проект?
34. Наведіть основні вимоги користувача і функціональні вимоги до СППР для прогнозування часових рядів.
35. Опишіть основні елементи проекту СППР.
36. Вкажіть послідовність програмування модулів та тестування інформаційної системи.
37. Розкрийте особливості проектування і реалізації СППР на основі створення її прототипу.
38. Перелічіть недоліки та переваги прототипу.
39. Що означає термін «еволюціонуючий прототип»?
40. Охарактеризуйте основні СППР, що пропонується на ринку інформаційних послуг.

РОЗДІЛ 7 ІНФОРМАЦІЙНІ СИСТЕМИ ПІДТРИМКИ ПРИЙНЯТТЯ РІШЕНЬ НА ОСНОВІ РОЗПОДІЛЕНИХ СИСТЕМ

Розподілена система – це комплекс незалежних комп'ютерів, які сприймаються користувачем єдиною об'єднаною системою. Розподілена інформаційна система (РІС) є розподіленою системою з єдиними інформаційними ресурсами. Доступ до інформаційних ресурсів регламентується встановленими правилами доступу до неї, процесами оброблення, передавання, зберігання, перетворення й використання інформації.

Розподілені інформаційні системи підтримки прийняття рішень (РІСППР) використовують РІС як середовище для реалізації розподілених алгоритмів прийняття рішень на ресурсах розподіленої системи.

7.1 Архітектура розподілених інформаційних систем

7.1.1 Узагальнена структура розподіленої інформаційної системи

Сучасний рівень розроблення РІС зумовлено кількома факторами, основними з яких є:

1) значними темпами зростання продуктивності комп'ютерів з одночасним зменшенням вартості оброблення інформації. Наприклад, при вартості комп'ютера у 100 млн дол. в 50-х роках при швидкості одна операція на секунду критерій ціна/кількість операцій становив 10^8 долар/операцій/с. Для сучасних систем швидкість ПК складає 10^7 операцій/с при вартості 1000 доларів, тобто критерій ціна/кількість операцій становить 10^{-4} долар/операцій/с. Отже темпи підвищення продуктивності з урахуванням вартості становлять 10^{12} ;

2) широким застосуванням локальних комп'ютерних мереж, починаючи з 80-х років минулого століття, при швидкості обміну інформацією 10 – 1000 Мбіт/с і вище;

3) розвитком телекомунікаційних систем і глобальних комп'ютерних мереж WAN. Збільшення швидкості обміну даними становить більше 10^4 від перших представників (arpanet, milnet) з швидкістю передавання інформації 56 Кбіт/с до сучасних систем в Internet, Intranet, VPN із швидкістю обміну даними 1-10 Гбіт/с і вище;

4) розвитком теорії й практики створення програмних засобів розподілених БД (БД) та засобів графічного інтерфейсу для них (Oracle, SQL Server, Informix);

5) впровадженням еталонної моделі взаємодії відкритих систем OSI;

6) мобільністю надання інформаційних сервісів (послуг) і створенням потужних обчислювальних середовищ на основі GRID- систем і систем хмарних обчислень (cloud computing).

Комп'ютери ПІС зазвичай мають власні операційні системи (ОС). Узагальнену структурну організацію ПІС для комп'ютерів А, В, С з локальними операційними системами (ЛОС) показано на рис. 7.1.

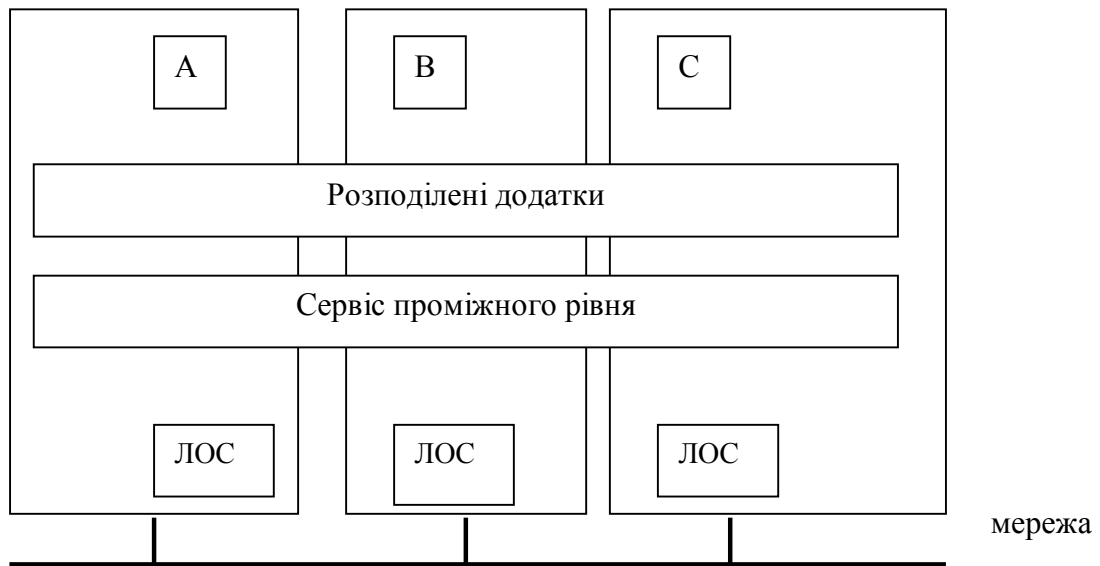


Рис.7.1 Узагальнена структурна організація РІС

Прикладний рівень розподілених додатків забезпечує єдине середовище взаємодії додатків. Сервіси проміжного рівня надають єдине середовище додаткових сервісів при взаємодії з операційною окремих підсистем А, В, С.

До основних задач створення РІС відносяться:

1. Організація з'єднання користувачів із ресурсами, яке спрощує обмін інформацією, зокрема, мультимедійною.
2. Забезпечення прозорості, яка полягає у прихованні для користувачів процесів і ресурсів на множині комп'ютерів.
3. Забезпечення відкритості РІС, яка передбачає наявність стандартних засобів або служб доступу до системи широкому колу користувачів.
4. Масштабованість, яка полягає у можливості розширення й модифікації РІС.

Найбільш поширеними є масштабованість:

- за розмірами (підключення користувачів і ресурсів);
- географічна масштабованість (за місцеположенням);
- адміністративна (тобто взаємодія між собою адміністративно незалежних організацій).

Організація з'єднання у глобальних мережах передбачає підтримку віртуальних з'єднань за допомогою схем групової роботи. Віртуальні організації можуть включати десятки і сотні організацій, які розглядаються як члени однієї групи для розв'язання спільних задач із використанням систем групового доступу.

Організації можуть виступати як у якості клієнтів, так і серверів з розподіленими ресурсами. Організація за клієнт-серверними технологіями передбачає різні варіанти реалізації проміжного рівня РІС, адаптованість ОС до забезпечення сервісів з широким спектром вимог. Зокрема, клієнтам РІСППР можуть надаватись широкий спектр послуг для роботи у розподіленому інформаційному, апаратному і програмному середовищі, або надаватись права для роботи в існуючих розподілених системах з визначеною архітектурою і функціональними можливостями.

7.1.2 Основні характеристики розподілених систем

7.1.2.1 Прозорість

Основні характеристиками розподілених систем є: прозорість, відкритість, масштабованість.

Основними формами прозорості є:

- 1) прозорість доступу – приховує відмінності у поданні даних і доступу до ресурсів;

- 2) прозорість місцеположення – приховує від користувача фізичне місцеположення ресурсу;
- 3) прозорість перенесення – приховує факт переміщення ресурсу в інше місце;
- 4) прозорість зміни місцеположення – приховує переміщення ресурсу під час обробки даних;
- 5) прозорість реплікації – приховує факт реплікації від користувача;
- 6) прозорість паралельного доступу – приховує факт сумісного використання ресурсу кількома конкуруючими процесами користувачів;
- 7) прозорість відмови – приховує факт виникнення відмов в системі й відновлення ресурсу;
- 8) прозорість схоронності – приховує факт розміщення ресурсу на диску або кешування ресурсу.

Прозорість місцеположення досягається, зокрема, наданням логічних імен.

Прозорість доступу дозволяє приховати різний формат передавання даних для різних процесорів (від молодшого байта до старшого, молодший – молодший), а також відмінності у поданні імен файлів тощо.

Прозорість зміни місцеположення актуальна для мобільних розподілених систем, в яких відбувається пересування процесорів у процесі обробки даних.

Прозорість реплікації може значно зменшити продуктивність роботи РІС. Несуперечливість прозорості паралельного доступу, наприклад, в БД, підтримується механізмом взаємоблокувань або використанням транзакцій.

Прозорість відмов і приховування збоїв є однією з найскладніших проблем побудови РІС і забезпечення її відмовостійкості. Визначення відмов і збоїв може приводити до сповільнення роботи окремих компонентів системи.

Прозорість схоронності маскує реальне розміщення ресурсу на диску і на віртуальному диску. Воно характерно для об'єктно-орієнтованих БД, які мають можливість безпосереднього виклику методів.

Існує баланс між ступенем прозорості й продуктивності системи. У разі забезпечення прозорості проміжного сервера може бути заборонено доступ до кінцевого сервера, із якого отримують дані.

Прозорість реплікації за умови підтримки їх цілісності може виявитись недоцільною через великі часові витрати.

7.1.2.2 Відкритість

До основних характеристик відкритості РІС відносяться:

1) наявність загальної специфікації інтерфейсу сервісів РІС, яка достатня для реалізації інтерфейсу різними виробниками і нейтральною для різних реалізацій інтерфейсу. Така нейтральність і самодостатність забезпечує простоту перенесення окремих програмних компонент системи і спроможність їх взаємодіяти для різних реалізацій системи на різних платформах (у різних ОС);

2) гнучкість конфігурування системи з різних компонентів системи різних виробників. Це можуть бути компоненти з одних ОС, які переносяться на іншу, зокрема, іншу файлову систему. Практично така відкритість не є характерною для сучасних ОС, оскільки вони є монолітними. При забезпеченні масштабованості необхідно розв'язати задачі. При масштабуванні за розміром доцільно використати децентралізовані сервіси, дані та алгоритми.

7.1.2.3 Масштабованість

Географічна масштабованість породжує низку проблем, зокрема, недотримання принципу синхронізації зв'язку характерну для локальних і глобальних мереж. Синхронізація WAN пов'язана з великими часовими витратами. Наприклад, для LAN запит клієнтом сервісу блокується до отримання відповіді і становить сотні мікросекунд. В інтерактивних додатках в РІС на основі WAN ці затримки можуть складати сотні мілісекунд.

При географічній масштабованості підтримується зв'язок «точка-точка», а не ширококомовний зв'язок. Це породжує проблему масштабованості централізованих рішень при застосуванні масштабування за географічною ознакою. При географічній масштабованості збільшується ризик здійснення атак з інших регіонів, а процедура узгодженості потребує реєстрації користувача з однієї області (або домену) в іншому регіоні (в інших доменах).

Існуючий механізм дружніх стосунків між контролерами окремих доменів (наприклад, технологія Windows NT) не завжди є ефективним розв'язком цієї проблеми.

До основних технологій масштабування відносяться:

- 1) приховування часу очікування зв'язку;
- 2) розподілення;
- 3) реплікація, тобто розмноження компонентів систем.

Приховування часу при географічному масштабуванні полягає у можливому уникненні відповіді на запит віддаленого сервера. Додатки розраховують на асинхронний зв'язок, який передбачає переривання роботи додатку при відповіді на раніше поданий запит. При перериванні

процесу додаток викликає обробник запитів. Це характерно для РІС пакетної обробки даних і паралельних додатків.

Для інтерактивних РІС частину обчислень доцільно перекладати із сервера на клієнта. Такий підхід підтримується в системах Internet, Intranet із застосуванням Java- аплетів, які передаються клієнту.

Масштабованість за технологією розподілення передбачає розділення компонентів на маленькі частини й рознесення їх по системі. Приклад: сервісу DNS імен Internet, в яких імена організуються у вигляді дерева доменів. Домени розділяють адреси на зони, які не перетинаються. У WWW застосовуються URL-адреси документів. Користувачам вони представляються єдиною системою документообігу. Однак, середовище WEB фізично рознесено по множині серверів, а ім'я сервера визначається за URL- адресою документа. При реплікація збільшується продуктивність РІС і поліпшується доступність до інформації.

7.2 Концепція апаратних і програмних рішень

7.2.1 Класифікації РІС

Єдиної класифікації РІС не існує. Визначальним для обробки даних в РІС є доступ процесора до пам'яті та кількість процесорів. У зв'язку з цим мультипроцесорні розподілені системи поділяють на:

1) мультипроцесорні РІС із загальним для процесорів модулями пам'яті;

2) мультикомп'ютерні РІС із локальною пам'яттю для кожного процесора.

За мережами зв'язку між процесорами і пам'яттю РІС поділяють на: системи з шинною архітектурою та з комутованою архітектурою.

Системи із шинною архітектурою (рис. 7.2.) мають загальний доступ до єдиної шини, через яку здійснюється обмін даними.

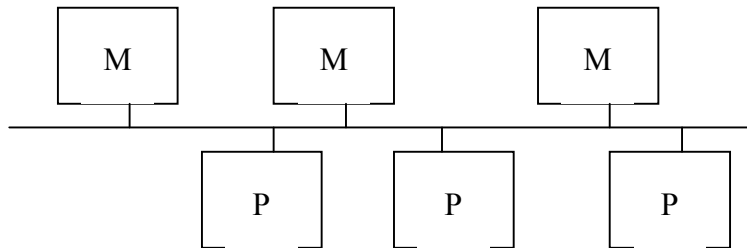


Рис. 7.2 Мультипроцесорна система із шинною архітектурою

У комутованих мультипроцесорних системах (рис.7.3) використовуються комутатори чи комутаційне середовище, яке дозволяє підключатися процесорам до певних модулів пам'яті.

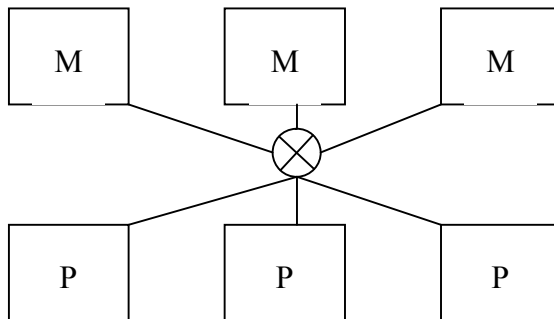


Рис. 7.3 Комутована мультипроцесорна система

Мультикомп'ютерні системи поділяються на мультикомп'ютерні РС із шинною організацією (архітектурою) (рис. 7.4) і з комутованою архітектурою (рис. 7.5).

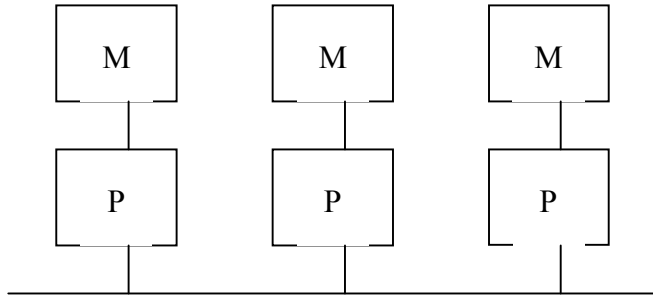


Рис.7.4 Мультикомп'ютерна РС із шинною архітектурою

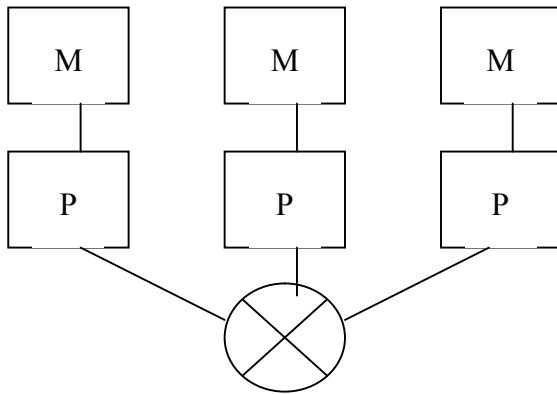


Рис.7.5 Мультикомп'ютерна система з комутованою архітектурою

Мультикомп'ютерні системи поділяють на:

- гомогенні, в яких застосовується одна мережна технологія;
- гетерогенні з використанням кількох мережних технологій.

У корпоративних комп'ютерних мережах популярною є архітектура мережі на основі комбінації кількох мережних технологій, зокрема, FDDI і Ethernet (рис. 7.6).

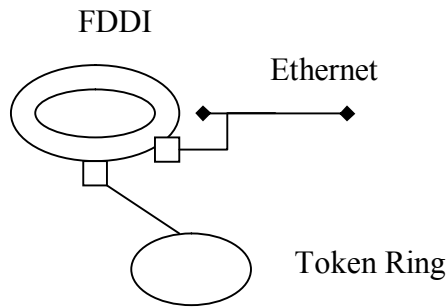


Рис.7.6 Корпоративна комп'ютерна мережа

Для одночасного доступу до пам'яті використовують кілька підходів у мультипроцесорних системах:

1. Застосування кешів (рис. 7.7).

Під час роботи системи здійснюється реплікація даних у вигляді сторінок (наприклад, розміром 4КБ або 8 КБ). Недоліками застосування кешів є:

дублювання даних у пам'яті;

необхідність узгодженості даних для різних процесорів;

обмежена масштабованість.

2. Застосування комутованих матриць (рис.7.8).

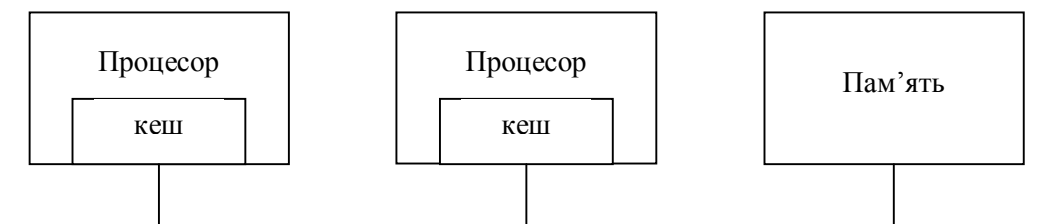


Рис. 7.7 Застосування кешів

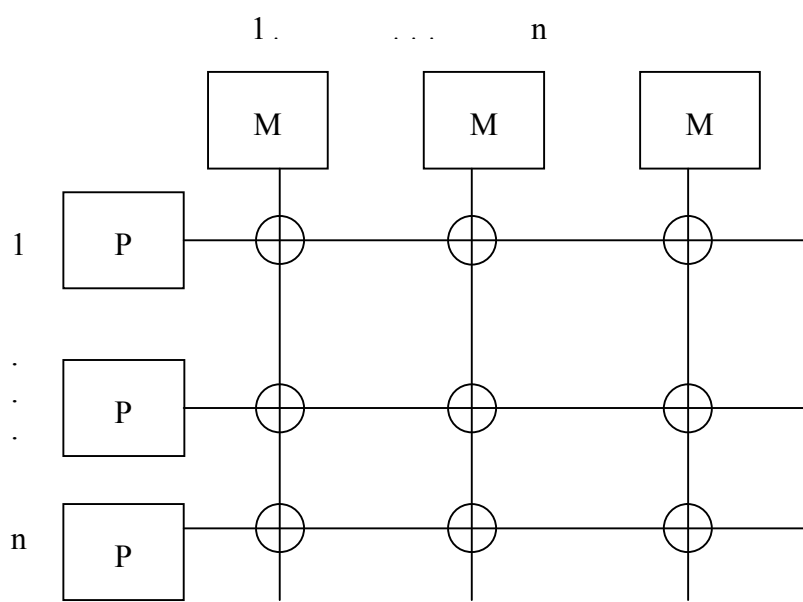


Рис. 7.8 Застосування комутованих матриць

Це потребує при кількості n модулів або процесорів n^2 комутаторів. Альтернативою є використання комутаторів з обмеженою кількістю входів (рис. 7.9).

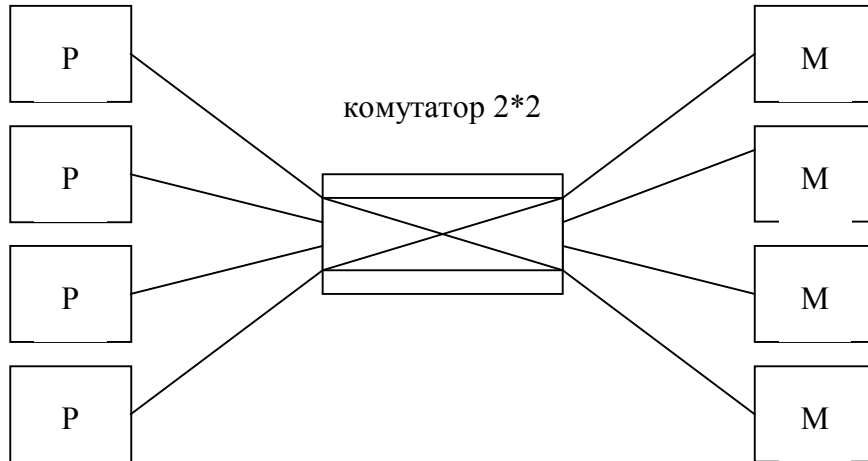


Рис. 7.9 Використання комутаторів з обмеженою кількістю входів

У гомогенних мультикомп'ютерних системах трафік зменшується за рахунок локальних взаємодій «процесор – пам'ять».

Зв'язок між комп'ютерами здійснюється за шинною архітектурою, або з використанням комутаторів, до яких належить архітектура Fast Ethernet. Масштабованість таких РС обмежена, а кількість комп'ютерів складає 25 ... 100.

Шинна архітектура таких систем зумовлює переповнення трафіку за рахунок взаємоблокувань при реалізації великої кількості звертань до шини.

У гомогенних мультикомп'ютерних системах застосовують регулярні структурні з'єднання. Найбільш поширені – квадратні решітки (рис.7.10). Гіперкуб, який є кубом розміром n . У вузлах гіперкуба розміщують процесори.

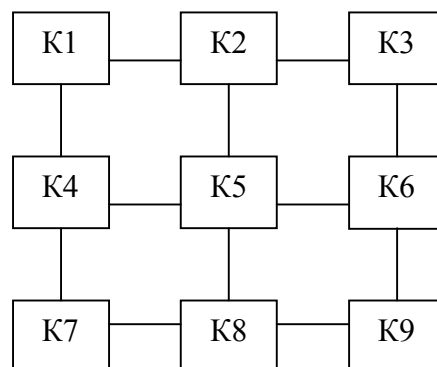


Рис. 7.10 Квадратна решітка

У мультикомп'ютерних системах зв'язок між процесорами здійснюється шляхом маршрутизації повідомлень за допомогою спеціальних швидкісних мереж комутації. Розрізняють:

- процесори з масовим паралелізмом (із кількістю процесорів до 10000);
- кластер з n робочих станцій.

У гетерогенних мультикомп'ютерних системах зв'язок між комп'ютерами не утворює регулярної структури. В них можуть застосовуватись швидкісні мережі телекомунікації або інші мережні технології. До гетерогенних систем можна віднести GRID- системи, які часто включають кластери серверів.

7.2.2 Концепції програмних рішень

Основними програмними компонентами РІС є ОС і системи проміжного рівня.

Операційні системи поділяють на розподілені та мережеві. Розподілені ОС —це сильнозв'язані ОС з одним глобальним поданням ресурсів, якими вони керують, що застосовуються у мультипроцесорних та гомогенних мультикомп'ютерних РІС. Слабозв'язані мережеві системи використовують переважно у гетерогенних мільтикомп'ютерних системах. Мережеві ОС надають локальні сервісу, доступні віддаленим клієнтам. Тому зазвичай, мережеві ОС мають спеціальні сервісу віддаленого доступу.

Системи проміжного рівня (middleware) забезпечують координовану роботу мережних ОС з можливим використанням їх, програмного інтерфейсу.

Характеристики і призначення ОС і програмних засобів проміжного рівня наведено у табл. 7.1.

Таблиця 7.1 Характеристики програмних рішень

Система	Опис	Основне призначення
Розподілена ОС	Сильнозв'язані ОС	Приховування і керування апаратними засобами
Мережна ОС	Слабозв'язані ОС локальних і глобальних гетерогенних мультикомп'ютерних мереж	Подання локальних служб віддалених клієнтів
Програмні засоби проміжного рівня	Додатковий рівень над мережевими ОС для реалізації служб загального призначення	Забезпечення прозорості розподілення

Розподілені мультипроцесорні ОС орієнтовані на підтримку високої продуктивності з одночасним забезпеченням потрібної кількості процесорів. Взаємодія додатків або їх частин відбувається через область даних сумісного використання.

Для захисту використовують примітиви синхронізації, семафори й монітори. Семафор підтримує операцію UP (збільшення) і DOWN (зменшення) цілого числа; якщо значення семафора дорівнює нулю, то процес, що викликається, блокується. Якщо значення більше за нуль, число зменшується на одиницю. Оператор збільшення перевіряє наявність заблокованих процесів. Якщо він виявляє заблоковані процеси, він розблокує один з них і продовжує роботу. Розблокований процес виконується до виклику операції зменшення. Якщо заблокованих процесів немає, значення семафору збільшується на одиницю. Коли запускається операція зменшення або збільшення, жодний процес не має доступу до семафора. Програмування з використанням семафорів спричиняє багато помилок обробки даних.

Монітор можна розглядати як модуль, який має змінні і процедури. Доступ до змінної виконується лише викликом однієї з процедур. Монітор дозволяє використовувати процедуру лише одному процесу. Застосування умовних змінних запобігає наявності двох активних процесів у моніторі.

Сучасні мультикомп'ютерні ОС складніші за мультипроцесорні. Загальну структуру мультикомп'ютерної операційної системи показано на рис. 7.11.

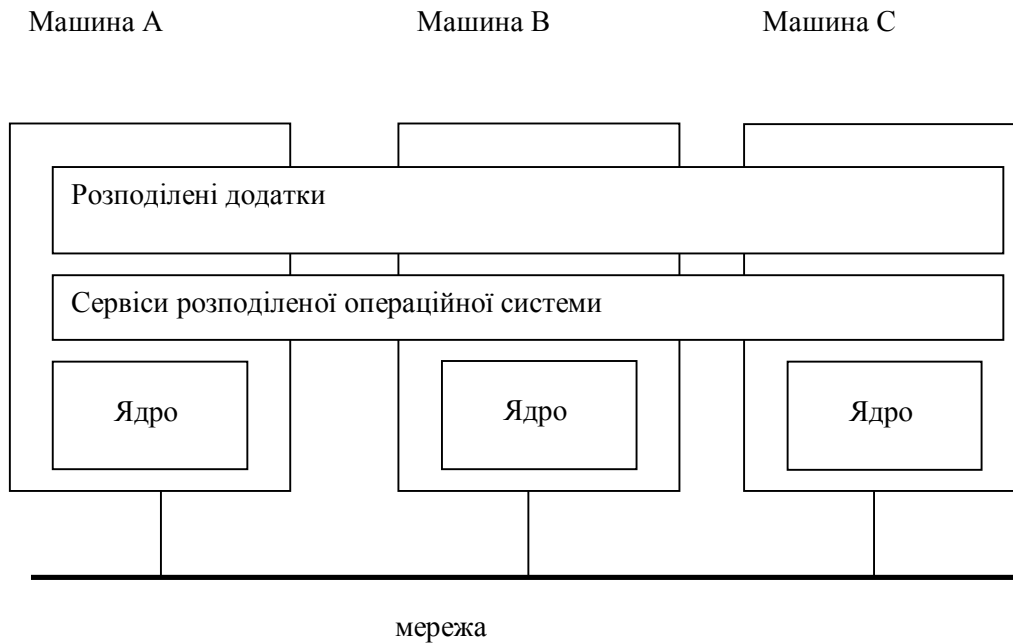


Рис. 7.11 Структура мультикомп'ютерної ОС

Кожний вузол (машина) містить ядро для керування локальними ресурсами і здійснення міжпроцесорної взаємодії між вузлами на основі повідомлень (message). Сервіси розподіленої ОС реалізують віртуальну машину для паралельного виконання задач. Прикладний рівень забезпечує реалізацію розподілених додатків. Сервіси проміжного рівня являють собою сервіси розподіленої ОС, які узгоджують роботу окремих ОС.

Ядро ОС здійснює керування локальними ресурсами: пам'яті, процесорів, принтерів, сканерів тощо. Пересилання повідомлень між машинами виконується окремим модулем приймання і передавання даних.

Сервіси розподіленої ОС забезпечують віддалене приєднання до іншої системи за допомогою спеціальних команд віддаленого доступу. При цьому робоча станція користувача переходить у режим віддаленого

термінала. Зокрема, в цьому режимі можливе копіювання файлів з машини в машину. Для створення загальних файлових систем використовують сервери; за запитом програм-користувачів (клієнтів) до сервера виконується записування або зчитування файлу. Файлові системи або їх піддерева серверів можна монтувати або імпортувати на робочі станції (наприклад, у UNIX).

Сервіси розподіленої ОС здійснюють паралельну роботу з різними задачами, призначають задачі різним процесорам, маскують збої апаратури, забезпечують прозорість схоронності та загальний обмін між процесорами.

У системах з розподіленою розділеною пам'яттю (shared memory) використовується пам'ять із сторінковою організацією. Розмір сторінки становить 4-8 КБ. У разі звертання до пам'яті відбувається перезаписування її в локальну пам'ять і система перезапускається. У разі подальшого звернення до сторінки переривання не відбувається і вона використовується безпосередньо з локальної пам'яті.

Для підвищення продуктивності розподіленої системи частина сторінок може бути заздалегідь занесена (реплікована) у локальну пам'ять (довідники, константи). Зі збільшенням розміру сторінок збільшується час їх розподілення і пересилання.

Мережні ОС гетерогенних систем мають сервіси міжмережної підтримки в кожній однопроцесорній машині, що дозволяє уникнути створення сервісу розподіленої системи..

Мережні ОС значно простіші порівняно з розподіленими ОС. Основна їх відмінність полягає у непрозорості мережних ОС, що спонукає до копіювання файлів з однієї машини на іншу. Крім цього, потрібна реєстрація користувача в системі, у межах сервера-контролера домену. Це понижує захищеність РІС від атак зловмисників.

Перевагами є гнучка масштабованість, оскільки підключення нової машини не створює проблем. Наприклад, в Інтернеті достатньо зареєструвати мережну адресу або символічне ім'я в службі DNS. Практично сервіс мережевої виконують функції служб проміжного рівня.

7.3 Програмне забезпечення проміжного рівня

Програмне забезпечення проміжного рівня дозволяє поєднати масштабованість і відкритість мережних ОС з прозорістю та простотою у використанні розподілених ОС. Таке поєднання дозволяє створювати РІС з такою загальною структурою, як показано на рис. 2.12.

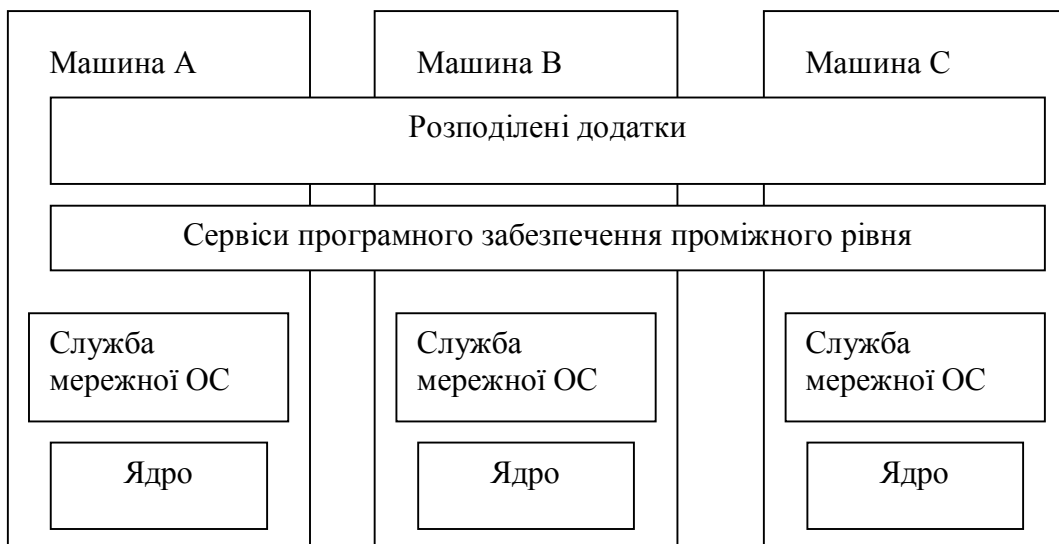


Рис. 7.12. Структура мільтикомп'ютерної системи із застосуванням проміжного рівня

Структура забезпечує поєднання можливостей окремих ОС і забезпечення прозорості, масштабованості та відкритості системи. Основне

завдання програмного забезпечення (ПЗ) проміжного рівня - приховати різноманіття базових платформ ОС від додатків.

Для розроблення ПЗ використовують деякі моделі або параметри. Основна частина ПЗ ґрунтується на моделі, яка визначає розподіленість і зв'язок. Найпростішими є моделі UNIX, коли всі ресурси системи (віддалені, локальні файли, принтери тощо) розглядаються як файли. Інша модель - це модель віддаленого виклику процедур RPC (Remote Procedure Call).

Мережний обмін приховується за рахунок виклику процедур RPC з віддалених машин. У разі виклику процедури параметри прозора передаються у віддалену машину, яка цю процедуру виконує, а результат повертається до машини виклику (у випадку oneway може не повертатись).

Відповідно до об'єктного підходу застосовують розподілені об'єкти. Кожен з них реалізує інтерфейс, який приховує внутрішні деталі процесу об'єкта від користувача. Інтерфейс реалізує методи, а процес бачить лише інтерфейс. Часто розподілені об'єкти розміщуються в одній машині, а доступ до його інтерфейсу відкривається з множини інших машин.

Щоб викликати процесором метод, інтерфейс перетворює його у повідомлення, яке надсилається об'єкту. Об'єкт виконує метод і повертає результат. Реалізація інтерфейсу перетворює повідомлення результату в значення, яке повертається і передається процесу. Так, у WWW використовується модель розподілених документів. Документ містить посилання на інші документи, тобто забезпечує їх зв'язок.

Основними сервісами проміжного рівня є:

- 1) засоби прозорого доступу до віддалених даних (файлових систем, розподілених БД, www);
- 2) сервіс віддаленого доступу, зокрема для виклику процедур і

звертання до розподілених об'єктів;

3) сервіс іменувань. Наприклад, у WWW для цього застосовують ім'я URL;

4) засоби зберігання даних (засоби схоронності persistence), зокрема, розподілені файлові системи, інтегровані бази даних або засоби зв'язку додатків з базами даних;

5) засоби розподілених транзакцій, які здійснюють множинну операцій зчитування і записування в мережах однієї атомарної операції. Атомарна операція або виконується, або не виконується (неуспішне виконання). Дані транзакції можуть розміщуватись на різних машинах;

6) засоби розподіленого захисту, які є однією з найскладніших служб проміжного рівня. Ці засоби не можуть спиратись на локальні ОС.

Програмне забезпечення проміжного рівня сучасних РІС орієнтується на кілька платформ ОС. Їх основні характеристики наведено в табл. 7.2.

Таблиця 7.2 Характеристика ОС і ПЗ проміжного рівня

Характеристика	Розподілена ОС		Мережна ОС	Розподілена система проміжного рівня
	Мультипроцесорна	Мультикомп'ютерна		
Ступінь прозорості	Дуже висока	Висока	Низька	Висока
Ідентичність ОС на всіх вузлах	Підтримується	Підтримується	Не підтримується	Не підтримується
Кількість копій ОС	1	N	N	N
Комутація на основі	Сумісного використання пам'яті	Використання повідомлень	Файлів	Залежить від моделі
Керування ресурсами	Глобальне централізоване	Глобальне розподілене	Окремо на вузлах	Окремо на вузлах
Масштабованість	Немає	Помірна	Так	Різна
Відкритість	Закрита	Закрита	Відкрита	Відкрита

Додатки зазвичай орієнтовані на певну РІС. Неповнота інтерфейсу проміжного рівня з додатком примушує створювати спеціальний інтерфейс

дodatка з ОС платформи. А це погіршує відкритість проміжного рівня РІС і системи, оскільки потребує модифікації інтерфейсу у разі зміни платформи.

7.4 Архітектури розподілених систем

7.4.1 Модель взаємодії клієнт-сервер розподілених систем

Модель клієнт-сервер набула широкого розповсюдження, особливо у зв'язку зі створенням інформаційних систем на основі централізованих і розподілених БД. Модель основана на процедурах надсилання клієнтом запита серверу і отримання від нього відповідь. Існує кілька механізмів організації взаємодії клієнт-сервер.

При застосуванні БД додатки клієнт-сервер при роботі з БД поділяються на такі рівні.

1. рівень інтерфейсу користувача.
2. рівень обробки запитів.
3. рівень даних.

Розподілені РІСППР включають засоби пошуку (пошукові системи) і засоби прийняття рішень. Розподілені пошукові системи у якості РІС виконують значну кількість функцій реалізації сервісів обробки даних за запитами РІСППР.

Наприклад для пошукових систем Internet БД містить проіндексовані Web-документи; ядро пошукової системи становить програма, яка трансформує введений користувачем рядок в один або декілька запитів до БД (рис. 7.13).

Програми рівня даних виконують видобування та первинну обробку даних. Властивістю цього рівня є схоронність. Для забезпечення цілісності даних підтримуються метадані загальні для додатків. Рівень інтерфейсу

користувача забезпечує взаємодію користувача з додатком. В реляційних БД рівень обробки даних можна відділити від рівня даних за рахунок незалежної обробки і подання даних.

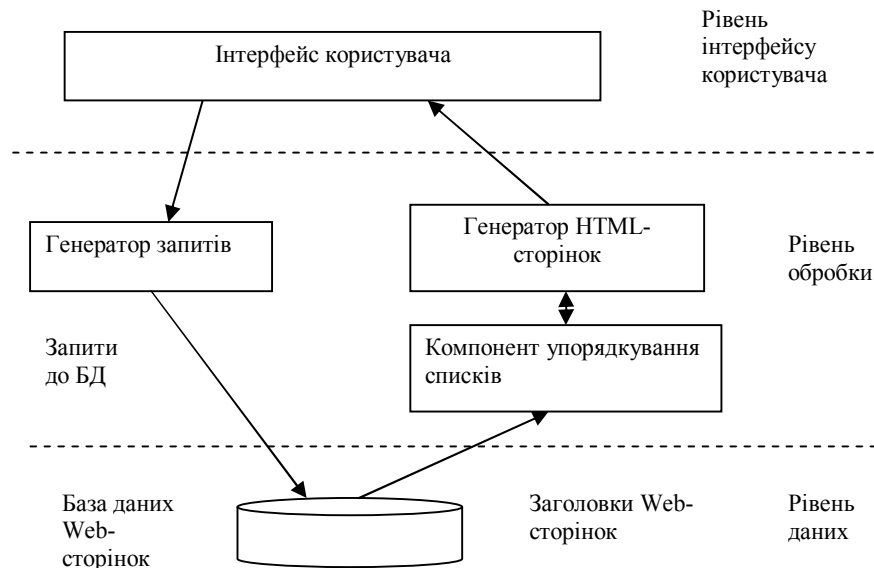


Рис. 7.13 Рівнева структура розподіленої пошукової системи

Для мультимедійних даних та інших форм подання інформації часто застосовують не відношення, а моделювання в поняттях об'єктів, що дозволяє працювати зі складними типами даних.

В об'єктно-орієнтованих БД рівень даних формується в поняттях роботи з об'єктами. При цьому підтримується організація даних у формі об'єктів і зберігається реалізація операцій над об'єктами.

7.4.2 Архітектура клієнт-сервер корпоративних інформаційних систем

Корпоративною інформаційною системою (КІС) називають систему управління бізнес-процесами підприємства, яка підтримує функціонування підрозділів забезпечуючи обробку фінансових та товарних потоків впродовж усієї технології керування даним підприємством. Єдиний інформаційний простір корпорації утворюється на основі об'єднання всіх інформаційних ресурсів корпорації з використанням інформаційно-телекомунікативної взаємодії.

На початку 90-х минулого століття створено технологію ERP (Enterprise Resource Planning). Вона поділяється на дві категорії:

1) концепція «Управління підприємством». Ця концепція забезпечує планування і розподіл матеріалів та завантаження потужностей;

2) концепція «Управління бізнесом» – для управління допоміжними процесами. Нова концепція ERP II будується на застосування Інтернет-технологій.

Лідерами ERP II є програмні продукти компаній SAP AG, Oracle, PeopleSoft J.D.Edwards.

Сучасні КІС поділяють на:

- середні (ID Edwards, Syteline(SOCAP));
- великі (R3 фірми SAP, BaanIV компанії Baan));
- інтегровані малі (Planinum SQL фірми Platinum).

Середні КІС будують на принципах багаторівневої клієнт-серверної архітектури з резервуванням критичних компонент, що забезпечує високий рівень їх безаварійності. Обчислення в таких КІС здійснюється на сервері БД, а оброблення даних на основі запитів на прикладних серверах.

Застосовують дві основні архітектури клієнт-сервер КІС (рис. 7.14):

1. Дворівнева архітектура (сервер БД та клієнтські додатки).
2. Трирівнева архітектура (включає проміжний сервер застосування).

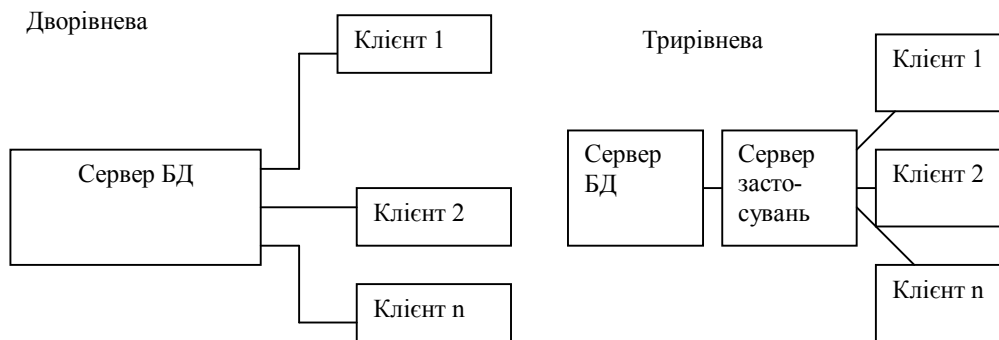


Рис. 7.14 Архітектури «клієнт-сервер» КІС

Сервер застосувань реалізує прикладні функції, оформлені у вигляді сервісів, та надає послуги всім іншим компонентам.

Для реалізації взаємодії застосовуються різні технології. Найбільш поширеними технологіями взаємодії об'єктів і програм є COM (Component Object Model – компонентна модель об'єктів) корпорації Microsoft і архітектура CORBA (Common Object Request Broker Architecture) групи OMG.

Для підтримки віддаленої взаємодії застосовуються RPC-виклики віддалених процедур і надсилання запитів об'єктами для виконання їх методів. Зокрема для спільного використання об'єктів розроблено брокер об'єктних запитів ORB консорціуму OMG, компоновник об'єктів OLE, модель COM і розподіленої COM (DCOM) компанії MS. В більш потужних КІС використовуються розподілені БД.

Для розподілених БД Дж. Дейт сформулював 12 цілей забезпечення розподіленості:

1. Локальна автономність.
2. Незалежність від центрального вузла.
3. Безперервне функціонування.
4. Незалежність від розташування.
5. Незалежність від фрагментації.
6. Незалежність від реплікації.
7. Можливість обробки розподілених запитів.
8. Управління розподіленими транзакціями.
9. Незалежність від апаратного забезпечення.
10. Незалежність від ОС.
11. Незалежність від мережі.
12. Незалежність від СУБД.

В діючих РІС ці цілі досягаються не в повній мірі. Тому при виборі архітектури КІС керуються найбільш придатними для конкретних застосувань. Основними варіантами таких архітектур є такі.

1. Незалежний варіант. Клієнтська машина реалізує інтерфейс користувача, а сервер реалізує рівень обробки і рівень даних.
2. Фізично дволанкова архітектура (ФДА) з розподілом інтерфейсу користувача між клієнтом і сервером. Сервер виконує додатки і має БД..
3. Фізична дволанкова архітектура з повною реалізацією інтерфейсу користувача на клієнті а решта (додатки і БД) на сервері.
4. Фізично збалансована архітектура з розподілом додатку між клієнтом і сервером.
5. Фізично збалансована архітектура з БД на сервері, а решти ПЗ на клієнті.

6. Фізично збалансована архітектура з розміщенням додатку і інтерфейсу на клієнті, а БД розподілені між клієнтом і сервером.
7. Фізична триланкова архітектура. Рівень обробки виносять на окремий сервер та можливо частково між клієнтом і сервером.
8. Вертикально розподілена багатоланкова архітектура з взаємодією ланок згідно з логічною організацією додатка.
9. Багатоланкова архітектура з горизонтальним розподіленням, коли клієнт або сервер може містити фізично розділені частини логічно однорідного модуля.

7.4.3 Модель розподіленої системи з проміжним рівнем

Згідно з моделлю OSI ISO повідомлення між двома машинами надсилаються на фізичному рівні у вигляді (рис. 7.15).

Заголовки рівнів						Завершення каналного рівня	
2	3	4	5	6	7	повідомлення	

Рис. 7.15 Протокольна взаємодія за еталонною моделлю OSI ISO

На каналному рівні 2 передавання даних ведеться по кадрам або фреймами. Для перевірки правильності кадру використовується бітова маска та обчислюється контрольна сума (checksum). При прийомі кадра з неспівпаданням контрольної суми виконується запит на повторне передавання саме потрібного кадра, що може призвести до його багаторазової пересилки.

На мережевому рівні 3 пакети даних маршрутизуються через низку проміжних сегментів. Алгоритми маршрутизації можуть керуватись різними критеріями оптимальності вибору шляху (маршруту). Оскільки

об'єм трафіка і кількість повідомлень у черзі на відправлення у кожній лінії маршруту різні, а визначальним при передаванні є мінімальна затримка приймання повідомлення отримувачем, то найкоротший маршрут може виявитись не оптимальним за часом затримки.

У протоколах IP мережевого рівня маршрути передавання різних пакетів є незалежними і попереднє з'єднання між відправником і отримувачем не встановлюється. У протоколах АТМ встановлюється віртуальний канал між абонентами. Набір віртуальних каналів групують у віртуальний шлях (virtual path). При зміні маршруту відбувається переключення всіх каналів віртуального шляху.

На транспортному рівні 4 реалізують всі сервіси мережних додатків, які не увійшли в мережний рівень. Протоколи фізичного, каналного, мережного та транспортного рівнів утворюють базовий стек мережних протоколів.

На транспортному рівні реалізують надійну взаємодію без втрат передавання пакетів з встановленням або без встановлення з'єднання на мережному рівні. В першому випадку забезпечується доставка пакетів у правильній послідовності за умов одночасного їх передавання. У другому випадку послідовність отримання пакетів може бути порушена, а транспортний протокол їх впорядковує.

Протокол UDP передавання дейтаграм транспортного рівня не потребує з'єднання і по суті є IP з невеликими змінами. Протокол TCP менш надійний, ніж UDP, але підтримує відкритість системи. Він створює додаткове навантаження на мережу. Протокол TCP часто застосовують в архітектурах клієнт-сервер для мереж LAN і WAN.

Для передавання даних по кадрам у реальному зв'язку розроблено протокол RTP (Real Time Transport Protocol), у якому визначено протокол для моніторингу і управління RTP-пакетами. .

В протоколі T/TCP (TCP for Transactions) зменшення часу передавання повідомлень забезпечує передаванням повідомлень із запитом, а відповідь - з розривом з'єднання. Протокол T/TCP автоматично перетворюється в TCP при відсутності його підтримки однією із сторін.

Протоколи верхнього рівня охоплюють 5, 6 та 7 рівні OSI і перший рівень стеку протоколів TCP/IP. Для реалізації проміжного рівня розподілених систем вони не є зручними.

Сеансовий рівень 5 і рівень подання 6 забезпечують управління з фіксацією машини, яка передає дані з забезпеченням засобів синхронізації. Засоби синхронізації потрібні для створення контрольних точок, довгих сеансів передавання даних і повідомлення про збої. В Інтернет та в значній кількості корпоративних систем цей рівень не використовується. Представницький рівень дозволяє визначити структуровану інформацію у відповідному форматі, що часто не використовується.

Прикладний рівень 7 моделі OSI ISO повинен містити набір стандартних протоколів і додатків (наприклад FTP та ftp), які забезпечують сервіс для користувача мережі. Але насправді цей рівень містить всі протоколи додатків, які не входять в інші рівні. Проміжний рівень розподілених систем логічно входить у прикладний рівень OSI, але містить значну кількість протоколів загального призначення.

Протоколи високого рівня можна поділити на протоколи взаємодії та протоколи подання служб проміжного рівня. Прикладом є протокол авторизації; протоколи розподіленого підтвердження (commit), які використовують у транзакціях; протоколи проміжного рівня підтримки комунікаційних служб високого рівня; сервіси групової розсилки..

З урахуванням цих особливостей модель розподіленої з урахуванням проміжного рівня має структуру: фізичної реалізації; передавання даних;

мережний рівень; транспортний рівень; проміжний рівень; рівень додатків (рис. 7.16).



Рис. 7.16 Модель розподіленої системи

7.4.4 Механізми реалізації взаємодій проміжного рівня

Існує ряд засобів забезпечення віддаленої взаємодії компонентів розподіленої системи. До найбільш поширених належить віддалений виклик процедур RPC (Remote Procedure Call). Виклик RPC було вперше запропоновано у 1976 р. Сутність RPC полягає у можливості виклику процедури, що міститься в інших машинах. Процес, запущений на машині А, викликає процедуру, розміщену у машині В. Виконання процедури в А призупиняється і виконується процедура на машині В.

Передавання інформації з процесу виклику до викликаної процедури відбувається через параметри і повертається до процесу у вигляді результату виконання процедури. Параметр може передаватись за значенням або за посиланням (рис. 7.17).

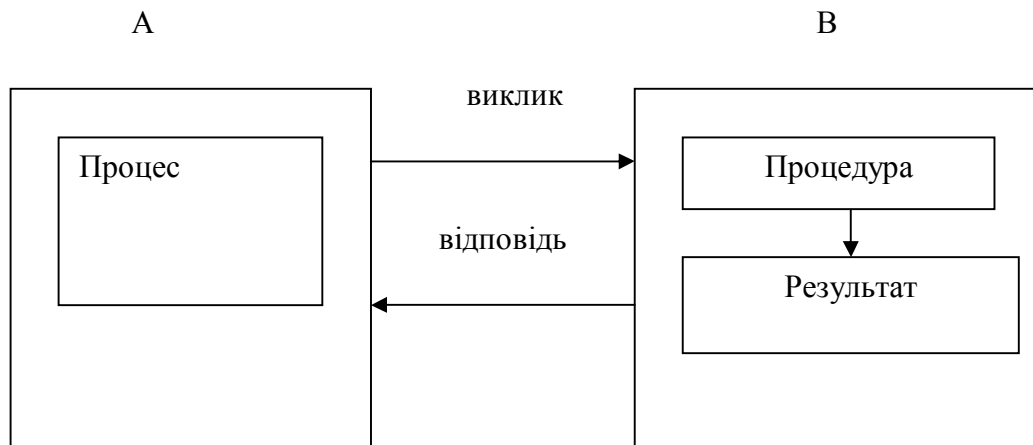


Рис. 7.17 Обмін повідомленнями між відправником і отримувачем RPC

У деяких мовах застосовують виклик через копіювання та відновлення (call-by-copy/restore). У разі такого виклику відбувається копіювання в стек змінної, як за виклику за значенням. Після завершення виклику виконується копіювання цієї змінної зі стека з вилученням значення змінної.

Прозорість команд RPC (зокрема, read) забезпечується за допомогою використання клієнтської заглушки, яка пакує параметри у повідомлення і надсилає процедурою send на сервер. Далі клієнтська заглушка викликає процедуру receive і блокується до отримання відповіді (рис. 7.18).

Повідомлення, отримане на сервері, операційною системою на сервері передається серверній заглушці (server stub), яка перетворює запит у виклик локальних процедур сервера.

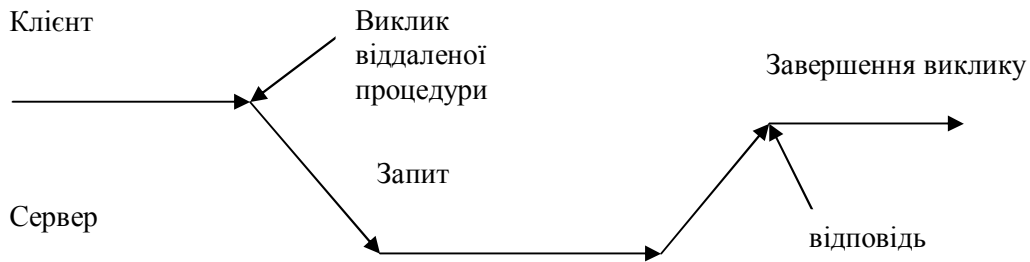


Рис. 7.18 Схема взаємодія клієнта і сервера у RPC

Для отримання повідомлення заглушка запускає процедуру receive і блокується, очікуючи повідомлення. Після завершення оброблення виклику результат спаковується і процедурою send передається клієнту. Далі викликається процедура receive.

Операційна система клієнта визначає, що отримане повідомлення належить залушці, копіює це повідомлення в буфер очікування, а клієнтський процес розблоковується. Клієнтська залушка розпаковує повідомлення, копіює у пам'ять програми, яка її викликала, і завершує роботу, передаючи в неї код повернення. Ця програма отримує результат у буфері, тобто без визначення місця його походження.

Таким чином, клієнтська залушка перетворює виклик процедури у локальний виклик процедури сервера (рис. 7.19).

Існують деякі особливості передавання в RPC. Передавання за значенням спакування параметрів (маршалінг параметрів – parameter marshaling) передбачає у повідомленні наявності імені віддаленої процедури, за яким на серверній залушці, можливо із застосуванням команд типу switch, визначають саму процедуру.

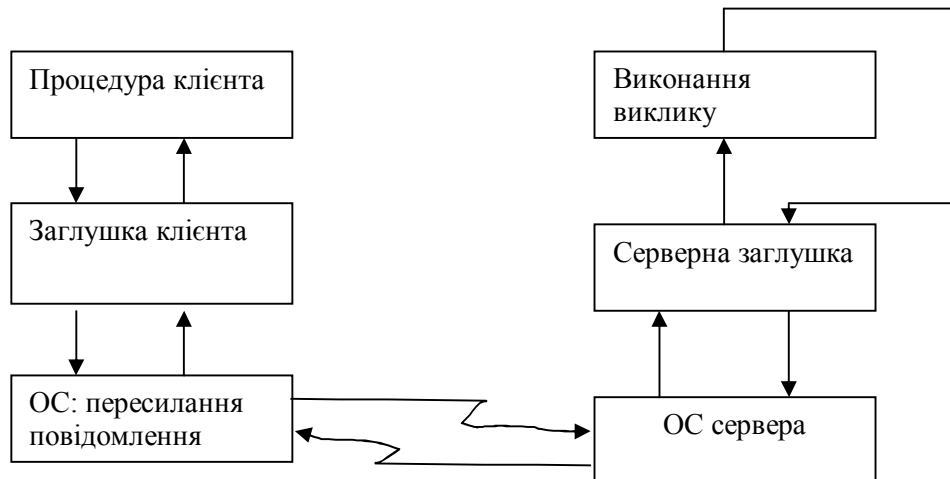


Рис. 7.19 Послідовність використання заглушок клієнта і сервера

Для організації локальної взаємодії процесів на одній машині застосовують засоби між процесорної взаємодії IPC (Inter Process Communication), які надають базові ОС. Так, в UNIX це засоби керування розподіленою пам'яттю (shared memory), каналами, чергами повідомлень сумісного використання.

Засоби IPC переважно ефективніші за мережні засоби RPC на окремій машині. Компромісом для деяких ОС (наприклад, Spring, Solaris) є застосування механізму, еквівалентного RPC, який має назву doors (двері або входи) і подібний механізм спрощеного виклику RPC (lightweight RPC).

Під час реєстрації входу сервером вхід отримує ідентифікатор, який можна використовувати як символічне ім'я входу. У Solaris виклик такого ідентифікатора з файловою назвою виконується простим запитом `fattach`. Запит `door_call` є системним і повертається через системний виклик `door_return`.

Виклик процедур для віддаленої машини виконується за допомогою механізмів розширеного RPC. Стандартним є призупинення роботи клієнта на час виклику і отримання відповіді від сервера. Удосконаленим є асинхронний виклик RPC (asynchronous RPC), за яким клієнт продовжує роботу після виклику після отримання від сервера підтвердження про початок оброблення запиту. Удосконаленням є відкладений синхронний виклик RPC (deferred synchronous RPC), за яким сервер отримує від клієнта перелік потрібних для виконання запиту хостів (host), і підтверджує їх отримання. Існує також варіант одностороннього RPC, за яким клієнт не очікує відповіді від сервера.

Механізм RPC застосовують як основу проміжного рівня. Найбільш відомими його варіантами реалізації є Sun RPC та DCE (Distributed Computing Environment) RPC. Версії DCE RPC розроблено для UNIX, VMS і продуктів Microsoft.

Основні сервіси DCE RPC такі.

1. Сервіси, що є частиною DCE:

- усесвітня сервіс розподілених файлів (distributed file service);
- сервіс каталогів (directory service) для відслідковування місцезнаходження ресурсу системи;
- сервіс захисту (security service);
- сервіс розподіленого часу (distributed time service).

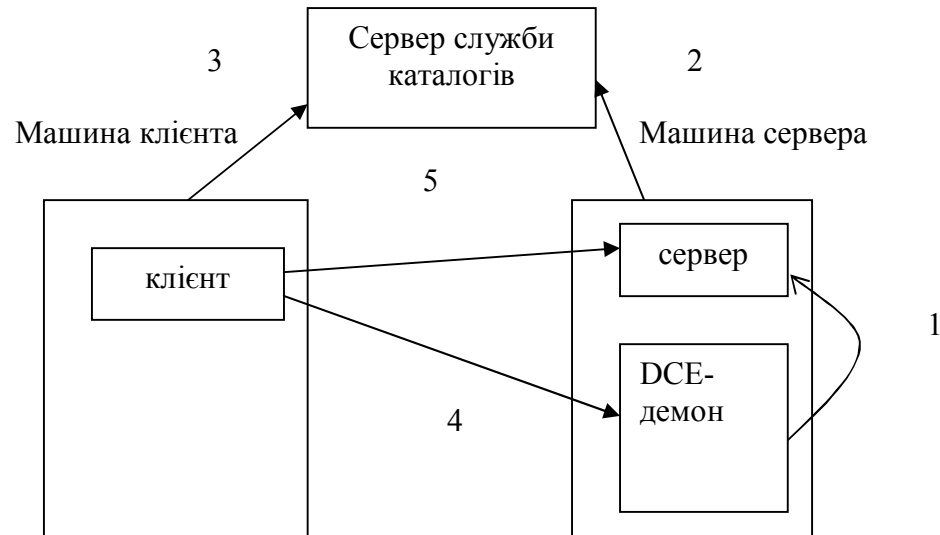
2. Сервіси на основі додатків.

Системи RPC виконують такі процеси:

- 1) автоматичне визначення потрібного сервера і встановлення зв'язку між клієнтом і сервером через прив'язування (binding);
- 2) керування транспортуванням повідомлень;
- 3) автоматичне відслідковування перетворень даних між клієнтом і сервером;

- 4) приховування реалізації взаємодії від користувача;
- 5) підтримання різноманіття мережних протоколів і подання даних.

Прив'язування клієнта до сервера виконується за схемою рис. 7.20.



1 – реєстрація кінцевої точки; 2- реєстрація сервісу; 3-пошук сервера сервісу каталогів; 4- запит кінцевої точки; 5 – виконання виклику RPC

Рис. 7.20 Схема прив'язування клієнта до сервера DCE

Клієнт до сервера прив'язується через реєстрацію клієнта на машині каталогів з подальшим використанням відповідного сервера і кінцевої точки або порту сервера. При цьому на машині сервера здійснюється фіксація сервера і відповідної кінцевої точки (порту) у спеціальній БД за допомогою DCE-демона. На підставі вихідних даних сервера, кінцевої точки (порту) і визначеного протоколу відбувається взаємодія клієнта із сервером.

Принципи RPC можуть застосовуватись для об'єктно-орієнтованих технологій на основі розподілених віддалених об'єктів за схемою заміщувача клієнта (рис. 7.21).

Звертання клієнта через замісника або представника (проху) називають методом віддаленого звернення до методу RMI (Remote Method Invocation). Інтерфейс об'єктів RMI пишуть мовою IDL, Java та іншими засобами, за умови, що інтерфейси об'єктів під час розроблення клієнтського додатка відомі, а у разі зміни клієнтського інтерфейсу виконується перекомпіляція.

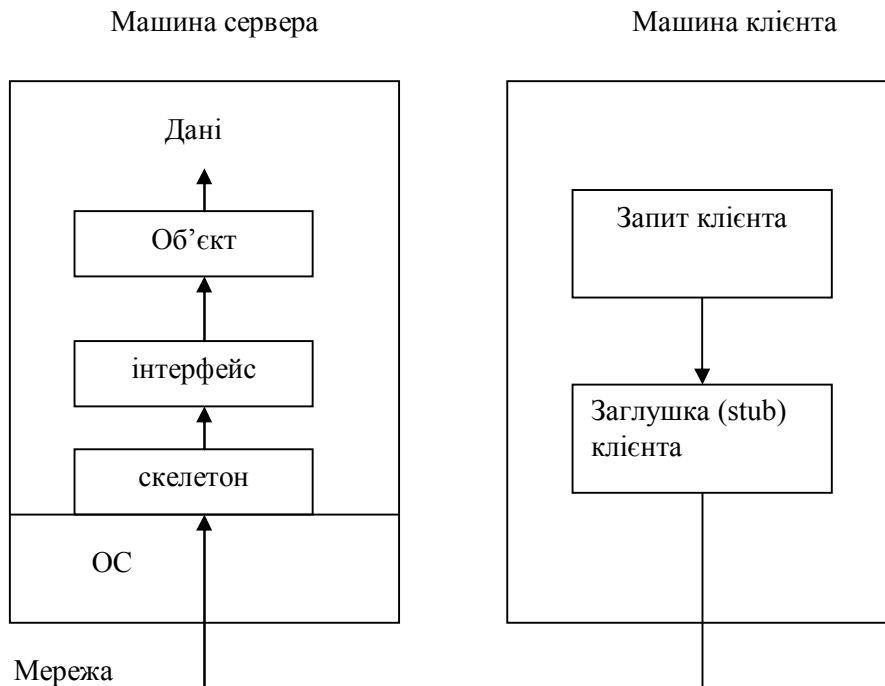


Рис. 7.21 Схема організації RPC для розподілених об'єктів

Передавання параметрів у RMI для розподілених об'єктів розрізняється залежно від того, чи є об'єкт локальним або віддаленим. Для локальних об'єктів у клієнтській машині, де є посилання, об'єкт копіюється повністю і в процесі копіювання передається клієнту.

7.5 Зв'язок на основі повідомлень

7.5.1 Застосування сокетів

Зв'язок на основі повідомлень дозволяє уникнути блокування клієнта у процесі здійснення операції, що характерно для RPC і RMI.

Сокет (Socket) є кінцевою точкою комунікації, в яку додаток може записувати дані про пересилання базовою мережею і з якої може зчитувати поточні дані, що надійшли. Сокети утворюють абстракцію над реальною кінцевою точкою мережі, яка працює з локальною ОС за деяким транспортним протоколом (табл. 7.3).

На початку сервер зазвичай виконує послідовність `socket`, `bind`, `listen`, `accept`. У примітиві `socket` для кінцевої точки ОС резервують ресурси для розміщення і відправлення повідомлень згідно з деяким протоколом. `bind` прив'язує локальні адреси до створеного сокету, наприклад, IP-адресу своєї машини з портом сокету. Примітив `listen` резервує буфер через ОС для максимальної кількості з'єднань (черги на встановлення з'єднання).

Таблиця 7.3 Примітиви сокетів для комунікації повідомлень TCP/IP

Примітив	Призначення
<code>socket</code>	Створити нову кінцеву точку комунікації
<code>bind</code>	Призначити сокету локальну адресу
<code>listen</code>	Позначити готовність до встановлення з'єднання
<code>accept</code>	Блокувати викликану сторону до прибуття запиту про з'єднання
<code>connect</code>	Зробити спробу встановлення зв'язку
<code>send</code>	Надіслати дані
<code>receive</code>	Отримати дані

Примітив `accept` блокує викликаний процес до надходження запиту на з'єднання, який ініціює ОС створити новий сокет з властивостями

базового і передачу його процесу. Це дозволяє розгалужити процес, який буде підтримувати зв'язок через нове з'єднання.

У клієнта після створення кінцевої точки виконується примітив `connect`, що вимагає від створеного процесу адресу чи порт транспортного рівня, за якими буде відправлено запит на з'єднання. До з'єднання клієнт блокується (у випадку блокованого введення-виведення), а після з'єднання починається обмін інформацією за допомогою примітивів `write` `read`.

Наявні примітиви взаємодії `send`, `receive`, сокети і орієнтація на застосування протоколів загального призначення типу TCP/IP обмежують можливості їх застосування у спеціальних протоколах взаємодії мереж у високопродуктивних мультикомп'ютерних системах. Тому такі системи зазвичай мають власні комунікаційні бібліотеки, які часто не сумісні між собою.

7.5.2 Стандарт MPI

Стандарт MPI (Message Passing Interface – інтерфейс передавання даних) орієнтовано на паралельні додатки. Стандарт MPI не передбачає автоматичного відновлення системи за наявності фатальних збоїв (наприклад, аварій процесів або ділянок мережі) та синхронізацію відправника й отримувача під час передавання повідомлень мережею. Разом з цим примітиви MPI орієнтовані на багатоваріантну реалізацію інших типів взаємодій у високопродуктивних паралельних додатках. Інтерфейс містить понад сотні функцій для створення додатків.

MPI – найпоширеніший стандарт передавання інформації у паралельному програмуванні. Існують його реалізації для великої кількості комп'ютерних платформ. Головним засобом комунікації між процесами в MPI є передавання повідомлень один одному. У стандарті MPI описано

інтерфейс передавання повідомлень, котрий має підтримуватися як на платформі, так і в додатках користувача. Є велика кількість безоплатних і комерційних реалізацій MPI.

7.5.3 Проміжний рівень на основі черг повідомлень

Сучасний асинхронний зв'язок на основі повідомлень ґрунтується на повідомленнях служб проміжного рівня, відомих під назвою «система черг повідомлень» (Message-queuing System), або орієнтований на повідомленнях проміжний рівень MOM (Message Oriented Middleware). На відміну від сокетів Берклі і MPI система MOM призначена для підтримання обміну повідомленнями протягом хвилин, а не секунд чи мілісекунд.

Модель MOM базується на слабозв'язаній взаємодії додатків через повідомлення, які вони розміщують у черги. Повідомлення передають ланкою комунікаційних серверів, які зазвичай мають прямий зв'язок безвідносно до активності отримувача повідомлень. Черга повідомлень доступна одному або одночасно кільком додаткам. Отримувач через деякий час одержить потрібне повідомлення зі своєї черги.

Слабозв'язана взаємодія між відправником і отримувачем передбачає чотири варіанти режиму їх взаємодії залежно від їх активності.

Адресація даних повідомлень здійснюється через надання унікального в межах системи, імені черги, що значно спрощує інтерфейс, який надається користувачам. Базовий інтерфейс містить такі примітиви:

Put – додати повідомлення в чергу;

Get – взяти повідомлення з черги, тобто призупинити свою роботу до появи повідомлення в черзі і витягнути перше з цих повідомлень;

Poll – перевірити наявність повідомлень у черзі;

Notify – вставити дескриптор функції, яку буде викликано в разі розміщення повідомлення у відповідну чергу.

Більшість систем черг повідомлень підтримує процес вставки дескриптора функції зворотного виклику `CallBack function`, який викликається автоматично при потраплянні повідомлення у чергу. Зворотні виклики можна застосовувати для автоматичного запуску процесу, для вилучення повідомлень з черги, якщо жоден з процесів не запущено. Ці функції реалізуються за допомогою демона на боці отримувача, який перевіряє черги на наявність вхідних повідомлень.

Загальна архітектура системи черг містить кілька типів менеджерів черг (`queue managers`):

1) менеджери, які взаємодіють з додатками для відправлення і отримання повідомлень;

2) спеціалізовані менеджери маршрутизації і ретрансляції, які перенаправляють отримані повідомлення іншим менеджерам черг.

Система надає такі черги:

1) вихідні локальні черги (`source queue`) відправника, які перебувають на тій же машині або деякій машині в межах локальної мережі. Повідомлення містить опис черги призначення (`destination queue`), у яку воно має бути переміщено;

2) черги призначення отримувача повідомлень;

3) бази даних імен-черг (`queue names`), яке визначає, подібно DNS, відображення черг на мережні адреси.

Система черг повідомлень надає черги відправникам і отримувачам повідомлень і забезпечує переміщення повідомлення з черги відправника до черги отримувача повідомлення. Адресація на рівні черг і мережному рівні пов'язані між собою (рис. 7.22).

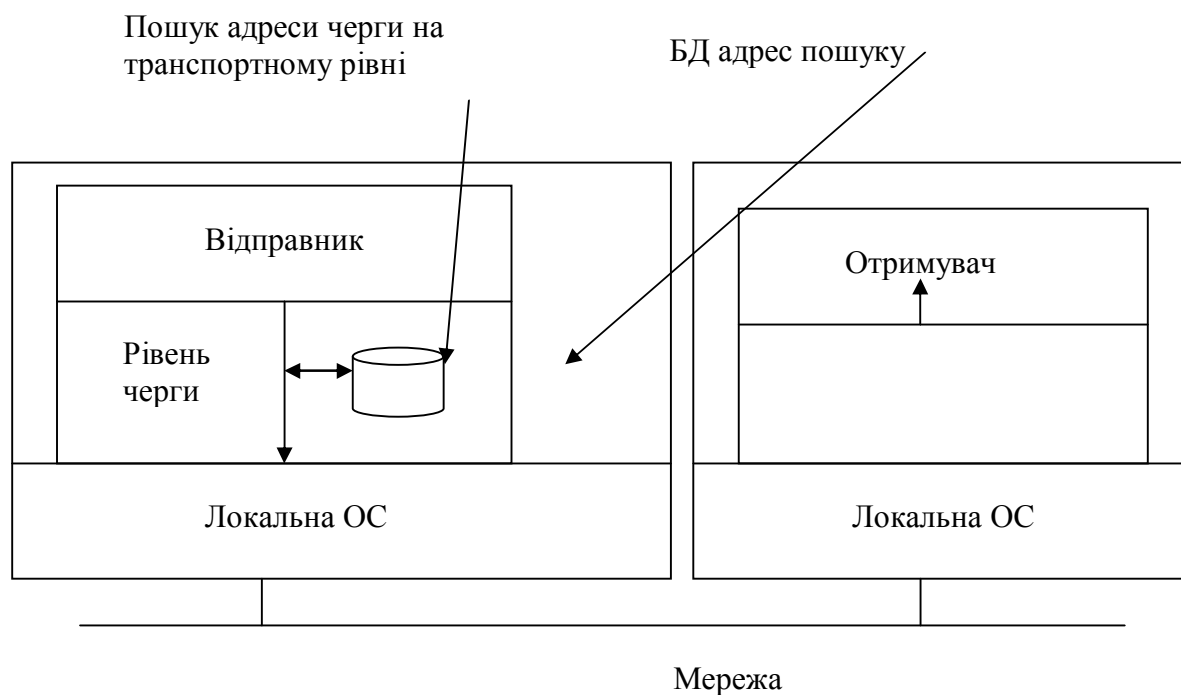


Рис. 7.22 Взаємодія на рівні черг і мережевому рівні

7.5.4 Зв'язок на основі потоків даних

Зв'язок описує взаємодію процесів для даних, чутливих до часу передавання (зокрема, мультимедійних даних). Основне завдання – це підтримання неперервних середовищ, яке має певний спосіб подання інформації (GIF, JPEG, мультимедійні дані тощо). Для передавання таких даних використовують потоки даних (date streams, streams), які являють собою послідовність елементів даних.

Основні режими їх передавання такі:

- синхронний, коли для кожного елемента потоку визначено максимальну затримку наскрізного передавання;
- асинхронний з дотриманням лише послідовності передавання елементів даних, а не їх часових характеристик;

– ізохронний, коли час передавання елементів обмежено максимальним і мінімальним часом затримання передавання, що характерно для розподілених систем мультимедіа.

Розрізняють прості й комплексні (складені) потоки, які містять кілька взаємопов'язаних потоків, взаємодія між якими також може залежати від часу. У разі групового розсилання потоків кільком приймачам даних постає проблема забезпечення якості передавання за різних вимог цих приймачів до потоків. Оскільки ланки від джерел до отримувача інформації мають різні характеристики, то доцільно визначити певну якість передавання (наприклад, часові діаграми обміну).

Опис якості обслуговування QoS (Quality of Service) для різних служб передачі даних є певною проблемою. Вимоги до якості визначаються специфікацією передачі, згідно з якими розробляються протоколи передавання. Наприклад, протокол RSVP (Resource ReSerVation Protocol) є протоколом резервування ресурсів і керування транспортного рівня для мережних маршрутизаторів. Протокол використовується хостами для підтримання потоків даних від додатків, що потребують заданої якості обслуговування від мережі для окремих потоків даних, а в мережних маршрутизаторах для доставки керувальних запитів QoS усім вузлам.

7.6 Реалізації розподілених систем

7.6.1 Промисловий стандарт CORBA

Узагальнена архітектура брокера об'єктних запитів CORBA (Common Object Request Broker Architecture) є специфікацією побудови розподілених систем, технологічний стандарт написання розподілених додатків, який розроблено консорціумом OMG (Object Management

Group). Перші версії специфікації з'явилися на початку 90-х років. Архітектуру CORBA показано на рис. 7.23.

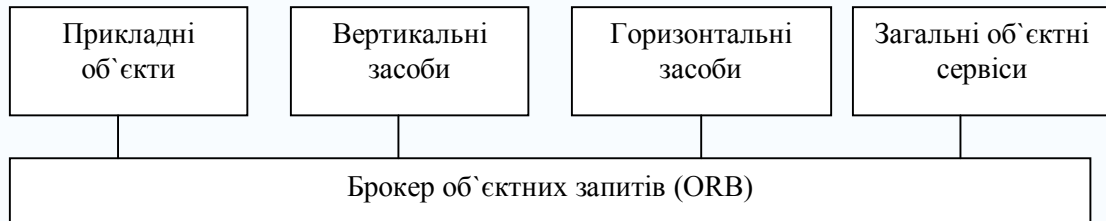


Рис. 7.23 Архітектура CORBA

Завдання CORBA — виконати інтеграцію ізольованих систем, дати змогу програмам, написаним різними мовами, тим, що працюють на різних вузлах мережі, взаємодіяти так само просто, якби вони були в адресному просторі одного процесу.

Є кілька реалізацій CORBA. Широко використовуються SOM і DSOM фірми IBM. CORBA також є складовою частиною в платформі ONE (Open Network Environment) фірми Netscape.

Брокер ORB підтримує віддалений зв'язок між об'єктами та клієнтами і приховує проблеми, пов'язані з розподіленням і різномірністю частин системи. Визначаючи конкретну архітектуру ORB, не обов'язково реалізувати його як один компонент. Але кожна реалізація має виконувати три категорії операцій:

- 1) однакові для всіх реалізацій ORB;
- 2) специфічні для конкретного об'єктного типу;
- 3) специфічні для окремих видів реалізацій об'єктів.

Брокер ORB може підтримувати різні види реалізацій, а адаптери об'єктів можуть забезпечувати потрібні набори сервісів для клієнта і реалізацій. Ядро ORB забезпечує основні механізми для маніпулювання з

об'єктами і виконання запитів. Специфікація CORBA призначена для підтримання різних механізмів реалізації об'єктів, тому структура ядра не визначається. Замість цього задається набір інтерфейсних функцій, які мають бути в кожній реалізації ORB, і тим самим маскувати відмінності між різними реалізаціями ORB.

Горизонтальні засоби (horizontal facilities) незалежні від прикладного рівня і являють собою високорівневі сервіси загального призначення для обслуговування користувацького інтерфейсу, керування інформацією, системою і задачами для визначення систем робочих потоків. Вертикальні засоби (vertical facilities) – це високорівневі сервіси для конкретних предметних галузей (електронна комерція, банківська справа, виробництво тощо).

Об'єкти і сервіси описуються в CORBA мовою опису інтерфейсів IDL (Interface Definition Language). Мову розроблено OMG для опису інтерфейсів розподілених об'єктів – набору методів і їх параметрів – типів змінних-аргументів в рамках узагальненої архітектури CORBA. Специфікація інтерфейсу, задана в IDL, відображається мовами програмування: C, C++, Java, SmallTalk, ADA, Cobol.

Брокер ORB надає :

1) операції маршалінгу і демаршалінгу посилань на об'єкти під час обміну між процесами і операції для порівняння посилань;

2) операції початкового пошуку доступних для процесу служб. Зазвичай – ORB надає дані для отримання початкового посилання на об'єкт, який реалізує конкретну службу CORBA.

Загальну схему взаємодії клієнта і сервера показано на рис. 7.24.

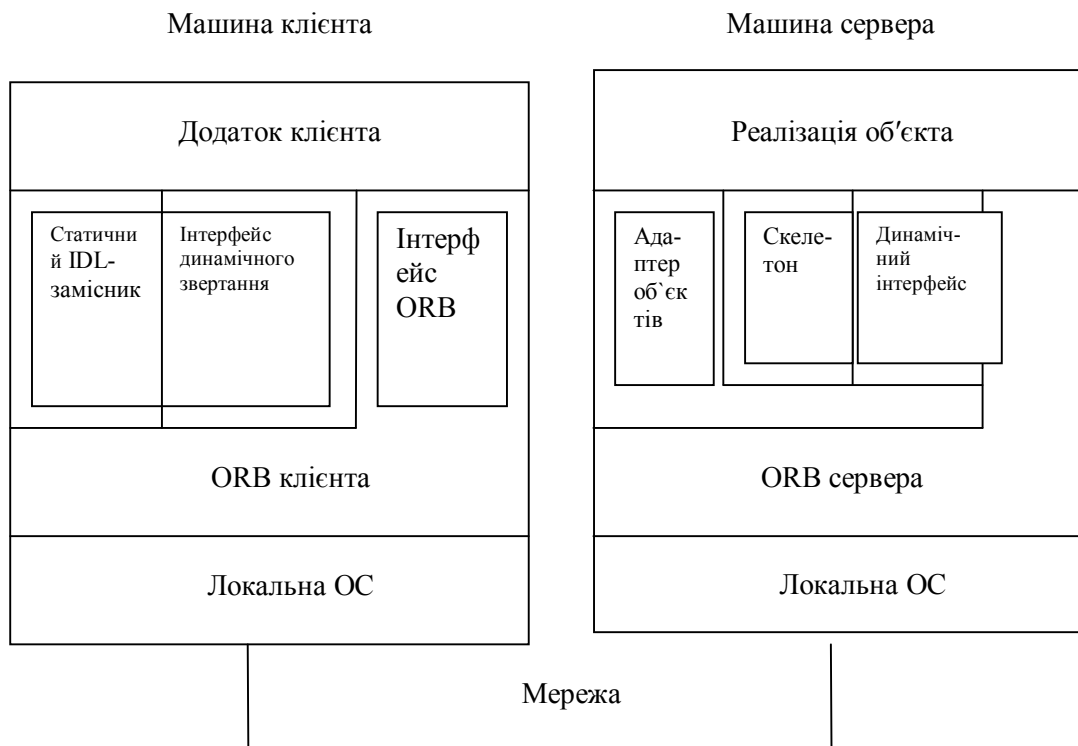


Рис. 7.24 Структура клієнта і сервера на основі брокера ORB

Адаптер об'єктів відповідає за передавання вхідних запитів необхідному об'єкту, а демаршалінг запиту виконує скелетон сервера (stub-замісник) або сам об'єкт.

Усі визначені інтерфейси зберігаються в сховищі інтерфейсів (interface repository) і викликаються за ідентифікатором зберігання (repository identifier), призначеним компілятором IDL. Сховище реалізації CORBA (implementation repository) надає потрібні дані для реалізації і активізації об'єкта для адаптера об'єктів в адресному просторі сервера (наприклад, таблицю серверів і портів).

Проходження запиту від клієнта до реалізації об'єкта показано на рис. 7.25. Інтерфейс – це набір методів, а об'єкт визначає, які сервіси клієнт може запросити. Система об'єктів забезпечує клієнта набором сервісів.

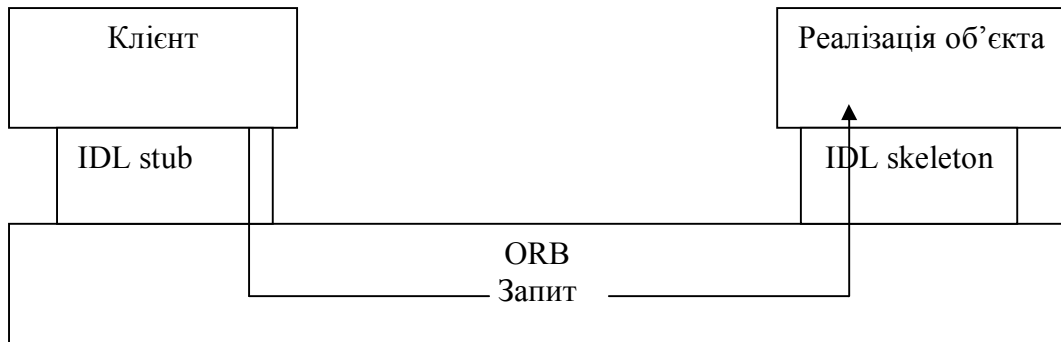


Рис. 7.25 Послідовність проходження запиту від клієнта до сервера

Схему взаємодії мереж через зв'язок двох брокерів від ORB1 до ORB2 показано на рис. 7.26.

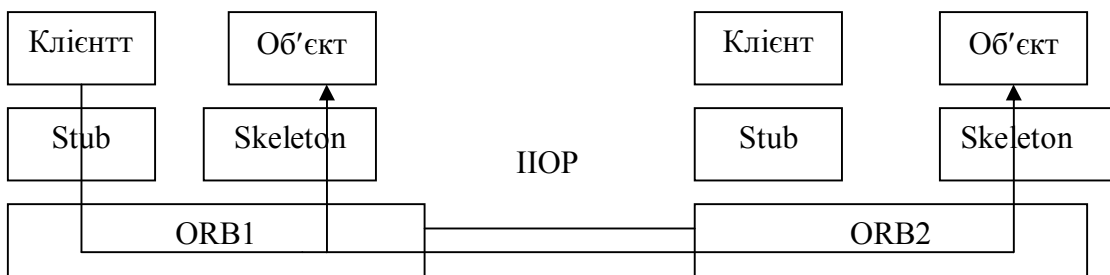


Рис. 7.26 Взаємодія мереж на основі ORB і протоколу IIOP

Компілятор IDL компілює код зв'язку між клієнтським і серверним ORB. Якщо статично визначений інтерфейс не доступний клієнтові, то CORBA надає інтерфейс динамічного звертання DII (Dynamic Invocation Interface), який дозволяє створювати запити під час виконання додатків.

Типи служб CORBA подібні до служб ОС і включають запитів, транзакцій, подій, сповіщень, сервіси обміну, схоронності, захисту тощо. Сервіси написано мовою IDL.

Зв'язок здійснюється звертанням клієнта до методу об'єкта через відсилання запиту відповідному серверу. Якщо запит асинхронний без

відповіді, використовується односторонній запит (one-way request) з використанням сервісу доставлення з максимальним зусиллям (best effort delivery) без гарантії якості доставки. У разі відкладеного синхронного запиту (deferred synchronous request) ресурс повертається клієнтові асинхронно після отримання результату сервера. Існує також одноразовий синхронний запит.

Сервіс подій (event service) асоціюється з посиланням на одиничний елемент даних. Зазвичай сервіс передає подію каналами подій (event channel) без гарантії схоронності. Для фільтрації подій користувача застосовують окремі канали подій за допомогою сервісу сповіщення (notification service).

Повідомлення між непрацездатним клієнтом або сервером зберігається базовою системою.

Сумісність версій CORBA від різних виробників забезпечується стандартизованим протоколом GIOP (General Inter-Orb Protocol) взаємодії між брокерами та універсальними посиланнями на об'єкти. Реалізацію GIOP над рівнем TCP називають Інтернет – протоколом обміну між брокерами ІІОР (Internet Inter-Orb Protocol). Протоколи GIOP, ІІОР має вісім типів повідомлень (табл. 7.4).

Для модифікації запиту на шляху клієнт–сервер і зворотного повідомлення сервера використовують кілька типів фрагментів кодів перехоплювачів (interceptors). Так, перехоплювач рівня запитів (Request Level Interceptor) модифікує записи від замісника, а після проходження ORB перехоплювач повідомлення (Message Level Interceptor) модифікує повідомлення (наприклад, ділить за фрагментами), що надходить через ОС на сервер за схемою, показаною на рис. 7.27.

Таблиця 7.4 Повідомлення GIOP

№ з/п	Тип повідомлення	Джерело	Опис
1	Request	Клієнт	Повідомлення після маршалінга з посиланнями і параметрами об'єкта
2	Reply	Сервер	Повернення значень і параметрів
3	Local request	Клієнт	Запит до сховища реалізацій про розміщення об'єкту
4	Local reply	Сервер	Відповідь на Local request
5	Cancel request	Клієнт	Відміна дій раніше відправлених повідомлень для сервера
6	Close connection	Клієнт/сервер	Припинення зв'язку
7	Message error	Клієнт/сервер	Заголовок повідомлення помилки
8	Fragment	Клієнт/сервер	Частина повідомлення

Перехоплювач рівня повідомлень сервера розташовано між ORB й адаптером. Він збирає фрагменти повідомлення. Перехоплювачі розпізнають лише ORB.

Адаптери об'єкта (WRAPPERS) реалізують правила активізації групи об'єктів. Вони викликають методи об'єкта і відповідають за створення цілісного образу об'єкта; адаптують для клієнта програми як об'єкти. Так, наприклад, переносний адаптер об'єктів POA (Portable Object Adapter), без прив'язки до конкретного ORB дає змогу сприймати серверний код об'єктом для клієнта.

Для використання додатків-агентів у CORBA застосовують модель сумісної роботи агентів різних типів, яка передбачає наявність стандартного інтерфейсу для роботи з агентами. Система агентів AS (agent system) визначає платформу створення, виклику і перенесення на іншу машину агентів та припинення їх роботи. Система AS має профіль, який визначає вигляд агента і складається з множини процесів, кожний з яких підтримує одного або кількох агентів.

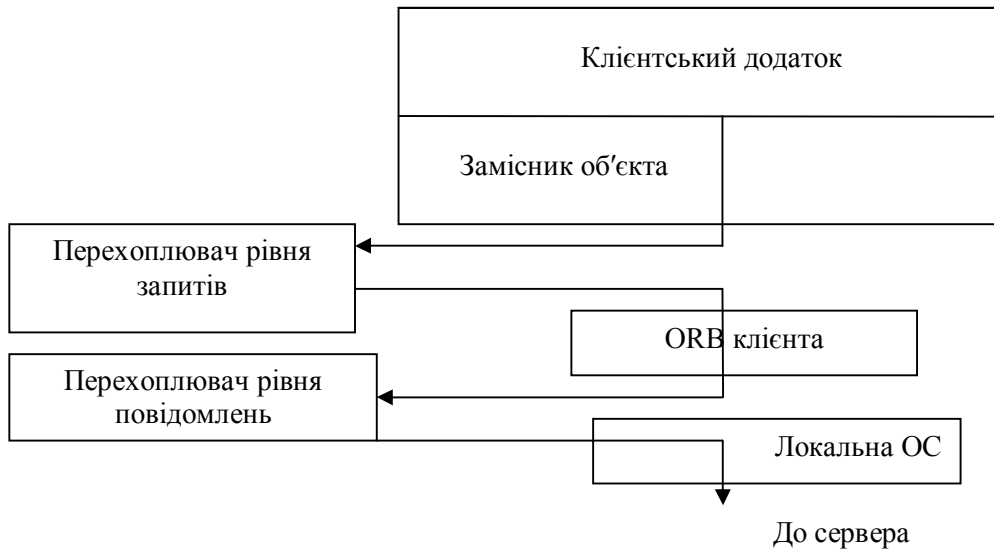


Рис. 7.27 Використання перехоплювачів у CORBA

Систему агентів групують за регіонами. Загальну модель агентів і регіонів показано на рис. 7.28.

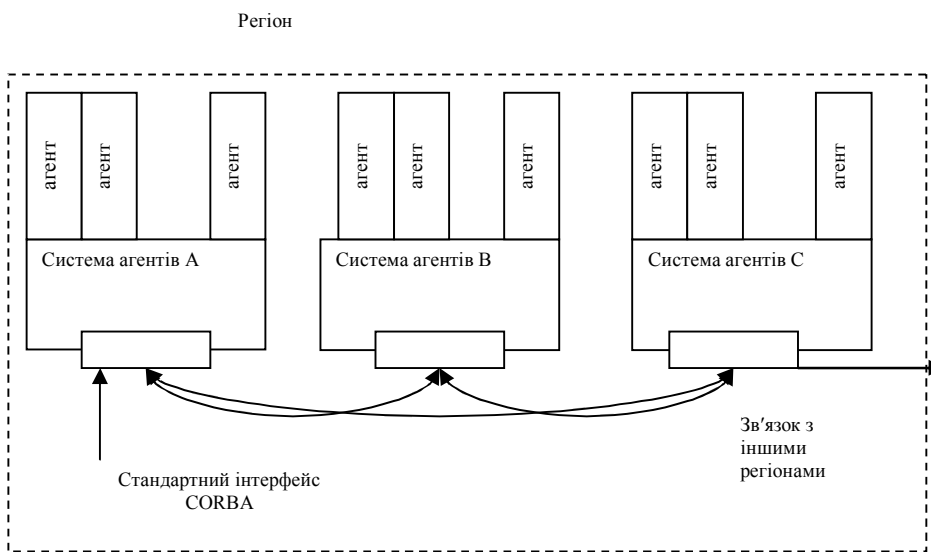


Рис. 7.28 Моделі агентів CORBA

Агент розглядають як сутність, складену з набору класів або файлу з текстом програми. Реалізація агента через ім'я агента передається в систему агентів з подальшою можливістю створювати цей агент і передавати його на інші машини.

Система AS реалізує стандартні операції, зокрема створення, припинення роботи агента, ведення списку агентів і місць, де вони розташовані, здійснює призупинення і повторний запуск агента. Регіон має шукача (Finder), який визначає місцеположення агента, реєструє та знімає агента з реєстрації.

Сервіс іменування (Naming Service) об'єктів підтримує відображення імен на ідентифікатори об'єктів через імена нижнього рівня (посилання на об'єкти і символічні імена) і через властивості об'єктів за допомогою сервісу властивостей (property service), яка асоціює об'єкт з парою <атрибут, значення >. Для цих двох служб сервіс обміну (trading service) дозволяє об'єктам регламентувати свої можливості, а клієнтам – шукати потрібні сервіси.

Брокер ORB має власне подання посилань на об'єкти у формі міжпроцесорного посилання на об'єкт IOR (Interoperable Object Reference), зрозуміле іншому брокеру ORB, і містить потрібну інформацію для ідентифікації об'єкта.

Посилання IOR містить, зокрема, типовий профіль (tagged profile), який включає усю інформацію, потрібну для звернення до об'єкта. Організацію IOR за специфікацією IIOP показано на рис. 7.29.

Зокрема, якщо сервером об'єктів підтримується кілька протоколів, то інформація про кожен з них міститься в окремому теговому профілю. Хост може, наприклад, задаватись DNS або IP-адресою. Порт сервера очікує повідомлення. Ключ об'єкта дозволяє серверу виокремити потрібні запити

(наприклад, створений POA ідентифікатор об'єкта) та ідентифікувати POA.



Рис. 7.29 Структура IOR CORBA

Розрізняють безпосередні посилання в IOR на об'єкт – пряму прив'язку (direct binding) і непрямую прив'язку (indirect binding), за якою спочатку пересилаються у сховище реалізацій, яке є процесом, ідентифікованим в IOR об'єкта. Сховище реалізацій переглядає свої таблиці реєстру і шукає потрібний сервер об'єктів.

Сервіс паралельного доступу реалізовується центральним менеджером блокувань, який підтримує кілька ступенів деталізації блокувань, що особливо важливо для БД (зокрема, блокування записів, таблиць).

CORBA не підтримує узагальненого кешування та реплікації, але окремі їх реалізації використовують для забезпечення відмовостійкості. Тому в додатках передбачають спеціальні засоби, зокрема для їх реалізації з використанням перехоплювачів.

У системах DSS CORBA часто застосовують для реалізації окремих функцій по узгодженню програмних продуктів API DSS з системами аналізу реальної ситуації. Наприклад, у системі ATACKS (The Advanced Tactical Architecture for Combat Knowledge System) забезпечують інтеграцію зовнішньої підсистема специфікації подій DEVS (the Discrete Event System Specification) (рис. 7.30, 7.31) .

Взаємодія процесів оброблення даних відбувається з використанням механізмів IPC.

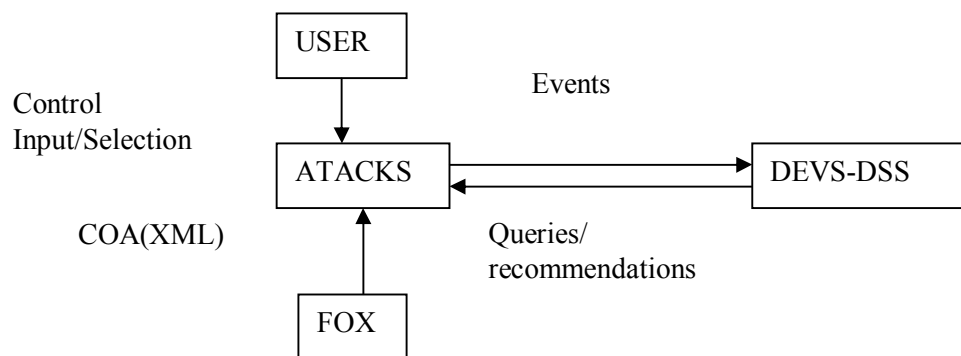


Рис. 7.30 Організація інтерфейсу користувача у прийнятті рішень

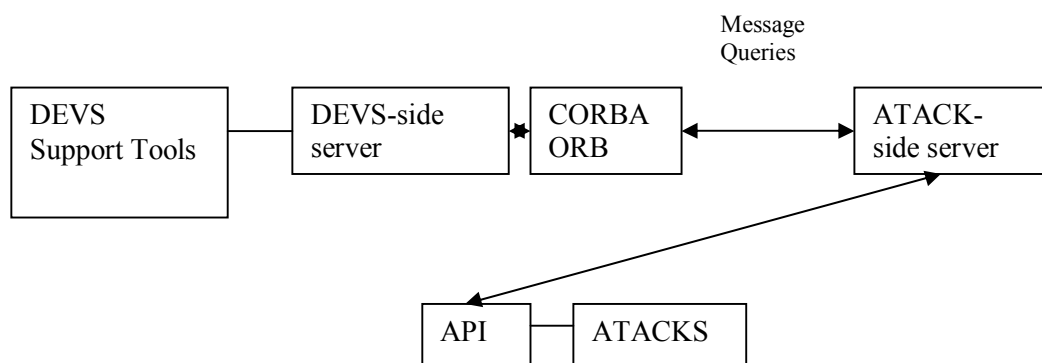


Рис. 7.31 Взаємодія DEVS з ATACKS

7.6.2 Система GLOBE

Систему середовища глобальних об'єктів GLOBE (Global Object-Based Environment) розроблено в університеті Vrije (Амстердам). Систему створено для підтримання дуже великої кількості користувачів і об'єктів Інтернет зі зберіганням прозорості розподілу.

Основою системи є метод перегляду об'єктів за кількома процесами на різних машинах. Кожний об'єкт визначає розподілення своїх станів за репліками з можливим переміщенням станів, правилами захисту тощо. Система Globe не підтримує моделі доступу до віддалених об'єктів.

У загальному випадку об'єкт є розподілений розділений об'єкт DSO (distributed shared object). Процес, пов'язаний з DSO, має локальну реалізацію інтерфейсів об'єкта, яку називають локальним поданням або локальним об'єктом LO (Local Object). Локальні об'єкти розподілені по різних машинах і доступні (розділені) для різних процесів. Взаємодія локальних об'єктів відбувається через інтерфейс локального об'єкта. Загальну структуру DSO показано на рис. 7.32.

Об'єкти LO мають загальний інтерфейс SOInf, подібний IUnknown DCOM, який за методом getInf за отриманими вихідними даними ідентифікатора інтерфейсу повертає вказівник на цей інтерфейс. Вказівник забезпечує можливість доступу до цього інтерфейсу іншим процесам. Локальний об'єкт має відповідний об'єкт класу, що дозволяє створювати нові локальні об'єкти.

Об'єкти LO поділяються на примітивні PLO (primitive local object) та композитні CLO (composite local object) об'єкти. Об'єкти PLO реалізують бінарні інтерфейси, які складаються переважно з таблиць вказівників на функції і описуються мовою IDL. Об'єкти CLO мають кілька складових

локальних об'єктів (підоб'єктів): підоб'єктів керування, реплікацій зв'язку, семантики тощо, які взаємодіють між собою. Загальну структуру локального об'єкта показано на рис. 7.33.

Підоб'єкт керування CLO-CTS (Control Subobject) є зв'язуючою ланкою між процесом і локальним об'єктом. Він передає реалізацію локального інтерфейсу підоб'єкта семантики і стану локального об'єкта процесу. Підоб'єкт семантики CLO-SS (Semantic Subobject) забезпечує створення локальних реалізацій інтерфейсу DSO для заданого процесу. Підоб'єкт реплікації CLO-RS (replication Subobject) реалізує політику розподілення DSO у вигляді локальних об'єктів за допомогою реплікації цього об'єкта для заданого процесу. Підоб'єкт зв'язку (Communication Subobject, CLO-CS) дозволяє здійснювати пересилати повідомлення у процесі взаємодії локальних об'єктів між собою.

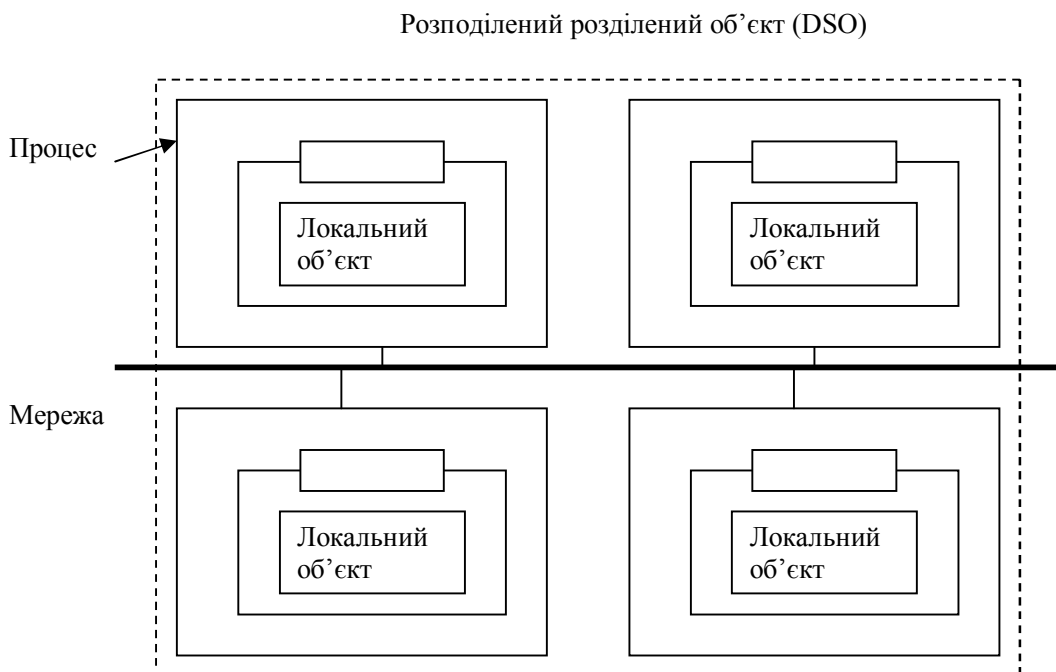


Рис. 7.32 Організація об'єкта Globe

Прив'язування процесу до об'єкта виконується за п'ять етапів:

1. Клієнтом сервісу іменувань надсилається ім'я сервісу іменувань, яка запитується клієнтом. Сервіс іменувань побудовано подібно до DNS.
2. Сервіс іменувань створює глобальне ім'я і дескриптор об'єкта.
3. Сервіс локалізації повертає набір контактних адрес, за якими може реалізуватись процес під час роботи з об'єктом.
4. У сховище класів DSO завантажується локальний об'єкт локальної реалізації DSO, яка відповідає запиту процесу.
5. Ініціалізується локальний об'єкт і викликаються дії, передбачені вихідними даними обробки даних під час реалізації процесу з використанням DSO.



Рис. 7.33 Структура локального об'єкта Globe

Сервіси GLOBE поділяють на власне сервіси GLOBE і сервіси, які реалізуються об'єктами DSO. На відміну від DCOM і CORBA значну кількість служб перекладено на об'єкт.

До власних сервісів належать:

- 1) сервіси зовнішніх зв'язків, за якими кожний процес реалізує свій власний маршалінг даних, що пересилаються;
- 2) сервіс життєвого циклу, за яким виокремлення об'єкта класу об'єкта DSO реалізується у самому об'єкті;
- 3) сервіс іменувань, що дозволяє створювати глобальні імена;
- 4) сервіс схоронності, яка призначена для зберігання класів DSO у сховищі;
- 5) сервіс захисту, що реалізується підоб'єктом захисту з використанням служб локального захисту;
- 6) сервіс реплікації, яка підтримує реалізації DSO у вигляді локальних об'єктів;
- 7) сервіс відмовостійкості.

У GLOBE передбачено застосування тільки синхронного зв'язку під час звернення до методів. Після завершення звернення до DSO через локальний інтерфейс локального об'єкта діяльність DSO припиняється, тобто DSO виконує роль пасивного об'єкта.

У процесі взаємодії з іншими локальними об'єктами об'єкти в DSO можуть мати різний склад підоб'єктів, зокрема, може не бути підоб'єкт CLO-SS, який визначає стан і посилання на локальну реалізацію інтерфейсу при взаємодії з процесом. У цьому разі процес забезпечує взаємодію з іншими локальними об'єктами DSO за CLO-CTS, CLO-RS, CLO-SS. Це дозволяє виконувати розподілення DSO під час взаємодії процесів. Наприклад, процес А не має CLO-SS (рис. 7.34).

Після завершення DSO стає пасивним. Реалізація клієнтських і серверних процесів у GLOBE дещо розрізняються.

Для реалізації клієнтських процесів застосовують реалізацію локального об'єкта DSO цього об'єкта і CLO-CTS. Для реалізації

серверних процесів застосовують CLO-SS, що забезпечує взаємодію локальних об'єктів з використанням ресурсів об'єкта DS0.

У GLOBE існують серверні процеси для підтримання об'єктів DSO. До основних методів, використовуваних цим сервером, належить метод BIND прив'язування до вказаного об'єкта DSO. Для прив'язування серверу надаються дескриптор Object Handle і критерій вибору контактної адреси, якщо таких кілька.

Метод Create LR створює локальний об'єкт нового об'єкта DSO. Контактна адреса створеного об'єкта стає доступною службі локалізації, що уможлиблює прив'язування процесів до об'єкта.

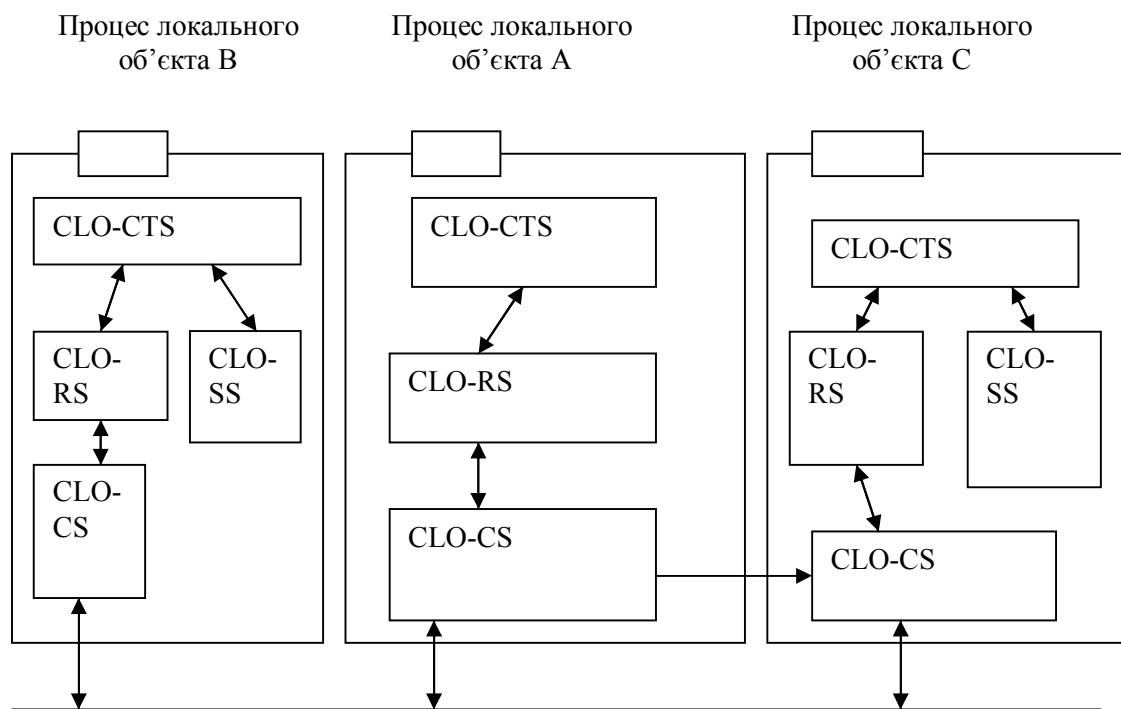


Рис. 7.34 Використання активної реплікації у разі звернення до об'єкта Globe

Прив'язування може зумовити створення постійного локального об'єкта PLO (Persistent local object). Такі об'єкти зберігаються в постійному сховищі і дозволяють відновити стан DSO.

У GLOBE немає механізму відкату. Це спрощує реалізацію взаємодії процесів і дає змогу уникнути створення аварійних непередбачуваних ситуацій. Деякою компенсацією цього механізму є використання постійних локальних об'єктів і взаємодія локальних об'єктів між собою.

До основних методів належать такі:

Add Binding – прив'язування до об'єкту за умови, що таке прив'язування вже відбувалося;

Remove LR – вилучення локального об'єкта зі складу даного DSO;

UnBinding DSO – зняття прив'язування об'єкта, тобто вилучення всіх локальних об'єктів DSO;

List DSO – виведення переліку всіх локальних об'єктів DSO;

Stat LR – отримання стану поточного локального об'єкту.

Іменування DSO у GLOBE здійснюється за допомогою глобального ідентифікатора OID довжиною 256 біт. Його використовують для аналізу посилань різних процесів на DSO. Вважається, що процеси посилаються на один об'єкт DSO, якщо ці DSO мають однаковий OID. Для отримання контактної адреси в службі локалізації використовують дескриптор об'єкта object handler, який підтримує два типи адрес: упаковану адресу (stacked address) і адресу екземплярів (instance address). Упакована адреса є багаторівневим набом протоколів, кожен рівень якого описується трьома полями: ідентифікатором протоколу, адресою протоколу, дескриптором реалізації, який містить посилання на сховище класів (за допомогою адресації URL).

Ідентифікатор протоколу відповідає відомому протоколу (наприклад, протоколу UDP, TCP, IP). Адреса протоколу містить номер порту протоколу TCP або мережні адреси протоколу IPV4.

Адреса екземпляра містить дескриптор реалізації локального об'єкта і рядок ініціалізації, який передається об'єкту після завантаження реалізації і завантаження об'єкта.

Сервіс іменувань GLOBE створюється на основі DNS. Вихідне ім'я, яке задається, спочатку перетворюється на правильне ім'я за допомогою спеціального перетворювача імен GLOBE. Ім'я подано уніфікованим ідентифікатором ресурсу URI (Uniform Resource Identifier).

Синхронізація в GLOBE реалізується у вигляді частини підоб'єкта CLO-RS, оскільки спеціального механізму не передбачено. Кожен DSO має свою власну стратегію реплікації.

Інтерфейси CLO-RS стандартизовано. Стан і операції над станами DSO реалізуються в CLO-SS, розподілених по кількох процесах. Тип несуперечності копій реплікації CLO-SS підтримується CLO-RS з урахуванням методу читання і модифікації (Read And Modify).

У разі звернення процесу до методу DSO підоб'єкт CLO-CS інформує викликом методу Start інтерфейсу CLO-RS. Дані стосуються можливості змін у CLO-SS під час звернення до DSO. За цією інформацією CLO-RS визначає наступні дії CLO-CTS.

У результаті виклику Start CLO-CTS у CLO-RS може бути передано запит за методом send. Інший варіант дії CLO-CTS – виконання запиту у разі звернення до CLO-SS.

Метод send отримує запит, виконує маршалінг запиту і відправлення повідомлення іншим реплікам DSO. Підоб'єкт реплік CLO-RS збирає репліки локальних об'єктів і передає їх процесу.

Для захисту DSO передбачено спеціальний підоб'єкт захисту CLO-SCS (Security Subobject), який взаємодіє з іншими підоб'єктами (рис.7.35).

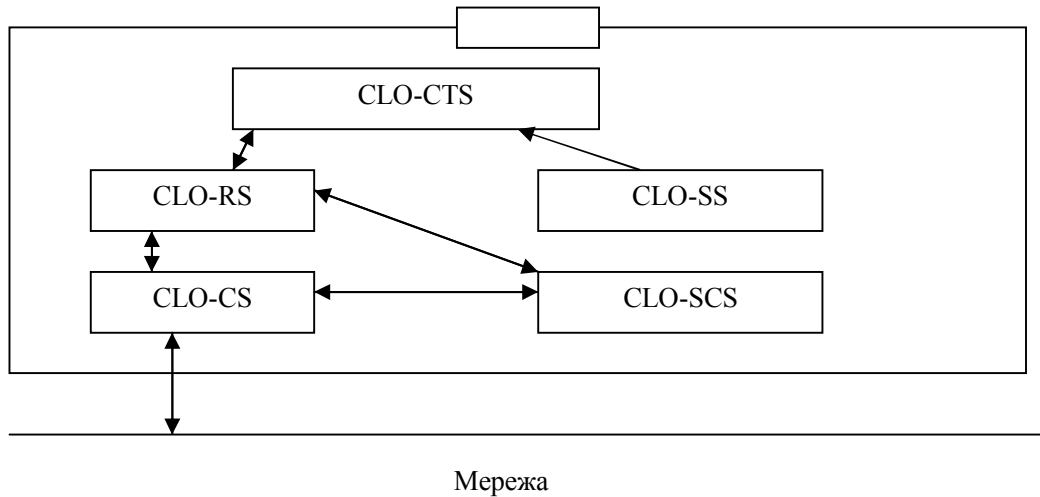


Рис. 7.35 Схема застосування підоб'єкту захисту

Для автентифікації звернень віддалених процесів CLO-CS передає дані про запити в CLO-SCS, який після завершення автентифікації обмінюється даними автентифікації з підоб'єктом CLO-RS з виконанням подальшої взаємодії і контролю доступу до ресурсів. Контроль доступу реалізовано для кожного методу окремо, тобто для кожного методу уможлиблюється звернення до нього процесу із запитом.

Для перевірки відповідності параметрів запиту певним умовам підоб'єкт CLO-SCS може додатково звертатися до CLO-CTS. Наприклад, такою умовою може бути перевищення певного значення обчислених даних.

Підоб'єкт CLO-SCS може звертатись до інших локальних служб захисту, наприклад, для отримання сеансового ключа через виділений DSO підтримання (principal object), який подає користувача конкретного

DSO в заданому захищеному домені. Цей об'єкт відповідає, наприклад, за взаємодією з довіреними третіми особами, зокрема сервісами автентифікації і сертифікації.

7.6.3 Система CODA

Файлова система CODA розроблена в університеті Карнегі-Мелона CMU (Carnegie Mellon University) і інтегрована у ОС UNIX, LINUX. В системі забезпечена масштабованість і висока доступність користувачів за рахунок удосконаленої схеми кешування, яка дозволяє клієнту продовжувати роботу після відключення від сервера, і прозорість відмов системи. Прототипом системи є система AFS на 10000 робочих станцій і має ту ж саму AFS Virtue організацію.

Вузли AFS поділяються на дві групи: робочі станції клієнтів Virtue і виділені файлові сервери Vice з централізованим адмініструванням.

Внутрішня архітектура вузла Virtue має такий вигляд (рис. 7.36).

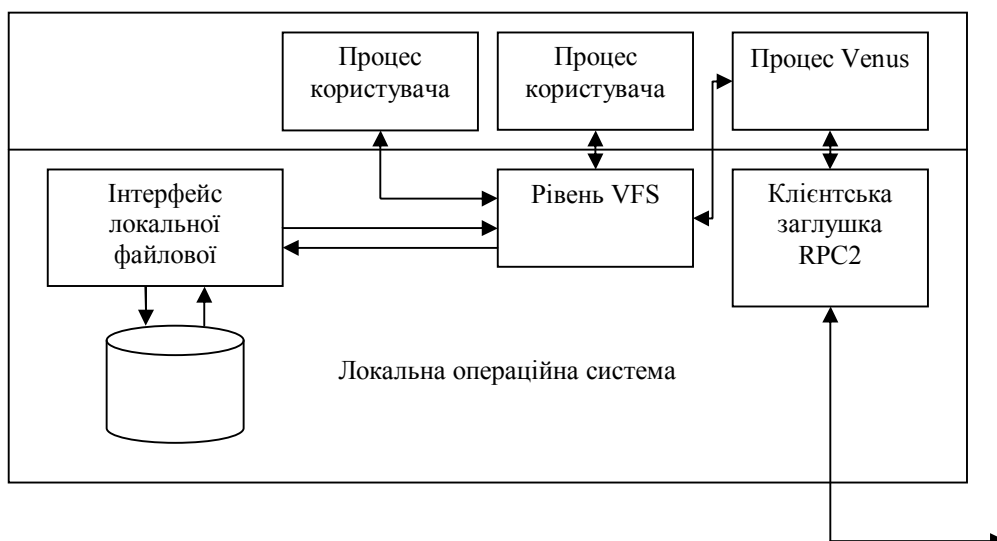


Рис. 7.36 Архітектура вузла Virtue

Прикладний процес Venus за надання доступу до файлів серверів Vice, а також за продовження виконання операцій клієнтами, коли сервери тимчасово недоступні. Рівень VFS перехоплює виклики з клієнтських додатків і передає виклику у локальну файлову систему або процесу Venus.

Взаємодію з сервером здійснює спеціальна програма VENUS. Ця програма також підтверджує роботу вузла VIRTUE в умовах відсутності зв'язку з серверами VICE. Venus спілкується з Vice через RPC користувацького рівня, надбудованим над UDP. На стороні сервера є три різних процеси: файловий сервер Vice, сервер автентифікації і процеси відновлення для збереження непротивічності метаінформації на кожному з серверів Vice.

На користувацькому рівні підтримується більшість операцій NFS, але на відміну від NFS в CODA є глобальний розподілений простір імен, який підтримується серверами Vice. Клієнти отримують доступ до цього простору через спеціальний вкладений каталог у своєму локальному адресному просторі.

Система RPC2 є складнішою за звичайний RPC. Запити RPC2 реалізують надійні виклики RPC на базі ненадійного протоколу UDP. При кожному виклику RPC код клієнта RPC2 запускає потік виконання, який передає запит на сервер і блокується. Сервер регулярно відповідає повідомленнями про його обробку. Якщо таких повідомлень не надходить потік повертає повідомлення про помилку у додаток.

Система RPC2 підтримує побічні ефекти (side effects), під якими розуміють взаємодію клієнта і сервера з використанням специфічного протоколу додатка. Сервер і клієнт встановлюють постійний потік даних в

режимі ізохронної передачі з гарантованою затримкою передавання даних між мінімальним і максимальним значенням.

Зокрема, він здійснює групову розсилку повідомлень про наявні кешовані файли в локальній файловій системі VIRTUE. RPC2 надбудована над ненадійним протоколом UDP і виконує всі операції ля забезпечення надійного зв'язку. RPC2 запускає потік виконання, що передає запит на сервер, і блокується. Сервер регулярно повідомляє про обробку отриманого запиту. При відсутності зворотного повідомлення робиться висновок про відключення серверу. Тоді вузол VIRTUE переходить в автономній режим роботи з керованими файлами отриманими з серверу.

Сервер і клієнт встановлюють постійний потік даних в режимі ізохронної передачі даних з гарантованою затримкою в межах мінімального і максимального значення. Це дозволяє передавати в реальному часі відео дані з використанням спеціального інтерфейсу побічних процесів, які реалізуються додатком.

RPC2, на відміну від звичайних RPC підтримує групове розсилання . Сервер відслідковує наявність копій файла у клієнта. Модифікація файла призводить до повідомлення через RPC клієнтів. На це затрачається достатньо багато часу. Найкращим є об'явлення копію неправильною. Це повідомляється всім клієнтам. Сервер через потрібну паузу визначає, які клієнти не в змозі прийняти RPC і визначає їх як такими, що відмовили.

При паралельних викликах RPC застосовується система MultiRPC, яка є частиною RPC2 і прозорою для отримувача. Отримувач сприймає виклик MultiRPC як звичайний RPC тз подібною семантикою.

Процеси в CODA чітко поділяються на клієнтські і серверні. Клієнтські реалізуються за допомогою програми VENUS, а серверні за допомогою Vice. Ці типи процесів організовано у вигляді паралельних потоків в адресному просторі імен користувача. При організації

безперервної операції за умови блокуючих запитів вводу/виводу застосовують окремий потік виконання, який перебирає на себе операції виконання багаторівневих асинхронних операцій базової ОС. Такий потік виконання ефективно емулює синхронне вводу/виводу без блокування процесу в цілому.

Іменування в CODA подібно до UNIX: Файли групуються у модулі (томи –Volumes), яким відповідають певні області на диску. За звичай том – це колекція файлів, яка належить одному користувачу. Томи можуть монтуватись, монтажною точкою служить кореневий вузол іншого тому.

Повне ім'я файлу часто містить кілька монтажних точок. Для прозорості сервер VICE повертає в VENUS інформацію про монтування. Це дозволяє VENUS автоматично монтувати томи у просторі імен клієнта з використанням рівня VFS інтерфейсу локальної файлової системи.

VENUS підтримує структуру загального простору імен. Клієнтам гарантується однаковість файлів сумісного використання, хоча розв'язання імен файлів ґрунтується на локальній реалізації простору імен.

Файлам відповідають ідентифікатори файлів. Ідентифікатори є унікальними і пов'язані із зберіганням файлів точно в одному томі. Том може бути репліковано на кількох серверах.

Розрізняють логічні і фізичні томи. Логічні томи передбачають можливість реплікації фізичного тому і має ідентифікатор реплікованого тому RVID (replicated Volume Identifier). З RVID асоціює кілька реплік. Фізичний том має ідентифікатор VID (Volume Identifier) для уточнення конкретної репліки, незалежно від її місцезнаходження.

Кожен файл має 96-бітний ідентифікатор файла, який складається з 2-х частин: RVID і дескриптору файла. Перша частина складається з 32 біт являє собою ідентифікатор логічного тому. При пошуку файлу клієнт передає RVID в базу даних реплікації томів (Volume Replication Database),

яка повертає список VID (Volume ID). Він відповідає ідентифікатору RVID і може мати кілька значень: VID1, VID2.(рис. 7.37).

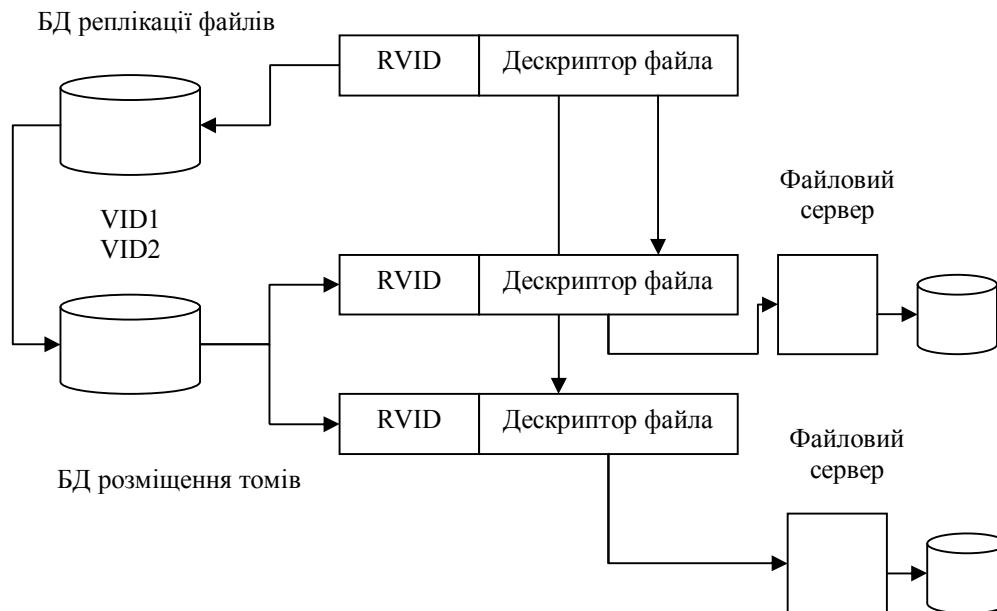


Рис. 7.37 Доступ до томів CODA

Отримавши VID, клієнт шукає сервер на якому знаходиться робоча репліка логічного тому. Пошук здійснюється надсиланням VID в базу даних розміщення томів. VLDB (Volume Location Database), яка повертає поточне місцезнаходження конкретного фізичного тому.

Друга частина складається з 64 біт і є дескриптором, який унікально ідентифікує файл в межах тому. Він є ідентифікатором вузла індексу віртуальної файлової системи VFS і має назву віртуального вузла (VNODE) (аналогічно i-node в UNIX).

В транзакціях застосовують поняття розділу (partition), під яким розуміють частину мережі, ізольованої від решти мережі і яка містить колекцію серверів, клієнтів або тих та інших. При відновленні з розділу файлова система перевіряє множину транзакцій розділу.

За системним одночасним викликом на одній машині встановлюються тип сеансу для кожного клієнта з відповідними встановленнями необхідних блокувань читання і запису файла за допомогою процесу VENUS. Семантика розподілу файлів виглядає як технологія двофазного блокування.

При виникненні конфлікту оновлення файлу в паралельних сеансах відміняється, але версії зберігаються клієнтом для подальшого узгодження.

Кешування і реплікація в CODA забезпечують високу доступність до файлів. Кешування на клієнті покращує масштабованість і відмовостійкість. Сервер фіксує дані про кешування файлу на клієнті і за допомогою зворотного виклику (callback promise). Після першої зміни файлу клієнтом за допомогою відміни зворотного виклику (callback break) сервер повідомляє про некоректність копії. При подальшому відкриванні файлу в наступному сеансі всі клієнти отримають нову версію файлу з сервера.

Реплікація томів для колекції серверів, які називають групою сховища файлів VSG (Volume Storage Group), дозволяє клієнту обмежитись доступними серверами VSG – AVSG (Accessible Volume Storage Group). Якщо AVSG порожня для даного клієнта, то клієнт вважається від'єднаним (disconnected).

Підтримка несуперечності реплікованого тому здійснюється протоколом реплікованого запису, зокрема, з використанням варіанту «читаємо раз-пишемо все» ROWA (read one write all). При зміні прочитаного файлу при закритті файла клієнт пересилає його всім членам AVSG за допомогою паралельної пересилки MultyRPC.

Захист даних в CODA забезпечується наявністю захищених каналів і використанням захищених викликів RPC із застосуванням системної автентифікації. Авторизація в серверах VICE здійснюється для каталогів, а не для файлів.

7.6.4 Система World Wide Web

За стандартизацію протоколів удосконалення міжопераційної взаємодії і удосконалення систем на основі World Wide Web (WWW) з 1994 року відповідає консорціум World Wide Web, який є органом CERN і M.I.T.

Система WWW є системою клієнт–сервер із мільйонами серверів, які обробляють запити на отримання документів і створення нових. Документи мають уніфікований вказівник ресурсу URL (Uniform Resource Locator) за форматами:

- протокол `:// DNS-ім'я [:порт/файл файлової системи]`;
- протокол `:// IP адреса [:порт/файл файлової системи]`.

Документи створюють на основі мов HTML, XML, Java, Java-Script та інших і читають на клієнтському боці за допомогою браузера. Браузер надає спектр методів і засобів надання інформації, яка оброблялась Web-серверами і Web-клієнтами.

Документ після синтаксичного розбору подається у браузері у вигляді дерева синтаксичного розбору. Вузол дерева є елементом документа і реалізує стандартний інтерфейс з методами доступу до змісту, посилання на батьківські і дочірні вузли тощо згідно з об'єктною моделлю документів DOM (Document Object Model), яку називають динамічною мовою HTML (DHTML). Інтерфейс дозволяє визначити в документі сценарії обходу дерева синтаксичного розбору, перегляду і модифікації

вузлів, їх значення тощо. Сучасна мова розмітки гіпертексту є мова HTML (Hyper Text Markup Language).

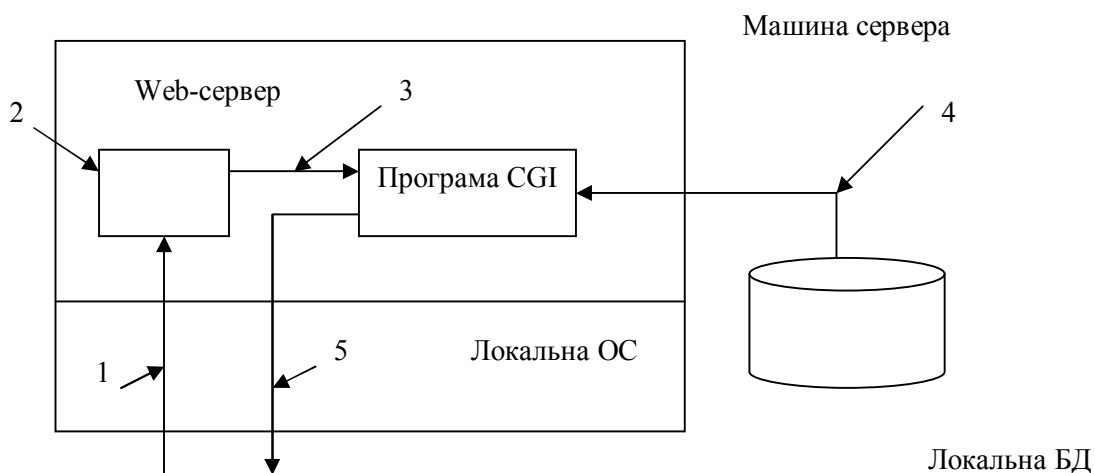
Розширеною мовою розмітки є мова XML (eXtensible Markup Language), у якій, на відміну від HTML, передбачено структурування документа і роботу з різними типами документів. Правила форматування XML-документа реалізується, зокрема, за допомогою вбудовування XML-документа в HTML-документ, або з використанням мови стилів XSL (eXtensible Style Language).

Мови XML і HTML підтримують входження елементів мультимедіа, зокрема зображень, відеокліпів, аудіофайлів, інтерактивну анімацію. Тип документа часто визначають типом багаторольового розширення пам'яті Інтернет у MIME Style (Multipart Internet Mail Extensions). У MIME розрізняють типи і підтипи, основні з яких наведені в табл. 7.5. Останній тип застосовують для складених документів.

Архітектуру клієнт-сервер WWW доповнено підтриманням взаємодії з клієнтами за допомогою узагальненого інтерфейсу шлюза CGI (Common Gateway Interface). Взаємодія з машиною сервера показано на рис. 7.38.

Таблиця 7.5 -Типи даних MIME-документів

Тип	Підтип	Опис
Text	Plain	Неформатований текст
	HTML	Текст на HTML
	XML	Текст на XML
Image	GIF	Відображення у форматі GIF
	JPEG	Відображення у форматі JPEG
Audio	BASIC	Подання аудіоданих восьмибітовим кодом PCM, 8000 Герц
	Tone	Тон
Video	MPEG	Відео у форматі MPEG
	Pointer	Подання вказівника «миші»
Application	Octet-stream	Не інтерпретована послідовність байтів
	Postscript	Друковані документи у форматі Postscript
	PDF	Друковані документи у форматі PDF
Multipart	Mixed	Незалежна частина у заданому порядку
	Parallel	Частини, що переглядаються одночасно



1-запит на отримання документа, надісланого на сервер; 2 – процес введення;
 3 – запуск програми отримання документа; 4 – взаємодія з БД; 5 – створений
 HTML - документ; 6 – надсилання відповіді на запит браузера клієнта.

Рис. 7.38 Використання CGI на боці сервера

Отримавши запит, сервер запускає вказану в запиті програму і передає їй параметри запиту. Результат повертається у вигляді HTML-документа, який відсилається браузеру клієнта. Перед цим сервер може обробити документ, зокрема з використанням серверного сценарію (server – side script), який виконується сервером у разі локального отримання документа.

Сценарій задається тегом `<SERVER>` HTML залежно від сервера. Клієнту також можуть передаватись скомпільовані аплети (applets) для виконання в просторі адрес браузера клієнта. Серверні аплеті – сервлети (servlets) може виконувати сервер із застосуванням методів з комунікаційним протоколом передавання гіпертексту HTTP (Hypertext Transfer Protocol).

В основу протоколу HTTP покладено протокол TCP. За запитом клієнта до сервера, або навпаки, встановлюється з'єднання TCP, за яким надсилається сам запит. Сучасні версії HTTP підтримують схоронний зв'язок з'єднань (persistent connections), які дозволяють передавати кілька запитів і виконувати конвеєризацію (pipelining), тобто надсилати кілька запитів однією командою.

Протокол HTTP орієнтовано на передавання в обох напрямках між клієнтом і сервером. Основні операції клієнта наведено в табл. 7.6.

Таблиця 7.6 Операції клієнта за протоколом HTTP

Операція	Опис
head	Запит на повернення заголовка документа
get	Запит на повернення клієнту документа
put	Запит на збереження документа
post	Дані, що додаються до документа
delete	Запит на вилучення документа

У HTTP розрізняють повідомлення-запити і повідомлення-відповіді, які мають структуру, як показано на рис. рис. 7.39. Існує близько 20 типів заголовків, які уточнюють дані запити і відповіді.

У WWW використовують два типи процесів: Web-браузери для доступу до документів і Web-сервери для оброблення запитів браузерів. Для підтримання різних типів документів у браузерах застосовують модулі розширення (plug-ins). Ці модулі являють собою невеликі програми для оброблення документів певного типу. Тип документа зазвичай задається як тип MIME.

Для роботи з різними протоколами і локальним кешем використовують клієнтські процеси – Web-замісники (Web-proxy), які

вбудовуються в браузер. Принцип використання замісника показано на рис.7.40.

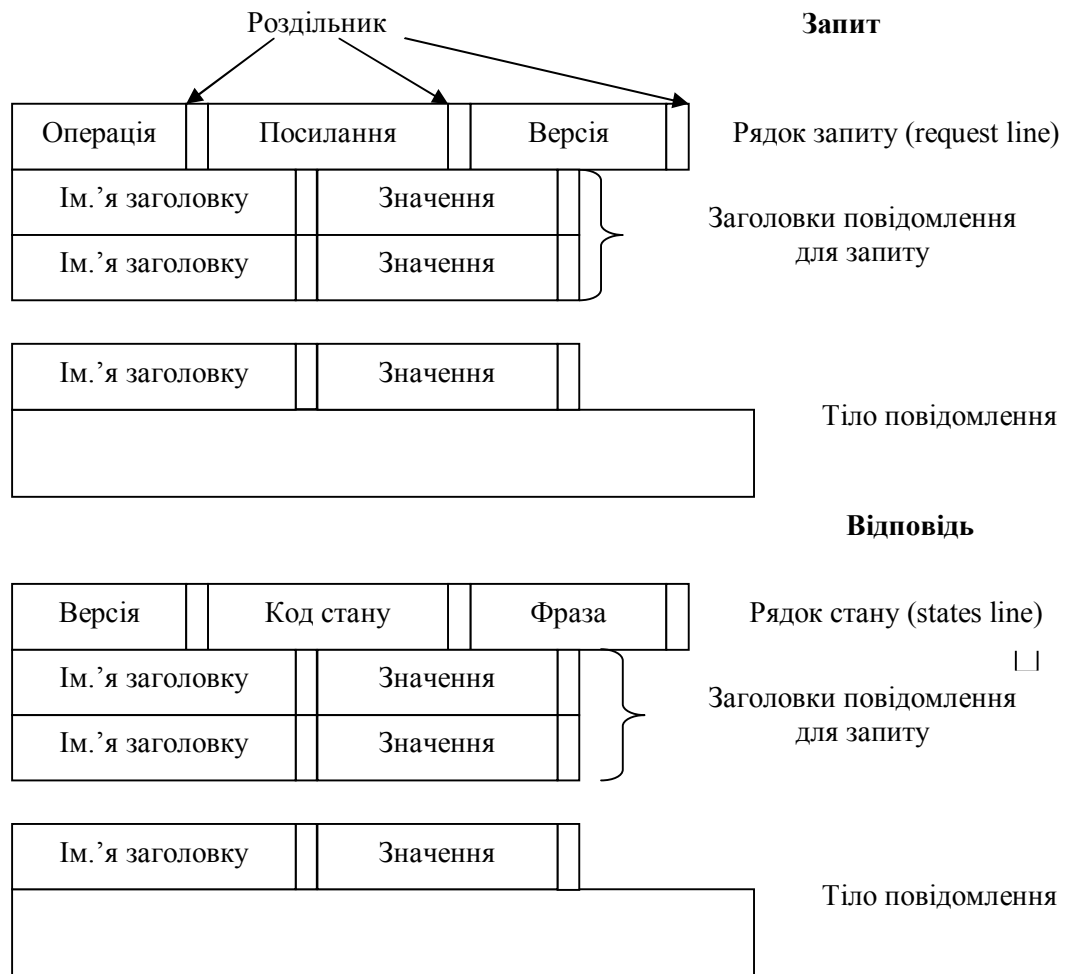


Рис. 7.39 Структури повідомлень протоколу HTTP

Web-сервери обробляють HTTP-запити і повертають клієнту відповіді. Наприклад, для Web-сервера Apache на платформі UNIX створюються модулі під керуванням ядра (рис. 7.41).

Для HTTP-запиту модуль створює запис запиту з посиланням на документ, який міститься в HTTP – запису. Довільний модуль Apache працює з цим записом, змінюючи поля. Після завершення документ повертається клієнту. Обробники (handlers) модуля отримують як єдиний параметр вказівника на запис запиту. Вони викликаються модулем у

певній фазі виконання: розв’язання посилань на документи в локальне ім’я файлу автентифікації клієнта; контроль доступу клієнта; контроль доступу запиту; визначення типу MIME-відповіді; загальна фаза оброблення; пересилання відповіді; протоколювання даних про оброблення запиту.

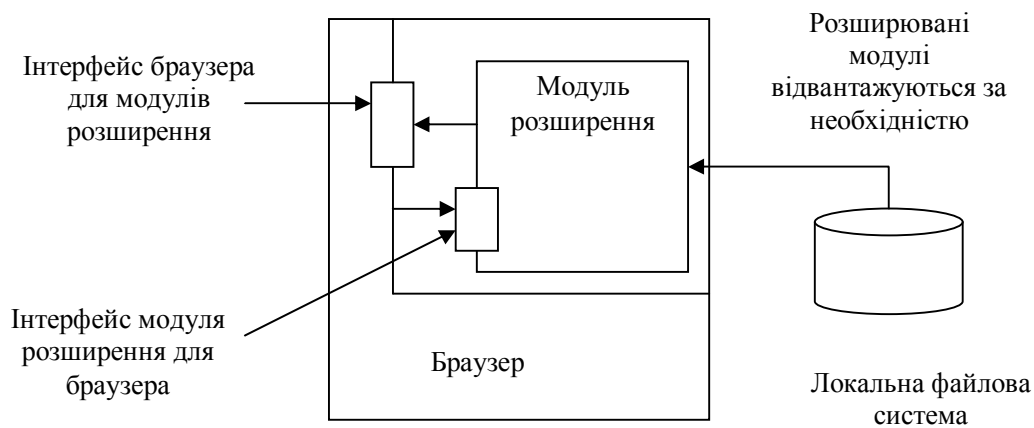


Рис. 7.40 Використання у Web-браузері модуля розширення

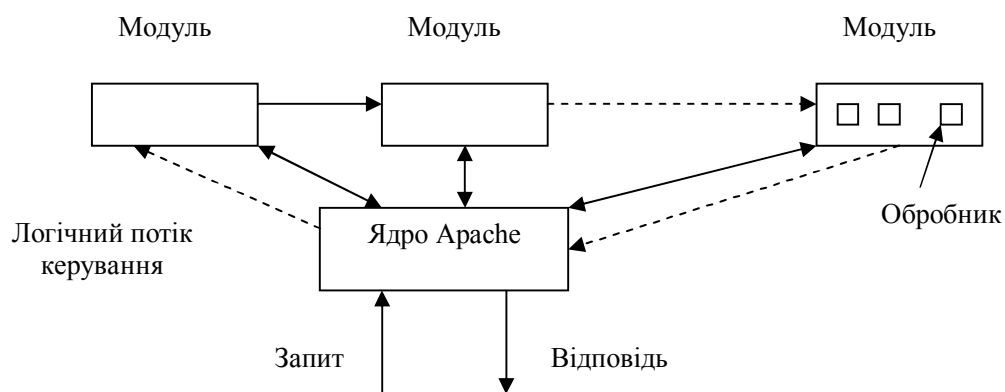


Рис. 7.41 Структура Web-сервера Apache

Для розвантаження потоку запитів у структурі Web передбачають кластери реплікованих Web-серверів. Зовнішній інтерфейс розподіляє HTML-запити між репліками кластера серверів.

Розподіляти запити можна за змістом (content-aware request distribution) на підставі перегляду змісту HTML-запиту, за яким з'єднання TCP прозора для клієнта перемикається безпосередньо на сервер. У результаті сервер безпосередньо відповідає на запит клієнта.

Іменування у WWW реалізується на основі уніфікованого ідентифікатора ресурсів URI (Uniform Resource Identifier), який має дві функції – уніфікованого вказівника ресурсів URL (Uniform Resource Locator) і уніфікованого імені ресурсу URN (Uniform Resource Name) , який є унікальним і глобальним і не залежить від місцеположення ресурсу.

В URL для визначення місцеположення застосовують DNS-імена або IP-адреси і, в разі потреби – порт та ім'я документу. В URN вирізняють три частини (рис. 7.42).

“urn”	Простір імен	Ім'я ресурсу
-------	--------------	--------------

Рис. 7.42 Структура імені URN

Наприклад

urn : left : rfc2648 документ rfc

urn : isbn : O-B-349945-6 книга за isbn

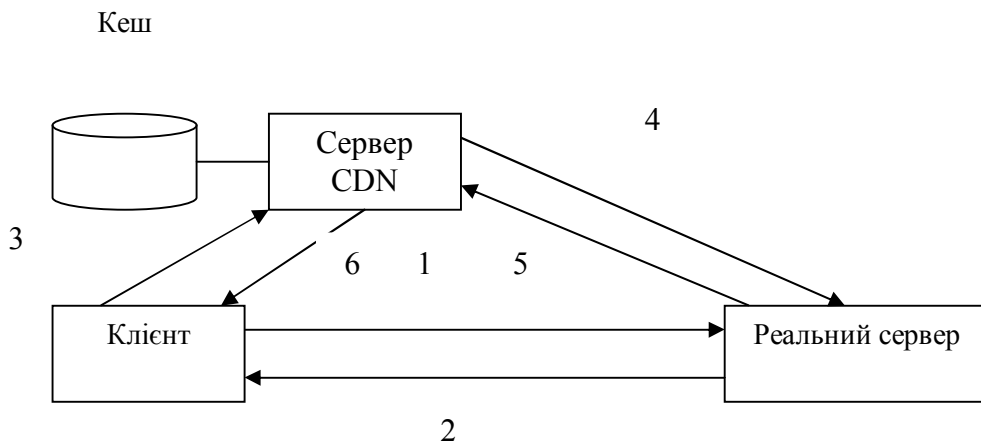
urn : ietf : rfc ; 2648 документ IETF

Синхронізація WWW стає дедалі актуальною у зв'язку з підтриманням сумісної роботи над Web-документами. Для цього HTTP розширено і розроблено протокол розподіленої підготовки документів з

контролем версії Web DAV (Web Distributed Authoring and Versioning). Синхронізація у Web DAV підтримується простим механізмом блокування записів. Блокування виключного записування може бути надано лише одному клієнту. У блокуванні сумісного записування передбачено поділ документа на частини з використанням маркерів блокувань клієнтами, які підтверджують їх права на запис.

Кешування даних клієнтом і реплікації серверів забезпечують у WWW підвищення продуктивності. Кешування виконується через кеш браузера з використанням Web-замісника, який може кешувати отримані від сервера відповіді. В останньому випадку можливе використання цих даних кількома клієнтами, тобто кеш використовується так само, як і в разі розподілу.

Реплікацію в Web використовують такими способами: із використанням кластерів Web-серверів і дзеркальних серверів; із застосуванням мережі доставлення змісту CDN (Content Delivery Network) або мережі розподіленого змісту (Content Distribution Network) для колекцій серверів, розподілених в Інтернеті. У CDN застосовують різні технології реплікації і розподілення документів. Так, у службі RADAR реплікація виконується за регіональним принципом. У CDN кожний документ має головну HTML-сторінку із вбудованими іншими документами (відео, аудіо тощо). За частотою запитів до документів на вбудований документ посилення за допомогою вказівника URL змінюється так, що він посилається на віртуальний образ (virtual ghost) реального сервера CDN. Послідовність операцій розв'язання вказівника URL показано на рис. 7.43.



1 – запит основного документа; 2 – документ з посиланнями на убудовані документи; 3 – запит убудованих документів; 4 – отримання убудованого документа із локального кеша або із сервера; 5 – убудовані документи; 6 – убудовані документи.

Рис. 7.43 Принцип роботи мережі CDN

7.6.5 Система Lotus Notes

Система Lotus Notes орієнтована на бази даних і працює під управлінням операційних систем Windows, UNIX. Система має архітектуру клієнт-сервер і спочатку розроблялась для локальних мереж. Загальну структуру показано на рис. 7.44.

Сервер Domino (Domino servers) керує колекцією БД і надає доступ RPC до розподілених БД віддаленим клієнтом на іншим серверам. Клієнти, сервери і бази даних пов'язані між собою за допомогою компонента проміжного рівня, який має назву служб об'єктів Notes NOS (Notes Object Services), що розташовані на базовими ОС. Це дозволяє використовувати розподілені БД.

Основним елементом даних є нотатка (note), яка є списком елементів (items). Елементом є комірка для збереження нотатки даних і має тип даних (текстовий, числовий тощо). Нотатки утворюють ієрархію, яка

включає початкову нотатку і відповідь на неї. Розрізняють нотатки за даними (data notes) та інші типи нотаток. Нотатка з даними подібна до Web- документів з різними типами даних.

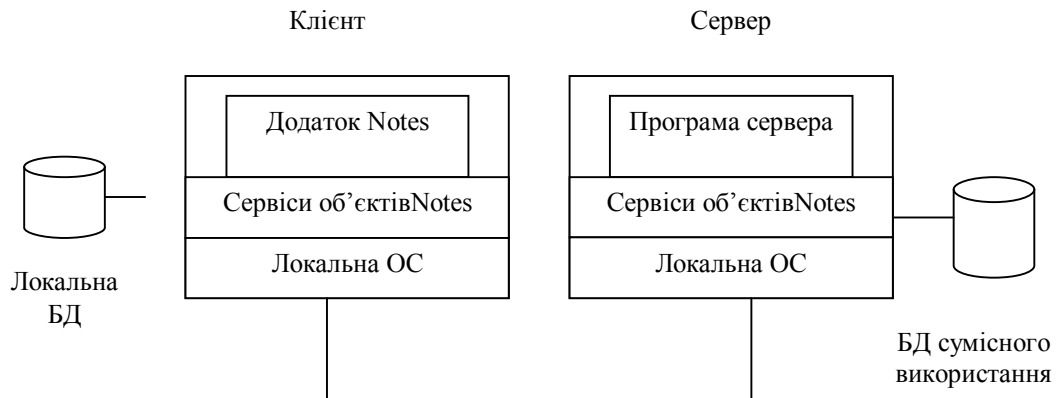


Рис. 7.44 Загальна структура Lotus Notes

Основними категоріями нотаток є конструкції БД (типи Form – форма, Field – поле, View – структура для показу колекції документів); дані (тип Document) і адміністрування (типи ACL - список контролю доступу до БД, Rep1Formula - реплікація бази даних). Взаємопосилання між нотатками здійснюється за допомогою посилань на нотатки (notelinks), які включають базу даних і нотатку у ній. На нотатку можна посилатись за допомогою URL.

Система RPC є частиною проміжного рівня операційної системи (NDS). Взаємодія процесів на різних машинах має обмежений характер порівняно з CORBA, DCOM і реалізується за допомогою інтерфейсів системи RPC Notes. Поштові повідомлення надсилаються в форматі MIME із використанням протоколів SMTP, Lotus Notes тощо. В системі існують засоби автоматичного надсилання електронних листів для реагування на події в базах даних. Ці посилання можуть автоматично міняти зміст бази даних.

Розділяють машини-сервери і машини-клієнти. Клієнт, крім браузера і звичайних програм взаємодії з серверами має програму для розробки нотаток, форм, подій тощо, і виконання додатків. Програма розробки подібна до існуючих для БД.

Сервер обробляє запити за допомогою основної програми і завдань, що в неї входять, відкриває і закриває локальні БД, підтримує непротивірччя даних і керує кластерами серверів. Існують також додаткові завдання сервера (server tasks) у вигляді процесів, які виконуються на одній машині з головним сервером і знаходяться під його повним контролем. Цей головний сервер і його завдання утворюють сервер Domino, який має таку структуру (рис.7.45).

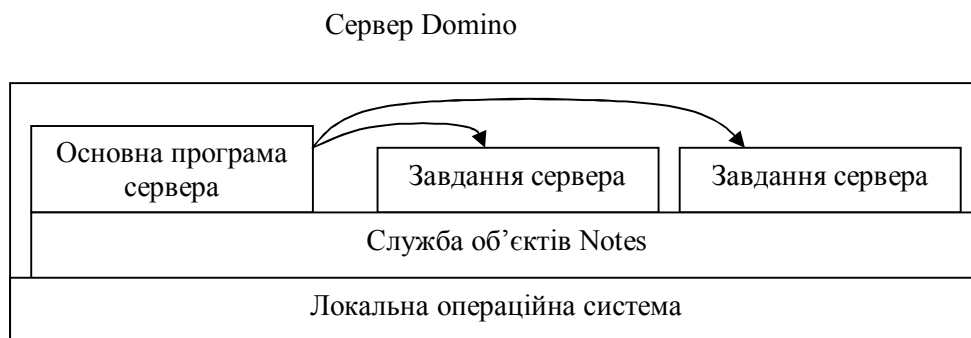
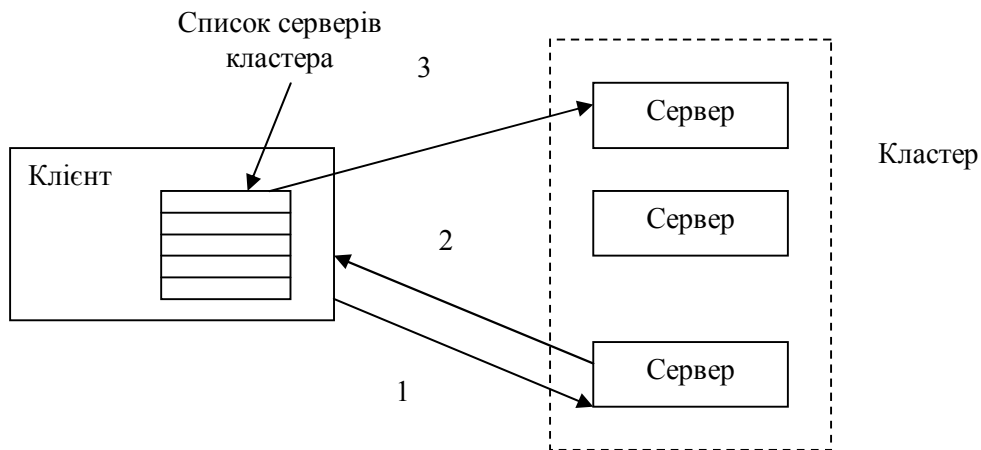


Рис. 7.45 Сервер Domino

Сервери Domino можна об'єднувати у кластери з двох-шести машин з однаковими серверами і ідентичними копіями БД. Клієнти незрозумілі клієнту, і клієнт може вибирати сервер для запиту на основі наданого списку серверів кластера. , як показано на рис. 7.46.



- 1- запит даних про сервер;
 2- вибір найменш завантаженого сервера;
 3 – запит до вибраного сервера.

Рис. 7.46 Кластер серверів Domino

Іменування нотаток здійснюється за допомогою звичайних операцій БД, з використанням подань і посилань. Додаткові сутності – користувачі, сервери, відкриті ключі тощо зберігаються у спеціальній базі даних – каталозі Domino (Domino directory).

Імена у вигляді рядка символів підтримуються службою розпізнаних імен (distinguished names), в тому ієрархічних, на основі LDAP, та за допомогою URL, який застосовується для доступу до серверів Domino через Web. Адреса URL в Notes складається з DNS-імені сервера Domino, шляху до БД Notes на тому ж сервері, і ідентифікатора імені операції і параметрів операції (необов'язково). За виконання операції відповідає сервер Domino. Для кластерів зміни передають на всі репліки кластера.

Відмовостійкість забезпечується системою відновлення на основі журналу з упереджаючим записом перед виконанням операції транзакції. Автентифікація ґрунтується на застосуванні відкритих ключів, які

містяться у сертифікатах. Дійсність сертифікатів забезпечується використанням посвідчень для користувачів, серверів і сертифікуючи організацій. Посвідчення реалізовано у вигляді файла з ключем і сертифікатом. При реєстрації нового користувача, він отримує два набори пар (відкритий ключ, закритий ключ). Перша пара з 512 бітним ключем RSA використовується для шифрування даних за межами США, а друга пара з 630-бітним ключем в межах США і для автентифікації і підпису повідомлень. Сертифікат зберігається у файлі користувача. Закриті ключі шифруються ключем користувача. Аналогічно створюють файли для Domino.

7.6.6 Система Jini

Система Jini є розподіленою системою, орієнтованою на Java. Вона складається із взаємопов'язаних елементів з моделлю узгодженого регенеративного зв'язку на основі системи узгодження JavaSpace. Система JavaSpace є розподіленим простором даних, у якому зберігаються кортежі, що є типізованими наборами посилань на об'єкти Java.

Система JavaSpace може містити кілька просторів JavaSpace. Організація простору показано на рис. 7.47.

Під час записування кортежу процесом виконується маршалінг усіх його полів. Кортежі вміщуються у простір Java Space операцією write, за якою зберігається нова копія кортежа після маршалінгу. Посилання на копію використовується так само як і на екземпляр кортежу (tuple instance), використовуючи кортеж – еталон (template).

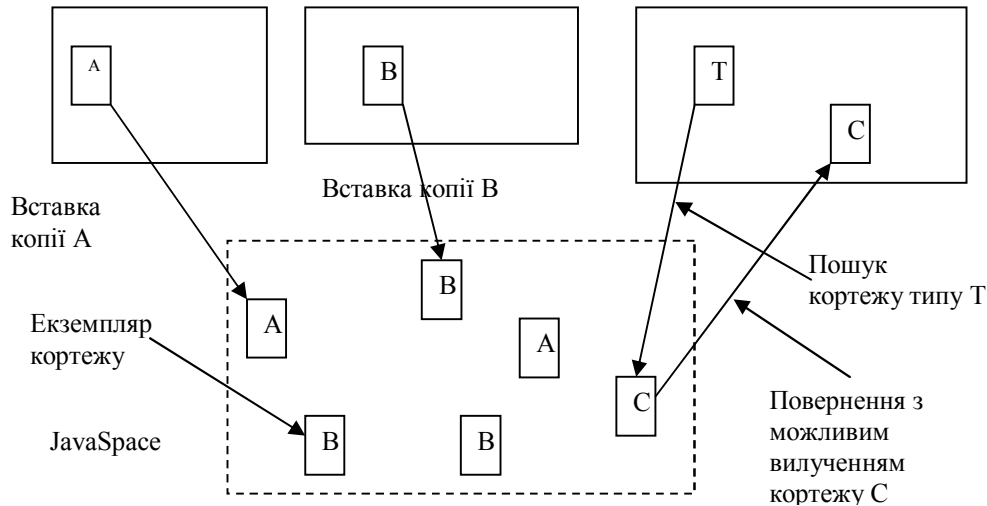


Рис. 7.47 Організація простору JavaSpace в Jini

Під час читання екземпляра в Java Space процес надає інший кортеж – еталон, який відповідає екземплярам кортежу, що зберігається в просторі. Еталонний кортеж є типізованим набором посилань на об'єкти. Поля кортежу можуть мати значення NULL або посилання на реальні об'єкти. Далі відбувається порівняння по полях екземпляра кортежу і еталонного кортежу. У разі збіжності полів виконується демаршалінг екземпляра і він повертається процесу, який ініціював читання. Архітектуру системи Jini показано на рис. 7.48.

Нижній рівень інфраструктури містить базові механізми, зокрема підтримання взаємодії за допомогою звернень RMI (віддалений доступ до методів) мови Java. Сервіси надаються звичайним процесам, а також пристроям, які програмне забезпечення Jini і Java не можуть виконувати.

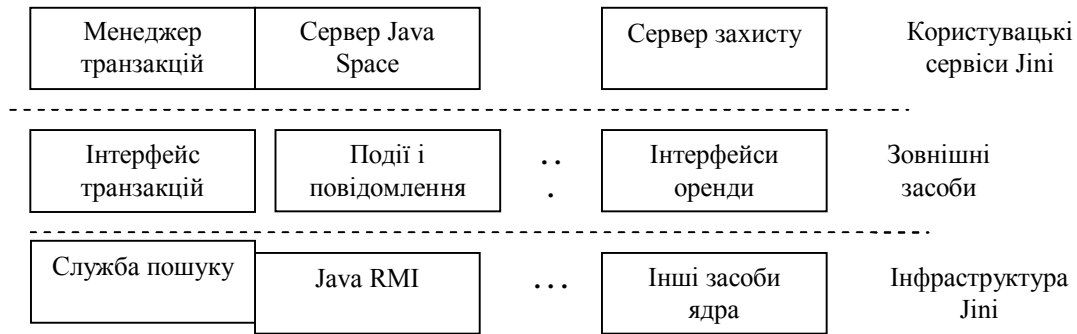


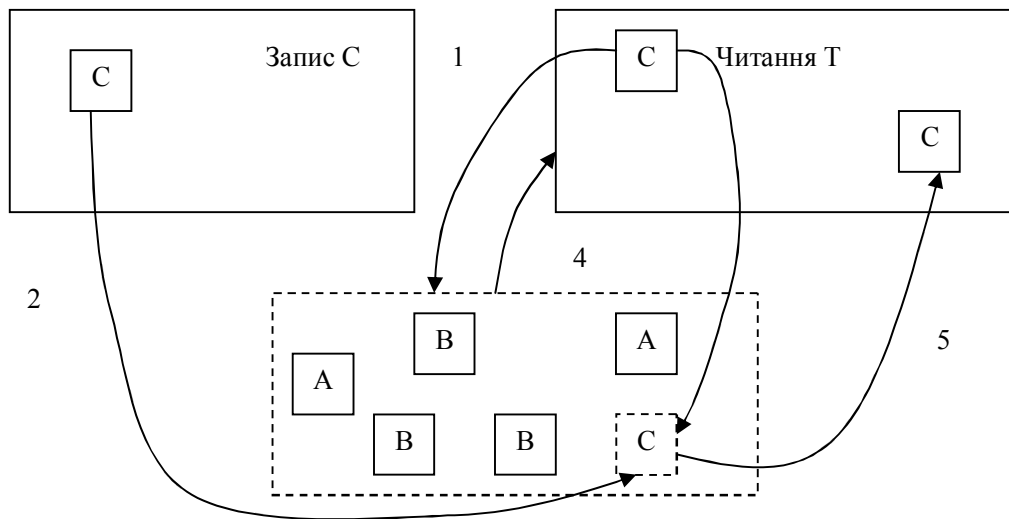
Рис. 7.48 Архітектура системи Jini

На рівні зовнішніх засобів використовуються засоби ефективної реалізації служб. Верхній рівень складається з клієнтів і служб. Програмам цього рівня дозволяється напряму використовувати механізми інфраструктури Jini. Зв'язок у Jini реалізовується за допомогою RMI, а механізм взаємодії має просту підсистему подій і сповіщення. Подія відбувається в межах об'єкту або в JavaSpace. Якщо вона зацівила клієнта, він дозволяє зареєструвати її на цьому об'єкті. Події в Java Space виконуються так, як показано на рис. 7.49.

Реєстрація події орендується, тому після її завершення клієнт (або процес, якому надсилались сповіщення за дорученням клієнта припиняють отримувати сповіщення.

Сповіщення про подію реалізується об'єктом віддаленим викликом об'єкта за схемою RMI прослуховувача (listener object). Цей об'єкт може викликатись повторно з поданням сповіщення порядкових номерів.

Набір екземплярів кортежів для розподілених серверів Java Script може бути розподілено по кількох машинах. Ефективна розподілена реалізація передбачає розв'язання двох проблем: емуляції асоціативної адресації без глобального пошуку; розподілення екземплярів кортежів по машинах.



1 – запит про сповіщення Т; 2 – вставка копії С; 3 – сповіщення про вставку С;
 4 – пошук кортежу типу Т; 5 – повернення і, можливо, вилучення кортежу С

Рис. 7.49 Використання подій в Java Space

Для розв'язання проблем кожний кортеж вважають структурою даних певного типу. Це дозволяє спростити програмування і розподілення кортежів. Для розсилання кортежів використовують широкомовного розсилання і створення підпросторів JavaSpace на різних машинах. Такий підхід стикається з проблемами масштабованості, які полягають у застосуванні широкомовної розсилки для вставлення кортежу у простір кортежів. При цьому треба враховувати, що глобального простору кортежів немає.

Іменування реалізується за допомогою сервіси пошуку Java Space прикладного рівня, однак Jini має власну службу пошуку (lookup service) нижнього рівня.

Кожна сервіс має ідентифікатор Service ID (Service Identifier), який є унікальним 128-бітним значенням. Ідентифікатор сервісу реєструється як елемент сервісу (service item) у службі пошуку (можливо, кількох). Поле

Service елемента сервісу надсилається на об'єкт (можливо віддалений через RMI), який реалізує службу, а поле Attributesets містить набір кортежів, що описують службу.

У системі Jini підтримується кілька механізмів синхронізації, зокрема з використанням операцій записування read, take, а також транзакцій. Організація транзакцій показано на рис. 7.50.

Клієнт може надіслати запит менеджеру транзакцій, й який повертає ідентифікатор транзакцій і може визначати час до підтвердження і переривати транзакцій. Менеджер може визначити оренду створеної транзакції, перериваючи її після завершення. Клієнт може також вимагати приєднання до транзакції інших процесів. Простір Java Space може брати участь у транзакціях.

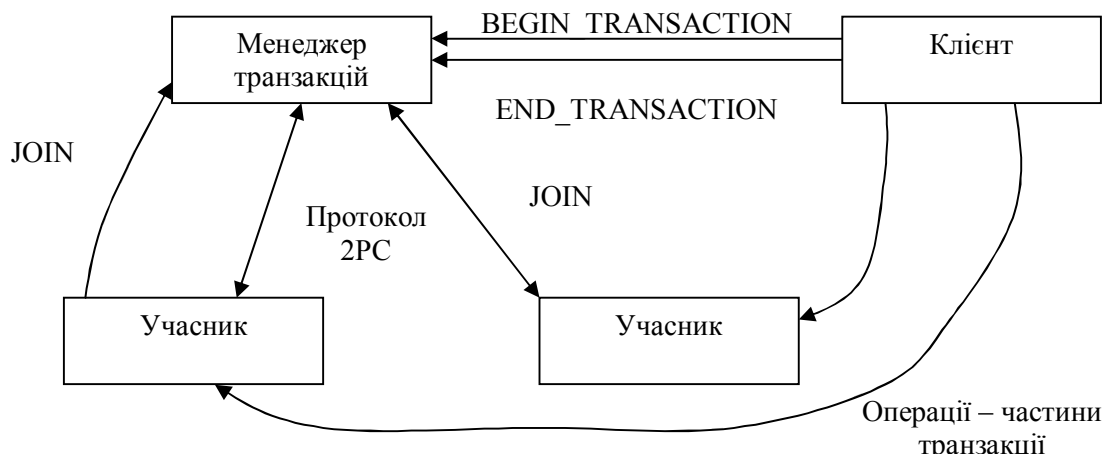


Рис. 7.50 Діаграма організації транзакції у системі Jini

Система Jini не підтримує відмовостійкості. Захист реалізується відповідно до процедур RMI мови Java. За допомогою спеціального сервісу авторизації Java JAAS (Java Authentication and Authorization Service)

здійснюються автентифікація і авторизація користувачів системи за допомогою відділеного підключення модуля автентифікації PAM (Pluggable Authentication Module). Він займає середнє положення між додатком і ОС (рис. 7.51).

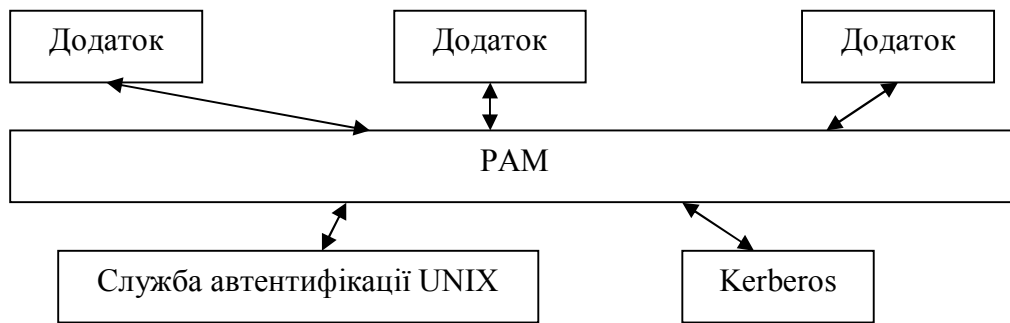


Рис. 7.51 Взаємодія PAM із сервісами захисту

7.7 Програмні агенти

Програмним агентом (software agent) називають автономний процес, спроможний реагувати на середовище виконання і викликати зміни у середовищі виконання, можливо, разом з іншими програмними агентами. Виділяють кооперативні (collaborative agent), мобільні (mobile agent), інтерфейсні (interface agent) і інформаційні (information agent) агенти.

Кооперативним агентом називають агента мультиагентної системи, який вирішує деякі загальні задачі. Мобільним агентом називають агента, який може переміщуватися з машини на машину. Інтерфейсний агент дозволяє користувачу працювати з кількома додатками. Інформаційний агент може керувати інформацією, яка надходить з множини інформаційних джерел.

Узагальнена модель платформи агента згідно з розробкою FIP (Foundation for Intelligent Physical Agents) має вигляд (рис. 7.52).

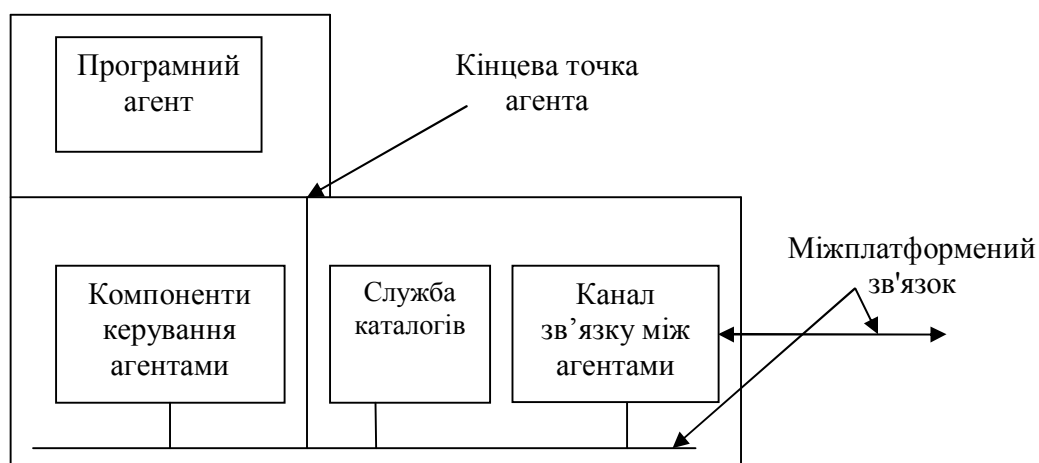


Рис. 7.52 Узагальнена модель платформи агента

Компонент керування агентами надає службі іменувань і механізми створення і знищення агентів, перегляду кінцевої точки на наявність агента. Сервіс каталогів базується на використанні атрибутів і дозволяє виявити наявність більших агентів на платформі. Канал зв'язку між агентами АСС (Agent Communication Channel) відповідає за взаємодію між різними платформами агентів, зокрема у вигляді сервера (d'Agent). Зв'язок між АСС здійснюється за допомогою протоколу ІОР (Internet Inter-ORB Protocol).

Зв'язок між агентами відбувається за допомогою комунікаційного протоколу прикладного рівня, який називають мовою взаємодії агентів АСЛ (Agent Communication Language). В АСЛ для повідомлення виділяють обмежену кількість цілей і зміст. Агент – відправник і агент - отримувач однаково розуміють ціль повідомлення, яка однозначно визначає реакцію отримувача (табл. 7.7).

Таблиця 7.7 Цілі повідомлення

Ціль	Опис	Зміст повідомлення
INFORM	Інформувати, що повідомлення істинно	Припущення
QUERY-REF	Запит даного об'єкта	Вираз
ACCEPT_PROPOSAL	Повідомлю про прийняття повідомлення	Ідентифікатор пропозиції
PROPOSE	Надати пропозицію	Пропозиція
REJECT	Повідомити, що дана пропозиція відхилена	Ідентифікатор
PROPOSAL	Запитати виконання дії	Специфікація дії
REQUEST	Підписатись на джерело інформації	Посилання на джерело
SUBSCRIBE	Запитати, чи істина дана пропозиція	Пропозиція
QUERY - IF	Запитати пропозицію	Залежить від пропозиції
CFP		

Запитання та завдання для самоконтролю

1. Описати структурну організацію розподілених систем.
2. Які виникають задачі при створенні розподілених систем?
3. Обґрунтувати форми прозорості у розподілених інформаційних системах.
4. Навести основні характеристики відкритості розподілених інформаційних систем.
5. Як побудована архітектура і реалізації у стандарті CORBA?
6. Описати особливості функцій ORB і його реалізацій.
7. Описати взаємодію клієнта і сервера на основі ORB.
8. Які є типи служб в CORBA?
9. Якою є модель взаємодії агентів-додатків в CORBA?
10. У чому полягають особливості посилань IOR і які є типи посилань в IOR?
11. Яку структуру має розподілений об'єкт DSO в

системі GLOBE?

12. Особливості взаємодії локальних об'єктів LO в DSO системи GLOBE.

13. Як виконується прив'язка процесу до об'єкта в системі GLOBE?

14. Охарактеризувати типи служб системи GLOBE і їх застосування при різних видах зв'язку.

15. Дати аналіз серверних процесів в системі GLOBE.

16. Описати способи уникнення аварійних ситуацій в системі GLOBE.

17. Описати механізм іменування DSO в системі GLOBE.

18. Як здійснюється захист DSO в системі GLOBE?

19. Описати особливості внутрішньої архітектури вузла Virtue у системі CODA.

20. Які особливості використання RPC2 у системі CODA?

21. Описати особливості іменувань та реплікації файлів з використанням ідентифікаторів у системі CODA.

22. Описати механізми обробки конфліктів доступу до файлів у системі CODA.

23. Як забезпечується відмовостійкість у системі CODA?

24. Описати особливості застосування мов HTML і XML у WWW.

25. Дати характеристику запитів і відповідей у протоколі HTTP.

26. Описати особливості обробки запитів і іменувань у Web-серверах.

27. Як виконується кешування і реплікація даних у клієнтах і серверах WWW?

28. Особливості будови сервера і нотаток у системі Lotus Notes.

29. Як взаємодіють клієнти і сервери у системі Lotus Notes?

30. Як здійснюється автентифікація у системі Lotus Notes?
31. Описати особливості я простору даних системи Java Space Jini.
32. В чому полягають відмінності архітектури системи Jini від інших?
33. Як здійснюється взаємодії у системі Jini?
34. Описати механізми синхронізації у системі Jini.
35. Як реалізують процеси серверів для процесів клієнтів?
36. Класифікація серверів розподілених систем.
37. У чому полягають особливості організації серверів розподілених об'єктів?
38. Як відбувається взаємодія між машинами розподіленої системи на різних рівнях OSI?
39. Які особливості передавання повідомлень за допомогою протоколів TCP, T/TCP?
40. Яка модель взаємодії в розподілених системах із урахуванням проміжного рівня?
41. Як відбувається взаємодія за допомогою виклику RPC між клієнтом і сервером?
42. Опишіть механізм взаємодії розширеного RPC.
43. У чому полягають особливості RMI?
44. Що описує взаємодія на основі сокетів?
45. Опишіть особливості моделі черг повідомлень.
46. Які є режими зв'язку на основі потоків даних?

РОЗДІЛ 8 МАТЕМАТИЧНІ МЕТОДИ ПОБУДОВА СППР ПРИ ПРОГНОЗУВАННІ ДИНАМІКИ ЧАСОВИХ РЯДІВ

8.1 Методи прогнозування на основі часових рядів

Прогнозування на основі часових рядів – один із самих популярних підходів до прогнозування розвитку економічних процесів, об'ємів торгових операцій, об'ємів виробництва та накопичення продукції на складах, оцінювання альтернативних економічних стратегій, формування бюджетів підприємств та держави, прогнозування та менеджмент економічних і фінансових ризиків та інше [6].

Методи прогнозування можна поділити на три основні класи [9]:

1. Прогнозування на основі суджень; прогнозування, що ґрунтується на суб'єктивних судженнях (оцінках), інтуїції, поглиблених знаннях конкретної області та іншій інформації, що має відношення до прогнозованого процесу (зокрема, так зване передбачення).

2. Методи прогнозування на основі використання часового ряду однієї змінної, тобто, на основі авторегресії, авторегресії з ковзним середнім (АРКС) та АРКС плюс модель тренду.

3. Методи прогнозування на основі використання часових рядів декількох змінних [10].

У третьому випадку ендогенна змінна, що прогнозується, залежить від кількох регресорів або екзогенних змінних у правій частині рівняння. Очевидно, що в загальному випадку метод прогнозування може поєднувати у собі 2-3 наведених вище методи.

Відомо багато методів прогнозування на основі використання часових рядів. Найбільш поширеними серед них є: метод групового врахування аргументів (МГВА), регресійний аналіз; нейромережі; байєсівські мережі; нечіткі множини; нечіткі нейромережі тощо.

Регресійний аналіз включає методи авторегресії (АР), авторегресії з ковзним середнім (АРКС), авторегресії з інтегрованим ковзним середнім (АРИКС), лінійної та нелінійної множинної регресії, квантильної регресії, регресійні дерева.

У загальному випадку прогноз може бути подано одним (точковим) значенням змінної, інтервалом, в який попадає випадкова змінна, а також ймовірністю прийняття змінною (чи подією) деякого значення у вибраному інтервалі. Якщо для опису процесу застосовують лінгвістичні змінні, то прогнозом буде нечітке лінгвістичне значення, але його також можна перетворити в чітке число.

8.2 Основні складові прогнозування процесу

Можна по-різному ставити задачу прогнозування в залежності від рівня прийняття рішення та конкретної поставленої задачі управління чи контролю. Прогнозування може стосуватись таких складових процесу:

- детермінованого тренду, як індикатора довгострокових змін процесу;
- випадкового (нерегулярного) тренду, як показника коротко- та середньострокових змін;
- короткострокових змін, тобто, прогнозування коливань (відхилень), що накладаються на тренд;
- сезонних ефектів;
- приростів (швидкості) зміни процесу, які визначаються першими різницями;
- дисперсії або стандартного відхилення, як міри розсіювання процесу (наприклад, волатильність, яку часто використовують за міру ризику у інвестуванні або міру якості на виробництві);
- якісних змінних (за допомогою нечітких множин, БМ тощо);

– комбінацій вказаних елементів процесів.

Відповідно до того, які складові процесу необхідно прогнозувати, ставиться задача побудови математичної, ймовірнісної (ймовірнісно-статистичної) або логічної моделі, що має меті забезпечити високу якість прогнозу на заданому горизонті. Розглянемо деякі можливості математичного опису складових процесів різної природи.

8.2.1 Детермінований тренд

Якщо описати детермінований тренд за допомогою полінома від часу довільного порядку

$$y(k) = a_0 + a_1 \cdot k + a_2 \cdot k^2 + \dots + a_p \cdot k^p + \varepsilon(k), \quad E[\varepsilon(k)] = 0,$$

то визначення прогнозу тренду зводиться до підстановки в це рівняння бажаного значення часу k і застосування безумовного математичного сподівання.

Прогнозування значень тренду вважається довгостроковим, оскільки детермінований тренд вказує на довгострокові зміни процесів. Обмеження на випадковий процес, $E[\varepsilon(k)] = 0$, необхідне для коректного застосування методів оцінювання моделей, а також для подальшого аналізу якості оцінок прогнозів. Ця вимога забезпечується автоматично, якщо метод оцінювання параметрів моделі вибрано коректно у відповідності до відомих статистичних характеристик процесу, що моделюється.

8.2.2 Стохастичний тренд

Для опису і прогнозування стохастичного тренду можна скористатись, наприклад, відомим рівнянням випадкового кроку із перетином (константою):

$$y(k) = a_0 + y(k-1) + \varepsilon(k),$$

розв'язок якого має вигляд:

$$y(k) = y_0 + k a_0 + \sum_{i=1}^k \varepsilon(i).$$

Сума $\sum_{i=1}^k \varepsilon(i)$ в правій частині останнього рівняння описує випадкову складову тренду. Цю складову називають ще нерегулярною.

8.2.3 Прогнозування коливань, що накладаються на тренд

Коливання, що накладаються на тренд, або короткострокові зміни процесу, можна описати рівняннями авторегресії з ковзним середнім:

$$y(k) = \sum_{i=1}^p a_i y(k-i) + \sum_{j=1}^q b_j \varepsilon(k-j) + \varepsilon(k).$$

Техніку обчислення прогнозу за допомогою такої моделі з використанням умовного та безумовного математичного сподівання розглянемо нижче в цьому розділі.

При переході до перших та різниць вищих порядків із процесу вилучається тренд відповідного порядку. Наприклад, якщо процес містить лінійний тренд, то його вилучають перші різниці; тобто після переходу до різниць ми маємо справу з коливаннями, що накладаються на тренд.

Можливість прогнозування сезонних ефектів досягається за рахунок введення у модель процесу відповідних значень основної змінної із затримками (лагами), що відповідають періодичності ефекту. Як буде показано нижче, сезонний ефект може бути врахований як за допомогою основної змінної, так і за допомогою відповідного процесу ковзного середнього в адитивній та мультиплікативній формах. При цьому якість моделі (і оцінки прогнозу) можуть бути різними. Сезонний ефект може супроводжуватись гетероскедастичністю, тобто зміною дисперсії процесу у часі. Це призводить до появи задачі прогнозування дисперсії.

8.2.4 Прогнозування дисперсії процесу

Якщо дисперсія процесу змінюється в часі, то для її опису можна вибрати рівняння для формування відповідної функції прогнозування та обчислення в подальшому оцінки прогнозованого значення. Процеси зі змінною дисперсією отримали назву гетероскедастичних. Методика побудови моделей гетероскедастичних процесів передбачає наступні кроки:

- математичний опис самої змінної рівнянням авторегресії невисокого порядку (наприклад, першого);
- математичний опис умовної дисперсії, як динамічної змінної, за допомогою рівняння прийнятної (за якістю прогнозу) структури.

8.3 Умовні та безумовні статистичні характеристики

При виконанні статистичного аналізу випадкових процесів використовують умовні та безумовні статистичні характеристики. Зокрема, для знаходження короткострокових та довгострокових прогнозів розвитку процесів застосовують умовне, E_k , та безумовне, E , математичне сподівання, відповідно. Різниця у визначенні цих характеристик полягає у наступному.

Безумовні статистичні характеристики – це константи, які розглядають і оцінюють на довільних часових інтервалах, не накладаючи умов на змінні, функції та об'єми інформації, необхідної для визначення цих характеристик. Тобто інформація для обчислення дисперсії вважається відомою і постійною на всьому інтервалі, що розглядається в процесі аналізу.

Так, безумовне математичне сподівання використовують для знаходження довгострокових прогнозів або умов економічної рівноваги (в економетриці). Наприклад, безумовні вибіркоче середнє і дисперсія обчислюються за відомими простими формулами:

$$\bar{x} = \frac{1}{N} \sum_{k=1}^N y(k), \quad \text{Var}(x) = \frac{1}{N-1} \sum_{k=1}^N [x(k) - \bar{x}]^2,$$

а безумовне математичне сподівання позначається символом E . У виразі для дисперсії немає необхідності зменшувати значення N на одиницю, якщо потужність ряду досить велика, наприклад, декілька десятків.

Умовні статистичні характеристики в аналізі динаміки процесів, представлених часовими рядами, – це функції часу, які визначають на кожний конкретний момент часу k . При цьому для їх обчислення

необхідно, щоб на вибраний (заданий) момент була наявна інформація стосовно значень змінних та функцій, яка необхідна для виконання відповідних обчислень.

Умовне математичне сподівання застосовують для визначення короткострокових та середньострокових прогнозів. Умовну дисперсію процесу і стандартне відхилення також часто використовують як міру ризику. Наприклад, при аналізі фінансових процесів, формуванні правил торгівлі на біржі, аналізі банківських та економічних ризиків. В технічних системах дисперсія також відіграє значну роль при визначенні ступеня відхилення вузла (наприклад, підшипника, корпусного елемента, крила) від нормального (заданого) робочого стану. Тому вміння правильно аналітично описати дисперсію дає можливість з високою точністю описати і спрогнозувати значення відхилень від норми.

Так, умовне математичне сподівання стохастичного процесу AP(1) визначається за виразом:

$$\begin{aligned} E_k[y(k+1) | y(k), y(k-1), \dots, y(0), \varepsilon(k), \varepsilon(k-1), \dots, \varepsilon(0)] = \\ = E_k[a_0 + a_1 y(k) + \varepsilon(k+1)] = a_0 + a_1 y(k) \end{aligned}$$

за умови, що $E_k[\varepsilon(k+l)] = 0, l \geq 1$.

Умовне вибіркове середнє та умовну вибіркєву дисперсію можна наближено обчислити за виразами:

$$\bar{x}(k) = \frac{1}{k} \sum_{i=1}^k x(i); \quad \text{var}_k(x) = \frac{1}{k-1} \sum_{i=1}^k [x(i) - \bar{x}(k)]^2, \quad k = 2, \dots, N.$$

В результаті застосування цих виразів отримаємо ряд значень умовного середнього та умовної дисперсії, тобто ще дві характеристики

процесу, якими можна скористатись при побудові математичних і статистичних моделей.

Наближені значення умовних вибірових статистичних характеристик можна визначити, також, за аналогією із обчисленням ковзного середнього. При такому підході необхідно вибрати ширину ковзного вікна і обчислювати значення статистичного параметра рухаючись крок за кроком від початку до кінця часового ряду. Наприклад, якщо вибрати вікно шириною п'ять значень ряду, то умовна дисперсія буде обчислюватись за виразом:

$$\text{var}_k(x) = \frac{1}{4} \sum_{i=k-2}^{k+2} [y(i) - \bar{y}]^2, \quad k = 3, \dots, N - 2,$$

або у загальному вигляді:

$$\text{var}_k(x) = \frac{1}{d-1} \sum_{i=k-(d-1)/2}^{k+(d-1)/2} [y(i) - \bar{y}]^2, \quad k = \frac{d-1}{2}, \dots, N - \frac{d-1}{2},$$

де d – ширина ковзного вікна.

Ширина вікна залежить від того, наскільки швидко змінюється у часі дисперсія. Якщо вона має високу динаміку, то ширину вікна вибирають рівною 5-9. Незважаючи на наближеність таких розрахунків, практика моделювання свідчить, що обчислені значення виявляються досить корисними при побудові моделей.

У складніших випадках обчислюють функцію умовної дисперсії, яка точніше відображає характер її зміни у часі. Наприклад, можна розглянути таку модель процесу:

$$y(k) = \sqrt{f[x(k)]} \varepsilon(k), \quad (8.1)$$

$$\varepsilon(k) = \beta \varepsilon(k-1) + v(k), \quad (8.2)$$

де $y(k) \in \mathfrak{R}, k = 1, 2, \dots, N$;

$\{v(k)\}$ – множина незалежних однаково розподілених (НОР) величин, що мають нормальний розподіл з параметрами $N(0,1)$;

$$\beta \in \Theta = (-1; 1);$$

$$f[x(k)] \in C^p [0, 1].$$

Припущення про те, що $v(k)$ – гаусів процес зроблено для зручності викладок та забезпечення коректної умови для оцінювання параметрів.

Змінна $x(k) \in [0, 1]$ упорядкована за значеннями, тобто:

$$x(1) \leq x(2) \leq \dots \leq x(N),$$

$$\text{де } x(k) = \frac{k}{N}, \quad k = 1, 2, \dots, N.$$

Функцію $f[x(k)]$ називають функцією дисперсії, хоча вона не повністю описує дисперсійно-коваріаційну структуру процесу (8.1) – (8.2). Припустимо, що $f[x(k)]$ має p неперервних похідних. Безумовну дисперсію процесу $y(k)$ можна визначити таким чином:

$$\begin{aligned} \text{var}[y(k)] &= f[x(k)] E[\varepsilon^2(k)] = f[x(k)] E\{\beta \varepsilon(k-1) + v(k)\}^2 = \\ &= \beta^2 E[\varepsilon^2(k-1)] + E[v^2(k)] f[x(k)] = \\ &= \beta^2 f[x(k)] E[\varepsilon^2(k-1)] + f[x(k)] = \\ &= \beta^2 \text{var}[y(k)] + f[x(k)]. \end{aligned}$$

Звідси отримаємо:

$$\text{var } [y(k)] = \frac{f[x(k)]}{1 - \beta^2}.$$

Таким чином, процес (8.1) – (8.2) є умовно і безумовно гетероскедастичним. Задача полягає у тому, щоб коректно побудувати та оцінити функцію $f[x(k)]$.

8.4 Побудова функцій прогнозування

8.4.1 Прогнозування без знаходження розв'язку рівнянь

Структура різницевого рівняння така, що воно дозволяє виконувати прогнозування на один крок (один період дискретизації вимірів) без додаткових перетворень. Тобто у праву частину необхідно підставити минулі значення змінних і обчислити оцінку прогнозу головної змінної в лівій частині. Але для того щоб знайти оцінку прогнозу на більше число кроків, необхідно застосувати деякі попередні перетворення різницевих рівнянь. Розглянемо можливі підходи до формування функцій прогнозування та обчислення оцінок прогнозованих значень [20].

8.4.1.1 Прогнозування для рівняння AP(1)

Розглянемо рівняння AP(1):

$$y(k) = a_0 + a_1 y(k-1) + \varepsilon(k), \quad E[\varepsilon(k)] = 0. \quad (8.3)$$

Збільшимо незалежну змінну, час на одиницю і запишемо рівняння :

$$y(k+1) = a_0 + a_1 y(k) + \varepsilon(k+1).$$

Якщо коефіцієнти a_0, a_1 відомі, то можна знайти умовне математичне сподівання на основі відомої інформації до моменту k включно:

$$\begin{aligned} E_k[y(k+1)] &= E_k[y(k+1) | y(k), y(k-1), \dots, \varepsilon(k), \varepsilon(k-1), \dots)] = \\ &= a_0 + a_1 E_k[y(k)] = a_0 + a_1 y(k), \end{aligned}$$

оскільки $y(k)$ у момент k є відомою константою.

По аналогії запишемо рівняння (8.3) для моменту $k+2$:

$$y(k+2) = a_0 + a_1 y(k+1) + \varepsilon(k+2)$$

і знайдемо умовне математичне сподівання:

$$\begin{aligned} E_k[y(k+2)] &= a_0 + a_1 E_k[y(k+1)] = a_0 + a_1 E_k[a_0 + a_1 y(k)] = \\ &= a_0 + a_0 a_1 + a_1^2 y(k). \end{aligned}$$

Для наступного моменту часу маємо:

$$E_k[y(k+3)] = a_0 + a_0 a_1 + a_0 a_1^2 + a_1^3 y(k).$$

Таким чином, для загального випадку прогнозування на s кроків можна записати вираз для оцінки прогнозу:

$$\begin{aligned} \hat{y}(k+s) &= \\ &= E_s[y(k+s)] = a_0 \left(\sum_{i=0}^{s-1} a_1^i \right) + a_1^s y(k) = a_0 \sum_{i=0}^{s-1} a_1^i + a_1^s y(k). \end{aligned} \quad (8.4)$$

Отримане рівняння називають функцією прогнозування на довільне число кроків. Прогноз представляє собою збіжний процес, якщо $|a_1| < 1$, тобто

$$\lim_{s \rightarrow \infty} E_k[y(k+s)] = \frac{a_0}{1-a_1}, \quad |a_1| < 1, \quad (8.5)$$

де a_1 є знаменник геометричної прогресії.

Вираз (8.5) свідчить про те, що для будь-якого стаціонарного процесу АР чи АРКС оцінка умовного прогнозу асимптотично ($s \rightarrow \infty$) збігається до безумовного середнього.

Знайдемо похибку прогнозування за умови, що $E[\varepsilon(k)] = 0$:

$$f_k(s) = y(k+s) - E_k[y(k+s)].$$

Похибка прогнозу на один крок:

$$f_k(1) = y(k+1) - E_k[y(k+1)] = a_0 + a_1 y(k) + \varepsilon(k+1) - a_0 - a_1 y(k) = \varepsilon(k+1).$$

Похибка прогнозу на два кроки:

$$\begin{aligned}
f_k(2) &= y(k+2) - E_k[y(k+2)] = \\
&= a_0 + a_1[a_0 + a_1y(k) + \varepsilon(k+1)] + \varepsilon(k+2) - E_k[y(k+2)] = \\
&= a_0 + a_0a_1 + a_1^2y(k) + a_1\varepsilon(k+1) + \varepsilon(k+2) - a_0 - a_0a_1 - a_1^2y(k) = \\
&= \varepsilon(k+2) + a_1\varepsilon(k+1).
\end{aligned}$$

Таким чином, можна записати вираз для похибки оцінки прогнозу для довільного числа кроків прогнозування як

$$f_k(s) = \varepsilon(k+s) + a_1\varepsilon(k+s-1) + a_1^2\varepsilon(k+s-2) + \dots + a_1^{s-1}\varepsilon(k+1).$$

Враховуючи те, що $E[f_k(s)] = 0$, оцінка прогнозу, яка обчислюється за виразом (8.4), є незміщеною. Дисперсія похибки прогнозування:

$$Var[f_k(s)] = \sigma^2 [1 + a_1^2 + a_1^4 + a_1^6 + \dots + a_1^{2(s-1)}],$$

тобто дисперсія є функцією кількості кроків s . Асимптотичне значення дисперсії похибки прогнозу для стаціонарного процесу

$$\lim_{s \rightarrow \infty} Var[f_k(s)] = \frac{\sigma^2}{1 - a_1^2},$$

де a_1^2 є знаменник геометричної прогресії.

8.4.1.2 Узагальнення функції прогнозування (на процес АРКС(p,q))

Знайдемо спочатку функцію прогнозування для процесу АРКС(2,1), який описується наступним рівнянням:

$$y(k) = a_0 + a_1 y(k-1) + a_2 y(k-2) + \varepsilon(k) + \beta_1 \varepsilon(k-1).$$

Для наступного моменту часу можна записати

$$y(k+1) = a_0 + a_1 y(k) + a_2 y(k-1) + \varepsilon(k+1) + \beta_1 \varepsilon(k).$$

Умовне математичне сподівання для $y(k+1)$ має вигляд:

$$E_k[y(k+1)] = a_0 + a_1 y(k) + a_2 y(k-1) + \beta_1 \varepsilon(k),$$

Значення $\varepsilon(k)$ розглядають як відому величину на момент k включно.

При цьому $E[\varepsilon(k+j)] = 0, \forall j > 0$.

Для моменту часу $k+2$ маємо:

$$y(k+2) = a_0 + a_1 y(k+1) + a_2 y(k) + \varepsilon(k+2) + \beta_1 \varepsilon(k+1),$$

і умовне математичне сподівання

$$\begin{aligned} E_k[y(k+2)] &= a_0 + a_1 E_k[y(k+1)] + a_2 E_k[y(k)] = \\ &= a_0 + a_1 [a_0 + a_1 y(k) + a_2 y(k-1) + \beta_1 \varepsilon(k)] + a_2 y(k) = \\ &= a_0 + a_0 a_1 + a_1^2 y(k) + a_1 a_2 y(k-1) + a_1 \beta_1 \varepsilon(k) + a_2 y(k) = \\ &= a_0(1 + a_1) + (a_1^2 + a_2) y(k) + a_1 a_2 y(k-1) + a_1 \beta_1 \varepsilon(k). \end{aligned}$$

Можна знайти також наступне умовне математичне сподівання для оцінки прогнозу на три кроки:

$$\begin{aligned}
E_k[y(k+3)] &= a_0 + a_1 E_k[y(k+2)] + a_2 E_k[y(k+1)] = \\
&= a_0(1 + a_1 + a_1^2 + a_2) + (a_1^3 + 2a_1 a_2)y(k) + (a_1^2 a_2 + a_2^2)y(k-1) + \\
&\quad + \beta_1(a_1^2 + a_2)\varepsilon(k) .
\end{aligned}$$

З отриманих виразів для умовного математичного сподівання видно, що рекурсивна формула для прогнозу на довільне число кроків s може бути записана тільки у наступному вигляді:

$$E_k[y(k+s)] = a_0 + a_1 E_k[y(k+s-1)] + a_2 E_k[y(k+s-2)]. \quad (8.6)$$

Якщо корені характеристичного рівняння, записаного для (8.6), знаходяться всередині одиничного кола, то оцінка прогнозу асимптотично збігається до безумовного середнього значення:

$$\lim_{S \rightarrow \infty} E[y(k+s)] = \frac{a_0}{1 - a_1 - a_2},$$

а для довільного процесу АРКС(p,q) оцінку умовного прогнозу можна записати так:

$$E_k[y(k+s)] = a_0 + \sum_{i=1}^p a_i E_k[y(k+s-i)].$$

8.4.2 Побудова функції прогнозування на основі різницевого рівняння

Розглянемо як приклад рівняння АРКС(1,1):

$$y(k) = a_0 + a_1 y(k-1) + \varepsilon(k) + \beta_1 \varepsilon(k-1), \quad |a_1| < 1,$$

де $\varepsilon(k)$ – білий шум з нульовим середнім;

$y(0) = y_0$ – відома початкова умова.

Для однорідного рівняння $y(k) - a_1 y(k-1) = 0$ розв'язком є $A a_1^k$, де A – довільна константа.

Частковий розв'язок можна знайти за допомогою лагового оператора L у вигляді:

$$y(k) = \frac{a_0}{1 - a_1} + \frac{\varepsilon(k)}{1 - a_1 L} + \frac{\beta_1 \varepsilon(k-1)}{1 - a_1 L}.$$

Використовуючи властивості лагового оператора, запишемо загальний розв'язок як

$$y(k) = \frac{a_0}{1 - a_1} + \sum_{i=0}^{\infty} a_1^i \varepsilon(k-i) + \beta_1 \sum_{i=0}^{\infty} a_1^i \varepsilon(k-i-1) + A a_1^k.$$

Для того щоб знайти значення довільної константи скористаємось початковою умовою:

$$k = 0: \quad y_0 = \frac{a_0}{1 - a_1} + \sum_{i=0}^{\infty} a_1^i \varepsilon(-i) + \beta_1 \sum_{i=0}^{\infty} a_1^i \varepsilon(-i - 1) + A.$$

Запишемо розв'язок із урахуванням отриманого значення довільної константи:

$$y(k) = \frac{a_0}{1 - a_1} + \sum_{i=0}^{\infty} a_1^i \varepsilon(k - i) + \beta_1 \sum_{i=0}^{\infty} a_1^i \varepsilon(k - i - 1) + \left[y_0 - \frac{a_0}{1 - a_1} - \sum_{i=0}^{\infty} a_1^i \varepsilon(-i) + \beta_1 \sum_{i=0}^{\infty} a_1^i \varepsilon(-i - 1) \right] a_1^k.$$

Оскільки

$$\begin{aligned} \sum_{i=0}^{\infty} a_1^i \varepsilon(k - i) &= \varepsilon(k) + a_1 \varepsilon(k - 1) + a_1^2 \varepsilon(k - 2) + \dots + a_1^k \varepsilon(0) + a_1^{k+1} \varepsilon(-1) + \dots, \\ - a_1^k \sum_{i=0}^{\infty} a_1^i \varepsilon(-i) &= -a_1^k \varepsilon(0) - a_1^{k+1} \varepsilon(-1) - a_1^{k+2} \varepsilon(-2) - \dots, \end{aligned}$$

то

$$y(k) = \frac{a_0}{1 - a_1} + \sum_{i=0}^{k-1} a_1^i \varepsilon(k - i) + \beta_1 \sum_{i=0}^{k-1} a_1^i \varepsilon(k - i - 1) + \left[y_0 - \frac{a_0}{1 - a_1} \right] a_1^k.$$

Знайдемо рівняння для оцінки прогнозу із урахуванням того, що на момент $k = 0$ відоме значення збурення $E_0[\varepsilon(0)] = \varepsilon_0$, тобто необхідно знайти $E_0[y(k)]$. Таким чином, функція прогнозу приймає такий вигляд:

$$E_0[y(0+k)|y(0),\varepsilon(0)] = \frac{a_0}{1-a_1} + E_0 \left[\sum_{i=0}^{k-1} a_1^i \varepsilon(k-i) + \beta_1 \sum_{i=0}^{k-1} a_1^i \varepsilon(k-1-i) \right] + E_0 \left(\left[y_0 - \frac{a_0}{1-a_1} \right] a_1^k \right).$$

Враховуючи, що

$$\sum_{i=0}^{k-1} a_1^i \varepsilon(k-i) = \varepsilon(k) + a_1 \varepsilon(k-1) + a_1^2 \varepsilon(k-2) + \dots + a_1^{k-1} \varepsilon(-1)$$

і

$$\beta_1 \sum_{i=0}^{k-1} a_1^i \varepsilon(k-1-i) = \beta_1 \varepsilon(k-1) + \beta_1 a_1 \varepsilon(k-2) + \beta_1 a_1^2 \varepsilon(k-3) + \dots + \beta_1 a_1^{k-1} \varepsilon(0),$$

то отримаємо:

$$y(0+k) = E_0[y(k)] = \frac{a_0}{1-a_1} + \beta_1 a_1^{k-1} \varepsilon_0 + \left[y_0 - \frac{a_0}{1-a_1} \right] a_1^k. \quad (8.7)$$

Рівняння (8.7) можна розглядати як функцію прогнозування на k кроків наперед на основі інформації, наявної на момент $k = 0$.

Знайдемо функцію прогнозування на s кроків вперед на основі інформації, наявної на момент k . Спочатку зробимо заміну індексів в рівнянні (8.7):

$$\begin{aligned} y(0+s) = E_0[y(s)] &= \frac{a_0}{1-a_1} + \beta_1 a_1^{s-1} \varepsilon_0 + \left[y_0 - \frac{a_0}{1-a_1} \right] a_1^s = \\ &= \left(\frac{a_0}{1-a_1} \right) (1-a_1^s) + \beta_1 a_1^{s-1} \varepsilon_0 + y_0 a_1^s. \end{aligned}$$

Тепер виконаємо оновлення часового індексу для змінних y і ε на k одиниць вперед:

$$y(k+s) = E_k[y(k+s)] = \left(\frac{a_0}{1-a_1} \right) (1-a_1^s) + \beta_1 a_1^{s-1} \varepsilon(k) + y(k) a_1^s.$$

Отримане рівняння є функцію прогнозування на основі відомої інформації про процес на момент k , включно. Використовуючи наведені вище викладки, можна записати функції прогнозування для довільної кількості кроків таким чином:

$$s=1: \quad E_k[y(k+1)] = a_0 + \beta_1 \varepsilon(k) + a_1 y(k);$$

$$s=2: \quad E_k[y(k+2)] = \left(\frac{a_0}{1-a_1} \right) (1-a_1^2) + \beta_1 a_1 \varepsilon(k) + a_1^2 y(k);$$

$$s=3: \quad E_k[y(k+3)] = \left(\frac{a_0}{1-a_1} \right) (1-a_1^3) + \beta_1 a_1^2 \varepsilon(k) + a_1^3 y(k);$$

При цьому
$$\lim_{S \rightarrow \infty} E_k [y(k + s)] = \frac{a_0}{1 - a_1}.$$

Можна досить просто перейти від моделі АРКС(1,1) до моделі АР(1), якщо покласти $\beta_1 = 0$.

Для АР(1) отримаємо функцію прогнозування у вигляді:

$$E_k [y(k + s)] = \left(\frac{a_0}{1 - a_1} \right) (1 - a_1^S) + a_1^S y(k).$$

По аналогії можна знайти функції прогнозування для моделей іншої структури, табулювати отримані функції і використовувати їх при прогнозуванні процесів різної природи.

Експериментально встановлено, що модель, яка може бути використана для прогнозування, повинна бути отримана (оцінена) на основі не менше ніж 50 спостережень (вимірів) основної змінної.

Якщо модель побудована на основі вибірки даних довжиною N, то для рівняння АРКС(2,1)

$$y(k) = a_0 + a_1 y(k - 1) + a_2 y(k - 2) + \varepsilon(k) + \beta_1 \varepsilon(k - 1),$$

функцію прогнозування можна записати так:

$$E_N [y(N + 1)] = \hat{a}_0 + \hat{a}_1 E_N [y(N)] + \hat{a}_2 E_N [y(N - 1)] + \hat{\beta}_1 \hat{\varepsilon}(k);$$

$$E_N [y(N + 2)] = \hat{a}_0 + \hat{a}_1 E_N [y(N + 1)] + \hat{a}_2 E_N [y(N)];$$

$$\dots E_N [y(N + s)] = \hat{a}_0 + \hat{a}_1 E_N [y(N + s - 1)] + \hat{a}_2 E_N [y(N + s - 2)], \quad s \geq 2.$$

Приклад 8.1. Побудуємо функцію прогнозування для рівняння АР(2):

$$y(k) = 3,0 + 0,9y(k-1) - 0,2y(k-2) + \varepsilon(k)$$

з початковими умовами: $y(0) = y_0$; $y(1) = y_1$.

а) Знайдемо однорідний розв'язок:

$$\alpha^2 - 0,9\alpha + 0,2 = 0;$$

$$\alpha_1 = 0,5; \alpha_2 = 0,4.$$

$$y^h(k) = A_1 0,5^k + A_2 0,4^k;$$

б) Частковий розв'язок для детермінованої частини:

$$d = 3,0 + 0,9d - 0,2d, \quad \text{або} \quad d = \frac{3,0}{0,3} = 10;$$

в) Частковий розв'язок для стохастичної частини:

$$y^{ps}(k) = \sum_{i=0}^{\infty} \alpha_i \varepsilon(k-i),$$

де $\alpha_0 = 1$; $\alpha_1 = 0,9$; $\alpha_i = 0,9\alpha_{i-1} - 0,2\alpha_{i-2}$, $\forall i > 1$;

г) Повний розв'язок:

$$y(k) = 10,0 + A_1 0,5^k + A_2 0,4^k + \sum_{i=0}^{\infty} \alpha_i \varepsilon(k-i),$$

$$\alpha_i = 0,9\alpha_{i-1} - 0,2\alpha_{i-2}.$$

Для того щоб знайти значення невідомих довільних констант, використаємо початкові умови y_0, y_1 . В результаті отримаємо два рівняння:

$$y_0 = 10,0 + A_1 + A_2 + \sum_{i=0}^{\infty} \alpha_i \varepsilon(-i),$$

$$y_1 = 10,0 + 0,5A_1 + 0,4A_2 + \sum_{i=0}^{\infty} \alpha_i \varepsilon(1-i),$$

з яких можна визначити невідомі константи. Із урахуванням початкових умов розв'язок прийме такий вигляд:

$$y(k) = 10,0 + 0,4^k [5,0(y_0 - 10) - 10(y_1 - 10)] + \\ + 0,5^k [10(y_1 - 10) - 4(y_0 - 10)] + \sum_{i=0}^{k-1} \alpha_i \varepsilon(k-i).$$

Значення залежної змінної в момент $k + s$ при початкових умовах $y(k)$ і $y(k - 1)$ можна знайти так:

$$y(k + s) = 10,0 + 0,4^S \{5,0[y(k - 1) - 10] - 10[(y(k) - 10)]\} + \quad (8.8)$$

$$+ 0,5^S \{[10y(k) - 10] - 4[y(k - 1) - 10]\} + \sum_{i=0}^{S-1} \alpha_i \varepsilon(k + s - i),$$

де $y(k - 1), y(k)$ – початкові умови відносно k – го моменту часу.

В рівнянні (8.8) виконано заміну часової змінної k на $k + s$, але із урахуванням того, що прогнозоване значення визначається відносно k – го моменту часу. Тому верхнє граничне значення над знаком суми визначається як $k - 2 = k + s - 2$, а із урахуванням того, що початковим моментом часу (на який відома необхідна інформація) є $k = 1$, отримаємо $1 + s - 2 = s - 1$. Перевірка нижнього індекса свідчить, що $i = 0 \Rightarrow \varepsilon(k + s)$ і $i = s - 1 \Rightarrow \varepsilon(k + 1)$, тобто границі зміни часового індекса для збурення визначені коректно.

Запишемо умовне математичне сподівання або прогноз на короткий проміжок часу:

$$E_k[y(k + s)] = 10,0 + 0,4^S \{5,0[y(k - 1) - 10] - 10[(y(k) - 10)]\} +$$

$$+ 0,5^S \{[10y(k) - 10] - 4[y(k - 1) - 10]\}.$$

Безумовне середнє або асимптотичний прогноз приймає значення:

$$\lim_{S \rightarrow \infty} E_k [y(k + s)] = 10,$$

тобто прогноз на нескінченності дорівнює безумовному середньому.

8.4.3 Прогнозування з мінімальною дисперсією

Функцію прогнозування можна будувати різними способами, які відрізняються припущеннями стосовно процесу (який описується математично і прогнозується), критерію оптимальності оцінок (якщо він використовується) та вигляду самої функції. Побудова функції прогнозування, яка забезпечує мінімум дисперсії, ґрунтується на таких припущеннях:

1. Процес, що прогнозується, є стаціонарним, гаусовим і має дробово-раціональну спектральну густину.
2. Оптимальною вважається функція прогнозування, яка генерує оцінку прогнозу шляхом мінімізації дисперсії похибки прогнозу, тобто

$$E[y(k + s) - \hat{y}(k + s)]^2 \Rightarrow \min .$$

3. Значення $y(k + s)$, для якого оцінюється прогноз, є функцією всіх попередніх спостережень до моменту k включно.

Результат буде таким же, якщо замінити припущення про нормальність процесу припущенням про те, що він є двічі диференційованим, а функцією прогнозування є лінійна функція спостережень.

Розглянемо стохастичне різницеве рівняння

$$y(k+1) = -ay(k) + \varepsilon(k+1) + \beta\varepsilon(k), \quad (8.9)$$

де $\{\varepsilon(k)\} \sim N(0,1)$, тобто нормально розподілений процес із нульовим середнім та одиничною дисперсією.

Розглянемо спочатку функцію прогнозування на один крок за допомогою рівняння (8.9). Для того щоб знайти значення $y(k+1)$, необхідно знати $y(k)$, $\varepsilon(k)$. Значення $y(k)$ – це значення останнього виміру, а $\varepsilon(k)$ необхідно обчислити.

Нехай відомі початкові значення $\varepsilon(0) = \varepsilon_0$, $y(0) = y_0$. Запишемо рівняння (8.9) у вигляді:

$$\varepsilon(\tau+1) - y(\tau+1) + \beta\varepsilon(\tau) = ay(\tau), \quad 0 \leq \tau \leq k$$

і знайдемо розв'язок для цього рівняння. Додамо до лівої і правої частини член $-\beta y(\tau)$. Отримаємо:

$$[\varepsilon(\tau+1) - y(\tau+1)] + \beta[\varepsilon(\tau) - y(\tau)] = (a - \beta)y(\tau).$$

Тепер введемо нову змінну $x(\tau) = \varepsilon(\tau) - y(\tau)$ і запишемо різницеве рівняння для нової змінної:

$$x(\tau+1) = -\beta x(\tau) + (a - b)y(\tau)$$

і знайдемо розв'язок цього рівняння методом прямої ітерації:

$$\begin{aligned}
 x(1) &= -\beta x(0) + (a - \beta)y(0), \\
 x(2) &= -\beta x(1) + (a - \beta)y(1) = -\beta[-\beta x(0) + (a - \beta)y(0)] + (a - \beta)y(1) = \\
 &= -\beta^2 x(0) - \beta(a - \beta)y(0) + (a - \beta)y(1), \\
 x(3) &= -\beta x(2) + (a - \beta)y(2) = \\
 &= -\beta^3 x(0) + \beta^2(a - \beta)y(0) - \beta(a - \beta)y(1) + (a - \beta)y(2).
 \end{aligned}$$

Для довільного k розв'язок має наступний вигляд:

$$x(k) = (-\beta)^k x(0) + (a - \beta) \sum_{i=0}^{k-1} (-\beta)^{k-1-i} y(i). \quad (8.10)$$

Виконаємо перевірку індексів:

- при $k = 1$ останній член в правій частині має вигляд: $(a - \beta)y(0)$;
- при $k = 2$ маємо наступне рівняння:

$$(a - \beta) \sum_{i=0}^1 (-\beta)^{1-i} y(i) = -\beta(a - \beta)y(0) + (a - \beta)y(1),$$

тобто, часові індекси в рівнянні (8.10) змінюються в належних інтервалах.

Використовуючи рівняння (8.10), можемо записати вираз для $\varepsilon(k)$:

$$\varepsilon(k) = [\varepsilon(0) - y(0)](-\beta)^k + y(k) + (a - \beta) \sum_{i=0}^{k-1} (-\beta)^{k-1-i} y(i), \quad (8.11)$$

Для довільного початкового моменту часу k_0 отримаємо:

$$\varepsilon(k) = [\varepsilon(0) - y(0)](-\beta)^{k-k_0} + y(k) + (a - \beta) \sum_{i=0}^{k-1} (-\beta)^{k-1-i} y(i). \quad (8.12)$$

Оскільки для стаціонарного процесу $|\beta| < 1$, то перший член (8.11) і (8.12) наближається до нуля при $k_0 \rightarrow \infty$ для довільних початкових умов. Таким чином, $\varepsilon(k)$ може бути обчислено безпосередньо за допомогою результатів вимірів із рівняння:

$$\varepsilon(k) = y(k) + (a - \beta) \sum_{i=-\infty}^{k-1} (-\beta)^{k-1-i} y(i).$$

Таким чином, можемо записати рівняння для прогнозування значення $y(k+1)$ на основі спостережень $y(k), y(k-1), y(k-2), \dots$ у вигляді:

$$\begin{aligned} y(k+1|k) &= y(k+1, k) = y(k+1) = -ay(k) + \beta\varepsilon(k) = \\ &= -ay(k) + \beta y(k) + \beta(a - \beta) \sum_{i=-\infty}^{k-1} (-\beta)^{k-i-1} y(i) = \\ &= -(a - \beta)y(k) + \beta(a - \beta) \sum_{i=-\infty}^{k-1} (-\beta)^{k-i-1} y(i) = (\beta - a) \sum_{i=-\infty}^k (-\beta)^{k-i} y(i). \end{aligned}$$

Це рівняння можна переписати у зручнішій рекурсивній формі:

$$\begin{aligned}
 y(k+1, k) &= (\beta - a)[y(k) + \sum_{i=-\infty}^{k-1} (-\beta)^{k-i} y(i)] = \\
 &= (\beta - a)[y(k) + (-\beta) \sum_{i=-\infty}^{k-1} (-\beta)^{k-1-i} y(i)] = -\beta y(k, k-1) + (\beta - a)y(k). \quad (8.13)
 \end{aligned}$$

Тобто функція прогнозування на один крок описується РР першого порядку. Динаміка прогнозу визначається коефіцієнтом β .

За визначенням, похибка прогнозу дорівнює:

$$f(k+1, k) = y(k+1) - y(k+1, k) = \varepsilon(k+1),$$

а її математичне сподівання $E[f(k+1, k)] = 0$.

8.4.4 Альтернативна форма функції прогнозування на один крок

Введемо оператор зсуву $zx(k) = x(k+1)$ або $z^{-1}x(k) = x(k-1)$.

Тепер рівняння:

$$y(k+1) = -ay(k) + \varepsilon(k+1) + \beta\varepsilon(k) \quad (8.14)$$

можна переписати у вигляді:

$$\begin{aligned}
y(k+1) &= \frac{1+\beta z^{-1}}{1+az^{-1}} \varepsilon(k+1) = \varepsilon(k+1) - \varepsilon(k+1) + \frac{1+\beta z^{-1}}{1+az^{-1}} \varepsilon(k+1) = \\
&= \varepsilon(k+1) + \frac{-\varepsilon(k+1) - a\varepsilon(k) + \varepsilon(k+1) + \beta\varepsilon(k)}{1+az^{-1}} = \\
&= \varepsilon(k+1) + \frac{\beta-a}{1+az^{-1}} \varepsilon(k). \tag{8.15}
\end{aligned}$$

Із рівняння (8.14) знайдемо вираз для випадкової величини:

$$\varepsilon(k) = \frac{1+az^{-1}}{1+\beta z^{-1}} y(k)$$

і підставимо цей вираз в (8.15) щоб уникнути змінної $\varepsilon(k)$ з метою зменшення невизначеності прогнозу:

$$y(k+1) = \varepsilon(k+1) + \frac{\beta-a}{1+\beta z^{-1}} y(k).$$

Відніmemo від обох частин цього рівняння прогнозоване значення $y(k+1, k)$ і знайдемо математичне сподівання квадрата похибки прогнозу:

$$E_k [y(k+1) - y(k+1, k)]^2 = E[\varepsilon^2(k+1)] + E \left[\frac{\beta-a}{1+\beta z^{-1}} y(k) - y(k+1, k) \right]^2$$

за умови, що $E[y(k)\varepsilon(k+1)] = 0$.

Якщо дисперсію збурення нормувати до одиниці, тобто $E[\varepsilon^2(k)] = 1$, то можна записати наступне співвідношення:

$$E[y(k+1) - y(k+1, k)] \geq E[\varepsilon^2(k+1)] = 1,$$

де рівність виконується тільки при умові, що:

$$y(k+1, k) = \frac{\beta - a}{1 + \beta z^{-1}} y(k).$$

Таким чином, оптимальне з точки зору мінімуму дисперсії похибки прогнозу значення для прогнозу на один крок можна знайти за допомогою рівняння:

$$y(k+1, k) + \beta y(k, k-1) = (\beta - a)y(k)$$

або

$$y(k+1, k) = -\beta y(k, k-1) + (\beta - a)y(k),$$

що співпадає з отриманим вище рівнянням (8.13).

8.4.5 Функція прогнозування на два кроки

Побудуємо функцію прогнозування на два кроки $y(k+2, k)$ для випадкового процесу АРКС(1,1):

$$y(k) = -ay(k-1) + \varepsilon(k) + \beta\varepsilon(k-1),$$

де $\{\varepsilon(k) : N(0,1)\}$ - випадковий процес з одиничною дисперсією та нульовим середнім значенням.

Для моменту $k+2$ можна записати попереднє рівняння за допомогою оператора зсуву в наступній формі:

$$y(k+2) = \frac{1 + \beta z^{-1}}{1 + a z^{-1}} \varepsilon(k+2).$$

Права частина цього рівняння є лінійною функцією від $\varepsilon(k+2), \varepsilon(k+1), \varepsilon(k), \varepsilon(k-1), \dots$. При цьому змінні $\varepsilon(k), \varepsilon(k-1), \dots$ можна визначити за допомогою спостережень $y(k), y(k-1), \dots$, а випадкові змінні $\varepsilon(k+1), \varepsilon(k+2)$ не залежать від отриманих спостережень на момент k .

Виділимо в правій частині члени з $\varepsilon(k+1), \varepsilon(k+2)$. З цією метою додамо та віднімемо $\varepsilon(k+2)$ в правій частині:

$$\begin{aligned}
y(k+2) &= \varepsilon(k+2) - \varepsilon(k+2) + \frac{1+\beta z^{-1}}{1+z^{-1}} \varepsilon(k+2) = \\
&= \varepsilon(k+2) + \frac{-\varepsilon(k+2) - a\varepsilon(k+1) + \varepsilon(k+2) + \beta\varepsilon(k+1)}{1+az^{-1}} = \\
&= \varepsilon(k+2) + \frac{\beta-a}{1+az^{-1}} \varepsilon(k+1).
\end{aligned}$$

Тепер домножимо чисельник на багаточлен $(1+az^{-1}-az^{-1})$:

$$\begin{aligned}
y(k+2) &= \varepsilon(k+2) + \frac{(\beta-a)(1+az^{-1}-az^{-1})\varepsilon(k+1)}{1+az^{-1}} = \\
&= \varepsilon(k+2) + \frac{(\beta-a)(1+az^{-1})}{1+az^{-1}} \varepsilon(k+1) - \frac{a(\beta-a)\varepsilon(k)}{1+az^{-1}} = \\
&= \varepsilon(k+2) + (\beta-a)\varepsilon(k+1) - \frac{a(\beta-a)}{1+az^{-1}} \varepsilon(k), \tag{8.16}
\end{aligned}$$

де $\varepsilon(k)$ можна обчислити за спостереженнями:

$$\varepsilon(k) = \frac{1+az^{-1}}{1+\beta z^{-1}} y(k).$$

Якщо підставити цей вираз в (8.16), то отримаємо наступне рівняння:

$$\begin{aligned}
y(k+2) &= \varepsilon(k+2) + (\beta-a)\varepsilon(k+1) - \frac{a(\beta-a)}{1+az^{-1}} \varepsilon(k) = \\
&= \varepsilon(k+2) + (\beta-a)\varepsilon(k+1) - \frac{a(\beta-a)}{1+\beta z^{-1}} y(k). \tag{8.17}
\end{aligned}$$

Нехай $y(k+2, k)$ – прогноз на два кроки на основі інформації, що є в наявності на момент k , включно. Запишемо дисперсію похибки прогнозу за допомогою рівняння (8.17):

$$E[y(k+2) - y(k+2, k)]^2 = E[\varepsilon^2(k+2)] + (\beta - a)^2 E[\varepsilon^2(k+1)] + E\left[y(k+2, k) + \frac{a(\beta - a)}{1 + \beta z^{-1}} y(k)\right]^2. \quad (8.18)$$

Таким чином, можна записати, що:

$$E[y(k+2) - y(k+2, k)]^2 \geq 1 + (\beta - a)^2.$$

Із (8.18) випливає, що рівність можлива при умові, що:

$$y(k+2, k) = -\frac{a(\beta - a)}{1 + \beta z^{-1}} y(k).$$

Тобто функція двокрокового прогнозу задовольняє такому різницевому рівнянню:

$$y(k+2, k) = -\beta y(k+1, k) - a(\beta - a)y(k).$$

Знайдемо похибку двокрокового прогнозу:

$$\begin{aligned} f(k+2, k) &= y(k+2) - \hat{y}(k+2, k) = \\ &= \varepsilon(k+2) + (\beta - a)\varepsilon(k+1) - \frac{a(\beta - a)}{1 + az^{-1}}y(k) + \frac{a(\beta - a)}{1 + \beta z^{-1}}y(k) = \\ &= \varepsilon(k+2) + (\beta - a)\varepsilon(k+1), \end{aligned}$$

тобто, це процес КС(2).

8.5 Загальна постановка задачі прогнозування з мінімумом дисперсії

Модель процесу можна подати у вигляді:

$$A(z^{-1})y(k) = B(z^{-1})\varepsilon(k), \quad (8.19)$$

де $A(z^{-1}), B(z^{-1})$ є поліномами стосовно z виду:

$$A(z^{-1}) = 1 + a_1z^{-1} + \dots + a_nz^{-n},$$

$$B(z^{-1}) = 1 + b_1z^{-1} + \dots + b_nz^{-n},$$

$$\{\varepsilon(k)\} \sim N(0,1),$$

Тобто $\varepsilon(k)$ є нормальною розподіленою випадковою величиною з нульовим середнім та одиничною дисперсією.

Необхідно побудувати найкращу, у розумінні мінімуму дисперсії похибки, функцію прогнозування на s кроків за умови, що відомі значення $y(k), y(k-1), y(k-2), \dots$, тобто за умови:

$$E_k [y(k+s) - \hat{y}(k+s)]^2 \rightarrow \min_{\hat{y}},$$

де \hat{y} – оцінка прогнозу.

На основі рівняння (8.19) можна записати:

$$y(k+s) = \frac{B(z^{-1})}{A(z^{-1})} \varepsilon(k+s). \quad (8.20)$$

Права частина цього рівняння є лінійною комбінацією змінних $\varepsilon(k+s), \varepsilon(k+s-1), \varepsilon(k+s-2), \dots, \varepsilon(k+1), \varepsilon(k), \varepsilon(k-1), \varepsilon(k-2), \dots$.

Значення змінних $\varepsilon(k), \varepsilon(k-1), \dots$ можна обчислити за результатами вимірів $y(k), y(k-1), \dots$, а значення $\varepsilon(k+1), \dots, \varepsilon(k+s)$ не залежать від вимірів, тому що ці виміри ще невідомі.

Для того щоб виділити вказані групи змінних, переписемо праву частину (8.20) у вигляді (це допустимо для лінійних систем):

$$\frac{B(z^{-1})}{A(z^{-1})} \varepsilon(k+s) = F(z^{-1}) \varepsilon(k+s) + z^{-s} \frac{G(z^{-1})}{A(z^{-1})} \varepsilon(k+s), \quad (8.21)$$

де $F(z^{-1}), G(z^{-1})$ – поліноми від z^{-1} степені $s-1$ та $n-1$, відповідно.

Тобто:

$$F(z^{-1}) = 1 + f_1 z^{-1} + \dots + f_{s-1} z^{-(s-1)},$$

$$G(z^{-1}) = g_0 + g_1 z^{-1} + \dots + g_{n-1} z^{-(n-1)}.$$

Таке розщеплення є допустимим для лінійних систем, якщо правильно визначити степені z та коефіцієнти поліномів F, G . Для першої складової в правій частині рівняння (8.21) можна записати:

$$F(z^{-1})\varepsilon(k+s) = \varepsilon(k+s) + f_1 \varepsilon(k+s-1) + \dots + f_{s-1} \varepsilon(k+s-s+1)$$

або:

$$F(z^{-1})\varepsilon(k+s) = \varepsilon(k+s) + f_1 \varepsilon(k+s-1) + \dots + f_{s-1} \varepsilon(k+1).$$

Враховуючи, що $z^{-s} \varepsilon(k+s) = \varepsilon(k)$, для чисельника другого члена в правій частині (8.21) отримуємо ковзне середнє порядку $n-1$:

$$g_0 \varepsilon(k) + g_1 \varepsilon(k-1) + \dots + g_{n-1} \varepsilon(k-n+1).$$

Тепер рівняння (8.20) можна записати у вигляді:

$$y(k+s) = F(z^{-1})\varepsilon(k+s) + \frac{G(z^{-1})}{A(z^{-1})}\varepsilon(k). \quad (8.22)$$

Другий член в правій частині цього рівняння є функцією випадкових змінних $\varepsilon(k), \varepsilon(k-1), \dots$, а тому його можна обчислити за результатами спостережень $y(k), y(k-1), \dots$, тобто скориставшись рівнянням (8.19), отримаємо:

$$\varepsilon(k) = \frac{A(z^{-1})}{B(z^{-1})} y(k).$$

Тепер рівняння (8.22) можна звести до рівняння:

$$y(k+s) = F(z^{-1})\varepsilon(k+s) + \frac{G(z^{-1})}{B(z^{-1})} y(k). \quad (8.23)$$

Перший член у правій частині цього рівняння представляє собою лінійну комбінацію змінних $\varepsilon(k+1), \varepsilon(k+2), \dots, \varepsilon(k+s)$, які не залежать від спостережень, а другий – лінійна функція результатів спостережень.

Нехай $\hat{y}(k+s, k)$ – оцінка прогнозу на s кроків. Запишемо похибку цього прогнозу як:

$$f(s) = y(k+s) - \hat{y}(k+s, k),$$

де $y(k+s)$ є спостереження, яке точно описується рівнянням (8.23).

Відніmemo $\hat{y}(k+s, k)$ від обох частин рівняння (8.23), що приводить до виразу:

$$y(k+s) - \hat{y}(k+s, k) = F(z^{-1})\varepsilon(k+s) + \frac{G(z^{-1})}{B(z^{-1})}y(k) - \hat{y}(k+s, k),$$

і знайдемо квадрат похибки прогнозу:

$$\begin{aligned} [y(k+s) - \hat{y}(k+s, k)]^2 &= \left[F(z^{-1})\varepsilon(k+s) + \frac{G(z^{-1})}{B(z^{-1})}y(k) - \hat{y}(k+s, k) \right]^2 = \\ &= \left[F(z^{-1})\varepsilon(k+s) \right]^2 + 2 \left[F(z^{-1})\varepsilon(k+s) \right] \left[\frac{G(z^{-1})}{B(z^{-1})}y(k) - \hat{y}(k+s, k) \right] + \\ &\quad + \left[\frac{G(z^{-1})}{B(z^{-1})}y(k) - \hat{y}(k+s, k) \right]^2. \end{aligned}$$

Тепер запишемо умовне математичне сподівання для правої частини:

$$\begin{aligned} E_k \left[F(z^{-1})\varepsilon(k+s) \right]^2 + 2 E_k \left\{ \left[F(z^{-1})\varepsilon(k+s) \right] \left[\frac{G(z^{-1})}{B(z^{-1})}y(k) - y(k+s, k) \right] \right\} + \\ + E_k \left[\frac{G(z^{-1})}{B(z^{-1})}y(k) - \hat{y}(k+s, k) \right]^2. \end{aligned}$$

Оскільки випадкові величини $\varepsilon(k+s), \varepsilon(k+2), \dots$ не залежать від $y(k), y(k-1), \dots$, то середній член в правій частині дорівнює нулю, а тому:

$$E \left[y(k+s) - y(k+s, k) \right]^2 = E_k \left[F(z^{-1}) \varepsilon(k+s) \right]^2 + E_k \left[\frac{G(z^{-1})}{B(z^{-1})} y(k) - \hat{y}(k+s, k) \right]^2.$$

Якщо знехтувати другим членом у правій частині цієї рівності, то можна записати нерівність:

$$E \left[y(k+s) - \hat{y}(k+s, k) \right]^2 \geq \left[\varepsilon(k+s) + f_1 \varepsilon(k+s-1) + f_2 \varepsilon(k+s-2) + \dots + f_{s-1} \varepsilon(k+1) \right]^2$$

У цій нерівності рівність досягається за умови, що:

$$E_k \left[\frac{G(z^{-1})}{B(z^{-1})} y(k) - \hat{y}(k+s, k) \right]^2 = 0,$$

або:

$$\hat{y}(k+s, k) = \frac{G(z^{-1})}{B(z^{-1})} y(k).$$

Звідси випливає, що оптимальне (за мінімумом дисперсії) значення прогнозу на s кроків можна обчислити за виразом:

$$\begin{aligned} y(k+s, k) + b_1 y(k+s, k-1) + \dots + b_n y(k+s-n, k-n) = \\ = g_0 y(k) + g_1 y(k-1) + \dots + g_{n-1} y(k-n-1). \end{aligned} \quad (8.24)$$

Похибка прогнозування:

$$\begin{aligned} f(k+s, k) = y(k+s) - y(k+s, k) = F(z^{-1})\varepsilon(k+s) = \\ = \varepsilon(k+s) + f_1 \varepsilon(k+s-1) + \dots + f_{s-1} \varepsilon(k+1). \end{aligned} \quad (8.25)$$

Мінімально можливе значення дисперсії похибки прогнозування при даному підході можна визначити так:

$$\text{var}[f(k+s, k)] = 1 + f_1^2 + f_2^2 + \dots + f_{s-1}^2.$$

Для того щоб скористатись формулами (8.24) і (8.25), необхідно знати коефіцієнти поліномів $F(z^{-1})$ і $G(z^{-1})$. Знайти невідомі коефіцієнти можна за допомогою тотожності:

$$B(z^{-1}) = A(z^{-1})F(z^{-1}) + z^{-s} G(z^{-1}). \quad (8.26)$$

Фактично, $F(z^{-1})$ – це частка від відношення $B(z^{-1})/A(z^{-1})$, а поліном $G(z^{-1})$ – залишок від ділення. Коефіцієнти поліномів $F(z^{-1})$ і $G(z^{-1})$ можна визначити тепер прирівнюванням коефіцієнтів при однакових степенях z^{-1} .

Для заданої структури поліномів $A(z^{-1})$, $B(z^{-1})$, $F(z^{-1})$ і $G(z^{-1})$:

$$\begin{aligned} B(z^{-1}) &= b_0 + b_1 z^{-1} + b_2 z^{-2} + \dots + b_n z^{-n}, \\ A(z^{-1}) &= 1 + a_1 z^{-1} + a_2 z^{-2} + \dots + a_n z^{-n}, \\ F(z^{-1}) &= 1 + f_1 z^{-1} + f_2 z^{-2} + \dots + f_{s-1} z^{-(s-1)}, \\ G(z^{-1}) &= g_0 + g_1 z^{-1} + \dots + g_{n-1} z^{-(n-1)}, \\ z^{-s} G(z^{-1}) &= g_0 z^{-s} + g_1 z^{-(s+1)} + g_2 z^{-(s+2)} + \dots + g_{n-1} z^{-(n-1+s)} \end{aligned}$$

отримаємо такі рівності для визначення коефіцієнтів:

$$\begin{aligned} \text{при } z^{-1}: \quad b_1 &= a_1 + f_1, \\ \text{при } z^{-2}: \quad b_2 &= a_2 + a_1 f_1 + f_2, \\ \text{при } z^{-(s-1)}: \quad b_{s-1} &= a_{s-1} + a_{s-2} f_1 + a_{s-3} f_2 + \dots + a_1 f_{s-2} + f_{s-1}, \\ \text{при } z^{-s}: \quad b_s &= a_k + a_{k-1} f_1 + a_{k-2} f_2 + \dots + a_1 f_{s-1} + g_0, \\ \text{при } z^{-(s+1)}: \quad b_{s+1} &= a_{k+1} + a_k f_1 + a_{k-1} f_2 + \dots + a_2 f_{s-1} + g_1, \\ &\quad \vdots \\ \text{при } z^{-n}: \quad b_n &= a_n + a_{n-1} f_1 + a_{n-2} f_2 + \dots + a_{n-s-1} f_{s-1} + g_{n-s}, \\ \text{при } z^{-(n+1)}: \quad 0 &= a_n f_1 + a_{n-1} f_2 + \dots + a_{n-s+2} f_{s-1} + g_{n-s+1}, \\ &\quad \vdots \\ \text{при } z^{-(n+s-1)}: \quad 0 &= a_n f_{s-1} + g_{n-1}. \end{aligned} \tag{8.27}$$

Таким чином, коефіцієнти f_i , g_i можна обчислити за допомогою простих рекурсивних співвідношень при відомих $A(z^{-1})$ і $B(z^{-1})$, а також вибраній кількості кроків s для прогнозування. Очевидно, що це число не може бути великим.

Для даної постановки задачі кращою є лінійна оцінка прогнозу і результат прогнозування не залежить критично від вибору критерію мінімізації похибки прогнозу.

Якщо послідовність $\{y(k)\}$ має нормальний розподіл, то функція прогнозування не зміниться, якщо вибрати за критерій мінімізації:

$$E w[y(k+s) - \hat{y}(k+s, k)]^2,$$

де w – довільна симетрична функція.

Вирішальну роль для справедливості наведеного вище виведення функції прогнозування відіграє той факт, що $\varepsilon(k)$ і $\varepsilon(l)$ є взаємно незалежними, якщо $k \neq l$.

Якщо випадкові змінні $\varepsilon(k)$ і $\varepsilon(l)$ є залежними одна від одної, то математичне сподівання добутку $\varepsilon(k+l)$ на довільну функцію від $y(k), y(k-1), \dots$ не обов'язково буде дорівнювати нулю при $l > 0$.

8.6 Приклад побудови функцій прогнозування

Приклад 8.2. Для стохастичного процесу $\{y(k)\}$, який визначається рівнянням:

$$y(k) = 1,5y(k-1) + 0,5y(k-2) = 2[\varepsilon(k) - 1,2\varepsilon(k-1) + 0,6\varepsilon(k-2)],$$

де $\{\varepsilon(k)\} \sim N(0, 1)$, запишемо функцію прогнозування на три кроки, $s = 3$, яка мінімізує середньоквадратичну похибку прогнозу.

Для даного прикладу поліноми $A(z^{-1})$ і $B(z^{-1})$ мають вигляд:

$$A(z^{-1}) = 1 - 1,5z^{-1} + 0,5z^{-2},$$

$$B(z^{-1}) = 1 - 1,2z^{-1} + 0,6z^{-2}.$$

Використовуючи співвідношення (8.27), можна визначити коефіцієнти функції прогнозування:

$$b_1 = a_1 + f_1$$

або:

$$f_1 = b_1 - a_1 = -1,2 + 1,5 = 0,3;$$

$$b_2 = a_2 + a_1 f_1 + f_2$$

або:

$$f_2 = b_2 - a_2 - a_1 f_1 = 0,6 - 0,5 + 1,5 \cdot 0,3 = 0,55;$$

Таким чином, $F(z^{-1}) = 1 + 0,3z^{-1} + 0,55z^{-2}$.

Використовуючи тотожність (8.26), запишемо:

$$1 - 1,2z^{-1} + 0,6z^{-2} = (1 - 1,5z^{-1} + 0,5z^{-2}) \cdot (1 + 0,3z^{-1} + 0,55z^{-2}) + z^{-3}(g_0 + g_1z^{-1}).$$

Звідси визначимо, що

$$G(z^{-1}) = 0,675 + 0,275z^{-1},$$

тобто, всі необхідні коефіцієнти для функції прогнозування знайдено.

Отримаємо функції прогнозування. Запишемо функції прогнозування динаміки процесу на довільне число s кроків, отримані в даному розділі.

Для рівняння першого порядку без розв'язку:

$$E_k[y(k+s)] = \hat{y}(k+s, k) = a_0 \sum_{i=0}^{s-1} a_1^i + a_1^s y(k).$$

Для рівняння AP(2) без розв'язку:

$$E_k[y(k+s)] = y(k+s, k) = a_0 + a_1 E_k[y(k+s-1)] + a_2 E_k[y(k+s-2)]$$

на основі повного розв'язку рівняння АРКС(1,1):

$$E_k[y(k+s)] = \frac{a_0}{1-a_1} (1-a_1^s) + \beta_1 a_1^{s-1} \varepsilon(k) + a_1^s y(k) ;$$

на основі повного розв'язку рівняння AP(1):

$$E_k[y(k+s)] = \frac{a_0}{1-a_1} (1-a_1^s) + a_1^s y(k) ;$$

функція прогнозування з мінімальною дисперсією:

$$y(k+s, k) = \frac{G(z^{-1})}{B(z^{-1})} y(k)$$

8.7 Схема створення СППР для прогнозування часових рядів

Для реалізації СППР при прогнозуванні динаміки часових рядів вибирається технологія на основі застосування методів структуризації задач, методів попередньої оброблення даних, математичних і статистичних моделей процесів, множини методів оцінювання моделей і множини критеріїв визначення якості прогнозів. Застосування такого підходу забезпечує отримання високої якості прогнозів та прийнятих рішень, які на них ґрунтуються.

Функціональна схема процесу аналізу даних та прогнозування в СППР на основі часових рядів представлена на рис. 8.1.

8.8 Алгоритм процесу аналізу та прогнозування

Розглянемо алгоритм процесу аналізу та прогнозування на основі часових рядів при підтримці прийняття рішень за схемою на рис. 8.1. Алгоритм складається із послідовності кроків:

Крок 1. Попереднє оброблення та аналіз даних.

Крок 2. Перевірка наявності нелінійностей.

Крок 3. Визначення порядку не лінійності Перехід на крок 9.

Крок 4. Перевірка процесу на стаціонарність Якщо процес стаціонарний, переходимо до кроку 9, інакше до кроку 5.

Крок 5. Перевірки наявності гетероскедастичності Якщо процес гетероскедастичний, то переходимо до пункту 6, якщо містить тренд, то до кроку 7, інакше до кроку 8.

Крок 6. Визначення типу моделі для опису гетероскедастичності Після вибору найбільш підходящої моделі переходимо до кроку 9.

Крок 7. Визначення способу вилучення або моделювання тренду.
Перейти до кроку 9.

Крок 8. Для коінтегрованих процесів будується модель корекції похибок і переходимо до кроку 10.

Крок 9. Будується модель часового ряду та обчислюються критерії адекватності отриманої моделі і переходимо до кроку 10.

Крок 10. На основі обраної моделі, будується функція прогнозування. Обчислюється прогноз поведінки ряду та визначаються оцінки точності прогнозу.

Крок 11. Якщо точність прогнозу не задовольняє особу, що приймає рішення, то уточнюються вхідні дані і переходимо до кроку 1. Інакше, процес закінчується.

8.8.1 Попереднє оброблення та аналіз даних

Крок 1. Попереднє оброблення та аналіз даних, що включає в себе:

- заповнення пропусків;
- згладжування екстремальних значень;
- логарифмування;
- нормування в діапазоні від -1 до $+1$;
- диференціювання (можливості обчислення перших різниць та різниць вищих порядків);
- пряме і зворотне перетворення Фур'є;
- цифрова фільтрація;
- бутстреп (розмноження вибірки даних).

Перехід на крок 2.

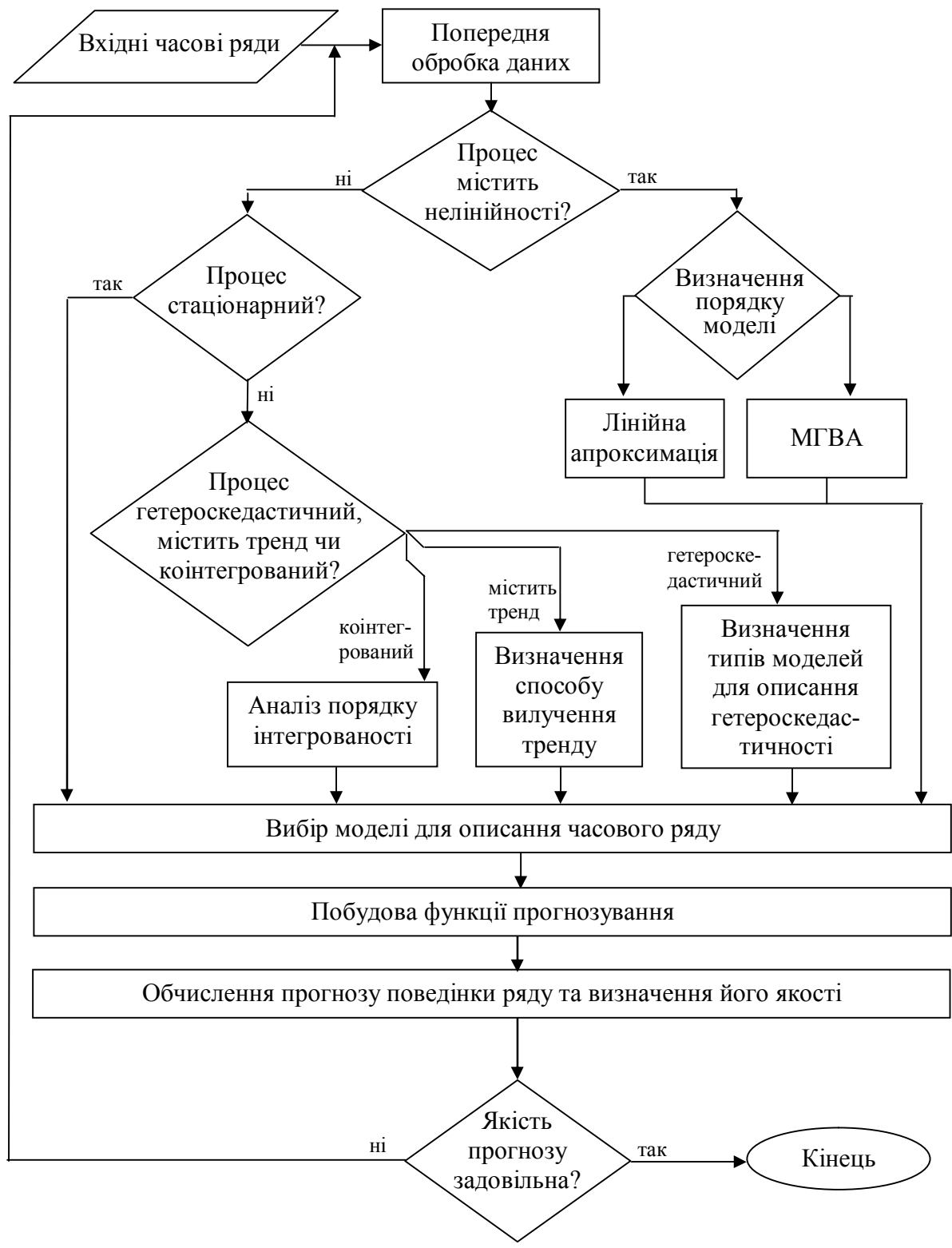


Рис. 8.1 Функціональна схема процесу прогнозування на основі часових рядів

8.8.2 Перевірка наявності нелінійностей

Крок 2. Перевірка наявності нелінійностей визначається за допомогою тесту Фішера або кореляційних функцій вищих порядків. Якщо процес містить нелінійності, то переходимо до кроку 3, інакше до кроку 4.

Для розв'язання цієї задачі можна користуватися різними критеріями. Однак при цьому необхідно знати про їх можливості. Покажемо на простому прикладі, що застосування лінійних коваріаційних функцій не завжди призводить до позитивних результатів.

Нехай при визначенні структури моделі не були обчислені деякі пояснюючі змінні; у результаті корельовані залишки описуються таким рівнянням:

$$\xi(k) = cu(k-1)e(k-1) + e(k), \quad (8.28)$$

$$E[e(k)] = 0, \quad E[u(k)] = 0, \quad E[e(k)u(k)] = 0,$$

де $e(k)$ – білий гаусівський шум;

c – масштабний коефіцієнт.

Тобто змінні $e(k)$ і $u(k)$ некорельовані і мають нульове середнє.

Можна показати, що нормована автокореляційна функція залишків із нормованої функції взаємної кореляції між вхідним сигналом $u(k)$ і залишками мають вигляд: $\Phi_{\xi\xi}(\tau) = \delta(\tau)$, $\Phi_{u\xi}(\tau) = 0$, $\forall \tau$.

Проте, із рівняння (8.28) випливає, що $\xi(k)$ – корельована послідовність, що буде вносити зміщення в оцінки параметрів моделі. Таким чином, у загальному випадку лінійні кореляційні методи не дають

можливості встановити факт наявності нелінійних ефектів та ступінь їх впливу на процес.

Для того щоб оцінити тип зв'язку між входом і виходом (тобто, зв'язок лінійна або нелінійна) можна скористатися спектральною функцією високого порядку вигляду:

$$X_{ij} = \frac{|S_{\omega}(\omega_i, \omega_j)|^2}{S_{\omega}(\omega_i)S_{\omega}(\omega_j)S_{\omega}(\omega_i/\omega_j)},$$

де $S_{\omega}(\omega_i, \omega_j)$ – біспектральна щільність потужності;

$S_{\omega}(\omega_i)$ – спектральна щільність потужності часового ряду.

При $S_{\omega}(\omega_i, \omega_j) = 0, \forall \omega_i, \omega_j$ процес буде лінійним і третій момент вхідного сигналу $\mu_3 = 0$.

Проте, якщо $X_{ij} = const$, то процес лінійний, але $\mu_3 \neq 0$.

Такий підхід до встановлення наявності нелінійностей має два недоліки. По-перше, оцінювання спектральної щільності потужності потребує застосування спеціального попередньої оброблення сигналів у вигляді застосування часових вікон, усереднення, цифрової фільтрації тощо. По-друге, він не завжди може бути використаний при розв'язанні задач ідентифікації систем, оскільки він не дає можливості одержати оцінки параметрів моделі в явному вигляді. Крім того, при розв'язанні цих же задач не завжди є можливість одержати виміри вхідного сигналу або ж інформативний вхідний сигнал одержують штучно у вигляді спеціально генерованих послідовностей, що не завжди можна подавати на вхід об'єкта внаслідок особливостей його функціонування.

Що стосується економічних процесів, то в цьому випадку, зазвичай, не можна поставити експеримент із процесом. Тому використовують тільки ті статистичні дані, які можна реально зібрати в процесі дослідження. У загальному випадку при ідентифікації систем використовують три типи сигналів: вхідні, вихідні і збурення. При цьому вхідний керуючий сигнал вважають незалежним від збурення. У результаті виявляється неможливим з'ясувати деякі типи зв'язків.

Можливо використання також дисперсійного методу визначення присутності нелінійностей, який заснований на застосуванні наступної функції:

$$\Psi_{zu}(t_1, t_2) = E_{u(t_2)} [E_{z(t_1)} [z(t_1) | u(t_2)] - E_{z(t_1)} [z(t_1)]]^2,$$

яка обчислюється за допомогою достатньо складаного інтегрального рівняння, якщо відомі відповідні щільності розподілу можливостей сигналів, що не завжди можна визначити.

Крім розглянутих підходів до визначення наявності нелінійностей, при побудові регресійних моделей можна скористатися простішими тестами. Наприклад, статистикою Фішера:

$$\hat{F} = \frac{\frac{1}{k-2} \sum_{i=1}^k \sum_{j=1}^{n_i} n_i (\bar{y}_i - \hat{y}_{ij})^2}{\frac{1}{n-k} \sum_{i=1}^k \sum_{j=1}^{n_i} (y_{ij} - \bar{y}_i)^2},$$

де k – число груп даних;

n_i – кількість вимірів у групі;

\bar{y}_i – групове середнє;

\hat{y}_i – значення, що оцінюють по прямій регресії;

n – загальне число вимірів.

Фактично така статистика є відношенням:

$$\hat{F} = \frac{\text{Відхилення середніх значень від прямої регресії}}{\text{Відхилення значень } y(k) \text{ від групових середніх}}$$

Якщо статистика \hat{F} зі $\nu_1 = k - 2$, $\nu_2 = n - k$ ступенями свободи досягає або перевершує рівень значущості, то гіпотезу про лінійність потрібно відкинути.

8.8.3 Визначення порядку нелінійності

Крок 3. Визначення порядку нелінійності, побудова моделі за МГВА і лінійна апроксимація процесу з метою її порівняння з нелінійною моделлю. Перехід на крок 9.

8.8.4 Перевірка процесу на стаціонарність

Крок 4. Перевірка процесу на стаціонарність за допомогою тесту Дікі-Фуллера. Якщо процес стаціонарний, переходимо до кроку 9, інакше до кроку 5.

Перевірка процесу на стаціонарність виконують за допомогою тесту Дікі – Фуллера.

При визначенні наявності нестационарності (тобто присутності одиничного кореня), пропонується скористатися тестом Дікі-Фуллера [20], суть якого полягає в наступному: для визначення наявності одиничного кореня запропоновано скористатись трьома рівняннями:

$$\Delta y(k) = \gamma y(k-1) + \varepsilon(k), \quad (8.29)$$

$$\Delta y(k) = a_0 + \gamma y(k-1) + \varepsilon(k), \quad (8.30)$$

$$\Delta y(k) = a_0 + \gamma y(k-1) + a_2 k + \varepsilon(k), \quad (8.31)$$

де k – дискретний час;

$\gamma = a_1 - 1$ – коефіцієнт у рівнянні:

$$y(k) = a_0 + a_1 y(k-1) + \varepsilon(k).$$

Різниця між рівняннями (8.29) та (8.30), (8.31) полягає у присутності детермінованих членів a_0 і $a_2 k$ у рівняннях (8.30) і (8.31), відповідно. Рівняння (8.29) представляє собою модель випадкового кроку (або «блукання»), друге включає зміщення у вигляді константи a_0 , а третє включає зміщення та детермінований лінійний часовий тренд.

У всіх трьох рівняннях нас цікавить параметр γ . Якщо $\gamma = 0$, то послідовність $\{y(k)\}$ містить одиничний корінь. Застосування тесту Дікі-Фуллера передбачає оцінювання одного або більше з наведених вище трьох рівнянь за допомогою МНК або ММП з метою отримання оцінки параметра γ та стандартної похибки цієї оцінки. На основі оцінки та її стандартної похибки обчислюється t -статистика, яка порівнюється із значеннями, наведеними в таблицях Дікі-Фуллера. На основі цього порівняння приймається рішення щодо справедливості або відхилення нуль-гіпотези, що $\gamma = 0$.

Нехай для рівняння $y(k) = a_1 y(k-1) + \varepsilon(k)$ на основі 100 спостережень отримана така оцінка параметра $\hat{a}_1 = 0,9459$ із стандартною

похибкою $SE_{a_1} = 0,031$. Очевидно, що оцінювання рівняння $\Delta y(k) = \gamma y(k-1) + \varepsilon(k)$ приведе до оцінки $\hat{\gamma} = -0,0541$ з тією ж стандартною похибкою 0,031.

Таким чином, для нуль-гіпотези $\gamma_0 = 0$ відповідна t -статистика буде дорівнювати: $t = -0,0541/0,031 = -1,74516$.

Із таблиць Дікі-Фуллера для випадку $a_0 = a_2 = 0$ при $N = 100$ знаходимо, що критичні значення t -статистики дорівнюють $-1,61$; $-1,95$ та $-2,60$ на рівнях значущості 10%, 5% та 1%, відповідно. Таким чином, в розглянутому гіпотетичному випадку при $\hat{\gamma} = -0,0541$ нуль-гіпотеза щодо $\gamma = 0$ (тобто, одиничний корінь присутній) не може бути відхилена при рівнях значимості 5% та 1%, але вона відхиляється на рівні значущості 10%.

Як і у більшості інших випадків перевірки гіпотез, для будь-якого рівня значущості, критичні значення t -статистики зменшуються при збільшенні розміру вибірки.

Методика тестування на наявність одиничного кореня залишається незмінною для всіх трьох рівнянь (8.29)-(8.31). Однак, критичні значення t -статистики залежать від структури моделі, тобто від того чи наявні у моделі зміщення α_0 та детермінований тренд $\alpha_2 k$. Автори методики визначили, що критичні значення для $\gamma = 0$ залежать від структури регресійного рівняння та від довжини вибірки. Так, для рівняння (8.29) використовується статистика, що позначається через τ , для рівняння (8.30) – статистика позначається τ_μ , а для рівняння (8.31) – через τ_τ .

Якщо у модель включити константу (зміщення), але $a_2 = 0$, то необхідно користуватись іншою частиною таблиці критичних значень для t -статистики. Оцінюючи рівняння для гіпотетичного прикладу у формі:

$\Delta y(k) = a_0 + \gamma y(k-1) + \varepsilon(k)$, знайдемо, що $\gamma = 0,9135 - 1 = -0,0865$ із стандартною похибкою 0,041. Таким чином, отримаємо таке значення t -статистики: $\tau_\mu = -0,0865/0,041 = -2,1098$.

З таблиць знову знаходимо, що для 100 спостережень критичні значення дорівнюють $-2,58$; $-2,89$ та $-3,51$ на рівнях значущості 10%, 5% та 1%, відповідно. Таким чином, нуль-гіпотеза щодо наявності одиничного кореня ($\gamma = 0$) не може бути відхилена при всіх рівнях значущості, закладених в таблицю Дікі-Фуллера.

Якщо ж скористатись структурою моделі у вигляді: $\Delta y(k) = a_0 + a_1 y(k-1) + a_2 k + \varepsilon(k)$, то знайдемо, що критичні значення статистики τ_τ дорівнюють $-3,45$ та $-4,04$ на рівнях значущості 5% і 1%, відповідно.

Критичні значення не зміняться, якщо рівняння (8.29), (8.30) і (8.31) замінити такими рівняннями авторегресії:

$$\Delta y(k) = \gamma y(k-1) + \sum_{i=2}^p \beta_i \Delta y(k-i+1) + \varepsilon(k), \quad (8.32)$$

$$\Delta y(k) = a_0 + \gamma y(k-1) + \sum_{i=2}^p \beta_i \Delta y(k-i+1) + \varepsilon(k), \quad (8.33)$$

$$\Delta y(k) = a_0 + \gamma y(k-1) + a_2 k + \sum_{i=2}^p \beta_i \Delta y(k-i+1) + \varepsilon(k). \quad (8.34)$$

Тобто при використанні цих рівнянь для тестування нуль-гіпотези $\gamma = 0$ використовують ті ж статистики: τ , τ_μ і τ_τ . Для перевірки об'єднаних гіпотез щодо коефіцієнтів Дікі і Фулер запропонували ще три F -статистики: ϕ_1 , ϕ_2 і ϕ_3 .

Так, статистика ϕ_1 використовується для перевірки нуль-гіпотези щодо $\gamma = a_0 = 0$ в рівняннях (8.30) і (8.33). Статистика ϕ_2 використовується для перевірки об'єднаної гіпотези $a_0 = \gamma = a_2 = 0$ в рівняннях (8.31) і (8.34), а статистика ϕ_3 – для перевірки об'єднаної гіпотези $\gamma = a_2 = 0$.

Статистики ϕ_1, ϕ_2 і ϕ_3 обчислюються по аналогії із звичайними F – статистиками:

$$\phi_i = \frac{[RSS_1 - RSS_2]r}{RSS_2(N - n)},$$

де RSS_1 і RSS_2 – суми квадратів похибок (СКП), обчислених для моделей з обмеженнями та моделей без обмежень;

r – число обмежень;

N – число використаних спостережень;

n – число параметрів, оцінених для необмеженої моделі.

Можливими обмеженнями можуть бути обмеження на порядок моделі та їх структуру.

Обчислення значення ϕ_i та його порівняння із відповідним значенням, запропонованим Дікі і Фулером, дозволяє визначити рівень значущості, на якому обмеження на модель відіграють суттєву роль. При цьому за нуль-гіпотезу приймають те, що дані генеруються моделлю з обмеженнями, а за альтернативну – що дані генеруються моделлю без обмежень.

Якщо обмеження не відіграє суттєвої ролі, то сума квадратів похибок для моделі з обмеженнями буде близькою до СКП для моделі без

обмежень. Відповідно, ϕ_i буде мати при цьому невелике значення, тобто, при великих значеннях ϕ_i обмеження відіграють суттєву роль і нуль-гіпотеза відхиляється.

Таким чином, якщо розраховане значення ϕ_i є меншим ніж відповідна статистика Дікі і Фуллера, то приймається модель з обмеженнями (тобто, приймається нуль-гіпотеза). Якщо ж розраховане значення ϕ_i є більшим ніж відповідна статистика Дікі і Фуллера, то нуль-гіпотеза відхиляється і приймається альтернативна, тобто, що обмеження грають суттєву роль і модель повинна бути без обмежень.

Необхідно також виконати перевірку гіпотез щодо значущості константи a_0 та коефіцієнта a_2 , значущість якого означає наявність часового тренду. Якщо нуль-гіпотеза визначена як $\gamma = 0$, то тестування на наявність часового тренду виконується за допомогою статистики $\tau_{\beta\tau}$. Таким чином, ця статистика виконує перевірку чи $a_2 = 0$ при $\gamma = 0$. Для перевірки гіпотези стосовно $a_0 = 0$, необхідно скористатись статистикою $\tau_{\alpha\tau}$, якщо оцінюється модель (8.34), або статистикою $\tau_{\alpha\mu}$, якщо оцінюється модель (8.33).

8.8.5 Перевірки наявності гетероскедастичності

Крок 5. Для перевірки наявності гетероскедастичності застосовуються тести: Уайта, Бройша-Пагана/Годфрі, Голдфельда-Квандта. Наявність тренду визначається за допомогою тесту Дікі-Фуллера. Аналіз коінтегрованості процесів виконується за методикою Інгла-Грейнджера або Йохансена. Якщо процес гетероскедастичний, то переходимо до пункту 6, якщо містить тренд, то до кроку 7, інакше до кроку 8.

8.8.5.1 Тест Уайта

У відповідності до цього тесту (фактично ми ним вже користувались в попередньому параграфі), для виявлення наявності гетероскедастичності необхідно побудувати допоміжну модель регресії для квадратів залишків, які генеруються у результаті застосування звичайного методу найменших квадратів (МНК) до часових рядів. Регресія квадратів залишків містить у собі (в правій частині) константу, а також всі ненадлишкові регресори на множині всіх регресорів, яка включає самі регресори, їхні квадрати та взаємні добутки. Наприклад, нехай будується початкова регресія у вигляді:

$$y(k) = a_0 + a_1x_1(k) + a_2x_2(k) + \varepsilon(k),$$

Тобто вектор вимірів незалежних змінних має вигляд: $[1 \ x_1 \ x_2]^T$. У даному випадку всього є дев'ять можливих змінних, але 1 в квадраті залишається одиницею, а взаємні добутки регресорів на 1 також нічого не змінюють. Тому множина всіх ненадлишкових змінних, яка складається із регресорів, їхніх квадратів та взаємних добутків, має вигляд: $[1 \ x_1 \ x_2 \ x_1^2 \ x_2^2 \ x_1x_2]$.

Якщо висунути гіпотезу про існування гетероскедастичності, то добуток NR^2 буде мати в асимптотиці розподіл хі-квадрат $\chi^2(5)$, де 5 означає кількість регресорів без константи; R^2 – коефіцієнт множинної детермінації. У загальному випадку можна записати, що $NR^2 \leftrightarrow \chi^2(q)$. Тобто добуток NR^2 приблизно має розподіл хі-квадрат при використанні в регресії q регресорів (константа не враховується). Цей тест

дає можливість виявити наявність гетероскедастичності, але не вказує на її форму і, як наслідок, на тип алгоритму оцінювання параметрів, який необхідно використовувати. Використання тесту Уайта передбачає використання МНК для оцінювання параметрів початкової моделі.

Ще однією проблемою, пов'язаною із використанням цього тесту, є те, що кількість ступенів свободи в розподілі χ^2 може приймати велике значення, що знижує якість тестування. Наприклад, нехай початкова регресійна модель має r регресорів (разом із константою). Тоді в загальному випадку $q = [r(r+1)/2] - 1$ [20, 53]. Якщо $r = 10$, то $q = 54$. Якщо серед регресорів є фіктивні змінні (змінні, що вводяться у регресію з метою виключення із алгоритму оцінювання частини вимірів), то кількість ступенів свободи буде дещо меншою. Іноді значення q зменшують шляхом введення в регресію квадратів регресорів, але виключають взаємні добутки.

8.8.5.2 Тест Бройша-Пагана/Годфрі

Розглянемо лінійну регресію

$$y(k) = \mathbf{x}^T(k)\boldsymbol{\beta} + \varepsilon(k), \quad (8.35)$$

де $\mathbf{x}^T(k) = [1 \ x_2(k) \ x_3(k) \ \dots \ x_r(k)]$.

Припустимо, що гетероскедастичність має таку форму:

$$E[\varepsilon(k)] = 0, \quad \forall k,$$

$$\text{var}[\varepsilon(k)] = E[\varepsilon^2(k)] = \sigma_\varepsilon^2 = h(\alpha \mathbf{z}^T(k)), \quad (8.36)$$

де $\mathbf{z}^T(k) = [1 \ z_2(k) \ z_3(k) \ \dots \ z_p(k)]$ – вектор відомих змінних;

$\alpha = [\alpha_1 \ \alpha_2 \ \dots \ \alpha_p]$ – вектор невідомих коефіцієнтів;

$h(\cdot)$ – деяка невизначена функція, яка може приймати тільки додатні значення.

Нуль-гіпотеза щодо гомоскедастичності формулюється так:

$$H_0 : \alpha_2 = \alpha_3 = \dots = \alpha_p = 0.$$

А це означає, що $\sigma_\varepsilon^2 = h(\alpha_1) = \text{const}$. При такій нуль-гіпотезі можна оцінювати коефіцієнти моделі (8.35) за допомогою МНК у припущенні про нормальність розподілу збурень у правій частині рівняння. Процедура застосування тесту на гетероскедастичність в даному випадку досить проста і її можна представити у вигляді таких кроків:

1. Оцінити параметри початкового рівняння (8.35) за допомогою звичайного МНК, сформувати масив залишків $e(k) = y(k) - \mathbf{x}^T(k)\beta$ та обчислити дисперсію $\sigma_\varepsilon^2 = \sigma_e^2 = N^{-1} \sum e^2(k)$.

2. Знайти оцінки регресії $e^2(k)/\sigma_e^2$ на $z(k)$ за допомогою ЗМНК та обчислити значення похибки ESS (explained sum of squares) за виразом $ESS = \beta^T \mathbf{X}^T \mathbf{X} \beta - N\mu_y^2$, де μ_y – середнє значення послідовності $\{y(k)\}$.

3. При прийнятій нуль-гіпотезі $H_0 \frac{1}{2}ESS \leftrightarrow \chi^2(p-1)$.

Таким чином, гіпотеза стосовно гомоскедастичності відхиляється, якщо $ESS / 2$ перевищує вибране критичне значення із розподілу χ^2 .

4. Простішим, але асимптотично еквівалентним підходом, є оцінювання регресії $e^2(k)$ на $z(k)$. Величина NR^2 , обчислена для цієї регресії, буде мати в асимптотиці розподіл $\chi^2(p-1)$ при прийнятій нуль-гіпотезі.

Нагадаємо, що зв'язок між різними видами похибок регресійної моделі визначається за виразом: $(\mathbf{y}^T \mathbf{y} - N\mu_y^2) = (\beta^T \mathbf{X}^T \mathbf{X} \beta - N\mu_y^2) + \mathbf{e}^T \mathbf{e}$, або $TSS = ESS + RSS$, де TSS – загальна похибка регресії (total sum of squares); RSS – сума квадратів похибок моделі (residual (unexplained) sum of squares).

Для застосування цього тесту необхідно знати змінні z , які спричиняють гетероскедастичність, але немає необхідності знати функціональну форму гетероскедастичності. Ця інформація може бути невідомою. На практиці кандидати в змінні z можуть бути вибрані з вектора регресорів $x(k)$. У такому випадку послідовність використання даного тесту така ж як і тесту Уайта.

8.8.5.3 Тест Голдфелда-Квандта

Цей спрощений тест застосовують у тих випадках, коли є одна змінна (зазвичай, з числа регресорів), що приводить до гетероскедастичності. Припустимо, наприклад, що σ_ε^2 додатно корельована з i – м регресором x_i .

Тоді процедура тестування складається із наведених нижче кроків:

1. Упорядкувати (за зростанням чи зменшенням) масив значень регресора x_i .

2. Виключити із аналізу c середніх значень змінної.

3. Побудувати окремо регресії для перших та останніх $(N - c)/2$ значень при умові, що їх достатньо для оцінювання заданого числа параметрів моделі.

4. Обчислити значення похибок RSS_1 і RSS_2 , де індекс «1» відноситься до регресії, яка оцінювалась за меншими значеннями x_i , а «2» відноситься до регресії, яка оцінювалась за більшими значеннями x_i . Тоді відношення $R = \frac{RSS_2}{RSS_1}$ буде мати (в припущенні існування гетероскедастичності) F -розподіл із $[(N - c - 2r)/2, (N - c - 2r)/2]$ ступенями свободи.

При висуванні альтернативної гіпотези F матиме велике значення.

Таким чином, якщо $R < F_{0,95}$, то гіпотеза про існування гетероскедастичності буде відхилена на рівні 5%.

Якість тестування при використанні даного тесту залежить (окрім інших факторів) від кількості середніх спостережень, які виключаються із аналізу. Так, при великому значенні c якість тестування буде низькою, оскільки RSS_1 і RSS_2 матимуть невелику кількість ступенів свободи. Однак, якщо значення c невелике, то якість тестування також буде низькою, оскільки зменшуватиметься різниця між RSS_1 і RSS_2 . Рекомендується вибирати значення $c \approx N/3$.

8.8.6 Визначення типу моделі для опису гетероскедастичності

Крок 6. Визначення типу моделі для опису гетероскедастичності: УАРУГ(p, q), експоненційна УАРУГ(p, q), УАРУГ-М або інша. Після вибору найбільш підходящої моделі переходимо до кроку 9.

Основними є такі моделі гетероскедастичних процесів:

1. Рівняння для умовної дисперсії квадратів залишків першого порядку (АРУГ(1)):

$$h^2(k) = \beta_0 + \beta_1 \varepsilon^2(k-1) + \varepsilon_1(k),$$

де $h^2(k)$ - умовна дисперсія процесу;

$\varepsilon^2(k)$ - квадрат залишків;

$\varepsilon_1(k)$ – похибка моделі в момент часу k .

2. Авторегресійна умовно гетероскедастична модель порядку p (АРУГ(p)):

$$h^2(k) = \beta_0 + \sum_{i=1}^p \beta_i \varepsilon^2(k-i) + \varepsilon_1(k).$$

3. Узагальнена авторегресійна умовно гетероскедастична модель (УАРУГ(p, q)) :

$$h^2(k) = \beta_0 + \sum_{i=1}^p \beta_i \varepsilon^2(k-i) + \sum_{i=1}^q \alpha_i h^2(k-i) + \varepsilon_1(k).$$

Це рівняння показує, що поточне значення умовної дисперсії є функцією від константи, деяких значень квадратів залишків з рівняння умовного середнього та деяких значень попередньої умовної дисперсії.

4. Експоненційна авторегресійна умовно гетероскедастична модель (E - УАРУГ(p, q)):

$$\log(h^2(k)) = \alpha_0 + \sum_{i=1}^p \alpha_i \frac{|\varepsilon(k-i)|}{h(k-i)} + \sum_{i=1}^p \gamma_i \frac{\varepsilon(k-i)}{h(k-i)} + \sum_{i=1}^q \beta_i \log(h^2(k-1) + \varepsilon_1(k)).$$

У моделях УАРУГ(p, q) умовна дисперсія залежить від розміру залишків, а не від їх знаків. Наведена модель описує умовну дисперсію як асиметричну функцію значень ε . Це дозволяє додатнім і від'ємним попереднім значенням ε мати різний вплив на волатильність. Подання в логарифмічному вигляді дає можливість включити від'ємні значення залишків, не отримуючи при цьому від'ємну умовну дисперсію.

5. Авторегресійна мовно-гетероскедастична модель модифікована (УАРУГ - М) – моделювання премії за ризик:

$$y(k) = \beta + \gamma h(k) + \varepsilon_1(k),$$

$$h^2(k) = a_0 + a_1 \sum_{i=1}^p \varepsilon^2(k-i) + \sum_{i=1}^q h^2(k-i) + \varepsilon_2(k).$$

У цьому рівнянні умовна середня дисперсія перетворена в середньо-квадратичну так, що вона виражається в тих же одиницях вимірювання, що і премія за ризик.

6. Модель для прогнозування волатильності за допомогою УАРУГ:

$$y(k+1) = \beta_0 + \beta_1 \varepsilon^2(k) + \gamma_1 h^2(k),$$

$$y(k+j) = \beta_0 + (\beta_1 + \gamma_1) h^2(k+j-1).$$

У другому рівнянні ε^2 , значення якого не відоме, замінюється на умовну оцінку h^2 . Таким чином, друге рівняння дозволяє прогнозувати h^2 в момент часу $t + 1$ ($j = 1$), потім в момент часу $t + 2$ ($j = 2$) тощо.

7. Двофакторна модель УАРУГ. Припустимо, що моделі умовних середніх мають такий вигляд:

$$s(k) = \alpha_0 + \alpha_1 s(k-1) + \varepsilon_s(k), \quad f(k) = \beta_0 + \beta_1 f(k-1) + \varepsilon_f(k).$$

Рівняння умовної дисперсії та коваріації мають наступний вигляд:

$$h^2(s(k)) = \alpha_0 + \alpha_1 \varepsilon^2(s(k-1)) + \alpha_2 h^2(s(k-1)),$$

$$h^2(f(k)) = \beta_0 + \beta_1 \varepsilon^2(f(k-1)) + \beta_2 h^2(f(k-1)),$$

$$\text{cov}[s(k), f(k)] = \gamma_0 + \gamma_1 \varepsilon(s(k-1)) + \gamma_2 \text{cov}[s(k-1), f(k-1)].$$

Рівняння корекції похибки, що впливає із наявності коінтеграції:

$$s(k) = \alpha_0 + \alpha_1 [(s(k-1) - \lambda f(k-1)) + \varepsilon_s(k)],$$

$$f(k) = \beta_0 + \beta_1 [(s(k-1) - \lambda f(k-1)) + \varepsilon_f(k)].$$

Коефіцієнт хеджування при використанні двофакторної моделі:

$$\hat{b}^*(k) = \frac{\text{cov}[s(k), f(k)]}{\hat{h}^2(f)}.$$

Перевагою двофакторної моделі УАРУГ з використанням коефіцієнта хеджування є те, що він визначається на основі дисперсій та коваріацій, що змінюються в часі.

8. Статична регресійна модель.

Для того щоб порівняти якість функціонування регресійних УАРУГ моделей, описаних вище, використовується статична регресійна модель, де середнє і дисперсія незмінні у часі. Вона має такий вигляд:

$$fp(k) = a_0 + u(k) u(k)F(k - 1) \sim N(0; \omega^2).$$

Дисперсія ω^2 незмінна й має бути додатною. Для моделювання збурень використовується гаусівська випадкова величина.

8.8.7 Визначення способу вилучення або моделювання тренду

Крок 7. Визначення способу вилучення або моделювання тренду включає в себе визначення порядку інтегрованості процесу або можливість описання тренду поліноміальною функцією, експоненціальною, логарифмічно або іншими функціями. Після вилучення тренду переходимо до кроку 9.

Крок 8. Для коінтегрованих процесів будується модель корекції похибок і переходимо до кроку 10.

8.8.8 Побудова моделі часового ряду

Крок 9. Будується модель часового ряду та обчислюються критерії адекватності отриманої моделі і переходимо до кроку 10.

Критерії адекватності моделі дозволяють оцінити окремо значущість коефіцієнтів математичної моделі в статистичному сенсі, визначити інтегральну похибку моделі стосовно вихідного часового ряду, встановити наявність кореляції між значеннями похибки моделі (нагадуємо, що вони повинні бути не корельованими), а також визначити ступінь адекватності моделі фізичному процесу в цілому. У цю множину входять такі статистичні параметри:

1. t – статистика Стьюдента.
2. Коефіцієнт детермінації R^2 .
3. Сума квадратів похибок моделі.
4. Інформаційний критерій Акайке.
5. Статистика Дарбіна-Уотсона.
6. Статистика Фішера F .
7. Коефіцієнт Тейла.

8.8.8.1 t - статистика Стьюдента

Значимість кожного коефіцієнта регресії в статистичному сенсі визначають за допомогою t – статистики, що, зазвичай, обчислюється всіма пакетами статистичної оброблення даних за формулою:

$$t_a = \frac{\hat{a} - a_0}{SE_a},$$

де \hat{a} – оцінка коефіцієнту, яка отримана за допомогою пакета;

a_0 – нуль-гіпотеза у відношенні значення цього коефіцієнту (звичайно $a_0 = 0$);

SE_a – стандартна похибка оцінки коефіцієнта, що обчислюється пакетом.

Очевидно, що чим менше значення стандартної похибки, тим кращою є оцінка коефіцієнта для моделі.

Для визначення значущості оцінки коефіцієнта необхідно знати довжину вибірки N , кількість оцінюваних параметрів p і задатися рівнем значущості α (зазвичай задаються $\alpha = 1\%$, $\alpha = 5\%$ або $\alpha = 10\%$). Рівень значущості, рівний 5% , означає, що при оцінюванні регресії ми допускаємо, що помилкове прийняття рішення стосовно значущості оцінок можливо у 5% випадків. Ці параметри дозволяють вибрати по таблицях значення $t_{крит}$. Якщо $-t_{крит} < t_a < t_{крит}$, то нуль-гіпотеза про незначущість коефіцієнта приймається; у протилежному випадку вона відхиляється і коефіцієнт вважається значущим. Оскільки значення статистики t_a обернено пропорційне стандартній похибці SE_a , то для більшого значення t_a , буде вищою значущість конкретного коефіцієнта.

8.8.8.2 Коефіцієнт детермінації R^2

За міру інформативності часового ряду часто використовують його дисперсію. Коефіцієнт R^2 – це відношення дисперсії тієї частини часового ряду основної змінної, що описується отриманим рівнянням, до вибіркової

дисперсії цієї змінної. Він обчислюється за формулою:
$$R^2 = \frac{\text{var}(\hat{y})}{\text{var}(y)}$$

Очевидно, що для адекватної моделі коефіцієнт детермінації повинен прямувати до одиниці, тобто: $R^2 \rightarrow 1$.

8.8.8.3 Сума квадратів похибок моделі

Сума квадратів похибок моделі $\sum e^2(k)$, тобто:

$$SSE = \sum_{k=1}^N [\hat{y}(k) - y(k)]^2,$$

де $\hat{y}(k) = \hat{a}_0 + \hat{a}_1 \hat{y}(k-1) + \hat{a}_2 \hat{y}(k-2) + \hat{b}_1 x(k) + b_2 z(k)$;

$y(k)$ – вимірювання;

N – довжина вибірки.

Очевидно, що з можливих кандидатів необхідно вибирати ту модель, для якої $\sum e^2(k)$ приймає мінімальне значення.

8.8.8.4 Інформаційний критерій Акайке

Інформаційний критерій Акайке (AIC) враховує суму квадратів похибок, кількість вимірів N і кількість оцінюваних параметрів моделі p :

$$AIC = N \ln \left[\sum_{k=1}^N e^2(k) \right] + 2p.$$

Очевидно, що для кращої моделі критерій має менше значення, оскільки він залежить від суми квадратів похибок (СКП). Проте, крім СКП, даний критерій враховує довжину вибірки і кількість оцінюваних параметрів, що робить його інформативнішим.

8.8.8.5 Статистика Дарбіна-Уотсона

Статистика Дарбіна-Уотсона (Durbin-Watson) обчислюється за формулою:

$$DW = 2 - 2\rho,$$

де ρ – коефіцієнт кореляції між сусідніми значеннями випадкової змінної $\varepsilon(k) \approx e(k)$.

Тобто:

$$\rho = \text{cov}[e(k)] = E[e(k)e(k-1)].$$

Цей параметр дозволяє визначити ступінь корельованості похибок моделі. При повній відсутності кореляції між похибками $DW = 2$, це найбільше прийнятне значення даного параметра.

8.8.8.6 Статистика Фішера

Статистика Фішера F визначає ступінь адекватності моделі в цілому. Для адекватної моделі виконується умова:

$$F > F_{\text{крит}},$$

де $F_{\text{крит}}$ визначається по таблиці аналогічно t – статистиці.

Значення F пропорційно $R^2/(1-R^2)$, де R^2 – коефіцієнт детермінації. Таким чином, більшому значенню F відповідає модель вищого ступеня адекватності.

8.8.8.7 Коефіцієнт Тейла

Коефіцієнт Тейла є дуже важливим індикатором точності моделі та її сумісності:

$$U = \frac{\sqrt{\frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2}}{\sqrt{\frac{1}{N} \sum_{i=1}^N (y_i)^2 + \frac{1}{N} \sum_{i=1}^N (\hat{y}_i)^2}}.$$

За побудовою, його величина знаходиться між 0 і 1. Якщо $U=1$, модель не може бути використана для прогнозу. Прогнозовані, на основі отриманої моделі, і реальні ряди некорельовані. У протилежному випадку, якщо $U=0$, прогнозовані ряди співпадають з реальними рядами і модель ідеально описує дані.

Цей коефіцієнт може бути розкладений на суму відношення упередженості U^M , відношення варіацій U^S і відношення коваріацій U^C .

Відношення упередженості:

$$U^M = \frac{(y_i - \hat{y}_i)^2}{\frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2}$$

використовується для перевірки наявності систематичних відхилень для середніх реальних і прогнозованих рядів.

Інакше кажучи U^M визначає, чи модель весь час завищує прогноз. Чим менша величина U^M , тим краща модель. Якщо $U^M = 0$, то в прогнозованих значеннях відсутня упередженість і модель високоякісна.

Відношення варіацій U^S визначається так:

$$U^S = \frac{(\sigma_{actual} - \sigma_{fitted})^2}{\frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2}.$$

Відношення варіацій використовується для перевірки того, що модель має достатньо динамічних властивостей для поглинання варіацій реальних рядів. Наприклад, модель може забезпечити систематично менші коливання, ніж коливання реальних рядів. Аналогічно U^M , менші значення U^S є індикатором меншого зміщення.

Значення відношення коваріацій U^C визначається так:

$$U^C = \frac{2(1-\rho)(\sigma_{actual} - \sigma_{fitted})^2}{\frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2}.$$

Відношення коваріацій U^C показує, наскільки корельованими є прогнозовані та реальні ряди. Рівність $U^C = 0$ є свідченням того, що прогнозовані і реальні ряди ідеально корельовано.

Три зазначені статистики пов'язані між собою таким чином:

$$U^C + U^M + U^S = 1.$$

8.8.9 Побудова функції прогнозування

Крок 10. На основі обраної моделі, будується функція прогнозування. Обчислюється прогноз поведінки ряду та визначаються оцінки точності прогнозу .

Для оцінювання якості моделі необхідно визначити, наскільки добре модель відтворює реальні часові ряди. Завжди рекомендується робити повторний (ретроспективний) прогноз після моделювання. Формальними критеріями оцінки якості прогнозу є такі.

- формальні статистики;
- поворотні точки (точки перегину);
- чутливість до зміни початкових даних;
- чутливість до зміни коефіцієнтів.

Критерії вибору кращого прогнозу (формальні статистики):

1. Середньоквадратична похибка (СКП):

$$СКП = \sqrt{\frac{1}{S} \sum_{i=1}^S (y(k+s) - \hat{y}(k+s, k))^2} .$$

2. Середня похибка прогнозу (СП):

$$СП = \frac{1}{S} \sum_{i=1}^S y(k+s) - \hat{y}(k+s, k) .$$

3. Середня похибка в процентах:

$$СПП = \frac{1}{S} \sum_{i=1}^S \frac{y(k+s) - \hat{y}(k+s, k)}{y(k+s)} \times 100\% .$$

4. Середня абсолютна похибка у процентах:

$$АСПП = \frac{1}{S} \sum_{i=1}^S \frac{|y(k+s) - \hat{y}(k+s, k)|}{|y(k+s)|} \times 100\%$$

5. Максимальна абсолютна похибка (МАП):

$$МАП = \max \{|y(k+1) - \hat{y}(k+1, k)|, \dots, |y(k+s) - \hat{y}(k+s, k)|\}.$$

6. Мінімальна абсолютна похибка (МіАП):

$$МіАП = \min \{|y(k+1) - \hat{y}(k+1, k)|, \dots, |y(k+s) - \hat{y}(k+s, k)|\}.$$

Оцінка моделей за точками перегину є важливим показником, оскільки деякі моделі можуть мати більшу точність, але можуть погано спрацьовувати при прогнозуванні змін трендів або циклів. Інші моделі можуть бути менш точними, але можуть проявляти більш багатий динамічний характер. Підсумовуючи можна говорити про компроміс між точністю і динамічними властивостями. На жаль, не існує формульного тесту цих властивостей. Проте візуальна перевірка прогнозованих і реальних рядів швидко визначає, чи включає модель точки перегину [7].

Іншим важливим тестом якості моделі є аналіз чутливості до початкових (стартових) даних. Якщо модель дає результати, в цілому грубо незалежні від початкових даних, то така модель вважається якісною.

8.8.10 Уточнення вхідних даних

Крок 11. Якщо точність прогнозу не задовольняє особу, що приймає рішення, то уточнюються вхідні дані і переходимо до кроку 1. Інакше, процес закінчується.

Запитання та завдання для самоконтролю

1. Які складових процесу використовують для прогнозування?
2. Як описують детермінований тренд?
3. Як описують стохастичний тренд?
4. Як враховують коливання, які накладаються на тренд?
5. Що собою являють гетероскедастичні процеси і як будують їх моделі?
6. Як обчислюють безумовні вибіркове середнє і дисперсію?
7. Як обчислюють умовне математичне сподівання стохастичного процесу $AR(1)$?
8. Як обчислюють умовну дисперсію?
9. Що називають функцією дисперсії?
10. Як прогнозують процес без знаходження розв'язку рівнянь?
11. Що таке узагальнення функції прогнозування (на процес $ARCS(p,q)$)?
12. Як будують функції прогнозування на основі розв'язку різницевого рівняння?
13. Що таке альтернативна форма функції прогнозування на один крок?
14. Як будують функцію прогнозування на два кроки?

15. Описати основні кроки алгоритм процесу аналізу та прогнозування на основі часових рядів.
16. Як виконують попередню обробку та аналіз даних?
17. Як перевіряють наявність нелінійностей?
18. Як перевіряють процес на стаціонарність?
19. Що описує тест Дікі – Фуллера?
20. Як перевіряють процес на наявність гетероскедастичності?
21. Пояснити Тест Уайта.
22. Пояснити Тест Бройша-Пагана/Годфрі.
23. Пояснити Тест Голдфельда-Квандта.
24. Описати кроки тесту на гетероскедастичність.
25. Пояснити моделі гетероскедастичних процесів.
26. Як визначають спосіб вилучення або моделювання тренду?
27. Як будують модель часового ряду?
28. Пояснити критерії адекватності моделі.

РОЗДІЛ 9 ОБ'ЄКТНО-ОРІЄНТОВАНІ СППР НА ОСНОВІ ПРОГНОЗУВАННЯ ЧАСОВИХ РЯДІВ

Розглянуті математичні основи побудови СППР на основі числових рядів застосовані у цілому ряді існуючих систем. Значний практичний інтерес викликають СППР прогнозування мікро і макроекономічних процесів.

Розробка СППР ґрунтується на застосуванні сучасних, випробуваних підходах проектування таких систем, зокрема об'єктно-орієнтованого підходу. У даному розділі розглянемо деякі особливості застосування розглянутих математичних методів прогнозування у створенні і використанні діючих систем [11].

Реалізацію СППР при прогнозуванні динаміки часових рядів розглянемо у вигляді послідовності необхідних етапів відповідно до розробленого алгоритму. Детально описано класи компонентів СППР.

9.1 Приклад реалізації СППР на основі часових рядів

9.1.1 Основні етапи побудови СППР прогнозування часових рядів

Реалізація СППР при прогнозуванні динаміки часових рядів розглянемо у вигляді послідовності необхідних етапів відповідно до розробленого алгоритму.

Етап I. Ідентифікація часового ряду.

На цьому етапі вибирається модель, яка з найбільшим ступенем адекватності описує часовий ряд.

Етап II. Вибір методів прогнозування.

На цьому етапі вибирається кращий метод прогнозування для наявного часового ряду.

Етап III. Вибір методів попередньої оброблення даних.

Попередня оброблення та аналіз даних включає в себе: нормування, логарифмування, заповнення пропусків, згладжування екстремальних адитивних значень і таке інше.

Етап IV. Вибір методів оцінювання параметрів моделей.

Для лінійних та квазілінійних моделей:

- метод найменших квадратів (МНК) та його рекурсивна версія (РМНК);
- метод максимальної правдоподібності (ММП) та його рекурсивна версія;
- метод допоміжної змінної (МДЗ).

Для нелінійних моделей:

- нелінійний метод найменших квадратів (НМНК);
- метод максимальної правдоподібності (ММП);
- узагальнений метод моментів (УММ);
- метод Монте Карло для Маяковських ланцюгів.

Етап V. Вибір критеріїв вибору кращих моделей з множини оцінюваних кандидатів.

На цьому етапі вибираються відомі та формуються нові критерії добору кращих моделей. Критерії адекватності моделей розглянуті на Кроці 9 алгоритму процесу прогнозування в СППР, що наведений в підрозділі 10.4.

Етап VI. Вибір методів обчислення якості прогнозу

Оцінка якості прогнозу може бути виконана за методами Кроку 10 алгоритму процесу прогнозування, що розглянутий у підрозділі 10.4.

Етап VII. Розробка функціональної структури системи та інтерфейсу.

9.1.2 Архітектура і структурна схема СППР

Розглянута СППР при прогнозуванні часових рядів є прототипом СППР універсального типу, яка проектується на основі технічного завдання конкретного замовника. Однак, створений прототип є функціонально повним і цілісним програмним продуктом, який містить необхідні складові для підтримки прийняття рішень при прогнозуванні динаміки змінних процесу, що моделюється. Структурна схема СППР зображена на рис. 9.2.

Реалізована СППР є найбільш простою з погляду архітектури, тому її впровадження може бути доцільним в організаціях, що не ставлять перед собою глобальних завдань і що мають відносно невисокий рівень розвитку інформаційних технологій.

Архітектура створеної СППР складається з таких рівнів:

- 1) завантаження і оброблення даних;
- 2) аналіз даних;
- 3) побудова та вибір кращої моделі;
- 4) прогнозування.

Перший рівень надає можливість завантаження даних шляхом імпорту із текстового файлу чи ручним введенням. Після чого можливе перетворення даних для усунення їх надлишковості та підготовка даних до аналізу.

Другий рівень забезпечує можливість візуальної оцінки даних, проведення статистичного та кореляційного аналізу.

Третій рівень надає засоби для побудови моделей авторегресії із ковзним середнім. Він надає можливість оцінити параметри якості моделі для вибору кращої.

Четвертий рівень реалізує динамічне та статистичне прогнозування на базі створених моделей.

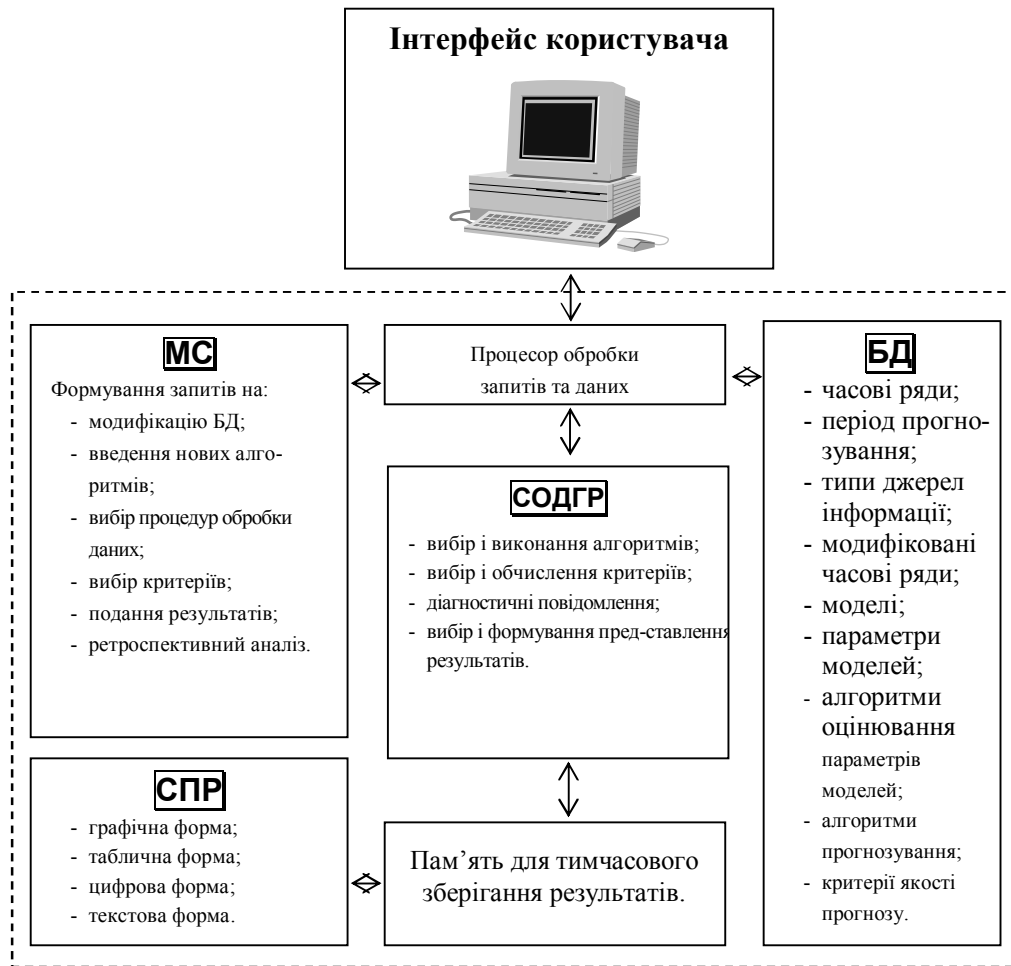


Рис. 9.2 Структурна схема СППР

9.2 Класи компонентів СППР

9.2.1 Діаграма основних класів

Відображення взаємозв'язків між окремими компонентами СППР, опис їх внутрішньої структури та типи відношень зручно розглянути в термінології класів об'єктно-орієнтованого проектування [11]. На рис.9.3 представлена діаграма основних класів розробленої СППР.

Головним класом на діаграмі є клас *BasicForm*, оскільки він реалізує головне вікно програми. Також важливими класами є класи *DataClass* та *GlobalDataTables*, що виступають в якості накопичувачів даних в процесі роботи програми. Розглянемо кожний клас докладніше.

9.2.2 Опис основних класів

9.2.2.1 Клас DataClass

Клас *DataClass* використовується для зберігання завантажених часових рядів, надає методи доступу та редагування даних.

Атрибутами класу виступають наступні параметри:

- `countAllRows` – вказує кількість завантажених рядів за час роботи програми;
- `dataAr` – ступінчатий масив для збереження завантажених рядів;
- `nameAr` – масив імен завантажених рядів.

Методи класу:

- `AddDataAr(string, double[])` – додає ряд в масив `dataAr`;
- `AddElements(int,int)` – додає вказану кількість невизначених елементів у кінець вказаного ряду;
- `CountRows()` – повертає кількість завантажених рядів;
- `DelDataAr(int)` – видаляє вказаний ряд;
- `DelElement(int,int)` – видаляє вказаний елемент із вказаного ряду;
- `EditElement(int,int,double)` – присвоює нове значення елементу ряду;
- `EditNameRow(int,string)` – змінює ім'я ряду;
- `Element(int,int)` – повертає значення заданого елементу ряду;
- `IndexRow(string)` – повертає індекс ряду по його імені;
- `LenghtRow(int)` – повертає довжину вказаного ряду;
- `NameRow(int)` – повертає ім'я ряду по його індексу;

9.2.2.2 Клас BasicForm

Клас BasicForm утворює головне вікно програми з якого розпочинають свій життєвий цикл об'єкти усіх інших класів.

Методи цього класу наступні:

- LoadData () – зчитує дані із текстового файлу та зберігає їх;
- NonstatTest() – виконує тест на не стаціонарність ряду;
- RefreshRowDataGridView() – оновлює вікно;
- StaticBehaviorForm() – викликає вікно із статистичними характеристиками ряду;
- ARMAForm() – викликає вікно для побудови АРКС моделі;
- CorrelationForm() – викликає вікно із коваріаційною та кореляційною матрицями;
- CorrelationForm() – викликає вікно із графіками АКФ та ЧАКФ;
- ShowRows() – викликає віно перегляду та редагування ряду.

9.2.2.3 Клас ShowRowsForm

Клас ShowRowsForm використовується для перегляду елементів рядів та їх редагування.

Методи класу:

- AddClearRow(string) – додає пустий ряд;
- AddRow(int) – додає уже існуючий ряд;
- Copy(int) – копіює ряд;
- Del(int) – видаляє ряд;
- Graph(int[]) – графічне подання ряду;
- Rename(int) – перейменовує ряд;

- Save(int) – зберігає внесені зміни;
- SaveInFile(int) – зберігає ряд у текстовий файл.

9.2.2.4 Клас GraphRowForm

Клас GraphRowForm виконує графічне подання даних ряду. Атрибут pane – містить криві, що відображаються. Метод Draw() – відображає задані криві у вікні.

9.2.2.5 Клас DataConversionClass

Клас DataConversionClass містить методи оброблення рядів:

- DifferRow(double[],int) – обчислює різниці заданого порядку;
- IsNegativeNumber(double[]) – вказує, чи містить ряд від’ємні елементи;
- IsUndefinedElement(double[]) – вказує, чи містить ряд невизначені елементи;
- LogRow(double[], double) – логарифмує ряд;
- MaxElementAr(double[]) – повертає максимальний елемент ряду;
- MinElementAr(double[]) – повертає мінімальний елемент ряду;
- NormalizationRow(double[]) – нормує ряд.

9.2.2.6 Клас StatisticalAnalysisClass

Клас StatisticalAnalysisClass містить методи обчислення статистичних показників:

- ACF(double[],int) – обчислює АКФ;
- Correlation(double[][]) – обчислює кореляцію;

- Covariation(double[][][]) – обчислює коваріацію;
- Dispersion(double[]) – обчислює дисперсію;
- DW(double[]) – обчислю статистику Дарбіна-Уотсона;
- F(double[]) – обчислює F-статистику;
- JB(double[]) – обчислює статистику Жака-Бера;
- Kurtosis(double[]) – обчислює куртозис (ексцес);
- MAE(double[]) – обчислює середню абсолютну похибку у відсотках;
- Middle(double[]) – обчислює середнє;
- PACF(double[],int) – обчислює ЧАКФ;
- R2(double[]) – обчислює коефіцієнт детермінації;
- SeofRegression(double[]) – обчислює середньоквадратичну похибку;
- Skewness(double[]) – обчислює коефіцієнт асиметрії;
- SSE(double[]) – обчислює суму квадратів похибок;
- U(double[]) – обчислює коефіцієнт Тейла.

9.2.2.7 Клас ARMAForm

Клас ARMAForm утворює вікно для побудови та вибору кращої моделі.

Атрибути:

- `_paramsAR` – порядок AP;
- `_paramsMA` – порядок КС;
- `ARMA` – клас побудованої моделі;
- `Method` – метод обчислень параметрів.

Методи:

- Calc() – обчислення параметрів моделі;
- CreateHMatrix() – створення матриці H;
- CreateYMatrix() – створення матриці Y;
- PACFResid() – обчислення ЧАКФ залишків.

9.2.2.8 Клас MethodClass

Клас MethodClass містить методи обчислення коефіцієнтів моделі:

- LSM() – реалізує МНК;
- RLSM() – реалізує РМНК.

9.2.2.9 Клас PrognosisForm

Клас PrognosisForm утворює вікно для прогнозування.

Атрибути:

- ARMA – клас побудованої моделі;
- PrognosisAr – масив прогнозованих значень.

Методи:

- DunPrognosis() – динамічне прогнозування;
- StatPrognosis() – статичне прогнозування;
- OutPrognosis() – виведення прогнозованих значень.

9.2.2.10 Клас ARMAModlClass

Клас ARMAModlClass створений для збереження моделі АРКС.

Атрибути:

- _paramsAR – порядок АР;

- `_paramsMA` – порядок КС;
- `dataAr` – вхідний ряд даних;
- `MatrxB` – матриця параметрів моделі;
- `MatrxH` – матриця H;
- `MatrxResid` – матриця залишків;
- `rowName` – ім'я ряду.

Методи:

- `Clear()` – очищує всі атрибути;
- `Exist()` – вказує чи існує модель.

9.2.2.11 Клас `MatrixClass`

Клас `MatrixClass` створений для роботи з матрицями. Атрибут `matrix` – зберігає двохвимірний масив даних.

Методи:

- `Add(MatrixClass, MatrixClass)` – обчислює суму двох матриць
- `Array()` – повертає масив даних, що представляє матрицю;
- `CountColumns()` – кількість стовпців матриці;
- `CountRows()` – кількість рядів матриці;
- `DiagonalMatrix(double)` – створює діагональну матрицю;
- `Diff(MatrixClass, MatrixClass)` – виконує віднімання двох матриць;
- `IncrConst(MatrixClass, double)` – перемножує матрицю на число;
- `Increase(MatrixClass, MatrixClass)` – перемножує дві матриці;
- `Inverse(MatrixClass)` – повертає обернену матрицю;
- `Method_Haleckogo()` – реалізує метод Халецького;
- `Transpose(MatrixClass)` – транспонує матрицю.

9.2.2.12 Клас Leksem

Клас Leksem створений для визначення та збереження окремої лексеми.

Атрибути:

- Alphabet – алфавіт;
- Commands – набір допустимих команд;
- Delimiters – набір допустимих розділювачів;
- Logics – набір допустимих логічних команд;
- Operators – набір допустимих операторів;
- Txt – текстове подання лексеми;
- Type – тип лексеми.

Методи:

- GetLeksems(string) – розбиває текст команди на лексеми;
- SetNewValues(string, LeksemType) – змінює лексему;
- WhithLeksemType() – повертає тип лексеми.

9.2.2.13 Клас Expression

Клас Expression виконує команди користувача.

Атрибути:

- Active – індекс активної команди;
- Commands – масив команд.

Методи:

- ArrayRun() – виконує команду оголошення масиву;
- Calc() – виконує обчислення виразу;
- CommandRun() – виконує обчислення команди;

- DelSpace() – видаляє пробіли із тексту команди;
- DoRun() – виконує цикл do {...}while(..);
- ElementaryCalc() – обчислює елементарний вираз, що не містить змінних, функцій та дужок;
- FunctionRun() – команда виконання функції;
- IfRun() – виконання команди умовного оператора if-else;
- OverrideVariable() – заміню у виразі змінні та функції на їх абсолютні значення;
- Run() – запуск команди;
- WhileRun() – виконання циклу while() {}.

9.2.2.14 Клас GlobalDataTables

Клас GlobalDataTables зберігає усі змінні та масиви.

Атрибути:

- Arrays – таблиця масивів;
- Variables – таблиця змінних.

Методи:

- AddArray(string,int) – додає новий масив;
- AddValue(string,double) – додає нову змінну;
- GetArrEl(string,int) – повертає елемент масиву;
- GetValue(string) – повертає значення змінної;
- SetArrEl(string,int,double) – встановлює значення елементу масиву;
- SetValue(string,double) – встановлює значення змінної.

9.2.2.15 Клас InnerCLIFunctions

Клас InnerCLIFunctions містить додаткові методи:

- Write(Leksem) – виводить дані на консоль;
- WriteLn(Leksem) – виводить дані на консоль і переміщує курсор у новий рядок.

9.3 Функціональні можливості СППР

9.3.1 Функціональна схема СППР

Функціональну схему СППР прогнозування подано на рис. 9.4. Завантаження даних до програми можливе двома способами:

1. Головне меню «Ряд» → «Завантажити». Після чого вибрати текстовий файл з даним. Необхідною умовою успішного виконання команди є розділення цілої та дробової частини комою. А також запис даних у файлі у один стовпчик.
2. Головне меню «Ряд» → «Додати пустий ряд». Після чого вказати ім'я та розмірність ряду.

Також через головне меню «Ряд» можна копіювати, видаляти та зберігати ряд даних у текстовий файл.

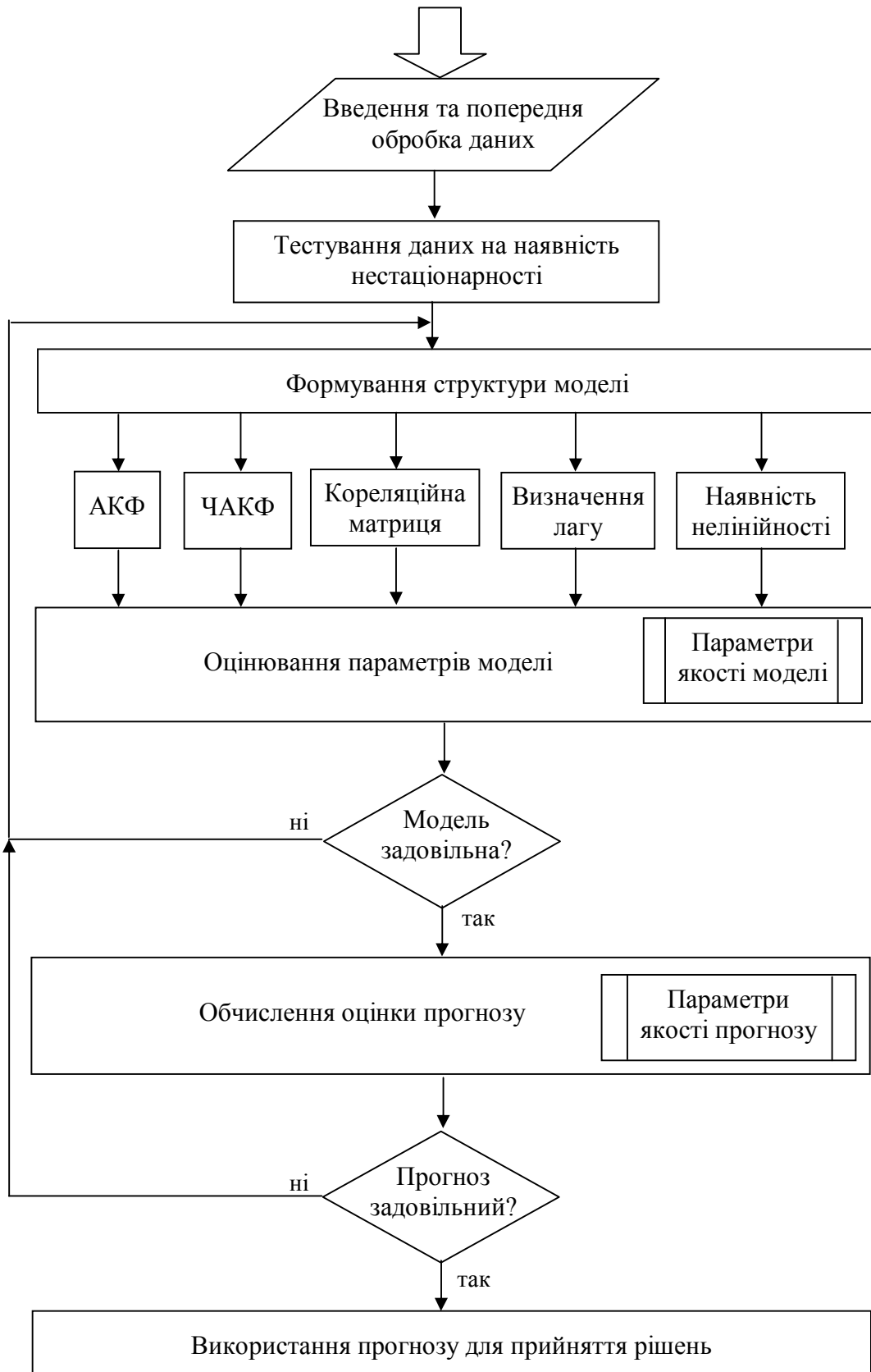


Рис.9.4 Функціональна схема СПДР

9.3.2 Редагування та попереднє оброблення даних

Щоб відкрити ряд для перегляду та редагування достатньо двічі натиснути на ньому ліву кнопку мишки, чи виконати команду меню «Ряд» → «Показати». Після чого відкриється вікно для маніпулювання даними, в якому можна виконувати такі функції:

- додавати чи видаляти елементи ряду;
- відкривати чи створювати нові ряди;
- нормувати чи логарифмувати дані;
- обчислювати різниці заданого порядку;
- будувати графіки рядів даних;
- змінювати ім'я ряду;
- зберігати внесені значення у програму чи у текстовий файл.

9.3.3 Графічне подання даних

Графічне подання рядів можливе за допомогою вибору в меню «Графік» вікна «Перегляду та редагування ряду» (рис. 9.5).

У вікні «Графік» за допомогою консольного меню можливе редагування масштабу зображення, збереження у файл чи роздрукування на принтері. А також за допомогою головного меню «Налаштування» можливо змінювати імена та колір кривих, тип ліній та точок.

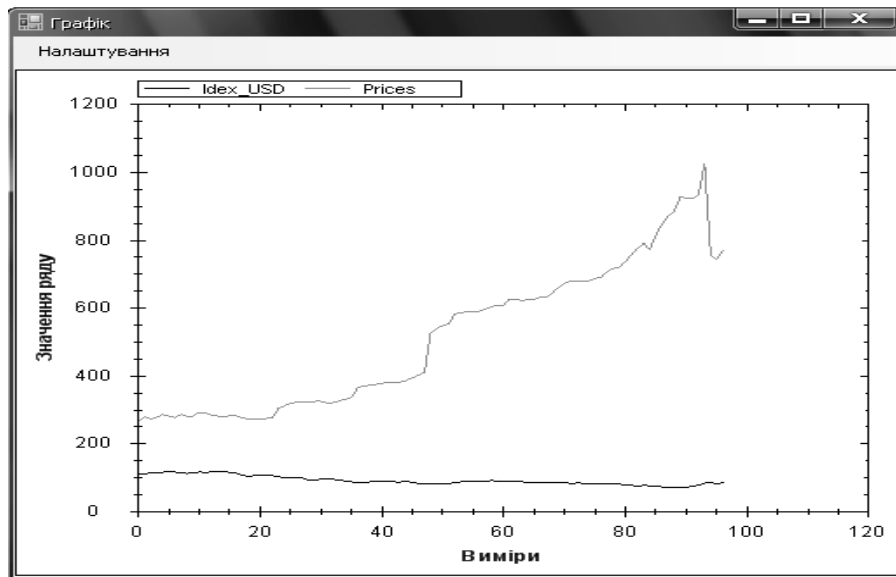


Рис. 9.5 Графічне подання рядів даних

9.3.4 Аналіз даних

Аналіз даних здійснюється через меню «Аналіз» головного вікна програми, що надає можливість:

- обчислення описових статистик;
- обчислення АКФ та ЧАКФ;
- обчислення коваріаційної та кореляційної матриці;
- проведення тесту на стаціонарність/нестаціонарність ряду.

9.3.5 Побудова та вибір кращої моделі АРКС

На рис. 9.6 представлено вікно побудови моделі, в якому є можливість вибирати ряд, за яким будується модель, параметри які будуть оцінюватись, а також метод оцінки параметрів. Після побудови моделі через меню «Дані» доступні такі операції:

- оцінити характеристики моделі;

- переглянути матрицю H;
- зберегти залишки моделі для тесту на гетероскадестичність.

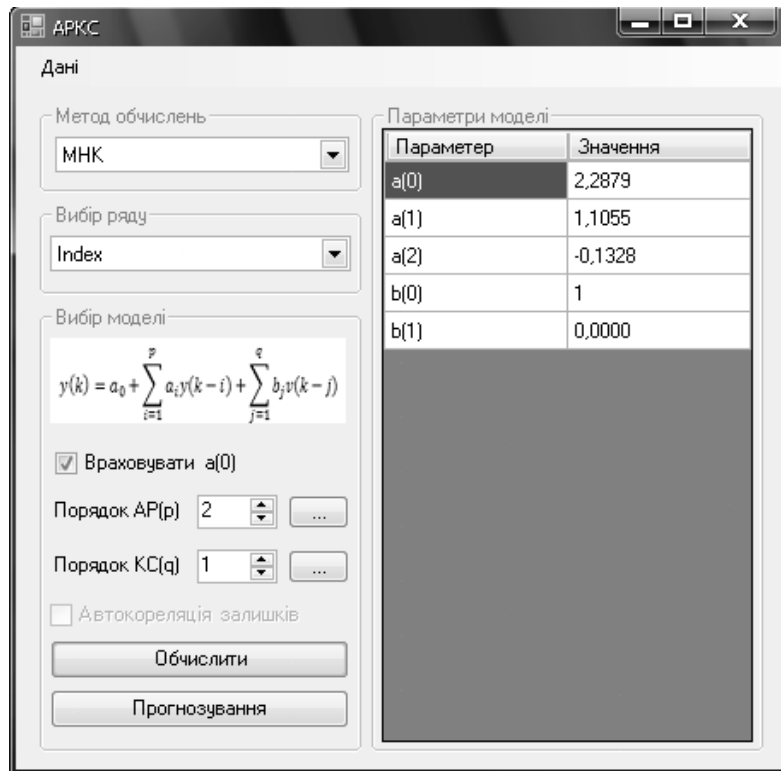


Рис. 9.6 Вікно побудови моделі АРКС

9.3.6 Прогнозування

Після побудови та вибору моделі стає доступним заключний етап аналізу – прогнозування. У вікні для прогнозування (рис. 9.7) вибирається тип прогнозу: статичний чи динамічний; і «вікно прогнозу» – початкова та кінцева точки прогнозу. Після обчислення прогнозних результатів для користувача доступне графічне подання вхідних та прогнозованих даних для візуального порівняння.

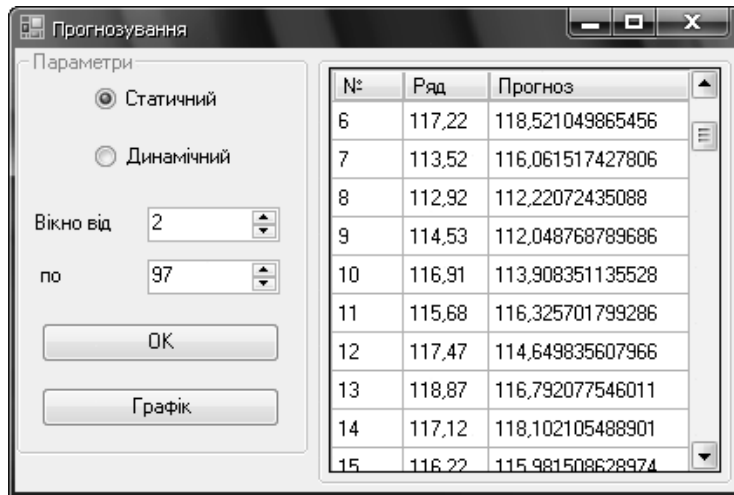


Рис. 9.7 Вікно прогнозування

9.3.7 Командний блок СППР

Для покращення та розширення можливостей редагування часових рядів в СППР реалізовано модуль командного блоку, який відображається в головному вікні системи та складається з двох частин:

- консолі, що виводить команди та результати їх виконання;
- текстового поля для введення команд.

Завдяки наявності циклів, функцій та масивів режим командного блоку надає можливість швидко формувати нові чи редагувати вже існуючі елементи ряду за допомогою обчислення довільної математичної формули, що введенняють користувач.

Синтаксис команд, що можуть введеннятись до командного блоку, є надзвичайно простим та легким для користувача. Послідовні команди необхідно розділяти крапкою з комою. Всі пробіли при виконанні команд будуть проігноровані і не є значимими. Також реалізовано синтаксичний аналізатор команд, який є регістронезалежним, тобто не розрізняє верхній та нижній регістри.

Алфавіт мови командного блоку (допустимі символи):

- символи латинського алфавіту: `qwertyuiopasdfghjklzxcvbnm`
- оператори: `+ - / * ^ = ! | & > <`
- розділювачі: `; () { } []`.

Оголошувати змінні не потрібно. За замовчанням тип усіх змінних – `double`, а початкове значення за замовченням – нуль. Логічні вирази обчислюються за принципом, що нуль – це `false`, а усі значення відмінні від нуля – `true`.

Оголошення масиву: **array** *ідентифікатор* [*розмірність масиву*] ;

Звернення до елемента масиву: *ідентифікатор* [*індекс елемента*] ;

Виклик функції: *ідентифікатор* (*парм.1* , *парм.2* , ..) ;

Цикл `while`: **while** (*вираз*) { *ком.1* ; *ком.2* ; .. } ;

Цикл `do`: **do** { *ком.1* ; *ком.2* ; .. } **while** (*вираз*) ;

Оператор `if-else`: **if** (*вираз*) { *ком.1* ; *ком.2* ; .. } **else** { *ком.1* ; *ком.2* ; ... } ;

Інші команди: **return, continue, break**

Вбудовані функції:

- **Write**(*змінна/число/вираз*) – виводить дані на консоль;
- **WriteLn**(*змінна/число/вираз*) – виводить дані на консоль зміщує курсор на новий рядок;
- **Set**(*індекс ряду; індекс елемента; значення*) – встановлює нове значення елемента числового ряду;
- **Get**(*індекс ряду; індекс елемента;*) – повертає значення елемента ряду;
- **Del**(*індекс ряду; індекс елемента;*) – видаляє елемент заданого ряду;
- **Add**(*індекс ряду; кількість нових невизначених елементів*) – додає задану кількість невизначених елементів в кінець заданого ряду;

- **Abs**(число) – повертає модуль числа;
- **Log**(число, базис логарифму) - повертає логарифм числа;
- **Exp**(число) – підносить експоненту до заданого степеня;
- **Cos**(радіани) – обчислює косинус кута заданого в радіанах;
- **Sin**(радіани) – обчислює синус кута заданого в радіанах;
- **Tan**(радіани) – обчислює тангенс кута заданого в радіанах.

Текст, що введеться в текстове поле командного блоку, потрапляє до синтаксичного аналізатору інтерпретатора командного блоку. Після зчитування команди аналізатором видаляються із неї усі пробіли. Далі команда розбивається на лексеми, з якими в подальшому працює інтерпретатор. Блок-схема розбиття команд синтаксичним аналізатором на лексеми зображена на рис. 9.8.

Групи лексем:

- Оператори: +/*^=;
- Змінні – повинні розпочинатися із латиської літери, а також містити лише латинські символи та цифри;
- Розділювачі: ;(){}[];
- Команди: if,else,do,while,array,return,continue,break;
- Числа;
- Масиви; імя масива повинне відповідати формату змінної і закінчуватися розділювачами []. Наприклад : a[1], b111[10];
- Логічні оператори: !,!=,==,|,&,<=,>=,<,>;
- Функції – імя функції повинне відповідати формату змінної і закінчуватися розділювачами (). Наприклад: write(10).

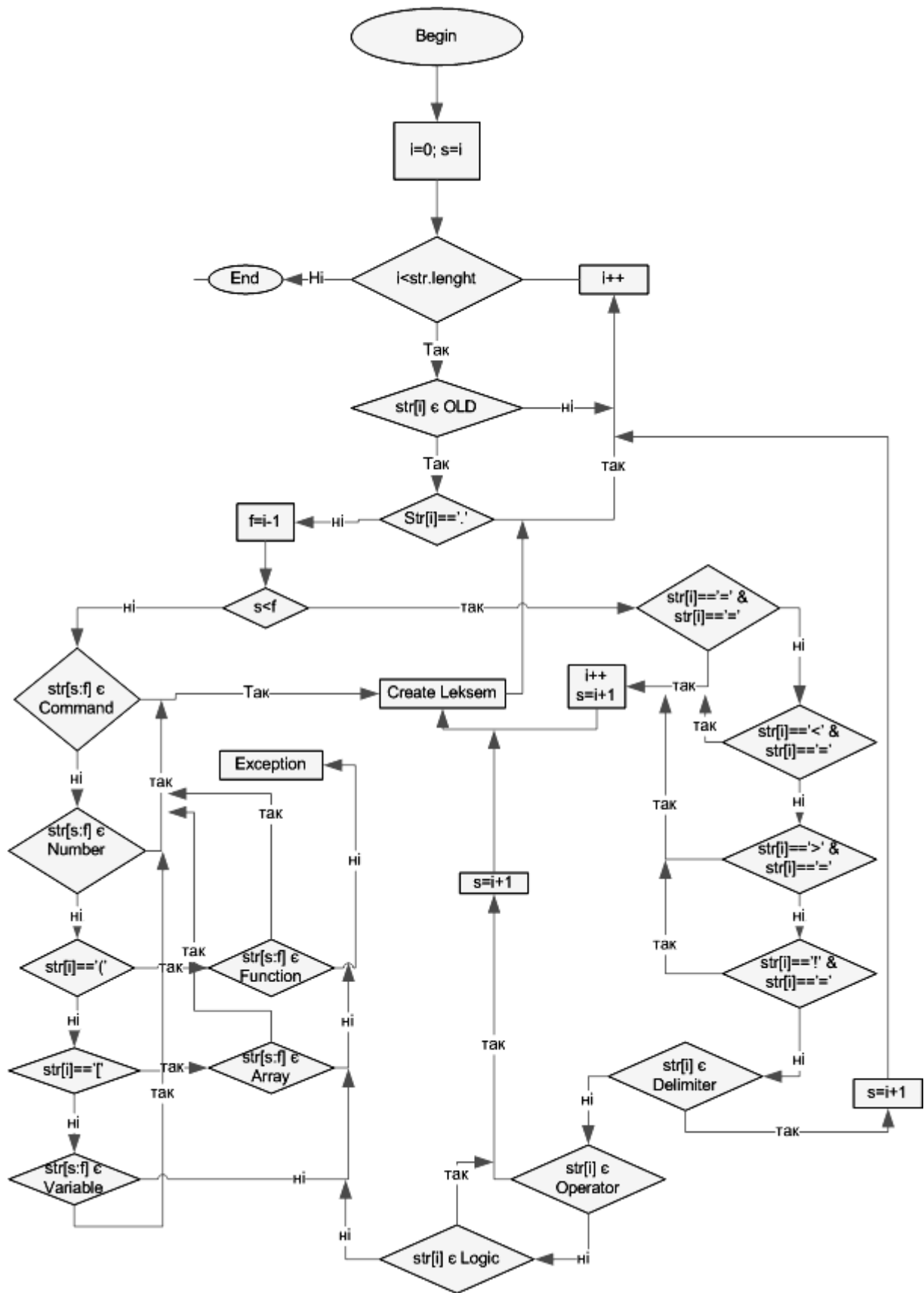


Рис.9.8 Блок-схема розбиття команди на лексеми

Можливі варіанти початку команди:

- із змінної;
- із масиву;
- із функції;
- із команди.

Обчислення виразів складається із таких етапів:

1. Заміна усіх змінних, масивів та функцій їх значеннями.
2. Перевірку на наявність мінуса на початку виразу.
3. Обчислення усіх виразів, що дані в дужках.
4. Обчислення кінцевого елементарного вираження, що не містить змінних, масивів, функцій та дужок в порядку пріоритетності операторів.

Усі змінні зберігаються у глобальній таблиці змінних «Variables».

Усі масиви зберігаються у глобальній таблиці масивів «Arrays».

9.4 Приклад застосування СППР для моделювання та прогнозування ВВП України

Розглянемо приклад застосування розробленої СППР для дослідження та прогнозування ВВП України. За значеннями поквартальних показників ВВП за допомогою СППР побудуємо авторегресійну модель і зробимо однокроковий статичний прогноз.

За графічним поданням даних можна зробити припущення про наявність тренду (рис. 9.9). Результати тесту Дікі-Фуллера (рис. 9.10) підтверджують таку гіпотезу, тому перед подальшими дослідженнями та моделюванням варто виконати попередню оброблення даних, щоб

привести ряд до стаціонарного виду. Для цього обчислимо перші різниці і отримаємо стаціонарний ряд з яким будемо працювати в подальшому (рис.9.11).

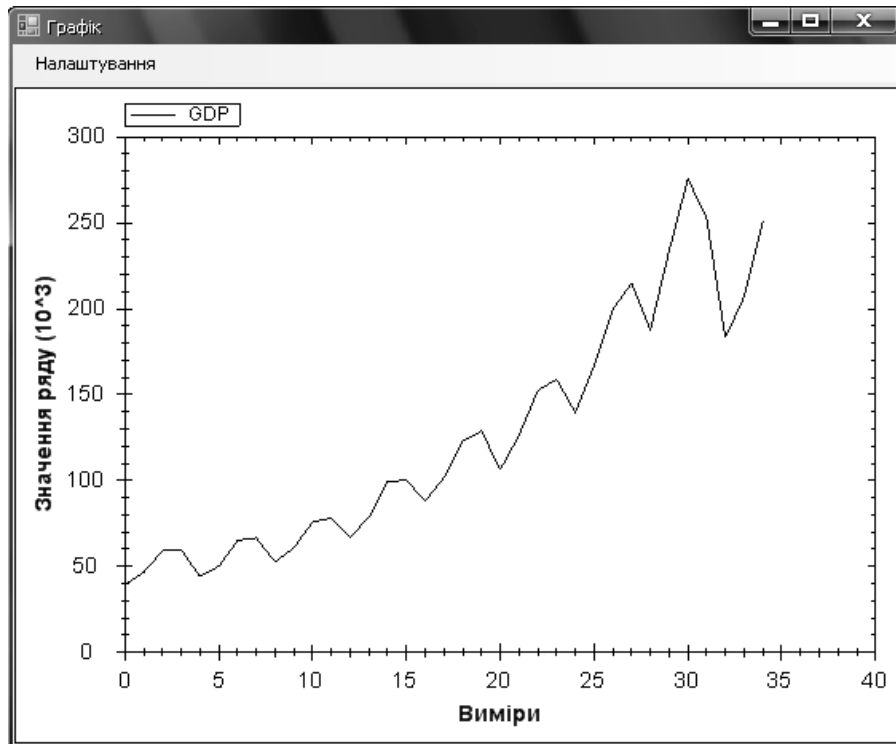


Рис. 9.9 Поквартальні показники ВВП України з 2001 року, у мільйонах гривень

Тест Дік-Фуллера. Ряд "GDP"		
Тобч.	Ткр. при 5%	Ткр. при 1%
1,111	-1.94	-2.57

Рис.9.10 Результати тесту на стаціонарність ряду

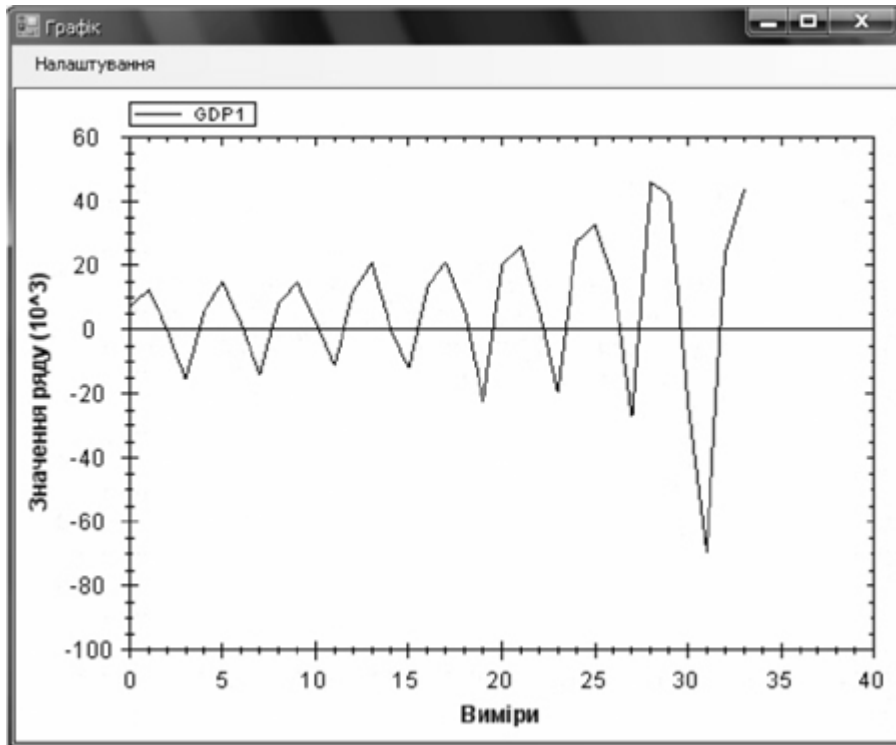


Рис. 9.11 Ряд ВВП України після взяття перших різниць

Після розрахунків АКФ та ЧАКФ (рис.9.12) для подальшого аналізу виберемо такі моделі АРКС:

- 1) $y_1(k) = a_2 * y_1(k-2) + a_5 * y_1(k-5) + e(k) + b_3 * e(k-3) + b_4 * e(k-4) + b_5 * e(k-5);$
- 2) $y_1(k) = a_2 * y_1(k-2) + a_4 * y_1(k-4) + a_5 * y_1(k-5) + e(k) + b_1 * e(k-1),$

Тут $y_1(k) = ВВП(k) - ВВП(k-1)$

Після обчислення характеристик вибраних моделей (рис. 7.13) остаточно зупиняємось на другому рівнянні АРКС(5,1), що краще описує наш процес:

$$y_1(k) = a_2 * y_1(k-2) + a_4 * y_1(k-4) + a_5 * y_1(k-5) + e(k) + b_1 * e(k-1)$$

Тут $y_1(k) = ВВП(k) - ВВП(k-1)$

Оцінені параметри обраної моделі АРКС(5,1) за методами МНК та РМНК наведені в таблиці 9.1.

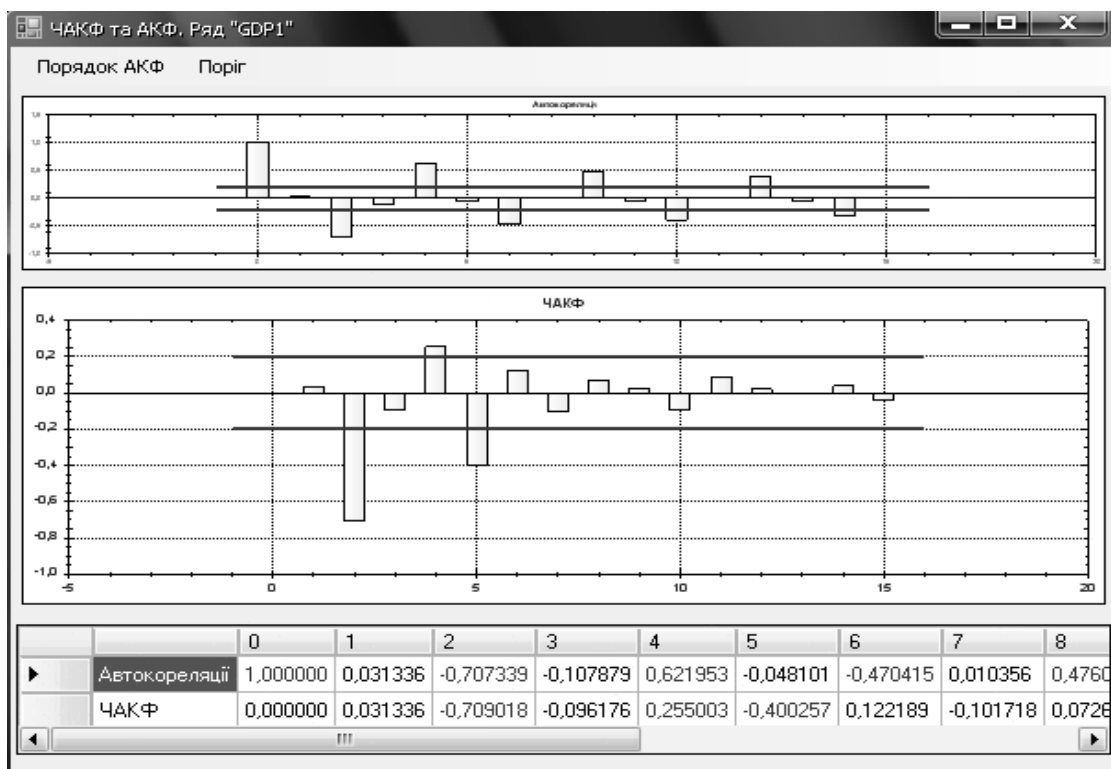


Рис.9.12 АКФ та ЧАКФ ВВП після взяття перших різниць

Таблиця 9.1 Оцінка параметрів моделі АРКС(5,1)

Коефіцієнт моделі	Оцінка в СППР, метод МНК	Оцінка в СППР, метод РМНК
a_2	-0,3258	-0,3258
a_4	0,9875	0,9875
a_5	-0,1574	-0,1574
b_1	0,6493	0,6493

Параметри якості моделі.		Параметри якості моделі.	
Коэф. детермінації	0,6528	Коэф. детермінації	0,9872
СКП	18602,8407	СКП	11781,0849
DW	2,2125	DW	0,8660
F-статистика	1,8798	F-статистика	230,6912
Кр. Тейла	0,3970	Кр. Тейла	0,2325

а)АРКС(5,5)

б) АРКС(5,1)

Рис.9.13 Характеристики обраних моделей

В таблиці 9.2 представлені результати прогнозування ВВП за вибраною моделлю АРКС(5,1).

Таблиця 9.2 Результати прогнозування ВВП України

Вимір	Реальне значення, млн грн	СППР			
		МНК		РМНК	
		Прогноз, млн грн	Похибка, млн грн	Прогноз, млн грн	Похибка, млн грн
2008-4	252 670	268 803	-16 133	268 803	-16 133
2009-1	183 217	210 871	-27 654	210 871	-27 654
2009-2	207 096	242 582	-35 486	242 570	-35 474
2009-3	250 614	265 792	-15 178	265 788	-15 174
2009-4		214 168		214 171	
Середня абсолютна похибка			-18 890		-18 887
Середня абсолютна похибка, %			11,2%		11,2%

Скорочення ВВП, СППР, МНК, РМНК наведені за переліком скорочень даної роботи

Запитання та завдання для самоконтролю

1. Дати характеристику основних етапів побудови СППР прогнозування часових рядів.
2. Які рівні містить структурна схема СППР?
3. Дати оцінку основних класів СППР.
4. Які основні компоненти містить функціональна схема СППР і як вони взаємодіють?
5. Що відображає графічне подання рядів даних?
6. Як виконують аналіз даних через меню «Аналіз» СППР?
7. Описати послідовність побудови та вибору кращої моделі АРКС.
8. Як виконують прогнозування. у вікні для прогнозування?
9. Пояснити призначення командного блоку СППР.
10. Як виконують розбиття команди на лексеми?
11. Як виконують прогнозування на основі АКФ, ЧАКФ та моделі АРКС?
12. Для чого необхідні прогнози? Які процеси відносять до нелінійних?
13. В чому полягає різниця між умовними та безумовними статистичними характеристиками випадкових процесів?
14. Вкажіть, які умовні та безумовні статистичні характеристики використовуються при виконанні статистичного аналізу випадкових процесів. Які існують типи нестационарності?
15. Розкрийте процес знаходження оцінок прогнозів без знаходження розв'язків лінійних різницевого рівнянь.
16. Опишіть процедури побудови функції прогнозування на основі розв'язку різницевого рівняння.

17. Сформулюйте припущення, на яких ґрунтується побудова функції прогнозування з мінімумом дисперсії.
18. Наведіть алгоритм процесу побудови математичної моделі та оцінювання прогнозів на основі часових рядів.
19. Поясніть здійснення перевірки процесу на стаціонарність за допомогою тесту Дікі – Фуллера. Який тип нестаціонарності аналізується в даному випадку.
20. Що таке гетероскедастичність і де вона може зустрічатись?
21. Опишіть відомі вам тести на гетероскедастичність процесу.
22. Сформулюйте статистичні критерії перевірки моделі на адекватність – коефіцієнт детермінації і статистика Дарбіна-Уотсона.
23. В чому суть статистики Фішера?
24. Що означає термін «перенавчання моделі»?
25. Які існують критерії вибору кращої оцінки прогнозу процесу?
26. Чому для оцінювання ступеня адекватності моделі і якості прогнозу не можна використовувати?

РОЗДІЛ 10 ПОБУДОВА СППР НА ОСНОВІ БМ

БМ знаходять все ширше застосування в інформаційних системах обробки статистичних, даних, представлених часовими рядами і часовими перерізами, а також якісними даними, представленими експертними оцінками, лінгвістичними змінними, інтервальними значеннями тощо.

У розділі розглянуто широке коло питань, пов'язаних поданням і застосуванням БМ. Даний розділ присвячено розробці практичної методики побудови БМ, яка може бути використана за наявності достатньої статистичної інформації стосовно досліджуваної системи, необхідної для побудови БМ. Пропонована методика може бути використана також тими дослідниками, хто вже має уяву про БМ, але не має достатнього досвіду щодо їх побудови та застосування. Спочатку розглянуто понятійні питання стосовно використання теореми Байєса, а на далі, на їх основі загальні принципи побудови та навчання БМ на основі експериментальних (статистичних) даних.

10.1 Основи застосування байєсових мереж у прийнятті рішень

10.1.1 Сфери використання

Широке застосування БМ знайшли у розв'язанні задач медичної діагностики, де вони допомагають ставити та уточнювати діагнози самих різних хвороб в умовах неточної та неповної інформації [5, 18]. Відомі застосування БМ у системах технічної діагностики, зокрема у системі моніторингу космічного корабля багаторазового використання, у діагностиці двигунів різних типів та призначення, аналізі стану технологічних процесів і технічних систем [58, 62]. Широке застосування знаходять БМ в системах класифікації даних різної природи, системах

автоматичного розпізнавання мовних сигналів, маркетингу і бізнесі, а також у багатьох інших сферах діяльності [5, 8, 58, 62].

Суттєвою перевагою застосування БМ є можливість встановлення причинно-наслідкових зв'язків між подіями та визначення ймовірності виникнення тієї чи іншої ситуації при отриманні нової інформації стосовно зміни стану будь-якого вузла (змінної) мережі.

Ступінь успішності застосування БМ у методах моделювання та формування статистичного висновку залежить від вміння коректно сформулювати постановку задачі, вибрати змінні процесу, які в достатній мірі характеризують його динаміку або статику, зібрати статистичні дані та використати їх для навчання мережі, а також коректно сформувати результат – висновок за допомогою побудованої мережі.

10.1.2 Побудова БМ

10.1.2.1 Постановка задачі

Побудова БМ пов'язана з необхідністю послідовного розв'язання кількох задач, зокрема це задачі обчислювального характеру, що зустрічаються при навчанні мережі. В загальному випадку навчання мережі відноситься до NP -повних задач, тобто об'єм обчислень зростає поліноміально із збільшенням кількості вузлів (змінних) мережі [51, 52].

Необхідно розробити методику побудови (формування структури) БМ у вигляді спрямованого ациклічного графа, який призначений для моделювання та візуалізації інформації щодо конкретної задачі навчання мережі на основі наявної інформації та формування статистичного висновку – прийняття рішення щодо поставленої задачі. БМ можна розглядати як модель подання ймовірнісних залежностей (взаємозв'язків) між його вершинами. Зв'язок $A \rightarrow B$ називають причинним, якщо подія A

є причиною виникнення B , тобто якщо існує механізм впливу значень змінної A на значення, які приймає змінна B . БМ називають причинною (каузальною) тоді, коли всі її зв'язки є причинними.

Формально, БМ – це трійка $\mathbf{N} = \langle \mathbf{V}, \mathbf{G}, \mathbf{J} \rangle$, першою компонентою якої є множина змінних \mathbf{V} ; другою – спрямований ациклічний граф \mathbf{G} , вузли якого відповідають випадковим змінним модельованого процесу; J – спільний розподіл ймовірностей змінних $\mathbf{V} = \{X_1, X_2, \dots, X_n\}$. При цьому стосовно множини змінних виконується марковська умова, тобто кожна змінна мережі не залежить від усіх інших змінних, за винятком батьківських попередників цієї змінної.

Спочатку ставиться задача обчислення значень взаємної інформації між усіма вершинами (змінними) мережі. Потім необхідно знайти оптимальну структуру мережі з використанням за критерій якості оцінку опису мережі мінімальної довжини (ОМД), значення якої аналізується і оновлюється на кожній ітерації алгоритму навчання.

10.1.2.2 Теорема Байєса і формування висновку на її основі

Ймовірність одночасної появи двох незалежних подій D і S визначається за виразом:

$$p(D, S) = p(D)p(S).$$

Якщо події D і S залежні, то поява однієї з них дає деяку інформацію про можливість появи іншої: $p(D, S) = p(D)p(S | D)$, де $p(S | D)$ – ймовірність появи події S при умові, що вже мала місце

подія D . Наприклад, подію D можна інтерпретувати як захворювання, а S , як симптом.

Якщо є інформація про те, що пацієнт має деяке захворювання, то можна присвоїти вищу ймовірність появи визначеного симптому. Враховуючи комутативність наведеного вище виразу, можна записати:

$$p(D,S) = p(S)p(D|S) = p(D)p(S|D),$$

а звідси маємо просту форму теорему Байєса (ТБ):

$$p(D|S) = \frac{p(D)p(S|D)}{p(S)}.$$

Теорему Байєса можна розглядати як механізм формування висновку (прийняття рішення). Припустимо, що розглядається проста задача постановки діагнозу. В даному випадку маємо: $p(D|S)$ – ймовірність захворювання при наявності симптому S , тобто це подія, відносно якої необхідно сформулювати висновок; $p(D)$ – ймовірність захворювання на конкретну хворобу в межах деякої популяції, яку можна оцінити на основі аналізу історії розвитку цієї популяції; $p(S|D)$ – ймовірність появи симптому, якщо пацієнт вже хворий. Останню величину можна оцінити за допомогою даних, взятих з історій хвороб. Ймовірність появи даного симптому S у вибраній популяції позначимо через $p(S)$; цю величину також можна обчислити на основі статистичних даних, але в цьому, зазвичай, немає необхідності (покажемо це нижче).

Припустимо, що змінна захворювання D має два стани (або може приймати два можливих значення): D_t – істинне значення ймовірності, яке

означає, що пацієнт має хворобу; D_f – неістинне (протилежне) значення. Ці два значення ймовірності дають в сумі одиницю незалежно від того, яке значення приймає S :

$$p(D_t | S) + p(D_f | S) = 1.$$

Застосуємо до останньої рівності теорему Байєса:

$$\frac{p(D_t)p(S | D_t)}{p(S)} + \frac{p(D_f)p(S | D_f)}{p(S)} = 1$$

або:

$$p(S) = p(D_t)p(S | D_t) + p(D_f)p(S | D_f).$$

Тобто знаючи оцінку $p(S)$, її можна виключити з подальшого розгляду. В даному прикладі змінна D має тільки два стани, але, очевидно, що $p(S)$ можна виключити з розгляду при довільному числі станів D .

Теорему Байєса можна розглядати як вираз (механізм), який об'єднує «апріорну» та «правдоподібну» інформацію, запишемо її у вигляді:

$$p(D | S) = \alpha p(D) p(S | D),$$

де $\alpha = 1/p(S)$ – нормуюча константа.

Тепер $p(D)$ можна розглядати як апіорну інформацію, оскільки вона була відома до отримання будь-яких вимірів; $p(S|D)$ – правдоподібна інформація, оскільки ми отримуємо її з аналізу (вимірів) симптомів.

У деяких випадках ми можемо обчислити апіорні ймовірності на основі статистичних даних. Наприклад, апіорну ймовірність появи захворювання можна визначити в результаті ділення числа випадків захворювання на загальне число пацієнтів, які проходять огляд. Однак, в більшості випадків це неможливо зробити внаслідок суб'єктивних труднощів отримання статистичних даних, але апіорні знання можна подати у інших формах. Розглянемо ілюстративний приклад з розпізнавання образів.

10.1.2.3 Застосування БМ у розпізнаванні кіл

Приклад 10.1. Розглянемо задачу розпізнавання двох кіл у цифровому образі, які мають бути розташовані на визначеній відстані одне від одного. Алгоритми розпізнавання ґрунтуються, зазвичай, на обчисленні множини ознак та їх порівнянні з відомими.

Для розпізнавання зображення кіл можна скористатись багатьма ознаками, але у даному прикладі виберемо простий варіант розпізнавання.

Наприклад, розробимо алгоритм розпізнавання двох кіл в даному образі, які мають однакові радіуси та знаходяться на деякій відстані S одне від одного. Якщо вдається знайти два суміжних кола, то далі необхідно встановити, чи є ці кола саме тими, які ми шукаємо?

Припустимо, що центри кіл знаходяться на відстані $S = 2(r_i + r_j)$, де r_i, r_j – радіуси кіл, знайдених в образі. Для простоти приймемо, що радіуси

однакові. Для кожної пари кіл, знайдених в цифровому образі, обчислимо міру M наближення до шуканої пари кіл за виразом:

$$M = \frac{|r_i - r_j|}{r_i} + \frac{|S - 2(r_i + r_j)|}{r_i}.$$

Очевидно, що $M = 0$ при ідеальному узгодженні міри з вибраною парою кіл. Міру M можна перетворити за деякою логікою у ймовірність, наприклад, за допомогою розподілу ймовірностей. Таким чином можна знайти суб'єктивну оцінку ймовірності за допомогою обчислених значень міри M .

Альтернативною стратегією є застосування об'єктивних методів. Для цього необхідно виконати деякі експерименти. Для даного прикладу необхідно знайти розміри фігур (кіл) для множини фотографій. Для кожного виміру параметрів двох кіл обчислюємо міру M , а також запитуємо експерта – чи є вибрана пара кіл шукані кола? На основі цього експерименту можна побудувати гістограму та відповідний дискретний розподіл.

10.1.2.4 Правдоподібність прийняття рішення

Зазвичай, апріорні ймовірності ґрунтуються на фактах, які знову і знову підтверджуються із плином часу. Їх можна оцінювати на основі відомих обґрунтованих знань щодо проблеми, яка моделюється. Разом з тим, експериментальні дані зазвичай містять похибки вимірів (або похибки збору статистичних даних), що призводить до невизначеності, яку виражають через правдоподібність. У розглянутому прикладі ці похибки

можуть бути пов'язані з методичними та обчислювальними похибками алгоритму розпізнавання образів.

Алгоритм розпізнавання не може взяти і виділити коло, але він може допомогти визначити ступінь наближення деякої фігури до кола. Наприклад, можна підрахувати число пікселів, що формують коло. Знаючи число пікселів, можна обчислити ймовірність наближення цієї фігури до кола. Тобто правдоподібність можна обчислити за аналогією із обчисленням апріорних ймовірностей.

Тепер можна сформулювати правило прийняття рішення (висновку) щодо наявності шуканого зображення двох кіл в деякому образі: $p(C | I) = \alpha p(C) p(I | C)$, де $p(C)$ – апріорна ймовірність того, що два кола представляють шукані кола; вона визначається на основі міри M , а також апріорного знання щодо перетворення M у ймовірність; $p(I | C)$ – ймовірність отримання необхідної інформації щодо образу за умови, що два кола представляють собою шукані кола – це інформація щодо правдоподібності, отримана в процесі обробки вимірів.

10.1.3 Проста БМ

Проста БМ. Розглянутий спрощений підхід до формування Байєсового висновку не дає можливості застосовувати його у більш складних ситуаціях обробки апріорної інформації. Так, у виразі для міри подібності деякого образу до шуканої пари кіл

$$M = \frac{|r_i - r_j|}{r_i} + \frac{|S - 2(r_i + r_j)|}{r_i}$$

обидва члени у правій частині в однаковій мірі впливають на значення M , але це не кращий спосіб формування міри. В цю міру можна ввести нові члени, які характеризують, наприклад, колір фону навколо кіл, які шукаємо.

Тобто складнішою мірою подібності довільного образу до шуканого може бути така:

$$M = \alpha \frac{|r_i - r_j|}{r_i} + \beta \frac{|S - 2(r_i + r_j)|}{r_i} + \gamma \cdot (\text{Ознака кольору}),$$

де α , β і γ – евристичні константи, які можна визначити, наприклад, експертним шляхом.

Розглянемо випадок, коли дані щодо проблеми можуть надходити з декількох джерел. Тепер теорема Байєса приймає вигляд:

$$p(D | S_1, S_2, \dots, S_n) = \frac{p(D) p(S_1, S_2, \dots, S_n | D)}{p(S_1, S_2, \dots, S_n)}.$$

У даному випадку виникає проблема оцінювання умовної ймовірності $p(S_1, S_2, \dots, S_n | D)$ при великих значеннях n . Однак, якщо припустити незалежність подій $S_i, i = 1, \dots, n$ при відомому D , то отримаємо:

$$p(S_1, S_2, \dots, S_n | D) = p(S_1 | D) p(S_2 | D) \dots p(S_n | D).$$

У результаті подальшого нормування можна позбутися знаменника $p(S_1, S_2, \dots, S_n)$, що спрощує задачу формування висновку. Таким чином, отримуємо рівняння для формування висновку за теоремою Байєса:

$$p(D | S_1, S_2, \dots, S_n) = \alpha p(D) p(S_1 | D) p(S_2 | D), \dots, p(S_n | D).$$

Це рівняння можна подати графічно, як показано на рис. 10.1. На графі змінні подано колами, а стрілки вказують на зв'язок (умовні ймовірності) між незалежними і залежними змінними. Незалежні змінні називають батьківськими або попередниками, а залежні – дитячими або нащадками.

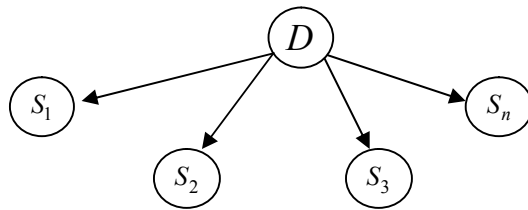


Рис. 10.1 Проста мережа Байєса

Задачу розпізнавання шуканого образу двох кіл також можна представити у вигляді простої БМ, представленої на рис. 10.2. Зазначимо, що використання деревовидної структури дає можливість точніше врахувати вплив кожного члена міри наближення деякого довільного образу до шуканого зображення. Відповідні змінні описані у табл. 10.1, а висновок можна сформулювати за виразом:

$$p(M | S, D, F) = \alpha p(M) p(S | M) p(D | M) p(F | M).$$

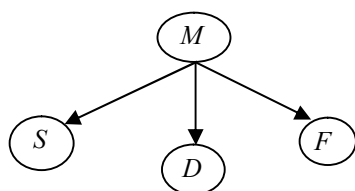


Рис. 10.2 Проста БМ для розпізнавання шуканого образу

Таблиця 10.1 Опис змінних простої БМ для розпізнавання образу

Змінна	Інтерпретація	Тип	Значення
M	міра подібності до шуканого образу	дискретна (2 значення)	істина або фальш
S	відстань між центрами кіл	неперервна	$(S - 2(r_i + r_j))/r_i$
D	різниця в розмірі кіл	неперервна	$ (r_i - r_j)/r_i $
F	колір фону навколо кіл	дискретна (20 значень)	за наближеною гістограмою пікселів для відтінків кольорів

За оцінку кольору фону навколо кіл, які представляють шуканий образ, можна взяти гістограму пікселів для відтінків кольорів в безпосередній близькості до кіл. Це може бути дискретна змінна, яка приймає обмежене число значень. З іншого боку, відстань між очами – це неперервна змінна, хоча точність її виміру можна обмежити точністю розміру пікселя. Можна дещо змінити вираз для визначення ступеня рознесення кіл у просторі, наприклад, можна ввести додатні та від’ємні значення (шляхом видалення модуля):

$$\text{Рознесення кіл} = \frac{S_i - 2(r_i + r_j)}{r_i} = \frac{2r_i - 2r_i - 2r_j}{r_i} \approx -2,$$

при $r_i \approx r_j$ та $S_i = 2r_i$. Це приведе до того, що міра рознесення кіл буде змінюватись приблизно від $-2,0$ (кола розташовані дуже близько, $S_i = 2r_i$) до 2 (кола знаходяться далеко одне від одного, $S_i = 3,0r_i$). Діапазон значень змінної «рознесення кіл» можна поділити на будь-яке число станів, але для ілюстрації зупинимось на таких 7 станах:

$$\{\text{менше } -2,0\}, \{-2,0 \div (-1,5)\}, \{-1,5 \div (-1,0)\}, \{-1,0 \div (-0,5)\}, \\ \{-0,5 \div 0\}, \{0 \div 0,5\}, \{\text{більше } 0,5\}.$$

Залежно від конкретної постановки задачі кількість станів змінної можна визначати різними способами.

10.1.4 Подання і оброблення БМ

Кожній дузі БМ ставиться у відповідність матриця зв'язку як матриця умовних ймовірностей. Матриця, яка зв'язує вузол D з вузлом M для кожної пари станів має наступний вигляд:

$$\mathbf{P}(D|M) = \begin{bmatrix} p(d_1|c_1) & p(d_1|c_2) \\ p(d_2|c_1) & p(d_2|c_2) \\ p(d_3|c_1) & p(d_3|c_2) \\ p(d_4|c_1) & p(d_4|c_2) \end{bmatrix}.$$

Значення елементів матриць умовних ймовірностей можна знайти експериментально. Для цього необхідно мати результати великого числа дослідів з відомими значеннями всіх змінних. Їх можна отримати шляхом цифрової обробки реальних образів для вузлів-нащадків (іншими словами листових вузлів) S, D і F плюс експертний висновок щодо вузла M .

Отримані таким чином матриці зв'язку являють собою об'єктивні ймовірності, які визначаються так:

$$p(d_3|c_1) = (\text{Число разів появи в образі } d_3 \text{ і } c_1) / (\text{Загальне число разів появи } c_1).$$

Очевидно, що навіть для даного простого прикладу кількість значень умовних ймовірностей буде значним. Тому для отримання прийнятних оцінок умовних ймовірностей необхідно мати великі масиви даних.

БМ, що розглядається в даному прикладі, називають по різному: класифікатор Байєса, наївний класифікатор Байєса та проста БМ. Це проста і зручна форма мережі, яка знаходить застосування у багатьох практичних задачах.

Для того щоб скористатись цією мережею, необхідно задати значення змінних, представлених вузлами. Задавання значень вузлам (змінним) називають інстанціюванням.

Формування висновку за допомогою мережі, поданої на рис. 10.2, можливе після того, як задані значення змінних S, D і F за допомогою інформації (вимірів), що міститься в образі, та вироблених правил дискретизації змінних, як показано вище.

Для отримання висновку необхідно перемножити значення всіх умовних ймовірностей для кожного стану M , які беруть з матриць зв'язку. Далі необхідно нормувати результат таким чином, щоб сума умовних

ймовірностей дорівнювала 1. Таким чином отримаємо ймовірність появи шуканого образу з двох кіл в експериментальних даних.

Звичайно, що змінні, які входять до мережі, можуть бути взаємозалежними. Так, для прикладу з розпізнаванням зображення двох кіл, змінні S = «рознесення кіл» та D = «різниця в розмірі кіл» можуть бути в деякій мірі корельованими. Зокрема, можна виставити контраргументи проти того, що S і D – це дійсно ті змінні, які можна використати для встановлення факту наявності двох шуканих кіл в деякому образі. Тобто ідея розпізнавання може бути сформульована дещо по іншому.

10.1.5 Байєсове дерево прийняття рішень

Розглянемо ускладнену мережу, представлену на рис. 10.3. Ця структура є кращою моделлю процесу розпізнавання, оскільки вона містить нову семантичну одиницю (вузол) «кола». Тобто такий елемент може бути виявлений в образі, але він не обов'язково зумовлений появою шуканого зображення. Тепер вузол «кола» можна розглядати як загальну причину введення вузлів S = «рознесення кіл» та D = «різниця в розмірі кіл», що дає можливість не розглядати проблему їх можливої залежності.

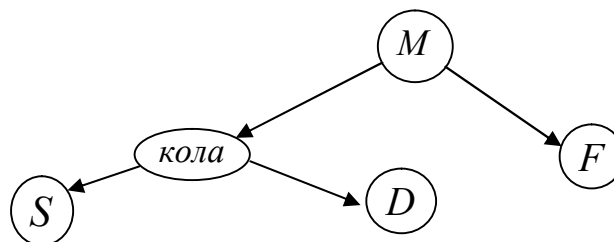


Рис. 10.3 Байєсове дерево прийняття рішень

На рис. 10.3 вузли M і $Кола$ мають матрицю зв'язку $\mathbf{P}(Кола|M)$; вузли M і F – матрицю $\mathbf{P}(F|M)$; вузли $Очі$ і S – матрицю $\mathbf{P}(S|Кола)$, а вузли $Кола$ і D – матрицю $\mathbf{P}(D|Кола)$. Для нового вузла необхідно встановити число його станів. В самому простому випадку – це дихотомічна змінна із двома станами, але в даному випадку краще ввести три таких стани: o_1 = «ймовірно це не шукані кола»; o_2 = «це можуть бути шукані кола» та o_3 = «ймовірно це шукані кола». Значення елементів матриці зв'язку можна знайти за експериментальними даними, але в даному випадку необхідно отримати експертну оцінку стосовно значення нетермінального вузла O та вузла M , за яким формується гіпотеза.

Продемонструємо роботу мережі, починаючи з вузла O («кола»). За теоремою Байєса маємо:

$$p(O|S,D) = \frac{p(O)p(S|O)p(D|O)}{p(S)p(D)}.$$

Однак, тут виникає проблема визначення ймовірності $p(O)$ – апріорної ймовірності появи шуканих кіл у образі. У даному випадку O є проміжною змінною, що не вимірюється, але ймовірності її значень необхідно знати. Ми можемо обчислити правдоподібність значення O за умови, що S і D отримують деякі значення, тобто можна записати:

$$l(O|S,D) = \frac{p(S|O)p(D|O)}{p(S)p(D)},$$

або у простішій формі:

$$l(O) = \alpha p(S|O) p(D|O).$$

Як і раніше, значення $p(S)$ і $p(D)$ можна виключити з розгляду шляхом нормування суми значень $l(O)$ до одиниці. Обчислена таким способом правдоподібність – це ймовірність, яка обчислена за припущенням, що апіорні ймовірності кожного стану змінної O є однаковими, тобто

$$p(o_1) = p(o_2) = p(o_3) = 1/3.$$

Тепер для кореневого вузла M можна записати:

$$p(M|O,F) = \frac{p(M) p(O|M) p(F|M)}{p(O) p(F)},$$

або простіше:

$$p(M|O,F) = \alpha p(M) p(O|M) p(F|M).$$

Якщо відоме значення (вимір) F , наприклад, $F = f_5$, то з матриці зв'язку можна визначити $p(F|M)$. Однак ми не маємо значення стану змінної O , а тільки оцінку правдоподібності для неї: $l(O)$, яка є елементом розподілу можливих станів змінної O . Для того щоб знайти оцінку $p(O|M)$, необхідно знайти середнє цього розподілу; це можна зробити так:

$$p(o|m_1) = p(o_1|m_1)l(o_1) + p(o_2|m_1)l(o_2) + p(o_3|m_1)l(o_3),$$

$$p(o|m_2) = p(o_1|m_2)l(o_1) + p(o_2|m_2)l(o_2) + p(o_3|m_2)l(o_3).$$

Тепер можна обчислити розподіл ймовірностей для M :

$$p'(m_1) = p(m_1 | O, f_5) = \alpha p(m_1) \{p(o_1 | m_1)l(o_1) + p(o_2 | m_1)l(o_2) + p(o_3 | m_1)l(o_3)\} p(f_5 | m_1),$$

$$p'(m_2) = p(m_2 | O, f_5) = \alpha p(m_2) \{p(o_1 | m_2)l(o_1) + p(o_2 | m_2)l(o_2) + p(o_3 | m_2)l(o_3)\} p(f_5 | m_2),$$

де p' – середня апостеріорна ймовірність.

Середня апостеріорна ймовірність p' – є ймовірність прийняття змінною деякого значення за умови, що відома деяка інформація (в даному випадку це значення F, S і D).

Хоча ми не маємо апріорної ймовірності для вузла O , її можна оцінити за допомогою апріорної (або апостеріорної) ймовірності для M та матриці зв'язку $\mathbf{P}(O|M)$. У векторній формі це рівняння має вигляд: $\mathbf{p}(O) = \mathbf{P}(O|M)\mathbf{p}(M)$.

На відміну від наведеної вище теореми Байєса (у скалярній формі), це векторне рівняння, тобто $p(o_1) \neq p(o_1 | m_2) p(m_2)$. Припустимо, що $\mathbf{p}(M) = \{0,4 \ 0,6\}$; це означає, що

$$\mathbf{p}(O) = \begin{bmatrix} p(o_1 | m_1) & p(o_1 | m_2) \\ p(o_2 | m_1) & p(o_2 | m_2) \\ p(o_3 | m_1) & p(o_3 | m_2) \end{bmatrix} \begin{bmatrix} 0,4 \\ 0,6 \end{bmatrix} = \begin{bmatrix} 0,4 p(o_1 | m_1) + 0,6 p(o_1 | m_2) \\ 0,4 p(o_2 | m_1) + 0,6 p(o_2 | m_2) \\ 0,4 p(o_3 | m_1) + 0,6 p(o_3 | m_2) \end{bmatrix}.$$

Оскільки суми елементів стовпчиків матриці зв'язку дорівнюють 1, то цей результат відноситься також до обчислених значень $\mathbf{p}(O)$.

Тепер можна обчислити розподіл ймовірностей для значень станів змінної O , за умови, що є виміри, скажемо, $\{s_3, d_2\}$:

$$\begin{aligned} p(o_1 | s_3, d_2) &= \alpha p(o_1) p(s_3 | o_1) p(d_2 | o_1), \\ p(o_2 | s_3, d_2) &= \alpha p(o_2) p(s_3 | o_2) p(d_2 | o_2), \\ p(o_3 | s_3, d_2) &= \alpha p(o_3) p(s_3 | o_3) p(d_2 | o_3), \end{aligned}$$

а той факт, що $p(o_1 | s_3, d_2) + p(o_2 | s_3, d_2) + p(o_3 | s_3, d_2) = 1$, дозволяє виключити з розгляду α .

Очевидно, що наведена процедура обчислення ймовірностей та формування висновку є досить складною і громіздкою, а при збільшенні розмірів мережі вона стає недосяжною для сприймання. Тобто виникає необхідність розробки спеціальних методів та відповідних обчислювальних алгоритмів для виконання подібних розрахунків. До таких методів належать евристичні методи побудови БМ.

10.2 Евристичний метод побудови БМ

10.2.1 Складність побудови БМ

Побудову БМ можна виконати простим перебором множини усіх можливих нециклічних графічних моделей та вибрати з них ту, що з максимальною адекватністю відповідає експериментальним (навчальним) даним. Ця задача є NP-складною, оскільки при повному переборі кількість усіх моделей дорівнює $3^{\frac{n(n-1)}{2}} - k_{cycle}$, де n – число вершин; k_{cycle} – число моделей з циклами. Число усіх можливих нециклічних моделей можна порахувати за рекурсивною формулою Робінсона [62]:

$$f(n) = \sum_{i=1}^n (-1)^{i+1} \cdot C_n^i \cdot 2^{i(n-i)} \cdot f(n-i),$$

де n – число вершин, а $f(0) = 1$.

10.2.2 Спрощений евристичний метод

Виконати повний перебір можливих структур моделей можна тільки для мереж, які містять не більше семи вузлів. Якщо кількість вузлів перевищує 7, то виконати простий перебір практично неможливо, оскільки при виконанні обчислень на звичайних персональних комп'ютерах не вистачає обчислювальних ресурсів. Тому для побудови мережі пропонується спрощений евристичний метод [42], який полягає у виконанні таких кроків: (1) обчислення так званої взаємної інформації між усіма вершинами за допомогою експериментальних даних; (2) виконання цілеспрямованого пошуку з використанням оціночної функції на основі принципу опису мінімальної довжини (ОМД); (3) повторення ітерацій до отримання структури мережі заданої якості.

Для оцінювання ступеня залежності двох довільних випадкових змінних x^i і x^j Чау і Ліу [5] запропонували використовувати значення взаємної інформації $MI(x^i, x^j)$, яка обчислюють за виразом:

$$MI(x^i, x^j) = \sum_{x^i, x^j} p(x^i, x^j) \cdot \log \left(\frac{p(x^i, x^j)}{p(x^i) \cdot P(x^j)} \right).$$

За своєю суттю взаємна інформація є деяким аналогом кореляції, але за змістом – це оцінка кількості інформації, що міститься в змінній x^i про змінну x^j . Взаємна інформація приймає невід'ємні значення, $MI(x^i, x^j) \geq 0$,

а у випадку, якщо вершини x^i і x^j є повністю незалежними одна від одної, то $MI(x^i, x^j) = 0$, оскільки $p(x^i, x^j) = p(x^i) \cdot p(x^j)$ і

$$\log\left(\frac{p(x^i, x^j)}{p(x^i) \cdot p(x^j)}\right) = \log\left(\frac{p(x^i) \cdot p(x^j)}{p(x^i) \cdot p(x^j)}\right) = \log(1) = 0.$$

У випадку, коли мережа Байєса складається з N вершин, для обчислення $MI(x^i, x^j)$ для всіх можливих пар x^i і x^j необхідно виконати $\frac{N \cdot (N-1)}{2}$ обчислень, при цьому $MI(x^i, x^j) = MI(x^j, x^i)$.

10.2.3 Принцип формування опису БМ мінімальної довжини

Принцип формування опису БМ мінімальної довжини (ОМД) ґрунтується на теорії Шеннона. Згідно з теорією кодування Шеннона, за умови відомого розподілу $P(X)$ випадкової величини X довжина оптимального коду для передачі конкретного значення x через канал зв'язку прямує до значення $L(x) = -\log P(x)$.

Ентропія джерела $S(P) = -\sum_x P(x) \cdot \log P(x)$ є мінімальною очікуваною довжиною закодованого повідомлення. Будь-який інший код, який ґрунтується на неправильному представленні про джерело повідомлення, призведе до більшої очікуваної довжини повідомлення. Іншими словами, чим кращою є модель джерела, тим компактнішими можуть бути закодовані дані.

В задачі навчання мережі джерелами даних є деяка невідома істинна функція розподілу

$$P(D|h_0),$$

де $D = \{d_1, \dots, d_N\}$ – набір даних;

h – гіпотеза щодо ймовірнісного походження даних;

$L(D|h) = -\log P(D|h)$ – емпіричний ризик, який є адитивним щодо числа спостережень і пропорціональним емпіричній похибці.

Відмінність між $P(D|h_0)$ і модельним розподілом $P(D|h)$ за мірою Кульбака-Левлера визначається так:

$$|P(D|h) - P(D|h_0)| = \sum_D P(D|h_0) \cdot \log \frac{P(D|h_0)}{P(D|h)} = \sum_D P(D|h_0) \cdot |L(D|h) - L(D|h_0)| \geq 0.$$

Тобто це є різниця між очікуваною довжиною коду даних, отриманою за допомогою гіпотези та мінімально можливою довжиною. Ця різниця є завжди невід'ємною і дорівнює нулю лише у випадку повного співпадання двох розподілів. Іншими словами, гіпотеза буде тим кращою, чим коротшою є середня довжина коду даних [5]. Принцип ОМД у своєму нестрогому і найбільш загальному формулюванні проголошує: з множини можливих моделей-кандидатів необхідно вибрати ту, яка дозволяє описати дані найбільш коротко і без втрат інформації [5, 42].

У загальному вигляді задача формування ОМД формулюють так: спочатку задається множина навчальних даних $D = \{d_1, \dots, d_n\}$, $d_i = \{x_i^{(1)} x_i^{(2)} \dots x_i^{(N)}\}$ (нижній індекс – номер спостереження, а верхній – номер змінної), n – число спостережень; кожне спостереження складається з N ($N \geq 2$) змінних $X^{(1)}, X^{(2)}, \dots, X^{(N)}$. Кожна j -я змінна ($j = 1, \dots, N$) має $A^{(j)} = \{0, 1, \dots, \alpha^{(j)} - 1\}$ ($\alpha^{(j)} \geq 2$) станів, а кожен структуру $g \in G$ БМ подають N множинами предків ($\Pi^{(1)}, \dots, \Pi^{(N)}$), тобто для кожної вершини $j = 1, \dots, N$,

$\Pi^{(j)}$ – це множина батьківських вершин, така, що $\Pi^{(j)} \subseteq \{X^{(1)}, \dots, X^{(N)}\} \setminus \{X^{(j)}\}$ (вершина не може бути предком самої себе, тобто петлі у графі відсутні).

Таким чином, ОМД структури $g \in G$ при заданій послідовності з n спостережень $x^n = d_1 d_2 \dots d_n$ обчислюють за виразом:

$$L(g, x^n) = H(g, x^n) + \frac{k(g)}{2} \cdot \log(n),$$

де $k(g)$ – кількість незалежних умовних ймовірностей у мережевій структурі g ;

$H(g, x^n)$ – емпірична ентропія.

Емпіричну ентропію визначають за формулою

$$H(g, x^n) = \sum_{j \in J} H(j, g, x^n), \quad k(g) = \sum_{j \in J} k(j, g),$$

де ОМД j -ї вершини

Значення ОМД j -ї вершини обчислюють за формулою:

$$L(j, g, x^n) = H(j, g, x^n) + \frac{k(j, g)}{2} \cdot \log(n);$$

де $k(j, g)$ – кількість незалежних умовних ймовірностей j -ї вершини.

Значення $k(j, g)$ визначають за виразом:

$$k(j, g) = (\alpha^{(j)} - 1) \cdot \prod_{k \in \phi(j)} \alpha^k,$$

де $\phi(j) \subseteq \{1, \dots, j-1, j+1, \dots, N\}$ – така множина, що $\Pi^{(j)} = \{X^{(k)} : k \in \phi(j)\}$.

Емпіричну ентропію j -ї вершини обчислюють за виразом:

$$H(j, g, x^n) = \sum_{s \in S(j, g)} \sum_{q \in A^{(j)}} -n[q, s, j, g] \cdot \log \frac{n[q, s, j, g]}{n[s, j, g]},$$

$$n(s, j, g) = \sum_{i=1}^n I(\pi_i^{(j)} = s) \quad ; \quad n[q, s, j, g] = \sum_{i=1}^n I(x_i = q, \pi_i^{(j)} = s),$$

де $\pi^{(j)} = \Pi^{(j)}$ означає $X^{(k)} = x^{(k)}, \forall k \in \phi(j)$.

Функція $I(E) = 1$ коли предикат $E = true$, інакше $I(E) = 0$.

10.2.4 Простий алгоритм навчання БМ з використанням опису мінімальної довжини

Простий алгоритм навчання БМ з використанням ОМД будують так: циклічно виконується перебір всіх можливих нециклічних мережевих структур. В g^* зберігається оптимальна мережева структура. Оптимальною структурою буде та, для якої функція $L(g, x^n)$ приймає найменше значення.

Простий алгоритм навчання БМ з використанням ОМД має такий вигляд:

1. $g^* \leftarrow g_0 (\in G)$;
2. для $\forall g \in G - \{g_0\}$: якщо $L(g, x^n) < L(g^*, x^n)$ то $g^* \leftarrow g$;
3. за розв'язок приймають g^* .

Приклад використання методу ОМД. Нехай є 10 спостережень для навчання БМ (табл. 10.2).

Таблиця 10.2 Десять спостережень для навчання БМ

n	$X^{(1)}$	$X^{(2)}$	$X^{(3)}$	n	$X^{(1)}$	$X^{(2)}$	$X^{(3)}$
1	0	1	1	6	0	1	1
2	1	0	0	7	1	0	1
3	0	1	1	8	1	0	0
4	1	0	0	9	0	1	1
5	0	1	1	10	1	1	1

У випадку повного перебору всіх можливих структур необхідно розглянути 25 структур. Після того, як будуть розглянуті всі 25 структур, за оптимальну буде вибрана структура, зображена на рис. 10.4.

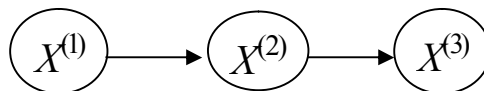


Рис. 10.4 Оптимальна структура, що відповідає таблиці 10.3

Довжину опису цієї структури обчислюють таким чином. Вершина $X^{(1)}$ не має предків, тобто $\Pi^{(1)} = \{\}$.

Емпіричну ентропію обчислюють за виразом:

$$H(j=1, g) = -5 \cdot \log\left(\frac{5}{10}\right) - 5 \cdot \log\left(\frac{5}{10}\right) = 6,9315,$$

а кількість незалежних умовних ймовірностей дорівнює $k(j=1, g) = 2 - 1 = 1$.

Таким чином, довжина опису вершини $X^{(1)}$ дорівнює $L(1, g) = 6,9315 + \frac{1}{2} \cdot \log(10) = 8,0828$.

При обчисленні можна використовувати логарифм с будь-якою основою; в даному прикладі використано основу $e = 2,7183$, тобто натуральний логарифм.

Вершина $X^{(2)}$ має одного предка $X^{(1)}$, тобто $\Pi^{(2)} = \{X^{(1)}\}$. Емпірична ентропія:

$$H(j=2, g) = \left(-0 \cdot \log\left(\frac{0}{5}\right) - 5 \cdot \log\left(\frac{5}{5}\right) \right) + \left(-4 \cdot \log\left(\frac{4}{5}\right) - 1 \cdot \log\left(\frac{1}{5}\right) \right) = 2,502,$$

а кількість незалежних умовних ймовірностей: $k(j=2, g) = (2-1) \cdot 2 = 2$.

Довжина опису вершини $X^{(2)}$ дорівнює:

$$L(2, g) = 2,502 + \frac{2}{2} \cdot \log(10) = 4,8046 \text{ (табл. 10.3 – 10.5).}$$

Таблиця 10.3 Таблиця значень параметрів вершини $X^{(1)}$

$X^{(1)}$	$n[q, s, j, g]$	$n[s, j, g]$
0	5	10
1	5	

Таблиця 10.4 Таблиця значень параметрів вершини $X^{(2)}$

$X^{(1)}$	$X^{(2)}$	$n[q, s, j, g]$	$n[s, j, g]$
0	0	0	5
0	1	5	
1	0	4	5
1	1	1	

Таблиця 10.5 Таблиця значень параметрів вершини $X^{(3)}$

$X^{(2)}$	$X^{(3)}$	$n[q, s, j, g]$	$n[s, j, g]$
0	0	3	4
0	1	1	
1	0	0	6
1	1	6	

Вершина $X^{(3)}$ має одного предка $X^{(2)}$, тобто $\Pi^{(3)} = \{X^{(2)}\}$; емпірична ентропія:

$$H(j=3, g) = \left(-3 \cdot \log\left(\frac{3}{4}\right) - 1 \cdot \log\left(\frac{1}{4}\right) \right) + \left(-0 \cdot \log\left(\frac{0}{6}\right) - 6 \cdot \log\left(\frac{6}{6}\right) \right) = 2.2493,$$

а кількість незалежних умовних ймовірностей:

$$k(j=3, g) = (2-1) \cdot 2 = 2.$$

Довжина опису вершини $X^{(3)}$ дорівнює:

$$L(3, g) = 2.2493 + \frac{2}{2} \cdot \log(10) = 4.5519.$$

Тобто довжина опису структури g , представленої на рис. 5.4, складає:

$$H(g, x^n) = \sum_{j=1}^3 H(j, g, x^n) = 17.4393.$$

10.2.5 Евристичний алгоритм побудови БМ

Вихідними даними для алгоритму є: навчальна вибірка $D = \{d_1, \dots, d_n\}$, $d_i = \{x_i^{(1)} x_i^{(2)} \dots x_i^{(N)}\}$ (нижній індекс – номер спостереження, а верхній – номер змінної), n – число спостережень; N – число вершин (змінних).

Алгоритм виконують у два етапи.

Перший етап

Для всіх пар вершин обчислюють значення взаємної інформації

$Set_MI = \left\{ MI(x^i, x^j); \forall i, j \right\}$. Після цього елементи множини Set_MI

упорядковують за спаданням:

$$Set_MI = \{MI(x^{m_1}, x^{m_2}), MI(x^{m_3}, x^{m_4}), MI(x^{m_5}, x^{m_6}), \dots\}.$$

Другий етап включає такі три кроки.

Крок 1. З множини значень взаємної інформації Set_MI вибирають перші два максимальних значення $MI(x^{m_1}, x^{m_2})$ и $MI(x^{m_3}, x^{m_4})$.

За отриманим значенням $MI(x^{m_1}, x^{m_2})$ и $MI(x^{m_3}, x^{m_4})$ будують множину моделей G вигляду:

$\{ (m_1 \rightarrow m_2; m_3 \rightarrow m_4), (m_1 \rightarrow m_2; m_3 \leftarrow m_4), (m_1 \leftarrow m_2; m_3 \leftarrow m_4), (m_1 \leftarrow m_2; m_3 \rightarrow m_4), (m_1 \leftarrow m_2; m_3 \text{ не залежить від } m_4), (m_1 \rightarrow m_2; m_3 \text{ не залежить від } m_4), (m_1 \text{ не залежить від } m_2; m_3 \rightarrow m_4), (m_1 \text{ не залежить від } m_2; m_3 \leftarrow m_4), (m_1 \text{ не залежить від } m_2; m_3 \text{ не залежить від } m_4) \}$.

Запис вигляду $m_i \rightarrow m_j$ означає, що вершина x^{m_i} є предком вершини x^{m_j} .

Крок 2. Виконують пошук серед множини моделей G . У параметрі g^* зберігається оптимальна мережева структура. Оптимальною структурою буде та, у якій найменше значення функції; $L(g, x^n)$ – ОМД структури моделі при заданій послідовності з n спостережень $x^n = d_1 d_2 \dots d_n$.

1. $g^* \leftarrow g_0 (\in G)$;
2. для $\forall g \in G - \{g_0\}$: якщо $L(g, x^n) < L(g^*, x^n)$ то $g^* \leftarrow g$;
3. на виході g^* – шукане рішення.

Крок 3. Після того як знайдено оптимальну структуру (структури) g^* з G , з множини значень взаємної інформації Set_MI вибирають максимальне значення: $MI(x^{i_next}, x^{j_next})$.

За отриманим значенням $MI(x^{i_next}, x^{j_next})$ і структурою (структурами) g^* будують множини моделей G вигляду: $\{ (g^*; i_next \rightarrow j_next), (g^*; i_next \leftarrow j_next), (g^*; i_next \text{ не залежить від } j_next) \}$. Перейти на крок 2.

Умовою закінчення процедури пошуку є аналіз певної кількості елементів. Евристичний пошук продовжується до тих пір, поки не буде виконано аналіз визначеної кількості елементів множини або ж всіх $\frac{N \cdot (N - 1)}{2}$ елементів множини Set_MI . Як показує практика, у більшості випадків немає смислу виконувати аналіз більше половини (тобто $\frac{N \cdot (N - 1)}{4}$) елементів множини Set_MI . Результатом роботи алгоритма є оптимальна структура g^*

10.3 Оцінки якості навчання БМ

Для оцінювання якості навчання БМ можна використати кількість зайвих, відсутніх та реверсованих дуг у навченій мережі у порівнянні з оригінальною БМ. За міру похибки навчання можна використати структурну різницю або перехресну ентропію (cross entropy) між навченою БМ та оригінальною мережею. Для обчислення структурної різниці застосовують формулу симетричної різниці структур [5]:

$$\begin{aligned}\delta &= \sum_{i=1}^n \delta_i = \sum_{i=1}^n \text{card}(\Pi^{(i)}(B) \Delta \Pi^{(i)}(A)) = \\ &= \sum_{i=1}^n \text{card}((\Pi^{(i)}(B) \setminus \Pi^{(i)}(A)) \cup (\Pi^{(i)}(A) \setminus \Pi^{(i)}(B))),\end{aligned}$$

де B – навчена БМ;

A – оригінальна БМ;

n – кількість вершин мережі;

$\Pi^{(i)}(B)$ – множина предків i -ї вершини навченої мережі B ;

$\Pi^{(i)}(A)$ – множина предків i -ї вершини оригінальної мережі A ;

$\text{card}(\xi)$ – потужність скінченної множини ξ , яка визначається кількістю елементів, що належать множині ξ .

Перехресна ентропія – це відстань між розподілами навченої БМ та оригінальної БМ. Якщо $p(v)$ – спільний розподіл оригінальної БМ, а $q(v)$ - спільний розподіл навченої БМ, то перехресна ентропія обчислюють так [5]:

$$\begin{aligned}
 H(p, q) &= \sum_v p(v) \cdot \log \frac{p(v)}{q(v)} = \\
 &= \sum_{j \in J} \sum_{s \in S(j, g)} \sum_{a \in A^{(j)}} p(X^{(j)} = a | \Pi^{(j)} = s) \cdot \log \frac{p(X^{(j)} = a | \Pi^{(j)} = s)}{q(X^{(j)} = a | \Pi^{(j)} = s)}
 \end{aligned}$$

10.4 Практичне застосування БМ

10.4.1 Прогнозування Нью-Йоркського біржового композитного індексу

Приклад 10.2. Прогнозування Нью-Йоркського біржового композитного індексу.

За даними, представленими на сайті Нью-Йоркської фондової біржі <http://www.nyse.com> сформована навчальна вибірка. На рис. 10.5 представлена БМ (вона описує композитний індекс), яка побудована за навчальною вибіркою з 500 значень. Варто відзначити, що при первісному аналізі розглядалось більша кількість індексів, але в моделі представлені найбільш значущі індекси для множини навчальних даних, за якими будувалася БМ. На рис. 10.6 представлена загальна структурна схема динамічного прогнозування композитного індексу БМ. При побудові БМ використано евристичний метод побудови, алгоритм якого докладно наведено вище. Для формування ймовірнісного висновку застосовувався

алгоритм формування ймовірного висновку у БМ на основі навчальних даних, який розглянуто в роботі [5]. Наведемо деякі терміни, що мають відношення до задачі.

Композитний індекс – відображає зміни курсів всіх акцій, розміщених на Нью-Йоркській фондовій біржі, що включають у себе більше 1500 найбільших американських компаній.

Фінансовий індекс відображає зміни курсів акцій непривілейованих акції фінансового сектора США та сільськогосподарських підприємств.

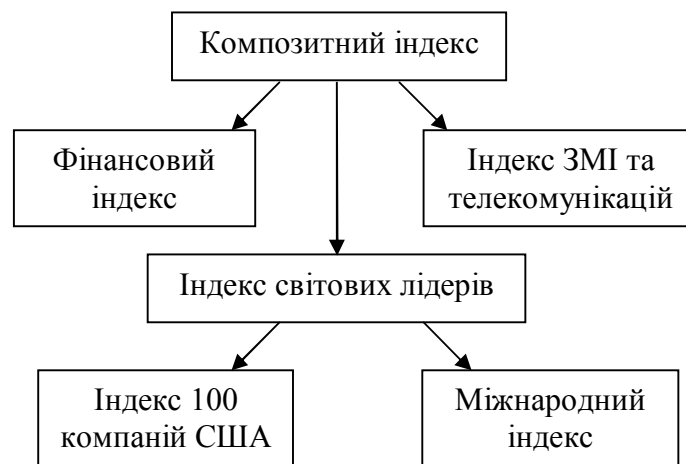


Рис. 10.5 БМ для опису композитного індекса Нью-Йоркської фондової біржі

Індекс ЗМІ та телекомунікацій – відображає зміни 100 компаній лідерів із сектора ЗМІ та телекомунікацій.

Індекс світових лідерів – відображає зміни курсу непривілейованих акцій 200 найбільших компаній світу (включаючи 100 найбільших компаній США).

Індекс 100 компаній США – відбиває зміни курсів непривілейованих акцій 100 найбільших компаній США із загальною капіталізацією 6,7 трильйонів доларів.

Міжнародний індекс – відбиває зміни курсів непривілейованих акцій ста найбільших компаній світу не із США із загальною капіталізацією 5,1 трильйона доларів.

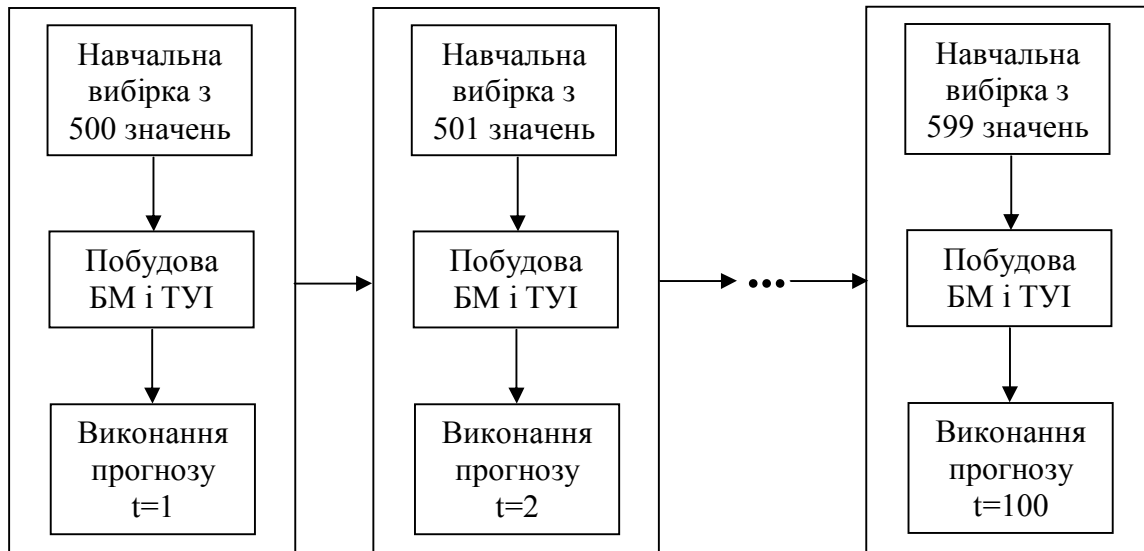


Рис. 10.6 Структурна схема послідовності прогнозування композитного індексу мережею Байєса

При виконанні обчислювального експерименту виконано 100 ітерацій алгоритму прогнозування композитного індексу. В 96% випадків було точно спрогнозовано зростання або спадання значення композитного індексу; у 52% випадків прогнозне та реальне значення відхилення композитного індексу від значення в попередній момент часу повністю збіглися. На рис. 10.7 показана діаграма зміни прогнозного та реального відхилень композитного індексу на перших 50 часових інтервалах.



Рис. 10.7 Діаграма зміни прогнозованого та реального відхилень композитного індексу на 50 часових інтервалах

10.4.2 Застосування БМ для постановки діагнозу

Приклад 10.3. Застосування БМ для постановки діагнозу.

На рис. 10.8 наведена медична БМ, яка показує зв'язок між захворюваннями пацієнта, хірургічним втручанням і станом здоров'я. Ця БМ побудована на основі відомої комп'ютерної бази даних http://www.umdnj.edu/idsweb/idst6122/data/patlos_sample.sav, яка складається з інформації про 1481 пацієнта із серцево-судинними захворюваннями. При побудові структури БМ, представленої на рис. 10.8, виконано аналіз 686 структур на 90 ітераціях евристичним методом [5] за 1 хвилину 11 секунд.

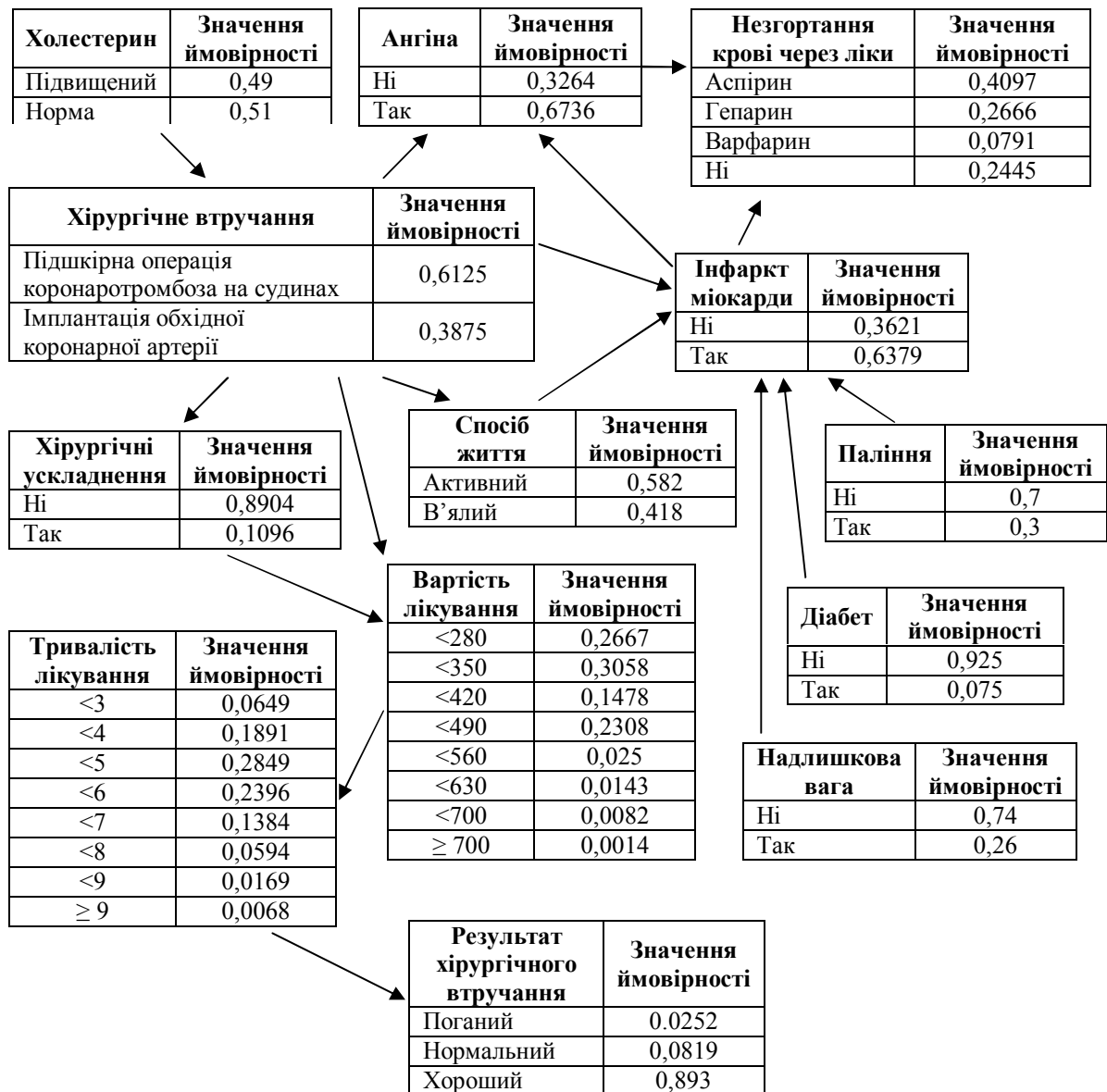


Рис. 10.8 Приклад застосування БМ у медицині

Застосовуючи імовірнісний висновок [5], змодельовано декілька ситуацій. Результати моделювання представлені у вигляді табл. 10.6. Так, у випадку коли пацієнт веде здоровий активний образ життя і при цьому ніколи не був хворим ангіною та діабетом, вірогідність того, що у нього відбудеться інфаркт міокарда дорівнює 5,28%, інакше – 97,89%.

Таблиця 10.6 Результати моделювання за БМ (рис. 10.8)

Номер ситуації	Інстанційовані вершини	Ймовірність настання інфаркту міокарда
1	Незгортання крові через ліки = «Аспірин»	73,80%
2	Незгортання крові через ліки = «Гепарин»	76,69%
3	Незгортання крові через ліки = «Варфарин»	66,63%
4	Незгортання крові через ліки = «Ні»	32,33%
5	Ангіна = «Так»	73,13%
6	Хірургічне втручання = «Підшкірна операція коронаротромбоза на судинах»	56,56%
7	Хірургічне втручання = «Імплантація обхідної коронарної артерії»	75,41%
8	Спосіб життя = «Активний»	55,89%
9	Спосіб життя = «В'ялий»	69,59%
10	Надлишкова вага = «Ні»	60,18%
11	Надлишкова вага = «Так»	74,32%
12	Діабет = «Ні»	62,04%
13	Діабет = «Так»	86,32%
14	Паління = «Ні»	58,23%
15	Паління = «Так»	77,12%
16	Незгортання крові через ліки = «Ні » та Ангіна = «Ні » та Спосіб життя = «Активний» та Надлишкова вага = «Ні» та Діабет = «Ні » та Паління = «Ні »	5,28%
17	Незгортання крові через ліки = «Гепарин » та Ангіна = «Так» та Спосіб життя = «В'ялий» та Надлишкова вага = «Так» та Діабет = «Так » та Паління = «Так »	97,89%

Таким чином, моделювання процесів різної природи за допомогою БМ – перспективний сучасний напрям в галузі інтелектуального аналізу даних. Основні напрями застосування: – розпізнавання образів; автоматична діагностика у медицині, технічних системах і технологічних процесах; прогнозування динаміки розвитку процесів. Головна ідея побудови БМ – коректне створення спрямованого ациклічного графа, що відображає причинно-наслідкові зв'язки процесу. Проста БМ може бути побудована на основі експертних оцінок стосовно існування причинних зв'язків між змінними процесу та оцінок апріорних ймовірностей.

Однак у складних випадках структура мережі ґрунтується на використанні значних об'ємів експериментальних даних, на основі яких формуються зв'язки між змінними і таблиці умовних ймовірностей. За допомогою чисельних експериментів встановлено, що швидкість побудови оптимальної БМ за евристичним методом залежить, головним чином, від кількості вершин у мережі.

Якщо число вершин (вузлів) мережі перевищує 7, то виконати простий перебір варіантів структури практично неможливо. Тому запропоновано метод побудови структури на основі евристичного підходу, який ґрунтується на обчисленні взаємної інформації. Наведений вище ітераційний алгоритм евристичного методу побудови БМ дозволяє значно зменшити обчислювальну складність побудови БМ.

Відомо, що у більшості випадків навчання мережі немає необхідності виконувати аналіз більше половини елементів множини даних. За міру якості побудованої мережі використано структурну різницю, яка є цілим числом. Це число є кумулятивною величиною, яка враховує відсутні, зайві та реверсовані дуги спрямованого ациклічного графа. Наведені приклади застосування методу побудови мережі свідчать про його високу ефективність у оптимізації структури.

10.5 Інформаційні СППР на основі БМ

10.5.1 Передумови застосування сучасних технологій обробки даних

Для прийняття обґрунтованих об'єктивних рішень на всіх рівнях управління виробничими, фінансово-економічними та іншими процесами довільної природи необхідно впроваджувати комп'ютерні інформаційні системи у всі ланки ділових стосунків між суб'єктами господарювання. Сучасна економічна та соціально-політична обстановка характеризується

швидкими змінами ринкової та регуляторної інформації, яка стосується асортименту і якості продукції, балансу попиту і пропозиції, кон'юнктури ринку, рівня оподаткування тощо.

Однак, рівень автоматизації обробки даних практично на всіх рівнях реалізації виробничих процесів характеризується, на сьогодні, досить низькими показниками. Відсутність інформаційних систем широкого профілю призводить до неможливості застосування сучасних інтелектуальних методів і систем до аналізу та моделювання ситуацій, прогнозування подальшого розвитку процесів та формування оптимальних або раціональних управлінських рішень.

Умовою успішного застосування сучасних технологій обробки даних і прийняття рішень є чітке усвідомлення задач, які має розв'язати підприємство в процесі розробки та впровадження інформаційно-аналітичних систем (ІАС) або систем підтримки прийняття рішень. В роботі [43] розглянуто основи побудови ІАС для накопичення та обробки економетричних даних та створення на їх основі СППР. Однак, в цій роботі наведено тільки відомі класичні методи статистичної обробки даних. В роботі [35] розглянуто основні принципи побудови СППР з акцентуванням на відповідних інформаційних технологіях.

Типова схема процесу автоматизації обробки даних і підтримки прийняття рішень на підприємстві виглядає на перший погляд тривіально і включає такі етапи: (1) автоматизація окремої сфери діяльності підприємства; (2) автоматизація суміжних сфер; (3) повна автоматизація бізнес-системи [35]. Ключовим моментом цього процесу є зростаюча складність та ієрархічність переходу від попереднього до наступного етапу. Саме внаслідок несумісності розроблених систем в одній сфері з аналогічними розробками у суміжних галузях найчастіше призводить до зупинки діяльності на першому етапі.

Наприклад, на розробці і впровадженні системи автоматизації бухгалтерської діяльності. Автоматизація всієї галузі (галузей) в цілому ставить додаткові задачі, наприклад, розв'язання проблеми забезпечення надійності та конфіденційності інформації при розподілі ролей між ланками підприємства та розширення ролі типової облікової інформаційної системи до рівня потужної системи підтримки прийняття рішень. Виходом з цієї ситуації є створення єдиної інформаційної системи підприємства, яка має можливість інтегрування в галузеву систему. Це означає, що має існувати єдина база даних (БД) з уніфікованими форматами подання даних, мають бути спроектовані і створені сучасні адаптивні інтерфейси для роботи з БД та системою в цілому, а також має бути створена СППР, яка використовує інформацію з цієї бази даних [12, 23]. При такому підході створюються сприятливі умови для отримання інформації в стандартизованому представленні, що дає можливість обробляти її сучасними аналітичними методами.

Таким чином можна створити реальні передумови для впровадження в діяльність підприємства формалізованих СППР на основі сучасних методів побудови математичних моделей і прийняття рішення. Надалі ті ж дані і алгоритмічні процедури можуть стати основою для бази знань потужнішої інтелектуальної системи – експертної системи для широкого кола задач на галузевому рівні.

Тобто повноцінна комп'ютеризація діяльності суб'єкта бізнесу спирається на ядро ІАС у вигляді бази даних, інтерфейсної частини для організації інтерактивної взаємодії користувач-комп'ютер, належної множини процедур для обробки даних і системи подання результатів у зручній для сприйняття формі.

10.5.2 Технологія розробки інформаційної системи на основі БМ

Ставиться задача побудови комп'ютерної інформаційної системи для оцінювання і прогнозування стану підприємства на основі сучасних ймовірнісних методів моделювання процесів та формування статистичного висновку (прийняття рішень). Для розв'язання задачі необхідно синтезувати модель у вигляді мережі Байєса на множині зв'язаних подій $X_i, i=1, \dots, n$ з відомими апіорними ймовірностями, тобто створити методику побудови ациклічного графа G , що характеризується множиною ймовірнісних параметрів B ; дослідити характеристики отриманої мережі з метою її подальшого застосування до аналізу та оцінювання стану бізнес-процесів на виробництві. При цьому вважаємо, що на події $X_i, i=1, \dots, n$ можуть впливати невизначеності різного характеру і природи; також існують дані, що описують події, зв'язані з виробничою діяльністю підприємства.

Розробка інформаційної СППР на основі мережі Байєса відповідає актуальному напрямку розвитку СППР з інтелектуалізацією процесів обробки даних. Інтелектуалізація СППР означає надання користувачеві принципової можливості отримувати нову інформацію на основі поглибленого інтелектуального аналізу даних за множиною взаємодоповнюючих методів і використовувати цю інформацію разом з накопиченими професіоналами досвідом і знаннями. Крім того, інтелектуалізація означає активне застосування методів обробки даних, що ґрунтуються на нейронних і байєсівських мережах, деревах рішень, методах нечіткої логіки, м'яких обчисленнях тощо.

Під інформаційною СППР будемо розуміти комп'ютерну інформаційну систему, яка надає будь-яку допомогу ОПР при прийнятті

рішень виробничого або ділового характеру. Тобто це допомога у аналізі (в тому числі візуальному) даних, у пошуку кращих моделей процесів, в обчисленні альтернативних рішень тощо.

Різновидом СППР є експертні системи (ЕС) [18], які поєднують у собі можливості комплексного використання експертних оцінок та результатів аналітичної обробки даних. Існують певні технології розробки ЕС, що складаються з таких шести етапів (рис. 10.9): ідентифікація, концептуалізація, формалізація, реалізація, тестування і дослідно-експериментальна експлуатація.

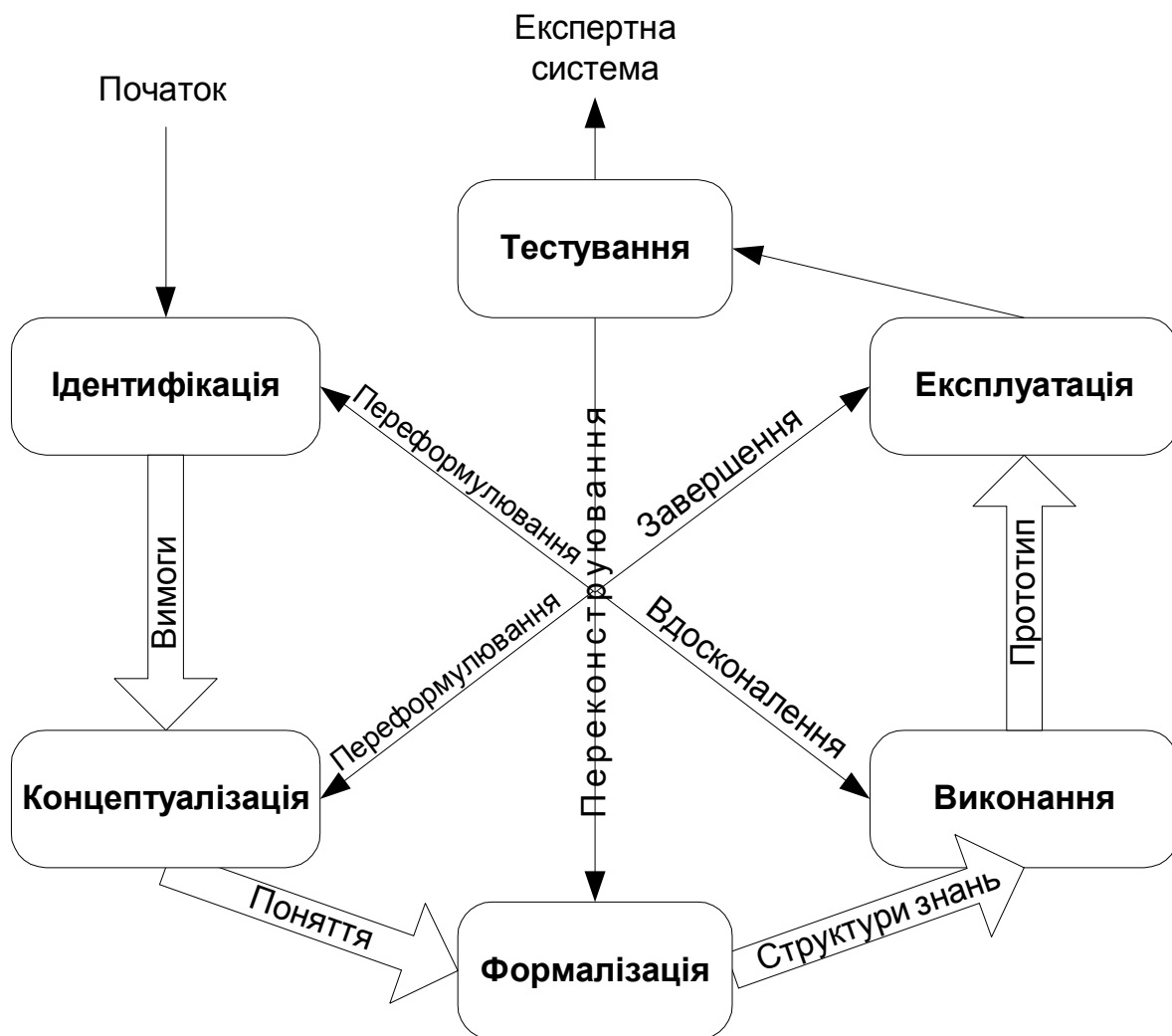


Рис. 10.9 Технологія розробки ЕС

На етапі ідентифікації визначаються задачі, які підлягають розв'язанню, виявляються проміжні цілі розробки, визначаються експерти за напрямом і типи користувачів.

Етап концептуалізації призначений для змістовного аналізу проблемної області, виявлення наявної інформації та визначення множини альтернативних методів розв'язання поставлених задач.

На етапі формалізації обирається інструментарій і визначаються способи зберігання та подання всіх типів знань, формалізуються основні поняття, визначаються способи інтерпретації знань, моделюється робота системи, оцінюється адекватність понять, методів розв'язання, засобів подання і маніпулювання знаннями.

На етапі виконання здійснюється наповнення експертом бази знань. Розповсюджені такі підходи до розробки ЕС: системи на основі правил, системи з використанням нейронних мереж та нечіткої логіки (або нейронечіткі системи), експертні системи на основі мереж довіри Байєса та інші.

10.5.3 Системи на основі БМ

Мережі довіри Байєса, або просто БМ, складаються з множини вузлів і сукупності спрямованих ребер, що з'єднують ці вузли між собою [5, 51]. Ребра визначають причинно-наслідкові зв'язки у предметній області, що більшою частиною не завжди визначені однозначно. Вірогідність твердження (або події) представляється за допомогою ймовірностей.

Концепція байєсових мереж полягає у оновленні ймовірностей подій (станів процесу або системи) при надходженні додаткової інформації. Математичним підґрунтям цього процесу обробки даних є теорема Байєса. На відміну від систем, заснованих на правилах, метод оновлення

ймовірностей у БМ є фундаментальним і, якщо модель та інформація коректні, то нові ймовірності будуть обчислені також коректно (стосовно аксіом класичної теорії ймовірностей) [51].

Інформацію може отримувати кожний вузол (змінна) мережі, оскільки оновлення ймовірностей інваріантне стосовно напрямку розповсюдження інформації ребрами мережі. Таким чином, БМ значно розширює можливості аналізу даних та прийняття рішень, оскільки дозволяє робити прямий і зворотний логічний висновок. Крім того, одночасне введення інформації про стани декількох вузлів не змінює алгоритму обробки даних мережі, що дає можливість виключити ситуації логічної суперечності, які часто мають місце при застосуванні інших методів.

Побудова БМ вимагає докладних знань стосовно причинно-наслідкових зв'язків між подіями предметної області. Якщо система має надавати можливість використання завчасно отриманих експертних знань, а тим більше таких, що мають інтуїтивну складову, то моделювання краще виконувати за допомогою БМ. Крім того, БМ може навчатись, тобто адаптивно корегувати ймовірності подій при отриманні нової інформації стосовно станів її вузлів. БМ в експертних системах мають такі значні переваги, як можливість чисельного трактування алгоритмів формування логічного висновку, гнучкість процесу розповсюдження інформації і врахування рівня суб'єктивізму експертів. Із сказаного випливає, що експертну систему для аналізу стану виробничого процесу доцільно будувати саме на основі БМ, які представляють собою потужний інструмент для моделювання процесів довільної природи та прийняття рішень.

БМ використовують для моделювання предметних областей, які характеризуються невизначеністю. Невизначеність може бути спричинена

недостатнім розумінням предметної області, неповним знанням її стану у момент прийняття рішення (неповні дані), випадковим характером механізмів, що визначають поведінку цієї області, негативним впливом випадкових збурень або ж комбінацією вказаних факторів.

Логічний висновок в БМ означає обчислення умовних ймовірностей станів вибраних змінних на основі інформації про інші змінні. Ключовим поняттям обчислення ймовірностей в БМ є процес оновлення ймовірностей, або зміна міри довіри. Алгоритм цього процесу визначає спосіб отримання апостеріорних ймовірностей вершин мережі на основі отриманої інформації. Таким чином, оновлення міри довіри для вершин може розглядатися як синонім формування ймовірнісного висновку у БМ. По відношенню до вершини БМ довіра – це розподіл ймовірностей.

Нехай БМ – мережа на множинні змінних U , і нехай e – множина тверджень вигляду «змінна A знаходиться у стані a ». Таким чином, e є твердженням «спільна конфігурація вершин A, \dots, B задана як (a, \dots, b) ». Необхідно знайти апостеріорний розподіл ймовірностей $P(X|e)$ для усіх змінних $X \in U$.

Математично ця задача може бути розв'язана таким чином:
використати ланцюгове правило для обчислення $P(U)$;
відокремити $P(U, e)$ – частину $P(U)$, відповідну конфігурації (a, \dots, b) ;
отримати $P(X, e)$ шляхом маржиналізації $P(U, e)$ для кожного $X \in U$ (тобто для кожного стану $x \in X$ підсумувати усі елементи $P(U, e)$, для яких X знаходиться у стані x);

обчислити $P(X|e)$, як результат нормування $P(X, e)$, тобто слід розділити $P(X, e)$ на суму всіх його членів.

Однак, зазвичай $P(U)$ настільки об'ємна, що необхідні обчислення можуть виявитися неприйнятно великими. Відмова від використання

повної ймовірності $P(U)$ може бути здійснена при переході на послідовне застосування теореми Байєса.

Дійсно, поставлена задача знаходження ймовірностей $P(X|e)$ на основі множини свідочств e відносно БМ може бути представлена як задача оновлення ймовірностей на (під)мережі, до складу якої входить лише певна підмножина вузлів графу. Зменшення мережі до цієї підмножини відбувається шляхом послідовного виключення вузлів (маржиналізації) і механізму інверсії ребер графа на основі теореми Байєса. Приклад цього процесу продемонстровано на рис. 10.10, де показано, що для отримання розподілу $P(C)$ вершини C , виконується послідовне звуження графа за допомогою виключення вершин (кроки 1,2,4,5) та інверсії ребер (крок 3).

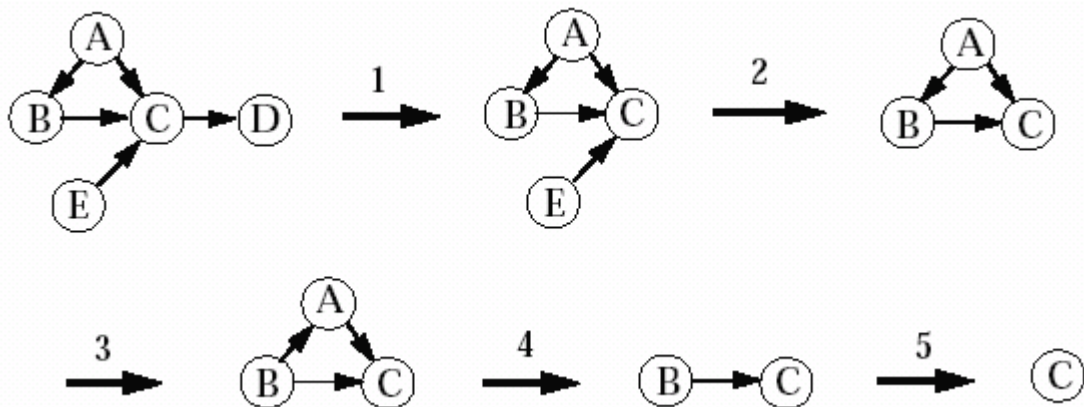


Рис. 10.10 Приклад формування ймовірнісного висновку

Сутність розповсюдження ймовірностей по мережі полягає у використанні поняття повідомлення, згідно з яким оновлення ймовірностей вершин мережі здійснюється шляхом розсилання кожною вершиною мережі двох типів повідомлень про свій стан: (1) повідомлення

батьківським вершинам – π -повідомлення; (2) повідомлення дочірнім вершинам – λ -повідомлення.

Міра довіри для події $X = x$ за цим алгоритмом розраховується як нормований добуток числового еквіваленту повідомлень $\lambda(x)$ і $\pi(x)$. Однак оглянемо основні концепції архітектури обміну повідомленнями для обчислення апостеріорних ймовірностей. Звернемося знову до БМ, наведеної на рис. 10.10. Якщо отримано новий розподіл $P^*(A)$ для вершини A , то фундаментальне правило числення ймовірностей і маржиналізація можуть бути використані для обчислення нового розподілу B за формулою:

$$P^*(B) = \sum_A P(B | A) P^*(A)$$

Таким чином, розповсюдження в напрямі зв'язків досить просте. Можна розглядати це розповсюдження як повідомлення, надіслане від A до B . Це повідомлення є розподіл A , і на цій основі у вузлі B відбувається оновлення розподілу B .

Оновлення довіри відбувається і в протилежному напрямі: інформація щодо B може бути використана для зміни рівня довіри для A . У БМ інструментом розповсюдження ймовірностей у зворотному напрямі є теорема Байєса. Цю ситуацію можна розглядати як розповсюдження розподілу від B до A у вигляді повідомлення. У вузлі A це повідомлення також використовується для оновлення міри довіри.

Розглянемо БМ, граф якої є деревом. Повідомлення можуть надсилатися в обох напрямках, тобто вершина дерева X може надіслати повідомлення до будь-якої сусідньої вершини Y . Повідомлення, що

надсилається, є поточним розподілом X , і вершина Y використовує це повідомлення для оновлення власного розподілу.

Нехай тепер вершини надсилають повідомлення не упорядковано, після чого на деякий час переходять в режим очікування. Можна довести, що у такому разі існує стан рівноваги (стійкий стан), в якому жодне подальше повідомлення не буде змінювати ніякого розподілу. Більше того, цього стійкого стану можна досягти після скінченої кількості пересилань повідомлень. У цьому стані кожна вершина буде зберігати коректний розподіл ймовірностей. На рис. 10.11 наведено приклад обміну повідомленнями для п'яти моментів часу, необхідних для досягнення стійкого стану дерева після введення даних про стани двох вершин.

Отже, в алгоритмі, що ґрунтується на повідомленнях, вплив кожного нового свідчення розглядається як збурення, що розповсюджується по мережі шляхом обміну повідомленнями між сусідніми вершинами. Однак досягнення стійкого стану при застосуванні цього алгоритму, так само як і гарантія коректності результатів, вимагають існування деревоподібної структури мережі. БМ має бути полідеревом, тобто спрямованим ациклічним графом, в якому між будь-якими двома вершинами існує лише один маршрут без урахування напрямку ребер. Це суттєве обмеження, оскільки переважну більшість реальних предметних областей неможливо змодельовати на графі без (неорієнтованих) циклів.

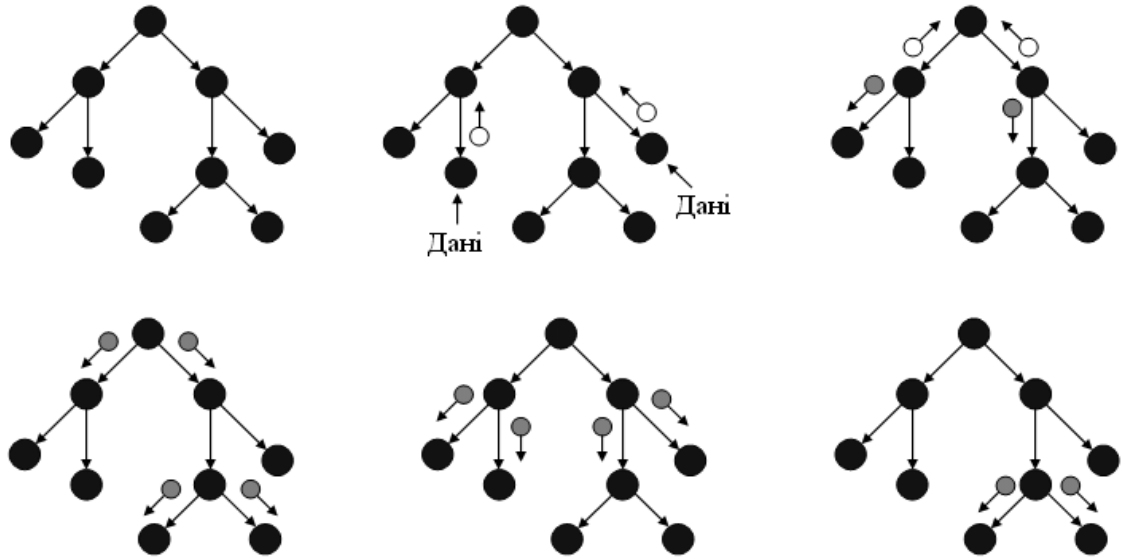


Рис. 10.11 Розповсюдження інформації через повідомлення

Для того щоб обійти це обмеження, запропоновано кілька методів. Наприклад, метод обумовлення (conditioning) ґрунтується на тому, що будь-яка мережа з циклом може бути зведена до полідерава перебором станів кореневої вершини циклу (послідовним обумовленням цієї вершини). Коли вершина обумовлена, тобто знаходиться у певному стані, можна вважати, що вона більше не є частиною мережі, а замість неї розглянути стільки мереж, скільки станів ця вершина має. Після цього теорема Байєса дозволить обчислити умовні ймовірності станів цієї вершини за умови, що відомі розподіли ймовірностей її сусідських (дочірніх) вершин. Таким чином, за даним методом доведеться обчислити стільки мереж, скільки станів має коренева вершина кожного циклу, а тому цей метод дуже уповільнює обчислення.

Інша ідея використовується в методі кластеризації [5, 58]. Метод кластеризації схожий на метод розповсюдження в деревах, описаний вище. Однак, в ролі вершин дерева у ньому виступають не окремі вершини, а їх

групи (кластери). За допомогою методів теорії графів аналізуються властивості незалежності вершин мережі і формується множина кластерів, які поєднуються у дерево. Отримане дерево має властивість сполученого дерева: для кожної пари (V, W) вершин дерева усі вершини маршруту між V та W містять їх перетин $V \cap W$. Сполучене дерево, отримане в результаті кластеризації вершин вихідного дерева, є полідеревом, і в ньому можна застосувати обмін повідомленнями. Після отримання апостеріорних ймовірностей для кластерів обчислюють ймовірності вершин вихідного дерева. Полідерево формується на основі графа, отриманого внаслідок триангуляції. Уяву про процес формування сполученого дерева надає рис. 10.12.

Структура мережі найчастіше визначається експертами предметної області, хоча існують методи структурного навчання БМ на основі даних (наприклад, розглянутий вище алгоритм). Таблиці ймовірностей, навпаки, часто генеруються на основі даних за допомогою статистичних методів. Проте необхідно підкреслити, що суб'єктивний байєсівський підхід не вимагає „об'єктивності” ймовірностей, а тому дозволяє при формуванні таблиць умовних ймовірностей спиратися на суб'єктивні оцінки експертів. Можна зазначити, що результати логічного висновку більше чутливі до якісної структури БМ, ніж до кількісних значень ймовірностей [5].

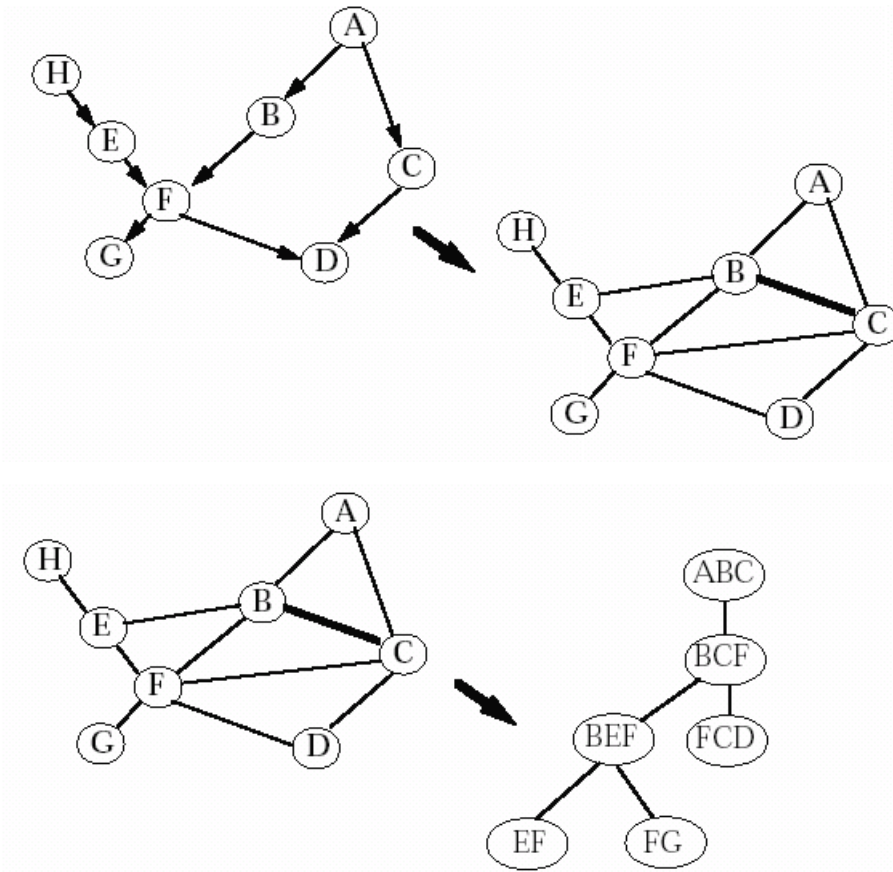


Рис. 10.12 Метод кластеризації для отримання сполученого дерева

10.6 Розробка експертної СППР для підприємства на основі БМ

10.6.1 Виділення класу задач експертної системи

Для суб'єктів бізнесу існує клас задач, які розв'язуються за допомогою експертів. Створюючи експертну систему для підприємства, необхідно виділити такий клас задач цієї множини, який був би практично важливим для підприємства та водночас якісно вирішувався за допомогою цього інструментарію. До таких задач можна віднести завдання стратегічного аналізу стану бізнесу, тобто порівняльного аналізу привабливості тих чи інших рішень та бізнесу в цілому.

Мета стратегічного аналізу – виявити риси внутрішнього й зовнішнього середовища підприємства, які впливають на його стратегічне бачення й можливості. Оцінювання бізнесу у стратегічному менеджменті спирається на результати конкурентного і ситуаційного аналізу.

Конкурентний аналіз зазвичай використовується для аналізу зовнішньої ситуації підприємства; ситуаційний аналіз стосується поточної ситуації на підприємстві та його найближчого оточення. На рис. 10.13 представлена структура стратегічного аналізу для підприємства. Логічним виходом цього аналізу є оцінка привабливості стратегічних альтернатив (чи бізнесу в цілому).



Рис. 10.13 Оцінка стану бізнесу в рамках стратегічного аналізу

10.6.2 Побудова БМ у експертній системі

Експертна система має реалізувати процес стратегічного аналізу у режимі консультації з користувачем, надавши йому на основі акумульованого експертного досвіду свою оцінку привабливості стратегічних альтернатив. Реалізація цієї системи на основі байєсівських мереж довіри означає, що необхідно побудувати мережу, виділивши у ній вхідні вузли (тобто такі, ймовірності станів котрих визначає користувач) і результуючі вузли. Далі, надавши станам вхідних вузлів певні значення, необхідно отримати ймовірності станів результуючих вузлів.

Для задачі оцінювання стану бізнесу з позицій стратегічного аналізу природним видається виділити одну основну результуючу вершину мережі, що відображає привабливість стратегії чи бізнесу в цілому. Назвемо її Рейтинг (Business rating) і покладемо, що вона може знаходитися у станах Привабливий та Непривабливий.

До вхідних вузлів мережі логічно віднести такі економічні характеристики бізнесу, як очікуваний попит на продукцію підприємства, темпи зростання ринку, диференціація постачальників, спеціалізація конкурентів підприємства, наявність товарів-замінників, лояльність покупців, обсяги капіталовкладень, наявність довгострокових контрактів.

Ключовими для оцінки привабливості бізнесу є також такі поняття, як прибутковість підприємства, рівень конкуренції в галузі, наявність чи відсутність надлишкового продукту, можливість подолання економічних бар'єрів входу/виходу (стратегічна гнучкість) та стабільність позиції підприємства. Економічні характеристики також доцільно виділити в окремі вершини БМ. Сукупність частини вузлів БМ визначена; їх можливі стани і стратегічне значення наведені у табл. 10.7.

Таблиця 10.7 Частина вершин БМ для оцінювання бізнесу

Вершина	Можливі стани	Стратегічне значення
Рейтинг	<i>Привабливий</i> <i>Непривабливий</i>	Результуюча вершина; визначає привабливість альтернативи чи підприємства
Затрати для входу	<i>Значні затрати</i> <i>Незначні затрати</i>	Великі затрати збільшують бар'єри входу/виходу, зменшуючи гнучкість підприємства
Спеціалізація конкурентів	<i>Інша продукція</i> <i>Така сама продукція</i>	Конкуренти з такою ж продукцією створюють більшу загрозу цінової війни
Лояльність клієнтів	<i>Звичайна</i> <i>Висока</i>	Лояльні клієнти надають можливість безпечніше варіювати стратегії
Надлишок продукції	<i>Низький</i> <i>Високий</i>	Надлишок продукції знижує рівень цін і, як наслідок, загострює конкуренцію
Очікуваний попит	<i>Очікується зменшення</i> <i>Очікується збільшення</i>	Збільшення попиту веде до зменшення надлишків і потенційного зросту прибутків

Стратегічне значення розглянутих економічних показників дозволяє визначити причинно-наслідкові зв'язки між вузлами мережі, визначивши тим самим ребра мережі та їх спрямованість. Заповнення таблиць умовних ймовірностей виконується на основі експертної інформації.

Використання статистичних даних в ЕС. Розглянемо такі вхідні вершини побудованої мережі, як Зростання обороту та Очікуваний попит. Ці вершини, на відміну від інших, є кількісними, а не якісними, та ще й стосуються очікувань у майбутньому періоді часу, а тому навряд чи користувач експертної системи може точно визначити їх стан. Зрозуміло, що точне визначення ймовірностей станів цих вузлів на перший погляд видається складним завданням; однак, ці вершини є вхідними для ЕС і тому їх стани мають бути визначеними.

Значення станів інших вхідних вузлів можуть бути визначені в результаті простого вибору користувачем одного із станів вершини (при цьому обраний стан отримує ймовірність 1, а усі інші – 0). Для двох розглянутих вершин така методика визначення ймовірностей скоріше за все принципово неможлива. Навряд чи можна сказати, що очікується

зростання попиту з ймовірністю 1. Тобто виникає питання коректного визначення відповідних ймовірностей.

База даних підприємства, зазвичай, містить достатньо інформації для автоматизації цього процесу. Як частина загальної інформаційної системи підприємства, БД містить великий об'єм статистичної інформації, у тому числі і про загальний оборот підприємства, і про попит на його продукцію. Якщо виконати відповідні запити до бази, то отримаємо ряди числових даних, які можна обробляти статистичними методами. Таким чином, за наявності відповідних рядів даних, можна спрогнозувати значення показників обороту і очікуваного попиту на майбутній період, а маючи ці прогнозні значення, можна визначити стани вершин БМ.

10.6.3 Модель виробничої діяльності підприємства

Експертна система, в основі якої лежить модель предметної області у вигляді БМ, може бути використана для розв'язку задачі вибору оптимальної стратегії розвитку підприємства. Цілями моделювання при розробці стратегії розвитку підприємства є: оцінка поточної управлінської стратегії; визначення можливих шляхів і засобів керування розвитком підприємства; проведення порівняльного аналізу наслідків впровадження тих чи інших управлінських рішень; обрання найкращої стратегії розвитку.

Для досягнення поставлених цілей необхідно визначити критерії, які визначають ступінь ефективності стратегії, що впроваджується. Крім того, слід визначити фактори, що впливають на кінцевий результат, та їх взаємозв'язок. Основними критеріями, що визначають рівень розвитку, конкурентоспроможності та перспективності підприємства, можна назвати: цінність кадрів підприємства; рівень розвитку підприємства та фінансовий результат діяльності підприємства.

Цінність кадрів підприємства визначає рівень вмотивованості та кваліфікації працюючих. Цей критерій дозволяє оцінити якість трудових ресурсів підприємства. Рівень розвитку підприємства є показником технологічної структурованості підприємства. Цей критерій визначає ефективність внутрішньої організації підприємства та ступінь довершеності бізнес-процесів, що протікають на підприємстві. Фінансовий результат – це один з найважливіших інтегральних показників бізнес-діяльності підприємства. Він дозволяє оцінити стан підприємства на основі кількісних фінансових показників діяльності, таких, як доходи і витрати.

За фактори, що впливають на значення критеріїв діяльності підприємства, можна вибрати такі укрупнені економічні показники: обсяг виробництва; виробничі фонди; невиробничі витрати; собівартість виробництва. Іншими впливовими факторами є також показники, що відносяться до трудових ресурсів підприємства – чисельність працюючих, продуктивність праці і середня заробітна плата працівників. Ці фактори прямо чи опосередковано впливають на всі критерії розвитку підприємства. При моделюванні враховано такі фактори: продуктивність праці на підприємстві, середня заробітна платня працюючих, чисельність персоналу, виробничі фонди підприємства, невиробничі витрати, обсяг виробництва та собівартість виробництва.

Для побудови відповідної байєсової мережі і подальшої роботи з нею скористаємося графічною оболонкою системи Hugin Lite 5.1. В термінології БМ вищенаведені показники та інтегральні критерії складатимуть вершини мережі, які ми позначимо наступним чином: продуктивність праці – ПП; середня заробітна платня працюючих – СЗП; чисельність працюючого персоналу – ЧП; виробничі фонди підприємства - ВФ; невиробничі витрати – НВ; обсяг виробництва – ОВ; собівартість виробництва – С; цінність кадрів – ЦК; рівень розвитку

підприємства – РРП; фінансовий результат – ФР. Врахувавши причинно-наслідкові зв'язки між наведеними факторами, можемо побудувати БМ, представлену на рис.10.14.

Надалі для кожної вершини необхідно визначити множину можливих станів. Для спрощення прийmemo, що кожна з вершин може приймати лише два стани – Збільшується (позначення Inc) чи Зменшується (позначення Dec).

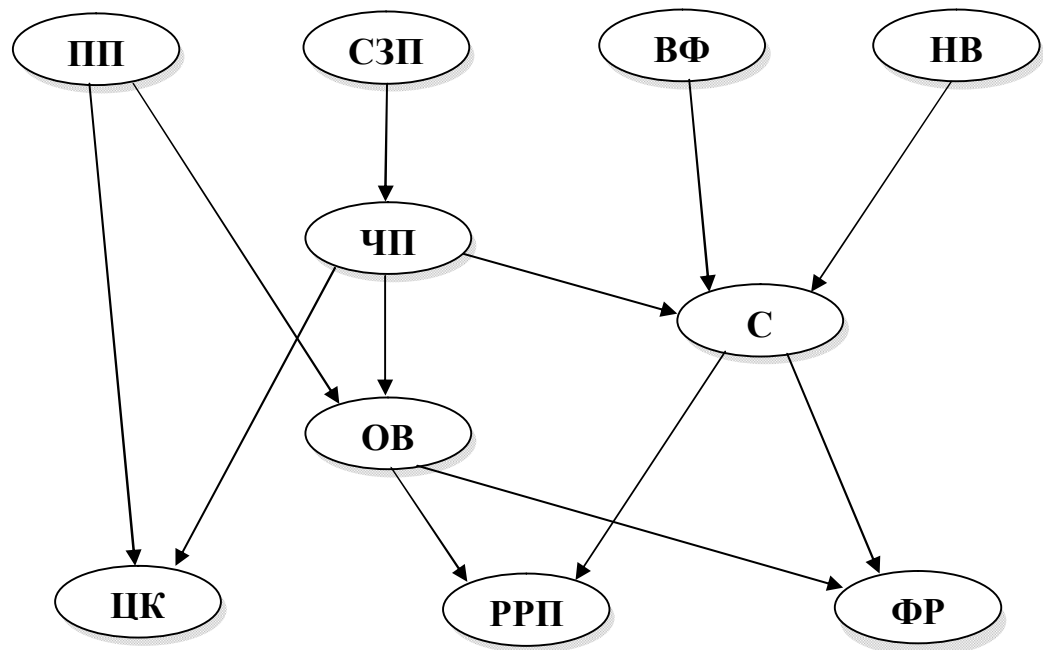


Рис. 10.14 БМ для моделювання стратегій розвитку підприємства

Тоді результатом моделювання будуть ймовірності $P(Inc)$ та $P(Dec)$ для вершин ЦК, РРП та ФР. Наприклад, якщо в результаті моделювання деякої стратегії отримаємо, що для вершини ЦК $P(Inc)=0,7$, то це означає, що при застосуванні цієї стратегії цінність кадрів збільшиться з ймовірністю 0,7.

Останнім кроком при побудові БМ є визначення таблиць умовних ймовірностей для кожної вершини мережі. Наведемо таблицю умовних

ймовірностей для вершини мережі, що відповідає обсягу виробництва. Відповідно до топології мережі, стан для цієї вершини (ОВ) визначається станом вершин ЧП і ПП, тобто на обсяг виробництва впливають чисельність працівників та продуктивність їх праці (рис. 10.15).

ЧП	Inc		Dec	
ПП	Inc	Dec	Inc	Dec
Inc	0.9	0.8	0.5	0.05
Dec	0.1	0.2	0.5	0.95

Рис. 10.15 Таблиця умовних ймовірностей для вершини ОВ

З наведеної таблиці видно, що при зростанні чисельності працюючих і одночасному зростанні продуктивності праці обсяг виробництва буде зростати, і ймовірність цього зростання, за думкою експертів, складе 0,9. Також можна бачити, що у разі одночасного спадання кількості працюючих і зростанні продуктивності їх праці експертна оцінка ускладнена, і ймовірності зростання та спадання цього показника покладені рівними.

10.6.4 Моделювання управлінських рішень

Першим етапом роботи є аналіз поточної стратегії економічного розвитку підприємства. Апріорні ймовірності задаються для корневих вершин дерева ПП, СЗП, ВФ і НВ. Кореневі вершини дерева трактуються як об'єкти керування. Листові вершини дерева трактуються як результати моделювання. Отримані в ймовірності станів вершин представляють собою результати моделювання стратегії розвитку підприємства. Покладемо для корневих вершин мережі ПП, СЗП, ВФ і НВ значення

ймовірностей $P(Inc)=P(Dec)=0,5$. Ініціалізація мережі, тобто розповсюдження по мережі цих ймовірностей, дає такі значення критеріїв (табл. 10.8).

Таблиця 10.8 Поточний стан підприємства

Вершина	ЦК	РРП	ФР
$P(Inc)$	0,5644	0,5345	0,5223

Таким чином, експертна оцінка поточної стратегії дає початкові значення критеріїв розвитку, з якими можна порівнювати результати моделювання інших стратегій. Наступним етапом аналізу є моделювання впровадження у виробництво інших стратегічних рішень. У відповідності із розробленою моделлю виділимо три можливих стратегії.

Стратегія 1: інтенсифікація виробництва. Ця стратегія припускає досягнення бажаних критеріїв розвитку підприємства шляхом підвищення продуктивності праці персоналу. Комплекс заходів щодо інтенсифікації передбачає впровадження прогресивних технологій виробництва, збільшення норми виробітки на одного працюючого, зростання частки наукомістких високотехнологічних операцій у виробничому циклі тощо.

В рамках побудованої моделі впровадження цієї стратегії означає варіювання (підвищення) ймовірності $P(Inc)$ для вершини ПП. Серед інших розглянутих стратегій – стимулювання персоналу і зменшення втрат. Отже, для моделювання стратегії розвитку підприємства по черзі встановимо $P(Inc)=1$ для вершини ПП, $P(Inc)=1$ для вершини СЗП і $P(Dec)=1$ для вершини НВ, залишаючи ймовірності інших вершин незмінними (табл. 10.9).

Таким чином, отримані ймовірності збільшення критеріїв розвитку підприємства у разі застосування тієї чи іншої стратегії. Наступним і

останнім етапом моделювання є визначення найкращої стратегії, яку слід рекомендувати управлінському персоналу підприємства.

Таблиця 10.9 Результати моделювання на мережі

Стратегія	Значення P(Inc) для критерію		
	ЦК	РРП	ФР
№1	0,5712	0,5801	0,5793
№2	0,5700	0,5570	0,5628
№3	0,5644	0,6335	0,5975

Вибір оптимальної стратегії розвитку підприємства. Для прийняття управлінського рішення необхідно визначити інтегральний критерій розвитку підприємства для кожної стратегії, краще значення якого і визначить оптимальну стратегію з числа розглянутих. Розрахунок інтегрального критерію пропонується виконати на основі наступних міркувань. Нехай при моделюванні розглядаються K критеріїв і S стратегій розвитку підприємства. В результаті моделювання кожної стратегії на байєсівській мережі отримані значення ймовірностей збільшення критеріїв $p_{ks}, k=1, \dots, K, s=1, \dots, S$ і відомі початкові значення p_{k0} для поточного стану підприємства. Тоді для кожної стратегії розраховується зважений показник збільшення критеріїв розвитку підприємства:

$$p_s = \sum_{k=1}^K w_k p_{ks}, \quad s = 0..S,$$

де w_k – ваговий коефіцієнт, що визначає важливість збільшення k -го

критерію розвитку для підприємства, $\sum_{k=1}^K w_k = 1$.

Зважений показник p_s можна розглядати як ймовірність збільшення деякого інтегрального критерію розвитку підприємства, тобто p_s – це величина, яка характеризує ймовірність загального поліпшення рівня розвитку, конкурентоспроможності та перспективності підприємства при впровадженні s -ої стратегії.

Інтегральний критерій розвитку підприємства може бути знайдений як відношення зваженого показника $p_s, s = 1, \dots, S$ до показника поточного стану p_0 : $I_s = \frac{p_s}{p_0}, s = 1, \dots, S$, і тоді за оптимальну стратегію s^{opt} слід приймати таку, для якої інтегральний критерій розвитку має найбільше значення:

$$s^{opt} = \arg \max_{s=1..S} I_s .$$

У нашій моделі $K=3, S=3$, величини $p_{ks}, k = 1, \dots, K, s = 1, \dots, S$ наведені в табл. 10.9, величини p_{k0} – в табл. 10.10.

Будемо вважати, що на цьому етапі розвитку підприємства цінність кадрів, рівень розвитку виробництва і фінансовий результат однаково важливі для управлінського персоналу; тоді

$$w_1 = w_2 = w_3 = \frac{1}{3} .$$

Тепер, відповідно до наведених вище формул, отримуємо:

$$p_1 = 0.5769, p_2 = 0.5633, p_3 = 0.5985, p_0 = 0.5404;$$

$$I_1 = \frac{0.5769}{0.5404} = 1.07, I_2 = \frac{0.5633}{0.5404} = 1.04, I_3 = \frac{0.5985}{0.5404} = 1.11;$$

$$s^{opt} = \arg \max_{s=1..S} I_s = s_3.$$

Отже оптимальною стратегією розвитку підприємства є стратегія № 3 зі зменшенням витрат. При запровадженні цієї стратегії цінність кадрів на підприємстві збільшиться з ймовірністю 0,56, рівень розвитку виробництва з ймовірністю 0,63 і фінансовий результат з ймовірністю 0,6.

Таким чином, БМ, як потужний високоресурсний ймовірнісний інструмент моделювання процесів різної природи та прийняття рішень, дає можливість враховувати структурні і статистичні невизначеності досліджуваних процесів.

Запропонована послідовність побудови байєсівських мереж довіри може бути використана при моделюванні соціально-економічних процесів на будь-якому рівні функціонування держави, а також для опису динаміки і статички технічних систем. Основні проблеми, пов'язані з побудовою таких мереж, полягають у визначенні апіорних ймовірностей подій та оптимізації топології мережі. Процес розповсюдження ймовірностей по мережі реалізується шляхом послідовного застосування теореми Байєса або за допомогою алгоритму генерування і аналізу двох типів повідомлень стосовно стану батьківських і дочірніх вершин, які уточнюють апостеріорні розподіли ймовірностей подій (вершин). Однак, досягнення стійкого стану мережі при застосуванні алгоритму, що ґрунтується на повідомленнях, вимагає щоб мережа мала деревовидну структуру.

Цього обмеження можна уникнути за допомогою методів обумовлення та кластеризації. Останній приводить до побудови полідерава, для вершин якого можна обчислити апостеріорні ймовірності.

Побудована мережа Байєса для розв'язання задачі оцінювання привабливості стратегій розвитку малого бізнесу. Таблиці умовних апіорних ймовірностей якісних вершин заповнені за допомогою експертної інформації, а кількісних вершин – за допомогою статистичних даних (зростання обороту і очікуваний попит). Тобто значення умовних ймовірностей для кількісних вершин отримано на основі відповідних рядів статистичних даних і частотних ймовірностей. В цілому БМ для оцінювання стратегій розвитку підприємства складається з 10 вершин, які забезпечили моделювання кількох стратегій розвитку, зокрема, інтенсифікацію та зменшення витрат на виробництво. Для розглянутого прикладу кращою виявилась стратегія зменшення витрат.

Запитання та завдання для самоконтролю

1. Сфери застосування, переваги і передумови використання байєсових мереж у прийнятті рішень.
2. Як описують байєсову мережу?
3. Як формулюється проста теорема Байєса? Показати форми запису?
4. Як визначають апіорну та правдоподібну інформацію?
5. Описати загальну схему застосування мереж Байєса у розпізнаванні кіл.
6. Як визначають правило прийняття рішення щодо наявності шуканого зображення двох кіл в деякому образі?

7. Пояснити теорему Байєса для даних, які надходять з кількох джерел.
8. Як описують графічно просту мережу Байєса?
9. Як обробляють мережі Байєса у матричному вигляді?
10. Дати характеристику подання Байєсова дерева прийняття рішень.
11. Якими є оцінки складності побудови мереж Байєса?
12. У чому полягає спрощений евристичний метод Чау і Ліу?
13. Як формують опис мережу Байєса мінімальної довжини?
14. Що є джерелами даних у задачі навчання мережі?
15. У чому полягає простий алгоритм навчання мережі Байєса з використанням опису мінімальної довжини?
16. Пояснити евристичний алгоритм побудови мережі Байєса.
17. Як оцінюють якість мережі Байєса?
18. Описати технологію розробки інформаційної системи на основі БМ.
19. Як описують мережі довіри Байєса?
20. Як формують ймовірнісний висновок?
21. Описати розповсюдження інформації через повідомлення.
22. У чому полягає метод кластеризації для отримання сполученого дерева?
23. Як виділяють клас задач експертної системи?
24. Описати порядок побудови БМ у експертній системі.
25. Як визначають модель виробничої діяльності підприємства?
26. У чому полягає моделювання управлінських рішень?
27. Дайте визначення поняттю «мережа Байєса».
28. Поясніть процедуру формування висновку на основі теореми Байєса.

29. Опишіть наївний класифікатор Байєса.
30. Наведіть недоліки звичайних процедур обчислення ймовірностей та формування висновку.
31. Розкрийте принцип формування опису мережі Байєса мінімальної довжини.
32. Опишіть алгоритм методу опису мінімальної довжини.
33. Сформулюйте евристичний алгоритм побудови мережі Байєса.
34. Наведіть методи оцінювання якості навчання мережі Байєса.
35. Які недоліки і переваги мають мережі Байєса?
36. Опишіть технологію розробки експертних систем.
37. Сформулюйте структуру мережі Байєса для моделювання стратегій розвитку підприємства.

РОЗДІЛ 11 ПОБУДОВА СППР НА ОСНОВІ МЕТОДІВ ЕКСПЕРТНОГО ОЦІНЮВАННЯ

Прийняття рішень являє собою вибір найбільш преференційного варіанту рішення із множини допустимих альтернатив або упорядкування множини рішень. Задача вибору варіантів, та задача розподілу ресурсів (більш складна і охоплююча задача) є споконвічними проблемами, з якими стикається людство у всіх сферах свого буття. Ресурсами можуть виступати технічні засоби, людські резерви, ділянки частотного, часового та просторового діапазонів, кількість каналів зв'язку, інформаційні сигнали, фінансові активи і кошти, нематеріальні активи, енергоносії тощо.

11.1 Методи експертних оцінок для розв'язання задач прийняття рішень розподілу ресурсів

11.1.1 Змістовне формулювання задачі

Існує деякий фіксований граничний обсяг ресурсів, призначений для розподілу. Є множина проектів, які розглядаються як кандидати на використання цих ресурсів. Загальний обсяг необхідних ресурсів для всіх проектів може перевищувати наявний граничний обсяг. Необхідно вибрати із множини усіх проектів-кандидатів деякий набір проектів, задоволення яких ресурсами забезпечить найбільш раціональне використання наявного обсягу ресурсів.

Проблема розподілу ресурсів по суті являє собою вибір у певному сенсі „кращих” з наявних варіантів кандидатур на одержання ресурсів, що звичайно здійснюється за деяким критерієм оптимальності чи набором таких критеріїв за допомогою певної процедури пошуку екстремуму. Проблема розподілу ресурсів, крім власне вибору варіантів, передбачає

розв'язання оптимізаційних задач щодо розподілу ресурсів при задоволенні тих чи інших обмежень задачі.

Розв'язання проблеми вибору варіантів, а саме багатокритеріального вибору варіантів, є головним і необхідним стрижнем розв'язання загальної проблеми розподілу ресурсів. Проблема вибору варіантів фактично постає частиною проблеми розподілу ресурсів, проте у певних постановках може виступати окремою задачею вибору варіантів з множини допустимих альтернатив.

Таким чином, виявляється доцільним спільне дослідження проблеми розподілу ресурсів і проблеми вибору варіантів в спільній постановці проблеми розподілу ресурсів і вибору варіантів.

В сучасній теорії прийняття рішень можна виділити такі аспекти: теоретичний (доведення теорем існування, збіжності, тощо), прикладний (побудова моделей для розв'язання конкретних практичних задач) і обчислювальний.

Проблема розподілу ресурсів під кутом зору теорії прийняття рішень може бути розглянута насамперед у прикладному і обчислювальному аспектах. Застосування ТПР передбачає дослідження проблеми шляхом аналізу окремих її елементів, детального ізольованого вивчення цих елементів і тільки після цього здійснюють аналіз найпростіших взаємодій між ними.

Проте теоретичні і прикладні проблеми розподілу ресурсів потребують складніших і глибоких сучасних методів та підходів, які не вкладаються в рамки однієї дисципліни; вони є міждисциплінарними і виникають на стику різних наук – ґрунтуються на системному аналізі.

Системний аналіз проявляється у всесторонньому науковому підході до прийняття рішень – системному підході, за яким вся проблема

вивчається в цілому, визначаються цілі і шляхи їх реалізації із урахуванням можливих наслідків.

ТПР і системний аналіз ґрунтуються на застосуванні системного підходу, однак, якщо в ТПР і дослідженні операцій системний підхід спрямовується головним чином на аналіз зв'язків усередині системи, що призначена для вирішення певної окремої задачі, то у системному аналізі він застосовується ще й для виявлення зовнішніх зв'язків даної системи з суміжними системами, які впливають на рішення задачі [22].

Так, проблема розподілу ресурсів і вибору варіантів, задавання цілей і вибір стратегії завжди взаємопов'язані. Ресурси представляють собою свого роду „фільтр”, через який необхідно пропускати рішення. За умов неможливості раціонального розподілу наявних ресурсів, виникає необхідність перегляду цілей і стратегії до тих пір, поки не буде досягнута забезпеченість ресурсами та їх раціональний розподіл [3].

11.1.2 Задачі розподілу ресурсів в умовах невизначеності

Необхідно зазначити, що при моделюванні та формальному описі систем різної природи, у функціонуванні яких приймає участь людина, наприклад економічних системах, стикаються з проблемою домінування якісних, погано визначених факторів, які виявляються у нечітких, неточних, розпливчастих властивостях процесів та явищ [3, 22].

В результаті аналізу задачі розподілу ресурсів встановлено, що для задач такого класу, особливо для задач розподілу інвестиційних ресурсів, властиві умови невизначеності, що зумовлені наявністю як внутрішніх так і зовнішніх чинників, зокрема: невизначеність цілей, структурна, ситуаційна, інформаційна, статистична і комбінаторна невизначеності, що утворюють системну невизначеність. В ТПР об'єктивну дійсність, що

спричиняє невизначеність, у тому випадку коли вона постає незацікавленою інстанцією, поведінка якої невідома і яка не містить елементу свідомої протидії цілям задачі, називають „природою”. Природа виступає у ролі керуючого рухом випадкових подій.

До розв’язання проблеми розподілу ресурсів можна підходити через розв’язання наступних задач, кожна з яких виступає окремою задачею прийняття рішень, дослідження операцій та системного аналізу: визначення цілей, критеріїв оптимальності, критеріїв і методів добору „кандидатів” на отримання ресурсів; формування множини допустимих альтернатив; порівняння та упорядкування множини альтернатив за обраними критеріями; добір кращих варіантів за критерієм оптимальності та вибір рішення.

Розв’язання задач, приналежних до класу проблем, що виникають у системах, у функціонуванні яких приймає участь людина, – класу проблем розподілу ресурсів і вибору варіантів, може бути класифіковане як погано структуроване, динамічне, рішення в умовах невизначеності, багатоосібне та, в залежності від певної ситуації, стратегічне чи тактичне.

Існуючі детерміновані підходи з використанням точних характеристик об’єктів, явищ та процесів, точних методів моделювання та прийняття рішень і процедур оптимізації не враховують зазначені фактори, тому не можуть бути успішно використані при моделюванні реальних процесів. Безперечно, для розв’язання задач зазначеного класу і прийняття рішень слід спиратись на досвід, знання та інтуїцію фахівців-експертів [22].

11.1.3 Методологія евристичного розв'язання задач розподілу ресурсів

Методологія розв'язання задач за участю людини має базуватись не лише на формальних методах, а значною мірою на евристичних міркуваннях, а також на інтуїції. По-перше, це виявляється у необхідності враховувати різноманітні якісні фактори, значення яких крім як експертним чином оцінити неможливо (наприклад, соціальна значимість інвестиційних проектів). По-друге, використання відомих кількісних значень оцінок за деякими характеристиками проектів є недостатнім для такого класу задач, і потребує їх доповнення експертними оцінками.

Наприклад, строк окупності 2 роки для одного проекту може виявитись задовільним, а для іншого дуже тривалим. ОПР змушені робити висновки на основі невеликого числа спостережень, які, зазвичай, не можуть бути відтвореними. Необхідно зазначити, що при підтримці прийняття рішень необхідно не лише вірно описати поведінку системи, але й передбачати вплив людського фактору [41].

Задачу вибору можна розв'язати, якщо у деякий спосіб виконати структурування множини альтернатив. Структурування полягає у класифікації, кластеризації чи ранжируванні альтернатив, які здійснюється за допомогою некрітеріальних та критеріальних методів [1, 44]. Некритеріальні методи ґрунтуються на використанні здатності людського мозку створювати загальне уявлення про об'єкт. За цими методами експерти виконують попарне порівняння альтернатив «в цілому», тобто порівнюють цілісні образи об'єкту [41].

Критеріальне структурування полягає у співставленні альтернатив за деяким набором кількісних та якісних критеріїв. Розв'язання реальних задач із застосуванням некритеріальних методів є практично неприйнятним, оскільки у такому випадку отримують грубі та часто недостатньо обґрунтовані рішення [34, 44, 65].

Людський вимір в СППР, наприклад при розподілі ресурсів і виборі варіантів, мають відігравати значну роль [1], внаслідок чого використання експертних оцінок у СППР має знайти достатньо широке застосування.

Більше того, як уже встановлено, розв'язок задачі здійснюється в умовах неповної, неточної і невизначеної інформації, що насамперед зумовлено постановкою задачі та її середовищем. Серед класу задач розподілу ресурсів найбільш складною постає задача розподілу інвестиційних ресурсів, оскільки вона в більшій мірі споріднена з величезною відповідальністю ОПР за прийняте рішення. Все це спричиняє необхідність застосування лише прозорих методів підтримки рішень у СППР, необхідність надання можливості ОПР безпосередньо впливати на рух процесу прийняття рішення і бути безпосередньо задіяними в процесі підготовки і прийняття рішення, більш того, рішення задачі має враховувати особисті уподобання і переваги ОПР.

Отже, застосування підходів на основі експертних оцінок виявляється не лише бажаним інструментом для одержання якісної інформації в певних класах задач, але і необхідністю для розробки практично прийнятної системної методології для реальних задач прийняття рішень в умовах невизначеностей, особливо, у стратегічних і надзвичайно важливих галузях.

11.1.4 Застосування методу експертних оцінок Дельфі

Метод експертних оцінок відомий у зарубіжній науковій літературі як метод Дельфі [3, 22] (на честь древньогрецького оракула Delphos). Суть методу полягає у отриманні висновку групи експертів про поведінку однієї чи декількох пов'язаних характеристик досліджуваної системи.

Практичне застосування методу експертних оцінок складається з таких етапів [16, 22].

1. Добір групи експертів-фахівців з предметної області проблеми, що розв'язується.
2. Формулювання цілей, які мають бути досягнуті в результаті вирішення проблеми.
3. Розробка спеціальної „форми опитування”, що може включати набір об'єктів, що оцінюються, набір критеріїв оцінювання, шкали оцінок тощо.
4. Опитування експертів – експертне оцінювання об'єктів з проблеми, що розв'язується.
5. Статистична оброблення даних оцінювання для синтезу попередніх результатів.
6. Аналіз отриманих результатів кожного експерта. Врахування експертами оцінок і висновків всієї групи.
7. У випадку корегування своїх оцінок деякими експертами у п.6., виконання повторної обробки даних оцінювання за п.5.
8. Повторне виконання пп. 5–7 до припинення корегування експертами своїх результатів.

9. Отримання консенсусного результату. У випадку неможливості отримання такого результату, наприклад, відсутній стабільний результат при багаторазовому виконанні пп. 5–7 – повернення до п 1, зміна складу групи експертів і повторне виконання пп. 1–9.

10. Аналіз консенсусного результату опитування експертів для його імплементації у розв’язанні проблеми.

11.1.5 Методологія розподілу ресурсів в умовах невизначеності

11.1.5.1 Основні етапи розв’язання задачі

Застосування експертних оцінок за сучасними технологіями сполучене із використанням механізмів опису і оперування якісними неточними нечіткими поняттями.

Задача розподілу ресурсів і вибору варіантів, що розв’язується із залученням експертів формулюється таким чином.

Постановка задачі. Є множина запропонованих проектів $P = \{P_i\}$, $i = \overline{1, n}$, відомі необхідні обсяги ресурсів b_i для кожного проекту P_i , задано загальне ресурсне обмеження B , що є меншим за загальний обсяг необхідних ресурсів: $\sum_{i=1}^n b_i \geq B$. Необхідно у найкращий спосіб розподілити ресурси між проектами P при задоволенні обмеженню B .

Методологія розв’язання задачі розподілу ресурсів між альтернативними проектами в умовах системної невизначеності, ґрунтується на застосуванні методів експертних оцінок. Суть її полягає у одержанні кількісних і якісних експертних оцінок проекту від групи ОПР (експертів), агрегуванні їх у єдину оцінку проекту (ступінь привабливості проекту) з аналізом погодженості експертних оцінок, та у подальшому виборі проектів відповідно до їх привабливості і розподілу ресурсів

між ними. Розв'язання задачі складається з таких послідовних етапів (рис. 11.1.)



Рис. 11.1 Структура методології розв'язання задачі розподілу ресурсів і вибору варіантів

11.1.5.2 Постановка і формалізація задачі

Формується група ОПР – експертів у певній предметній області, що приймають участь в розв’язанні задачі, $D = \{D_t\}$, $t = \overline{1, k}$. Постановка задачі з урахуванням створеної групи експертів трансформується в таку. Необхідно у найкращий, з точки зору вибраного критерію, спосіб розподілити ресурси між проектами P у відповідності до індивідуальних переваг ОПР D при задоволенні обмеження B .

Задавання параметрів. ОПР $D = \{D_t\}$, $t = \overline{1, k}$ для проектів $P = \{P_i\}$, $i = \overline{1, n}$ експертним шляхом визначають набір важливих критеріїв $C = \{C_l\}$, $l = \overline{1, h}$, за якими буде проводитись оцінювання проектів.

Всі критерії C_l будемо вважати максимізуючими, тобто більше значення оцінки за таким критерієм є кращим з точки зору вибору проекту. Якщо деякий критерій виявиться мінімізуючим, тобто кращим з точки зору вибору проекту для інвестування є менше значення оцінки за таким критерієм, то цей критерій змінимо на зворотній.

Всі критерії структуруються в логічну ієрархічну структуру, що складається з послідовно та паралельно з’єднаних блоків, кожен з яких має на вході 5-7 докладних (конкретизуючих) критеріїв, а на виході один критерій, який узагальнює вхідні критерії даного блоку (рис. 11.2).

Наприклад, в задачі розподілу інвестиційних ресурсів між комерційними проектами СНЛВ початкового нижчого рівня може визначати рівень керівника *BossLevel* організації–заявника проекту за входами bl_1 – досвід роботи керівником; bl_2 – освіта; bl_3 – надійність; bl_4 – комунікабельність; bl_5 – успішність. Вихід *BossLevel* цієї СНЛВ буде входом для СНЛВ наступного рівня, яка у свою чергу може визначати рівень *FirmLevel* організації–заявника проекту за такими входами:

$fl_1 = BossLevel$ – рівень керівника організації; fl_2 – активи організації; fl_3 - пасиви організації; fl_4 – дебіторська заборгованість організації; fl_5 - кредиторська заборгованість організації; fl_6 – балансовий прибуток організації. Вихід СНЛВ $FirmLevel$ буде входом СНЛВ наступного рівня, і так далі до СНЛВ найвищого останнього рівня, виходом якої буде ступінь привабливості проекту $P_i, i = \overline{1, n}$, що аналізується.

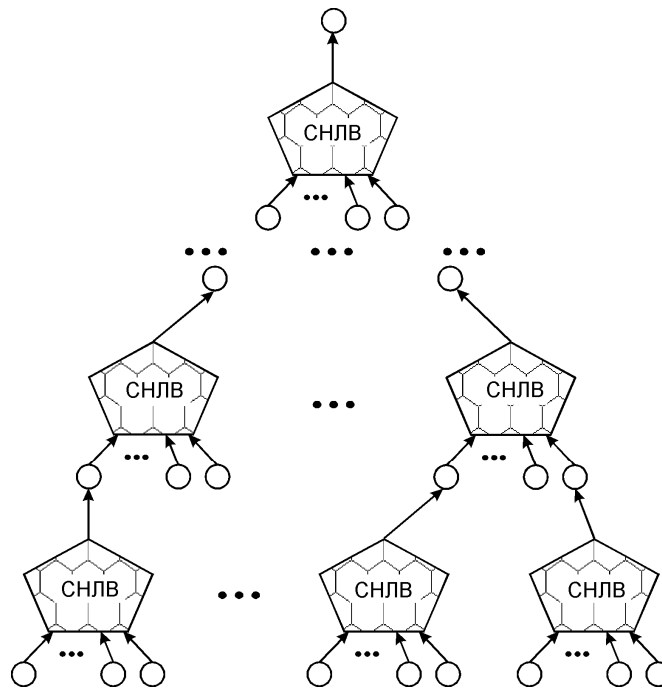


Рис. 11.2 Ієрархічна структура критеріїв оцінювання проектів

Вся ієрархічна структура критеріїв, таким чином, збігається на верхньому рівні ієрархії в єдиний критерій: привабливість проекту для його вибору і надання йому ресурсів. Для розподілу ресурсів потрібні значення оцінок за докладними критеріями, саме вони і складають сформований ОПР набір критеріїв оцінювання $C = \{C_l\}, l = \overline{1, h}$.

11.1.5.3 Оцінювання варіантів

В задачі розподілу ресурсів необхідно враховувати, що визначені критерії C_l , $l = \overline{1, h}$, за якими ОПР будуть оцінювати проекти, можуть відрізнитися своєю важливістю, та в результаті мати різну вагу впливу на рішення задачі. Більше того, кожна ОПР може мати свій погляд на ранжирування та розподіл ваг критеріїв, тому логічно щоб кожна ОПР D_t надавала свої індивідуальні вагові коефіцієнти W_{lt} для кожного критерію C_l , де $l = \overline{1, h}$, $t = \overline{1, k}$.

Окрім цього, доцільним постає врахування значимості, досвіду, рівню підготовки, посади, тощо кожної ОПР. Для цього, групою ОПР $D = \{D_t\}$ колегіально, чи її керівником – особою, що є відповідальною за прийняте рішення, задаються вагові коефіцієнти V_t для ОПР D_t , $t = \overline{1, k}$ [27]. Визначення значень вагових коефіцієнтів критеріїв W_{lt} та ОПР V_t зручно проводити за допомогою геометричного прийому, наведеного далі. Тут вагові коефіцієнти є чіткими, оскільки немає необхідності у їх фазифікації для використання у подальших етапах методу. Зведені до нечіткого вигляду в даному випадку вагові коефіцієнти будуть умовними нечіткими множинами, тобто такими, які визначаються чітким значенням – центром абсолютного чи відносного інтервалу, та стандартною функцією належності. Це призвело б лише до ускладнення практичної реалізації методу при відсутності поліпшення якості рішення.

Для подальшого використання вагові коефіцієнти необхідно піддати процедурі нормування:

$$W_{lt}^{norm} = W_{lt} / \left(\sum_{l=1}^h W_{lt} \right), \quad l = \overline{1, h}, \quad t = \overline{1, k}; \quad (11.1)$$

$$V_t^{norm} = V_t / \left(\sum_{t=1}^k V_t \right), \quad t = \overline{1, k}. \quad (11.2)$$

Множина критеріїв $C = \{C_l\}$ умовно поділяється на дві підмножини. За однією підмножиною критеріїв $C_1 = \{C_{l_1}\}$ кожному проекту P_i надаються суб'єктивні окремі експертні оцінки S_{ilt} кожною ОПР D_l . За іншою підмножиною критеріїв $C_2 = \{C_{l_2}\}$ оцінки проектів визначаються як розраховані числові характеристики проектів, що надаються разом з проектами (наприклад, для комерційних інвестиційних проектів: рентабельність, строк окупності, тощо).

Для зручності реалізації подальших етапів і процедур оцінки кожного проекту за другою підмножиною критеріїв C_2 , тобто вхідні числові характеристики проектів, будемо розглядати спільно з експертними і такими, що є однаковими для всіх ОПР.

Отже, після виконання етапу оцінювання варіантів є визначеними нормовані вагові коефіцієнти критеріїв W_{lt}^{norm} та ОПР V_t^{norm} , кожний проект P_i , $i = \overline{1, n}$ характеризується своїм набором оцінок S_{ilt} , $l = \overline{1, h}$, $t = \overline{1, k}$ за кожним критерієм C_l від кожної ОПР D_l .

11.1.5.4 Узагальнення оцінок

На цьому етапі задача, що розглядається, постає задачею багатокритеріального оптимального вибору проектів, для розв'язання якої скористуємось підходом зведення її до задачі однокритеріального вибору [22, 27], тобто вибір проектів буде відбуватись за узагальненою

агрегованою оцінкою кожного проекту P_i – ступенем привабливості проекту A_i , $i = \overline{1, n}$.

Необхідно зазначити, що деякі існуючі підходи [16, 22] до зваженого агрегування оцінок проектів в задачі багатокритеріального вибору мають суттєвий недолік. За такими підходами спочатку агрегують оцінки S_{ilt} деякого проекту P_i за кожним критерієм C_l , $l = \overline{1, h}$, що надані всіма ОПР D_t , $t = \overline{1, k}$, а потім, використовуючи вагові коефіцієнти критеріїв, отримують остаточні узагальнені оцінки для кожного проекту (рис. 11.3а).

Однак, за такої послідовності, по-перше, ускладнюється врахування значень вагових коефіцієнтів одного критерію, що надані різними експертами, та вагових коефіцієнтів самих експертів; по-друге не враховується логічний когнітивний зв'язок значення оцінки експерта з його суб'єктивним міркуванням щодо ваги такої оцінки за даним критерієм та значеннями оцінок за іншими критеріями; по-третє відсутня методична відповідність підходу аналізу погодженості думок експертів і видалення оцінок некомпетентних експертів.

Для усунення зазначених недоліків і методичних перешкод у зведенні задачі до однокритеріальної, пропонується такий метод знаходження узагальненої оцінки привабливості A_i проекту P_i , $i = \overline{1, n}$ (рис. 11.3б) [27]:

1. Знаходження зважених агрегованих оцінок A_{it}^D кожного проекту P_i від кожної ОПР D_t , де $i = \overline{1, n}$, $t = \overline{1, k}$, відбувається на основі оцінок S_{ilt} ОПР D_t проекту P_i за кожним критерієм C_l та вагових коефіцієнтів W_{lt} критеріїв, де $l = \overline{1, h}$.

2. Знаходження остаточної узагальненої оцінки A_i кожного проекту P_i , де $i = \overline{1, n}$. Розраховується на основі агрегованих оцінок A_{it}^D кожного проекту P_i та вагових коефіцієнтів V_t ОПР, де $i = \overline{1, n}$, $t = \overline{1, k}$.

За умов застосування нелінійних підходів до агрегування оцінок послідовність агрегування суттєво впливає на значення узагальненої оцінки і результат розв'язку задачі.

На жаль, традиційні методи агрегування експертних оцінок та вхідних параметрів проектів не прийнятні в реальних задачах розподілу ресурсів, тому для знаходження узагальнених агрегованих оцінок проектів пропонується застосовувати систему нечіткого логічного висновку.

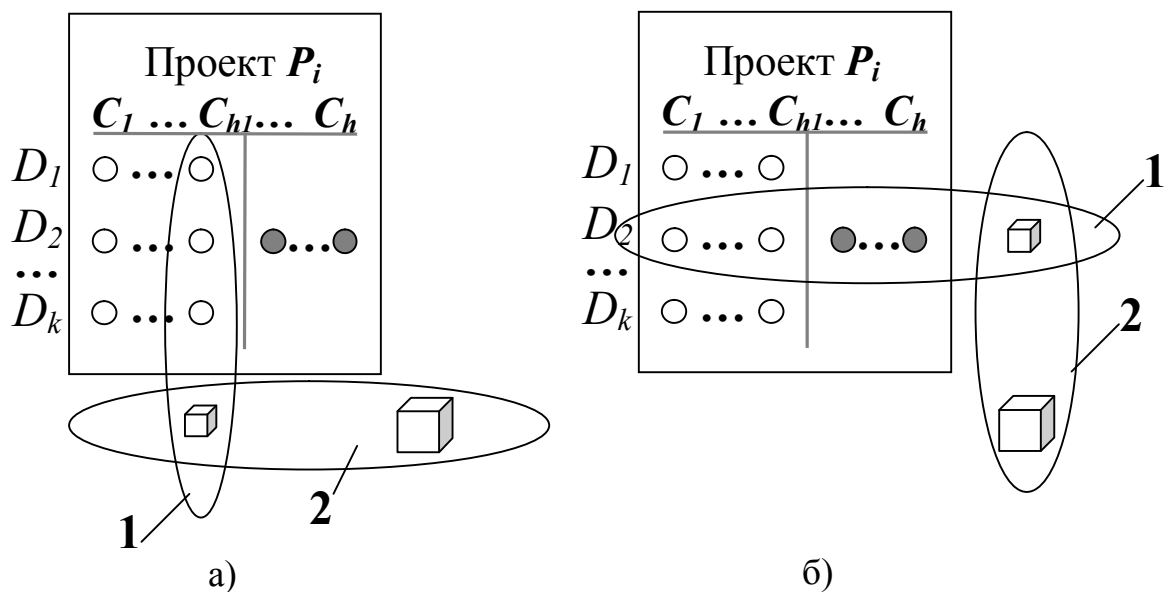
Етап зведення задачі складається з таких трьох операцій:

операції узагальнення оцінок проектів від кожної ОПР за критеріями системою НЛВ;

операції аналізу погодженості оцінок ОПР;

операції агрегування оцінок проектів за ОПР.

Отже, відповідно до ієрархічної структури критеріїв відбувається послідовне багаторівневе узагальнення значення оцінок кожного проекту P_i за всіма критеріями $C = \{C_t\}$ для кожної ОПР D_i з урахування ваг W_t^{norm} критеріїв, в результаті чого для кожного проекту P_i знаходяться $t = \overline{1, k}$ зважених агрегованих оцінок A_{it}^D .



- - S_{il_1} експертна оцінка проекту P_i від ОПР D_i за критерієм C_{i1} .
- - S_{il_2} числова характеристика проекту P_i за критерієм C_{i2} .
- - агреговане значення оцінок.
- - єдина узагальнена оцінка проекту P_i - ступень його привабливості A_i .

Рис. 11.3 Схеми послідовності подвійного агрегування оцінок проектів: а) яка використовується; б) запропонована

З цією метою для кожного структурного блоку ієрархії критеріїв будується відповідна локальна система нечіткого логічного висновку, яка має 5-7 входів і один вихід. СНЛВ нижнього, останнього рівня має своїм виходом: ступінь привабливості проекту на думку ОПР D_i . Необхідно зазначити, що в СНЛВ для експертних оцінок застосовуються побудовані раніше індивідуальні когнітивні ФН, а для розрахованих числових значень відповідні наметоподібні ФН.

За допомогою методу α, β -коаліцій здійснюється аналіз погодженості зважених агрегованих оцінок A_{it}^D , для чого формуються коаліції ОПР зі схожими міркуваннями (оцінками) та з подальшого розгляду видаляються оцінки експертів, що входять до несуттєвих коаліцій.

Тепер всі зважені агреговані оцінки A_{it}^D , що пройшли процедуру аналізу погодженості, для кожного проекту P_i із урахуванням ваг експертів V_i^{norm} за допомогою методу агрегування за міжгруповим консенсусом агрегуються в єдину остаточну узагальнену оцінку A_i – ступінь привабливості проекту.

11.1.5.5 Пошук рішення

Даний етап полягає у розв'язанні однокритеріальної задачі вибору проектів P_i за узагальненими оцінками $A_i, i = \overline{1, n}$ та призводить до добору тих проектів, які максимізують ефективність виділення їм ресурсів, що розподіляються, та задовольняють ресурсному обмеженню B .

За узагальненими оцінками A_i проектів $P_i, i = \overline{1, n}$ та наявними обмеженнями формулюється однокритеріальна задача оптимізації, розв'язання якої приводить до остаточного результату – розв'язання первісної задачі розподілу ресурсів і вибору варіантів.

Зазначена однокритеріальна задача формулюється в залежності від конкретної проблеми і може бути розв'язана за тим чи іншим підходом до розв'язання класичних задач лінійного програмування дослідження операцій. Для задач великої розмірності пропонується застосовувати еволюційні алгоритми. Ефективність даного підходу підтверджена численними прикладами.

Створена методологія розподілу ресурсів і вибору варіантів розроблена на принципах системності і має такі переваги: ґрунтується на спеціально розроблених методах аналізу даних, що виявляють ефективність і результативність, „прозорість” процедур для ОПР, враховують комплекс умов системної невизначеності, передбачають безпосереднє залучення експертів до процесу розв’язання задачі і врахування їх особистих уподобань та переваг, оперують кількісними і якісними даними. Крім цього, розроблена методологія є інваріантною до параметрів і обмежень задачі, масштабною та зручною для практичного використання.

11.2 Підходи до аналізу невизначеності

У випадках системної невизначеності, пов’язаної з нечіткими, неточними, розпливчастими властивостями процесів (наприклад, за наявності інформаційної, ситуаційної і стратегічної невизначеностей, де не завжди можливо коректно застосувати існуючі детерміновані методи), використовують підходи до формального опису невизначених, неточних і ненадійних факторів, зокрема, до формального подання якісних експертних оцінок.

Дослідження виявили, що у мисленні людини використовуються не числа, а образи і слова [41, 61], тому відповіді експертів у процедурах експертного опиту, тобто експертні оцінки, є якісними, що являють собою такі об’єкти нечислової природи, як градації якісних ознак, ранжирування, розбивки, результати парних порівнянь, нечіткі переваги, тощо [41, 61]. У такому разі невизначені якісні поняття подають у вигляді інтервальних чисел або об’єктів багатозначних логік.

Найбільш поширені такі підходи до аналізу невизначеності: апарат інтервальних чисел, нечітка логіка (НЛ), що ґрунтується на теорії нечітких множин (ТНМ) та логіка антонімів. Неперервнозначні чи багатозначні логіки нечітка логіка та логіка антонімів оперують приблизними міркуваннями є розширеннями класичної Булевої логіки до обробки понять часткової істинності між «повністю хибний» до «повністю істинний».

У задачах, в яких використовують інтервальні числа, недетерміновані коефіцієнти та невідомі постають у вигляді замкнених інтервалів можливих значень, наприклад, $\tilde{c}_i = [c_i^l, c_i^r]$ та $\tilde{x}_i = [x_i^l, x_i^r]$. Такі задачі іноді розв'язують за допомогою методу імовірнісного аналізу, де інтервальне число розглядають як випадкову величину з рівномірним розподілом; або найчастіше вирішують «безпосередньо», користуючись теорією інтервальних чисел. У такому випадку інтервальну задачу методом детермінації [33] зводять до двох аналогічних задач, які визначають нижню та верхню границі вектора невідомих. Цей метод базується на теорії порівняння інтервальних чисел [33], за якою порівняння таких чисел розглядається як порівняння їх відповідних границь – нижньої та верхньої.

Для оперування як кількісними і якісними поняттями в реальних задачах, що розв'язуються за допомогою СППР, пропонується використання апарату нечіткої логіки та теорії нечітких множин, які, у найкращий спосіб є застосовними і прийнятними для класу задач розподілу ресурсів і вибору варіантів, що розглядається.

Доцільно зазначити, що виходячи з публікацій у доступних джерелах, серед неперервнозначних логік у світі немає рівних нечіткій логіці Л.Заде, як за ступенем розробленості, так і за кількістю застосувань.

11.3 Нечітка логіка Л. Заде

Нечітка логіка і теорія нечітких множин були запропоновані Л.Заде [70] у 1965р. Назва «нечітка логіка» припускає, що ця логіка оперує наближеними поняттями; це робить її подібною до людських міркувань. Функціонал істинності в НЛ приймає значення з відрізка $[0, 1]$.

Нехай X – область визначення змінної x . Нечітка множина \tilde{A} , що належить X , визначається функцією належності $\mu_{\tilde{A}}: X \rightarrow [0, 1]$ [69, 70, 71]. Значення функції $\mu_{\tilde{A}}(x)$ являє собою значення належності величини x множині \tilde{A} . Нечітка множина \tilde{A} у загальному випадку може бути графічно проілюстрована як зображено на рис. 11.4.

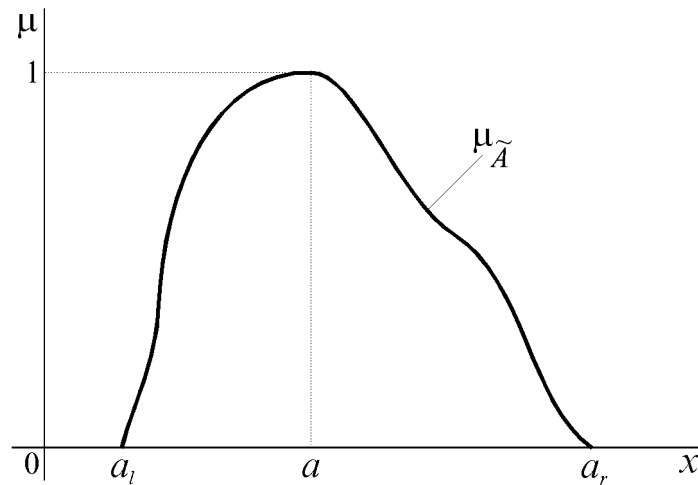


Рис. 11.4 Графічне зображення нечіткої множини \tilde{A}

Носієм нечіткої множини \tilde{A} є звичайна множина з $x \in X$, для яких $\mu_{\tilde{A}}(x) > 0$, тобто інтервал $[a_l, a_r]$ є носієм нечіткої множини \tilde{A} . Нечітка множина \tilde{A} є пустою, якщо $\forall x \in X \mu_{\tilde{A}}(x) = 0$. Величина $\mu_{\tilde{A}}(a) = \sup_{x \in X} \mu_{\tilde{A}}(x)$ називається висотою нечіткої множини \tilde{A} . Нечітка множина \tilde{A} є

нормальною, якщо її висота дорівнює 1, тобто $\sup_{x \in X} \mu_{\tilde{A}}(x) = 1$, та субнормальною при $\sup_{x \in X} \mu_{\tilde{A}}(x) < 1$. Нечітка множина \tilde{A} є унімодальною, якщо $\mu_{\tilde{A}}(x) = 1$ лише для одного $x \in X$. Елементи $x \in X$, для яких $\mu_{\tilde{A}}(x) = 0,5$ називаються точками переходу нечіткої множини \tilde{A} .

Нечіткі множини часто характеризують ФН трикутного вигляду (рис. 11.5) та дзвоноподібного вигляду (рис. 11.6). Найбільш поширене застосування набула трикутна симетрична ФН (рис. 11.5), графік якої являє собою рівнобедрений трикутник. Такий вибір спричинений легкістю сприйняття таких ФН при дослідженні задач прийняття рішень та простотою їх практичного застосування у обчислювальних алгоритмах.

Нечітка множина \tilde{A} на інтервалі $[a_l, a_r]$ з ФН трикутного вигляду може бути параметризована трійкою чисел: $\tilde{A} = (a_l, a, a_r)$, тобто

$$\mu_{\tilde{A}}(x) = \begin{cases} (x - a_l)/(a - a_l), & a_l \leq x \leq a, \\ (x - a_r)/(a - a_r), & a \leq x \leq a_r, \\ 0, & \text{інакше.} \end{cases} \quad (11.3)$$

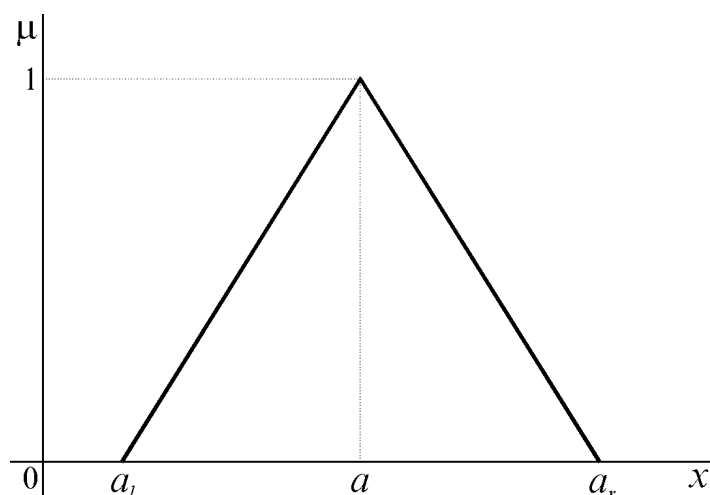


Рис. 11.5 Графік функції належності трикутного вигляду

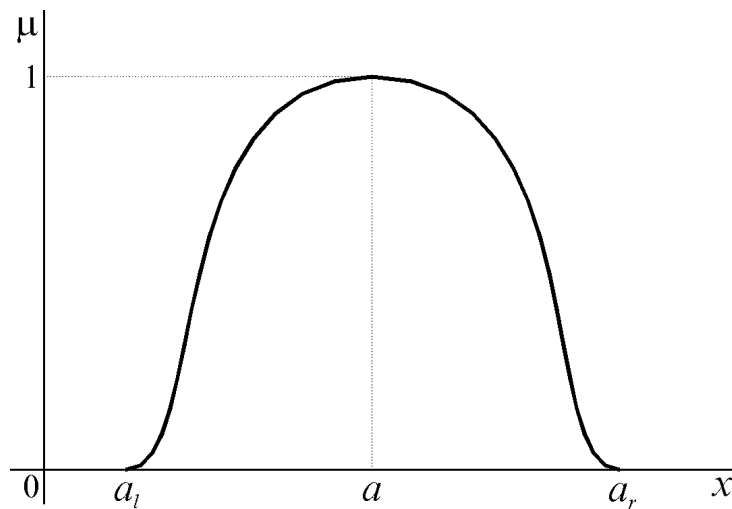


Рис. 11.6 Графік функції належності дзвоноподібного вигляду

Параметри трійки $\tilde{A} = (a_l; a; a_r)$ мають таке призначення: параметр a визначає максимально можливий ступінь приналежності, тобто $\mu_{\tilde{A}}(a) = 1$, параметри a_l, a_r являють собою ліву і праву границю носія нечіткої множини. Так, для точного значення числа a можна записати, що $\tilde{A} = (a; a; a)$.

11.4 Система нечіткого логічного висновку

11.4.1 Схема системи нечіткого логічного висновку

Нечіткий логічний висновок визначає відображення вектора вхідних даних в скалярне вихідне значення за допомогою нечітких правил. Систему НЛВ з багатовимірним виходом розглядають як набір незалежних систем НЛВ з одновимірними виходами.

Як показано на рис. 11.7, система НЛВ складається з трьох основних компонентів: фазифікатора, механізму формування логічного висновку та дефазифікатора.

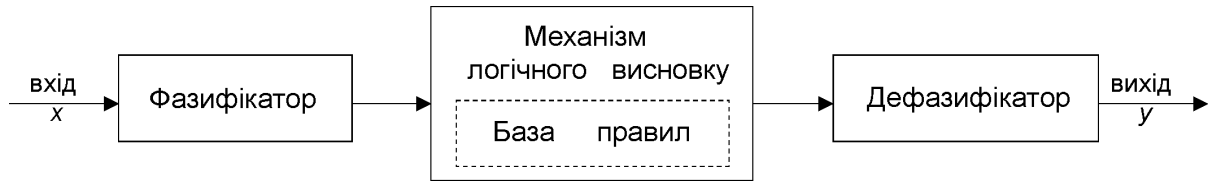


Рис. 11.7 Загальна схема системи нечіткого логічного висновку

Фазифікатор визначає ступінь належності вхідних значень x_i , $i = \overline{1, n}$ до нечітких множин входу – лінгвістичних змінних з відповідної лінгвістичної шкали $T_{x_i} = \{T_{x_i}^1, T_{x_i}^2, \dots, T_{x_i}^{m_{x_i}}\}$, де m_{x_i} – кількість лінгвістичних змінних у шкалі для i -того входу, яка, зазвичай, є рівною для всіх входів. Ця процедура зумовлена використанням у системі НЛВ лінгвістичних правил, вона здійснюється задля визначення ступеню істинності кожної передумови кожного правила.

Ядром механізму логічного висновку є база правил, яка містить лінгвістичні правила. Ці правила можуть бути задані експертним шляхом, чи отримані із числових статистичних даних. Механізм логічного висновку відображає вхідні нечіткі множини $T_{x_i}^{any}$, $i = \overline{1, n}$ кожного правила у вихідну T_y^{any} з набору вихідних лінгвістичних змінних $T_y = \{T_y^1, T_y^2, \dots, T_y^{m_y}\}$. Відмінною рисою НЛВ є те, що порядок виконання правил не впливає на результат – правила виконуються паралельно. Правила в базі правил $Rules = \{Rule_j\}$, $j = \overline{1, r}$ містяться у такому форматі [56]:

$$\begin{aligned}
 Rule_j = & \text{"якщо } x_1 \in T_{x_1}^{any} \text{ і } x_2 \in T_{x_2}^{any} \dots \\
 & \dots \text{ і } x_n \in T_{x_n}^{any}, \text{ то } y^j \in T_y^{any} \text{ " .}
 \end{aligned} \tag{11.4}$$

Далі вихідні нечіткі множини y^j кожного правила об'єднуються в одну нечітку множину висновку \tilde{y} .

Дефазифікатор відображає нечітку множину висновку \tilde{y} у чітке число \bar{y} , яке є результатом системи НЛВ для заданих вхідних значень $x_i, i = \overline{1, n}$. Тобто діапазон вихідних значень дефазифікатор перетворює в одне числове значення зручне для подальшого використання.

11.4.2 Методи дефазифікації

11.4.2.1 Центроїдний метод

На практиці користуються такими поширеними методами дефазифікації [56, 71]: центроїдний, методи максимуму, метод центру максимумів, висотна дефазифікація.

Центроїдний метод (рис. 11.8) полягає у знаходженні центру ваги (центроїду), який і обирається за результат \bar{y} . Для безперервно та дискретно заданих нечітких множин, відповідно:

$$\bar{y} = \frac{\int_a^b y \mu(y) dy}{\int_a^b \mu(y) dy}, \quad \bar{y} = \frac{\sum_{i=1}^n y_i \mu(y_i)}{\sum_{i=1}^n \mu(y_i)}. \quad (11.5)$$

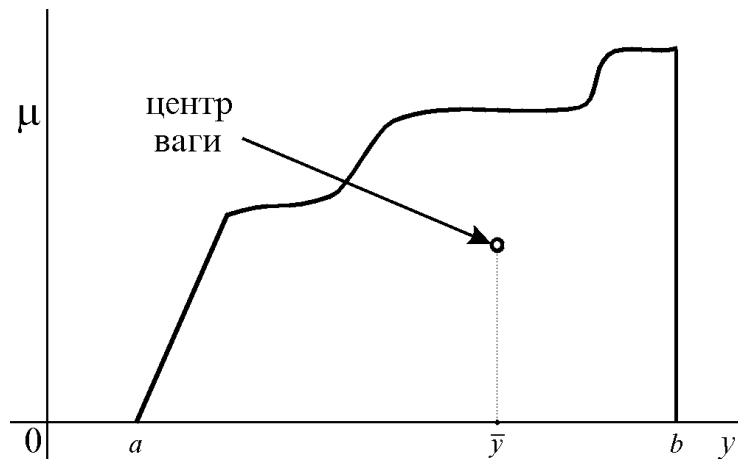


Рис. 11.8 Дефазифікація центроїдним методом

11.4.2.2 Методи максимуму

Методи максимуму, які графічно показані на рис. 11.9, полягають у виборі чіткого результату \bar{y} серед тих значень, для яких ступінь належності $\mu(y)$ є найбільшою. Серед методів максимуму найпоширеніші: метод першого максимуму та метод середнього максимуму.

У методі першого максимуму чітке значення \bar{y} знаходять як найменше значення, при якому досягається найбільше значення ступеню належності $\mu(y)$:

$$\bar{y} = \min \left(y \mid \max_{[a, b]} \mu(y) \right), \quad (11.6)$$

тобто $\bar{y} = \min_{[a_1, b_1]}(y) = a_1$.

У методі середнього максимуму чітке значення \bar{y} знаходять, відповідно, як середнє значення з тих, при яких досягається найбільше значення ступеня належності $\mu(y)$. (Середній максимум на рис. 11.9

позначено через \bar{y}_a .) Для безперервно та дискретно заданих нечітких множин відповідно маємо:

$$\bar{y} = \frac{\int_{a_1}^{b_1} y dy}{\int_{a_1}^{b_1} dy}, \quad \bar{y} = \frac{1}{N} \sum_{i=1}^N y_i, \quad y_i \in [a_1, b_1]. \quad (11.7)$$

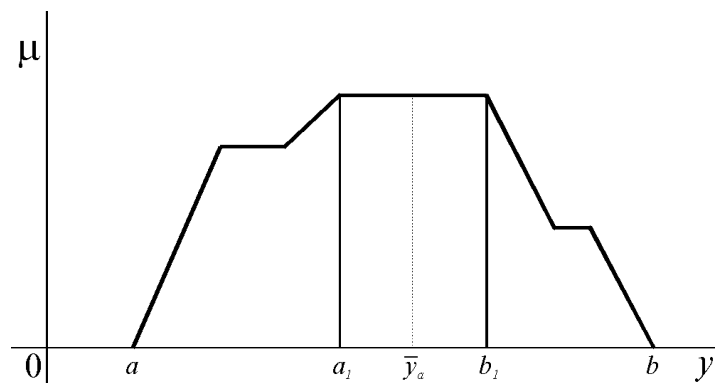


Рис. 11.9 Дефазифікація методами максимумів

Вихід методів максимумів дуже чутливий до домінуючого правила у базі правил.

11.4.2.3 Метод центра максимумів

У методі центра максимумів виходом \bar{y} – це середня точка між центрами областей значень y , при яких функція належності $\mu(y)$ утворює найвищі «плато». Застосування даного методу зображено на рис. 11.10.

Метод висотної дефазифікації [56] полягає у знаходженні центроїду нечіткої множини α -рівня \tilde{y}_α , тобто з нечіткої множини виходу \tilde{y} до уваги приймаються лише ті значення, для яких $\mu(y) \geq \alpha$, $0 \leq \alpha \leq 1$. Для безперервного та дискретного випадку нечітких множин відповідно:

$$\bar{y} = \frac{\int_{y \in \tilde{y}_\alpha} y \mu(y) dy}{\int_{y \in \tilde{y}_\alpha} \mu(y) dy}, \quad \tilde{y}_\alpha = \{y \mid \mu(y) \geq \alpha\}, \quad 0 \leq \alpha \leq 1; \quad (11.8)$$

$$\bar{y} = \frac{\sum_{i=1}^n y_i \mu(y_i)}{\sum_{i=1}^n \mu(y_i)}, \quad y_i \in \tilde{y}_\alpha = \{y \mid \mu(y) \geq \alpha\}, \quad 0 \leq \alpha \leq 1. \quad (11.9)$$

При $\alpha = 0$ з методу висотної дефазифікації (11.8), (11.9) виходить центроїдний метод (11.5).

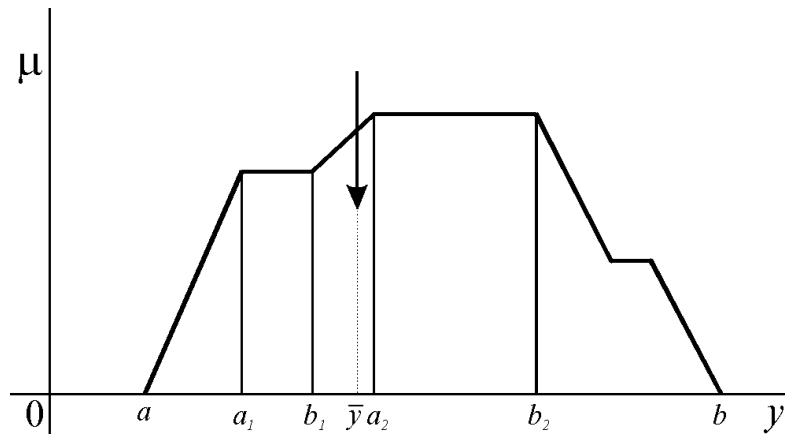


Рис. 11.10 Дефазифікація методом центра максимумів

11.4.3 Модифікації процесу формування нечіткого логічного висновку

11.4.3.1 Нечіткий логічний вивід Мамдані

У сучасній теорії розроблено декілька модифікацій процесу формування нечіткого логічного висновку. Розглянемо найпоширеніші з них.

Нечіткий логічний вивід Мамдані описано у [60]. База правил $Rules = \{Rule_j\}$, $j = \overline{1, r}$ складається з правил у вигляді (11.4). На етапі фазифікації визначаються ступені належності вхідних значень x_i , $i = \overline{1, n}$ до нечітких множин входу, тобто визначаються ступені істинності $\mu_i^j(x_i)$ для кожної передумови i кожного правила j . Далі для кожного правила j на основі ступенів істинності передумов μ_i^j розраховується ступінь його виконання α_j . Для цього застосовують композицію на основі оператора мінімуму:

$$\alpha_j = \min(\mu_1^j(x_1), \mu_2^j(x_2), \dots, \mu_n^j(x_n)), \quad j = \overline{1, r}. \quad (11.10)$$

Далі для кожного правила на основі ступеню виконання α_j , $j = \overline{1, r}$ виконується імплікація, тобто розраховується результат його виконання – вихідна нечітка множина із зрізаною функцією належності $\ddot{\mu}^j(y)$, визначення якої також відбувається за допомогою оператора мінімуму:

$$\ddot{\mu}^j(y) = \min(\alpha_j, \mu^j(y)), \quad j = \overline{1, r}. \quad (11.11)$$

Наприкінці механізму логічного висновку вихідні нечіткі множини виконаних правил за допомогою оператора максимуму агрегуються в нечітку множину висновку \tilde{y} , функція належності якої має такий вигляд:

$$\mu_{\tilde{y}} = \max(\ddot{\mu}^1(y), \ddot{\mu}^2(y), \dots, \ddot{\mu}^r(y)). \quad (11.12)$$

На останньому етапі приведення до чіткості для знаходження остаточного результату \bar{y} користуються будь-яким з наведених вище методів дефазифікації. Нечіткий логічний вивід Мамдані для системи НЛВ з двома входами та двома виконаними правилами графічно ілюструє рис. 11.11.

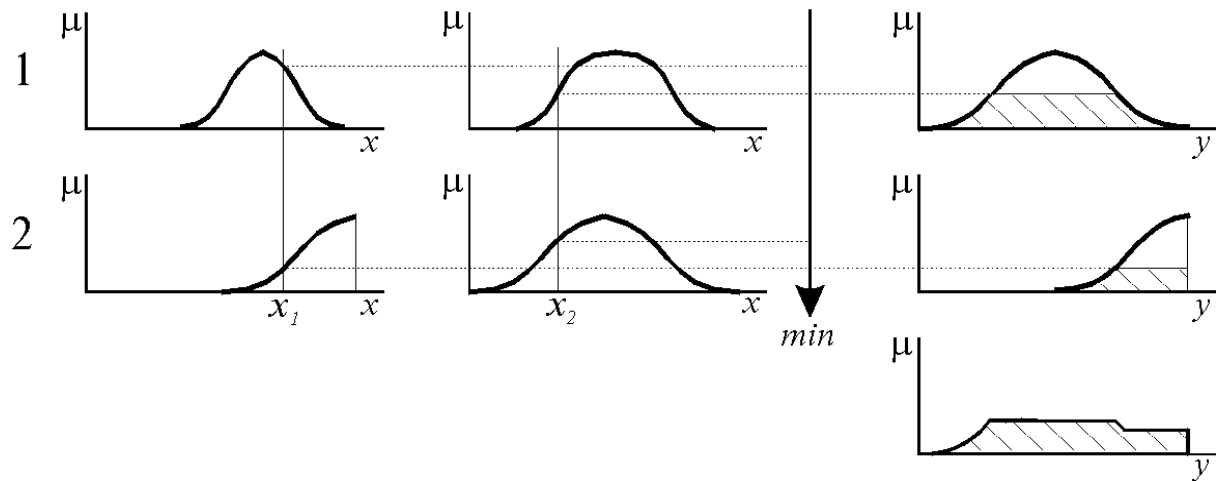


Рис. 11.11 Ілюстрація нечіткого логічного висновку Мамдані

11.4.3.2 Нечіткий логічний висновок Ларсена

Нечіткий логічний висновок Ларсена викладено у [57]. Підхід Ларсена цілком аналогічний до підходу Мамдані, але ФН $\ddot{\mu}^j(y)$ результату виконання кожного правила розраховується на основі оператора добутку, на відміну від використання оператора мінімуму в (11.11):

$$\ddot{\mu}^j(y) = \alpha_j \cdot \mu^j(y), \quad j = \overline{1, r}.$$

Агрегування вихідних нечітких множин виконаних правил в нечітку множину висновку \tilde{y} також відбувається за (11.12).

Графічний приклад процедури нечіткого логічного висновку Ларсена показаний на рис. 11.12.

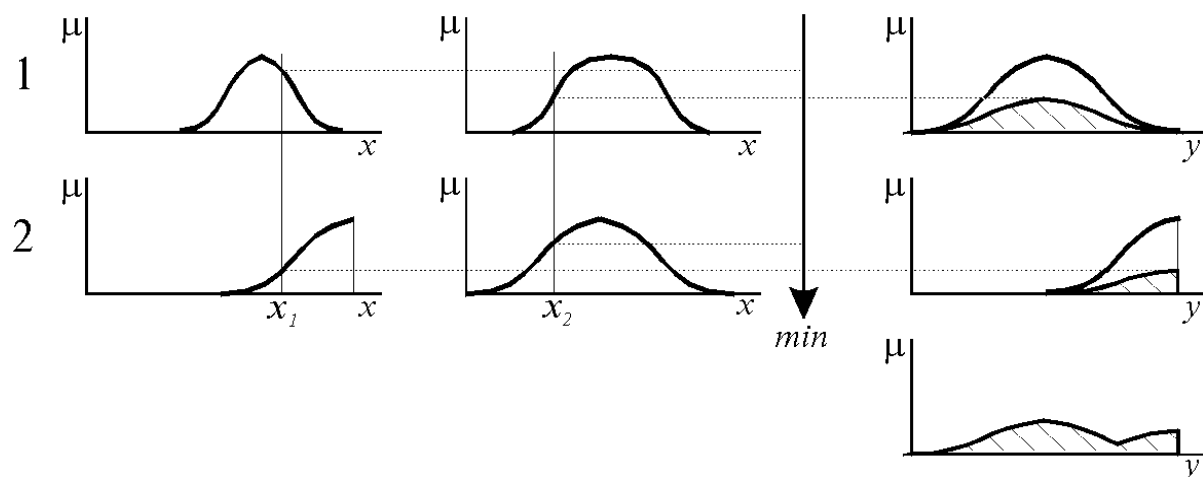


Рис. 11.12 Ілюстрація нечіткого логічного висновку Ларсена

11.4.3.3 Нечіткий логічний висновок Цукамото

Нечіткий логічний висновок Цукамото [4]. Початкові ідеї такі, як у підходах Мамдані та Ларсена, але вважається, що ФН вихідних нечітких множин T_y^{any} є монотонними. У механізмі логічного висновку, як у Мамдані, для кожного правила спочатку розраховується ступінь його виконання α_j за (11.10), а потім відразу знаходиться чітке значення \bar{y}^j результату його виконання. Чітке значення \bar{y}^j є тим, при якому ступінь належності $\mu^j(y)$ вихідної нечіткої множини правила дорівнює ступеню його виконання α_j , та знаходиться як розв'язання відповідного рівняння:

$$\bar{y}^j = \{y \mid \alpha_j = \mu^j(y)\}, \quad j = \overline{1, r}. \quad (11.13)$$

Остаточний результат логічного висновку \bar{y} розраховується як зважене середнє чітких результатів \bar{y}^j виконаних правил, тобто за дискретним варіантом центроїдного методу (9.5):

$$\bar{y} = \frac{\sum_{j=1}^r \alpha_j \cdot \bar{y}^j}{\sum_{j=1}^r \alpha_j} \quad (11.14)$$

Процедуру нечіткого логічного висновку Цукамото ілюструє рис.11.13.

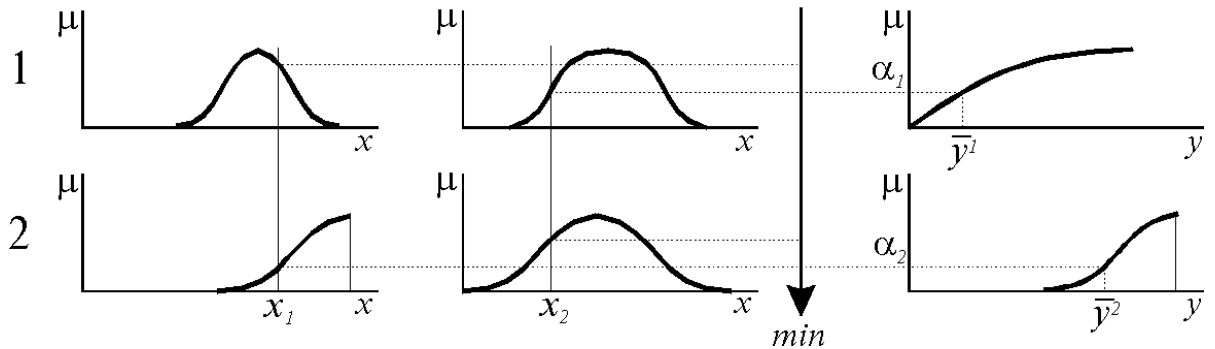


Рис. 11.13 Ілюстрація нечіткого логічного висновку Цукамото

11.4.3.4 Нечіткий логічний вивід Сугено

Нечіткий логічний вивід Сугено [67]. База правил $Rules = \{Rule_j\}$, $j = \overline{1, r}$ складається з правил, у яких виходом є чітке значення, що визначається лінійною функцією:

$$Rule_j = " \text{якщо } x_1 \in T_{x_1}^{any} \text{ і } x_2 \in T_{x_2}^{any} \dots \dots \text{ і } x_n \in T_{x_n}^{any}, \text{ то } \bar{y}^j = \sum_{i=1}^n a_i^j x_i + c^j, " \quad (11.15)$$

де a_i^j ($i = \overline{1, n}$), c^j – константи.

Етап фазифікації такий, як у нечіткому висновку Мамдані. У механізмі логічного висновку для кожного правила розраховується

ступінь його виконання α_j за (11.10), а потім відразу знаходиться чітке значення $\overline{y^j}$ результату його виконання, але, на відміну від Цукамото (11.13), без урахування знайдених ступенів виконання правил α_j , а за формулою, що задана в кожному правилі (11.15).

Остаточний результат логічного висновку \overline{y} розраховується у той самий спосіб, що у Цукамото – як зважене середнє чітких результатів $\overline{y^j}$ виконаних правил (11.14). На рис. 11.14 проілюстровано приклад застосування процедури нечіткого логічного висновку Сугено.

11.4.3.5 Спрощений нечіткий логічний висновок

Спрощений нечіткий логічний висновок [67]. Даний метод по суті є методом нечіткого логічного висновку Сугено нульового порядку, тобто використовується метод Сугено з базою правил, в якій виходом кожного правила є чітке значення, що визначається константою:

$$\begin{aligned} \text{Rule}_j = & \text{" якщо } x_1 \in T_{x_1}^{any} \text{ і } x_2 \in T_{x_2}^{any} \dots \\ & \dots \text{ і } x_n \in T_{x_n}^{any}, \text{ то } \overline{y^j} = c^j \text{ "}, \end{aligned}$$

де c^j – константа.

Графічно процедура спрощеного нечіткого логічного висновку показана на рис. 11.15.

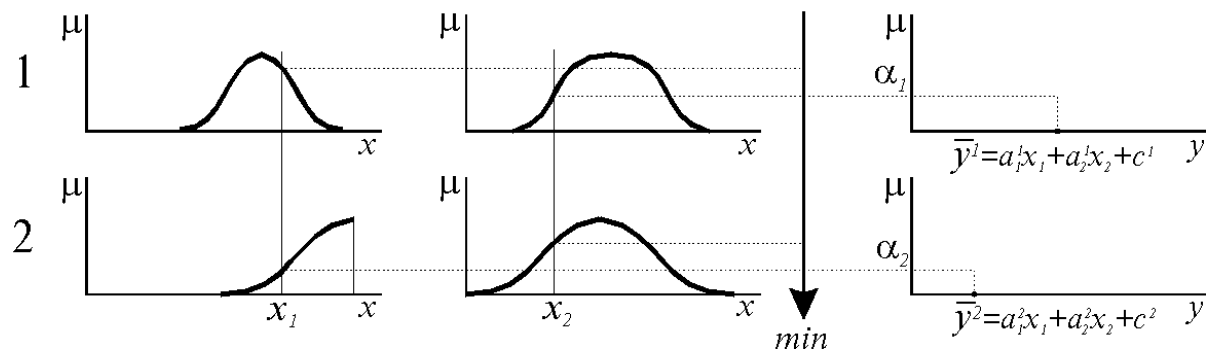


Рис. 11.14 Ілюстрація нечіткого логічного висновку Сугено

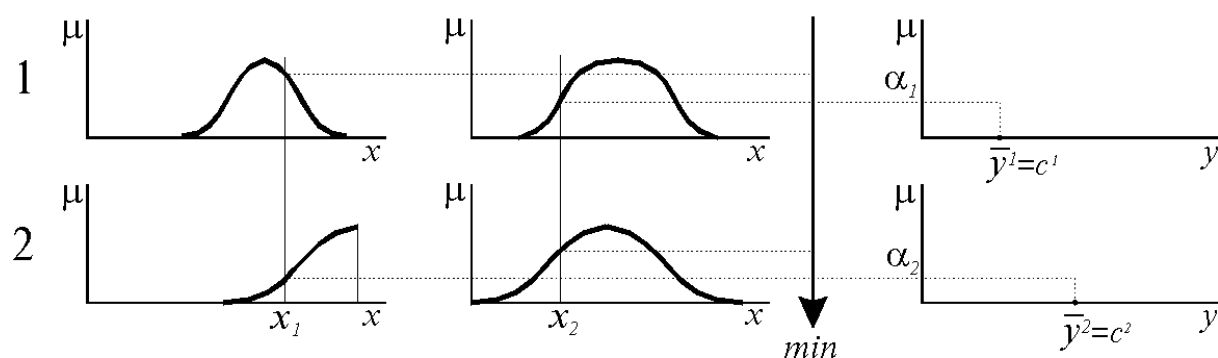


Рис. 11.15 Ілюстрація спрощеного нечіткого логічного висновку

11.5 Експертне оцінювання проектів

11.5.1 Підходи до оцінювання проектів

Оцінки проектів умовно поділяються на дві підмножини – розраховані числові характеристики проектів та експертні оцінки від ОПР. Числові характеристики є супровідними до кожного проекту, проте як механізми одержання експертних оцінок потребують додаткового аналізу і розробки відповідного інструментарію.

Застосування підходу лінгвістичних змінних [21], що розроблений на теорії нечітких множин [69, 71], і також запроваджений Л. Заде надає широкі можливості для опису та оперування якісними нечіткими та невизначеними поняттями, що не можуть бути описані у рамках традиційних математичних формалізмів.

11.5.2 Застосування лінгвістичних змінних

Значеннями лінгвістичних змінних, що постають нечіткими множинами, є слова або фрази повсякденної або синтезованої мови. Лінгвістичні змінні в задачах прийняття рішень досить часто характеризують саме трикутними функціями належності (11.3).

Наприклад, таку характеристику деякого проекту як «важливість» можна оцінити значенням лінгвістичної змінної Imp з такого набору:

$$Imp = \{EN, N, M, I, EI\}, \quad (11.16)$$

де EN = «зовсім не важливий»;

N = «важливість нижче середнього»;

M = «середньої важливості»;

I = «важливий»;

EI = «надзвичайно важливий».

Якщо значення даної лінгвістичної змінної Imp характеризувати на числовому відрізку $[1, 5]$ нечіткими множинами з трикутною ФН, то їх можна представити таким чином: $EN = (1; 1; 2)$, $N = (1; 2; 3)$, $M = (2; 3; 4)$, $I = (3; 4; 5)$, $EI = (4; 5; 5)$.

Графічна інтерпретація нечітких множин значень лінгвістичної змінної Imp , за якою оцінюється важливість проекту, наведена на рис.11.16.

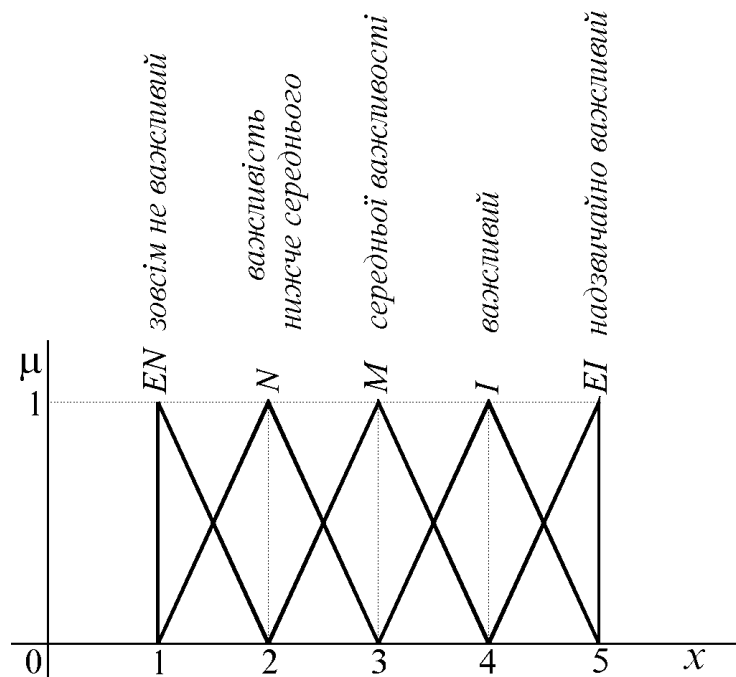


Рис. 11.16 Нечіткі множини значень лінгвістичної змінної Imp

Таким чином, за лінгвістичним підходом для надання експертних оцінок проектів ОПР вибирають значення з множини нечітких лінгвістичних змінних.

Однак, практичні дослідження виявили, що за таким підходом для даного класу задач ОПР надають достатньо грубі оцінки ситуацій, проектів і процесів, що зменшує реалістичність експертних оцінок, та у значній мірі негативно впливає на якість отриманого рішення [37]. Наприклад, при наданні якісних оцінок двом проектам за характеристикою «важливість» вибором значення лінгвістичної змінної із наведеної п'ятибальної шкали лінгвістичних змінних (11.4) ОПР змушена надати цим проектам однакові оцінки I = «важливий», хоча, у дійсності, могла би надати перевагу

другому проекту, наприклад, оцінку $VI = \langle \text{дуже важливий} \rangle$, $VI = (3,5; 4,5; 5)$, яка знаходилась би між змінними $I = \langle \text{важливий} \rangle$ та $EI = \langle \text{надзвичайно важливий} \rangle$.

Збільшення кількості значень наведеної шкали лінгвістичних змінних в деякій мірі збільшує реалістичність експертних оцінок, але не вирішує дану проблему внаслідок того, що градація значень по шкалі лінгвістичних змінних, зазвичай, відбувається у рівномірний спосіб [21, 71]. Крім того, велика кількість значень у шкалі лінгвістичних змінних може спричиняти внутрішній опір у багатьох ОПР [41, 69] під час надання оцінок.

11.5.3 Геометричний спосіб визначення оцінок

Доцільним виявляється підхід одержання якісних нечітких експертних оцінок, який полягає у ранжируванні оцінок на неперервному числовому відрізку та подальшій їх фазифікації – перетворенні у нечіткі множини [21, 38]. За таким підходом ОПР для надання якісних експертних оцінок характеристик проекту чи деякого об'єкту чи процесу використовує прийом геометричного позиціонування показчика між крайніми полярними значеннями, тобто між точками \min та \max , які є мінімально та максимально можливими значеннями оцінок (рис. 11.17).

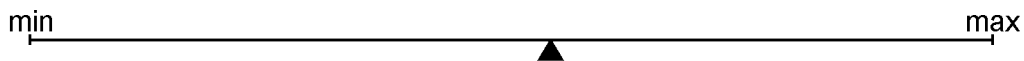


Рис. 11.17 Геометричний спосіб визначення оцінок

У такий спосіб відбувається фактичне ранжирування оцінок на неперервному числовому відрізку $[\min, \max]$, а далі здійснюється перетворення наданих у такий спосіб ОПР значень у нечіткі множини, які і постають якісними експертними оцінками. Визначення нечіткої множини якісної експертної оцінки \tilde{E} при використанні, наприклад, трикутної ФН (11.3) здійснюється таким чином:

$$\tilde{E} = \begin{cases} (\min; e; e+c), & \text{якщо } (e-c) < \min, \\ (e-c, e, \max), & \text{якщо } (e+c) > \max, \\ (e-c, e, e+c), & \text{інакше,} \end{cases} \quad c = \frac{\max - \min}{n-1}, \quad n > 1,$$

де e – дійсне значення позиції показника, яка обрана ОПР, на неперервному числовому відрізку $[\min, \max]$;

c – допоміжний коефіцієнт;

n – ціле значення потенційної лінгвістичної потужності числового відрізку $[\min, \max]$.

Значення n , що є числом значень шкали лінгвістичних змінних, використовується для визначення розміру носія нечіткої множини оцінки за лінгвістичною змінною, яка характеризується на відрізку $[\min, \max]$. Необхідно зазначити, що кількість градацій n лінгвістичної шкали експертного оцінювання доцільно формувати у відповідності до числа Міллера, що дорівнює 7 ± 2 [22]. Чим більше значення n , тим коротшим буде відрізок носія нечіткої множини оцінки.

Наприклад, якщо ОПР виставила показник на позицію, яка відповідає значенню $e = 4,2$, мінімальному можливому значенню \min відповідає значення 1, а максимальному \max відповідно 5. Тому можливу кількість значень n шкали лінгвістичних змінних покладемо 5 і якісній

експертній оцінці буде відповідати нечітка множина \tilde{E} , яка характеризується, при використанні трикутної ФН (11.3) так $\tilde{E} = (3,2; 4,2; 5)$.

За дослідженнями Норвича і Турксена [69] геометричний підхід виявляється дуже легким, не сприяє запам'ятовуванню попередніх оцінок, не збуджує внутрішнього опору у ОПП, дозволяє одержувати реалістичні, більш точні якісні експертні оцінки об'єктів та процесів, зокрема проектів, і значно поліпшує якість отриманих рішень в задачах прийняття рішень, зокрема в задачах розподілу інвестиційних ресурсів.

11.6 Вибір функції належності нечітких понять

11.6.1 Обґрунтування функції належності

У більшості існуючих нечітких моделей для прийняття рішень відсутнє обґрунтування вибору функції належності нечітких понять [4, 57, 0, 69]. В силу простоти застосування в них зазвичай використовуються ФН симетричного трикутного вигляду (11.3), в якій центральне значення a нечіткої множини \tilde{A} є середнім значенням інтервалу-носія $[a_l, a_r]$. Рідше користуються дзвоноподібними (рис. 11.6) і гаусовими ФН, хоча їх практичне застосування також виявляється відносно простим і зручним. Проте такий вибір на користь трикутних чи дзвоноподібних ФН, зазвичай, не сприяє підвищенню якості рішення.

Розглянемо проблему вибору ФН нечітких оцінок критеріїв проекту в задачі розподілу ресурсів. Виконані експерименти з різноманітними ФН, у тому числі і з трикутними та дзвоноподібними, що набули широкого застосування, виявляють залежність якості прийнятого рішення від виду та вигляду ФН, що підкреслює значущість та актуальність даної проблеми.

Однак, встановити характер залежності, який дозволить розробити загальну методику вибору тієї чи іншої ФН неможливо. У деяких випадках, при стійкості рішення до змін ФН, незмінною залишається послідовність остаточного ранжирування проектів $P_i, i = \overline{1, n}$ за узагальненим критерієм, проте чисельна оцінка проектів за цим критерієм змінюється в залежності від ФН. У цьому випадку для задач простого вибору найбільш привабливого проекту P^{opt} з множини альтернатив P за узагальненим критерієм вигляд ФН не впливає на прийняте рішення. А для задач розподілу інвестицій з обмеженнями, вибір ФН виявляється критичним.

11.6.2 Методи побудови функцій належності

Виділяють дві групи методів побудови функцій належності: прямі та непрямі.

В прямих методах безпосередньо задаються правила визначення значень ФН. До прямих методів належать завдання ФН за допомогою таблиці, формули чи прикладу [4, 56, 57, 60]. Зазвичай, прямі методи побудови ФН застосовують для вимірних понять, таких як час, рентабельність, доход та інші, або коли виділяються полярні значення. У багатьох об'єктів в задачах прийняття рішень можливо встановити набір ознак, для яких можна виділити полярні значення, що відповідають значенням функції належності 0 та 1.

Непрямі методи побудови ФН використовують у випадках, коли не існує елементарних вимірних властивостей, через які визначається нечітка множина, що розглядається [69]. У непрямих методах значення ФН обирається таким чином, щоб задовольнялись заздалегідь сформульовані

додаткові умови. Додаткові умови можуть накладатись як на вид одержуваної інформації, так і на процедуру її обробки. Наприклад, додаткові умови можуть бути такими: ФН повинна відображати близькість до заздалегідь виділеного еталону; об'єкти множини є точками у параметричному просторі [56, 69, 71]; ФН задовольняє умовам інтервальної шкали [33]; умови оцінювання об'єктів при попарному порівнянні [69] тощо. Непрямі методи побудови ФН при практичній реалізації виявляються трудомісткішими ніж прямі методи.

В задачах розподілу інвестиційних ресурсів існує два способи одержання нечітких оцінок параметрів інвестиційних проектів: експертні оцінки, що надаються ОПР, та як результат обробки чітких розрахованих числових характеристик проектів. В основі обох способів полягає необхідність визначення та побудови ФН нечітких множин.

11.6.3 Когнітивні функції належності та метод їх побудови

При розробці та застосуванні системи підтримки прийняття рішень з розподілу ресурсів, істотно, що найкращим рішенням задачі буде таке, яке у найбільшій мірі відповідає логіці міркувань ОПР. Дослідження виявляють, що ОПР через суб'єктивне сприйняття нерідко спотворюють оцінки, наприклад, зрушують їх у напрямках кінців оціночної шкали [41, 61, 69], що може спричиняти порушення правил, визначених у концепції лінгвістичних змінних [21].

Результати досліджень зумовили активний розвиток напрямку розробки методів підтримки прийняття рішень, які враховують характеристики та особисті переваги ОПР. Крім цього, при побудові системи підтримки прийняття рішень з розподілу ресурсів, особливо інвестиційних, також необхідно приймати до уваги підвищену

відповідальність ОПР за прийняте рішення, тому для нечітких оцінок ОПР пропонується використовувати ФН, що відповідають аналітичним моделям ОПР. Тобто, пропонується для суб'єктивних оцінок, які надає кожна ОПР, визначати індивідуальну ФН. Таким чином буде здійснено «калібрування» системи підтримки прийняття рішень під множину ОПР.

В літературі з математичної психології можна зустріти розгляд методів шкалювання суб'єктивного сприйняття властивостей [21, 41, 61, 71]. Кожний з них має свої переваги та недоліки, але всі вони достатньо складні та трудомісткі при практичному застосуванні.

11.6.4 Евристичний підхід до побудови функції належності нечітких оцінок

Розглянемо евристичний підхід до побудови ФН нечітких оцінок, що оснований на дослідженнях Норвича і Турксена [69] та полягає у складанні кожною ОПР D_i з групи всіх ОПР $D = \{D_i\}$ спеціально розробленого тесту, який дозволяє виявити індивідуальне сприйняття ОПР. За результатами складання тесту будують індивідуальна ФН F_i^D для кожної ОПР D_i , яка потім узагальнюється на всю множину оцінок, що надаються данною ОПР D_i .

Тест базується на «калібруючій» вибірці, відносно якої ОПР формулює свої переваги, і таким чином виявляється її індивідуальне сприйняття. Вибірка складається з більш ніж 20 понять: істотних зовнішніх подразників ОПР; характеристик об'єктів і явищ реального світу; абстрактних якісних понять тощо.

11.6.5 Методи визначення властивостей у мисленні ОНР

Визначення індивідуальних властивостей у мисленні ОНР відбувається за трьома методами [69]: методом прямого оцінювання, методом зворотного оцінювання та методом відносного оцінювання.

Метод прямого оцінювання. ОНР D_i пропонують оцінку належності μ властивості π об'єкту O , тобто вказати ступінь згоди з тим, що « O має властивість π ».

Метод зворотного оцінювання. ОНР D_i повідомляється оцінка належності μ властивості π деякому об'єкту, та пропонується обрати об'єкт O , котрий, на погляд ОНР, має властивість π зі ступенем μ .

Метод відносного оцінювання. ОНР D_i надає оцінку належності μ_α властивості π об'єкту O_α по відношенню до оцінки належності μ_β властивості π об'єкту O_β .

Запитання тесту пропонуються у випадковому порядку за різними методами. ОНР надає оцінки у тому числі і за допомогою запропонованого вище геометричного прийому (рис. 11.17). За результатами порівнянь відповідей ОНР з еталонними визначається індивідуальна ФН F_i^D ОНР D_i , тобто ФН її нечітких експертних оцінок [37].

Процедури зазначених методів для певної ОНР D_i повинні привести до тієї самої ФН, а їх спільне використання надає можливість виконувати перевірку на еквівалентність та забезпечує необхідну умову існування індивідуальної ФН оцінок ОНР. Для кожної із зазначених процедур для певної ОНР D_i повторні випробування не завжди приводять до збіжних результатів. За остаточну оцінку значення ФН в деякій точці приймається середнє чи медіана. Необхідно зазначити, що в даній ситуації може

спостерігатись ефект субнормальності ФН певної ОПР D_t , тобто значення максимальної належності $\max_{x \in X} F_t^D(x)$ менше повної належності, що дорівнює 1. Це пояснюється тим, що при повторних пред'явленнях певного об'єкту ОПР може в деяких випробуваннях (хоча би в одному) оцінити ступінь належності деякої властивості меншим, ніж в інших.

Тому часто на практиці при знаходженні середньої оцінки значення верхньої границі ФН (висоти) нечіткої множини виходить менше значення, ніж повна належність: $\max_{x \in X} F_t^D(x) < 1$. Отже, визначена ФН F_t^D ОПР D_t піддається такій процедурі нормування:

$$F_t^D(x) = \frac{F_t^D(x)}{\max_{x \in X} F_t^D(x)}, \quad \forall x \in X.$$

За умови, що найбільше значення $\max_{x \in X} F_t^D(x)$ ФН приймає на деякому відрізку носія X , тобто утворюється найвище «плато», ФН піддається перетворенню, за яким найбільше значення $\max_{x \in X} F_t^D(x)$ буде відповідати лише одному значенню $x \in X$ – середині носія зазначеного «плато».

Під час виконання експериментальних досліджень отримано такі важливі спостереження: кожна ОПР керується своїм загальним підходом при оцінюванні різноманітних понять; ніякі два індивідууми не сприймають будь-яку властивість $\pi \in O$ у точності однаково. Людині притаманно не ототожнювати за значенням антонім $antT$ деякого терму T і його заперечення \bar{T} , наприклад, якщо в якості терму взяти «високий», то його антонім «низький» не сприймається синонімом до його заперечення

«невисокий» (саме ця особливість полягає в основі логіки антонімів). Ці результати підкріплюють необхідність застосування при розв'язанні задач прийняття рішень в гуманістичних системах, зокрема задач розподілу інвестицій, ФН нечітких експертних оцінок, які відповідають індивідуальному сприйняттю ОПР; та підтверджують вірність передумов і практичну застосовність запропонованого підходу побудови індивідуальних ФН нечітких оцінок ОПР.

Необхідно зазначити, що аналітичний вираз одержаних у такий спосіб ФН досить складний, але при інтеграції даного підходу у систему підтримки прийняття рішень з використанням засобів обчислювальної техніки не виникає жодних перешкод для успішного практичного застосування цього методу. Приклад побудованої за даним методом ФН експертних оцінок із урахуванням індивідуального сприйняття ОПР зображено на рис. 11.18.

Виходячи з методу одержання таких індивідуальних ФН ОПР та враховуючи те, що такі ФН характеризують індивідуальне сприйняття і мислення експертів, запропоновано називати їх когнітивними функціями належності нечітких експертних оцінок [37].

Даний метод дозволяє одержувати реалістичні, більш точні якісні нечіткі експертні оцінки об'єктів та процесів з урахуванням індивідуального сприйняття ОПР, і значно поліпшує якість отриманих рішень при розв'язанні задач прийняття рішень в гуманістичних системах, зокрема в соціально-економічних.

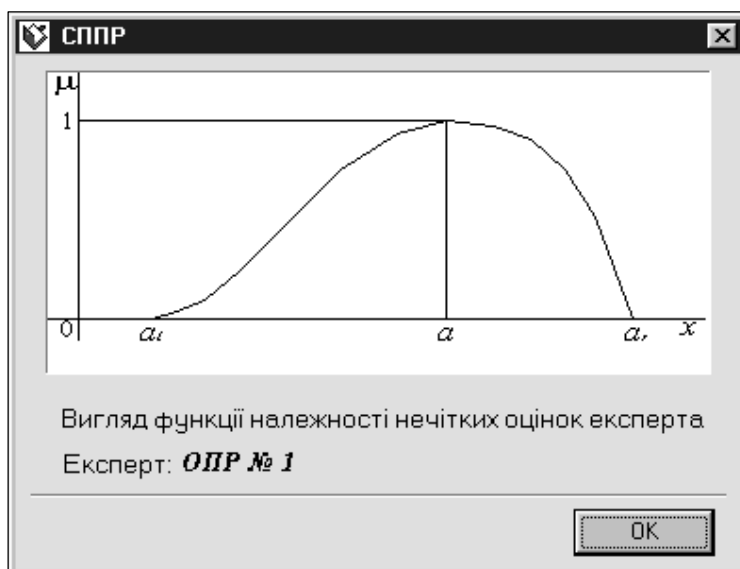


Рис. 11.18 Когнітивна функція належності нечітких експертних оцінок

11.6.6 Наметоподібні функції належності та метод α, β -рівнів їх побудови

Як зазначено вище, в задачах розподілу ресурсів разом із експертними оцінками майже завжди застосовують розраховані числові характеристики проектів, як, наприклад, для інвестиційних проектів: необхідний обсяг інвестиційних ресурсів, рентабельність, строк окупності тощо. При застосуванні підходів, що ґрунтуються на нечіткій логіці та ТНМ, такі чіткі числові характеристики проектів піддають процедурі фазифікації – приведення до нечіткості [56, 71]. Слід зазначити, що при цьому постає проблема побудови набору лінгвістичних змінних та їх функцій належності, яким ставляться у відповідність вхідні дані – числові характеристики проектів.

Існують підходи розв’язання цієї задачі, наприклад [4, 68], що ґрунтуються на обробці навчальної вибірки з вхідних та вихідних даних. На жаль, такі підходи мають суттєві недоліки, оскільки є дуже

громіздкими та трудомісткими, що значно ускладнює їх практичне застосування. Дуже часто логічно некоректною є побудова за такими методами множини лінгвістичних змінних для вхідних даних та їх ФН. Наприклад, на рис. 11.19 зображені нечіткі множини побудованого за методом [68] набору лінгвістичних змінних для вхідної змінної x_1 , що характеризує обсяг активів.

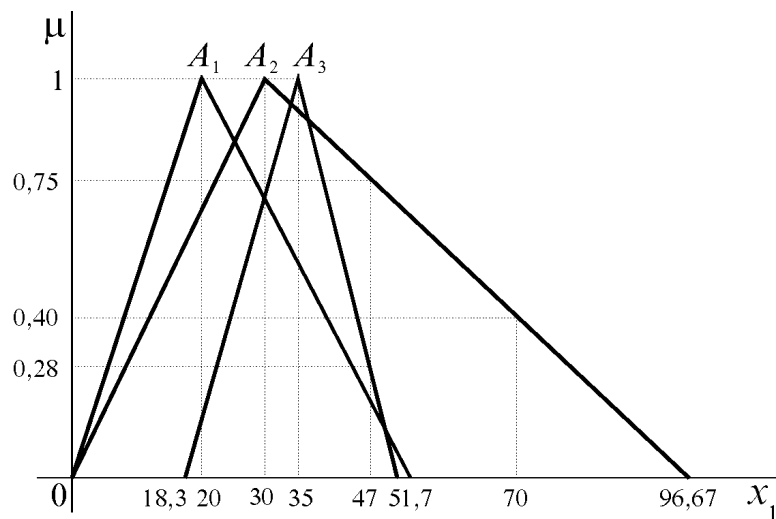


Рис. 11.19 Нечіткі множини набору лінгвістичних змінних для x_1

Нечітким множинам A_1 , A_2 , A_3 можна поставити у відповідність лінгвістичні змінні «малий», «середній», «великий». Отже, значення, наприклад 47, більше значення 35, яке має найбільшу ступінь належності 1 до лінгвістичної змінної «великий». Однак 47 має значно меншу ступінь належності до лінгвістичної змінної «великий», ніж до лінгвістичної змінної «середній»: 0,28 проти 0,75 (рис. 11.19). А значення, наприклад 70, взагалі має нульову ступінь належності до лінгвістичної змінної «великий», проте до лінгвістичної змінної «середній» належить зі ступінню 0,4.

Вочевидь, що такий набір лінгвістичних змінних з їх ФН не може бути використаний при розв'язанні задач розподілу ресурсів, оскільки зазначені методи побудови набору лінгвістичних змінних прямо суперечать принципам лінгвістичного підходу [4, 21, 71]. Забороняється існування у базовій множині пар сусідніх термів, де відсутня істотна розмежованість понять, що апроксимуються термами; та забороняється наявність ділянок області визначення, яким не відповідає жодне лінгвістичне поняття.

Більш того, вищенаведені підходи потребують наявності вибірки даних як з вхідними значеннями так із вихідними. Однак в реальних задачах розподілу інвестиційних ресурсів часто стикаються з відсутністю вибірок з вихідними даними, тому що часто не має знань про результати впровадження та виконання тих чи інших проектів. Все це робить існуючі методи побудови набору лінгвістичних змінних та їх ФН не прийнятними для застосування у розв'язанні реальних практичних задач розподілу інвестиційних ресурсів.

11.7 Побудова набору лінгвістичних змінних на основі кластерів

У роботі [29] запропоновано наступний підхід до побудови набору лінгвістичних змінних та ФН їх значень. Даний підхід застосовується окремо для значень кожної вхідної змінної x_i , $i = \overline{1, n}$.

Нехай є p значень вхідної змінної x_i . Вони упорядковуються за зростанням i , таким чином, утворюють набір: $X_i = \{x_1, x_2, \dots, x_p\}$. Для побудови набору лінгвістичних змінних набір числових значень X_i поділяється на r кластерів G_j , $j = \overline{1, r}$. Кластеризація реалізується за

допомогою « α -перерізів» нечіткого відношення еквівалентності $R^T(x_{p_1}, x_{p_2})$, $x_{p_1}, x_{p_2} \in X_i$ [29]:

$$G_j = \{x_g \mid R^T(x_{p_1}, x_{p_2}) \geq \alpha; \alpha \in [0, 1]; 1 \leq g \leq p, 1 \leq p_1 \leq p, 1 \leq p_2 \leq p\}. \quad (11.17)$$

Нечітке відношення еквівалентності R^T , яке є симетричним, рефлексивним та транзитивним, знаходиться як max–min транзитивне замикання нечіткого відношення близькості $R(x_{p_1}, x_{p_2})$, $x_{p_1}, x_{p_2} \in X_i$:

$$R^T(x_{p_1}, x_{p_2}) = R^k, \quad (11.18)$$

$$R^k = R^{k-1} \circ R, \quad R^{k+1} = R^k, \quad k \geq 2,$$

де « \circ » – операція композиції, яка для $R_1(x_{p_1}, x_{p_2})$ і $R_2(x_{p_2}, x_{p_3})$, $x_{p_1}, x_{p_2}, x_{p_3} \in X_i$ задається таким чином:

$$\mu_{R_1 \circ R_2}(x_{p_1}, x_{p_3}) = \max_{x_{p_2} \in X_i} \min [\mu_{R_1}(x_{p_1}, x_{p_2}), \mu_{R_2}(x_{p_2}, x_{p_3})]$$

Нечітке відношення близькості $R(x_{p_1}, x_{p_2})$, $x_{p_1}, x_{p_2} \in X_i$, яке є симетричним та рефлексивним, визначимо за допомогою Евклідової відстані як:

$$R(x_{p_1}, x_{p_2}) = 1 - \frac{|x_{p_1} - x_{p_2}|}{\delta}, \quad (11.19)$$

де коефіцієнт δ розраховується у такий спосіб:

$$\delta = \frac{\sum_{i=1}^{p-1} |x_i - x_p|}{p-1}. \quad (11.20)$$

Значення x_p є максимальним значенням з упорядкованого набору $X_i = \{x_1, x_2, \dots, x_p\}$. Відношення близькості $R(x_{p_1}, x_{p_2})$ приймає значення з інтервалу $[0, 1]$, тобто якщо $R(x_{p_1}, x_{p_2}) < 0$, то приймаємо $R(x_{p_1}, x_{p_2}) = 0$.

Після проведення кластеризації за правилом (11.17) є r кластерів $G_j, j = \overline{1, r}$, які характеризують шукані лінгвістичні змінні. Далі відбувається побудова ФН для цих лінгвістичних змінних.

Пропонується такий метод побудови ФН μ^j для деякого кластеру G_j , який складається з такого набору значень $G_j = \{x_1^j, x_2^j, \dots, x_{g_j}^j\}$.

Знаходиться умовний центр кластеру c^j як середнє арифметичне між елементами кластеру:

$$c^j = \frac{\sum_{i=1}^{g_j} x_i^j}{g_j}, \quad (11.21)$$

або ж центр кластеру може бути визначеним, наприклад, як медіана - оцінка, яка найменш віддалена від всіх інших значень кластеру за δ (11.20) [29]:

$$c^j = \arg \min_{x_p \in G_j, p=1, g_j} \left(\sum_{i=1}^{g_j} \delta(x_p^j, x_i^j) \right). \quad (11.22)$$

В деяких реалізаціях доцільним може виявитись обрання за центр кластеру середнього значення серед найбільш частих значень – середнього «зони купчастості» кластеру оцінок. У такому випадку функція належності буде проявляти більшу схильність до центрування, тобто значенням, які найбільш часто зустрічаються, будуть відповідати більші значення ступенів належності.

В окремому випадку центр кластеру c^j може дорівнювати деякому значенню з $G_j = \{x_1^j, x_2^j, \dots, x_{g_j}^j\}$. Отже, розрахований за (11.21), (11.22) центр кластеру c^j приймається за значення h^j , при якому ФН кластеру G_j буде мати найбільше значення 1, тобто $\mu^j(h^j)=1$:

$$h^j = c^j. \quad (11.23)$$

Значення h^j будемо називати точкою максимуму ФН кластеру G_j . Розміщення h^j серед значень кластеру виглядає так:

$$\{x_1^j, x_2^j, \dots, x_{l-1}^j, x_l^j, h^j, x_m^j, x_{m+1}^j, \dots, x_{g_j}^j\}, \quad 1 \leq l < m \leq g_j.$$

Далі визначаються лівий μ^{jL} і правий μ^{jR} фрагменти ФН μ^j кластеру для значень, що є меншими та більшими ніж точка максимуму h^j кластеру, відповідно. Для визначення фрагментів ФН пропонується ввести правило α, β -рівнів, за яким найближчому до h^j значенню (x_l^j та x_m^j) ставиться у відповідність ступінь належності, що дорівнює α , а найдалшому (x_1^j та $x_{g_j}^j$) – ступінь належності β , при цьому $\alpha, \beta \in [0, 1]$.

У випадку, коли $x_l^j = x_l^j$ або $x_m^j = x_{g_j}^j$, ці значення вважаються найдалшими. Значення між h^j та найближчими значеннями будуть мати

ступінь належності з інтервалу $[\alpha, 1]$, а між найближчими та найдальшими від h^j будуть мати ступінь належності з інтервалу $[\beta, \alpha]$, обернено пропорційну до їх відстані.

Таким чином, частини лівого та правого фрагментів будуть виглядати так:

$$\mu^{jL}(x) = \begin{cases} \frac{(x - x_l^j)(1 - \alpha)}{h^j - x_l^j} + \alpha, & \text{якщо } x_l^j < x \leq h^j; \\ \frac{(x - x_1^j)(\alpha - \beta)}{x_l^j - x_1^j} + \beta, & \text{якщо } x_1^j < x \leq x_l^j. \end{cases} \quad (11.24)$$

$$\mu^{jR}(x) = \begin{cases} \frac{(x_m^j - x)(1 - \alpha)}{x_m^j - h^j} + \alpha, & \text{якщо } h^j < x \leq x_m^j; \\ \frac{(x_{g_j}^j - x)(\alpha - \beta)}{x_m^j - x_{g_j}^j} + \beta, & \text{якщо } x_m^j < x \leq x_{g_j}^j. \end{cases} \quad (11.25)$$

Далі визначають частини лівого та правого фрагментів ФН, які характеризують значення ступенів належності з інтервалу $[0, \beta]$. Значеннями, які будуть мати найменші, тобто нульові, ступені належності для лівого та правого фрагментів обирають точки максимумів сусідніх кластерів, тобто h^{j-1} і h^{j+1} відповідно, та у такий саме спосіб лінійно продовжують фрагменти ФН до цих значень. Отже, заключні частини фрагментів ФН виглядають так:

$$\mu^{jL}(x) = \begin{cases} \frac{(x - h^{j-1})\beta}{x_1^j - h^{j-1}}, & \text{якщо } h^{j-1} \leq x \leq x_1^j. \end{cases} \quad (11.26)$$

$$\mu^{jR}(x) = \begin{cases} \frac{(h^{j+1} - x)\beta}{h^{j+1} - x_{g_j}^j}, & \text{якщо } x_{g_j}^j < x \leq h^{j+1}. \end{cases} \quad (11.27)$$

Об'єднуючи отримані частини фрагментів ФН (11.24)-(11.27) кластеру як лінгвістичної змінної одержуємо остаточно ФН деякого кластеру G_j (11.28).

Зазначимо, що ФН (11.28) є функцією належності деякого кластеру G_j , який не є ні першим, ні останнім, $j \neq 1$, $j \neq r$. Для першого G_1 та останнього G_r кластерів ФН складається лише з правого та лівого фрагментів відповідно, та їх точки максимумів h^1 і h^r визначаються за іншими правилами.

$$\mu^j(x) = \begin{cases} \frac{(x - h^{j-1})\beta}{x_1^j - h^{j-1}}, & \text{якщо } h^{j-1} \leq x \leq x_1^j; \\ \frac{(x - x_1^j)(\alpha - \beta)}{x_1^j - x_1^j} + \beta, & \text{якщо } x_1^j < x \leq x_l^j; \\ \frac{(x - x_l^j)(1 - \alpha)}{h^j - x_l^j} + \alpha, & \text{якщо } x_l^j < x \leq h^j; \\ \frac{(x_m^j - x)(1 - \alpha)}{x_m^j - h^j} + \alpha, & \text{якщо } h^j < x \leq x_m^j; \\ \frac{(x_{g_j}^j - x)(\alpha - \beta)}{x_{g_j}^j - x_m^j} + \beta, & \text{якщо } x_m^j < x \leq x_{g_j}^j; \\ \frac{(h^{j+1} - x)\beta}{h^{j+1} - x_{g_j}^j}, & \text{якщо } x_{g_j}^j < x \leq h^{j+1}. \end{cases} \quad (11.28)$$

Для першого кластеру G_1 значення x_1^1 вважається найближчим, а $x_{g_1}^1$ – найдалшим від точки максимуму h^1 , при цьому $h^1 < x_1^1$. Значення h^1 розраховується пропорційно, виходячи з того, що h^1 відповідає ступінь належності 1, x_1^1 відповідає ступінь належності α , а $x_{g_1}^1$ – ступінь належності β :

$$h^1 = x_1^1 - \frac{(x_{g_1}^1 - x_1^1)(1 - \alpha)}{\alpha - \beta}. \quad (11.29)$$

ФН для першого кластеру G_1 визначається у такий спосіб:

$$\mu^1(x) = \begin{cases} \frac{(x_{g_1}^1 - x)(\alpha - \beta)}{x_{g_1}^1 - x_1^1} + \beta, & \text{якщо } h^1 \leq x \leq x_{g_1}^1; \\ \frac{(h^2 - x)\beta}{h^2 - x_{g_1}^1}, & \text{якщо } x_{g_1}^1 < x \leq h^2. \end{cases} \quad (11.30)$$

Аналогічно, для останнього кластеру G_r значення $x_{g_r}^r$ вважають найближчим, а x_1^r – найдалшим від точки максимуму h^r , за умови $x_{g_r}^r < h^r$. Значення h^r розраховують пропорційно, виходячи з того, що h^r відповідає ступінь належності 1, $x_{g_r}^r$ відповідає ступінь належності α , а x_1^r – ступінь належності β :

$$h^r = x_{g_r}^r + \frac{(x_{g_r}^r - x_1^r)(1 - \alpha)}{\alpha - \beta}. \quad (11.31)$$

ФН для останнього кластеру G_r визначається у такий спосіб:

$$\mu^r(x) = \begin{cases} \frac{(x - h^{r-1})\beta}{x_1^r - h^{r-1}}, & \text{якщо } h^{r-1} \leq x \leq x_1^r; \\ \frac{(x - x_1^r)(\alpha - \beta)}{x_{g_r}^r - x_1^r} + \beta, & \text{якщо } x_1^r < x \leq h^r. \end{cases} \quad (11.32)$$

Як можна бачити з (11.30) і (11.32), правий та лівий фрагменти ФН відповідно першого та останнього кластеру мають на один вигин менше ніж фрагменти ФН внутрішніх кластерів.

Для даного методу побудови ФН необхідно отримати як мінімум два кластери. У випадку, коли в результаті кластеризації утворюється лише один кластер, він вважається першим і останнім, тобто на ньому, як на носії будуються дві ФН.

Отже, підсумуємо запропонований метод визначення набору значень лінгвістичної змінної відповідної до x_i та побудови ФН її значень за вхідними числовими даними, який запропоновано називати методом α, β -рівнів, у такому алгоритмі [29]:

1. З p значень вхідної змінної x_i утворюється упорядкований за зростанням набір $X_i = \{x_1, x_2, \dots, x_p\}$.

2. На X_i визначається нечітке відношення близькості $R(x_{p_1}, x_{p_2}), \forall x_{p_1}, x_{p_2} \in X_i$ за (11.19).

3. На X_i визначається нечітке відношення еквівалентності $R^T(x_{p_1}, x_{p_2}), \forall x_{p_1}, x_{p_2} \in X_i$ за (11.18) як $\max - \min$ транзитивне замикання нечіткого відношення близькості $R(x_{p_1}, x_{p_2})$.

4. Задається значення граничного коефіцієнту відношення еквівалентності α^{cut} , та потім за допомогою α^{cut} -перерізів нечіткого відношення еквівалентності $R^T(x_{p_1}, x_{p_2})$ набір вхідних значень X_i поділяється на r кластерів $G_j, j = \overline{1, r}$ за (11.17).

5. Якщо кластер один ($r=1$), то вважається, що він є першим та останнім, тобто покладається кількість кластерів рівною двом, а набір значень другого кластеру дорівнюється набору першого ($r := 2, G_2 := G_1$).

6. Задаються значення коефіцієнтів α і β , які будуть визначати ступені належності відповідно для найближчого та найдалшого значень від точки максимуму h^j в кожному кластері $G_j, j = \overline{1, r}$.

7. Для кожного кластеру $G_j, j = \overline{1, r}$ визначається його точка максимуму h^j , при якій ФН μ^j на кластері G_j буде мати найбільше значення 1 ($\mu^j(h^j)=1$). При цьому для першого кластеру G_1 значення h^1 визначається за (11.29). Для останнього G_r значення h^r визначається за (11.31), а для всіх інших кластерів $G_j, j = \overline{2, r-1}$ значення h^j визначається за (11.21)-(11.23).

8. Для кожного кластеру $G_j, j = \overline{1, r}$ за правилом α, β -рівнів будується його ФН μ^j . При цьому для першого кластеру G_1 ФН μ^1 визначається за (11.30). Для останнього G_r ФН μ^r визначається за (11.32), а для всіх інших кластерів $G_j, j = \overline{2, r-1}$ ФН μ^j визначається за (11.28).

Вигляд графіків так званих наметоподібних ФН, що утворюються за цим методом, наведений на рис.11.20.

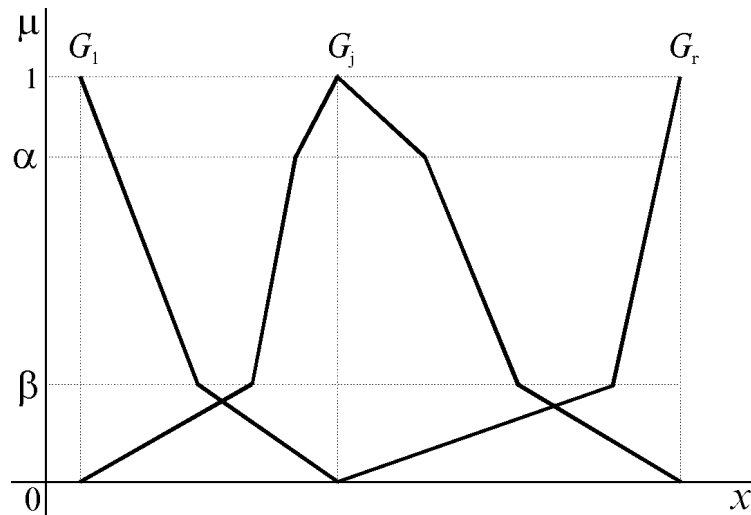


Рис. 11.20 Вигляд графіків наметоподібних ФН нечітких множин, що побудовані на кластерах за α, β -рівнями

11.8 Процедури агрегування та аналізу погодженості експертних оцінок проектів

На даній стадії розв'язання задачі кожний проект $P_i, i = \overline{1, n}$ характеризується своїм набором оцінок $S_{ilt}, l = \overline{1, h}, t = \overline{1, k}$ за кожним критерієм C_l від кожної ОПР D_t . Наступним є етап зведення багатокритеріальної задачі до однокритеріальної, що складається з таких трьох операцій: операції агрегування оцінок проектів від кожної ОПР за критеріями системою НЛВ, операції аналізу погодженості оцінок ОПР та операції агрегування оцінок проектів за ОПР.

Операція агрегування оцінок проектів кожної ОПР полягає у визначенні для кожного проекту $P_i, i = \overline{1, n}$ k узагальнених оцінок від ОПР $D_t, t = \overline{1, k}$, що знаходяться для кожної ОПР D_t шляхом агрегування її оцінок S_{ilt} даного проекту за всіма критеріями $C_l, l = \overline{1, h}, t = \overline{1, k}$.

Традиційні методи агрегування, наприклад, такі як визначення середнього, не прийнятні для значень експертних оцінок, а тому для

знаходження узагальнених агрегованих оцінок проектів скористуємось системою нечіткого логічного висновку.

Додаткові зручності системи НЛВ полягають у тому, що процедура фазифікації є складовою частиною системи НЛВ. Тому для одержання нечітких оцінок \tilde{S}_{it} проектів немає необхідності у попередній фазифікації наявних експертних оцінок S_{it} . Важливо, що результатом системи НЛВ є чітке значення завдяки оскільки процедура дефазифікації є складовою частиною системи НЛВ.

Для операції агрегування оцінок проекту, що одержані від певного експерта, пропонується обрати СНЛВ Мамдані (11.10)-(11.12), (рис. 11.11), що у найбільший спосіб відповідає наявним даним та методологічним вимогам етапу зведення задачі розподілу ресурсів до однокритеріальної. Система нечіткого логічного висновку для одержання узагальнених оцінок проектів має h входів (кількість критеріїв оцінки), та один вихід, який є власне узагальненою оцінкою проекту.

Система НЛВ використовується для кожного проекту P_i , $i = \overline{1, n}$ та для кожної ОПР D_t , $t = \overline{1, k}$ окремо, тобто система НЛВ відпрацьовує $n \cdot k$ циклів. За один цикл на вхід системи НЛВ подаються оцінки S_{it} за критеріями C_l проекту P_i від ОПР D_t , де $l = \overline{1, h}$. На виході система видає узагальнену чітку оцінку A_{it}^D проекту P_i від ОПР D_t .

11.9 Аналіз погодженості оцінок ОПР

11.9.1 Коефіцієнт конкордації Кендала

Вважається, що рішення може бути прийняте тільки на основі погоджених думок експертів [2, 16], тобто лише на основі погоджених узагальнених чітких оцінок A_{it}^D проектів від ОПР.

Поширеними та вживаними методами розрахунку погодженості думок експертів є непараметричні методи визначення коефіцієнту конкордації Кендала та коефіцієнту рангової кореляції Спірмена [2, 31].

Коефіцієнт конкордації Кендала для n об'єктів, що аналізуються, та m експертів розраховується за такою формулою [31]:

$$W = \frac{12}{m^2(n^3 - n)} \cdot \sum_{i=1}^n \left(\sum_{j=1}^m R_{ij} - \frac{m(n+1)}{2} \right)^2, \quad (11.33)$$

де R_{ij} – ранг i -того об'єкту, який наданий йому j -тим експертом.

За наявності в'язок, тобто однакових значень рангів, коефіцієнт конкордації Кендала (11.33) здобуває такий вигляд [31]:

$$W = \frac{1}{\frac{1}{12} m^2(n^3 - n) - m \sum_{j=1}^m T_j} \cdot \sum_{i=1}^n \left(\sum_{j=1}^m R_{ij} - \frac{m(n+1)}{2} \right)^2, \quad (11.34)$$

$$T_j = \frac{1}{12} \sum_{i=1}^{L_j} (n_i^3 - n_i),$$

де L_j – кількість в'язок;

n_i – кількість елементів в i -тій в'язці для j -того експерта.

Коефіцієнт конкордації Кендала W (11.33), (11.34) може приймати значення з інтервалу $[0,1]$. При цьому значення $W = 0$ означає повну непогодженість оцінок експертів, а значення $W = 1$ відповідно означає наявність повної погодженості думок експертів.

11.9.2 Коефіцієнт рангової кореляції Спірмена

Коефіцієнт рангової кореляції Спірмена може бути застосований для визначення погодженості оцінок двох експертів для n об'єктів, що аналізуються, та розраховується таким чином:

$$p = 1 - \frac{6}{n^3 - n} \cdot \sum_{i=1}^n (R_{i1} - R_{i2})^2, \quad (11.35)$$

де R_{i1} та R_{i2} – ранги i -го об'єкту, які надані йому першим та другим експертом відповідно.

Коефіцієнт рангової кореляції Спірмена p (11.35) може приймати значення з інтервалу $[-1,1]$. При цьому значення $p=0$ означає непогодженість оцінок експертів.

Однак, наявність погодженості оцінок експертів, що визначена за такими методами, означає лише відхилення гіпотези про незалежність оцінок ОПР на множині всіх оцінок. Перевірка погодженості у зазначеному статистичному сенсі зовсім не є перевіркою погодженості у сенсі практики експертних оцінок [2, 16, 31].

11.9.3 Жорсткі і м'які методи погодження оцінок

Часто намагаються штучно досягти погодженості оцінок ОПР – для цього вдаються до зменшення впливу оцінок тих ОПР, оцінки яких відрізняються від оцінок більшості, так званих експертів-дисидентів [2, 16, 31]. Жорсткий метод боротьби з такими ОПР полягає у неврахуванні їх оцінок при розв'язанні задачі, тобто фактичне видалення цих ОПР з групи ОПР D .

Одним з поширених підходів є видалення не тільки крайніх оцінок, тобто оцінок з мінімальним та максимальним значеннями, а й усіх, що не належать до більшості [2, 16, 31]. Необхідно зазначити, що у такому разі можуть бути відсіяні оцінки як некваліфікованих ОНР (які потрапили до експертної групи D за непорозумінням або виходячи з міркувань, що не мають відношення до їх професійного та посадового рівня), так і найбільш неординарні ОНР, які глибше збагнули проблему ніж більшість.

Таке відбраковування оцінок ОНР та взагалі відбраковування результатів спостережень, що різко виділяються, як показано у [2], призводить до отримання процедур з поганими або невідомими статистичними властивостями.

М'який метод боротьби з оцінками експертів-дисидентів полягає у застосуванні стійких статистичних процедур на наступному кроці агрегування оцінок ОНР. Наприклад, оцінка, що різко виділяється, сильно впливає на середнє арифметичне оцінок ОНР, та не впливає на їх медіану. Тому в якості погодженої думки експертів, тобто агрегованої оцінки ОНР, можуть розглядати медіану. У кожному з двох наведених методів боротьби з оцінками ОНР-дисидентів відбуваються необґрунтовані з позицій логіко-сислової коректності оцінок неврахування та видалення думок тих чи інших ОНР.

Передумова, що рішення може бути прийняте тільки за наявністю погодженості думок експертів не відповідає реальним практичним задачам прийняття рішень, зокрема задачам розподілу ресурсів. На практиці часто спостерігаються ситуації, коли ОНР поділяються на дві чи більше груп, які мають спільні групові точки зору. В таких ситуаціях фактично виявляється відсутність єдності думок ОНР.

Так, в [36] наведено приклад поділу експертів при оцінюванні результатів науково-дослідних робіт на дві групи: «теоретиків», котрі явно надають перевагу роботам, в яких отримані теоретичні результати, і «практиків», котрі обирають ті роботи, які надають можливість отримувати безпосередні прикладні результати.

Прагнення забезпечити погодженість оцінок ОПР за будь-якої ціни може спричиняти свідомий односторонній підбір експертів, ігнорування всіх точок зору, окрім однієї, яка найбільш подобається та є найбільш вигідною організатору процесу вирішення проблеми.

Існуючі методи і підходи для визначення погодженості оцінок ОПР, зокрема на основі коефіцієнтів Спірмена та Кендала, є неприйнятними для практичних задач розподілу ресурсів, однак все-таки необхідно припускати наявність оцінок некомпетентних експертів, які можуть негативно впливати на якість рішення задачі. Тобто необхідно виявляти та боротись з некомпетентними, несуттєвими та неважливими експертними оцінками.

Таким чином, для успішного вирішення задач прийняття рішень, необхідно не намагатись забезпечити повну погодженість думок експертів, а проводити аналіз реальної ситуації у розподілі їх оцінок, і виходячи з результатів такого аналізу, приймати рішення про доцільність врахування чи неврахування думок тих чи інших експертів.

11.9.4 Загальний підхід до аналізу і формування погоджених оцінок

В загальному випадку підхід до аналізу і формування погоджених експертних оцінок складається з таких етапів [16, 22].

1. Введення метрики. На множині експертних оцінок визначається відношення, за яким для будь-якої пари оцінок можна

встановити міру їх розрізнення.

2. Введення функціонала якості. Функціонал якості дозволяє визначити вагу оцінки, при чому вага оцінки тим більша, чим більш достовірною і обґрунтованою вона є з погляду організаторів експертного оцінювання.

3. Груповий аналіз оцінок. Полягає у виділенні на множині оцінок груп близьких одна до одної оцінок – кластерів, які використовуються для пошуку найбільш погодженої думки експертів відповідної групи.

Для цього для кожного кластеру виконують:

а) знаходження „центра” групи, наприклад, медіани – значення оцінки, яке є у найменшій мірі віддаленим від всіх інших оцінок у відповідному кластері за метрикою з п.1;

б) визначення довірчої множини оцінок, яка складається з оцінок, що найменш віддалені від центру „групи” та мають найбільшу вагу (функціонал якості).

4. Аналіз погодженості думок. В результаті аналізу погодженості думок приходять до підсумкової погодженої оцінки об'єкту.

11.9.5 Метод α, β - коаліцій

Пропонується метод α, β -коаліцій для аналізу і вибору погоджених експертних оцінок. Метод полягає у виявленні груп експертів зі схожими погодженими думками (коаліцій), та у видаленні оцінок експертів, які входять до складу несуттєвих коаліцій. При цьому коефіцієнт α визначає об'єднання експертів у коаліції за значеннями їх оцінок, а коефіцієнт β визначає належність коаліцій до суттєвих чи несуттєвих.

Для кожної ОПР D_t є вектор $A_t^D = \{A_{it}^D\}$, $i = \overline{1, n}$ її узагальнених оцінок усіх проектів P_i , $i = \overline{1, n}$. Поділимо множину векторних узагальнених оцінок ОПР $A^D = \{A_t^D\}$, $t = \overline{1, k}$ на r кластерів G_j , $j = \overline{1, r}$, тобто фактично виділимо r коаліцій ОПР за значеннями їх оцінок.

Для цього скористаємось прийомом, аналогічним до розглянутого у методі α, β -рівнів побудови наметоподібних ФН. Кластеризація відбувається за допомогою α -перерізів нечіткого відношення еквівалентності $R^T(A_{t_1}^D, A_{t_2}^D)$, $A_{t_1}^D, A_{t_2}^D \in A^D$ аналогічно до (11.17):

$$G_j = \{A_t^D \mid R^T(A_{t_1}^D, A_{t_2}^D) \geq \alpha; \alpha \in [0, 1]; A_{t_1}^D, A_{t_2}^D \in A^D\}. \quad (11.36)$$

Нечітке відношення еквівалентності R^T , яке є симетричним, рефлексивним та транзитивним, знаходиться за (11.18) як \max - \min транзитивне замикання нечіткого відношення близькості $R(A_{t_1}^D, A_{t_2}^D)$, яке у свою чергу є симетричним та рефлексивним. Визначимо $R(A_{t_1}^D, A_{t_2}^D)$, $A_{t_1}^D, A_{t_2}^D \in A^D$ за допомогою Евклідової відстані як:

$$R(A_{t_1}^D, A_{t_2}^D) = 1 - \frac{\sqrt{\sum_{i=1}^n (A_{it_1}^D - A_{it_2}^D)^2}}{\delta}, \quad (11.37)$$

де коефіцієнт δ розраховується у такий спосіб:

$$\delta = \frac{\sum_{t=1}^k \sqrt{\sum_{i=1}^n \left(A_{it}^D - \max_t \{A_{it}^D\} \right)^2}}{k-1}.$$

При цьому $\max_t \{A_t^D\}$ є максимальне значення оцінки проекту P_i серед оцінок усіх ОПР D_t , $t = \overline{1, k}$. Відношення близькості $R(A_{t_1}^D, A_{t_2}^D)$ приймає значення з інтервалу $[0, 1]$, тобто якщо $R(A_{t_1}^D, A_{t_2}^D) < 0$, то приймаємо $R(A_{t_1}^D, A_{t_2}^D) = 0$.

Отже, після проведення кластеризації за правилом (11.36) є r кластерів G_j , $j = \overline{1, r}$ векторних оцінок ОПР, які будемо розглядати як коаліції ОПР за схожістю думок щодо оцінювання проектів. Тобто кластер $G_j = \{A_t^D\}$ векторних оцінок ОПР будемо вважати коаліцією $G_j = \{D_t\}$ тих ОПР D_t , оцінки яких входять до складу цього кластеру.

Далі для кожної коаліції G_j , $j = \overline{1, r}$ визначимо її вагу W_j^G як суму вагових коефіцієнтів V_t^{norm} ОПР, які входять до складу коаліції:

$$W_j^G = \sum_{t, D_t \in G_j} V_t^{norm}. \quad (11.38)$$

За значенням ваги W_j^G кожної коаліції G_j на множині всіх коаліцій G визначимо суттєві та несуттєві коаліції. Множину суттєвих коаліцій G^{Ess} утворюють такі коаліції, вага кожної з яких не менше деякого заданого порогу β , а множину несуттєвих коаліцій G^{NonEss} утворюють такі коаліції, вага кожної з яких менше порогу β :

$$G^{Ess} = \{G_j \mid W_j^G \geq \beta, j = \overline{1, r}\}. \quad (11.39)$$

$$G^{NonEss} = \{G_j \mid W_j^G < \beta, j = \overline{1, r}\}. \quad (11.40)$$

Виходячи з правил добору коаліцій у множини суттєвих та несуттєвих коаліцій, можна стверджувати, що

$$G^{Ess} \cup G^{NonEss} = G, \quad G^{Ess} \cap G^{NonEss} = \emptyset.$$

За допомогою порогу суттєвості коаліції β фактично відфільтровуються неважливі експертні оцінки, які самі собою мають незначний вплив на результат та лише «зашумлюють» множину оцінок. ОНР, які входять до несуттєвих коаліцій, та їх оцінки, які утворюють відповідні несуттєві кластери, будуть видалені з подальшого розв'язання задачі. Тобто первісна група ОНР $D = \{D_t\}, t = \overline{1, k}$ після відсіювання тих ОНР, які входять до несуттєвих коаліцій, перетворюється на нову групу ОНР $D^E = \{D_t\}, t = \overline{1, d} (d \leq k)$, до складу якої входять лише ОНР з суттєвих коаліцій:

$$D^E = \{D_t \mid D_t \subset G^{Ess}, t = \overline{1, k}\}; \text{ або} \quad (11.41)$$

$$D^E = \{D_t \mid D_t \not\subset G^{NonEss}, t = \overline{1, k}\}.$$

Таким чином, подальше розв'язання задачі буде відбуватись лише на основі суттєвих узагальнених оцінок проектів P_i від ОНР D_t з D^E :

$$A_{it}^D, i = \overline{1, n}, t = \overline{1, d}. \quad (11.42)$$

Оскільки $d \leq k$, тобто $D^E \subseteq D$, то вагові коефіцієнти ОНР з D^E можуть не відповідати умові

$$\sum_{t=1}^d V_t^{norm} = 1,$$

і тому можуть потребувати корегуального перенормування.

У будь-якому разі для подальшого використання вагові коефіцієнти ОПР з нової групи D^E доцільно піддати додатковій процедурі нормування аналогічно до (11.2):

$$V_t^{norm'} = V_t^{norm} / \left(\sum_{t=1}^d V_t^{norm} \right), \quad t = \overline{1, d}; \quad (11.43)$$

$$\text{та} \quad V_t^{norm} = V_t^{norm'}, \quad t = \overline{1, d}. \quad (11.44)$$

За допомогою (11.43) та (11.44) відносні вагові коефіцієнти V_t^{norm} ОПР набувають нових скорегованих значень, які гарантовано задовольняють умові

$$\sum_{t=1}^d V_t^{norm} = 1.$$

Отже, підсумуємо запропонований метод α, β -коаліцій для аналізу погодженості думок експертів у такому алгоритмі:

1. На множині векторних узагальнених оцінок $A^D = \{A_t^D\}$ ОПР $D = \{D_t\}$, $t = \overline{1, k}$ усіх проектів $P = \{P_i\}$, $i = \overline{1, n}$ (кожній ОПР відповідає вектор її оцінок $D_t \sim A_t^D$) визначається нечітке відношення близькості $R(A_{t_1}^D, A_{t_2}^D)$, $\forall A_{t_1}^D, A_{t_2}^D \in A^D$ за (11.37).

2. На A^D визначається нечітке відношення еквівалентності $R^T(A_{t_1}^D, A_{t_2}^D)$, $\forall A_{t_1}^D, A_{t_2}^D \in A^D$ за (11.18) як \max - \min транзитивне замикання нечіткого відношення близькості $R(A_{t_1}^D, A_{t_2}^D)$.

3. Задається значення граничного коефіцієнту відношення еквівалентності α – коефіцієнту кластеризації, та потім за допомогою α -перерізів нечіткого відношення еквівалентності $R^T(A_{t_1}^D, A_{t_2}^D)$ за (11.36) множина векторних узагальнених оцінок ОНР A^D поділяється на r кластерів $G_j, j = \overline{1, r}$, тобто визначаються r коаліцій $G_j, j = \overline{1, r}$ ОНР за погодженістю їх оцінок.

4. Для кожної коаліції $G_j, j = \overline{1, r}$ за (11.38) визначається її вага W_j^G .

5. Задається значення коефіцієнту β – порогу суттєвості коаліції, та потім за (11.39) і (11.40) утворюються множини відповідно суттєвих G^{Ess} і несуттєвих G^{NonEss} коаліцій.

6. Утворюється нова група суттєвих ОНР $D^E = \{D_t\}, t = \overline{1, d}$ за (11.41) та за (11.42) новий набір суттєвих узагальнених оцінок A_{it}^D проектів P_i відповідний до цих ОНР D_t . У подальшому розв'язанні задачі розглядаються лише такі суттєві ОНР та їх оцінки.

7. Здійснюється перенормування вагових коефіцієнтів V_t^{norm} ОНР з нової групи $D^E = \{D_t\}, t = \overline{1, d}$ за (11.43) та (11.44).

Розглянутий метод виявляє певну гнучкість. Наприклад, якщо значення порогу суттєвості коаліції β покласти рівним значенню ваги керівника, який несе найбільшу чи навіть всю відповідальність за прийняте рішення, то таким чином можна гарантувати, що його оцінки будуть враховані при розв'язанні задачі, навіть якщо він не вступить ні з ким у коаліцію, а фактично утворить окрему одноосібну коаліцію, що є цілком доцільним за таких умов.

У деяких практичних задачах розподілу ресурсів може бути доцільним проводити аналіз оцінок ОНР окремо для кожного проекту, тобто проводити кластеризацію та видалення несуттєвих оцінок для кожного проекту окремо. У такому випадку в процедурі кластеризації нечітке відношення близькості $R(A_1^D, A_2^D)$ (11.37) буде розраховуватись не для векторних, а скалярних оцінок, тобто прийме вигляд (11.19).

Формування нової групи ОНР за (11.41) необхідно буде пропустити, а процедуру добору суттєвих оцінок необхідно буде виконувати для кожного проекту окремо. Також, залежно від певних умов задачі, кластеризацію можна виконувати не лише за допомогою α -перерізів нечіткого відношення еквівалентності, а за будь-яким придатним правилом, наприклад, за допомогою розглянутих коефіцієнту конкордації Кендала або коефіцієнту рангової кореляції Спірмена.

У такому разі коефіцієнт кластеризації α буде порогом для значень обраного показника кореляції. Однак, може виникнути необхідність вирішення неоднозначності об'єднання ОНР у коаліції за умов відсутності властивості транзитивності у правила кластеризації.

11.10 Агрегування оцінок всіх ОНР

11.10.1 Методи розрахунку агрегованих оцінок

Процедура агрегування оцінок проектів за всіма ОНР полягає у знаходженні остаточної агрегованої оцінки кожного проекту P_i – ступеню привабливості A_i , де $i = \overline{1, n}$. Це здійснюється на основі узагальнених за критеріями оцінок A_{it}^D (від кожної ОНР D_t , $t = \overline{1, d}$) кожного проекту P_i та вагових коефіцієнтів V_t^{norm} ОНР, де $i = \overline{1, n}$, $t = \overline{1, d}$.

Загальнопоширеним методом розрахунку таких зважених агрегованих оцінок є зважена сума [2, 31]:

$$A_i = \frac{\sum_{t=1}^d V_t^{norm} \cdot A_{it}^D}{\sum_{t=1}^d V_t^{norm}}. \quad (11.45)$$

Оскільки вагові коефіцієнти ОПР V_t^{norm} нормовані, то (11.45) буде виглядати як лінійна комбінація:

$$A_i = \sum_{t=1}^d V_t^{norm} \cdot A_{it}^D. \quad (11.46)$$

В загальному випадку узагальнені за критеріями оцінки ОПР A_i^D можуть представляти фактичне розділення думок ОПР на конкурентні групи, які є дещо розбіжними у поглядах, то результат застосування методу (11.45), (11.46) для всього набору оцінок, в якому припускається повна погодженість думок ОПР, може не відображати реальну ситуацію наявності несхожих міркувань та формування угруповань ОПР, і тому може спричиняти спотворення значення спільної, компромісної для всіх ОПР, оцінки проекту [2, 31, 36].

Розглянемо метод агрегування оцінок за міжгруповим консенсусом [30]. Він полягає у визначенні угруповань оцінок ОПР для кожного проекту, знаходженні спільних оцінок для кожної групи, і у подальшому агрегуванні значень цих спільних групових [28].

Визначення угруповань здійснюється для оцінок ОПР D_t , $t = \overline{1, d}$ окремо для кожного проекту P_i , $i = \overline{1, n}$ за допомогою процедури кластеризації аналогічної до процедур у запропонованих вище методі побудови наметоподібних ФН (11.17) і методі аналізу погодженості оцінок ОПР (11.36).

Після проведення кластеризації для кожного з утворених r кластерів G_j , $j = \overline{1, r}$ визначається значення спільної групової оцінки A_{ij}^G . Спосіб, у який здійснюється її визначення, вибирається при розв'язанні конкретної задачі в залежності від логіко-сміслового значення такої групової оцінки.

Поширеним і часто обґрунтованим є визначення медіани кластеру в якості спільної групової оцінки [22]. Медіана M_{ij}^G кластера G_j – це значення оцінки, яка найменш віддалена від інших оцінок в кластері G_j за відношенням еквівалентності R^T (11.18):

$$A_{ij}^G = M_{ij}^G = \arg \min \left(\sum_{t_1 | A_{it_1}^D \in G_j} \sum_{t_2 | A_{it_2}^D \in G_j} R^T(A_{it_1}^D, A_{it_2}^D) \right). \quad (11.47)$$

Проте, оскільки значення оцінок, що утворюють деякий кластер, характеризують схожі думки, то для визначення спільної групової оцінки доцільно застосувати зважену суму оцінок (11.45). Для кластеру G_j визначення агрегованої спільної оцінки A_{ij}^G виглядає так:

$$A_{ij}^G = \frac{\sum_{t | A_{it}^D \in G_j} V_t^{norm} \cdot A_{it}^D}{\sum_{t | A_{it}^D \in G_j} V_t^{norm}}. \quad (11.48)$$

Для кожного кластеру G_j , $j = \overline{1, r}$ визначається його вага W_j^G , тобто вага його агрегованої спільної оцінки A_{ij}^G , як сума вагових коефіцієнтів ОПР, оцінки яких утворюють цей кластер:

$$W_j^G = \sum_{t|A_{it}^D \in G_j} V_t^{norm}. \quad (11.49)$$

Таким чином утворюється набір $A_i^G = \{(A_{ij}^G, W_j^G)\}$, $j = \overline{1, r}$, який складається з пар агрегованих спільних оцінок кластерів A_{ij}^G та відповідних їм вагових коефіцієнтів W_j^G . За цим набором визначається загальна агрегована оцінка проекту P_i , яка буде компромісною для сформованих кластерів-груп оцінок.

Для цього за даним набором A_i^G будують нечітку множину \tilde{A}_i . Ключовими точками такої нечіткої множини є значення агрегованих спільних оцінок A_{ij}^G , а ступенями належності цих ключових точок будуть відповідні значення вагових коефіцієнтів W_j^G . ФН $f_{\tilde{A}_i}(x)$ нечіткої множини \tilde{A}_i утворюється за допомогою сплайн-інтерполяції за значеннями ступенів належності ключових точок.

Через відсутність можливості зробити будь-які припущення з наявних даних щодо вигляду ФН нечіткої множини \tilde{A}_i , побудованої у такий спосіб, доцільним постає використання лінійної сплайн-інтерполяції. Тобто ФН $f_{\tilde{A}_i}(x)$, $x \in [A_{i1}^G, A_{ir}^G]$ нечіткої множини \tilde{A}_i буде складатись з лінійних відрізків, кількість яких складає $r-1$, та які з'єднують значення ступенів належності сусідніх ключових точок, та мати такий вигляд:

$$f_{\tilde{A}_i}(x) = \begin{cases} \frac{(x - A_{i1}^G)(W_2^G - W_1^G)}{A_{i2}^G - A_{i1}^G} + W_1^G, & \text{якщо } A_{i1}^G \leq x < A_{i2}^G; \\ \dots \\ \frac{(x - A_{ij}^G)(W_{j+1}^G - W_j^G)}{A_{ij+1}^G - A_{ij}^G} + W_j^G, & \text{якщо } A_{ij}^G \leq x < A_{ij+1}^G; \\ \dots \\ \frac{(x - A_{ir-1}^G)(W_r^G - W_{r-1}^G)}{A_{ir}^G - A_{ir-1}^G} + W_{r-1}^G, & \text{якщо } A_{ir-1}^G \leq x \leq A_{ir}^G. \end{cases} \quad (11.50)$$

Приклад графічного зображення ФН нечіткої множини \tilde{A}_i наведено на рис. 11.21.

Далі за допомогою застосування до нечіткої множини \tilde{A}_i методу дефазифікації, який обирається в залежності від конкретної задачі, знаходиться чітке значення A_i , що і приймається за шукане агреговане значення. Найбільш придатним і зручним постає центроїдний метод (11.5), за яким визначення A_i виглядає таким чином:

$$A_i = \frac{\int_{A_{i1}^G}^{A_{ir}^G} x f_{\tilde{A}_i}(x) dx}{\int_{A_{i1}^G}^{A_{ir}^G} f_{\tilde{A}_i}(x) dx}. \quad (11.51)$$

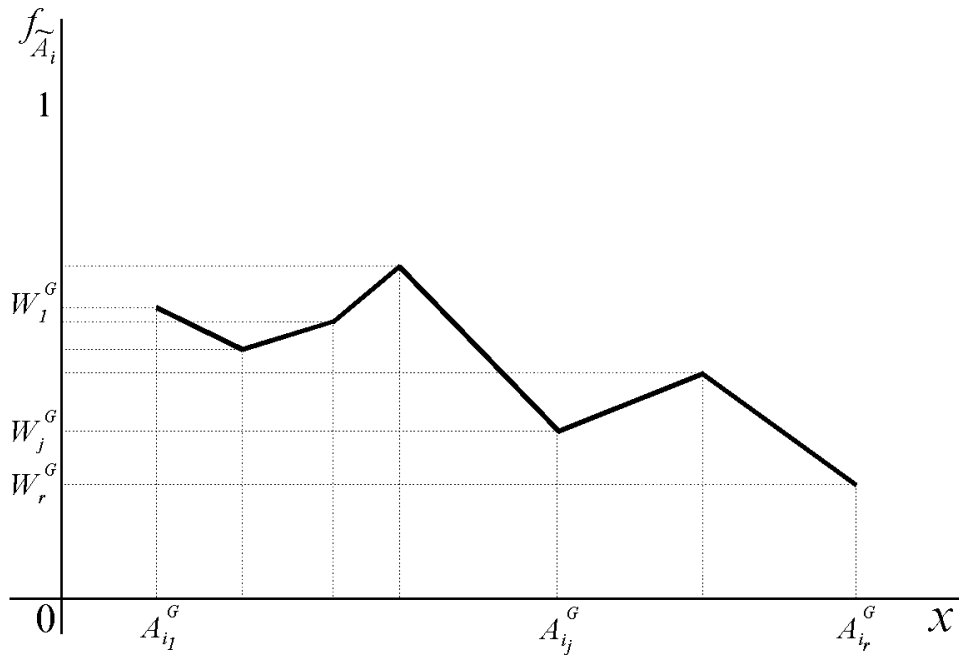


Рис. 11.21 Приклад вигляду графіку ФН нечіткої множини \tilde{A}_i

11.10.2 Алгоритм агрегування оцінок для проекту

Отже, підсумуємо запропонований метод агрегування оцінок для проекту P_i , $i = \overline{1, n}$ за міжгруповим консенсусом у такому алгоритмі.

1. Множина узагальнених оцінок проекту P_i від ОПР D_t , $t = \overline{1, d}$ упорядковується за зростанням в набір $A_i^D = \{A_{it}^D\}$, $t = \overline{1, d}$.

2. На множині A_i^D визначається нечітке відношення близькості $R(A_{it_1}^D, A_{it_2}^D)$, $\forall A_{it_1}^D, A_{it_2}^D \in A_i^D$ за (11.47).

3. На A_i^D визначається нечітке відношення еквівалентності $R^T(A_{it_1}^D, A_{it_2}^D)$, $\forall A_{it_1}^D, A_{it_2}^D \in A_i^D$ за (11.18) як $\max - \min$ транзитивне замикання нечіткого відношення близькості R .

4. Задається значення граничного коефіцієнту відношення еквівалентності α – коефіцієнту кластеризації, та потім за допомогою α -перерізів нечіткого відношення еквівалентності $R^T(A_{it_1}^D, A_{it_2}^D)$ за (11.17) та (11.36) множина узагальнених оцінок A_i^D проекту P_i поділяється на r кластерів $G_j, j = \overline{1, r}$.

5. Для кожного кластеру $G_j, j = \overline{1, r}$ за (11.48) визначається його агрегована спільна оцінка A_{ij}^G .

6. Для кожного кластеру $G_j, j = \overline{1, r}$ за (11.49) визначається його вага W_j^G .

7. За значеннями, що визначені у п.5,6, утворюється набір $A_i^G = \{A_{ij}^G, W_j^G\}, j = \overline{1, r}$.

8. За набором A_i^G будується нечітка множина \tilde{A}_i , ФН якої $f_{\tilde{A}_i}(x), x \in [A_{i1}^G, A_{ir}^G]$ визначається за (11.50).

9. За допомогою дефазифікації нечіткої множини \tilde{A}_i за (11.51) знаходиться чітке значення A_i – агрегована оцінка проекту P_i .

В окремих задачах розподілу ресурсів може виявитись доцільним визначення вагових коефіцієнтів W_j^G кластеру $G_j, j = \overline{1, r}$ не лише за значеннями вагових коефіцієнтів ОПР V_t^{norm} , оцінки яких утворюють кластер, але й з урахуванням потужності кластеру – кількості оцінок $N(G_j)$, що входять до складу даного кластеру. Наприклад, у такому випадку (11.49) може перетворитись на такий вираз:

$$W_j^G = \sum_{t|A_{it}^D \in G_j} V_t^{norm} \frac{N(G_j)}{d}, \quad \text{де } N(G_j) = \sum_{t|A_{it}^D \in G_j} 1. \quad (11.52)$$

Даний метод дозволяє варіювати коефіцієнтом кластеризації, методом визначення ваг кластерів оцінок, методом дефазифікації залежно від конкретної задачі, та надає можливість більш тонко виявляти вплив розподілення між угрупованнями міркувань ОПР у задачі розподілу ресурсів.

11.11 Процедури вибору варіантів

11.11.1 Виділення ресурсів для проекту

Етап вибору проектів є заключним у процесі розв'язання задачі розподілу ресурсів. Процедура вибору проектів для виділення їм ресурсів, що розподіляються, полягає у доборі тих проектів, які максимізують ефективність використання ресурсів та задовольняють ресурсному обмеженню B .

Ефективність виділення ресурсів для проекту P_i характеризується ступенем його привабливості A_i . Ресурсне обмеження B є максимальним обсягом ресурсів, запланованого для виділення проектам.

У реальних практичних задачах загальний обсяг ресурсів найчастіше менший ніж сумарна потреба проектів в ресурсах. Таким чином, процедура вибору проектів постає розв'язанням однокритеріальної задачі вибору, яка може бути сформульована як задача булевого програмування.

Є множина запропонованих проектів $P = \{P_i\} (i = \overline{1, n})$, множина потреб проектів в ресурсах $b = \{b_i\}$, множина визначених ступенів привабливості проектів $A = \{A_i\}$, які відповідають індивідуальним перевагам ОПР, задано

ресурсне обмеження B . Необхідно вибрати проекти, що максимізують сумарну привабливість та задовольняють ресурсному обмеженню.

Для кожного проекту P_i задається змінна x_i , яка може приймати значення 0 або 1, в залежності від обрання проекту для виділення йому ресурсів:

$$x_i = \begin{cases} 0, & \text{якщо проект } P_i \text{ не обирається;} \\ 1, & \text{якщо проект } P_i \text{ обирається.} \end{cases}$$

Задача булевого програмування формулюється таким чином:

$$\sum_{i=1}^n A_i x_i \rightarrow \max, \quad (11.53)$$

за обмеження

$$\sum_{i=1}^n b_i x_i \leq B, \quad x_i \in \{0, 1\}.$$

Задача (11.53) може бути вирішена будь-яким методом цілочисельного програмування дослідження операцій. Для задач великої вимірності рекомендується застосування евристичних підходів, зокрема, генетичних алгоритмів пошуку рішення.

11.11.2 Виділення ресурсів проектам з ресурсним обмеженням

Необхідно зазначити, що на практиці часто постає задача виділення ресурсів усім наявним проектам з ресурсним обмеженням [6, 27]. Наприклад, на державному рівні такою задачею є розподілення фінансових ресурсів між всіма статтями видатків бюджету певної установи між її статтями витрат. У такому разі виділення ресурсів для проектів пропонується здійснювати пропорційно до значень їх ступенів привабливості. Значення обсягу ресурсів I_i , що виділяються проекту P_i , може бути розраховано таким чином:

$$I_i = B \cdot \frac{A_i}{\sum_{ii=1}^n A_{ii}}, \quad I_i \leq b_i, \quad i = \overline{1, n}. \quad (11.54)$$

У деяких практичних задачах розподілу ресурсів, зокрема, інвестиційних задаються бажані потреби кожного проекту у фінансуванні та обмеження на мінімальні допустимі рівні фінансування проектів [6, 27].

В інших практичних задачах розподілу інвестицій запропоновані інвестиційні проекти можуть бути поділені на групи за їх приналежністю до певних галузей, за їх пріоритетністю, тощо, та накладені обмеження по кількості фінансованих проектів з тієї чи іншої галузі, тощо [6, 27]. У будь-якому разі розглянутий підхід до розподілу ресурсів та вибору варіантів є практично застосовним для вирішення такої задачі розподілу інвестицій. В залежності від конкретних умов задачі розподілу ресурсів та обмежень R змінам піддається лише постановка остаточної задачі (11.53) на останньому етапі методу – етапі вибору проектів. Оскільки така задача (11.53) є чіткою задачею однокритеріального вибору, то її

практичне розв'язання без ускладнень може бути здійснене за допомогою відомих методів дослідження операцій, евристичних підходів тощо.

11.11.3 Застосування процедури вибору проектів

Приклад застосування процедури вибору проектів в задачі розподілу ресурсів. Розглянемо застосування процедури вибору проектів на прикладі задачі розподілу інвестиційних ресурсів. В табл. 11.1 наведені дані про потреби проектів P_i в інвестиційних ресурсах b_i в тис. грн та ступені привабливості проектів A_i за десятибальною шкалою. Бюджет фінансування B становить 5 млн грн.

Для розподілу існуючого бюджету 5 млн грн необхідно розв'язати таку задачу за (11.53):

$$7x_1 + 4x_2 + 6,2x_3 + 6x_4 + 9,4x_5 + 3x_6 + \\ + 6,3x_7 + 5,84x_8 + 3,5x_9 + 7,2x_{10} \rightarrow \max'$$

за обмежень

$$\begin{cases} 500x_1 + 940x_2 + 620x_3 + 570x_4 + 300x_5 + 800x_6 + \\ + 256x_7 + 752,2x_8 + 1300x_9 + 480x_{10} \leq 5000; \\ x_i \in \{0,1\}. \end{cases}$$

При цьому

$$x_i = \begin{cases} 0, & \text{якщо проект } P_i \text{ не обирається;} \\ 1, & \text{якщо проект } P_i \text{ обирається.} \end{cases}$$

Таблиця 11.1 Дані про проекти прикладу

Проект P_i	Потреби у фінансуванні b_i , тис. грн	Ступінь привабливості A_i
P_1	500	7
P_2	940	4
P_3	620	6,2
P_4	570	6
P_5	300	9,4
P_6	800	3
P_7	256	6,3
P_8	752,2	5,84
P_9	1300	3,5
P_{10}	480	7,2

Розв'язок даної задачі показує, що при інвестуванні восьми з десяти запропонованих проектів на суму 4418,2 тис. грн на погляд ОПР буде досягнута максимальна ефективність 51,94, що визначається як сумарна ступінь привабливості обраних проектів. Розв'язок задачі представлено у табл. 11.2.

Таблиця 11.2 Розв'язок задачі вибору проектів

Проект P_i	P_1	P_2	P_3	P_4	P_5	P_6	P_7	P_8	P_9	P_{10}
x_i	1	1	1	1	1	0	1	1	0	1

Для задачі розподілу всього наявного бюджету серед усіх проектів розв'язок за (11.54) наведено у табл. 11.3. Як видно, лише три проекти з найменшою привабливістю недоотримають повне фінансування.

Таблиця 11.3 Рішення задачі розподілу всього бюджету

Проект P_i	Потреби у фінансуванні b_i , тис. грн	Фінансування I_i , тис. грн
P_1	500	500
P_2	940	579,73
P_3	620	620
P_4	570	570
P_5	300	300
P_6	800	434,80
P_7	256	256
P_8	752,2	752,2
P_9	1300	507,27
P_{10}	480	480

11.12 Побудова СППР для розподілу обмежених ресурсів

11.12.1 Структура інформаційної системи підтримки прийняття рішень

Розглянемо структуру інформаційної СППР при розподілі ресурсів на основі методів експертних оцінок, що складається з чотирьох основних підсистем і передбачає модульно-блочну побудову (рис. 11.22) [26].

Підсистема інтерфейсу користувача призначена для здійснення зв'язку між користувачами СППР та внутрішніми елементами системи і забезпечує ввід та вивід інформації для ОПР і експертів, а також надає доступ до зовнішніх запам'ятовуючих пристроїв ПЕОМ. Інтерфейс дозволяє операторові вводити інформацію, дані, команди, параметри й запити в систему та одержувати вихідну інформацію в зручному для сприйняття вигляді.

Підсистема накопичення і менеджменту даних забезпечує доступ до даних та знань СППР. Наявність Блоку менеджера даних надає можливість побудувати систему, незалежну від архітектури зберігання даних та дозволяє здійснювати ефективний уніфікований обмін даними між всіма структурними елементами СППР.

Така організація зберігання і одержання даних забезпечує можливість вибрати оптимальну систему управління даними для певної практичної задачі та знімає необхідність майбутнього структурного перетворення СППР при впровадженні нових, більш сучасних БД і БЗ.

Головна підсистема СППР через Головний операційний блок забезпечує реалізацію процесу аналізу і розв'язання задачі у відповідності із загальною структурою системної методології розподілу ресурсів і вибору варіантів (рис. 11.22). При цьому для здійснення певних процедур підключаються і застосовуються відповідні модулі, що входять до складу Підсистеми функціональних модулів. Такі модулі призначені для імплементації розроблених методів і підходів, що застосовуються в системній методології, та передбачають можливість подальшого вдосконалення і розвитку без необхідності коригування інших елементів СППР. У певних ситуаціях може виявитись доцільним розробити декілька модулів одного виду на основі різноманітних методів і вибирати один з них в залежності від конкретної задачі.

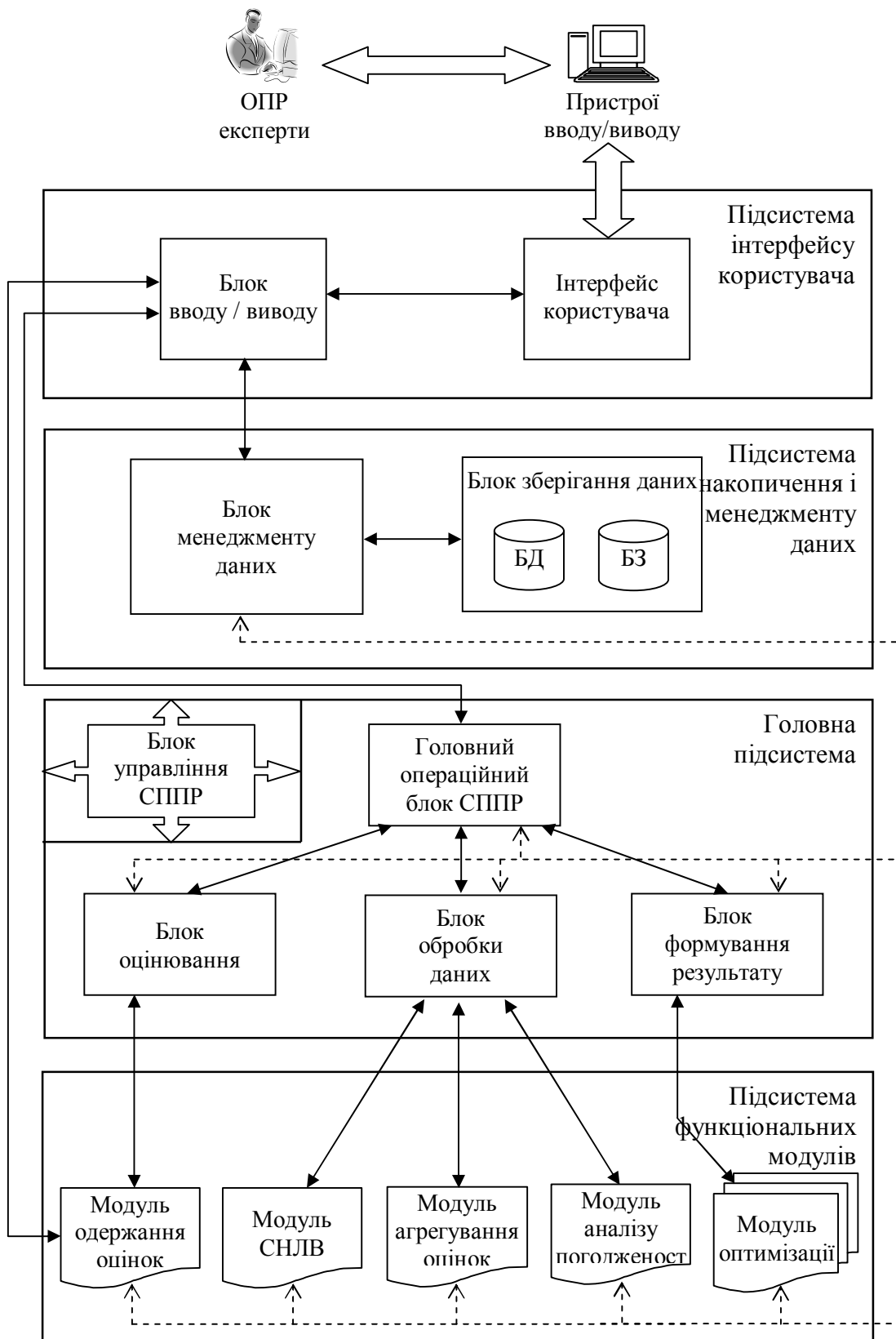


Рис. 11.22 Структура СППР для розподілу ресурсів і вибору варіантів

Головна підсистема СППР також містить службовий Блок управління СППР, який відповідає за режими і настройки функціонування СППР та забезпечує моніторинг і завдання параметрів для всіх підсистем і блоків СППР.

Представлена архітектура СППР легко модифікується до розв'язання задач інших класів, формування нових функцій інтерфейсу (адаптації до користувача), розширення множини функцій обчислювального та логічного характеру, а також можливостей застосування альтернативних методів обробки даних і прийняття рішень.

11.12.2 Застосування СППР для розподілу ресурсів

Розглянемо приклад застосування розробленої СППР для розподілу наявного бюджету B між основними напрямками фінансування галузі охорони здоров'я України. Наявний бюджет складає 3 млрд грн.

Визначено 9 проектів – альтернативних варіантів статей фінансування $P_i, i = \overline{1,9}$, які представлені в таблиці 11.4.

До розв'язання задачі залучена група D з 13 експертів-ОПР $D_t, t = \overline{1,13}$. Задані та нормовані відносні вагові коефіцієнти експертів V_t наведені в таблиці 11.5.

Для оцінювання та розподілу ресурсів між альтернативами експертами запропоновано 20 критеріїв C_l оцінювання і вибору проектів, які складають ієрархічну структуру $C = \{C_l\}, l = \overline{1,20}$ (рис.11.23).

Таблиця 11.4 Альтернативи P_i для фінансування

P_i	Стаття фінансування
P_1	Соціальні програми для медперсоналу (зокрема, забезпечення житлом, заохочення до праці в сільській місцевості, соціальний захист лікарів та медперсоналу тощо).
P_2	Ремонт старого існуючого обладнання та спеціальної медичної техніки.
P_3	Вітчизняне виробництво нового обладнання та спеціальної медичної техніки.
P_4	Закупка нового обладнання та спеціальної медичної техніки за кордоном.
P_5	Фінансування вітчизняних проектів наукових досліджень за біологічними, медичними та ветеринарними напрямками.
P_6	Фінансування установ середньої та вищої медичної освіти.
P_7	Перепідготовка та вдосконалення наявних медичних кадрів (зокрема, стажування за кордоном).
P_8	Фінансування спеціальних оздоровчих програм та заходів (зокрема, фінансування розвитку системи медичного страхування).
P_9	Розвиток матеріальної бази та інфраструктури системи охорони здоров'я (зокрема, будівництва сучасних клінік, забезпечення доступу до світових медичних інформаційних ресурсів).

Таблиця 11.5 Вагові коефіцієнти експертів V_t

Експерт-ОПР D_t	Ваговий коефіцієнт V_t
D_1	0,105263
D_2	0,105263
D_3	0,105263
D_4	0,105263
D_5	0,105263
D_6	0,094737
D_7	0,094737
D_8	0,094737
D_9	0,094737
D_{10}	0,031578
D_{11}	0,021053
D_{12}	0,021053
D_{13}	0,021053

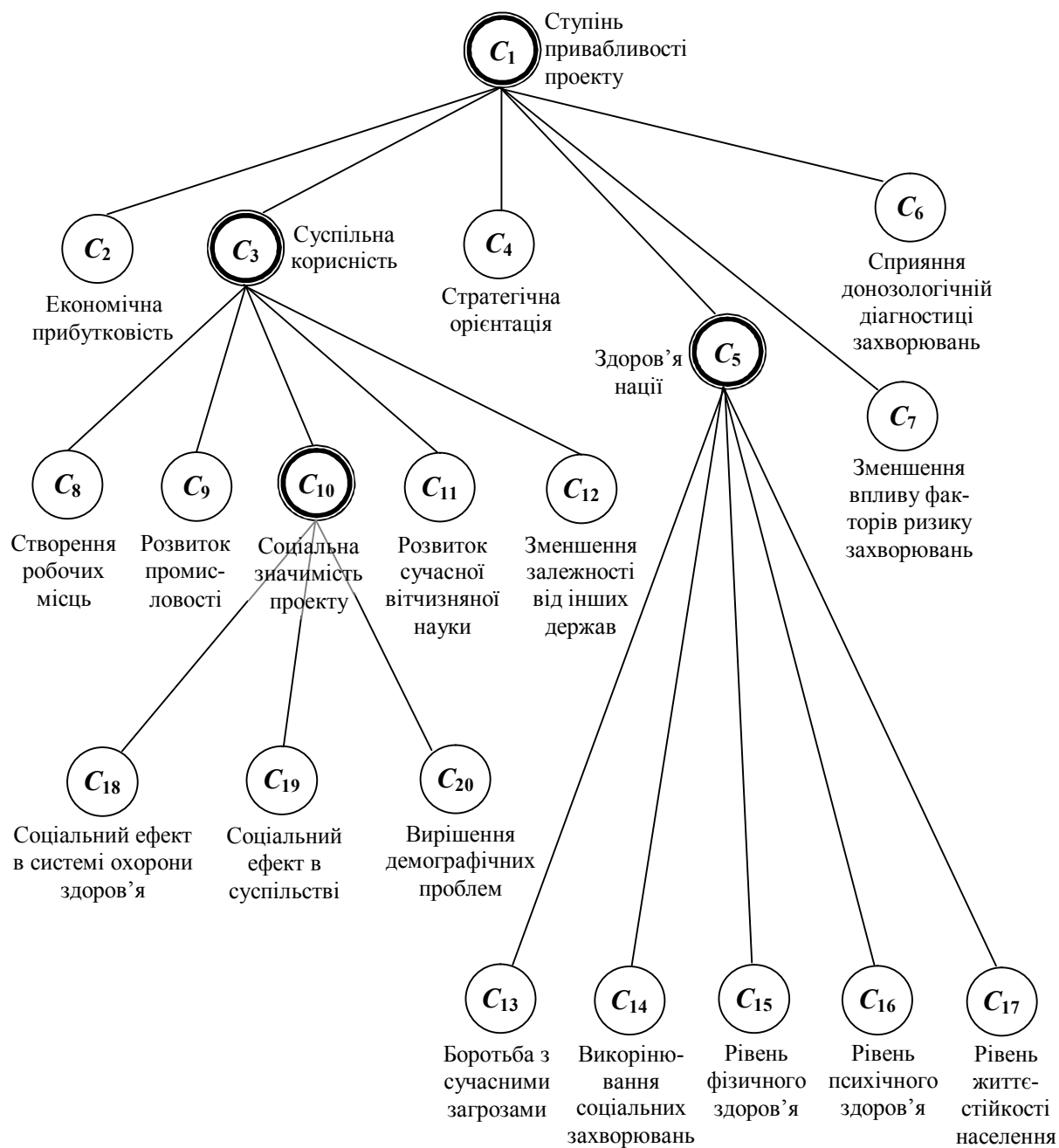


Рис. 11.23 Ієрархічна структура критеріїв оцінювання альтернатив

Критерії C_3 , C_5 , C_{10} є проміжними узагальненими критеріями, а критерій C_1 постає остаточним узагальненим критерієм добору альтернатив – ступінь привабливості проекту. Така структура критеріїв полегшує роботу експертів з контролю та корегування правил логічного висновку, при чому не призводить до погіршення якості результату.

Фактично кожна ОПР буде надавати оцінки альтернативам лише за 16 критеріями, такими, що не є агрегованими. Крім того, кожна ОПР D_t , $t = \overline{1,13}$ задає свої індивідуальні вагові коефіцієнти для критеріїв оцінювання проектів W_{lt} , $l \in \{2,4,6,7,8,9,11,12,13,14,15,16,17,18,19,20\}$. Вагові коефіцієнти агрегованих критеріїв C_3 , C_5 , C_{10} визначаються сумою ваг їх складових критеріїв.

У відповідності до запропонованої експертами структури критеріїв для визначення ступеня привабливості проектів необхідно створити та застосувати чотири системи нечіткого логічного висновку – для визначення значень за кожним агрегованим критерієм C_1 , C_3 , C_5 , C_{10} . Кількість градацій лінгвістичної змінної для кожного вхідного значення СНЛВ покладено 5 (схематично зображено на рис. 11.24), а кількість градацій лінгвістичної змінної для вихідного значення СНЛВ прийнято 7 (схематично зображено на рис. 11.25).

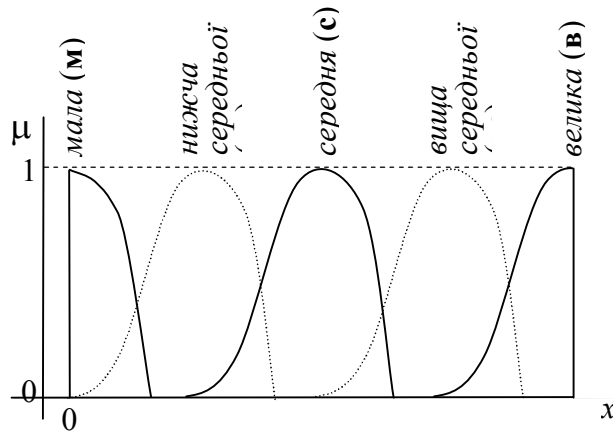


Рис. 11.24 Нечіткі множини значень вхідних лінгвістичних змінних СНЛВ

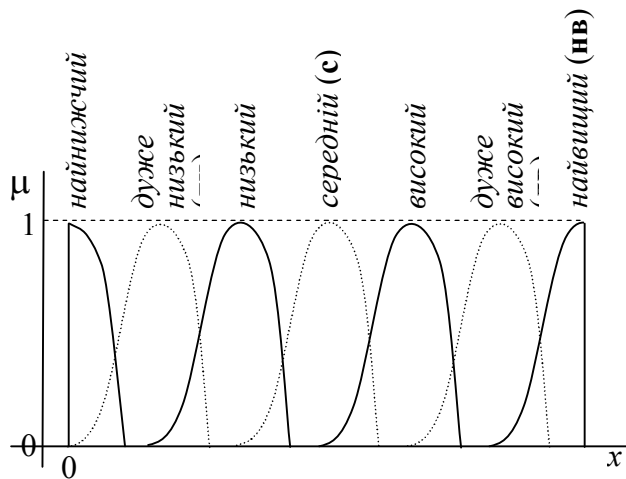


Рис. 11.25 Нечіткі множини значень вихідної лінгвістичної змінної СНЛВ

В результаті критеріального оцінювання проектів, аналізу погодженості експертних оцінок та їх агрегації отримані значення ступенів привабливості A_i для всіх проектів P_i , $i = \overline{1,9}$ (табл. 11.6). Відповідно до значень ступенів привабливості проектів до фінансування, враховуючи наявне бюджетне обмеження B та приймаючи до уваги необхідність фінансування всіх статей (проектів), за пропорційним принципом розрахуємо відносні та абсолютні обсяги B_i фінансування кожного проекту за (11.54):

$$B_i = B \cdot A_i / \sum_{ii=1}^9 A_{ii}, i = \overline{1,9}.$$

Таблиця 11.6 Обсяги виділення ресурсів для альтернатив P_i

Проект P_i	Ступінь привабливості A_i	Фінансування, %	Фінансування B_i , млн грн
P_1	0,410674	10,0	300,0
P_2	0,412623	10,4	312,0
P_3	0,478933	12,1	363,0
P_4	0,431399	10,9	327,0
P_5	0,364861	9,25	277,5
P_6	0,451492	11,4	342,0
P_7	0,531856	13,4	402,0
P_8	0,367292	9,35	280,5
P_9	0,524367	13,2	396,0

Запитання та завдання для самоконтролю

1. Як здійснюється змістовне формулювання задачі прийняття рішень розподілу ресурсів за варіантами?
2. Чим зумовлена невизначеність у задачах розподілу ресурсів?
3. Описати методологію евристичного розв'язання задач розподілу ресурсів.
4. Як застосовують метод експертних оцінок Дельфі?
5. Дати характеристику основних етапів розв'язання задачі розподілу ресурсів в умовах невизначеності.
6. Як структурують критерії оцінювання проектів?
7. Як виконують оцінювання варіантів за множиною критеріїв у задачі розподілу ресурсів?
8. Як багатокритеріальна задача вибору зводиться до однокритеріальної при узагальненні оцінок?
9. Як розв'язують однокритеріальні задачі вибору проектів за узагальненими оцінками?
10. Дати характеристику основних підходів до аналізу невизначеності.
11. У чому полягає нечітка логіка Л. Заде?
12. Як описують нечітку множину у загальному випадку?
13. Пояснити графік функції належності трикутного вигляду.
14. Що таке система нечіткого логічного висновку?
15. Описати схему системи нечіткого логічного висновку.
16. Що таке лінгвістичні правила?
17. Які існують методи дефазифікації?
18. Описати особливості центроїдного методу дефазифікації.

19. Що лежить в основі методів максимуму дефазифікації?
20. Як виконують дефазифікацію методами максимумів?
21. Дати характеристику першого і середнього максимуму.
22. У чому полягає метод центра максимумів?
23. Описати особливості нечіткого логічного виводу Мамдані.
24. У чому полягають основні особливості нечіткого логічного висновку Ларсена?
25. У чому полягають основні ідеї нечіткого логічного висновку Цукамото?
26. Описати основні правила і фазифікацію нечіткого логічного виводу Сугено.
27. У чому полягає спрощений нечіткий логічний висновок?
28. Які існують підходи до експертного оцінювання проектів?
29. У чому полягає застосування лінгвістичних змінних для опису нечітких множин?
30. У чому полягає геометричний спосіб визначення оцінок?
31. Як виконують обґрунтування функції належності для нечітких понять?
32. Які існують методи побудови функцій належності?
33. Що таке когнітивні функції належності та як їх будують?
34. Описати евристичний підхід до побудови функції належності нечітких оцінок.
35. Як визначають методи визначення властивостей у мисленні ОПР?
36. Описати сутність методу α, β -рівнів.
37. Як будують набори лінгвістичних змінних на основі кластерів?
38. У чому полягає визначення функції належності кластеру?
39. Що таке наметоподібні функції належності нечітких множин?

40. Пояснити застосування коефіцієнта конкордації Кендала.
41. Що визначає коефіцієнт рангової кореляції Спірмена?
42. У чому полягають відмінності жорстких і м'яких методів погодження оцінок?
43. Описати особливості загального підходу до аналізу і формування погоджених оцінок.
44. Описати особливості методу α, β -коаліцій.
45. Дати характеристику основних етапів алгоритму методу α, β -коаліцій для аналізу погодженості думок експертів.
46. Як можна агрегувати оцінки всіх ОПР?
47. Які існують методи розрахунку агрегованих оцінок?
48. Описати особливості алгоритму агрегування оцінок для проекту.
49. Які існують Процедури вибору варіантів?
50. Як виділяють ресурси проектам з ресурсним обмеженням?
51. Описати особливості застосування процедур вибору проектів на прикладі задачі розподілу інвестиційних ресурсів.
52. Як описують структуру інформаційної СППР при розподілі ресурсів на основі методів експертних оцінок?
53. Як застосовують СППР для розподілу ресурсів з урахуванням альтернатив фінансування?
54. Описати побудову ієрархічної структури критеріїв оцінювання альтернатив.
55. Вкажіть склад та послідовність етапів застосування методів експертних оцінок.
56. Опишіть структуру методології розв'язання задач розподілу ресурсів і задач вибору.

57. Доведіть необхідність ієрархічної структуризації критеріїв оцінювання.
58. Поясніть особливості узагальнення критеріальних оцінок.
59. Розкрийте зміст етапу зведення багатокритеріальної задачі прийняття рішення до однокритеріальної.
60. Дайте визначення таким поняттям: нечітка множина, функція належності, лінгвістична змінна. Вкажіть зв'язок між ними.
61. Наведіть основні види функцій належності, охарактеризуйте їх.
62. У чому полягають переваги трикутних функцій належності?
63. опишіть роботу і застосування системи нечіткого логічного висновку.
64. Сформулюйте зміст процедур аналізу погодженості оцінок.
65. Наведіть структуру СППР для задач розподілу ресурсів і задач вибору на основі методів експертних оцінок.

РОЗДІЛ 12 ПРОЕКТУВАННЯ СППР

12.1 Визначення типу СППР

СППР переважно застосовують при прогнозуванні динаміки економічних процесів або при розв'язанні конкретних задач. Тому проектування СППР полягає у розробці структури СППР вибраного типу і опису її основних функцій.

Основними типами застосування СППР є:

- при моделюванні та прогнозуванні стаціонарних процесів;
- при прогнозуванні часових рядів, нестаціонарних відносно тренду;
- при прогнозуванні часових рядів, нестаціонарних відносно дисперсії (гетероскедастичні процеси);
- при прогнозуванні нестаціонарних часових рядів;
- при прогнозуванні процесів на основі байєсівського підходу;
- при класифікації образів;
- при менеджменті банківських ризиків;
- при менеджменті економічних ризиків за умов ризику та/або невизначеності;
- при управлінні торговельним або виробничим підприємством;
- при виконанні політичних досліджень;
- при виконанні соціальних досліджень;
- при оптимізації перевезень (товарів, сировини і т.ін.);
- при оптимізації розміщення складів оптової торгівлі на території України;
- при багатокритеріальній оптимізації економічних рішень;
- при оптимізації реалізації приватизаційного процесу;

- при оптимізації розміщення обладнання (верстатів, роботів, складських приміщень) в цеху виробничого підприємства;
- при оптимізації розкрою матеріалів;
- при плануванні кар’єри;
- при виконанні маркетингових досліджень;
- при виборі та придбанні житла;
- при плануванні виробництва конкретної продукції;
- при управлінні якістю на виробництві;
- при діагностуванні та виявленні місця пошкодження (поламки) автомобіля, літака або космічного корабля;
- при медичному діагностуванні (необхідно вибрати конкретний напрям, наприклад, захворювання серця або нирок);
- при проектуванні фільтрів Калмана;
- на основі експертної системи будь-якого призначення.

Характерними для СППР є запити:

- на доповнення типів моделей;
- на розширення множини алгоритмів оцінювання параметрів моделей;
- на доповнення кількості критеріїв адекватності моделей;
- на введення даних (значень ряду);
- на доповнення і редагування даних;
- на побудову моделі ряду;
- на обчислення оцінки прогнозу;
- на виведення результату у заданій формі (таблиця, графік, діаграма, текст тощо);
- на ретроспективний перегляд раніше отриманих результатів моделювання;

- на ретроспективний перегляд раніше отриманих результатів прогнозування;
- на надання допомоги стосовно використання функцій системи;
- на розширення кількості допустимих запитів, тобто ми даємо завдання СППР розширити її власні можливості стосовно інтерактивної взаємодії з користувачем.

Кількість запитів можна розширити, а їх функції модифікувати. Наведений перелік можливих запитів являє собою один із можливих прикладів створення мовної підсистеми.

12.2 Визначення інших типів СППР

За умови вибору реалізації та дослідження СППР іншого типу необхідно сформулювати відповідні запити до неї. Для цього описують конкретні функції, які буде виконувати система обробки даних та генерування результатів (СОДГР) в СППР. Якщо це СППР при прогнозуванні динаміки часових рядів, то необхідно визначити такі елементи:

- типи математичних моделей, які можна будувати за допомогою СППР;
- критерії адекватності, які будуть використовуватись для вибору кращої моделі;
- конкретні типи функцій прогнозування, що будуть використовуватись в вашій системі;
- критерії оцінювання якості прогнозу дозволить обчислювати система;
- правило вибору кращої моделі;

- правило вибору кращого прогнозу.

Якщо система призначена для підтримки прийняття управлінських рішень, то описують метод прийняття рішення та обґрунтування його вибору.

Результатом проведених досліджень повинні бути визначені:

- тип СППР, яку ви вибрали для проектування та реалізації;
- опис призначення підсистем СППР;
- перелік всіх можливих видів запитів до системи;
- перелік функцій системи обробки даних та генерації результатів;
- типи математичних чи статистичних моделей, які ви плануєте використовувати в системі;
- перелік критеріїв вибору моделі при її побудові;
- перелік функцій прогнозування, які будуть використовуватись в СППР (якщо СППР проектується для прогнозування);
- критерії вибору кращого прогнозу чи управлінського рішення.

Необхідними умовами створення проекту СППР та його реалізації є:

- задача, яку передбачається розв'язувати за допомогою СППР, що проектується, повинна бути досяжна людині-експерту у даній галузі за допомогою доступних засобів і методів; повинен бути достатній об'єм знань для створення системи;
- процес прийняття рішення, який планується для реалізації, повинен бути узгодженим із процесом прийняття рішень, який застосовується людьми, тобто враховувати особливості процесу мислення людини; така СППР буде справді практично корисною;
- формулювання задачі проектування повинно бути структурованим, тобто, максимально чітким і зрозумілим, зокрема треба

чітко визначити, яку конкретно підтримку ми хочемо отримати від СППР і в якому об'ємі;

- з самого початку розробки треба визначити, що система не зможе робити; краще створити систему, яка зможе надійно розв'язувати простішу, обмежену задачу, ніж систему, що претендує на розв'язування відносно широкого класу задач, але вона не завжди даватиме вірне рішення;

- необхідно відпрацювати (тестувати) поведінку системи на наборі представницьких (репрезентативних) окремих випадків та сформулювати і описати бібліотеку таких випадків для системи, що проектується;

- треба визначити конкретні знання (теорія, алгоритми, критерії, числа та текстова інформація), які необхідні для розв'язання задачі у вибраній предметній області; це дозволить встановити можливість успішної реалізації системи;

- необхідно передбачити виконання тестування СППР за проектною концепцією на вибраних контрольних прикладах;

- варто описати контрольні приклади і результати їх застосування а також дані, які при цьому використовуються; такий підхід дозволяє уникнути грубих помилок при програмуванні і повторювання однієї й тієї ж роботи;

- при програмуванні необхідно притримуватись правила: «короткі програми (модулі) скорочують шлях до успіху і підвищують надійність системи»;

- можна жертвувати продуктивністю програми, якщо це робить її зрозумілішою і прозорою (в тому числі і в розумінні читання програмного коду), а також спрощує її супроводження.

Часто при розробці комерційних інформаційних систем спочатку розробляють прототип, який може постійно розширюватись і модифікуватись. Прототип хороший тим, що він не занадто складний і дозволяє рекламувати СППР до її завершення, а також за короткі строки тестувати окремі функції. Надалі цей прототип часто трансформується у завершену систему.

Створення прототипу починається з самих простих варіантів системи. Але розробка інших наступних прототипів варто не застосовувати попередні варіанти програмного коду, оскільки спроби подальшого використання неякісних прототипів призводять до марнування часу і фінансів.

12.3 Послідовність побудови СППР управління процесом приватизації

Процес побудови СППР при управлінні процесом приватизації складається з наступних кроків:

А) Аналіз проблеми на заданому рівні:

- галузь;
- кілька суміжних (зв'язаних) галузей;
- на рівні держави;

Б) Визначення входів процесу та інших змінних, що його характеризують (Рис. 12.1);

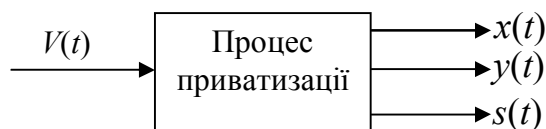


Рис. 12.1 Оптимальні траєкторії процесу приватизації

Основні змінні:

$v(t)$ – швидкість реалізації;

$y(t)$ – поточна зайнятість в галузі;

$x(t)$ – поточний об'єм виробництва в галузі.

Допоміжні змінні:

K – число підприємств, що підлягають приватизації;

T – час досягнення мети (горизонт керування);

$f(t)$ – поточний об'єм продукції, яка виробляється на одному приватному підприємстві;

N – число робітників на приватному підприємстві;

$g(t)$ – продуктивність державного підприємства;

$g(t)/N$ – продуктивність робітника на приватному підприємстві;

$f(t)/n$ – продуктивність робітника на приватному підприємстві;

$h(t)$ – поточне число підприємств, які знаходяться в процесі переходу від державної до приватної форми власності;

n' – число робітників на одному підприємстві, яке знаходиться в процесі переходу до приватної форми власності;

$f'(t)$ – поточна продуктивність підприємства, яке знаходиться в процесі переходу;

$\frac{f'(t)}{n'}$ – продуктивність робітника на підприємстві, яке знаходиться в процесі переходу;

$\alpha(t)$ – коефіцієнт, який характеризує зростання числа підприємств, що переходять до приватної форми власності; при цьому $0 \leq \alpha(t) \leq 1$, $t \in [0, T]$;

$S(t)$ – поточна сума виручки від продажу підприємства;

C_0 – початкова вартість 1 – го підприємства;

$C(t)$ – приріст вартості підприємства в часі.

В) Побудова математичної моделі процесу приватизації

$$x(t) = [f(t) - g(t)] v(t) + \frac{f'(t)}{n'} \alpha(t) [NK - y(t)] \quad (12.1)$$

$$y(t) = - (N - n) v(t) + \alpha(t) [NK - y(t)] \quad (12.2)$$

$$S(t) = v(t) [C_0 + C(t)].$$

Обмеження на змінні:

- обмеження на швидкість приватизації: $v(t) > 0$;
- обмеження на число підприємств, що приватизуються:

$$\int_0^t v(t) dt \leq K ;$$

де K -число підприємств, що приватизуються;

- обмеження на рівень безробіття:

$$y(t) \geq (1 - r) \cdot N \cdot K, \quad 0 \leq r < 1;$$

де r – рівень безробіття.

Початкові умови: $r, K, T, v(0) = v_0 > 0; x(0) = y(0) \cdot K; y(0) = N \cdot K$.

Можливі критерії оптимізації:

- максимізація об'єму виробництва: $I_1 = \max_{v(t) \in c} \int_0^T x(t) dt$;

де c -клас кусочно-неперервних функцій;

- максимізація прибутку від продажу підприємств:

$$I_2 = \max_{v(t)} \int_0^T s(t) d(t);$$

- максимізація зайнятості:

$$I_3 = \max_{v(t)} \int_0^T y(y) d(t).$$

Вибір методів розв'язку:

- варіаційний метод, як найбільш загальний;
- узагальнений метод множників Лагранжа;
- експертний метод (без наведеної моделі).

Оптимальні траєкторії реалізації процесу приватизації представлені на рис. 12.2.

Таким чином, при різних початкових умовах та обмеженнях отримуємо різні траєкторії реалізації процесу, з яких необхідно вибрати прийнятні.

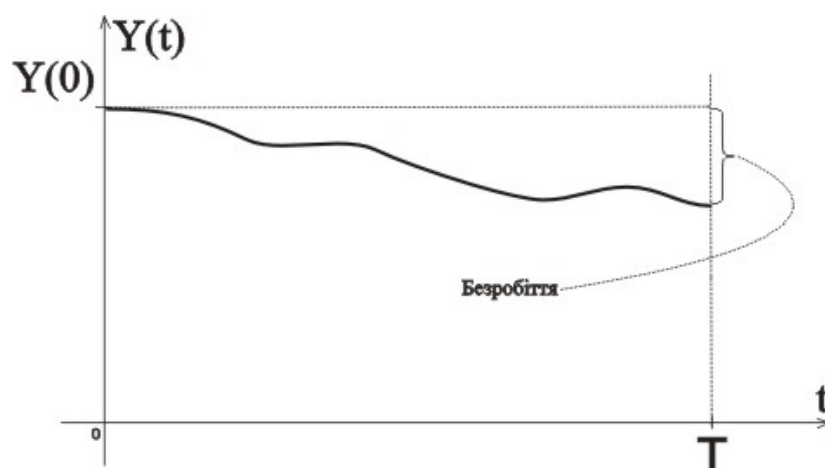


Рис. 12.2 Оптимальні траєкторії процесу приватизації

Ще одним результатом роботи СППР є множина значень критеріїв $I_1^{(1)}, \dots, I_1^{(n)}; I_2^{(1)}, \dots, I_2^{(n)}; I_3^{(1)}, \dots, I_3^{(n)}$, використаних для постановки і розв'язку оптимізаційних задач;

Г) Вибір кращого рішення.

Кращий розв'язок з множини отриманих можна знайти шляхом порівняння чисельних значень критеріїв оптимізації та траєкторії реалізації процесу. Так, не можна вважати прийнятними траєкторії, які закінчуються раніше моменту T (момент T'') або пізніше (момент T'), тому що процес повинен закінчитись за планом в момент T .

Процес вибору прийнятних траєкторій реалізації процесу можна автоматизувати, якщо створити ряд правил типу: якщо $I_1^{(i)} \geq j_1^{(i)}$ і траєкторія $v(t) = 0$ закінчується в момент T і рівень безробіття $\Gamma \leq \Gamma$ задане, то траєкторія прийнятна для реалізації.

Контрольні запитання

1. Як вибирають тип СППР для проектування та реалізації?
2. Які є типи запитів до системи?
3. Які основні функції виконує СОДГР?
4. Які теоретичні знання, алгоритми, критерії та інша інформація необхідні для проектування та реалізації системи?
5. Що може слугувати контрольним прикладом (прикладом) для перевірки СППР?
6. Що таке репрезентативний приклад при тестуванні системи?
7. У яких випадках є виправданим написання довгих складних програм?
8. Для чого створюють прототип системи?

РЕКОМЕНДОВАНА ЛІТЕРАТУРА

1. Айзерман М. А. Выбор вариантов: основы теории / М. А. Айзерман, Ф. Т. Алексеров. – Москва : Наука, 1990. – 240с.
2. Андреенков В. Г. Анализ нечисловой информации в социологических исследованиях / под ред. В. Г. Андреенкова, А. И. Орлова, Ю. Н. Толстовой. – Москва :Наука, – 1985. – 220с.
3. Анфилатов В. С. Системный анализ в управлении / В. С. Анфилатов, А. А. Емельянов, А. А. Кукушкин. – Москва : Финансы и статистика, 2002. – 368 с.
4. Прикладные нечеткие системы / [Асаи, Киедзи, Ватада и др.] Пер. с япон. Под ред. Тэрано и др.– Москва : Мир, 1993. - 368 с.
5. Бидюк П. И. Построение и методы обучения байесовских сетей / П. И. Бидюк, А. Н. Терентьев // Таврический вестник информатики и математики. – 2004. – № 2. – С. 139-153.
6. Бідюк П. І. Аналіз та моделювання економічних процесів перехідного періоду / П. І. Бідюк, О. В. Половцев. – Київ : НТУУ КПІ, 1999. – 230 с.
7. Моделювання та прогнозування нелінійних динамічних процесів / [П. І. Бідюк, І. В. Баклан, Я. І. Баклан, та ін.] – Київ :ЕКМО, 2004. – 120 с.
8. Бідюк П. І. Принципи побудови та застосування мережі Байеса / П. І. Бідюк, Д. В. Шехтер, О. М. Клименко // Наук. вісті НТУУ „КПІ”.- 2005. – № 5.- С. 14-25.
9. Бідюк П. І. Системний підхід до побудови математичних моделей на основі часових рядів / П. І. Бідюк, І. В. Баклан., В. М. Рифа. //

Системні дослідження та інформаційні технології. – 2002. – № 3. – С. 114-131.

10. Бідюк П. І. Часові ряди: моделювання та прогнозування / П. І. Бідюк, О. І. Савенков, І. В. Баклан. – Київ : ЕКМО, 2004. – 144 с.

11. Буч Г. Объектно-ориентированное проектирование с примерами применения / Буч Г; Пер. с англ. – Москва : Конкорд, 1992. - 519 с.

12. Гаврилова Т. А. Базы знаний интеллектуальных систем / Т. А. Гаврилова, В. Ф. Хорошевский. – СПб. : Питер, 2000. – 384 с.

13. Герасимов Б. М. Человеко-машинные системы принятия решений с элементами искусственного интеллекта / Б. М. Герасимов, В. А. Тарасов, И. Б. Токарев. – Київ : Наукова Думка. – 1993. – 184 с.

14. Глухов В. В. Математические методы и методы для менеджмента / В. В. Глухов, М. Д. Медников, С. Б. Коробко. – Москва : Лань, 2000. – 480 с.

15. Глушков В. М. Основы безбумажной информатики / В. М. Глушков. – Москва : Наука, 1987 – 552 с.

16. Гнатієнко Г. М. Експертні технології прийняття рішень / Г. М. Гнатієнко., В. Є. Снитюк. – Київ : ТОВ „Маклаут”, 2008. – 444 с.

17. Деннинг В. Диалоговые системы «Человек-ЭВМ». – Адаптация к требованиям пользователя: Пер. с англ. / Деннинг В., Эссинг Г., Маас С. – Москва : Мир, 1984. – 112 с.

18. Джексон П. Экспертные системы / Джексон П. – Москва , СПб, Київ : Вильямс, 2001. – 624 с.

19. Довгий С. А. Математическое прогнозирование процессов приватизации и инвестирования / С. А. Довгий, А. И. Савенков, П. И. Бидюк. – Київ : НТУУ, 2001. – 232 с.

20. Дрейпер Н. Прикладной регрессионный анализ (т2) / Дрейпер Н., Смит Г. – Москва : Финансы и статистика, 1986. – 366 с.
21. Заде Л. Понятие лингвистической переменной и его применение к принятию приближенных решений / Заде Л. Пер. с англ. Н. И. Ринго под ред. Н. Н. Моисеева и С. А. Орлова. – Москва : Мир, 1976. – 165 с.
22. Згуровский М. З. Системный анализ: проблемы, методология, приложения / М. З. Згуровский, Н. Д. Панкратова. – Київ : Наук. Думка, 2005. – 743 с.
23. Зудилова Е. В. Современное состояние в области проектирования адаптивных систем / Е. В. Зудилова // КИН-96. – Инженерия знаний. – 225 с.
24. Кігель В. Р. Математичні методи ринкової економіки / В. Р. Кігель. – Київ : Кондор, 2003. – 158 с.
25. Коршевнюк Л. А. Проблема поддержки принятия решений при управлении бизнес-процессами на предприятиях / Л. А. Коршевнюк, П. И. Бидюк // Системні технології. – 2000. – 3(11). – С. 40-51.
26. Коршевнюк Л. А. Разработка структуры системы поддержки принятия решений при управлении предприятием / Л. А. Коршевнюк, П. И. Бидюк // Матеріали міжнародної конференції «Автоматика – 2001» – Одеса: ОДПУ, 2001. – Т.2. – С. 49-51.
27. Коршевнюк Л. А. Решение задачи распределения инвестиций между альтернативными проектами / Л. А. Коршевнюк, П. И. Бидюк // Збірник наукових праць Інституту проблем моделювання в енергетиці ім. Г. Є. Пухова. – 2002.– Вип. 17. – С. 26 – 33.
28. Коршевнюк Л. О. Метод дефазифікаційної згортки векторного критерію оптимальності / Л. О. Коршевнюк, П. І. Бідюк // Системні технології.– 2006.-6(47). – С. 158 – 165.

29. Коршевнюк Л. О. Метод побудови наметоподібних функцій належності за α , β -рівнями / Л. О. Коршевнюк., П. І. Бідюк // Наук. вісті НТУУ "КПІ". – 2006.– 5(49).– С. 41 – 47.
30. Коршевнюк Л. О. Підхід групування оцінок в задачах прийняття рішень / Л. О. Коршевнюк, М. Ю. Мінін., П. І. Бідюк. // Информационные технологии в XXI веке: Сб. докладов и тезисов II-го Международного научно-практического форума. – Днепропетровск: ИПК ИнКомЦентра УГХТУ, 2004. – С. 85 – 86.
31. Лапач С. Н. Статистика в науке и бизнесе / С. Н. Лапач, А. В. Чубенко, П. Н. Бабич. – Київ : МОРИОН, 2002. – 640 с.
32. Ларичев О. И. Теория и методы принятия решений / О. И. Ларичев. – Москва : Логос, 2000. – 296 с.
33. Левин В. И. Теория сравнения интервальных величин и ее применение в задачах измерения / В. И. Левин // Измерительная техника. – 1998. – № 5. – С. 23-32.
34. Нейман Дж. Теория игр и экономическое поведение Пер. с англ. / Нейман Дж., Моргенштерн О. – Москва : Наука, 1970. – 707 с.
35. Олексюк О. С. Системи підтримки прийняття фінансових рішень на мікрорівні / О. С. Олексюк. – Київ : Наук. думка, 1998. – 508 с.
36. Орлов А. И. Допустимые средние в некоторых задачах экспертных оценок и агрегирования показателей качества / А. И. Орлов // Многомерный статистический анализ в социально-экономических исследованиях. Москва :Наука, – 1974. – С. 388–393
37. Пат. 71850 А України, МПК G 06 F 17/60, N 7/06. Спосіб одержання експертних оцінок з урахуванням індивідуального сприйняття експерта / Коршевнюк Л. О., Мінін М. Ю., Коршевнюк Д. О. (Україна). – № 20031213216; Заявл. 31.12.2003; Опубл. 15.12.2004, Бюл. «Промислова власність» № 12. – 8 с.

38. Пат. 71851 А України, МПК G 06 F 17/60, N 7/06. Спосіб одержання якісних експертних оцінок при моделюванні економічних, соціальних, біологічних систем / Коршевніук Л. О., Коршевніук Д. О., Мінін М. Ю. (Україна). – № 20031213217; Заявл. 31.12.2003; Опубл. 15.12.2004, Бюл. «Промислова власність» № 12. – 6 с.

39. Попова Э. В. Искусственный интеллект // Системы общения и экспертные системы. Кн.1. Под ред. проф. Э. В. Попова. – Москва : Радио и связь, 1990. – 430 с.

40. Ситник В. Ф. Системи підтримки прийняття рішень. – Київ: КНЕУ, 2004. – 614 с..

41. Солсо Р. Когнитивная психология / Солсо Р. – СПб. : Питер, 2002. – 592 с.

42. Терентьев А. Н. Алгоритм вероятностного вывода в Байесовских сетях / А. Н. Терентьев, П. И. Бидюк, Л. А. Коршевніук // Системні дослідження та інформаційні технології. – 2009. – № 2. – С. 107-111.

43. Цисарь И. Ф. Компьютерное моделирование экономики / И. Ф. Цисарь, В. Г. Нейман. – Москва : Диалог-МИФИ, 2002. – 304 с.

44. Черноруцкий И. Г. Методы оптимизации и принятия решений / И. Г. Черноруцкий. – СПб. : Лань, 2001. – 384 с.

45. Эддоус М. Методы принятия решений / Эддоус М., Стэнсфилд Р. – Москва : Аудит, ЮНИТИ, 1997. – 590 с.

46. Эндрю Ван Дам. Пользовательские интерфейсы нового поколения / Эндрю Ван Дам // Открытые системы. – 1997. – № 6. - С. 32 - 45.

47. Яременко С. В. Реформы менеджмента: Опыт лучших предприятий Украины и России / С. В. Яременко. – Харьков: Фолио, 1999. – 159 с.

48. Alter S. L. Decision support systems : current practice and continuing challenges. / Alter S. L. – Reading, Mass.: Addison-Wesley Pub., 1980.- 120 p.
49. Ardissono L. Dynamic User Modeling and Plan Recognition in Dialogue [Electron resource]/ Ardissono L. // PhD Thesis, Dipartimento di Informatica, Università di Torino. – Italy, www.di.uniti.it/~liliana/, World Wide Web, 1996. – Mode of access: <http://www.di.unito.it/~liliana/EC/tesi.pdf>
50. Bonczek R. H. Foundations of Decision Support Systems / Bonczek R.H., Holsapple C., Whinston A.B. – New York: Academic Press, 1981. - 453 p.
51. Cooper G. F. The computational complexity of probabilistic inference using Bayesian belief networks / Cooper G.F. // Artificial Intelligence, 1990. – vol. 42. – No. 2-3. – pp. 393-405.
52. Dagum P. Approximating probabilistic inference in Bayesian belief networks is NP-hard / Dagum P., Luby M. // Artificial Intelligence . – 1993. – vol. 45. – pp. 141–153.
53. Enders W. Applied econometric time series / Enders W. – New York: John Wiley & Sons, Inc., 1995. – 434 p.
54. Holsapple C.W. Decision Support Systems (a knowledge based approach) / Holsapple C.W., Whinston A.B. – New York: West Publishing Company, 2003. – 860 p.
55. Keen P. G. W. Decision support systems : an organizational perspective / Keen P.G.W., Scott Morton M. S. – Reading, Mass.: Addison-Wesley Pub. Co., 1978. – 620 p.
56. Kosko B. Fuzzy Engineering / Kosko B. – Prentice-Hall, New Jersey., 1997. – 549 p.
57. Larsen P. M. Industrial applications of fuzzy logic control / Larsen P.M. // International Journal of Man-Machine Studies. – 1980. – No 12. – pp. 3-10.

58. Lauritzen S. L. Local computations with probabilities on graphical structures and their application to expert systems / Lauritzen S. L., Spiegelhalter D. J. // Journal of the Royal Statistical Society, Series B. – 1988. – vol. 50. – No.2. – pp. 157–224.
59. Little J. D. C. Models and Managers: The Concept of a Decision Calculus / Little J.D.C. // Management Science. – 1970. – v. 16. – N 8. – pp. 67 - 73.
60. Mamdani E. H. An experiment in linguistic synthesis with a fuzzy logic controller / Mamdani E. H., Assilian S. // International Journal of Man-Machine Studies. – 1975. – No 7. – pp. 1–13.
61. Nunnally J. M. Psychometric Theory, 2nd ed. / Nunnally J. M. – McGraw-Hill, New York., 1978. – 515 p.
62. Pearl J. Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference / Pearl J. – San Mateo, CA (USA): Morgan Kauffmann Publishers, Inc., 1988. – 550 p.
63. Power D. J. A Brief History of Decision Support Systems [Electron resource]/ Power D.J. – DSSResources.COM, World Wide Web, version 2.8, May 31, 2003. – Mode of access: <http://dssresources.com/history/dsshhistory.html>
64. Scott Morton M. S. Management Decision Systems: Computer-based Support for Decision Making / Scott Morton M. S. — Boston: Harvard University, 1971. – 515 p.
65. Simon H.A. The New Science of Management Decision / Simon H.A. – New York: Harper and Row. 1960. – 516 p.
66. Sprague R. H. Building Effective Decision Support Systems / Sprague R. H., Carlson E. D. — Englewood Cliffs, NJ: Prentice-Hall, 1982. - 613 p.

67. Sugeno M. An introductory survey of fuzzy control / Sugeno M. // Information Sciences. – 1985.– No 36. – pp. 59–83.
68. Tzu-Ping Wu. A new method for constructing membership functions and fuzzy rules from training examples / Tzu-Ping Wu, Shyi-Ming Chen // IEEE Transactions on Systems, Man and Cybernetics – Part B: Cybernetics. – 1999. – vol. 29. – No 1, February. – pp. 25–39.
69. Yager Ronald R. Fuzzy Set and Possibility Theory. Recent Developments / Yager Ronald R. Edited by Ronald R. Yager. – N.Y.:Pergamon Press, 1982. – 420 p.
70. Zadeh L. A. Fuzzy Sets / Zadeh L. A. // Information and Control. – 1965. – 8. – pp. 338–353.
71. Zadeh L. A. Toward a theory of fuzzy information granulation and its centrality in human reasoning and fuzzy logic / Zadeh L. A. // Fuzzy Sets and Systems. – 1997. – 9. – pp. 111–127.

Додаток А Перелік технічної документації на розроблення інформаційних систем

1. Технічні пропозиції
2. Технічне завдання
3. Технічний проект
4. Експлуатаційна документація:
 - Відомість експлуатаційних документів
 - Інструкція для експлуатації комплексу технічних засобів
 - Посібник користувача
 - Формуляр
5. Організаційно-розпорядча документація

Додаток Б Технічна пропозиція

1. У даному документі:

- повинна бути описана оцінка можливості реалізації Вимог замовника;
- повинні бути розглянуті можливі варіанти реалізації Вимог;
- виробляється оцінка необхідних ресурсів на реалізацію Вимог і забезпечення функціонування системи;
- виробляється оцінка переваг і недоліків кожного варіанта;
- повинні бути співставлені вимоги замовника й характеристики запропонованого рішення, запропоновано оптимальний варіант;
- може бути запропоновано порядок оцінки якості й умов приймання системи.

2. Документ містить наступні розділи:

- I. Введення.
- II. Постановка завдання й запропонований варіант її рішення.
- III. Опис технічних і програмних засобів.
- IV. Інші варіанти реалізації вимог замовника. Порівняльний аналіз.
- V. Схеми й малюнки.
- VI. Специфікації запропонованого комплексу технічних і програмних засобів.
- VII. Інформаційні матеріали про фірму-розроблювача й фірми-виконавця.
- VIII. Рекламні матеріали.

Додаток В Технічне завдання

Основні положення розробки технічного завдання (ТЗ):

– ТЗ є основним документом, що визначає вимоги й порядок створення системи, відповідно до якого проводиться розробка системи і її прийняття при впровадженні в дію.

– Допускається розробка ТЗ на окремі частини системи.

– Спочатку допускається розробка проекту Технічного завдання. Проект ТЗ розробляє організація-розроблювач системи за участю замовника на підставі Вимог замовника (Форма1). Зауваження по проекту ТЗ повинні бути представлені з технічним обґрунтуванням. Рішення по зауваженнях повинні бути прийняті розроблювачем проекту ТЗ і замовником до затвердження ТЗ.

– Затвердження ТЗ здійснюють керівники організацій розроблювача й замовника.

– Зміни до ТЗ оформляють доповненням або підписаним замовником і розроблювачем протоколом (Форма 4).

– Номера сторінок ТЗ проставляють, починаючи з першого аркуша, що впливає за титульним, у верхній частині аркуша посередині.

– Якщо конкретні значення показників, норм і вимог не можуть бути встановлені в процесі розробки ТЗ, у ньому варто зробити запис про порядок встановлення й узгодження цих показників: «Остаточна вимога (значення) уточнюється в процесі ... і погоджується протоколом з ... на стадії ...». При цьому в текст ТЗ змін не вносять.

– Форма титульного аркуша ТЗ наведена в Додатку 4-1, а останнього аркуша ТЗ - у Додатку 4-2.

– Титульний аркуш доповнення до ТЗ оформляють аналогічно титульному аркушу ТЗ. Замість найменування «Технічне завдання» пишуть «Доповнення № ___ до ТЗ ___».

– Розділи, які повинне містити ТЗ:

1. Загальні відомості.

У даному розділі вказуються:

- повне найменування системи і її умовні позначки;
- номер договору (якщо є);
- найменування підприємств розроблювача й замовника і їхні реквізити;
- перелік документів, на підставі яких створюється система, ким і коли затверджені ці документи;
- планові строки початку й закінчення робіт;
- відомості про джерела й порядок фінансування;
- порядок оформлення й пред'явлення замовникові результатів робіт із створення системи.

2. Призначення й мети створення системи.

У довільній формі.

3. Характеристики об'єкта.

У цьому розділі наводяться:

- короткі відомості про об'єкт або посилання на документи, що містять таку інформацію;
- відомості про умови експлуатації системи.

4. Вимоги до системи.

Складаються з підрозділів:

4.1. Вимоги до системи в цілому:

- вимоги до структури та функціонування системи;
- вимоги до чисельності й кваліфікації персоналу системи;

- показники призначення (наводяться значення параметрів, що характеризують ступінь відповідності системи її призначенню);
- вимоги до надійності;
- вимоги безпеки;
- вимоги до експлуатації, технічному обслуговуванню, ремонту й зберіганню компонентів системи;
- вимоги до захисту інформації від несанкціонованого доступу;
- вимоги по збереженню інформації при аваріях (наводиться перелік подій, при яких повинна бути забезпечена схоронність інформації в системі);
- вимоги до захисту від впливу зовнішніх впливів;
- вимоги по стандартизації й уніфікації;
- додаткові вимоги.

4.2. Вимоги до функцій, які виконуватиме система:

- перелік функціональних підсистем, що вводять у дію в 1-у і наступних чергах;
- часовий регламент реалізації кожної функції;
- вимоги до якості реалізації кожної функції.

4.3. Вимоги до видів забезпечення

- вимоги до математичного, інформаційного, програмного, технічного й іншого видів забезпечення системи.

5. Склад і зміст робіт із створення (розвитку) системи.

Даний розділ містить:

- перелік стадій і етапів робіт із створення системи, строки їхнього виконання,
- перелік організацій-виконавців робіт, посилання на документи, що підтверджують згоду цих організацій на участь у створенні системи, або запис, що визначає відповідального за проведення цих робіт;

- перелік документів, пропонованих по закінченні відповідних стадій і етапів робіт.

6. Порядок контролю й приймання системи.

Вказуються:

- види, склад, обсяг і методи випробувань системи і її складових частин;

- загальні вимоги до приймання робіт (перелік підприємств і організацій, що беруть участь, місце й строки проведення), порядок узгодження й затвердження приймальної документації;

- статус приймальної комісії (державна, міжвідомча, відомча).

7. Вимоги до складу й змісту робіт по підготовці об'єкта до введення системи в дію.

Наводиться перелік основних заходів:

- зміни, які необхідно здійснити в об'єкті;

- створення умов функціонування об'єкта, при яких гарантується відповідність створюваної системи вимогам ТЗ;

- створення необхідних для функціонування системи підрозділів і служб;

- строки й порядок комплектування штатів і навчання персоналу.

8. Вимоги до документування.

У цьому розділі наводять:

- погоджений розроблювачем і замовником перелік робіт, які підлягають розробці комплектів і видів документів; перелік документів, що створюють на машинних носіях;

- при відсутності держ.стандартів, що визначають вимоги до документування елементів системи, додатково включають вимоги до складу й змісту таких документів.

9. Додатки:

- розрахунок очікуваної ефективності системи;
- оцінка науково-технічного рівня системи;
- інше.

В.1 Форма титульного аркуша технічного завдання на систему

АТ "PROFYSOFT "

ЗАТВЕРДЖУЮ

(ПІБ)

(посада)

(найменування підприємства-замовника)

(підпис)

« _____ » _____ 20__ р.

(Печатка)

ЗАТВЕРДЖУЮ

_____ І.І. Іваненко

(ПІБ)

_____ Президент

(посада)

_____ АТ "Profysoft"

(найменування підприємства-виконавця)

(підпис)

« _____ » _____ 20__ р.

(Печатка)

найменування

ТЕХНІЧНЕ ЗАВДАННЯ

На _____ аркушах

КИЇВ – 20__ р.

В.2 Форма останнього аркуша технічного завдання

СКЛАЛИ

Найменування організації, підприємства	Посада виконавця	Прізвище, ім'я, по батькові	Підпис	Дата

ПОГОДЖЕНО

Найменування організації, підприємства	Посада	Прізвище, ім'я, по батькові	Підпис	Дата

Додаток Г Технічний проект

1. Загальні положення.

– Процес підготовки документації включає підготовку ескізного проекту, технічного проекту й робочої документації. Але існуючі стандарти допускають об'єднання цих документів в один (ДСТУ 34.601-90).

– У загальному випадку, Технічний проект складається з наступних розділів (РД 50-34.698-90):

- загальносистемні рішення;
- рішення по організаційному забезпеченню;
- рішення по технічному забезпеченню;
- рішення по інформаційному забезпеченню;
- рішення по програмному забезпеченню;
- рішення по математичному забезпеченню.

– Перелік найменувань розроблювальних документів і їхня комплектність повинні бути визначені в Технічному завданні.

– Після назв документів у даному Додатку в дужках вказується їхній код відповідно до ДСТУ 34.201-89. Цей код використовується при формуванні позначення документів. Кожний з розділів Технічного проекту, що має самостійне позначення, може бути виділений в окремий документ.

2. Зміст розділів Технічного проекту.

2.1. Пояснювальна записка містить розділи:

2.1.1. Загальні положення:

– найменування проектованої системи й найменування документів, їхнього номера й дату завершення, на підставі яких ведуть проектування системи;

- перелік організацій, що беруть участь у розробці системи, строки виконання стадій;

- мета, призначення й області використання системи;

- відомості про передовий досвід, винаходи, використані при розробці проекту;

- черговість створення системи й обсяг кожної черги;

2.1.2. Опис процесу діяльності:

- як буде забезпечений взаємозв'язок і сумісність процесів автоматизованої й неавтоматизованої діяльності, вимоги до організації робіт в умовах функціонування системи;

2.1.3. Опис постановки завдання (комплексу завдань):

- характеристики комплексу завдань:

- вихідна інформація;

- вхідна інформація;

2.1.4. Основні технічні рішення:

- загальносистемні рішення:

- рішення за структурою системи, підсистем, засобам і способам зв'язку для інформаційного обміну між компонентами системи, підсистем;

- рішення по взаємозв'язках системи із суміжними системами, забезпеченню її сумісності;

- рішення по режимах функціонування, діагностуванню роботи системи;

- рішення стосовно організаційного забезпечення:

- опис організаційної структури (зміни в організаційній структурі керування об'єктом і у взаємозв'язках між підрозділами);

- опис технологічного процесу обробки даних (включаючи телеобробку);

- рішення по технічному забезпеченню:

- опис комплексу технічних засобів:
 - загальні положення;
 - структура комплексу технічних засобів;
 - апаратури передачі даних;
 - план розташування технічних засобів, що вимагають спеціальних приміщень або окремих площ для розміщення;
- рішення по інформаційному забезпеченню: склад інформації, обсяг, способи її організації, види машинних носіїв, вхідні й вихідні документи і повідомлення, послідовність обробки інформації та інші компоненти;
- рішення по програмному забезпеченню:
 - структура програмного забезпечення (перелік частин ПЗ із вказівкою їхніх взаємозв'язків і обґрунтуванням виділення кожної з них);
 - функції частин програмного забезпечення;
 - методи й засоби розробки програмного забезпечення;
 - операційна система;
 - засоби, що розширюють можливості операційної системи;
 - рішення по алгоритмах і методах їх реалізації;

2.1.5. Заходи щодо підготовки об'єкта до введення системи в дію:

- заходи щодо приведення інформації до вигляду, придатного для обробки на ЕОМ;
- заходи щодо навчання і перевірки кваліфікації персоналу;
- заходи щодо створення необхідних підрозділів і робочих місць;
- інші заходи, що виходять із специфічних особливостей створюваної системи.

2.2. Схема організаційної структури:

–склад підрозділів (посадових осіб) організації, що забезпечують функціонування системи або використовують при ухваленні рішення інформацію, отриману від системи;

–основні функції і зв'язки між підрозділами та посадовими особами, зазначеними на схемі, їхня підпорядкованість.

2.3. Схема функціональної структури:

–елементи функціональної структури системи; автоматизовані функції й завдання; сукупності дій (операцій), виконуваних при реалізації автоматизованих функцій тільки технічними засобами (автоматично) або тільки людиною;

–інформаційні зв'язки між елементами та із зовнішнім середовищем з короткою вказівкою змісту повідомлень і/або сигналів, переданих по лініях зв'язку, і при необхідності, зв'язку інших типів;

–деталізовані схеми частин функціональної структури.

2.4. Схема структурна комплексу технічних засобів.

Документ містить склад комплексу тех.засобів і зв'язку між цими технічними засобами або групами технічних засобів, об'єднаних за якими-небудь логічними ознаками.

2.5. Локальний кошторисний розрахунок:

Містить відомості про кошторисну вартість робіт, виконуваних при створенні системи, і кошторисної вартості об'єктів, що споруджують при створенні системи.

2.6. Програма і методика випробувань.

Призначена для встановлення даних, що забезпечують одержання й перевірку проектних рішень, виявлення причин відмов, визначення якості робіт, показників якості функціонування системи, перевірку відповідності системи вимогам ТБ, тривалість і режим випробувань.

Перелік перевірок, які варто здійснювати при випробуваннях для підтвердження виконання вимог ТЗ, з посиланнями на відповідні методики випробувань включає такі елементи:

- відповідність системи ТЗ;
- комплектність системи;
- комплектність і якість документації;
- комплектність, достатність складу і якість програмних засобів і програмної документації;
- кількість і кваліфікацію обслуговуючого персоналу;
- ступінь виконання вимог функціонального призначення системи;
- придатність системи для контролю;
- виконання вимог ТБ, протипожежній безпеці, промисловій санітарії, ергономіки;
- функціонування системи із застосуванням програмних засобів.

Г.1 Форма титульного аркуша технічного проекту

АТ "PROFYSOFT"

ЗАТВЕРДЖУЮ

_____ (ПІБ)

_____ (посада)

_____ (найменування підприємства-замовника)

_____ (підпис)

« _____ » _____ 20__ р.

(Печатка)

ЗАТВЕРДЖУЮ

_____ І.І. Іваненко
(ПІБ)

_____ Президент
(посада)

_____ АТ "Profysoft"
(найменування підприємства-виконавця)

_____ (підпис)

« _____ » _____ 20__ р.

(Печатка)

ТЕХНІЧНИЙ ПРОЕКТ

_____ (назва розроблюваної системи)

На _____ аркушах

ПОГОДЖЕНО

Керівник (посада,
найменування організації,
яка погоджує)

Особистий підпис

Печатка
Дата

Розшифровка
підпису

Г.2 Форма останнього аркуша технічного проекту

СКЛАЛИ

Найменування організації, підприємства	Посада виконавця	Прізвище, ім'я, по батькові	Підпис	Дата

ПОГОДЖЕНО

Найменування організації, підприємства	Посада	Прізвище, ім'я, по батькові	Підпис	Дата

Додаток Д Експлуатаційна документація

Складається з таких документів:

- Відомість експлуатаційних документів;
- Інструкція для експлуатації комплексу технічних засобів;
- Посібник користувача;
- Формуляр.

Відомість експлуатаційних документів

Містить перелік експлуатаційних документів і їхнього позначення. У нашому випадку експлуатаційними документами є:

- Інструкція для експлуатації комплексу технічних засобів;
- Посібник користувача;
- Формуляр.

Д.1 Інструкція стосовно експлуатації комплексу технічних засобів

Інструкція містить розділи:

1. Загальні вказівки

- вид устаткування, для якого складена дана інструкція;
- найменування функцій системи, реалізованих на даному устаткуванні;
- регламент і режими роботи устаткування стосовно реалізації функцій;
- перелік експлуатаційних документів, якими повинен додатково керуватися персонал при експлуатації даного устаткування.

2. Міри безпеки

- правила безпеки, яких необхідно дотримуватись під час підготовки устаткування до роботи та при його експлуатації.

3. Порядок роботи

- склад і кваліфікація персоналу, що допускається до експлуатації устаткування;
- порядок перевірки знань персоналу і допуску його до роботи;
- опис робіт і послідовність їх виконання.

4. Перевірка правильності функціонування

- зміст і короткі методики основних перевірок функціональності устаткування і правильності виконання функцій системи.

5. Вказівки про дії в різних режимах

- дії персоналу в нормальному режимі роботи, аварійному відключенні устаткування, передаварійному і аварійному стані об'єкта автоматизації, пусковому та зупинному режимах функціонування об'єкта.

Д.2 Посібник користувача

Містить розділи:

1. Введення

- область застосування;
- короткий опис можливостей;
- рівень підготовки користувачів;
- перелік документації, з якою необхідно ознайомитися користувачеві.

2. Призначення й умови застосування

- види діяльності, функції, для яких призначена розроблена система;
- умови, при дотриманні яких забезпечується застосування розробленої системи відповідно до призначення (наприклад, конфігурація технічних засобів, операційне середовище й програмні засоби, вхідна інформація, носії даних, бази даних, вимоги до підготовки фахівців тощо.).

3. Підготовка до роботи

- склад і зміст дистрибутивного носія інформації;
- порядок завантаження даних і програм;
- порядок перевірки функціональності системи.

4. Опис операцій

- опис всіх виконуваних функцій, завдань, процедур;
- опис операцій технологічного процесу обробки даних, необхідних для виконання функцій, завдань, процедур. Для кожної операції обробки даних вказують найменування, умови, при яких можливе виконання операції, підготовчі дії, основні дії в необхідній послідовності, заключні дії, ресурси, що витрачають на операцію. Наводиться перелік посад

персоналу, на які поширюються дані правила виконання операцій технологічного процесу обробки даних.

5. Аварійні ситуації

– дії у випадку недотримання умов виконання технологічного процесу, у тому числі при тривалих відмовах технічних засобів;

– дії по відновленню програм/даних;

– дії у випадку виявлення несанкціонованого втручання в дані;

– дії в інших аварійних ситуаціях.

6. Рекомендації з освоєння

– рекомендації стосовно освоєння та експлуатації системи, включаючи опис контрольного прикладу, правила його запуску і виконання.

Д.3 Формуляр

Зберігається у замовника і ним же заповнюється у процесі експлуатації системи. У ньому фіксуються всі несправності і аварійні ситуації, а також вжиті при цьому заходи.

Містить розділи:

1. Загальні відомості

- найменування системи і її позначення;
- найменування розробника;
- дата здачі системи в експлуатацію;
- загальні вказівки персоналу по експлуатації системи;
- вимоги стосовно ведення формуляра й місця його зберігання;
- перелік технічної документації, з якою повинен бути ознайомлений персонал.

2. Основні характеристики

- перелік реалізованих функцій;
- кількісні і якісні характеристики системи і її частин;
- опис принципів функціонування системи, регламент і режими функціонування;
- відомості про взаємодію системи з іншими системами.

3. Комплектність

- перелік технічних і програмних засобів;
- перелік експлуатаційних документів.

4. Свідчення про приймання

- дати підписання актів приймання системи і її частин у промислову експлуатацію;
- прізвища голів комісій, що здійснюють приймання системи.

5. Гарантійні зобов'язання

- гарантійні зобов'язання розроблювачів по системі в цілому й частинам, що мають різні гарантійні строки;
- перелік технічних засобів системи, що мають гарантійні терміни служби менше гарантійних строків для системи.

6. Відомості про стан системи

- відомості про несправності, у тому числі дата, час, характер, причина виникнення й особи, що усунула несправність;
- зауваження по експлуатації та аварійних ситуаціях, вжиті при цьому заходи;
- відомості про ремонт технічних засобів і зміни у програмному забезпеченні із вказівкою підстави, дати, а також зміст зміни;
- відомості про виконання регламентних (профілактичних) робіт і їхніх результатів.

7. Відомості про рекламації

- номер, дата й короткий зміст рекламаційного акту;
- відомості про усунення зауважень, зазначених в акті.

Додаток Е Організаційно-розпорядча документація

На етапі впровадження в дію розробленої системи готуються такі документи:

1. Наказ про склад приймальної комісії:

- найменування прийнятої системи в цілому або її частин;
- відомості про склад комісії;
- підстава для організації комісії;
- найменування організації-замовника;
- найменування організації-розроблювача, організацій-співвиконавців;
- призначення і мета роботи комісії;
- строки початку і завершення роботи комісії;
- вказівка про форму завершення роботи комісії.

2. План-графік робіт

2.1. Документ встановлює перелік робіт, строки виконання й виконавців робіт, пов'язаних із впровадженням у дію системи.

2.2. Для кожної роботи вказується:

- найменування роботи;
- дата початку й закінчення роботи;
- найменування підрозділу-учасника роботи;
- прізвище й посада відповідального виконавця;
- форма подання результатів роботи.

3. Накази про проведення робіт:

- наказ про готовність об'єкта до проведення будівельно-монтажних робіт;
- наказ про готовність об'єкта до проведення налагоджувальних робіт;
- наказ про початок дослідної експлуатації системи;

– наказ про введення в промислову експлуатацію системи.

4. Протокол випробувань:

– найменування об'єкта випробувань;

– список посадових осіб, що проводила випробування;

– ціль випробувань;

– відомості про тривалість випробувань;

– перелік пунктів ТЗ на створення системи, на відповідність яким виконано випробування;

– відомості про результати спостережень за правильністю функціонування системи;

– відомості про відмови, збої й аварійні ситуації, що виникають при випробуваннях;

– відомості про коректування параметрів об'єкта випробування.

5. Акт завершення робіт.

Документ містить:

– найменування завершеної роботи;

– список представників організації-розроблювача й організації-замовника, що склали акт;

– дату завершення робіт;

– найменування документа(ів), на підставі якого(их) проводилася робота;

– основні результати завершеної роботи;

– висновок про результати завершеної роботи.

6. Акт приймання в дослідну експлуатацію

Документ містить:

– найменування системи (або її частини), прийнятої в дослідну експлуатацію, і відповідного об'єкта автоматизації;

- найменування документа, на підставі якого розроблена система;
- склад приймальної комісії й підстава для її роботи;
- період часу роботи комісії;
- найменування організації-розроблювача, організації-співвиконавця та організації-замовника;
- склад функцій системи, прийнятої в дослідну експлуатацію;
- перелік складових технічних, програмних, інформаційних і організаційних забезпечень, що перевіряють у процесі дослідної експлуатації;
- перелік документів, запропонованих комісії;
- оцінку відповідності прийнятої системи технічному завданню на її створення;
- основні результати прийняття в дослідну експлуатацію;
- рішення комісії про прийняття системи в дослідну експлуатацію.

7. Акт приймання системи у промислову експлуатацію

7.1. Документ містить:

- найменування об'єкта автоматизації і системи, прийнятої в промислову експлуатацію;
- відомості про статус приймальної комісії (державна, міжвідомча, відомча), її складі й підстава для роботи;
- період часу роботи комісії;
- найменування організації-розроблювача, організацію-співвиконавця і організацію-замовника;
- найменування документа, на підставі якого розроблена система;
- склад функцій системи, прийнятої в промислову експлуатацію;

- перелік складових технічних, програмних, інформаційних і організаційних забезпечень, прийнятих у промислову експлуатацію;
- перелік документів, запропонованих комісії;
- висновок про результати дослідної експлуатації;
- оцінку відповідності прийнятої системи технічному завданню на її створення;
- коротку характеристику й основні результати виконаної роботи із створення системи;
- оцінку науково-технічного рівня системи;
- оцінку економічної ефективності від впровадження системи;
- рішення комісії;
- рекомендації комісії з подальшого розвитку системи.

7.2. Додаються протоколи випробувань, протоколи засідання комісії, акти приймання у промислову експлуатацію прийнятих раніше частин системи, перелік технічних засобів, які використала комісія при прийманні системи тощо.

Е.1 Оформлення та зберігання документів

1. Всі текстові документи готуються членами робочої групи у форматі Microsoft Word. Назви устаткування або програмних продуктів, описуваних у документі, з першою появою в тексті виділяються шрифтом **Bold**. Перше згадування яких-небудь проектних рішень, на які варто звернути увагу, супроводжується виділенням шрифтом *Italic*.

Дані установки можуть бути змінені Менеджером проекту.

2. При написанні Технічних пропозицій необхідно пам'ятати, що цей документ повинен зацікавити Замовника і бажано, щоб цей документ був прочитаний ним до кінця. Тому допускається деяка оформлювальна вільність.

Текст для титульної сторінки готується в довільній формі і повинен містити назву організації-розроблювача, назва документа, назва організації-замовника, рік розробки документа.

3. Малюнки й схеми виконуються за допомогою *Microsoft Word* або *Visio*.

4. Специфікації виконуються у форматі *Microsoft Excel*.

6. Для здійснення поточної роботи над проектом і зберігання розроблених документів Менеджером проекту при початку робіт створюється каталог *c:\...\Projects\{name_of_organization}*. Назви документів, що зберігаються в цьому каталозі, повинні бути очевидними і відповідати змісту документа. Всі розроблювальні ними документи члени робочої групи передають Менеджерові проекту у електронному виді.

Е.2 Заявка на виконання робіт

Форма 1

<i>Менеджер</i>	<i>Дата</i>
Замовник:	
Організація -	
Представник -	
Телефон, факс -	
Що зробити:	
Суть робіт:	
<i>Технічна пропозиція</i>	
<i>Специфікація</i>	
<i>Роботи</i>	
<i>Консультація</i>	
<i>Скласти контракт</i>	
<i>Розробити Технічний проект</i>	
<i>Інше (указати)</i>	
Строки виконання заявки:	
Прикладені документи:	
Умови оплати робіт:	
<i>Передплата</i>	
<i>за фактом виконання</i>	
<i>включено у вартість обладнання</i>	
<i>інше (вказати)</i>	
Додаткові відомості:	
Підпис менеджера	
Підпис директора ОПП	
Рішення директора ДИС:	
<i>Виконати/Не виконати</i>	
<i>Виконавці:</i>	
<i>Особливі вказівки:</i>	
Підпис	Дата