

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ
«ХАРКІВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

Людмила НІКІТІНА

ЕКСПЕРТНІ СИСТЕМИ

Навчальний посібник
для студентів спеціальності 122 – Комп'ютерні науки

Харків – 2023

УДК 004.891

Н 62

Рекомендовано до публікації рішенням кафедри систем інформації ім. В.О.Кравця
Національного технічного університету "Харківський Політехнічний Інститут"
(протокол № 9 від 21.04.2023 р.).

Рецензенти:

Сергій ЛЕОНОВ, доктор технічних наук, професор, професор кафедри
обчислювальної техніки та програмування Національного технічного університету
«Харківський Політехнічний Інститут»,

Олександр ЯНКОВСЬКИЙ, кандидат технічних наук, доцент, доцент кафедри
електронних обчислювальних машин Харківського національного університету
радіоелектроніки.

Людмила Нікітіна

Н 62 Експертні системи: навчальний посібник / Л.Нікітіна – Харків: НТУ «ХПІ», 2023.
– 210 с., рис. 74, табл. 20.

ISBN 000-000-000-000-0

У посібнику розглядаються основні поняття та перспективи розвитку прикладних систем штучного інтелекту в контексті їх використання в технологіях експертних систем. Проаналізовано методи подання й обробки знань, особливості побудови експертних систем, загальні відомості про методи набуття знань та інструментарії для розробки експертних систем. Здійснено теоретичний аналіз основних принципів та методів подання знань та їх моделей в експертних системах.

Матеріал навчального посібника рекомендовано для студентів спеціальності 122 - Комп'ютерні науки.

УДК 004.891

Н 62

ISBN 000-000-000-000-0

© Нікітіна Л. 2023 р.

© НТУ «ХПІ», 2023

ЗМІСТ

1 ВСТУП ДО ЕКСПЕРТНИХ СИСТЕМ	5
1.1 Поняття «експертна система»	5
1.2 Класифікація експертних систем	9
1.2.1 Класифікація за вирішуваною задачею	10
1.2.2 Класифікація за зв'язком з реальним часом	13
1.2.3 Класифікація за типом комп'ютера	13
1.2.4 Класифікація за ступенем інтеграції з іншими програмами	13
2 СТРУКТУРИ ТА ОСНОВНІ КОМПОНЕНТИ ЕКСПЕРТНИХ СИСТЕМ	15
2.1 Технологія експертних систем	15
2.2 Переваги ЕС	19
2.3 Основні концепції експертних систем	21
2.4 Структура та принцип роботи експертних систем	23
3 РОЗРОБКА ЕКСПЕРТНИХ СИСТЕМ	29
3.1 Процес розробки експертних систем	29
3.2 Тип проблемного середовища	36
3.3 Класифікація засобів розробки ЕС	40
3.3.1 Рівень мови, що використовується	40
3.3.2 Парадигми програмування (механізми реалізації виконуваних тверджень)	42
3.3.3 Спосіб представлення знань	43
3.3.4 Механізми виведення та система моделювання	43
3.3.5 Засоби набуття знань	45
3.4 Підходи до розробки ЕС	46
4 БАЗИ ЗНАНЬ ЕКСПЕРТНИХ СИСТЕМ	50
4.1 Мова опису поля знань	50
4.2 Семіотична модель поля знань	51
4.3 Стратегії отримання знань	55
4.4 Теоретичні аспекти вилучення знань	57
4.4.1 Психологічний аспект вилучення знань	57
4.4.2 Лінгвістичний аспект вилучення знань	61
4.4.3 Гносеологічний аспект вилучення знань	63
4.5 Теоретичні аспекти структурування знань	67
4.5.1 Ієрархічний підхід	67
4.5.2 Традиційні методології структурування	68
4.6 Моделі представлення знань	72
4.6.1 Продукційна модель представлення знань	73
4.6.2 Модель семантичних мереж	84
4.6.3 Фреймова модель представлення знань	95
4.6.4 Логічна модель представлення знань	108
5 РОБОТА З НЕЧІТКИМИ ЗНАННЯМИ	114
5.1 Види невизначеності	114
5.2 Коефіцієнти упевненості	115
5.3 Системи з дошкою оголошень	120
5.4 Ймовірісна логіка у експертних системах	125

5.4.1 Основні поняття теорії ймовірностей.....	127
5.4.3 Ймовірність Байєса.....	128
5.4.4 Таблиці ймовірностей.....	132
5.4.5 Подання і обробка нечітких знань: підхід на основі умовних ймовірностей	134
5.4.6 Байєсівський підхід.....	136
5.4.7 Байєсові мережі довіри.....	138
5.4.8 Застосування байєсових мереж.....	141
5.5 Теорія свідощів Демпстера-Шефера.....	141
5.6 Нечіткі множини і нечітка логіка	143
5.6.1 Нечіткі множини.....	144
5.6.2 Нечіткі і та лінгвістичні змінні	145
5.6.3 Операції над нечіткими множинами.....	148
5.6.4 Нечітке логічне виведення	153
5.6.4.2 Алгоритми нечіткого логічного виведення	155
6 МЕТОДИ ВИЛУЧЕННЯ ЗНАНЬ	162
6.1 Комунікативні методи.....	162
6.1.1 Пасивні методи	162
6.1.2 Активні індивідуальні методи.....	166
6.2 Текстологічні методи	177
6.3 Методи структурування	181
6.3.1 Спеціальні методи структурування знань	181
6.4 Методи експертних оцінок	184
6.4.1 Експертне оцінювання.....	185
6.4.2 Етапи експертного оцінювання.....	185
6.4.3 Характеристики методів експертних оцінок	187
6.4.4 Обробка експертних оцінок	189
7 НЕЙРОМЕРЕЖЕВА ТЕХНОЛОГІЯ У ЕКСПЕРТНИХ СИСТЕМАХ	192
7.1 Біологічна нейронна мережа.....	192
7.2. Штучна нейронна мережа	194
7.3 Робота з нейронною мережею	195
7.4 Деякі області застосування нейронних мереж.....	197
7.5 Проблеми нейромереж	199
7.6 Нейромережеві експертні системи	200
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	207

1 ВСТУП ДО ЕКСПЕРТНИХ СИСТЕМ

На початку вісімдесятих років дослідження штучного інтелекту утворили незалежний напрямок під назвою "експертні системи". Метою досліджень у галузі експертних систем є розробка комп'ютерних програм, які вирішують складні для експерта-людини проблеми, та отримують результати, що не поступаються за якістю та ефективністю рішенням експерта. Дослідники в галузі експертних систем часто також використовують термін "інженерія знань", введений Е.Фейгенбаумом як "застосування принципів та інструментів досліджень з галузі штучного інтелекту для вирішення складних прикладних проблем, що вимагають знань експертів".

Експертні системи (ЕС) відрізняються від систем обробки даних тим, що в основному використовують символний (а не числовий) спосіб подання даних, символне виведення і евристичний пошук рішення (а не виконання відомого алгоритму).

Рішення експертних систем наділені "прозорістю", тобто, вони можуть бути пояснені користувачеві на якісному рівні. Ця властивість експертних систем забезпечується їхньою здатністю "розмірковувати" на основі наявних знань. Експертні системи оперують знаннями та здатні поповнювати свої знання під час взаємодії з експертом.

Технологія експертних систем використовується для розробки програмних застосунків, орієнтованих на вирішення різних типів задач, таких як інтерпретація, передбачення, діагностика, планування, конструювання, контроль, налагодження, інструктаж, управління та ін.

ЕС застосовуються в різноманітних проблемних галузях, таких як медицина, фінанси, нафтова та газова промисловість, енергетика, транспорт, фармацевтичне виробництво, космос, металургія, гірнича справа, хімія, освіта, телекомунікації та зв'язок та ін.

Таким чином, експертна система є комп'ютерною системою, яка імітує (емулює) дії людини-експерта у процесі пошуку та прийняття рішень. Термін «імітує» означає, що ЕС має діяти в усіх відношеннях як людина-експерт.

1.1 Поняття «експертна система»

Першим кроком у вирішенні будь-якої проблеми є визначення галузі вирішуваної проблеми. Такий підхід доцільно використовувати для розробки штучного інтелекту (ШІ) у межах програмно-прагматичного напрямку. Наразі безліч реальних проблем з навколишнього світу вирішуються за допомогою ШІ, і існує безліч комерційних додатків ШІ.

Хоча в даний час загальні досконалі рішення класичних проблем ШІ, таких як переклад з однієї природної мови на іншу, розуміння мови, розпізнавання зорових образів та ін. ще не знайдені, за умови обмеження

предметної області можна знаходити корисні рішення. Наприклад, нескладно побудувати природномовну систему, якщо вхідні речення обмежені іменниками, дієсловами та об'єктами. Такі системи забезпечують дружній інтерфейс у різних програмних застосунках. Парсери, вбудовані в популярні комп'ютерні ігри, мають властивість розуміння природної мови на високому рівні.

Штучний інтелект має широкий спектр застосування (рис. 1.1).

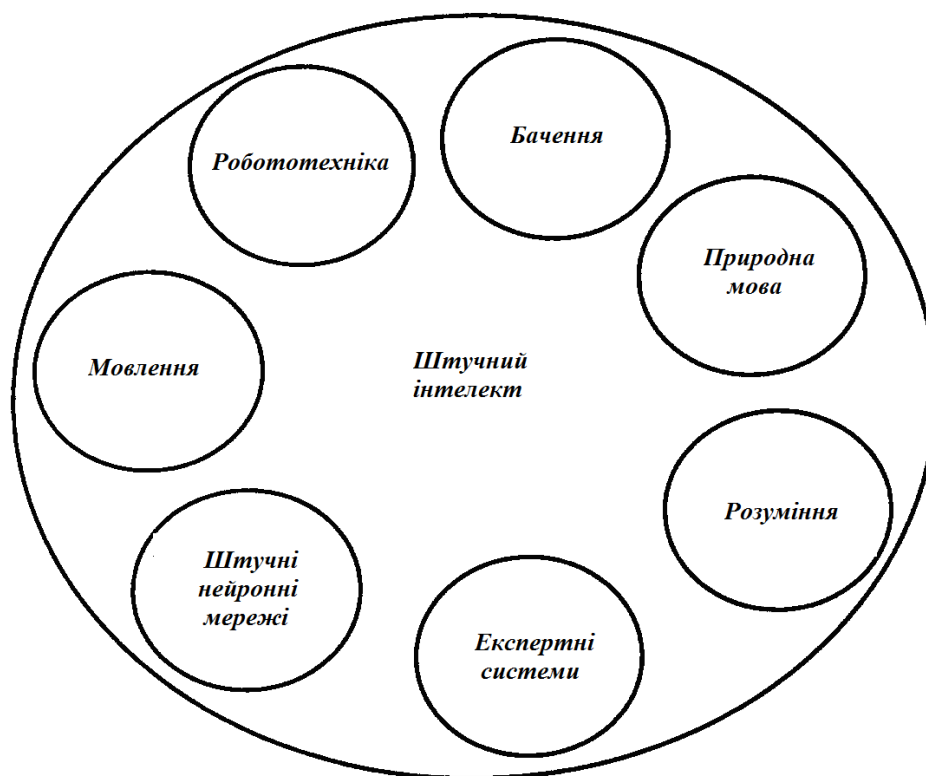


Рисунок 1.1 – Деякі сфери застосування штучного інтелекту

Експертні системи успішно вирішують класичні завдання інтелекту, що програмується. Професор Едвард Фейгенбаум зі Стенфордського університету, піонер технології експертних систем, дав визначення експертної системи як "інтелектуальної комп'ютерної програми, яка використовує знання та процедури виведення для вирішення проблем, які є досить складними, та потребують значних людських знань для їх вирішення". Тобто експертна система є комп'ютерною системою, яка імітує (емулює) здібності людини-експерта у процесі прийняття рішень. Термін «імітує» означає, що експертна система має діяти в усіх відношеннях як людина-експерт. Емуляція – це набагато більше, ніж моделювання, яке необхідно лише щоб діяти як реальний об'єкт тільки в деяких випадках.

Експертна система – це інтелектуальна система, призначена для вирішення слабо формалізованих задач, на основі накопиченого в базі знань досвіду роботи експертів в проблемній області. Вона містить базу знань з набором правил і механізмом виведення і дозволяє на підставі наданих

користувачем фактів розпізнати ситуацію, поставити діагноз, сформулювати рішення або дати рекомендацію для вибору дії.

Експертні системи призначені для відтворення досвіду, знань професіоналів високого рівня і використання цих знань в процесі управління. В основі побудови експертних систем лежить база знань, яка ґрунтується на моделях представлення знань. Вважається, що будь-яка експертна система – це система, заснована на знаннях.

Експертні системи добре працюють в обмежених областях. Доказом цього є багато застосунків експертних систем у бізнесі, медицині, науці та техніці, а також безліч опублікованих книг, журналів, конференцій, присвячених експертним системам.

Дослідження в галузі штучного інтелекту спрямовані на розробку та впровадження комп'ютерних програм, здатних емулювати (імітувати, відтворювати) ті сфери діяльності людини, які вимагають мислення, певної майстерності та накопиченого досвіду.

Робота зі знаннями є центральною проблемою сучасних інтелектуальних технологій. Системи, ядром яких є база знань чи модель предметної галузі, називають інтелектуальними. Такі системи застосовуються для вирішення складних завдань з використанням слабоформалізованих знань експертів, і в них логічна, смислова обробка інформації переважає обчислювальну.

Системи, що базуються на знаннях, — це комп'ютерні програми, що відтворюють роботу експертів у певних галузях знань.

ЕС є одним з основних типів інтелектуальних систем. Вони з'явилися в рамках досліджень ШІ в той період, коли ця наука переживала серйозну кризу, і був потрібний суттєвий прорив у розвитку практичних застосунків. Так виникла ідея моделювати конкретні знання фахівців-експертів. Досі єдиним критерієм інтелектуальності програмних систем є механізми роботи зі знаннями.

Термін "система, що базується на знаннях", з'явився в 1976 році одночасно з першими системами, що накопичували досвід та знання експертів. Це були експертні системи у галузях медицини та хімії. Вони діагностували інфекційні захворювання крові та розшифрували дані мас-спектрографічного аналізу.

ЕС дозволяють формалізувати конкретні змістовні знання про об'єкти управління та процеси, що протікають в них, тобто поряд з математичними створити комп'ютерні логіко-лінгвістичні моделі. Логіко-лінгвістичне моделювання розширює сферу застосування комп'ютерів за рахунок важко- або зовсім неформалізованих раніше галузей знань (диспетчерське управління, управління гнучким автоматизованим виробництвом, управління бойовими діями тощо).

ЕС ефективні у специфічних «експертних» галузях, де важливим є досвід фахівців. Розробка ЕС спрямована на використання комп'ютерної обробки інформації в тих галузях науки і техніки, де традиційні математичні методи моделювання малопридатні.

Хоча ЕС призначені для рішення неформалізованих завдань, в них не відкидається і не замінюється традиційний підхід до розробки програм, орієнтований на рішення формалізованих завдань.

Властивості неформалізованих завдань:

- завдання не можуть бути задані у числовій формі;
- цілі не можуть бути виражені в термінах точно визначеної цільової функції;
- не існує алгоритмічного вирішення завдань;
- вхідні дані можуть бути хибними, неоднозначними, неповними та суперечливими;
- знання про проблемну сферу та розв'язуване завдання бути хибними, неоднозначними, неповними та суперечливими;
- велика розмірність простору рішення;
- дані та знання динамічно змінюються.

Системи, що базуються на знаннях, називаються експертними системами, коли їхнє використання визначається фахівцем-людиною. Існує багато систем, які базуються на знаннях експертів. Задача-максимум систем, що базуються на знаннях, полягає в тому, щоб замінити людину, яка приймає рішення, але в дійсності, ця мета перевищує реальні можливості систем. Вони можуть взяти на себе лише частину роботи людини, але не можуть цілком замінити її.

Як правило, ЕС створюються для вирішення практичних завдань в деяких вузькоспеціалізованих областях, де велику роль відіграють знання досвідчених фахівців.

Експертне знання - це поєднання теоретичного розуміння проблеми і практичних навичок її рішення, ефективність яких доведена в результаті практичної діяльності експертів у даній області.

Фундаментом експертної системи будь-якого типу є база знань, яка формується на основі експертних знань, які отримують від фахівців-експертів.

Експерт надає необхідні знання про ретельно відібрані прикладні проблеми і шляхи їх вирішення. Правильно вибраний експерт і вдала формалізація його знань дозволяє створити експертну систему з унікальними та цінними знаннями.

Цінність експертної системи як закінченого продукту в основному визначається якістю створеної бази знань.

Характеристики експертних систем:

- чітка обмеженість предметної області - експертна система обмежена певною сферою експертизи;
- здатність приймати рішення в умовах невизначеності - експертна система здатна міркувати при наявності сумнівних знань;
- здатність пояснити ланцюжок міркувань зрозумілим для користувача способом;

- можливість поступового нарощування системи на основі поповнення бази знань;
- алгоритм рішення не описується заздалегідь, а будується самою експертною системою;
- чітке розділення декларативних і процедурних знань (фактів і механізмів виведення);
- рішенням формується не у вигляді таблиць з цифрами, а як чітка порада;
- результат видається у вигляді конкретних рекомендацій для дій у ситуації, що склалася; якість рекомендацій відповідає якості рішень найкращих фахівців;
- можливість рішення неформалізованих завдань (спосіб формалізації яких поки що невідомий);
- відсутність гарантії знаходження оптимального рішення з можливістю вчитися на помилках.

Проблеми експертних систем:

- передача експертним системам глибоких знань про предметні області; як правило, це є наслідком складності формалізації евристичних знань експертів;
- здатність надати повні усвідомлені пояснення своїх міркувань, як це робить людина у вигляді послідовності кроків, зроблених у процесі пошуку рішення;
- трудомісткість налагодження і тестування при застосуванні експертних систем у таких критичних областях, як управління повітряним і залізничним рухом, системами зброї і ядерною промисловістю;
- повноцінна здатність до самонавчання;
- для підтримки експертної системи в актуальному стані необхідне постійне втручання інженерів по знаннях в базу знань.

1.2 Класифікація експертних систем

Клас ЕС сьогодні поєднує кілька тисяч різних програмних комплексів, які можна класифікувати за різними критеріями. Одна з можливих класифікацій наведена на рис. 1.2.

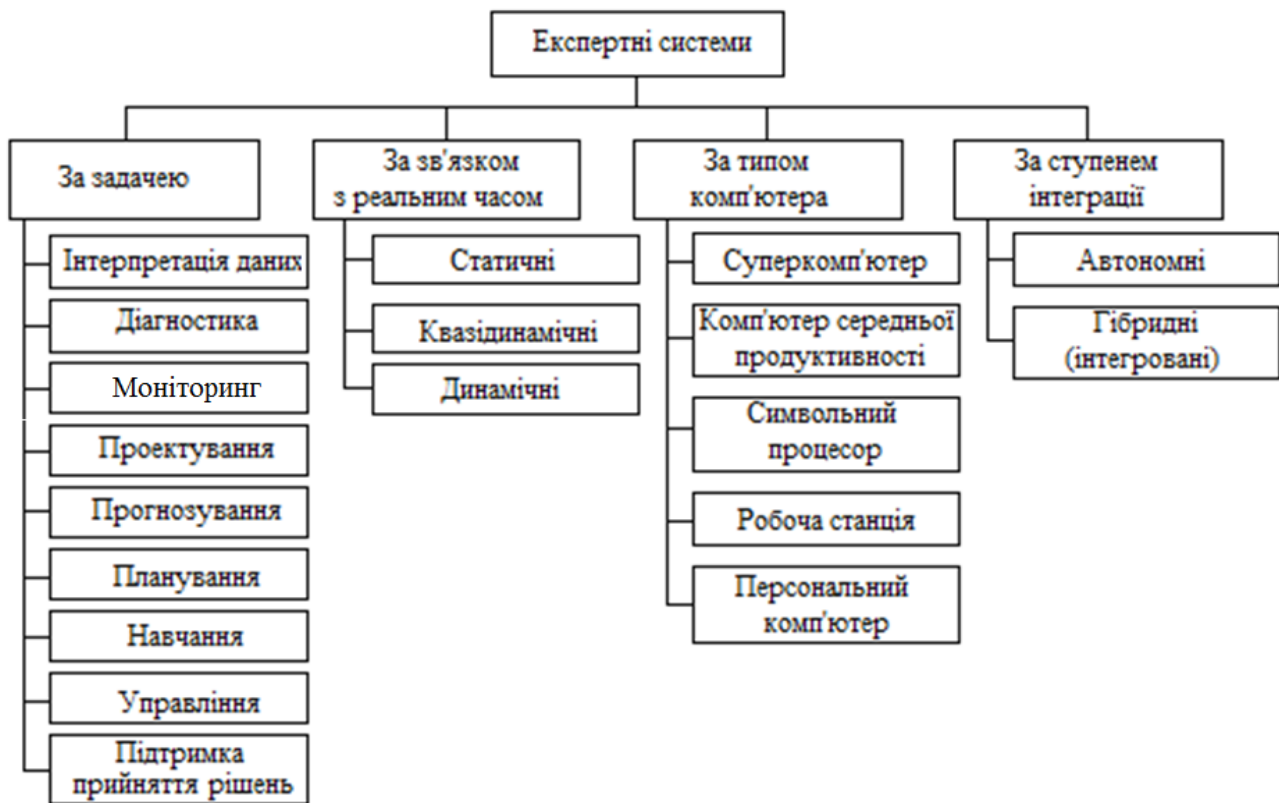


Рисунок 0.2 - Класифікація експертних систем

1.2.1 Класифікація за вирішуваною задачею

Інтерпретація даних

Це одне з традиційних завдань для ЕС. Інтерпретація - це процес визначення значення даних, результати яких повинні бути узгодженими та коректними. Зазвичай передбачається багатоваріантний аналіз даних.

Приклади.

- виявлення та ідентифікація різних типів океанських суден на основі результатів аерокосмічного сканування – ЕС SIAP;

- визначення основних типів властивостей особистості на основі результатів психодіагностичного тестування в системах АВАНТЕСТ і МИКРОЛЮШЕР.

Діагностика

Діагностика – процес співвідношення об'єкта з деяким класом об'єктів та/або виявлення несправності в деякій системі. Несправність – це відхилення від норми. Таке трактування дозволяє з єдиних теоретичних позицій розглядати несправність обладнання в технічних системах, захворювання живих організмів, і різні природні аномалії. Важливою специфікою тут є необхідність розуміння функціональної структури («анатомії») діагностованої системи.

Приклади:

- діагностика та терапія звуження коронарних судин – ANGY;

- діагностика помилок в апаратурі та математичному забезпеченні комп'ютера – CRIB.

Моніторинг

Основне завдання моніторингу – безперервна інтерпретація даних у реальному часі та сигналізація про вихід тих чи інших параметрів за допустимі межі. Головні проблеми – пропуск тривожної ситуації та інверсне завдання «хибного» спрацьовування. Складність цих проблем у розмитості симптомів тривожних ситуацій та необхідність урахування часового контексту.

Приклади:

- контроль роботи електростанцій СПРИНТ, допомога диспетчерам атомного реактора – REACTOR;
- контроль аварійних датчиків на хімічному заводі - FALCON.

Проектування

Проектування полягає у підготовці специфікацій створення «об'єктів» із заздалегідь заданими властивостями. Під специфікацією розуміють весь набір необхідних документів - креслення, пояснювальна записка і т.д. Основні проблеми – отримання чітко структурованого опису знань про об'єкт. Для організації ефективного проектування та ще більшою мірою перепроjektування необхідно формувати не лише самі проектні рішення, а й мотиви їх прийняття. Таким чином, у задачах проектування тісно пов'язуються два основні процеси, виконувані у межах відповідної ЕС: процес виведення рішення та процес пояснення.

Приклади:

- проектування конфігурацій EOM VAX-11/780 у системі XCON, проектування БІС – CADHELP;
- синтез електричних кіл – SYN.

Прогнозування

Прогнозування дозволяє передбачити наслідки деяких подій чи явищ виходячи з аналізу наявних даних. Прогнозуючі системи логічно виводять можливі наслідки із заданих ситуацій. У системі прогнозування зазвичай використовується параметрична динамічна модель, в якій значення параметрів «підганяються» під задану ситуацію. Наслідки, що виводяться з цієї моделі, становлять основу для прогнозів з ймовірнісними оцінками.

Приклади:

- прогноз погоди – WILLARD;
- оцінки майбутнього врожаю – PLANT;
- прогнози економіки – ECON.

Планування

Під плануванням розуміють знаходження планів дій, які стосуються об'єктів, здатних виконувати деякі функції. У таких ЕС використовуються

моделі поведінки реальних об'єктів для того щоб логічно вивести наслідки запланованої діяльності.

Приклади:

- планування поведінки робота - STRIPS;
- планування промислових замовлень – ISIS;
- планування експерименту - MOLGEN.

Навчання

Навчання означає використання комп'ютера для вивчення різних навчальних дисципліни чи предметів. Системи навчання діагностують помилки при вивченні певної дисципліни за допомогою комп'ютера та підказують правильні рішення. Вони містять знання про гіпотетичного «учня» та його характерні помилки, а потім у своїй роботі вони здатні діагностувати слабкі місця у знаннях учнів та знаходити відповідні засоби для їх ліквідації. Крім того, вони планують процес спілкування з учнем залежно від його успіху з метою передачі знань.

Приклади:

- викладання мови програмування LISP в системі вчителів LISP;
- навчання мови Паскаль - система PROUST.

Управління

Під управлінням розуміють функціонування організованої системи, що підтримує певний режим діяльності. Такі ЕС здійснюють управління поведінкою складних систем із заданими специфікаціями.

Приклади:

- допомога в управлінні газової котельні - GAS;
- управління системою календарного планування - Project Assistant.

Підтримка прийняття рішень

Підтримка прийняття рішень – це сукупність процедур, що забезпечує особу, яка приймає рішення, необхідною інформацією та рекомендаціями, що полегшують процес прийняття рішення. Ці ЕС допомагають фахівцям вибрати та/або сформулювати потрібну альтернативу серед множини виборів при прийнятті рішень.

Приклади:

- вибір стратегії виходу фірми з кризової ситуації - CRYISIS;
- допомога у виборі страхової компанії або інвестора - CHOICE.

У загальному випадку всі системи, засновані на знаннях, можуть бути розділені на системи, які вирішують проблеми аналізу та системи, що вирішують проблеми синтезу.

Основна відмінність задач аналізу від задач синтезу полягає в тому, що якщо в задачах аналізу множина рішень може бути перерахована та включена в систему, то у задачах синтезу множина рішень потенційно не обмежується та будується з компонентів або підзадач. Задача аналізу - це інтерпретація даних,

діагностика, підтримка прийняття рішень. До задач синтезу належать дизайн, планування, управління. Комбіновані задачі - навчання, моніторинг, прогнозування.

1.2.2 Класифікація за зв'язком з реальним часом

Статичні ЕС розробляються в предметних областях, в яких база знань та інтерпретовані дані не змінюються в часі, вони є стабільними.

Приклад:

- діагностика несправностей в автомобілі.

Квазидинамічні ЕС інтерпретують ситуацію, яка змінюється з деяким фіксованим інтервалом часу.

Приклад:

- мікробіологічні ЕС, у яких з технологічного процесу знімаються лабораторні вимірювання один раз на 4-5 годин (наприклад, виробництво лізину) та аналізується динаміка отриманих показників по відношенню до попереднього вимірювання.

Динамічні ЕС працюють у поєднанні з датчиками об'єктів у режимі реального часу з безперервною інтерпретацією даних, що надходять у систему.

Приклади:

- управління гнучкими виробничими комплексами, моніторинг в інтенсивних підрозділах;
- програмні засоби для розробки динамічних систем - G2.

1.2.3 Класифікація за типом комп'ютера

Станом на сьогодні є:

- ЕС для унікальних стратегічно важливих задач на супер-комп'ютерах – CRAY, CONVEX;
- ЕС на комп'ютерах середньої продуктивності типу Mainframe;
- ЕС на символічних процесорах і робочих станціях SUN, Silicon Graphics, APOLLO;
- ЕС на міні- и суперміні-комп'ютерах VAX, micro-VAX;
- ЕС на персональних комп'ютерах.

1.2.4 Класифікація за ступенем інтеграції з іншими програмами

Автономні ЕС працюють безпосередньо в режимі консультації з користувачем для конкретних «експертних» завдань, для рішення яких не потрібно застосування традиційних методів обробки даних (розрахунків, моделювання тощо).

Гібридні ЕС представляють програмний комплекс, який агрегує стандартні пакети прикладних програм (ППП) (наприклад, матстатистика, лінійне програмування, СУБД) та засоби маніпулювання знаннями. Це може бути інтелектуальна надбудова над PPP або інтегроване середовище для рішення складної задачі з елементами експертних знань.

Незважаючи на зовнішню привабливість гібридного підходу, слід зазначити, що розвиток таких систем є на порядок складнішим, ніж розвиток автономних ЕС. Поєднання не просто різних пакетів, а різних методологій (які виникають у гібридних системах) породжують цілий комплекс теоретичних та практичних труднощів.

Контрольні запитання

- 1 Дайте визначення експертної системи.
- 2 Назвіть задачі, які можуть вирішувати ЕС.
- 3 Наведіть характеристики ЕС.
- 4 Які проблеми існують у галузі ЕС?
- 5 Які системи називаються системами, що базуються на знаннях?
- 6 Які аспекти інтелектуальної поведінки відтворюють системи, що базуються на знаннях?
- 7 Перелічіть властивості неформалізованих завдань.
- 8 За якими критеріями можна класифікувати ЕС?
- 9 Поясніть класифікацію ЕС за розв'язуваною задачею.
- 10 Поясніть класифікацію ЕС у зв'язку з реальним часом.
- 11 Поясніть класифікацію ЕС за типом ЕОМ.
- 12 Поясніть класифікацію ЕС за ступенем інтеграції.

2 СТРУКТУРИ ТА ОСНОВНІ КОМПОНЕНТИ ЕКСПЕРТНИХ СИСТЕМ

2.1 Технологія експертних систем

Експертні системи є гілкою ІІІ, яка уможлиблює використання спеціальних знань для вирішення проблем на рівні людини-експерта. Експерт - це людина, яка має досвід у певній галузі. Тобто експерт має знання або спеціальні навички, які не відомі та не доступні для більшості людей. Експерт може вирішити проблеми, які більшість людей не можуть вирішити, або може вирішити їх ефективніше (але не дешево). У 1970-х роках, коли експертні системи були вперше розроблені, вони містили винятково експертні знання. Проте термін експертна система часто застосовується сьогодні для будь-якої системи, яка використовує технологію експертних систем. Ця технологія складається зі спеціальних мов експертних систем, програм та технічних засобів, призначених для надання допомоги у розробці та реалізації експертних систем.

Головна ідея використання технології експертних систем полягає в тому, щоб отримати від експерта його знання і, завантаживши їх у пам'ять комп'ютера, використовувати щоразу, коли в цьому виникне необхідність. ЕС являють собою комп'ютерні програми, що трансформують досвід експертів у будь-якій галузі знань у форму евристичних правил (евристик). Технологія ЕС використовується для побудови систем, що радять.

Подібність інформаційних технологій, що використовуються в експертних системах та системах підтримки рішень, полягає в тому, що обидві вони забезпечують високий рівень підтримки рішень. Однак, між ними є суттєві відмінності:

1) рішення проблеми в рамках СППР відображає рівень її розуміння користувачем та його здатність отримувати та розуміти рішення; технологія ЕС, навпаки, пропонує користувачеві прийняти рішення, яке перевищує його можливості;

2) ЕС здатні пояснювати свої міркування у процесі отримання рішення, часто ці пояснення виявляються важливішими для користувача, ніж саме рішення;

3) в ЕС використовується новий компонент інформаційної технології - знання.

Перевага застосування ЕС полягає у можливості прийняття рішень в унікальних ситуаціях, для яких алгоритм заздалегідь не відомий та формується за вхідними даними у вигляді ланцюжка міркувань із знань, що наявні у базі знань.

Знання в експертних системах можуть бути досвідом, або знаннями, які зазвичай містяться в опублікованих джерелах інформації (книгах, журналах), та знаннями, якими володіють компетентні люди. Терміни "експертна система", "система, заснована на знаннях" або "експертна система на основі знань", часто

використовується як синоніми. Більшість людей використовують термін "експертна система" просто тому, що він коротший.

У системі, заснованій на експертних знаннях (рис. 2.1), користувач вводить факти чи іншу інформацію. Експертна система обробляє їх і формує висновок у формі експертних порад. Експертна система складається з двох основних компонентів: *база знань*, що містить знання, *машина виведення*, що формує висновки на основі знань. Ці висновки є відповідями експертної системи на запити користувачів на виконання експертизи.

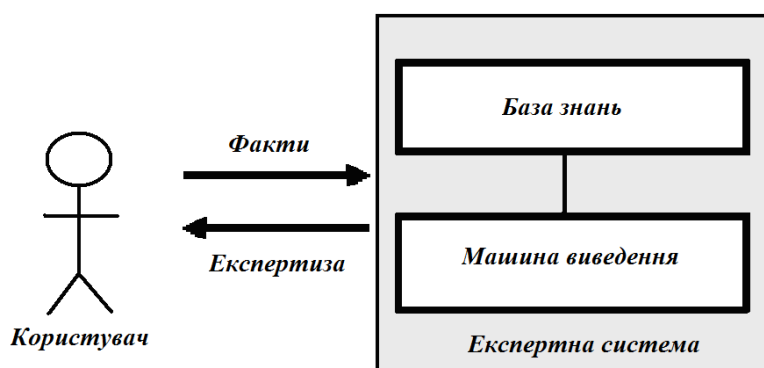


Рисунок 2.1 – Основні концепції функціонування ЕС

Системи, засновані на знаннях, можуть використовуватися як інтелектуальний помічник людини-експерта. Такі інтелектуальні помічники розробляються на основі технологій експертних систем і мають відповідні переваги. Що більше знань містить інтелектуальний помічник, то краще він повторює дії експерта. Отже, розвиток інтелектуальних помічників може бути корисним етапом у створенні повноцінної експертної системи. Крім того, така система-помічник може вивільнити більше часу експерта та прискорити вирішення проблеми.

Інтелектуальні програми, що навчають, є ще одним застосуванням штучного інтелекту. На відміну від звичайної комп'ютерної системи, що навчає, інтелектуальні системи можуть забезпечити контекстно-залежні інструкції навчання.

Знання експерта належать лише до однієї тематичної області, на відміну від знань про загальні методи вирішення проблем. Предметна область - це спеціально обмежена область проблеми, наприклад, медицина, фінанси, наука, техніка, тощо, в якій експерт може успішно вирішити проблеми. Експертна система, як і людина-експерт, зазвичай призначена для фахівців в одній предметній області. Знання, притаманні одній предметній області, не можна автоматично перенести в іншу.

Предметна область - сфера людської діяльності, виділена та описана згідно з встановленими критеріями. У опис предметної області повинні входити відомості про її елементи, явища, відношення і процеси, що відображають різні аспекти цієї діяльності. В описі предметної області повинні бути характеристики можливих впливів навколишнього середовища на елементи і

явища предметної області, а також зворотний вплив цих елементів і явищ на середовище.

Проблемна область (рис. 2.2) - комплексне поняття, що включає предметну область, розв'язувані завдання, цілі, можливі стратегії та евристики. Предметну область можна визначити як об'єкт або, наприклад, виробничу систему з усім комплексом понять та знань про її функціонування. При дослідженні проблемної області необхідно мати знання про завдання, що вирішуються, і цілі, що стоять перед нею.



Рисунок 2.2 – Компоненти проблемної області

Аналіз предметної області являє собою особливий вид наукової діяльності, у результаті якої будується інтерпретаційна модель предметних знань. У процесі аналізу знання діляться на інваріантні (здогадка, гіпотеза, думка, віра) та прагматичні (описують об'єкти та процеси щодо цілей розв'язуваного завдання).

Системний підхід до пізнання орієнтує аналітика на розгляд будь-якої предметної області з позицій закономірностей системного цілого та взаємодії складових його частин. Системність знань виходить із багаторівневої ієрархічної організації будь-якої сутності, тобто, всі об'єкти, процеси та явища можна розглядати як множину дрібніших підмножин (ознак, деталей) і, навпаки, будь-які об'єкти можна (і потрібно) розглядати як елементи вищих класів узагальнення.

Проблемна область - це модель всіх таких аспектів або компонент предметної області, з якими пов'язані (опосередковано або безпосередньо) знання, необхідні при вирішенні різних завдань у цій предметній області. Будь-яке проблемна область складається з двох блоків: інваріантної (відносно незмінної) частини та множини змінних частин, що відповідають окремим завданням.

У складі інваріантної частини, наприклад у методології SMEE (Structured Methodology for Elicitation of Expertise), виділяють сім типів компонентів:

об'єкти, інструменти, оператори, операції, кінцеві продукти, побічні продукти та обмеження.

Знання - це реальний стан справ, обґрунтований фактами та раціональними аргументами переконання людини. Знання експерта (або групи експертів) - це володіння інформацією, що дозволяє вирішити завдання.

У теорії штучного інтелекту та експертних систем знання - це сукупність тверджень про властивості об'єктів, закономірності процесів і явищ, правил логічного виведення одних тверджень з інших та правил використання їх для прийняття рішень. Головна відмінність знань від даних полягає у їхній структурності та активності: поява в базі знань нових фактів чи встановлення нових зв'язків між ними може стати джерелом змін у прийнятті рішень.

Термін «область знань» схожий на термін «знання», але безпосередньо стосується конкретної предметної області, котра досліджується з метою вирішення проблем.

Область знань - це набір інформації про те, що потрібно вміти, щоб зробити. Область знань містить завдання та методи їх вирішення.

Між проблемою та областю знань існує певний зв'язок (рис. 2.3). Знання експерта щодо вирішення конкретних проблем називають областю знань експерта. Наприклад, медична експертна система, призначена для діагностики інфекційних захворювань, містить множину знань про певні симптоми, спричинені інфекційними захворюваннями. У цьому випадку областю знань є медицина і вона складається зі знань про захворювання, симптоми та лікування. Конкретна сфера знань повністю включена в проблемну область.



Рисунок 2.3 – Проблемна область та область знань

У певній області знань експертна система знаходить причини або робить висновки так само, як і людина-експерт знаходить вирішення проблеми. Тобто, з урахуванням введених фактів виконується логічне виведення і робиться висновок. Наприклад, якщо ви протягом місяця не отримували повідомлень від друга електронною поштою можна зробити висновок, що йому нема чого вам сказати. Тим не менш, це лише один із кількох можливих висновків.

Як і в будь-якій новій технології, у технології експертних систем є ще багато не вивчених аспектів. Таблиця 2.1 відображає різні думки учасників технології експертних систем.

Таблиця 2.1 – Різні погляди на технологію ЕС

Учасник технології	Запитання
Менеджер	Для чого я її можу використовувати?
Технолог	Як я можу найкращим чином її реалізувати?
Дослідник	Як я можу її розширити?
Користувач	Як мені це допоможе? Чи виправдані клопоти та витрати? Наскільки вона надійна?

Технолог може бути інженером зі знань або розробником програмного забезпечення, а технологія може бути представлена апаратним або програмним забезпеченням. При вирішенні будь-якої проблеми це питання, на які потрібно знайти відповіді, інакше технологія не буде успішно використана. Як і будь-який інший інструмент, експертні системи можуть мати правильне та неправильне застосування. З набуттям досвіду ми можемо оцінити, чи правильно застосовуємо ЕС.

2.2 Переваги ЕС

ЕС мають низку переваг.

1. Підвищена доступність. Експертиза доступна на будь-якому комп'ютері. ЕС дозволяє широкому колу користувачів виконувати експертизу.
2. Зниження вартості експертизи. Вартість виконання експертизи для одного користувача значно знижується.
3. Зменшення небезпеки. ЕС можуть використовуватись у середовищах, які можуть бути небезпечними для людини.
4. Постійність. Виконання експертизи є незмінним. На відміну від людини-експерта, яка може звільнитися чи померти, знання експертної системи будуть доступні безкінечно.
5. Численні експертизи. Знання кількох експертів можуть бути надані та працювати одночасно і безперервно на вирішення проблеми у будь-який час дня та ночі. Рівень знань кількох експертів може перевищувати знання однієї людини-експерта.
6. Підвищена надійність. ЕС збільшує впевненість у правильності рішення шляхом використання думок інших експертів або шляхом виключення думки експерта у разі розбіжностей з іншими експертами. Звичайно, цей метод не працює, якщо експертна система була запрограмована на роботу з одним експертом. Рішення експертної системи завжди повинно співпадати з рішенням експерта, якщо експертом не було допущено помилки. Однак, помилки може припуститися людина-експерт, якщо вона втомила або перебуває у стані стресу.

7. Пояснення. ЕС може докладно пояснити міркування, які спричинили висновок. Людина може втомитися, не бажати або не могли робити пояснення весь час. Пояснення збільшує впевненість у правильності рішення.

8. Швидкий відгук. Швидка реакція або реакція в режимі реального часу може бути потрібна в деяких випадках. Залежно від програмного забезпечення та апаратних засобів, експертна система може реагувати швидше і бути доступнішою, ніж людина-експерт. У деяких надзвичайних ситуаціях можуть бути потрібні відповіді швидше, ніж їх може дати людина, тому для роботи в режимі реального часу експертна система є хорошим вибором.

9. Спокій, неупередженість і повна відповідь у будь-який час. Це може бути дуже важливо в режимі реального часу та надзвичайних ситуаціях, коли людина-експерт не може працювати з максимальною ефективністю через стрес чи втому.

10. Інтелектуальний наставник. ЕС може виступати в ролі інтелектуального наставника, дозволяючи особам, що навчаються, виконувати завдання-зразки та пояснюючи перебіг міркувань системи.

11. Інтелектуальна база даних. Експертні системи можуть бути використані для доступу до бази даних та виконувати інтелектуальну обробку та аналіз даних.

Процес розробки експертної системи має опосередковані переваги, оскільки знання людини-експерта слід чітко сформулювати для введення у базу знань ЕС. У ЕС очевидні знання та неявні знання, якими володіє експерт, можна перевірити на узгодженість, правильність, послідовність та повноту. Якщо знання суперечливі, непослідовні, вони повинні бути скориговані або переглянуті, тобто, ЕС покращує якість знань.

Хоча ці переваги очевидні, слід зазначити, що ЕС не має інтуїції та загальних знань про світ, хід її мислення та метод вирішення проблеми не можуть вийти за рамки знань про те, що в неї закладено. ЕС також буде безсилою при вирішенні проблеми за змінних умов, наприклад, при зміні методології рішення або появі нового обладнання. Експерти можуть безпосередньо сприймати весь комплекс вхідної сенсорної інформації - візуальної, звукової, тактильної чи нюхової. ЕС сприймає лише символи, якими представлені знання. Тому сенсорна інформація повинна бути проаналізована та перетворена в цифрову, символічну форму, придатну для обробки комп'ютером. При перетворенні людиною сенсорної інформації неминуче виникають спотворення та втрати. Лише людина здатна класифікувати інформацію зі всього потоку як значну, вторинну чи абсурдну. Так, наприклад, будь-яка людина негайно висловлює здивування, якщо її попросять знайти номер телефону Арістотеля, але навряд чи є програма, яка скаже, що давньогрецькі філософи не використовували телефони.

Таким чином, призначення експертних систем - консультування з вузькоспеціалізованих питань при прийнятті рішень людиною. Тобто, ЕС використовується для зміцнення та розширення професійних можливостей своїх користувачів.

2.3 Основні концепції експертних систем

Знання експертної системи можуть бути представлені різними способами – це можуть бути правила, описи ситуацій, описи об'єктів та відношень між ними. Один із найпоширеніших методів - подання знань у вигляді правил продукцій типу ЯКЩО – ТО, наприклад

ЯКЩО світло=червоний ТО стоп

Якщо світло дійсно червоне, це відповідає факту "світло=червоний". Наведене правило виконується, і об'єкт виконує дію "стоп". Це дуже простий приклад, але багато складних експертних систем були побудовані з поданням знань експертів у вигляді правил. Підхід до розробки експертних систем, що ґрунтується на знаннях, повністю витіснив ранній підхід до II 1950-х та 1960-х років, коли намагалися використовувати складні методи міркувань, без опори на знання. Деякі види інструментальних середовищ розробки ЕС, такі як CLIPS, Visual Prolog, дозволяють використовувати поєднання об'єктів і правил. Правила можуть використовувати шаблони об'єктів та шаблони фактів. Крім того, об'єкти можуть працювати незалежно від правил.

В даний час було розроблено багато експертних систем, заснованих на знаннях. Великі системи, що містять тисячі правил, такі як система XCON/RL корпорації цифрового обладнання Digital Equipment Corporation, «знають» набагато більше, ніж будь-який окремих експерт у налаштуванні комп'ютерних систем. Ряд невеликих систем для вирішення спеціалізованих проблем також було побудовано на основі декількох сотень правил. Ці невеликі системи не можуть працювати на рівні людини-експерта, але розроблені для використання переваг технології експертної системи для виконання завдань з високими технологіями. Системи знань цих невеликих систем можуть міститись у книгах, журналах або з інших опублікованих документах.

Класична експертна система втілює неявні знання, які інженеру зі знань слід здобувати від експертів з використанням інтерв'ю протягом тривалого періоду часу. Процес створення експертної системи називається інженерією знань і проводиться інженером знань (когнітологом). Інженерія знань передбачає здобуття знань від людини-експерта чи іншого джерела та його втілення в експертній системі у формі програмного коду.

Зміст процесу побудови експертної системи показаний на рис. 2.4.

Інженер зі знань встановлює діалог з експертом-людиною для отримання експертних знань. Цей етап у звичайному програмуванні схожий на процес обговорення системних вимог розробника з клієнтом, для якого буде побудована програма. Потім інженер зі знань кодує знання безпосередньо в базі знань. Експерт оцінює експертну систему та коментує знання, представлені когнітологом. Цей процес є ітераційним і повторюється, поки експерт не оцінить ефективність роботи системи як задовільну.

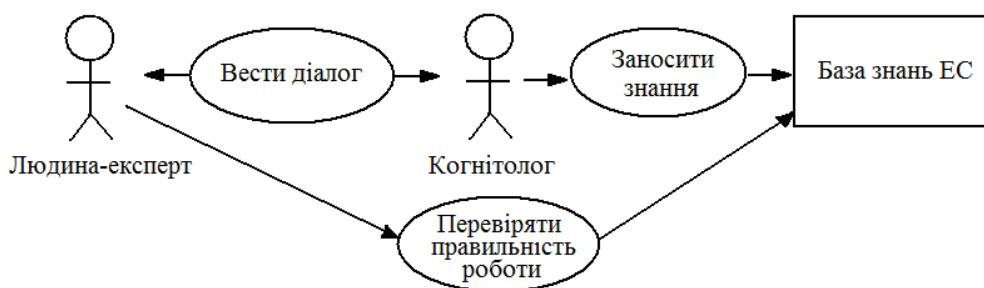


Рисунок 2.4 – Процес побудови ЕС

Вираз "система, заснована на знаннях" підходить для визначення як експертних систем, так і інших систем, побудованих на основі знань. Однак, як і термін штучний інтелект, цей термін використовується, коли мова йде про експертні системи, і системи, засновані на знаннях, навіть якщо знання людини-експерта у систему не закладені.

Розробка експертних систем, як правило, відрізняється від розробки звичайних програм, оскільки проблеми, які потребують вирішення, зазвичай не мають алгоритмічного рішення, і ЕС повинна базуватися на логічному виведенні для досягнення розумного рішення. Пошук розумного рішення - найкращий підхід, якщо немає алгоритму для отримання оптимального рішення. Оскільки експертна система заснована на логічному виведенні, вона повинна мати можливість пояснити свої міркування, щоб їх можна було перевірити. Компонент пояснення є невід'ємною частиною складних експертних систем.

Деякі експертні системи дозволяють виводити нові правила на основі аналізу таблиць даних, наприклад, за допомогою індукції. Це є частиною технології Data Mining.

Формалізація знань фахівців у вигляді правил є складним процесом, особливо якщо знання експертів не були до цього систематично досліджені. Знання експертів можуть бути суперечливими, неточними, дубльованими, і це може не бути очевидним, доки не буде зроблено спробу формально представити знання в експертній системі.

Практичним обмеженням багатьох експертних систем сьогодні є відсутність причинно-наслідкових знань. Тобто експертні системи насправді «не розуміють» основні причини та наслідки в системі. Набагато легше реалізувати експертну систему з поверхневими знаннями, заснованими на емпіричних та евристичних знаннях, ніж із глибокими знаннями, заснованими на базових структурах, функціях та поведінці об'єктів. Наприклад, набагато легше запрограмувати експертну систему, що призначає аспірин від головного болю людині, ніж запрограмувати усі основні біохімічні, фізіологічні, анатомічні та неврологічні знання про людський організм. Програмування причинно-наслідкової моделі людського тіла є складним завданням, і навіть у разі успіху час відгуку системи, ймовірно, буде дуже великим.

Одним із видів поверхневих знань є евристичні знання. Евристика не гарантує успіху, на відміну від алгоритму, який дає гарантоване рішення проблеми. Евристичні правила або емпіричні знання, отримані з досвіду,

можуть допомогти у пошуку рішення, але не гарантують його правильність. Тим не менш, у багатьох сферах, таких як медицина та техніка, евристика відіграє важливу роль у вирішенні деяких видів проблем. Евристика може дати цінні поради, які можуть скоротити час та витрати на вирішення проблеми.

Іншою проблемою експертних систем сьогодні є те, що їхній досвід обмежується предметною областю, яку знає система. Типові експертні системи не можуть у повній мірі узагальнювати свої знання, використовуючи аналогію міркувань у новій ситуації, як це можуть люди. Хоча можна застосовувати правила індукції, але лише обмежені види знань можуть бути виведені таким чином в експертній системі. Отримання та формальне подання людських знань в експертній системі є дуже складним та трудомістким завданням.

Незважаючи на існуючі обмеження, експертні системи успішно вирішують реальні проблеми, які традиційні методики програмування не в змозі вирішити, особливо такі, що мають справу з невизначеною чи неповною інформацією. Важливо розуміти переваги та недоліки технології експертних систем для її використання належним чином.

2.4 Структура та принцип роботи експертних систем

Усі ЕС мають схожу структуру. В основі цієї структури лежить поділ знань, закладених у систему, та алгоритмів їх обробки. Так, наприклад, програма, що вирішує квадратне рівняння, безсумнівно, використовує знання про те, як слід вирішувати цей вид рівнянь. Але це знання "зашито" в коді програми і його не можна ні прочитати, ні змінити, якщо вихідні коди програми недоступні.

Програми такого класу дуже зручні для тих, хто вирішує лише квадратні рівняння. Однак, якщо користувач хоче вирішити інший тип рівняння, йому не обійтися без програміста, який зможе написати йому нову програму. Тепер, припустимо, завдання поставлено трохи інакше: програма повинна при запуску зчитувати з текстового файлу тип рівняння та спосіб його вирішення, і користувач повинен мати можливість самостійно вводити нові способи розв'язання рівнянь, наприклад, щоб порівняти їх ефективність, точність та ін. Формат цього файлу має бути однаково «зрозумілий» як комп'ютеру, і користувачеві. Такий спосіб організації програми дозволить змінювати можливості без допомоги програміста.

Навіть якщо користувач вирішує тільки один тип рівнянь, новий підхід кращий за попередній хоча б тому, що зрозуміти принцип розв'язання рівнянь, можна просто вивчивши вхідний текстовий файл. Даний приклад, незважаючи на свою простоту та нетиповість предметної галузі для застосування технології ЕС (для вирішення математичних рівнянь зазвичай використовують спеціалізовані пакети програм, а не експертні системи), добре ілюструє особливість структури ЕС – наявність бази знань, яку користувач може переглянути безпосередньо або за допомогою спеціального редактора.

Базу знань також можна редагувати, що дозволяє змінювати роботу ЕС без її перепрограмування.

Реальні ЕС можуть мати складну, розгалужену структуру модулів, але для будь-якої ЕС необхідна наявність основних компонентів (рис. 2.5).

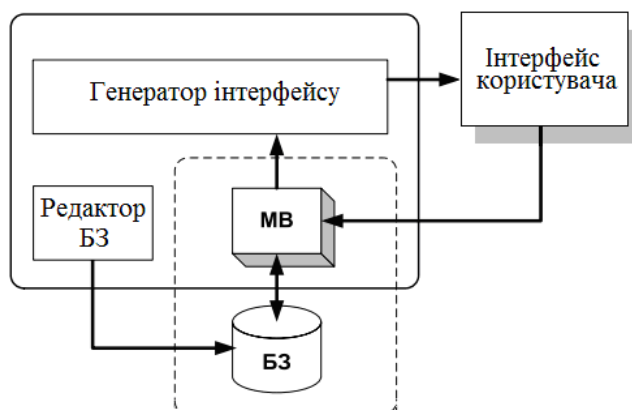


Рисунок 2.5 - Узагальнена структура ЕС

База знань (БЗ) – найбільш цінний компонент ядра ЕС. БЗ містить сукупність знань про предметну область та способи вирішення завдань. Знання подаються у формі, зрозумілій неспеціалістам у програмуванні: експерту, користувачеві та ін. Зазвичай знання в БЗ записуються у формі, наближеній до природної мови. Форма запису знань отримала назву мову подання знань (МПЗ). У різних системах можна використовувати різні МПЗ. Паралельно з таким "людським" поданням БЗ може існувати у внутрішньому "машинному" представленні. Перетворення між різними формами представлення БЗ має здійснюватися автоматично, оскільки редагування БЗ не передбачає участь програміста-розробника.

Машина виведення (МВ) – це блок, який моделює перебіг міркувань експерта, виходячи зі знань, закладених у БЗ. Машина виведення є постійною частиною ЕС. Однак більшість реальних ЕС мають вбудовані засоби управління ходом логічного виведення за допомогою так званих метаправил, що записуються в БЗ.

Редактор бази знань, призначений для розробників ЕС. За допомогою цього редактора до БЗ додаються нові знання або редагуються існуючі.

Інтерфейс користувача – це блок, призначений для взаємодії з користувачем. Через інтерфейс користувач вводить необхідні для виконання експертизи дані та отримує результат. Система може мати "жорсткий" інтерфейс, орієнтований на певний метод введення та виведення інформації, або може включати інструменти дизайну для спеціалізованих інтерфейсів (генератора інтерфейсу) для більш ефективної взаємодії з користувачем.

З точки зору технології експертних систем найбільшим інтересом вивчення є база знань та машина виведення.

У процесі функціонування ЕС читає інформацію зі своєї бази знань і намагається реалізувати логічне виведення для вирішення завдання. У базі знань можна зберігати два основні типи записів: факти, що описують стан предметної області та складаються з опису об'єктів та їхніх властивостей, і правила, що описують методи вирішення проблеми. Усі правила продукційної БЗ мають однакову форму запису і складаються з двох частин: умови та дії.

Попередній етап у роботі ЕС - це збірка початкових фактів, що описують проблему на мові подання знань. Ці факти можна вводити в систему різними способами: у діалоговому режимі через інтерфейс користувача, через файли або бази даних, із зовнішніх датчиків або пристроїв. Прочитавши початкову інформацію, машина виведення починає переглядати базу знань і послідовно порівнює опис проблеми із записами, наявними у БЗ, таким чином формуючи хід рішення. Якщо умова поточного правила БЗ підтверджується початково заданими фактами, то система виконує дії, записані в певному правилі, додаючи нові, похідні факти до БЗ.

На перший погляд, процес виведення здається досить простим - однаковий тип операцій прийому записів у БЗ та порівняння їх з існуючими фактами виконується до тих пір, поки не знайдеться рішення чи певний цільовий факт. Однак управління процесом виведення, який не залежить від контексту проблеми, є неефективним для практики. Вирішуючи реальні проблеми, людина рідко вдається до перебору усіх даних. Натомість люди використовують евристичні правила, які значно обмежують простір рішення і дозволяють швидко та ефективно вирішувати проблеми. Евристичні знання мають емпіричну природу, тобто вони формуються на основі досвіду та інтуїції експерта. Вражаючий приклад переваги евристичного підходу над алгоритмічного (на основі повного або часткового перебору) - це гра в шахи. На початку гри "білі" мають можливість зробити будь-який з 20 допустимих ходів, у відповідь, на який "чорні" також можуть зробити один із 20 ходів. Неважко підрахувати, що наступний хід "білих" може бути вибраний вже з 400 можливих ходів залежно від поточного стану партії. Крім того, у міру розвитку гри може настати неконтрольований комбінаторний вибух.

Особливо гостра проблема виникає в ендшпілі. Маючи на дошці кілька фігур, кожен з гравців має понад 50 варіантів можливих кроків. Очевидно, що шахові майстри з усім бажанням не змогли б виконати повний перебір ходів для пошуку найкращого варіанту. Натомість вони використовують короткотермінові та довготермінові стратегії. Кожна конкретна стратегія вибирається відповідно до поточної ситуації на ігровій дошці.

Ще одним простим прикладом є метод побудови стін «сухим» методом. На першому етапі роботи є велика кількість каменів різних форм, з яких потрібно скласти плоску і стабільну стіну. Більше того, камені можна підвезти за необхідності, а огляд усіх каменів для цілей підбору може бути в принципі неможливим. Спочатку будівельник не знає, як і які камені він вибере. У процесі будівництва час від часу він вивчає стіну, визначає, які камені залишаються, і вибирає коротку стратегію, зокрема, включаючи зняття каменів

зі стіни. Маючи той самий набір каменів, він, можливо, ніколи не зможе двічі побудувати однаково стіну.

Існує два основні типи логічного виведення: пряме та зворотне. Пряме виведення відповідає звичайному ходу рішення проблеми - від початкових фактів до цільових. Прикладом прямого виведення є задача класифікації. ЕС проводить поступове узагальнення початкових фактів, що описують властивості досліджуваного об'єкта, виявляючи найбільш характерні ознаки певного класу. Зворотне виведення відповідає протилежному напрямку для пошуку рішення: По-перше, машина виведення враховує правила БЗ, які дають цільовий факт. Потім з умов цих правил вибираються нові підцілі, і процес продовжується від цільових фактів до вихідних. Можна сказати, що під час зворотного виведення виконується конкретизація властивостей досліджуваного об'єкта. Цей тип логічного виведення дає ЕС нову фундаментальну властивість - здатність пояснити, як було отримано рішення, або що потрібно, щоб цей чи той факт відбувся. У реальних системах, як правило, використовується комбінація прямого та зворотного виведення.

Для управління процесом логічного виведення використовуються спеціальні правила - метаправила. Використовуючи метаправила, можна впорядкувати використання знань залежно від конкретних значень фактів та поточного стану БЗ. Можна продемонструвати різницю між метаправилами та звичайними правилами на прикладі. Нехай треба розмістити меблі (столи, стільці, парти та ін.) в аудиторії університету, враховуючи вимоги ергономіки, безпеки тощо. На основі знань про обладнання приміщень, залежно від місця розташування та розміру аудиторії, типу занять (лекції, практики чи лабораторні роботи) та інших параметрів, до БЗ внесено правила, що визначають той чи інший метод розміщення меблів. Це звичайний тип правил. Але в цій предметній області може знадобитися уточнити метод вирішення проблеми за допомогою метаправил типу "Якщо є властивість *X*, то спочатку застосувати групу правил *M*". Прикладом такого метаправила може бути: "Якщо аудиторія призначена для лабораторних занять, то спочатку застосувати правила щодо розміщення комп'ютерів та лабораторного обладнання, а потім правила щодо розміщення меблів". Якщо звичайні правила БЗ представляють кроки для вирішення проблеми, то метаправила описують стратегію пошуку рішень.

Оскільки БЗ в архітектурі ЕС є компонентом, що може змінюватися, виникає питання: "Чи можна взяти готову експертну систему з однієї тематичної області, занести знання з іншої тематичної області в неї та отримати нову ЕС?" Для редагування або навіть повної заміни вмісту БЗ не потрібно змінювати код ЕС та залучати програмістів, тому в принципі таке застосування готових програмних рішень є можливим. Дослідження в цьому напрямку призвели до створення так званих оболонок експертних систем. Оболонки ЕС включають машину виведення та інтерпретатор мови подання знань, інтерфейс розробника та інструменти дизайну інтерфейсу користувача. Наповнення оболонки БЗ дозволяє отримати ЕС для рішення різних завдань. Повторне використання розроблених компонентів ЕС значно скорочує час створення

нових ЕС. Однак, як показала практика використання оболонок ЕС, перенесення методів рішень та засобів представлення знань з однієї області знань до іншої не завжди є можливим. Інструменти, що успішно використовуються для одного типу завдань, можуть бути неефективними при спробі їх використання для вирішення інших типів завдань. Структура та методи опису знань у завданнях медичної діагностики та пошук несправностей в електронних схемах значно відрізняються від тих, що використовуються в розробці електричних кіл або вибору конфігурації комп'ютера. Таким чином, виникла нова область досліджень - класифікація експертних завдань, таких як медична діагностика, планування, інтерпретація сигналів тощо. Були зроблені спроби евристичної класифікації методів опису знань та методів рішення залежно від проблеми, що вирішується. Така класифікація почала розглядатися як етап, який передуює вибору методів та інструментів для вирішення проблем.

Структура ЕС, наведена на рис. 2.5, є базовою та може бути розширена (рис. 2.6). Штриховими лініями позначені компоненти, призначені для взаємодії з зовнішнім світом – підсистема моделювання зовнішнього світу, інтерфейс із зовнішнім світом, система датчиків, необхідна для отримання та інтерпретації даних у реальному часі. Експертні системи можуть накопичувати досвід у вигляді прецедентів (ситуацій, що вже відбувалися і є дозволеними), які зберігаються в базі знань і використовуються в майбутньому.

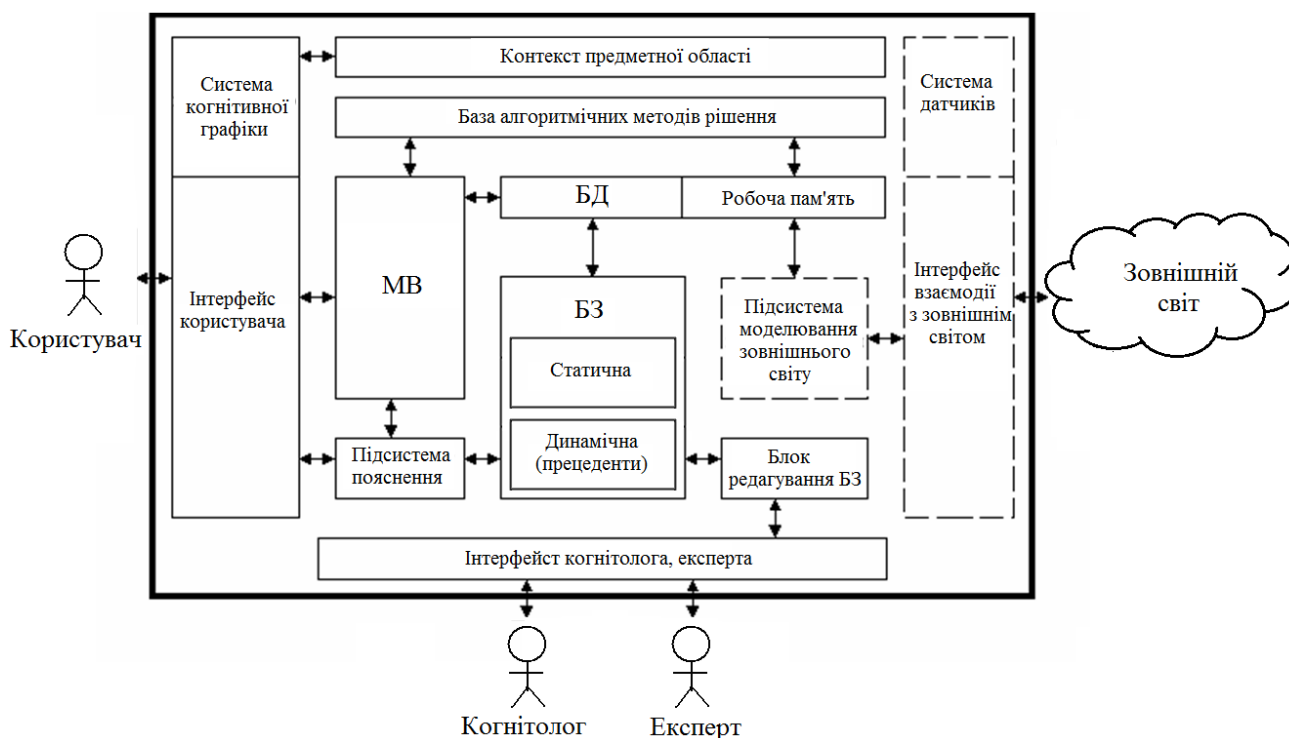


Рисунок 2.6 – Розширена структура ЕС

Блок алгоритмічних методів рішення включає всі обчислювальні операції та алгоритми, реалізовані традиційними методами програмування, інтегрованими в експертну систему. Асоціація традиційних методів

програмування та штучного інтелекту в рамках експертної системи може значно підвищити ефективність та якість прийняті рішення.

Специфіка предметної області, для якої побудована система, відображається та описується не лише в базі знань, але і в підсистемі "Контекст предметної області", дозволяє більш чітко подати вхідну і вихідну інформацію у формі, прийнятній для конкретної предметної області.

Сучасні інформаційні системи часто використовують архітектуру клієнт-сервер, яка дозволяє будувати розподілені та мережеві програми. Завдяки архітектурі клієнт-сервер, застосованій для побудови експертної системи, на стороні клієнта є інтерфейси користувача, експерта, зовнішнього середовища та системи датчиків. Решта блоків розташована на сервері.

Контрольні запитання

1. Поясніть поняття "експертна система".
2. Чому програмна ЕС є інтелектуальною комп'ютерною програмою?
3. Що являє собою технологія експертних систем?
4. Назвіть відмінності між ЕС та СППР.
5. Наведіть основні концепції функціонування ЕС.
6. Поясніть співвідношення предметної та проблемної області.
7. Хто є учасниками розробки експертних систем? Які функції вони виконують?
8. Назвіть переваги ЕС.
9. Назвіть обмеження та проблеми ЕС.
10. Які сфери застосування мають експертні системи?
11. Поясніть зміст процесу побудови експертної системи.
12. Поясніть структуру та принцип роботи експертних систем.
13. Що таке база знань? Прокоментуйте місце БЗ у ЕС.
14. Що таке машина виведення? Прокоментуйте місце МВ у ЕС.
15. Які компоненти є змінними в архітектурі ЕС?
16. Поясніть розширену структуру ЕС.

3 РОЗРОБКА ЕКСПЕРТНИХ СИСТЕМ

Процес розробки ЕС відрізняється від процесу розробки звичайної програмної системи. Використання традиційних технологій програмування або надмірно затримує процес розробки, або призводить до негативних результатів. Це пов'язано з тим, що вирішувані завдання є слабоструктурованими або неструктурованими. Найбільшою проблемою є процедура вилучення експертних знань та їх подання в ЕС. Для побудови ЕС та розробки програмного забезпечення була запропонована технологія створення ЕС.

3.1 Процес розробки експертних систем

Архітектура різних ЕС, з точки зору програмних модулів, що входять до них, є однаковою майже будь-яких завдань. Деталі реалізації модулів можуть значно відрізнятися в різних проектах, але їхній основний склад та взаємодія є чітко визначеними. Більшість сучасних ЕС базуються на знаннях і відрізняються від звичайних програм (табл. 3.1).

Таблиця 3.1 - Відмінності між звичайними програмами та експертними системами

Характеристика	Звичайні програми	Експертні системи
Управління	За порядком операторів	Машиною виведення
Управління та дані	Неявні об'єднання	Явні розділення
Потужність управління	Висока	Низька
Рішення	За алгоритмом	За механізмом виведення
Пошук рішення	Найчастіше відсутній	Значною мірою
Рішення проблем	За відомим алгоритмом	За правилами
Вхідні дані	Точні	Неповні, неточні
Несподівані вхідні дані	Не обробляються	Обробляються
Вихідні дані	Завжди коректні	Залежать від проблеми
Пояснення	Відсутнє	Зазвичай наявне
Застосування	Обробка чисел, текстових файлів	Символічні міркування
Виконання	Зазвичай послідовне	Пошук підходящих правил
Розробка програми	Структурований підхід	Неструктурований підхід

Основою систем, заснованих на знаннях, найчастіше є правила продукції, розділення знань та механізмів виведення та використання знань для експертизи (рис. 3.1).

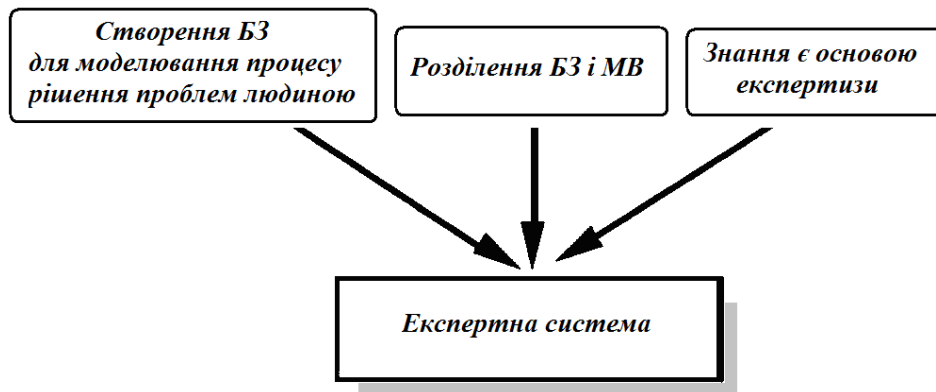


Рисунок 3.1 – Основні фактори впливу при створенні ЕС

Технологію розробки ЕС можна зобразити у вигляді схеми (рис. 3.2).

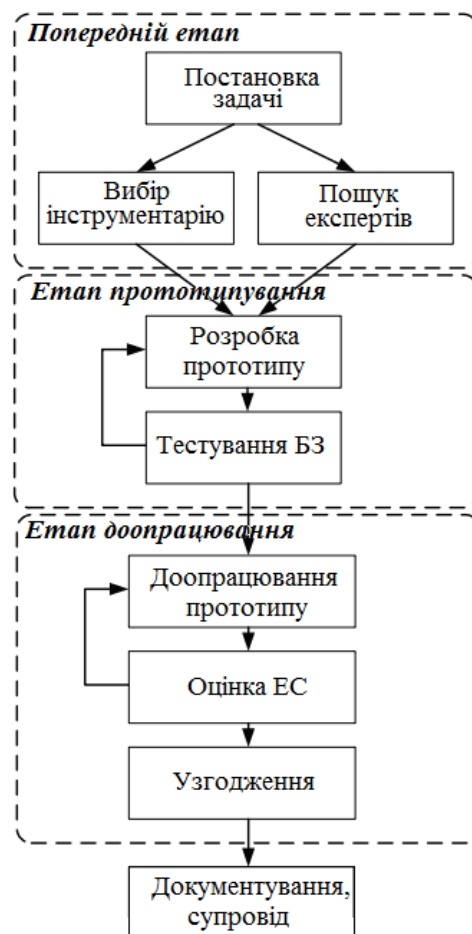


Рисунок 3.2 - Етапи розробки ЕС

Розробка ЕС складається з наступних етапів.

1. Попередній етап включає дії, що передують рішення про розробку нової ЕС. У рамках цього етапу проводиться конкретизація проблеми, вибір експертів у предметній області для співпраці, вибір відповідних інструментальних засобів. Основна особливість цього етапу полягає в тому, що може бути прийнято рішення про недоцільність розробки ЕС для рішення поставленого завдання.

2. Етап прототипування передбачає створення прототипу ЕС, призначеного для перевірки правильності обраних засобів та методів розробки нової ЕС. До прототипу системи не висуваються високі вимоги. Основне його завдання полягає в демонстрації можливостей майбутньої системи для фахівців, які безпосередньо беруть участь у розробці, а також для потенційних користувачів. На цьому етапі може бути здійснено коригування проекту, уточнено час, вартість та необхідні ресурси для завершення роботи.

3. Етап доопрацювання - основний, найбільш рутинний та тривалий етап роботи над ЕС. Усі компоненти багаторазово тестуються та доводяться до відповідності вимогам проекту. Найбільшу складність викликає доопрацювання та доказ адекватності та ефективності БЗ, оскільки кількість записів у ній може бути на порядок більшою, ніж у прототипі.

На практиці межа між етапами може бути розмитою, а сам процес проектування є досить неформальним, оскільки він пов'язаний з дослідженням і спробою копіювання діяльності людини. Велика кількість евристик, що застосовуються, інтуїтивний підхід до вирішення завдань експертами роблять процес створення ЕС творчим. Втім, формалізація технології ЕС, розробка в її рамках математичних методів та алгоритмів формування та опрацювання знань – це і є суть сучасної теорії ЕС. Ще однією особливістю розробки ЕС є її поетапне впровадження. Перші версії нової ЕС починають експлуатуватись в обмеженому обсязі вже на етапі прототипування.

Прототип, як перше наближення до нової ЕС, є індикатором успішності або невдачі попереднього етапу. У процесі постановки завдання найкритичнішою частиною розробки є правильний вибір проблеми. Якщо вибрати невідповідну проблему, можна дуже швидко ув'язнути у проектуванні завдань, які ніхто не знає, як вирішувати. Невідповідна проблема може також призвести до створення експертної системи, яка коштує набагато більше, ніж економить, і по суті нікому не потрібна. Справа буде ще гірше, якщо розробити систему, яка працює, але не прийнятна для користувачів через свою складність, низьку ефективність або інші причини. Якщо знання, необхідні для вирішення завдань у сфері застосування нової ЕС, чітко формулюються і пов'язані з обчислювальною обробкою, то звичайні алгоритмічні програми, ймовірно, будуть найдешевшим і найефективнішим способом вирішення проблем у цій галузі. Експертна система в жодному разі не усуває потреби у базах даних, статистичному програмному забезпеченні, електронних таблицях та системах текстової обробки. Але якщо результативність завдання залежить від суб'єктивного знання, що впливає частково з міркувань здорового глузду чи

інтуїції, тоді ефект від впровадження експертної системи у цій області може бути досить високим.

Необхідність створення ЕС зумовлена такими факторами:

- нестача висококваліфікованих фахівців у вузькій предметній галузі; приклад - система MYCIN була розроблена, щоб консультувати лікарів, що не спеціалізуються в галузі захворювань крові;
- потреба у великій кількості фахівців, якщо жоден з них не має достатнього об'єму знань; приклад - система ISIS допомагає планувати різні промислові замовлення;
- знижена продуктивність, оскільки завдання вимагає повного аналізу складного набору умов, а звичайний фахівець не в змозі переглянути (за відведений час) усі ці умови; приклад - система FALCON та ін. призначалися для контролю аварійних датчиків на хімічному заводі;
- велика розбіжність між рішеннями найдосвідченіших експертів та новачків; приклад - система XCON, розроблена для конфігурування комп'ютерів VAX – 1/780/ помилялася значно менше, порівняно з фахівцем середньої руки.

Для того, щоб знання, необхідні для вирішення поставленого завдання могли бути успішно представлені в базі знань, вони повинні бути досить вузькоспеціалізованими, не залежати значною мірою від загальнолюдських знань (знань про світ або про те, що для людини є само собою зрозумілим). Завдання, які вирішуються за допомогою цих знань, не повинні бути для людини-експерта ні надто легкими, що зробить застосування ЕС невиправданим, ні надто складними, оскільки формалізація необхідних знань може виявитися дуже трудомістким завданням. Також має існувати спосіб певним чином оцінити результати вирішення поставленого завдання, що дозволить судити про адекватність створюваної ЕС.

Спосіб, яким ЕС вирішуватиме завдання, залежить від експерта, знання якого будуть закладені в базу знань. Тому важливим етапом є вибір такого експерта. Деякі системи можуть реалізувати стратегії одного експерта, інші комбінувати підходи колективу фахівців. Отже, вибір відповідного експерта чи експертів - це один з найважливіших кроків у створенні експертної системи, оскільки визначає її авторитетність, що, зрештою, забезпечує комерційний успіх.

Реалізація прототипу можлива із застосуванням спеціалізованих інструментальних засобів. На ринку представлений досить великий їх спектр, починаючи від мов програмування (LISP, Prolog, Python та ін.), оболонок ЕС (Exsys та ін.) і закінчуючи дорогими комплексами (Gensym G2). Вибір інструментарію в основному залежить від наявних ресурсів та часу. Поширеним є також варіант, коли на ранніх стадіях для реалізації прототипу використовують інструмент, що дозволяє скоротити час розробки (наприклад, оболонку ЕС), а згодом для реалізації остаточної версії переходять на більш низькорівневу, але і більш ефективну мову програмування.

У процесі створення прототипу майбутньої ЕС виконується перевірка правильності кодування у БЗ фактів, зв'язків та стратегій міркування експерта. Також це хороший спосіб залучити експерта до активної участі в розробці експертної системи та зацікавити його, оскільки, як правило, авторитетні експерти не мають достатньо часу, щоб відволікатися від основної роботи для створення ЕС у повному обсязі.

Об'єм початкових версій БЗ прототипу зазвичай становить кілька десятків записів. Цього цілком достатньо, щоб змодельовати процес прийняття експертом окремих рішень з одного чи кількох питань. У той же час розробка цих кількох десятків правил вимагає проходження всього циклу проектування. Сама процедура створення прототипу має ітеративний характер. Кроки повторюються, доки не буде отримано задовільний результат.

Можна виділити основні стадії розробки прототипу: ідентифікація проблеми, вилучення знань, структурування та концептуалізація знань, формалізація знань, програмна реалізація системи, оцінка та тестування.

1. **Ідентифікація проблеми** - знайомство та навчання колективу розробників, а також створення неформального формулювання проблеми. Уточнюється завдання, планується хід розробки прототипу експертної системи, визначаються необхідні ресурси (час, люди, комп'ютери тощо), джерела знань (книги, додаткові експерти, методики), аналогічні експертні системи, цілі (поширення досвіду, автоматизація рутинних дій та ін.), класи розв'язуваних завдань тощо.

2. **Вилучення знань** – процес взаємодії експерта та інженера зі знань, що є основним розробником БЗ. В результаті цієї взаємодії інженер зі знань отримує найбільш повне уявлення про предметну область і способи вирішення завдань у ній. Відбувається перенесення компетентності експертів на інженерів із знань із використанням різних методів: аналіз текстів, діалоги, експертні ігри, лекції, дискусії, інтерв'ю, спостереження та інші.

3. **Структурування та концептуалізація знань** – розробка неформального опису знань про предметну область у вигляді графа, таблиці, діаграми або тексту, що відображає основні концепції та взаємозв'язки між поняттями предметної області. Виявляється структура отриманих знань предметної області, тобто, визначаються: термінологія, список основних понять та їх атрибутів, відношення між поняттями, структура вхідної та вихідної інформації, стратегія прийняття рішень, обмеження стратегій тощо. Такий опис називається *полем знань*.

4. **Формалізація знань** – розробка бази знань мовою, яка, з одного боку, відповідає структурі поля знань, з другого – дозволяє реалізувати прототип системи на наступній стадії програмної реалізації. Будується формалізоване подання концепцій предметної області з урахуванням вибраної мови подання знань (МПЗ). Традиційно на цьому етапі використовуються: логічна модель (обчислення предикатів першого порядку та інших), продукційна модель (з прямим і зворотним виведенням), модель семантичних мереж, модель фреймів, функціональне програмування, об'єктно-орієнтовані мови, засновані на ієрархії класів, об'єктів та інших, і навіть комбінація зазначених методів.

5. **Реалізація** – розробка комп'ютерної програми, яка демонструє життєздатність підходу загалом. Найчастіше перший прототип відкидається на етапі реалізації ЕС. Прототип експертної системи повинен включати базу знань і всі інші блоки, які можуть бути розроблені за допомогою будь-яких мов програмування, оболонок ЕС або спеціалізованих програмних комплексів.

6. **Тестування** – виявлення помилок у підході та реалізації прототипу та вироблення рекомендацій щодо подальшого доведення системи до промислового варіанту або повторної переробки. Оцінюється та перевіряється робота прототипу з метою приведення у відповідність до реальних запитів користувачів. Прототип перевіряється на зручність і адекватність інтерфейсів введення-виведення (характер питань у діалозі, зв'язність тексту результату та ін.), ефективність стратегії управління виведенням на знаннях (порядок перебору, використання нечіткого виведення та ін.), якість тестових прикладів, коректність бази знань (повнота і несуперечність правил).

Під час багаторазового проходження зазначених вище кроків прототип системи поступово вдосконалюється та розширюється. Таким чином відбувається плавний перехід від демонстраційного прототипу до комерційної системи. При цьому якщо програмний інструментарій вибраний вдало, відповідає необхідності переписування фінальної версії системи якоюсь мовою програмування або іншим, потужнішим інструментарієм.

У плавному процесі вдосконалення прототипу системи можна також виділити деякі стадії, межа між якими досить розмита, але в той же час мають певні характеристики:

- демонстраційний прототип ЕС – система вирішує частину завдань, демонструючи життєздатність підходу (кілька десятків записів БЗ);
- дослідницький прототип ЕС – система вирішує більшість завдань, але є нестійкою в роботі та не повністю перевіреною (кілька сотень записів БЗ);
- діючий прототип ЕС – система надійно вирішує всі завдання на реальних прикладах, але для складного завдання, можливо, знадобиться багато часу та пам'яті;
- робоча система – система забезпечує високу якість рішень при мінімізації необхідного часу та пам'яті, доопрацьовується з використанням більш ефективних інструментальних засобів;
- комерційна система – робоча система, придатна для продажу, тобто. добре документована та забезпечена необхідним сервісом для розповсюдження.

Основним завданням етапу створення діючого прототипу ЕС є додавання великої кількості додаткових евристик. Ці евристики зазвичай збільшують глибину системи, забезпечуючи більше правил для розуміння складних аспектів окремих випадків. У той же час експерт та інженер зі знань можуть розширити охоплення системи, включаючи правила, що керують додатковими підзавданнями або додатковими аспектами експертного завдання (метазнання). На цьому етапі розробки більшість експертів можуть вводити в систему нові

правила. Таким чином, починається процес, під час якого інженер зі знань передає право власності та контролю над системою експерту для уточнення, детальної розробки та обслуговування.

Після появи робочої експертної системи необхідно провести її тестування з погляду критеріїв ефективності та оцінити рівень її інтегрованості зі своїм оточенням (приладами, іншими програмними продуктами тощо). До тестування широко залучаються інші експерти для апробації працездатності системи на різних прикладах. Експертні системи оцінюються головним чином для того, щоб перевірити точність роботи програми та її корисність. Оцінку можна проводити, виходячи з критеріїв наступних категорій:

- критерії користувачів (зрозумілість та «прозорість» роботи системи, зручність інтерфейсів та ін.);
- критерії запрошених експертів (оцінка порад-рішень, пропонованих системою, порівняння їх з власними рішеннями, оцінка підсистеми пояснень та ін.);
- власні критерії колективу розробників (ефективність реалізації, продуктивність, час відгуку, дизайн, широта охоплення предметної області, несуперечність БЗ, кількість тупикових ситуацій, коли система не може прийняти рішення, аналіз чутливості програми до незначних змін у поданні знань, вагових коефіцієнтів, що застосовуються у механізмах логічного виведення, даних тощо).

Відлагоджену ЕС необхідно впровадити у її майбутнє оточення, тобто, виконати стикування модулів експертної системи з іншими програмними засобами, приладами та ін., що використовуються в середовищі, де функціонуватиме нова ЕС. Також необхідно забезпечити навчання користувачів системи. Іноді це означає внесення суттєвих змін, які потребують втручання розробників системи. Подібними змінами можуть бути, наприклад, зміна інтерфейсу користувача, реалізація низькорівневих засобів вводу/виводу та ін. Стикування передбачає також вдосконалення системних чинників, що залежать від часу, обчислювальних ресурсів і платформи функціонування ЕС, з метою забезпечення її більш ефективної роботи та поліпшення технічних характеристик, якщо система працює у неоднорідному середовищі (наприклад, зв'язок із вимірювальними пристроями).

Як приклад можна навести стикування системи PUFF зі своїм оточенням. Ця експертна система призначалася для діагностики захворювань легень. Після того, як PUFF була закінчена і всі були задоволені її роботою, систему перекодували з LISP на Бейсік. Потім систему перенесли на персональний комп'ютер, який працює у лікарні. У свою чергу, цей комп'ютер був пов'язаний із вимірювальними приладами. Дані з вимірювальних приладів відразу ж надходили до комп'ютера, PUFF обробляла ці дані та друкувала рекомендації для лікаря. Лікар у принципі не взаємодіє з PUFF, оскільки система повністю інтегрована зі своїм оточенням – вона була інтелектуальним розширенням апарату дослідження легень, який лікарі давно використовували.

Інший приклад стикування системи зі своїм оточенням – це система SAT-1 – експертна система для діагностики несправностей дизелів локомотивів. Ця система була розроблена також на LISP, а потім переведена на FORTH, щоб її можна було ефективно використовувати в різних локомотивних цехах. Майстер із ремонту вводив завдання у систему: визначити можливі причини несправності дизеля. Система, пов'язана з відеодиском, надавала майстру візуальні пояснення та підказки щодо докладних перевірок, які йому потрібно зробити. Крім того, якщо оператор не впевнений, як усунути несправність, система надавала йому навчальні матеріали, які фірма підготувала попередньо. Таким чином, майстер з ремонту за допомогою експертної системи міг діагностувати проблему, знайти потрібну тестову процедуру, отримати на дисплеї пояснення, як провести тест, або отримати інструкції про те, як впоратися з проблемою, що виникла.

При перекодуванні системи на мову, подібну до C, підвищується її швидкодія і збільшується переносимість, проте гнучкість при цьому зменшується. Це прийнятно лише в тому випадку, якщо система зберігає всі знання про проблемну область, і це знання не буде змінюватися в найближчому майбутньому. Однак, якщо експертна система створена саме через те, що проблемна область змінюється, необхідно підтримувати систему в інструментальному середовищі розробки. Вдалим прикладом ЕС, впровадженій в такий спосіб, є XCON (R1) – ЕС, яку фірма DEC використовувала для комплектації ЕОМ сімейства VAX. Одна з ключових проблем, з якою зіткнулася фірма DEC, – необхідність постійного внесення змін до БЗ для нових версій обладнання, нових специфікацій тощо. Для цієї мети XCON була стикована із середовищем проектування OPS5, до якого надходили всі необхідні матеріали про нове обладнання. Таким чином, нові дані автоматично потрапляли до БЗ ЕС, що дозволяло практично без додаткових витрат здійснювати оновлення БЗ.

3.2 Тип проблемного середовища

Поняття "проблемне середовище" охоплює предметну область (множину сутностей, що описують область експертизи, тобто множину об'єктів, значень їх характеристик і відношень між ними) і розв'язуваних в предметній області задач. Тобто, проблемне середовище включає сутності (структури даних) і розв'язувані з них задачі, що подаються у вигляді виконуваних тверджень (правил, процедур, формул тощо). Таким чином, проблемне середовище визначається характеристиками відповідної предметної області та характеристиками типів розв'язуваних у ній задач. Слід зауважити, що поряд із поняттям "проблемне середовище" використовується синонімічний йому термін "проблемна область".

Характеристики предметної області визначаються такими параметрами:

- 1) тип предметної області:

статична - вхідні дані не змінюються під час сеансу роботи програми, значення інших (не вхідних) даних змінюються лише ЕС;

динамічна - вхідні дані, які надходять від зовнішніх джерел, змінюються у часі, значення інших даних змінюються ЕС чи підсистемою моделювання зовнішнього оточення;

2) спосіб опису сутностей предметної області:

сукупність атрибутів та їхніх значень (фіксований склад сутностей);

сукупність класів (об'єктів) та їхніх екземплярів (змінний склад сутностей);

3) спосіб організації сутностей у БЗ:

неструктурована БЗ;

структурована БЗ - структурування сутностей БЗ по різних ієрархіях (найпоширеніші ієрархії: "загальне/приватне", "частина/ціле"), що забезпечує успадкування властивостей сутностей, що подаються в БЗ.

Характеристики типів задач:

1) тип вирішуваних задач:

задачі аналізу і (або) **синтезу**;

статичні або **динамічні** задачі;

2) частковість (загальність) виконуваних тверджень (правил, процедур, формул тощо):

часткові (спеціалізовані, конкретні) виконувані твердження;

загальні виконувані твердження.

Найбільш природним для людини способом опису сутностей предметної області є співвідношення з ними в пам'яті комп'ютера об'єктів, що складаються з атрибутів зі значеннями. Зазвичай вводиться опис об'єкта деякого типу, відповідно до якого створюються конкретні екземпляри цього типу. При цьому кількість екземплярів об'єкта не обмежується, тобто. склад сутностей, що представляються, при такому поданні проблемної області є змінним.

Однак для простих додатків при малій кількості об'єктів (від 1 до 5) іноді застосовують спрощене подання у вигляді атрибут/значення без вказівки на об'єкт, якому ці атрибути належать. Наслідком цього спрощення стало те, що реально існуючі об'єкти (сутності) предметної області стали представляти як фіксовану кількість розмножених імен відповідних атрибутів. Наприклад, замість посилання на атрибут i ($1 \times K$) об'єкта X ($1 \div N$) використовували посилання атрибут j ($1 (K \times N)$). Таким чином, замість K атрибутів вводили $K \times N$ атрибутів. Переваги цього підходу полягають у тому, що забезпечується пряме посилання атрибут (об'єкт).

Недоліки спрощеного подання у вигляді атрибут/значення:

1) неможливо використовувати "загальні" виконувані твердження (правила, процедури, функції тощо), що посилаються на довільну кількість сутностей; доводиться використовувати часткові (спеціалізовані) твердження, що збільшує їхню кількість; з іншого боку, за зміни складу сутностей у БЗ доводиться запроваджувати нові відповідні їм спеціалізовані твердження;

2) усунення об'єктів не дозволяє явно визначити природні взаємозв'язки між атрибутами одного об'єкта (всі атрибути є незалежними);

3) з усуненням об'єктів зникає можливість визначити ієрархію класів об'єктів.

Наступним параметром, що характеризує предметну область, є наявність (відсутність) **структурування БЗ**. Сенс структурування БЗ полягає в наступному:

1) обмежити коло сутностей, які мають розглядатися механізмом виведення, і таким чином скоротити перебір під час вибору рішення;

2) забезпечити **успадкування властивостей сутностей**, тобто, передачу властивостей вищих в ієрархії сутностей нижчим, що значно спрощує процес набуття та використання знань. Наприклад, загальні властивості класу "автомобіль" автоматично успадковуються всіма підкласами автомобілів та конкретними екземплярами цих підкласів. У подібній ієрархії успадковується відношення "є підмножиною (примірником)". Крім того, широко застосовується й інша ієрархія – "є частиною".

За типом розв'язуваних задач у першу чергу всі задачі доцільно поділити на задачі **аналізу** та **синтезу**.

У **задачі аналізу** задана модель сутності (об'єкта), і потрібно в результаті аналізу цієї моделі визначити деякі невідомі характеристики (функції) моделі.

У **задачі синтезу** задаються умови, яким мають задовольняти характеристики (функції) деякої "невідомої" моделі сутності, і потрібно побудувати модель цієї сутності. Розв'язання задачі синтезу є ітераційним процесом, що складається з наступних кроків:

- 1) створення дослідницької моделі сутності;
- 2) аналіз цієї моделі (тобто розв'язання задачі аналізу);
- 3) порівняння результатів аналізу з умовами задачі.

Таким чином, задача синтезу включає аналіз. Зазначимо, що у процесі створення конкретної ЕС, що вирішує задачу аналізу, розробник, створюючи БЗ (модель області експертизи), вирішує задачу синтезу, а побудована ЕС вирішуватиме задачу аналізу.

Задачі синтезу та аналізу можуть вирішуватися у статичних та динамічних областях. Якщо ЕС базується на припущенні, що вихідна інформація про предметну область (про навколишній світ), на основі якої вирішується задача, не змінюється за час вирішення задач, то говорять про **статичну предметну область** (точніше про статичне подання області в ЕС); якщо інформація про предметну область змінюється за час вирішення задач, то говорять про **динамічну предметну область**. При поданні динамічної області виникає задача моделювання навколишнього світу, зокрема моделювання активних агентів.

Якщо задачі, розв'язувані ЕС, явно не враховують фактор часу та (або) не змінюють у процесі свого вирішення дані (знання) про навколишній світ, то

кажуть, що ЕС вирішує *статичні задачі*; якщо задачі враховують фактор часу та (або) змінюють у процесі розв'язання дані про навколишній світ, то говорять про рішення *динамічних задач*.

Таким чином, ЕС працює у *статичному проблемному середовищі*, якщо вона використовує статичне подання та вирішує статичні задачі. ЕС працює у *динамічному проблемному середовищі*, якщо вона використовує динамічне подання та (або) вирішує динамічні задачі.

Враховуючи фактор часу у динамічних проблемних середовищах, багато фахівців називають їх додатками (ЕС), що працюють у реальному часі. Зазвичай виділяють такі системи реального часу: *"псевдореального" часу*, *"м'якого" реального часу* та *"жорсткого" реального часу*.

Системи "псевдореального" часу, як випливає з назви, не є системами реального часу, проте вони на відміну від статичних систем отримують і обробляють дані, що надходять від зовнішніх джерел. Системи псевдореального часу вирішують задачі швидше, ніж відбуваються значні зміни інформації про світ.

Системи "м'якого" реального часу працюють у тих додатках, де допустимий час реакції на події більше 0,1-1с. До них належать майже всі існуючі ЕС реального часу.

Системи "жорсткого" реального часу повинні забезпечувати час реакції швидше за 0,1-0,5с. Для досягнення такої швидкодії вони використовують не стандартні операційні системи (ОС) типу Unix та Windows NT, а спеціалізовані ОС та спеціалізовані бортові комп'ютери, що забезпечують швидкий час реакції.

Задачі, розв'язувані ЕС, розрізняють тим, як видаються твердження, що мають виконуватися. Використовуються як *часткові (спеціалізовані) твердження*, тобто, твердження, що містять посилання на конкретні сутності (об'єкти), так і *загальні твердження*, що застосовуються до будь-яких сутностей заданого типу (незалежно від числа та імені). Використання загальних тверджень дозволяє значно лаконічніше представляти знання. Однак оскільки загальні твердження не містять явних посилань на конкретні сутності, при їх використанні потрібно витратити значну роботу з визначення тих сутностей, до яких вони повинні застосовуватися, тобто, виконати, як стверджують фахівці, операцію зіставлення.

Не всі поєднання перерахованих вище властивостей, що характеризують проблемне середовище, зустрічаються на практиці. Виділимо кілька типів проблемних середовищ, що найбільш часто зустрічаються.

Тип 1. Статичне проблемне середовище: статична предметна область; сутності подаються як сукупність атрибутів та значень; склад сутностей незмінний; БЗ не структурована; вирішуються статичні задачі аналізу, використовуються лише часткові виконувані твердження.

Тип 2. Статичне проблемне середовище: статична предметна область; сутності подаються у вигляді атрибутів зі значеннями або породжених об'єктів (фреймів); склад сутностей незмінний; ієрархія БЗ або відсутня, або слабо

виражена (немає спадкування властивостей); вирішуються статичні задачі аналізу, використовуються часткові виконувачі твердження.

Тип 3. Статичне проблемне середовище: статична предметна область; сутності подаються у вигляді об'єктів; склад сутностей змінний; БЗ структурована; вирішуються статичні задачі аналізу і синтезу, використовуються загальні та часткові виконувачі твердження.

Тип 4. Динамічне проблемне середовище: динамічна предметна область; подаються у вигляді атрибутів та значень; склад сутностей незмінний; БЗ не структурована; вирішуються динамічні задачі аналізу, використовуються часткові виконувачі твердження.

Тип 5. Динамічне проблемне середовище: динамічна предметна область; подаються у вигляді об'єктів; склад сутностей змінний; БЗ структурована; вирішуються динамічні задачі аналізу і синтезу; використовуються загальні і часткові виконувачі твердження.

3.3 Класифікація засобів розробки ЕС

Інструментальні засоби (ІЗ) можна класифікувати за такими параметрами:

- рівень мови, що використовується;
- парадигма програмування;
- спосіб представлення знань;
- механізм виведення та моделювання;
- засоби набуття знань;
- технологія розробки системи.

Пропонована класифікація інструментальних засобів не є вичерпною, але, містить основні значущі параметри, за якими доцільно розрізняти інструментальні засоби.

3.3.1 Рівень мови, що використовується

Трудовитрати на розробку ЕС значною мірою залежить від використовуваних інструментальних засобів (ІЗ). У зв'язку з цим розглянемо класифікацію сучасних ІЗ за рівнем мови, що використовується.

1. Традиційні (зокрема об'єктно-орієнтовані) мови програмування типу С, С++ (зазвичай, ці ІЗ використовуються не для створення ЕС, а для створення ІЗ).

2. Символьні мови програмування (наприклад, Lisp, Prolog та їх різновиди). Ці ІЗ останнім часом, як правило, не використовуються в реальних додатках у зв'язку з тим, що вони погано пристосовані до поєднання програм, написаних мовами традиційного програмування.

3. Інструментарій, який містить багато, але не всі компоненти ЕС. Ці засоби призначені для розробника, від якого вимагається знання програмування

та вміння інтегрувати компоненти до програмного комплексу. Прикладами є такі засоби, як OPS 5, ИЛИС та ін.

4. Оболонки ЕС загального призначення, що містять усі програмні компоненти, але не мають знань про конкретні предметні середовища. Засоби цього та наступного типів не вимагають від розробника програми знання програмування. Прикладами є ЕКО, Leonardo, Nexpert Object, Карра та ін.

Останнім часом термін "оболонка" (shell) використовується рідше, його замінюють на ширший термін "середовище розробки" (development environment). Якщо хочуть підкреслити, що засіб використовується не тільки на стадії розробки програми, але й на стадіях використання та супроводу, то використовують термін "повне середовище" (complete environment). Прикладами таких засобів для створення статичних ЕС є: Nexpert Object, ProKappa, ART*Enterprise, Level 5 Object та ін.

5. Проблемно/предметно-орієнтовані оболонки (середовища):

- проблемно-орієнтовані засоби (problem-specific), орієнтовані на певний клас розв'язуваних задач; вони мають у своєму складі відповідні цьому класу альтернативні функціональні модулі (прикладами таких класів задач є задачі пошуку, управління, планування, прогнозування тощо);
- предметно-орієнтовані засоби (domain-specific), які включають знання про деякі типи предметних областей, що скорочує час розробки БЗ.

У наведеній класифікації інструментальні засоби наведені у порядку зменшення трудовитрат, необхідних для створення конкретної експертної системи. При використанні інструментарію першого, другого і третього типів до завдання розробника входить програмування всіх або частини компонентів ЕС мовою досить низького рівня. При застосуванні четвертого та п'ятого типів ІЗ розробник додатка повністю звільняється від робіт зі створення програм, його основні трудові витрати пов'язані з наповненням бази знань загальними та (або) специфічними знаннями.

При використанні інструментарію четвертого типу можуть виникнути такі проблеми:

1) керуючі стратегії, закладені в механізм виведення інструментарію, можуть не відповідати методам рішення, які використовує експерт, що взаємодіє з цією системою, що може призвести до неефективних, а можливо, і неправильних рішень;

2) мова подання знань, прийнята в інструментарії, може не підходити для конкретної системи.

Значна компенсація цих труднощів досягається застосуванням проблемно/предметно-орієнтованих засобів (ІЗ п'ятого типу).

3.3.2 Парадигми програмування (механізми реалізації виконуваних тверджень)

Способи реалізації механізму виконуваних тверджень часто називають *парадигмами програмування*, проте багато з цих парадигм не мають відношення до програмування у його загальноприйнятому значенні. До основних парадигм відносять:

- процедурне програмування;
- програмування, орієнтоване на дані;
- програмування, орієнтоване на правила;
- об'єктно-орієнтоване програмування.

Парадигма процедурного програмування найпоширеніша серед існуючих мов програмування (наприклад, C, Java). Тут явно виділяють два види різних сутностей:

1) процедури (функції), які виконують активну роль, тобто, вони задають поведінку (функціонування) програми;

2) дані, які виконують пасивну роль, тобто, вони є тим, що обробляється способом, заданим у функції; здатність скласти процедури (функції) з команд (операторів) та викликати їх є ключем даної парадигми; особливістю цієї парадигми є "побічні ефекти", що виникають у тих випадках, коли різні процедури (функції), що використовують загальні дані, незалежно змінюють ці дані.

У процедурній парадигмі активна роль організації поведінки відводиться процедурам, а не даним; причому процедура активізується викликом. Подібні способи задання поведінки зручні для описів детермінованої послідовності дій одного процесу, або декількох, але суворо взаємопов'язаних процесів.

У *парадигмі програмування, орієнтованій на дані*, активну роль відіграють дані, а не процедури. Тут із структурами активних даних пов'язують деякі дії (процедури), що активізуються тоді, коли здійснюється доступ до цих даних. Деякі фахівці називають цей спосіб активації дій "демонами". Програмування, орієнтоване на дані, дозволяє організувати поведінку мало залежних процесів, що важко реалізувати у процедурній парадигмі. Мала залежність процесів означає, що вони можуть розглядатися і програмуватися окремо. Однак при використанні парадигми, керованої даними, ці незалежно запрограмовані процеси можуть взаємодіяти між собою без їхньої зміни (тобто без перепрограмування).

У *парадигмі, орієнтованій на правила*, поведінка визначається множиною правил виду: умова $\rightarrow$ дія. Умова задає образ даних, у разі наявності яких дія правила може бути виконана. Правила у цій парадигмі грають таку ж роль, як і програми у процедурній парадигмі. Однак якщо в процедурній парадигмі поведінка задається детермінованою послідовністю операторів, яка не залежить для широкого класу оброблюваних даних від значень цих даних, то в парадигмі, орієнтованій на правила, поведінка (послідовність дій) задається

не задалегідь запропонованою послідовністю правил, а формується на основі значень тих даних, які зараз обробляються програмою (системою). Формування поведінки здійснюється за такою схемою. Умови правил порівнюються з поточними даними, і правила, умови яких задовольняються значеннями поточних даних, стають претендентами на виконання. Потім за певним критерієм здійснюється вибір одного правила серед претендентів та його виконання (тобто виконання дії, вказаної у правій частині правила). Правила-претенденти можуть виконуватись одночасно за наявності кількох процесів.

Підхід, орієнтований на правила, зручний для опису поведінки, що гнучко і різноманітно реагує на велику різноманітність станів даних.

Парадигма об'єктного програмування, на відміну процедурної парадигми, не поділяє програму на процедури і дані. Тут програма організується навколо сутностей, які називають об'єктами, які включають локальні процедури (методи) і локальні дані (змінні). Поведінка (функціонування) у цій парадигмі організується шляхом пересилання повідомлень між об'єктами. Об'єкт, отримавши повідомлення, здійснює його локальну інтерпретацію, ґрунтуючись на локальних процедурах та даних. Такий підхід дозволяє описувати складні системи найприроднішим чином. Особливо він зручний для вбудованих ЕС.

3.3.3 Спосіб представлення знань

Наявність різних способів представлення знань викликана насамперед прагненням з найбільшою ефективністю представити різні типи проблемних середовищ. Зазвичай спосіб представлення знань у ЕС характеризують моделлю представлення знань. До основних моделей представлення знань відносять: правила (продукції), фрейми або об'єкти, семантичні мережі, логічні моделі (числення предикатів) та ін. Нагадаємо, що ІЗ, що мають у своєму складі більше однієї моделі представлення знань, називають гібридними. Більшість сучасних ІЗ, як правило, використовуює об'єктно-орієнтовану парадигму, поєднану з парадигмою, орієнтованою на правила.

3.3.4 Механізми виведення та система моделювання

У статичній ЕС єдиним активним агентом, що змінює інформацію, є сама ЕС, та її механізми виведення (точніше механізм виконуваних тверджень, який обробляє не тільки правила, а й процедури, формули, функції тощо).

У динамічній ЕС зміна даних є наслідком:

- функціонування механізму виконуваних тверджень;
- змін, що відбуваються в навикишньому світі;
- функціонування системи моделювання зовнішнього середовища.

Розглянемо окремо параметри, що класифікують *ІЗ за механізмом виведення за системою моделювання*.

Стосовно статичних ІЗ механізм виведення можна характеризувати наведеними нижче параметрами.

1. Структура процесу отримання рішення:

- компіляція в режимі набуття знань з дерева виведення з навчальної вибірки (індуктивні методи набуття знань); вибір рішення з дерева виведення у режимі розв'язання задачі;
- компіляція у режимі набуття знань з дерева виведення зі специфічних правил; пошук рішення на дереві виведення у режимі розв'язання задачі;
- генерація дерева виведення та пошук рішення в режимі розв'язання задачі; генерація дерева виведення здійснюється під час виконання операції зіставлення, визначальної пари: правило n_i – сукупність даних, у яких умови цього правила задовольняються;
- у режимі рішення задач ЕС здійснює вироблення правдоподібних припущень (за відсутності достатньої інформації для вирішення); виконання міркувань щодо обґрунтування (спростування) припущень; генерацію альтернативних дерев виведення; пошук рішення у деревах виведення.

2. Пошук (вибір) рішення:

- спрямованість пошуку: від даних до мети, від мети до даних, двонаправлений пошук;
- порядок перебору вершин у дереві виведення: "пошук у ширину" - спочатку розкриваються (обробляються) всі вершини, безпосередньо пов'язані з поточною оброблюваною вершиною T , "пошук у глибину" - спочатку розкривається одна найбільш значуща вершина - $D1$, пов'язана з поточною T , потім вершина $D1$ робиться поточною, і розкривається одна найбільш значима вершина $D2$ тощо.

3. Процес генерації припущень та мережі виведення:

- режим генерації мережі виведення: генерація у режимі придбання знань, генерація у режимі розв'язання задачі;
- повнота мережі виведення, що генерується: операція зіставлення застосовується до всіх правил і всіх типів зазначених у правилах сутностей на кожному циклі роботи механізму виведення (забезпечується повнота мережі, що генерується); використовуються різні засоби для скорочення кількості правил та (або) сутностей, що беруть участь в операції зіставлення; наприклад, застосовується алгоритм зіставлення Rete або використовуються метазнання типу дій *focus* та *invoke*.

Механізм виведення для динамічних проблемних середовищ додатково містить:

- планувальник, який забезпечує всю діяльність ЕС за пріоритетами;

- засоби, що гарантують отримання кращого рішення за умов обмеженості ресурсів;
- систему підтримки істинності значень змінних, що змінюються у часі.

Динамічні ІЗ стосовно системи моделювання можуть бути охарактеризовані наступним чином:

- 1) система моделювання відсутня;
- 2) існує система моделювання загального призначення, що є частиною ІЗ;
- 3) існує спеціалізована система моделювання, що є зовнішньою стосовно ІЗ, у якому реалізується ЕС.

Системи моделювання загального призначення використовують для опису моделі алгебраїчних, різницевих і диференціальних рівнянь (зазвичай першого порядку). Як правило, до системи моделювання включається кілька альтернативних методів. Наприклад, на вибір користувача рівняння вирішується або методом Рунге-Кутта, що забезпечує кращу точність, або методом Ейлера, що забезпечує меншу точність, але дає більш швидке отримання рішення. Спеціалізовані системи моделювання враховують специфіку застосування (наприклад, модель хімічного підприємства).

3.3.5 Засоби набуття знань

ІЗ щодо набуття знань можуть бути охарактеризовані наступним набором параметрів.

1. Рівень мови, за допомогою якої здійснюється набуття знань:
 - формалізована мова;
 - обмежена природна мова (ПМ);
 - мова піктограм та зображень;
 - ПМ та мова зображень.
2. Тип знань, що набувають:
 - дані у вигляді таблиць, що містять значення вхідних та вихідних атрибутів (за якими індуктивними методами будується дерево виведення);
 - спеціалізовані правила;
 - загальні та спеціалізовані правила.
3. Тип даних:
 - атрибути зі значеннями;
 - об'єкти;
 - класи структурованих об'єктів та їх екземпляри, які отримують значення атрибутів при спадкуванні.

3.4 Підходи до розробки ЕС

Можна виділити принаймні чотири підходи до розробки ЕС:

- 1) підхід, що базується на поверхневих знаннях;
- 2) структурний підхід;
- 3) підхід, що базується на глибинних знаннях;
- 4) змішаний підхід, що базується на використанні поверхневих та глибинних знань.

Підхід, що базується на поверхневих знаннях, застосовується до складних задач, які не можуть бути точно описані. Цей підхід полягає в отриманні від експерта фрагментів знань (часто евристичних), які є релевантними розв'язуваній задачі. При цьому не робиться жодних спроб систематичного чи глибинного вивчення області, що зумовлює використання пошуку у просторі станів як універсального механізму виведення. Зазвичай в ЕС, які використовують даний підхід, як спосіб подання знань вибираються правила. Умова кожного правила визначає зразок певної ситуації, за дотримання якої правило може бути виконане. Пошук рішення полягає у виконанні тих правил, зразки яких зіставляються з поточними даними. У цьому передбачається, що у пошуку рішення послідовність ситуацій, що формуються таким чином, не обірветься до отримання рішення, тобто. не виникне невідомої ситуації, яка не зіставиться з жодним правилом. Даний підхід з успіхом застосовується до широкого класу додатків, проте він виявляється неефективним у тих додатках, коли задача може бути заздалегідь структурованою або при рішенні задачі може бути використана деяка модель.

Структурний підхід до побудови ЕС обумовлений тим, що для ряду додатків застосування тільки техніки поверхневих знань не забезпечує рішення задачі. Використання пошуку як механізму виведення в неструктурованій базі знань може призводити до ненадійних і (або) неякісних рішень. Структурний підхід до побудови ЕС подібний до структурного програмування. Однак стосовно ЕС не йдеться про те, що структурування має довести задачу до алгоритму (як у традиційному програмуванні), а передбачається, що частина завдання вирішується за допомогою пошуку. Структурний підхід у різних додатках доцільно поєднувати з поверхневим чи глибинним.

У **глибинному підході** компетентність ЕС базується на моделі того проблемного середовища, в якому ця ЕС працює. Модель може бути визначена у різний спосіб (декларативно, процедурно). Необхідність ряду додатків використовувати моделі викликана прагненням виправити недосконалість поверхневого підходу, що виникає за відсутності правил, що задовольняють поточної ситуації у робочій пам'яті. Глибинні ЕС мають властивості поверхневих ЕС. Крім того, у разі виникнення невідомої ситуації вони мають здатність визначити, які дії слід виконати. Ці дії ЕС визначають за допомогою деяких загальних принципів, справедливих для області експертизи.

Глибинний (модельний) підхід потребує явного опису структури та відношень між різними сутностями області. При цьому підході необхідно

використовувати ІЗ, які мають потужні моделюючі можливості: об'єкти з приєднаними процедурами, ієрархічне успадкування властивостей, активні знання (програмування, кероване даними), передача повідомлень об'єктам (об'єктно-орієнтоване програмування) тощо.

Змішаний підхід у загальному випадку може поєднувати поверхневий, структурний і глибинний підходи. Наприклад, поверхневий підхід може бути використаний для пошуку адекватних знань, які потім використовуються певною глибинною моделлю.

Відповідно до параметрів проблемних середовищ можна виділити п'ять типів ІЗ, що відповідають типам проблемних середовищ.

Тип 1. ІЗ застосовується для створення ЕС у статичних проблемних середовищах типу 1. ІЗ, зазвичай, є оболонкою (середовищем) і характеризуються такими особливостями: сутності подають як сукупність атрибутів і значень, БЗ не структурована, ІЗ орієнтований на рішення статичних завдань аналізу. Використовується програмування, орієнтоване на правила. Правила є конкретними. Правила та дані вводяться експертом або створюються автоматично з даних індуктивними методами. Мережа або дерево виведення компілюється в режимі набуття знань. Напрямок пошуку рішення: від мети до даних. Зіставлення не використовується. Під час розробки ЕС використовується технологія, що базується на поверхневих знаннях. Приклади ІЗ: 1-st class, ИЛИС та ін.

Тип 2. ІЗ застосовується до створення ЕС у статичних середовищах типу 2. ІЗ, зазвичай, є оболонкою (середовищем) з такими особливостями: сутності представляються як атрибути чи об'єкти (фрейми), структурування БЗ слабо виражено, орієнтація на вирішення статичних задач аналізу. Використовуються програмування, орієнтоване на правила, і частково об'єктно-орієнтоване програмування. Правила зазвичай конкретні, але у ряді ІЗ цього типу використовуються загальні правила. Правила та структури даних вводяться експертом. У ІЗ цього та наступних типів індуктивні методи зазвичай не застосовуються, оскільки за їх допомогою можна отримувати лише прості правила. Процес отримання рішення зазвичай полягає у генерації мережі виведення у режимі набуття знань. Напрямок пошуку рішення: від мети до даних. Зіставлення зазвичай не використовується. При розробці ЕС використовуються поверхневий та структурний підходи. Приклади ІЗ цього типу: ЕКО, Guru, Leonardo та ін.

Тип 3. ІЗ застосовується до створення ЕС у статичних середовищах типу 3. ІЗ, зазвичай, є оболонкою (середовищем) з такими особливостями: сутності представляються як об'єкти, БЗ структурована, орієнтація на статичні задачі аналізу та синтезу. Використовується програмування, орієнтоване на правила, об'єктно-орієнтоване та процедурне програмування. Правила конкретні та загальні, тобто, використовується операція зіставлення. Процес отримання рішення включає: генерацію мережі виведення та пошуку або вироблення

припущень з генерацією мережі виведення та пошуку. Напрямок пошуку рішення на вибір: від мети до даних і від даних до мети. Використовується (на вибір) пошук в ширину або в глибину. При розробці ЕС використовуються структурний, поверхневий та глибинний підходи. Дані та правила вводяться на обмеженій природній мові з використанням зображень. Приклади ІЗ цього типу: Nexpert Object, ART, Level 5 Object, ProKappa, ADS.

Тип 4. ІЗ застосовується до створення ЕС у динамічних середовищах типу 4. ІЗ є інструментарієм, тобто, сукупністю компонентів, з яких програмується ЕС. Сутності подаються як атрибути і значення, БЗ не структурована, вирішуються динамічні задачі аналізу у реальному часі, правила є конкретними. Використовується як пошук від мети, так і від даних. Зіставлення не використовується. Система моделювання відсутня. При розробці ЕС використовуються поверхневий та структурний підходи. Приклади ІЗ цього типу: TDC Expert, Activation Frame Work, Rocky.

Тип 5. ІЗ застосовується для створення ЕС у динамічних середовищах типу 5. ІЗ, зазвичай, є оболонкою (середовищем) з такими особливостями: сутності представляються класами об'єктів та екземплярами, БЗ структурована, вирішуються динамічні задачі аналізу та синтезу. Використовується програмування, орієнтоване на правила, об'єктно-орієнтоване програмування та процедурне програмування. Правила є конкретними та загальними. Процес отримання рішення складається з генерації мережі виведення та пошуку. Напрямок пошуку задається користувачем (від даних чи від мети). Використовується пошук у глибину та ширину. Операція зіставлення спрямовується метазнаннями шляхом фокусування на групах правил та (або) класі об'єктів (або на конкретному об'єкті). Використовується система моделювання. Планувальник забезпечує отримання найкращого рішення в умовах обмежених ресурсів. Система підтримки істинності забезпечує отримання нових значень змінних. При розробці ЕС використовуються поверхневий, структурний та глибинний підходи. Приклади ІЗ цього типу: G2, RTworks.

Контрольні запитання

- 1 Чим характеризується процес розробки ЕС?
- 2 Прокоментуйте різницю між звичайними програмами та експертними системами.
- 3 Поясніть основні фактори, що враховуються під час створення ЕС.
- 4 Поясніть етапи технології розробки ЕС.
- 5 Назвіть потрібні підстави для створення ЕС.
- 6 Навіщо необхідно реалізувати прототип ЕС?
- 7 Назвіть та охарактеризуйте основні стадії розробки прототипу ЕС.

- 8 Якими параметрами визначаються характеристики предметної області?
- 9 У чому суть задач аналізу та синтезу, які можуть вирішувати ЕС?
- 10 У чому суть динамічних та статичних задач, які можуть вирішувати ЕС?
- 11 Назвіть та охарактеризуйте типи проблемних областей, що найбільш часто зустрічаються.
- 12 Як можна класифікувати інструментальні засоби розробки ЕС?
- 13 Які мови програмування застосовуються для розробки ЕС?
- 14 Поясніть термін "Парадигми програмування".
- 15 Охарактеризуйте основні парадигми програмування ЕС.
- 16 Назвіть механізми виведення та моделювання, які використовуються при розробці ЕС.
- 17 Охарактеризуйте засоби придбання знань.
- 18 Назвіть підходи щодо розробки ЕС.
- 19 Поясніть підхід щодо розробки ЕС, що базується на поверхневих знаннях.
- 20 Поясніть структурний підхід щодо розробки ЕС.
- 21 Поясніть глибинний підхід щодо розробки ЕС.
- 22 Які характеристики мають п'ять типів інструментальних засобів розробки ЕС?

4 БАЗИ ЗНАНЬ ЕКСПЕРТНИХ СИСТЕМ

Будь-яка предметна область може бути описана у вигляді деякої сукупності відомостей про структуру цієї області, основні її характеристики, процеси, що протікають в ній, а також про способи вирішення завдань, що виникають у ній. Всі ці відомості утворюють знання про предметну область. При використанні ЕС для вирішення завдань у певній предметній області необхідно зібрати про неї відомості та створити концептуальну модель цієї області. Джерелами знань можуть бути документи, статті, книги, фотографії, кінозйомка та багато іншого. З цих джерел аналітик має отримати знання, що містяться в них. Цей процес може бути досить важким, бо треба заздалегідь оцінити важливість тих чи інших знань для роботи ЕС. Фахівці, які займаються всіма питаннями, пов'язаними зі знаннями, називаються інженерами зі знань чи когнітологами.

Напрямок штучного інтелекту «Інженерія знань» займається питаннями проектування баз знань, тобто, проблемами опрацювання знань.

Обробка знань включає:

- вилучення знань із джерел (під джерелами розуміються матеріальні засоби зберігання знань, а також події та явища, але при цьому вважається, що людина джерелом не є);
- набуття (отримання) знань від професіоналів (експертів);
- представлення знань, тобто, їх формалізація, що дозволяє надалі використовувати знання для логічного виведення експертною системою;
- маніпулювання знаннями, що включає поповнення, класифікацію, узагальнення знань та виведення на знаннях;
- пояснення на знаннях, що дозволяє дати відповідь, як і чому отримано той чи інший висновок.

Інженерія знань – напрямок ШІ, що займається проблемами набуття, представлення та управління знаннями. Процес організації баз знань на основі роботи з полем знань передбачає отримання та структурування знань фахівців у певній предметній області (ПрО).

Поле знань включає опис основних понять та відношень між поняттями предметної області, знань експертів, які подають у вигляді графа, таблиці, тексту.

4.1 Мова опису поля знань

Інженерія знань має справу з розмитими межами. Для опису знань поки не вистачає математичного апарату та потрібних засобів для наочного та компактного представлення моделі предметної області. Поле знань Pz

використовується для структурування знань і являє собою модель знань про предметну область, представлену деякою мовою представлення знань L . Згодом модель знань повинна бути реалізована у вигляді комп'ютерної програми. В даний час в інженерії знань віддають перевагу семіотичному моделюванню.

Семіотика, або семіологія, - наука, що досліджує властивості знаків і знакових систем, наука про комунікативні системи та знаки, що використовуються в процесі спілкування.

Семіотика включає в себе (рис. 4.1):

- синтаксис – сукупність правил побудови мови (відношень між знаками);
- семантику – зв'язок між елементами мови та їх значеннями або відношеннями між знаками та реальністю;
- прагматику – відношення між знаками і користувачами.

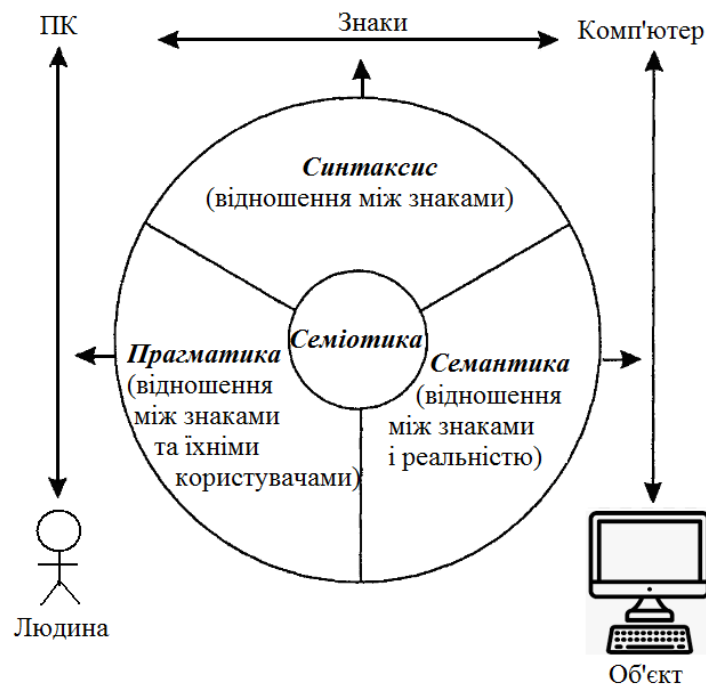


Рисунок 4.1 – Структура семіотики

4.2 Семіотична модель поля знань

Поле знань Pz може бути представлене як семіотична модель у вигляді графа, малюнка, таблиці, діаграми чи тексту. На вибір структури, форми та змісту Pz впливають особливості предметної області.

Синтаксис

Узагальнену синтаксичну структуру поля знань (рис. 4.2) можна як

$$Pz = (I, O, M) ,$$

де I – структура вхідних даних, які мають бути оброблені та інтерпретовані;
 O – структура вихідних даних;
 $M = (Sk, Sf)$ – операціональна модель предметної області, модель міркувань експерта, на основі котрої I перетворюється в O ;
 $Sk = (A, R)$ – концептуальна структура предметної області, A – множина об'єктів предметної області, R – множина відношень між ними;
 Sf – функціональна структура предметної області, тобто, модель міркувань та прийняття рішень.

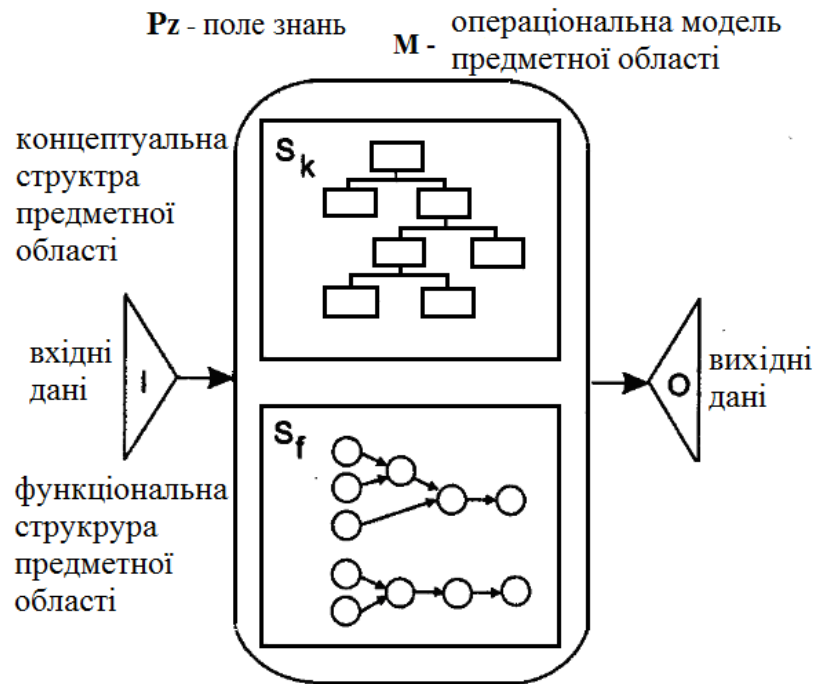


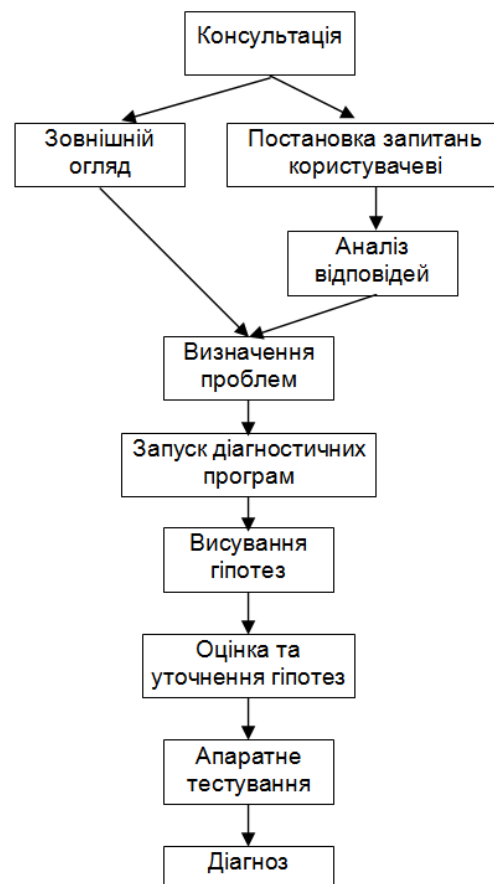
Рисунок 4.2 – Структура поля знань

Концептуальна структура Sk предметної області є статичною складовою, а функціональна структура Sf – динамічною і стратегічною.

Концептуальна (змістовна) структура предметної області – це абстрактна модель, що визначає структуру модельованої системи, властивості її елементів і причинно-наслідкові зв'язки, властиві системі і суттєві для досягнення мети моделювання.

Приклад Sk і Sf з області діагностики несправностей персонального комп'ютера наведено на (рис. 4.3, 4.4).

Структура Sf включає поняття предметної області та моделює основні функціональні зв'язки RA (відношення між поняттями з структури Sk). Зв'язки відображають модель (стратегію) прийняття рішення у вибраній предметній області.

Рисунок 4.3 - Концептуальна структура *Sk* предметної областіРисунок 4.4 - Функціональна структура *Sf* предметної області

Семантика

Семантика мови L залежить від особливостей PrO .

Семантику поля знань Pz можна розглядати на двох рівнях:

- 1 – P_{zg}^i – семантична модель знань експерта i про деяку предметну область Og ;
- 2 – модель деяких знань (поле знань Pz), яка є образом дійсності (рис. 4.5).

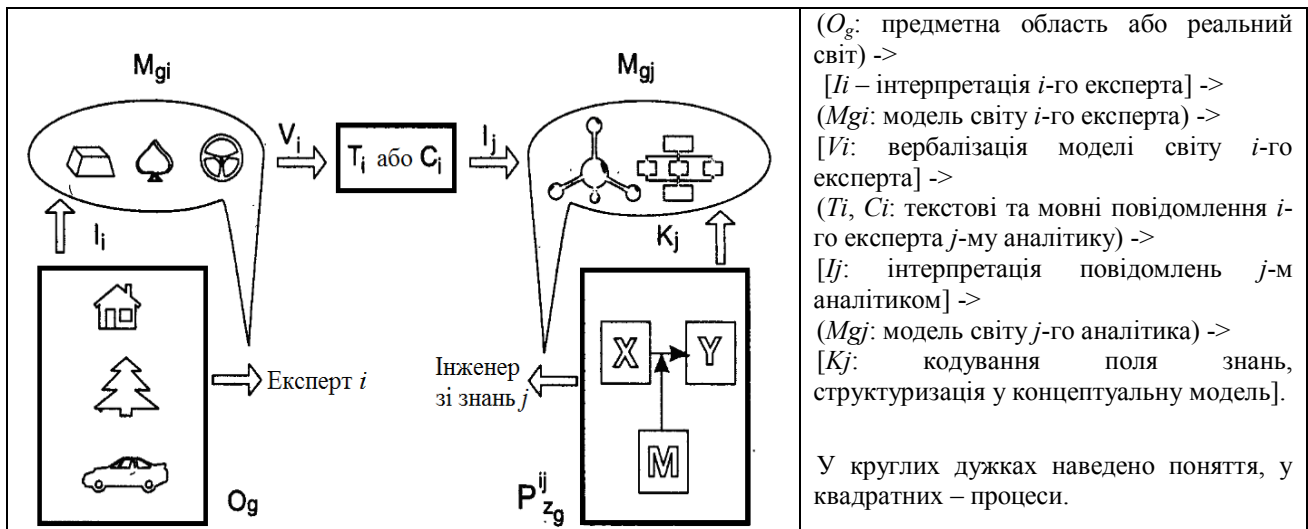


Рисунок 4.5 – Трансляція знань при формуванні поля знань

Поле знань P_{zg}^{ij} – результат, отриманий інженером зі знань.

Під час передачі від експерта до інженера зі знань знання зазнали 4 трансляції:

1 – I_i – сприйняття та інтерпретація дійсності Og предметної області g i -м експертом; у пам'яті експерта створюється модель M_{gi} як семантичне представлення дійсності та його особистого досвіду роботи з нею;

2 – V_i – вербалізація досвіду i -го експерта, який пояснює свої міркування S_i при передачі своїх знань Z_i інженеру зі знань; у пам'яті інженера створюється або текст T_i , або мовне повідомлення C_i ;

3 – I_j – сприйняття та інтерпретація T_i або C_i j -м інженером зі знань, в пам'яті котрого створюється модель світу M_{gj} ;

4 – K_j – кодування і вербалізація моделі M_{gj} у формі поля знань P_{zg}^{ij} .

Інженер зі знань повинен добитися максимальної відповідності між M_{gi} і P_{zg}^{ij} .

Приклад впливу суб'єктивних поглядів експертів на M_{gi} та V_i .

Реальність Og : два чоловіки прибігають на вокзал за 2 хвилини до відправлення поїзда; до кас черга. Біля кас-автоматів черги немає, але у жодного з двох немає дрібних грошей. Обидва запізнюються на важливу зустріч. Сприйняття і оцінка ситуації експертами:

	1-й експерт	2-й експерт
Інтерпретація <i>I</i>	Неможна приходити на вокзал менш ніж за 10хв.	Треба завжди мати дріб'язок.
Вербалізація <i>V</i>	Запізнився, тому що не розрахував час.	Запізнився, тому що на вокзалі безлад, у касах натовп.

Наступні трансляції ще більше спотворюватимуть модель вже на основі суб'єктивного сприйняття аналітиків.

Прагматика

У семіотичній моделі використовуються технології проведення структурного аналізу ПрО. З її допомогою інженер зі знань може сформувати *Pz* за результатами добування знань.

Прагматика – практичні аспекти розробки та використання поля знань.

Піраміда знань

Знання можна представляти як ієрархію чи піраміду, де кожен наступний рівень служить для узагальнення і поглиблення знань у ПрО. Таким чином, понятійну структуру *Sk* можна вважати багаторівневою. Відповідно і стратегії прийняття рішень (функціональні структури *Sf*) на різних рівнях суттєво відрізняються.

4.3 Стратегії отримання знань

Для означення процесу отримання знань використовують також терміни "придбання (acquisition), видобуток, вилучення, виявлення (elicitation), формування". Суть процесу видобутку знань полягає у перенесенні компетентності експертів на інженерів зі знань.

Можна виділити основні стратегії отримання знань (рис. 4.6):

- 1) вилучення знань без використання комп'ютерів, шляхом безпосереднього контакту інженера зі знань та джерела знань (експерта, літератури);
- 2) вилучення знань з використанням програмних засобів;
- 3) формування знань з використанням програм навчання за наявності репрезентативної вибірки прикладів прийняття рішень у предметній області.

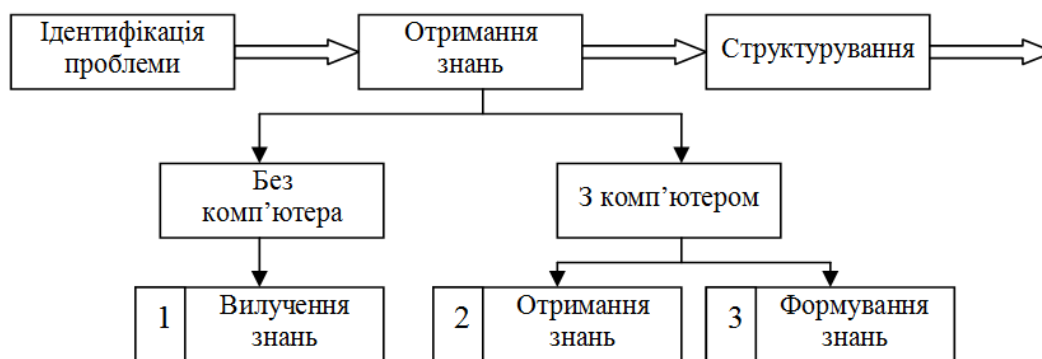


Рисунок 4.6 – Стратегії отримання знань

Вилучення знань (*knowledge elicitation*) – це процедура взаємодії інженера з джерелом знань, в результаті якої стають явними процес міркування фахівців при прийнятті рішення та структура їх уявлень про предметну область.

При вилученні знань виконується складання структури знань з потоку інформації. Вилучення знань – трудомісткий процес, що спирається на когнітивну психологію, системний аналіз, математичну логіку та інші науки.

Проблеми вилучення знань:

- організаційні проблеми;
- невдалий метод, що не збігається зі структурою знань у цій області;
- неадекватна модель представлення знань;
- невміння налагодити контакт з експертом;
- термінологічний різнобій;
- вилучення фрагментарних знань та відсутність цілісної системи;
- спрощення «картини світу» експерта та ін.

Часто експерт, знаючи що $A \rightarrow B$, не усвідомлює, що його ланцюжок міркувань був набагато довшим, тобто, $A \rightarrow D \rightarrow C \rightarrow B$.

У діалозі інженера зі знань та експерта виявляються експертні знання, які часто мають невербальний характер і виражені у вигляді образів. У процесі пояснення експерт пояснює словами асоціативні образи, тобто, вербалізує знання.

Експерту важко створити модель предметної області через глибину та обсяг інформації, якою він володіє. Об'єкти реального світу пов'язані більш ніж 200 типами відношень та утворюють велику і складну систему. Для виділення головної структури системи інженеру зі знань необхідно застосовувати системний аналіз та системну методологію.

Отримання знань (*knowledge acquisition*) – процес наповнення БЗ експертом із використанням спеціалізованих програмних засобів.

Набуття знань може виконуватися або як передача знань від експерта до БЗ, або як спосіб автоматизованої побудови БЗ через діалог експерта та спеціальної програми із заздалегідь закладеною структурою поля знань.

При набутті знань експерт спілкується з автоматизованою системою, в яку розробниками закладено структури знань, відповідно до яких система

набуває фрагментів знань. У цьому процесі вся конкретна предметна область жорстко обмежена і модель представлення знань задана.

Формування знань (*machine learning*) – процес аналізу даних та виявлення прихованих закономірностей з використанням спеціального математичного апарату та програмних засобів.

Формування знань – галузь інженерії знань, що займається розробкою моделей, методів та алгоритмів навчання. Вона включає індуктивні моделі формування знань та автоматичного породження гіпотез на основі навчальних вибірок, навчання за аналогією та інші методи. Ці моделі дозволяють виявити причинно-наслідкові емпіричні залежності БД з неповною інформацією, що містить структуровані числові і символічні об'єкти.

До завдань формування знань (машинного навчання) належать завдання прогнозування, ідентифікації (синтезу) функцій, розшифрування мов, індуктивного виведення, синтезу з додатковою інформацією, розпізнавання образів.

Інша стратегія отримання знань - навчання з прикладів (*case-based reasoning*), у якій джерелом знань є множина прикладів предметної області.

Навчання на прикладах тісно пов'язане з машинним навчанням. Різниця в тому, що результат навчання повинен інтерпретуватися в деякій моделі, в якій, можливо, вже містяться факти та закономірності предметної області. Результат має бути перетворений на спосіб подання, який може бути використаний для навчання БЗ, моделювання міркувань, роботи компонента пояснення. Результат навчання стає компонентом відповідної технології.

Компонентами машинного навчання є *data mining* та *knowledge discovery*, що базуються на аналізі даних та пошуку закономірностей.

4.4 Теоретичні аспекти вилучення знань

Процес вилучення знань має свої особливості, пов'язані з психологією, лінгвістикою та гносеологією. Аспекти отримання знань можна представити як

$$A = \{A1, A2, A3\},$$

де $A1$ – психологічний аспект;
 $A2$ – лінгвістичний аспект;
 $A3$ - гносеологічний аспект.

4.4.1 Психологічний аспект вилучення знань

Психологічний аспект є провідним і визначає успішність та ефективність взаємодії інженера зі знань (аналітика) та джерела знань (експерта).

Процес безпосереднього спілкування є складною та багатоплановою процедурою (рис. 4.7), яка передбачає, що потік інформації йде не тільки від

експерта до інженера зі знань, а й від інженера зі знань до експерта, інформація циркулює між ними.

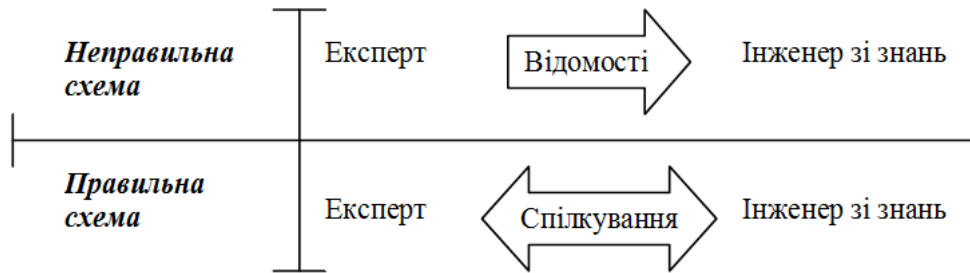


Рисунок 4.7 – Структура процесу спілкування

Можна виділити такі основні рівні спілкування.

1. Маніпулювання – один суб'єкт розглядає іншого як засіб чи перешкоду стосовно своєї діяльності.
2. Рефлексивна гра – суб'єкт враховує контрпроект іншого суб'єкта, але не визнає його самоцінність і прагне виграшу у реалізації свого проекту.
3. Правове спілкування – суб'єкти визнають право існування проектів діяльності одне одного і намагаються досягти хоча б зовнішнього узгодження.
4. Моральне спілкування – суб'єкти внутрішньо приймають спільний проект взаємної діяльності.

Професійним вважається спілкування інженера зі знань на 4-му рівні.

За будь-якого розмовного спілкування втрати інформації великі (рис. 4.8).



Рисунок 4.8 – Втрати інформації при розмовному спілкуванні

Інформативність спілкування аналітика та експерта може бути підвищена за рахунок використання психологічних знань.

Структурними компонентами моделі А1 спілкування при отриманні знань є відповідні шари проблем:

- учасники спілкування (партнери) – контактний шар S11;

- засоби спілкування (процедура) – процедурний шар S12;
- предмет спілкування (знання) – когнітивний шар S13.

Контактний шар S11.

У спілкуванні не можна гарантувати 100% психологічну сумісність учасників, але можна виділити ряд факторів, що впливають на ефективність процедури спілкування.

Параметри партнерів, що впливають на ефективність процедури спілкування під час вилучення знань, – стать, вік, особистість, темперамент, мотивація, вони формують модель суб'єкта спілкування (табл. 4.1).

Таблиця 4.1 – Параметри суб'єктів спілкування

Озна-чення	Назва параметра	Хороші результати
S11_1	стать	Гетерогенні суб'єкти (чоловік-жінка), не критично
S11_2	вік	Співвідношення віку експерта та аналітика $5 < (B_e - B_a) < 20$, не критично
S11_3	особистість	Доброзичливість, аналітичність, хороша пам'ять, уважність, спостережливість, уява, вразливість, зібраність, наполегливість, товариськість, винахідливість
S11_4	темперамент	Сангвініки й холерики – найкращий варіант. 3 флегматиками та меланхоліками – повільний темп спілкування. Холериків (поверхневе сприйняття інформації) – наводити на міркування і рефлексію. Меланхоліків - підбадьорювати
S11_5	мотивація	Стимулювання суб'єкта – матеріальне, збудження духу суперництва, конкуренції

Процедурний шар S12. Параметри шару S12 впливають на процес проведення процедури вилучення знань (табл. 4.2). Аналітик повинен мати спеціальні професійні знання для успішного отримання знань.

Таблиця 4.2 – Параметри процесу проведення процедури вилучення знань

Озна-чення	Назва параметра	Хороші результати
S12_1	Ситуація спілкування	
S12_1_1	місце	Невелике приміщення, наодинці, не на робочому місці експерта. Оптимальна дистанція – 1.2-3м, мінімальна дистанція – 0.7-0.8м.
S12_1_2	тривалість	≥ 1.5 -2 години з паузами через 20-25 хв.
S12_1_3	час	У першій половині дня для «жайворонків», у другій – для «сов»
S12_2	Обладнання	
S12_2_1	допоміжні засоби	Наочний матеріал – рисунки, графіки, діаграми, схеми (для суб'єктів художнього типу); формули, текстова інформація (для людей розумового типу). Протоколювання – стенографування розмови (недолік – низький темп, внесення суб'єктивності); звуковий запис (недолік – скутість суб'єкта); запам'ятовування з наступним записом (недолік – відсутність феноменальної пам'яті)

S12_2_2	освітленість	достатня, не дуже яскрава
S12_2_3	меблі	зручні
S12_3	Професійні прийоми	
S12_3	темп	враховувати темп суб'єкта, не нав'язувати свій
S12_3	стиль	враховувати стиль суб'єкта, не нав'язувати свій. Довжина фраз – 7 ± 2 слова. Враховувати та фіксувати у протоколі інтонацію, міміку, жести суб'єкта.
S12_3	методи	підбір методів до параметрів суб'єкта

Когнітивний шар S13. Когнітивні науки (психологія, нейрофізіологія, ергономіка, інженерія знань) досліджують пізнавальні процеси людини з можливістю подальшого моделювання.

Основними факторами, що впливають на когнітивну адекватність, є когнітивний стиль та семантична репрезентативність поля знань та концептуальної моделі (табл. 4.3).

Когнітивний стиль визначає не стільки ефективність діяльності, скільки спосіб досягнення результату. Когнітивний стиль дозволяє людям з різними здібностями досягати однакових результатів у діяльності, він є системою засобів та індивідуальних прийомів для організації діяльності людини.

Таблиця 4.3 - Основні фактори, що впливають на когнітивну адекватність

Озна-чення	Назва параметра	Хороші результати
S13_1	Когнітивний стиль	Інженеру зі знань корисно вивчати та прогнозувати свій когнітивний стиль.
S13_1_1	полезалежність-полenezалежність	Полenezалежність – увага акцентується на значимих для вирішення завдання аспектах проблеми. Пов'язана з невербальним інтелектом, аналітичності мислення, здатністю до розуміння суті, неконтактністю. Полезалежний суб'єкт більш контактний. Чоловіки більш полenezалежні, ніж жінки. Аналітик має бути полenezалежним, бажано, щоб експерт був полenezалежним. Поєднання в парі полenezалежний-полenezалежний є теж прийнятним.
S13_1_2	імпульсивність-рефлексивність	Імпульсивність - швидке прийняття рішень, часто необґрунтоване. Рефлексивність – схильність до розважливості, здатність до формування понять, продуктивність стратегій розв'язання логічних завдань. Рефлексивність бажана для інженера зі знань та експерта.
S13_1_3	ригідність-гнучкість	Гнучкість – здатність людини до зміни установок та точок зору відповідно до зміни ситуації для пристосування до нової обстановки. Ригідність – протилежність гнучкості, часто проявляється у експертів і людей старшого віку. Інженер зі знань має бути гнучким. Експерт може бути ригідним.
S13_1_4	когнітивна	Здатність людини до розрізнення понять та розбиття їх на

	еквівалентність	класи та підкласи. Чим вужчий діапазон когнітивної еквівалентності, тим тонша класифікація, тим більша кількість виділених ознак. У жінок діапазон когнітивної еквівалентності вужче, ніж у чоловіків.
S13_2	Репрезентативність	Семантична репрезентативність поля знань та концептуальної моделі виключає нав'язування експерту певної моделі уявлень. Інженер зі знань послідовно відтворює світ експерта, використовуючи формальні та неформальні методи. Семантична репрезентативність спрямована на досягнення когнітивної адекватності поля знань та концептуальної моделі.

4.4.2 Лінгвістичний аспект вилучення знань

Лінгвістичний аспект вилучення знань *A2* має справу з вивченням мовних проблем, оскільки мова є основним засобом вилучення знань.

В інженерії знань можна виділити три шари лінгвістичних проблем:

- *S21* – загальний код;
- *S22* – понятійна структура;
- *S23* – словник користувача.

Загальний код *S21* вирішує проблему мовної розбіжності професійної термінології експерта та повсякденної термінології інженера зі знань (табл. 4.4). Згодом загальний код перетворюється на семантичну мережу, що моделює поле знань предметної області (рис. 4.9).



Рисунок 4.9 – Структура загального коду

При спілкуванні аналітик використовує мову *V1* із невербальним компонентом *V1'*, а експерт - мову *V2* з невербальним компонентом *V2'*. Різниця мов *V1* та *V2* створює бар'єр у спілкуванні.

Етапи вироблення загального коду:

- 1) складання словника предметної області - аналітик виписує всі терміни, що вживаються експертом, та уточнює їх зміст;
- 2) групування термінів за змістом та вибір синонімів;
- 3) завершення складання словника предметної області з групування термінів за змістом.

Аналітик повинен максимально розуміти суть проблем і володіти відповідною термінологією, це базисом до створення адекватної БЗ.

Таблиця 4.4 – Параметри загального коду

Означення	Назва параметра	Хороші результати
S21_1 мова аналітика V1	Загально-наукова термінологія	Наявність достатнього обсягу теоретичних знань. Максимальний збіг із загальнонауковою термінологією експерта
S21_2 мова аналітика V1	Терміни, поняття ПрО	Спеціальні терміни та поняття, отримані з професійної літератури під час підготовки. Максимальний збіг зі спеціальною термінологією експерта
S21_3 мова аналітика V1	Побутова мова	Максимальний збіг з побутовою мовою експерта
S21_1 мова експерта V2	Загальнонаукова термінологія	Максимальний збіг із загальнонауковою термінологією аналітика
S21_2 мова експерта V2	Терміни, поняття ПрО	Максимальний збіг зі спеціальною термінологією аналітика
S21_3 мова експерта V2	Побутова мова	Максимальний збіг з побутовою мовою аналітика
S21_4 мова експерта V2	Неологізми, проф. жаргон	Максимальний збіг з неологізмами аналітика

Таким чином, шар S21 передбачає вивчення та управління процесом розробки спеціальної проміжної мови взаємодії аналітика та експерта.

Понятійна структура S22. Особливості формування понятійної структури полягають у необхідності врахування взаємозв'язків понять у пам'яті людини та наявності семантичної мережі, що поєднує окремі терміни у фрагменти, фрагменти у сценарії тощо. Побудова мережі понять (піраміди знань) є найважливішою ланкою у проектуванні інтелектуальних систем. У БЗ поняття мають пояснюватися та зв'язуватися з іншими термінами.

Лінгвістична робота аналітика в даному шарі полягає у побудові зв'язаних фрагментів за допомогою зв'язування термінів.

У процесі роботи аналітика та експерта зі структур понять складається ієрархія понять (онтологій) (рис. 4.10).

Ієрархія понять - глобальна схема, яка може бути основою концептуального аналізу структури знань будь-якої предметної області.

Для складання словника і понятійної структури аналітик повинен мати чуття мови, легко маніпулювати термінами, знаходити синоніми, мати багатий словниковий запас у галузі інженерії знань.

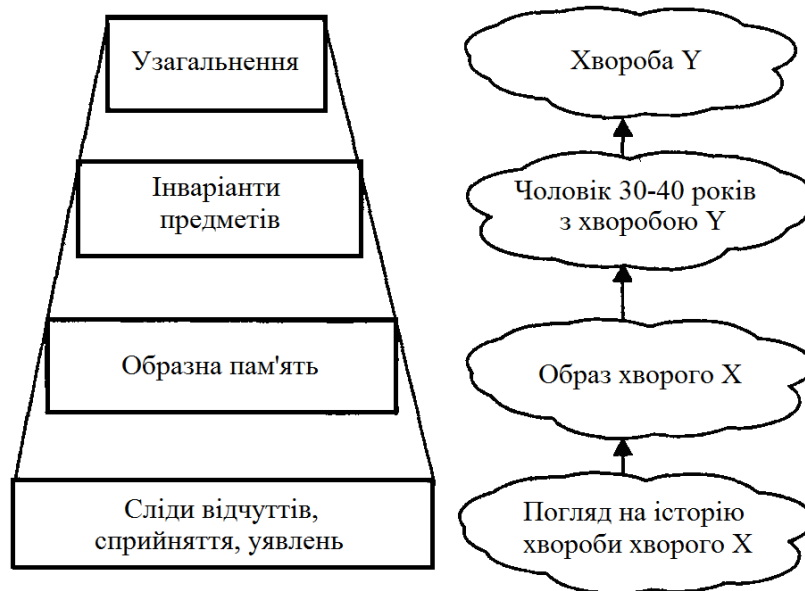


Рисунок 4.10 – Приклад ієрархії понять

Здатність до словесної інтерпретації залежить від статі аналітика: жінки надають великої значущості невербальним компонентам спілкування, а у вербальних мають більший алфавіт ознак. У експерта-чоловіка та експерта-жінки алфавіти для вербалізації ознак об'єктів можуть сильно відрізнятися.

Словник користувача S23. Словник користувача використовується для створення дружнього інтерфейсу з користувачем інтелектуальної системи. При складанні словника користувача необхідно доопрацювати словник загального коду для забезпечення доступності та прозорості функцій системи.

4.4.3 Гносеологічний аспект вилучення знань

Гносеологія – розділ філософії, що вивчає теорію пізнання та відображення дійсності у свідомості людини.

Гносеологічний аспект АЗ вилучення знань поєднує методологічні проблеми здобуття нового наукового знання, бо при створенні БЗ експертом часто вперше формулюються деякі закономірності, які раніше становили його особистий досвід.

При створенні БЗ експертні знання та теоретичний досвід перетворюються на поле знань, що є моделлю зовнішнього світу інженера зі знань.

У процесі отримання знань аналітика цікавлять індивідуальні знання експерта. Такі знання є емпіричними, що складаються з великого обсягу емпіричних фактів та спостережень, і надалі будуть теоретично узагальнені.

Аналітик зазвичай вилучає лише фрагменти знань експерта, а його завданням є створення моделі, адекватної системі знань експерта.

У ході спілкування аналітика та експерта у експерта виникають нові знання, коли він починає усвідомлювати деякі співвідношення та закономірності предметної області. Аналітик користується системною методологією, що представляє принципи логіки наукових досліджень, понятійної ієрархії науки. Ця методологія дозволяє за частковим побачити загальне та вибудовувати гносеологічні ланцюжки:

факт → узагальнений факт → емпіричний закон → теоретичний закон.

Знання, як і гносеологічний підхід, має два рівні:

- 1) емпіричний – спостереження, явища;
- 2) теоретичний – закони, абстракції, узагальнення.

Основними методологічними критеріями науковості, що дозволяють вважати науковим й саме знання, й спосіб його отримання, є компоненти гносеологічного аспекту АЗ:

- S31 – внутрішня узгодженість;
- S32 – системність;
- S33 – об'єктивність;
- S34 – історизм.

Внутрішня узгодженість S31. Основними характеристиками внутрішньої узгодженості емпіричного знання є модальність, суперечливість, неповнота. Часто емпіричні знання складаються з неузгоджених фактів, суперечливих визначень, дифузних критеріїв.

Таблиця 4.5 – Параметри внутрішньої узгодженості

Означення	Назва параметра	Хороші результати
S31_1	модальність	Модальність знання – можливість існування у різних категоріях, частина яких можлива, а інша – обов'язкова. Слід розрізняти відтінки модальності: Експерт знає, що..., думає, що... хоче, щоб... вважає, що.
S31_2	суперечливість	Суперечності не повинні вирішуватись у полі знань. Наявність протиріч часто є відправною точкою в міркуваннях експертів.
S31_3	неповнота	Неповнота – неможливість повністю описати предметну область. Проблема вирішується введенням обмежень, спрощень, припущень на предметну та проблемну галузі.

Системність S32. Будь-який об'єкт має багаторівневу ієрархічну організацію. Усі процеси і явища можна представити як множину підмножин (ознак, характеристик). У свою чергу, будь-які об'єкти слід розглядати як елементи вищих класів та узагальнень.

Об'єктивність S33. Процес пізнання є суб'єктивним, тобто, він істотно залежить від особливостей суб'єкта, що пізнає. Суб'єктивність починається вже з опису фактів та збільшується в міру ідеалізації об'єктів. Тому слід говорити про глибину розуміння, а не про об'єктивність знання.

Розуміння – процес тлумачення об'єкта з погляду суб'єкта. Це складний і неоднозначний процес, що відбувається в людській свідомості і потребує інтелектуальних і емоційних здібностей людини. Зусилля аналітика мають зосереджуватись на розумінні проблеми. Люди, які швидко і успішно вирішують інтелектуальні завдання, більшу частину часу витрачають на її розуміння, а погані вирішувачі швидко приступають до вирішення і часто не можуть його знайти.

Історизм S34. Пізнання сьогодення є пізнанням минулого, що його породило. Більшість систем, що ґрунтуються на знаннях, дають «горизонтальний» зріз знань, але аналітик повинен розглядати процеси з урахуванням часових змін як зв'язок з минулим, так і з майбутнім. Структура поля знань та БЗ повинні допускати підстроювання та корекцію як у період розробки, так і в період експлуатації системи.

Методологічна структура пізнання складається з етапів:

- E_1 – опис та узагальнення фактів;
- E_2 – встановлення логічних та математичних зв'язків, дедукція, індукція законів;
- E_3 – побудова ідеалізованої моделі;
- E_4 – пояснення та передбачення явищ.

Опис та узагальнення фактів E_1. Успіх цього етапу ґрунтується на ретельному та повному веденні протоколів у процесі вилучення знань та пунктуальній роботі над ними. Найчастіше на цьому етапі факти збирають і «складають»; досвідчений аналітик відразу намагається знайти місце факту у системі.

Встановлення зв'язків і закономірностей E_2. У пам'яті експерта всі поняття пов'язані та закономірності встановлені. Завдання аналітика – виявлення структури висновків експерта.

При реконструкції міркувань експерта аналітик спирається на логічну та асоціативну теорію мислення. Логічна теорія найчастіше використовується в роботах зі штучного інтелекту.

Традиційна логіка формує критерії, які гарантують точність, валідність, несуперечність загальних понять, міркувань та висновків.

Аналітик сам використовує операції традиційної логіки та виділяє їх у схемі міркувань експерта:

- визначення;
- порівняння та розрізнення;
- аналіз;
- абстрагування;
- узагальнення;
- класифікація;
- категоризація;
- утворення суджень;
- висновок;
- складання силогізмів та ін.

Теорія асоціацій представляє мислення як ланцюжок ідей, пов'язаних загальними поняттями. Основними операціями асоціативного мислення є:

- асоціації, набуті на основі різних зв'язків;
- пригадування минулого досвіду, спроби та помилки з випадковими успіхами;
- звичні, автоматичні реакції та ін.

Крім цих двох теорій пізнання використовують і гештальт-психологію. Гештальт – принцип цілісності сприйняття як основи мислення. У гештальт-психології намагаються у всьому виділити якийсь цілісний образ чи структуру як базис для розуміння процесів та явищ навколишнього світу. Ця теорія близька теорії фреймів та об'єктного підходу і спрямована на розуміння глибинного знання. У цьому виділяється центр ситуації, навколо якої розвивається знання предметної області. Аналітик, виявляючи різні фрагменти знань, не повинен забувати про гештальт фрагмента, який впливає на інші компоненти і пов'язує їх у єдину структурну одиницю. Гештальтом може бути принцип, ідея, гіпотеза експерта, його віра з окремих концепцій. Аналітик покликаний виявити основний гештальт експерта. Гарний гештальт є гармонійним, зв'язним, простим.

Побудова ідеалізованої моделі Е_3. Для побудови моделі предметної області необхідна мова опису та конструювання. Вона створюється поступово за допомогою категоріального апарату предметної області та формально-знакових засобів математики та логіки. Для емпіричних областей така мова поки що не розроблена.

Будь-яка модель будується на основі спрощення та ідеалізації. Аналітик повинен мати такі прийоми як ідеалізація, огрублювання, абстрагування без впливу несуттєвих деталей.

Аналітик повинен бути готовий змінювати свої сприйняття та оцінки, що утвердилися, перевіряти правильність суджень, які здаються очевидними.

Пояснення та передбачення явищ Е_4. Цей етап є завершальним у процесі пізнання та частковим критерієм істинності здобутого знання. На основі об'єктивної виявленої системи знань можна робити прогнози і пояснювати явища предметної області. Зазвичай БЗ є фрагментарними та модульними (незв'язними), що не дозволяє створювати справді інтелектуальні системи. Застосування запропонованої методології дозволяє уникнути помилок, що призводять до неповноти, суперечливості, фрагментарності БЗ.

4.5 Теоретичні аспекти структурування знань

Методологія структурування знань близька до сучасної теорії великих систем. Системи штучного інтелекту належать до класу великих систем, бо мають ознаки складності – мають ієрархії понять, внутрішньоелементні та міжелементні зв'язки.

Складність проектування інтелектуальної системи визначається складністю предметних областей та складністю процесу розробки, складністю забезпечення гнучкості кінцевого продукту та опису поведінки окремих підсистем.

4.5.1 Ієрархічний підхід

Ієрархічний підхід - методологічний прийом розчленування формально описаної системи на рівні (блоки, модулі). На вищих рівнях ієрархії використовуються найзагальніші уявлення, риси та особливості системи. На наступних рівнях ступінь деталізації зростає, система розглядається окремими блоками.

Об'єктно-структурний підхід дозволяє об'єднати низхідну та висхідну стратегії проектування. Низхідна (*STRtd – top-down*) стратегія є дедуктивною, із послідовною декомпозицією об'єктів згори донизу. Висхідна (*STRbu – bottom-up*) стратегія є індуктивною, з послідовним узагальненням понять та збільшенням ступеня абстрактності описів знизу догори.

Синтез цих стратегій та можливість повернення на попередні рівні узагальнень дозволили створити дуальну концепцію, що застосовується для формування концептуальної та функціональної структури предметної області.

Приклад застосування дуальної концепції під час проектування концептуальної структури для ЕС оператора енергоблоку наведено на рис. 4.11.

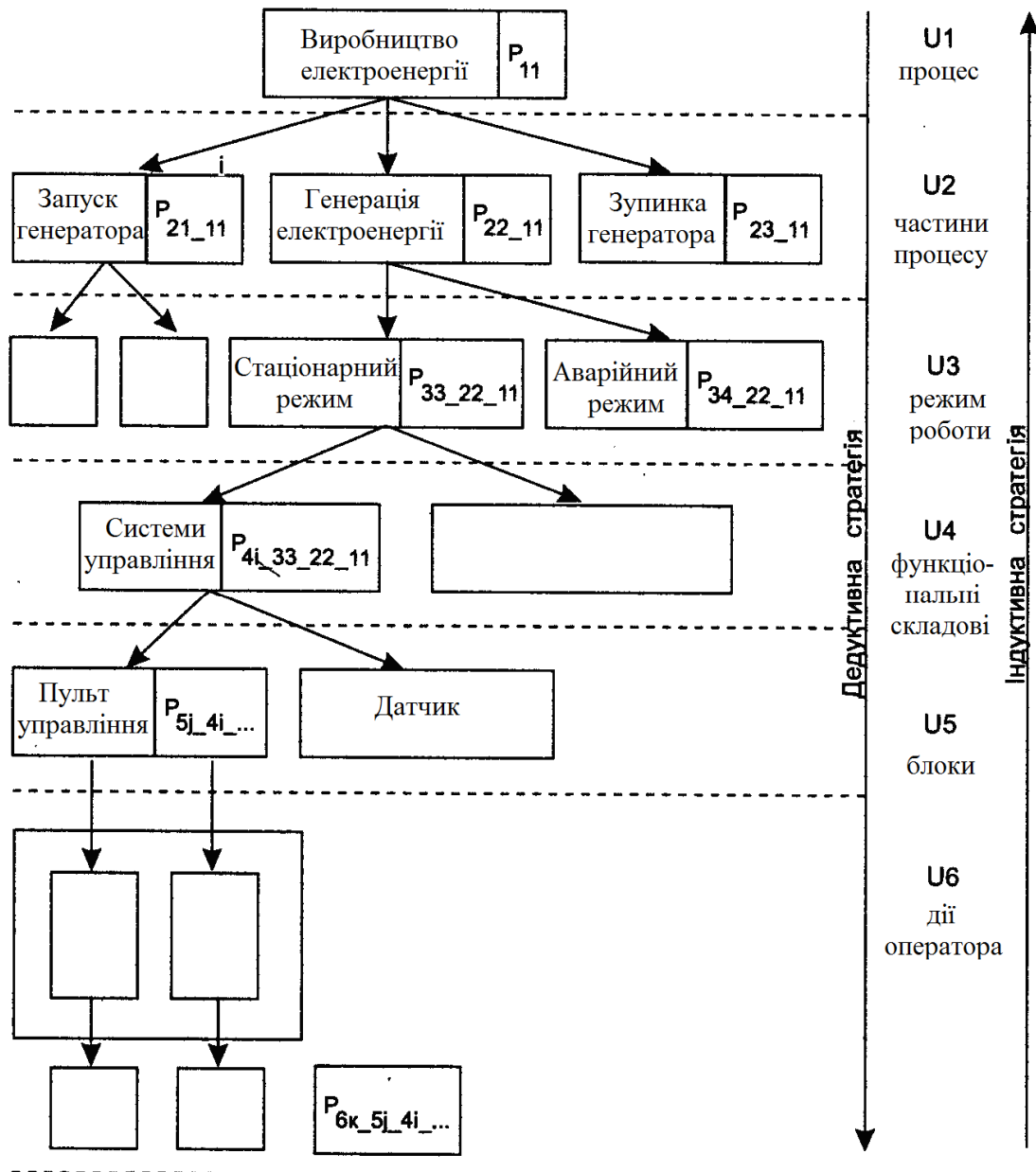


Рисунок 4.11 – Дуальна стратегія проектування

Підставою для припинення композиції і декомпозиції є використання словника термінів експерта.

4.5.2 Традиційні методології структурування

Існуючі підходи до проектування складних систем можна розділити на два класи:

- структурний (системний) підхід – ґрунтується на алгоритмічній декомпозиції; кожен модуль системи виконує один із важливих етапів загального процесу;
- об'єктний підхід – ґрунтується на декомпозиції та виділенні не процесів, а об'єктів, кожен об'єкт розглядається як екземпляр певного класу.

Засобами проектування у межах структурного підходу є DFD-діаграми потоків даних; ERD-діаграми об'єкт-відношення; PERT-діаграми управління проектом та ін. PERT (*Program (Project) Evaluation and Review Technique*) - техніка оцінки та аналізу програм (проектів), яка використовується при управлінні проектами. PERT – це спосіб аналізу завдань, необхідних для виконання проекту.

Об'єктно-орієнтований підхід ґрунтується на елементарних поняттях: об'єкти; класи як об'єкти, пов'язані загальною структурою та властивостями; класифікації як упорядкування знань; ієрархії з наслідуванням властивостей; інкапсуляції як засобі обмеження доступу; методах та поліморфізмі для визначення функцій та відношень.

Засобом проектування є UML-діаграми, представлені уніфікованою мовою моделювання.

Об'єктно-структурний підхід

Об'єктно-структурний підхід (ОСП) заснований на постулатах об'єктно-орієнтованого програмування, розширених для побудови БЗ.

Постулати ОСП:

- 1) системність - взаємозв'язок між поняттями;
- 2) абстрагування - виявлення суттєвих відмінних характеристик поняття;
- 3) ієрархія – впорядкування системи абстракцій за рангами;
- 4) типізація – виділення класів понять із частковим успадкуванням якостей у підкласах;
- 5) модульність - розбиття задачі на підзадачі, області на підобласті;
- 6) наочність та простота нотації (візуального подання).

Принцип модульності дозволяє будувати великі БЗ, у яких можна виділяти задачі за допомогою горизонтального і вертикального зрізів, і розбивати БЗ окремі модулі простору - описи предметної області.

Наочність уявлення моделі області знань спирається на можливість візуалізації процесу проектування.

При розробці БЗ названі вище постулати ОСП використовуються на усіх стадіях. Для проектування на основі ОСП використовується алгоритм об'єктно-структурного аналізу (ОСА) предметної області, що дозволяє оптимізувати та впорядкувати розмиті процедури структурування знань.

Стратифікація знань

Алгоритм об'єктно-структурного аналізу (ОСА) має на увазі поділ предметної області на страти – шари (табл. 6).

Таблиця 6 – Стратифікація предметної області

Шар	Назва знання	Призначення
S_1	НАВІЩО	Стратегічний аналіз – призначення і функції системи
S_2	ХТО	Організаційний аналіз – колектив розробників системи
S_3	ЩО	Концептуальний аналіз – основні концепти, понятійна структура
S_4	ЯК	Функціональний аналіз – гіпотези, моделі прийняття рішень
S_5	ДЕ	Просторовий аналіз – оточення, обладнання, комунікації
S_6	КОЛИ	Часовий аналіз – часові параметри та обмеження

S_7	ЧОМУ	Каузальний (причинно-наслідковий) аналіз – формування підсистеми пояснень
S_8	СКІЛЬКИ	Економічний аналіз – ресурси, витрати, прибуток, окупність

При виконанні об'єктно-структурного аналізу розробляють та використовують матрицю ОСА (табл. 4.7), яка містить подання всієї зібраної інформації, розділеної за шарами (стратами) – вертикальний аналіз, та за рівнями – від рівня проблеми до рівня підзадачі – горизонтальний аналіз. Порядок проведення аналізу може бути будь-яким.

Таблиця 4.7 – Матриця об'єктно-структурного аналізу

Рівні страт	Рівень області $U1$	Рівень проблеми $U2$	Рівень задачі $U3$	Рівень підзадачі $U4$	...	Рівень U_n
$S1$	$E11$	$E12$	$E13$	$E14$		$E1n$
$S2$	$E21$					
$S3$	$E31$					
$S4$	$E41$					
$S5$	$E51$					
$S6$	$E61$					
$S7$	$E71$					
$S8$	$E81$					
...					Eij	
Sm	$Em1$					Emn

При необхідності кількість страт може бути збільшено. У свою чергу, знання кожної страти піддаються подальшому ОСА і розбиваються на складові.

$$\|e_{mn}\|,$$

де m – номер страти;
 e_{mn} належить множині K усіх понять (концептів) предметної області.

Матриця (1) є матрицею над K

$$E_{mn} = \begin{bmatrix} e_{11} & \dots & e_{1n} \\ \dots & \dots & \dots \\ e_{m1} & \dots & e_{mn} \end{bmatrix}, \quad (1)$$

Нехай $M(K)$ – сукупність всіх матриць $m \times n$ над K . Тоді можна визначити клітинну матрицю E , в якій

$$m = m1 + \dots + mk,$$

$$n = n1 + \dots + nl,$$

де m і n – цілі позитивні числа. $E \in M_{m,n}(K)$, і її можна подати у вигляді:

$$E = \begin{bmatrix} E_{11} & \dots & E_{1l} \\ \dots & \dots & \dots \\ E_{k1} & \dots & E_{kl} \end{bmatrix}, \quad (2)$$

де $E_{\mu\nu} \in M_{m\mu}(K)$, $\mu=1, \dots, k$, $\nu=1, \dots, l$.

Матриця E є несиметричною, оскільки частина клітинних елементів $E_{\mu\nu}$ може декомпозуватися, а інші бути атомарними концептами з K , які не підлягають деталізації.

ОСП, яке виконується через ОСА, веде до об'єктно-структурної розробки (ОСР).

Алгоритм ОСА

Алгоритм об'єктно-структурного аналізу призначений для детального практичного структурування знань предметної області. В основі ОСА лежить заповнення ОСА-матриці E_{mn} . Алгоритм описує аналітичні процедури, що дозволяють скоротити та спростити процес структурування знань:

- A_1 – глобальний аналіз (вертикальний), що включає розбиття всієї предметної області на методологічні страти; результат – заповнення першого стовпця матриці (2);
- A_2 – аналіз страт (горизонтальний), що включає побудову багаторівневих структур з окремих страт; число рівнів визначається особливостями страт знань предметної області ($n < 3$ говорить про слабке опрацювання ПрО).

Перший рівень – рівень усієї ПрО; другий - рівень вирішуваної проблеми; третій – рівень розв'язуваної задачі. Подальші рівні відповідають виділеним завданням.

Можливе застосування висхідної, низхідної концепції проектування та їх комбінації.

Глобальний аналіз

Виконується розбиття простору основної задачі ПрО на підадачі, відповідно до особливостей ПрО. Мінімальний набір S -страт для формування БЗ:

- $S3$ – формування концептуальної структури Sk ;
- $S4$ – формування функціональної структури Sf ;
- $S7$ – формування підсистеми пояснень Sn .

Формування інших страт дозволяє оптимізувати процес розробки та уникнути помилок проектування. Страти $S4$ і $S5$ формуються тоді, коли знання ПрО істотно залежить від часових і просторових параметрів.

Алгоритм A_1 :

A_1_1 – зібрати всі матеріали щодо ідентифікації задачі з вилучення знань;

- A_1_2 – вибрати набір страт N для формування (мінімум 3);
- A_1_3 – зібрати інформацію з першої обраної страти;
- A_1_4 – повторити крок A_1_3 для усіх вибраних страт з набору N ;
- A_1_5 – якщо частина інформації залишиться невикористаною, збільшити число страт і застосувати крок A_1_3 до доданих страт, інакше перейти до алгоритму A_2 горизонтального аналізу страт.

Алгоритм A_2 аналізу страт залежить від номера страти і зводиться до реалізації концепції дуального структурування для вирішення конкретної підзадачі.

Алгоритм A_2 для страти S3 (ЩО-аналіз), що дозволяє сформувати концептуальну структуру Sk Про:

- A_2_3_1 - з інформації ЩО-страти вибрати всі значущі поняття та сформулювати концепти;
 - A_2_3_2 – виявити ієрархії та подати їх графічно як структури;
 - A_2_3_3 – деталізувати концепції по низхідній технології;
 - A_2_3_4 – сформувати метапоняття за висхідною технологією;
 - A_2_3_5 – виключити повтори, надмірність, синонімію;
 - A_2_3_6 – проаналізувати та обговорити з експертом поняття, які не увійшли до структури Sf , перенести їх до інших страт або виключити;
 - A_2_3_7 – отриманий граф або набір графів розділити на рівні та позначити згідно з матрицею ОСА (1).
- Аналогічні процедури виконати для усіх страт.

4.6 Моделі представлення знань

Моделі представлення знань - це один з найважливіших напрямків досліджень у галузі штучного інтелекту, тому що без знань штучний інтелект не може існувати в принципі.

На сьогоднішній день розроблено вже достатня кількість моделей. Кожна з них має свої плюси і мінуси, і тому для кожної конкретної задачі необхідно вибрати відповідну модель. Від цього залежить не стільки ефективність виконання поставленого завдання, скільки можливість її рішення взагалі.

Моделі подання знань належать до прагматичного напрямку досліджень в галузі штучного інтелекту. Цей напрямок ґрунтується на припущенні про те, що розумова діяльність людини - «чорний ящик». При такому підході не ставиться питання про адекватність використовуваних в комп'ютері моделей подання знань тим моделям, якими користується в аналогічних ситуаціях людина, а розглядається лише кінцевий результат вирішення конкретних завдань.

Найбільш часто використовуваними є такі моделі представлення знань:

- продукційна модель;
- модель семантичних мереж;

- фреймова модель;
- формально-логічна модель.

4.6.1 Продукційна модель представлення знань

Найбільш широко у США використовуються продукційні моделі представлення знань. Продукційна модель була запропонована в 1943 р. американським вченим Емілем Леоном Постом. Продукційні моделі - це набір правил виду "ЯКЩО-ТО". Продукції моделюють людський спосіб міркувань при вирішенні проблем.

Продукційні моделі можуть бути реалізовані як процедурально, так і декларативно. Їхня простота і строга форма послужили основою ряду цікавих властивостей, що зробило їх зручним засобом представлення знань.

Основними компонентами продукційної системи є неструктурована або структурована база фактів (БФ), кінцева множина продукційних правил або просто продукцій та інтерпретатор.

Кожна продукція складається з двох частин:

- умов (антецедент) – в цій частині визначаються певні умови, які повинні бути наявними в БФ для того, щоб були виконані відповідні дії;
- дій (консеквент) – в цій частині визначаються деякі дії, які повинні бути виконані над БФ у разі виконання відповідних умов; у найпростіших продукційних системах вони тільки визначають, які елементи слід додати до БД (або іноді видалити).

Інтерпретатор послідовно визначає, які продукції можуть бути активовані в залежності від умов, які в них містяться, вибирає одне з застосовних в даній ситуації правил продукцій, виконує дію згідно до вибраної процедури.

У загальному вигляді продукція - це вираз, який має вигляд:

$$(i): Q; P; A \longrightarrow B; N,$$

де (i) – ім'я продукції;
 Q – сфера застосування продукції;
 P – умова застосовності ядра продукції;
 $A \longrightarrow B$ - ядро продукції;
 A – умова ядра (антецедент);
 B – висновок ядра (консеквент);
 $\longrightarrow$ - знак логічної секвенції;
 N – післяумови продукції.

Іменем продукції (i) може бути сукупність літер або символів, що дозволяє однозначно ідентифікувати продукцію в системі. Часто як ім'я продукції використовують її унікальний номер.

Сфера застосування продукції Q явно або неявно описує предметну область знання, яку представляє дана продукція. Декомпозиція предметної

області на окремі незалежні області може істотно підвищити ефективність міркувань в продукційній системі. Сфера застосування продукції може виступати як метазнання або знання про знання.

Умова застосовності ядра продукції P являє собою логічний вираз (як правило, предикат). Якщо вона присутня у продукції, то активізація ядра продукції стає можливою тільки в разі істинності цієї умови. Часто параметр P може бути відсутнім або бути включеним у ядро продукції.

Ядро продукції $A \rightarrow B$ є центральним компонентом продукції. Найчастіше ядро продукції читається як «ЯКЩО A ТО B », де A і B - певні логічні вирази, які в тому чи іншому контексті можуть приймати значення «істина» або «хибність». Ядро слід розуміти так: якщо вираз A є істинним, то істинним є і вираз B . Якщо A є істинним, то про істинність B нічого сказати не можна. Вираз A може задаватися у вигляді складного логічного виразу, який може містити основні логічні зв'язки - заперечення, кон'юнкцію і диз'юнкцію.

Післяумова N описує дії і процедури, які необхідно виконати в разі реалізації ядра продукції, тобто отримання інформації про істинність B . Характер цих дій може бути самим різним і відображати обчислювальний чи інший аспект системи.

У продукційних системах можна виділити 3 основні компоненти:

1. Неструктурована чи структурована БД.
2. Множина продукційних правил або просто продукцій, кожна з яких складається із двох частин:
 - умова (антецедент); у цій частині визначаються деякі умови, які мають виконуватися у БД для того, щоб були виконані відповідні дії;
 - дія (консеквент); у цій частині визначаються деякі дії, які мають бути здійснені над БД у разі виконання відповідних умов. У найпростіших продукційних системах вони лише визначають, які елементи слід додати до БД або видалити з неї.
3. Інтерпретатор, який послідовно визначає, які продукції можуть бути активовані в залежності від умов, що містяться в них; вибирає одне із застосовних у цій ситуації правил продукцій; виконує дію з вибраної процедури.

Продукційні моделі в основному знаходять застосування як вирішувачі або механізми виведення.

У БД системи зберігаються відомі факти із певної предметної області. Продукції містять специфічні для даної області знання про те, які додаткові факти можуть бути допущені, якщо специфічні дані знайдені в БД (або подібно до демонів перевіряти самих себе на предмет того, чи виконуються умови активації). І тут форма подання знань є процедуральною, хоча й у дуже обмеженому вигляді. У наступних ітераціях факти, додані БД, можуть підключати (активувати) інші продукції, тощо.

У класичних продукційних системах БД є змінною частиною системи, тоді як правила та інтерпретатор найчастіше не змінюється. Будучи реалізовані процедурально, класичні продукційні моделі мають досить привабливу

властивість модульності. Тому правила можуть бути додані або видалені без несподіваних побічних ефектів. Причина цього полягає також у тому, що у класичних системах виклик процедур здійснюється лише залежно від стану даних; процедури, зазвичай, не активуються іншими процедурами. Тому продукційні системи можуть бути з великим успіхом використані для областей знань, про які маємо лише незначний набір правил (евристик), а не чітку теорію, цілком завершену і послідовну, і де, тому, немає алгоритмів, що прямо призводять до мети.

Продукційні системи останнім часом дедалі ширше використовуються для реалізації експертних систем, озброєних знаннями людини-експерта про певну предметну область.

Зазвичай, класичні продукційні системи не містять відомостей про застосування, тобто, знання того, наприклад, які продукції використовувати для досягнення мети. Це веде до значного зниження ефективності їх роботи: хоча в кожній ітерації лише одна продукція може бути активована, мають бути перевірені умови всіх продукцій. Для багатьох правил це може вимагати значної витрати ресурсів. Більше того, послідовність виконання продукції на кожній ітерації залежить від стану змінних системи. У цьому випадку виникає проблема комбінаторного "вибуху".

Для вирішення цих проблем пропонувалися підходи, пов'язані з методами структурного вдосконалення БД та умов у продукціях, що дозволило б підвищити ефективність функціонування. Робляться також спроби вплинути на перебіг управління виведенням.

У кожному циклі всі правила, умови яких є дійсними у результаті співставлення з вмістом БД, вважаються визначеними. Якщо існує кілька таких правил, то питання про те, яке правило вибрати, вирішується за допомогою певної стратегії "вирішення конфліктів" (наприклад, вибирається правило з найвищим пріоритетом із заздалегідь визначеного переліку пріоритетів). Усі дії, пов'язані з вибраним правилом, виконуються та викликають відповідні зміни у БД.

Інша можливість полягає у здійсненні точного контролю послідовності виконання продукції. У найпростішому випадку продукції могли б формувати спеціальні сигнали БД, які підключали б відповідні продукції в інших циклах.

У деяких системах самі продукції можуть активувати або деактивувати інші продукції і навіть впливати на роботу інтерпретатора.

Архітектура та особливості продукційної системи

Продукційні системи, у яких спочатку аналізується антецедентна частина (умови), мають так звану умово-вивідну архітектуру. Прикладом системи з такою архітектурою є МЕТА-ДЕНДРАЛ.

Альтернативним типом архітектури, яка досить часто використовується в продукційних системах, є ціле-вивідні (діє-вивідні або консеквент-вивідні) продукційні системи. Наприклад, правило виду

$$A \& B \& C \longrightarrow D$$

може бути інтерпретовано як

"логічна кон'юнкція A, B, C тягне за собою D "

або "щоб довести D , необхідно встановити A, B, C ".

В останньому випадку мета має бути досягнута шляхом дослідження консеквент правил для знаходження такого правила, яке дозволило б досягти мети. Коли таке правило знайдено, перевіряються на істинність його умови. Якщо умови дійсні, продукція активується. В іншому випадку продовжується пошук відповідної продукції.

У продукційних системах правила, за якими вирішуються проблеми у конкретній предметній області, зберігаються у базі знань (рис. 4.12). Проблемна область описується у вигляді фактів і правил. Проблеми ставляться перед системою як сукупності фактів (вхідної інформації), що описують деяку ситуацію, і система з допомогою бази знань намагається знайти висновки з цих фактів.

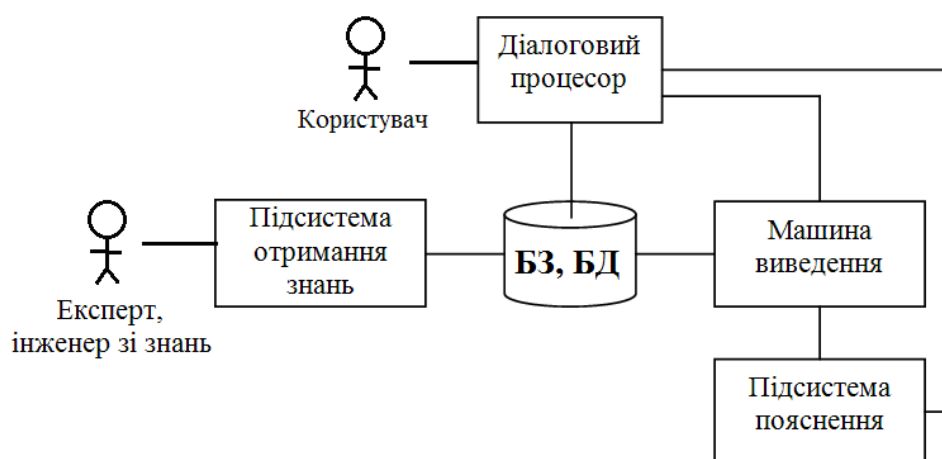


Рисунок 4.12 – Узагальнена архітектура продукційної експертної системи

Продукційна система може працювати у двох режимах:

- 1) отримання знань;
- 2) вирішення завдань (режим консультації).

Отримання знань – це передача системі потенційного досвіду вирішення проблем (задач) від деякого джерела знань та перетворення його до виду, допустимого для використання у комп'ютерній програмі. Найважливіша складова цього процесу – перенесення компетентності експертів на інженерів зі знань. Підсистема отримання знань призначена для додавання до бази знань нових правил та модифікації наявних. До її завдань входить приведення правила до виду, що дозволяє механізму виведення застосовувати це правило у процесі роботи. Найбільш поширений спосіб представлення знань – у вигляді конкретних фактів та правил, за якими з наявних фактів можуть бути виведені нові. Факти можна представляти триплетами (трійками ОБ'ЄКТ АТРИБУТ ЗНАЧЕННЯ). Так, (ПАЦІЄНТ1 ТЕМПЕРАТУРА 37.5) представляє факт «хворий ПАЦІЄНТ1 має параметр ТЕМПЕРАТУРА, що дорівнює 37.5».

У більш простих випадках факт виражається не конкретним значенням атрибуту, а якимось простим твердженням, яке може бути істинним чи хибним, наприклад: «Небо вкрите хмарами». При цьому факту досить дати певне ім'я (наприклад, ХМАРИ) або використовувати для його подання сам текст відповідної фрази.

Синтаксис правил наочно ілюструє приклад:

ЯКЩО A ТО S ,

де A – умова; S – дія.

Як умову A може виступати один або кілька фактів, пов'язаних логічними операціями «І» та «АБО»:

$A11 \& A12 \& \dots \& A_{mn} \vee \dots \vee A_k$

Приклад на природній мові:

Правило1: ЯКЩО	небо вкрите хмарами
І	атмосферний тиск падає
ТО	незабаром піде дощ.

Діалог із системою здійснюється через діалоговий процесор (рис. 4.12). При цьому термін «користувач» є багатозначним, оскільки, крім кінцевого користувача, застосовувати систему може і експерт, і інженер зі знань. У режимі консультації діалоговий процесор обробляє дані про завдання користувача, а також розподіляє ролі учасників (користувача та системи), організує їхню взаємодію в процесі спільного розв'язання задачі, перетворює дані користувача про задачу у внутрішню мову системи, а повідомлення системи її внутрішньою мовою виводить на природній мові користувача. Робота ЕС в режимі консультації може бути представлена у вигляді схеми на рис. 4.13.

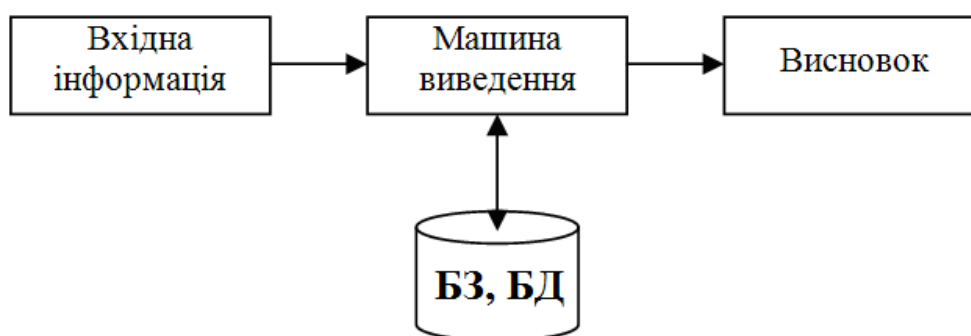


Рисунок 4.13 – Схема виконання виведення

Дії, що входять до складу правил, можуть додавати нові факти. При застосуванні правил із бази знань ці факти стають відомі системі, тобто, додаються до робочої пам'яті. Наприклад, якщо факти «Небо вкрите хмарами» та «Атмосферний тиск падає» вже є в робочій пам'яті, то після застосування наведеного вище Правила1 у робочу пам'ять також додається факт «Незабаром піде дощ». На основі даних з робочої пам'яті, загальних даних про проблемну

область і правил з бази знань механізм логічного виведення формує рішення задачі. У режимі консультації експертна система попередньо формує послідовність дій з рішення задачі, які сама ж і виконує. Якщо відповідь виявляється не зрозумілою користувачеві, він може звернутися до підсистеми пояснення та дізнатися, як було отримано дану відповідь. Такий процес триває доти, доки надійде інформація, достатня для остаточного висновку.

Якщо система не може вивести певний факт, істинність чи хибність якого потрібно встановити, система запитує про нього користувача. Наприклад: ЧИ Є ДІЙСНИМ, ЩО небо вкрите хмарами? При отриманні позитивної відповіді від користувача факт "Небо вкрите хмарами" додається до робочої пам'яті.

Механізми виведення у продукційній системі

Продукційні ЕС в основному знаходять застосування як вирішувачі чи механізми виведення.

Приклад правил для оцінки фінансового стану підприємства:

(П1) ЯКЩО коефіцієнт_рентабельності > 20

ТО рентабельність = «задовільно»

(П2) ЯКЩО заборгованість = «немає»

І рентабельність = «задовільно»

ТО фінансовий_стан = «задовільно»

(П3): ЯКЩО фінансовий_стан = «задовільно»

І репутація = «задовільно»

ТО надійність_фірми = «задовільно»

В БД системи зберігаються відомі факти з деякої предметної області. Продукції містять специфічні для даної області знань про те, які додаткові факти можуть бути припущені, якщо специфічні дані знайдені в БД (або подібно демонам перевіряти самих себе на предмет того, чи виконуються умови активації). В цьому випадку форма подання знань є процедуральною, хоча і в дуже обмеженому вигляді. У наступних ітераціях факти, додані в БД, можуть підключати (активувати) інші продукції і т.д.

У класичних продукційних системах БД являють собою змінну частину системи, в той час як правила і інтерпретатор найчастіше не міняються. Класичні продукційні моделі реалізуються процедурально і характеризуються модульністю. Тому правила можуть бути додані або видалені без виникнення несподіваних побічних ефектів. Причина цього полягає також у тому, що в класичних системах виклик процедур здійснюється тільки в залежності від стану даних; процедури, як правило, не активуються іншими процедурами. Тому продукційні системи можуть бути з великим успіхом використані для областей знань, про які мають тільки незначний набір правил (евристик), а не чітку теорію, цілком завершену і послідовну, і де, тому, немає алгоритмів, що прямо приводять до мети.

Продукційні системи останнім часом все більш широко використовуються для реалізації експертних систем, озброєних знаннями людини-експерта про цю предметну область.

Для вирішення проблем комбінаторного "вибуху" при пошуку рішення експертними системами та підвищення ефективності їхньої роботи пропонувалися підходи, пов'язані з методами структурного вдосконалення БД та умов у продукції, що дозволило б підвищити ефективність функціонування. Робляться також спроби вплинути на хід управління пошуком рішення.

У кожному циклі перегляду БЗ всі правила, умови яких підтверджені вмістом БД, вважаються дійсними. Якщо існує декілька таких правил, то з них формується конфліктний набір, і конфлікт вирішується за допомогою певної стратегії "вирішення конфліктів" (наприклад, вибір правила з найвищим пріоритетом, якщо правилам призначені пріоритети, вибір першого, вибір останнього, вибір найбільш часто використовуваного та ін.). Всі дії, пов'язані з вибраним правилом, виконуються і викликають відповідні зміни в БД.

Інша можливість полягає в здійсненні точного контролю послідовності виконання продукцій. У простому випадку продукції могли б формувати спеціальні сигнали в БД, які підключали б відповідні продукції в інших циклах.

У деяких системах самі продукції можуть активувати або деактивувати інші продукції і навіть впливати на роботу інтерпретатора.

У циклі виведення (рис. 4.14) виконуються такі операції:

1) співставлення – зразок правила співставляється з фактами, наявними у базі фактів;

2) вибір – якщо знайдено декілька підходящих правил, то вони створюють конфліктний набір; з конфліктного набору вибирається одне правило, яке найбільше підходить за заданим критерієм – тобто виконується рішення конфлікту;

3) спрацьовування – якщо співставляється антецеденту правила з фактами робочої пам'яті виконано успішно, то правило спрацьовує;

4) дія - до робочої пам'яті додається новий істинний факт, що є консеквентом правила, яке спрацювало.

Робота машини виведення залежить від стану робочої пам'яті та складу БЗ. Зазвичай враховується історія, тобто, поведінка механізму виведення у попередніх циклах. Інформація про поведінку механізму виведення запам'ятовується у стеку станів (рис. 4.15). Стек станів протоколює кроки роботи системи.

Вже на ранній стадії розробки продукційної ЕС необхідно знати, яку інформацію вводитиме кінцевий користувач. При цьому участь користувача виражається в наступному:

– конкретизація завдання - користувач, стикаючись з конкретними проблемами, може пояснити виникнення проблем та запропонувати можливі варіанти їх вирішення;

– спілкування - інтерфейс користувача повинен відповідати його словнику та рівню підготовки;

- встановлення зв'язків - знайомство користувача з причинами та наслідками, що викликають ту чи іншу дію в процесі функціонування системи, є важливим у визначенні взаємозв'язків фактів у базі знань;
- зворотний зв'язок - відмінною особливістю зручної у використанні системи є її здатність пояснити кінцевому користувачеві хід своїх міркувань.

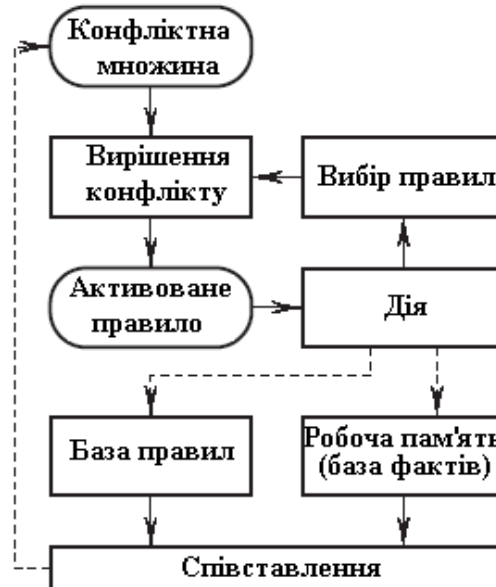


Рисунок 4.14 – Цикл роботи механізму виведення

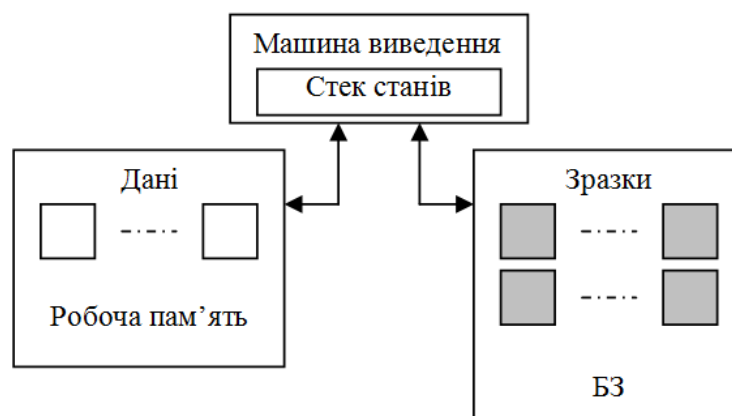


Рисунок 4.15 – Схема функціонування машини виведення

Управління пошуком рішень у продукційних системах

Пошук рішень у продукційних системах виконується за допомогою машини виведення. Машина виведення інтерпретує правила та використовує факти бази знань для вирішення проблем.

На кожному кроці роботи машина виведення будує трасу виведення і намагається підібрати правило, яке вважається найкращим кандидатом для

застосування та відповідає зразку-цілі виведення. При цьому ціль виведення може також задаватися у вигляді зразка-цілі.

Зміна контексту реалізується або внаслідок застосування правил виведення, або шляхом введення фактів, які запитують у користувача.

На машину виведення покладаються наступні основні задачі.

1. Побудова траси виведення.
2. Інтерпретація правил виведення та їх застосування.
3. Управління пошуком.

Процес побудови траси виведення полягає у запам'ятовуванні послідовності кроків виведення, застосовуваних правил і одержуваних значень змінних, тобто формування контексту. Мета побудови траси - можливість її перевизначення з деякого раніше породженого контексту.

Створення поточного контексту пов'язане з вибором правила виведення з множини альтернативних правил (конфліктного набору). Від такого вибору залежить ефективність роботи системи (швидкодія), тому необхідно вживати засоби для забезпечення достатньої ефективності машини виведення.

Відомо два методи управління виведенням:

- встановлення обмежень на генерацію конфліктного набору;
- визначення алгоритму розв'язання конфлікту.

Встановлення обмежень на генерацію конфліктного набору. Спосіб встановлення обмежень на генерацію конфліктного набору залежить від змісту правил. В якості цього способу може використовуватися:

- метод, за яким пошук умовної частини правил (антецедентів) певної категорії не здійснюється;
- метод з попередніми розбиттям правил на окремі категорії; в певних ситуаціях досліджується можливість застосування правил, що належать тільки до деякої категорії.

Для реалізації першого підходу розглядаються методи з використанням мета-правил, тобто правил, в умовній частині яких міститься умова застосування, що стосується змісту правила (умови та висновку) і вмісту робочої пам'яті (РП), або правил, що мають умовну частину, яка вказує атрибути (як умовні, так і заключні), що не підлягають пошуку або дослідження для даної проблеми. У другому випадку правила попередньо групуються по атрибутах, для кожної групи вказується умова (її форма аналогічна антецеденту правила), що стосується вмісту РП, і досліджується можливість застосування правила тільки з тієї групи, в межах якої виконується ця умова, або група вказується за допомогою заключної частини правила та допускається або забороняється застосування правил цієї групи.

Виведення за пріоритетом глибини використовується у випадках, коли явно задана умова зупинки системи. Виведення за пріоритетом глибини фактично відповідає спробі пріоритетного застосування правила з посиленням на останню інформацію, якою була доповнена РП.

При порівнянні Правил 2 і 3 на рис. 4.16 перевага віддається Правилу 3. Перевага: мінімальність шляху логічного виведення, реалізується зрозумілий користувачеві шлях логічного виведення. Якщо ж існує множина правил з посиленням на останню занесену в РП інформацію, то пріоритет може бути відданий правилом з більш жорсткою умовною частиною (з великою кількістю умов). У системах, в яких користувач по ходу виведення запитує дані, можуть бути застосовані алгоритми вирішення конфліктів, що використовують знання предметної області для оптимізації послідовності запитів.

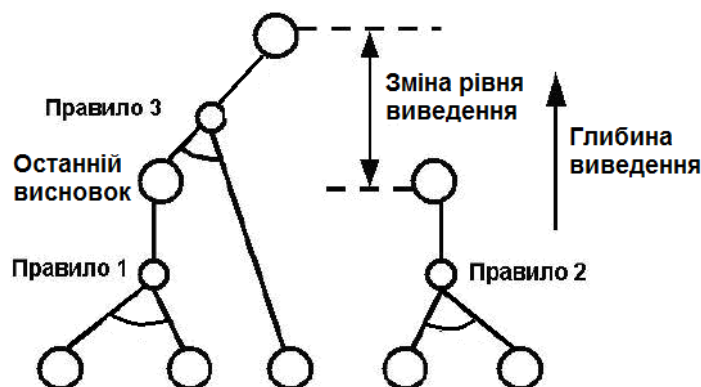


Рисунок 4.16 – Застосування правил при виконанні виведення в глибину

Проблеми реалізації стратегій пошуку виведення:

- обмеження конфліктного набору;
- вибір алгоритму розв'язання конфлікту;
- послідовність оцінки умов в умовній частині правила при зворотному виведенні; можливі випадки, коли дані, на які посилаються в кожній умові, повинні бути визначені за допомогою виведення, тобто вибір однієї умови породжує нову гілку пошуку; найчастіше ефективним вважається оцінювання з посиленням на дані, "близькі" до основних даних.

В системі продукцій розмір БП загострює проблему підвищення ефективності функціонування. Дійсно, якщо ми маємо NT правил і NP зразків, то для простого зіставлення зразків при прямому виведенні звернення до РП повинно виконуватися $NT \times NP$ разів. Якщо число правил і зразків перевищує кілька сотень (як у реальних експертних системах), то повторювати зіставлення зразків відповідне число разів практично неможливо і необхідно шукати нові шляхи підвищення ефективності.

Шляхи підвищення ефективності функціонування системи продукцій:

- 1) підвищення ефективності пошуку інформації в РП шляхом упорядкування її вмісту у формі "об'єкт-атрибут-значення";

- 2) використання алгоритму узгодження RETE, за яким кожен раз при доповненні РП новим зразком перевіряється правило, в якому він використовується; якщо зразок задовольняє частині умов правила, то він запам'ятовується саме в цій якості; якщо додавання зразка задовольняє всім умовам, дане правило включається в конфліктний набір;

3) попередня компіляція відношень між правилами і даними, а також між правилами і правилами попередньо компілюється у вигляді графа і при кожному поповненні основних даних відшукується гілка графа і позначаються пов'язані з нею вершини;

4) розбиття правил на групи, за рахунок чого досягається обмеження розмірів породжених одночасно конфліктних наборів;. типовий приклад-модель дошки оголошень.

Вибір методу пошуку рішення залежить від наступних параметрів:

- розмір предметної області;
- змінність предметної області;
- повнота моделі, яка описує предметну область;
- визначеність даних про розв'язувану задачу;
- кількість необхідних рішень;
- обмеження на результат і спосіб його одержання.

В залежності від цих параметрів використовуються наступні методи:

- пошук рішення в одному просторі станів (малі статичні області, точні і повні дані);
- пошук в ієрархічних просторах (робота у великих статичних областях);
- пошук при неточних і неповних даних;
- пошук в динамічних проблемних областях;
- пошук з використанням декількох моделей розгляду області (з різних точок зору).

Розглянемо стратегії вибору продукції з множини готових продукцій.

1. Принцип «стосу книжок». Базується на ідеї, що найбільш часто використовувана продукція є найбільш корисною. На самій горі «стосу» знаходиться продукція, яка використовувалася найчастіше. При актуалізації деякої множини готових продукцій для поповнення вибирається та продукція, яка використовувалася найчастіше. Управління за принципом стосу зручно застосовувати, якщо продукції є відносно незалежними, коли кожна з них є правилом виду «ситуація $A \rightarrow$ дія B ». Тому вони використовуються в плануючих системах для робіт.

2. Принцип найбільш довгої умови. Полягає у виборі з множини готових продукцій тієї, у якої став істинним найбільш довгий антецедент ядра. Цей принцип спирається на міркування здорового глузду, що приватні правила, що належать до вузького класу ситуацій, важливіші від загальних правил, оскільки враховують більше інформації про ситуацію. Управління за цим принципом доцільно виконувати в тих випадках, коли знання і самі продукції добре структуровані прив'язкою до типових ситуацій, на яких задане відношення типу «приватне - загальне».

3. Принцип метапродукцій. Спирається на ідеї введення в систему продукцій спеціальних метапродукцій, завданням яких є організація управління в системі продукцій при можливості неоднозначного вибору з фронту готових

продукцій (використовується в ЕС MYCIN при діагностиці інфекційних захворювань).

4. Принцип пріоритетного вибору. Пов'язаний з введенням статичних або динамічних пріоритетів на продукції. Статичні пріоритети можуть формувати апріорі на підставі відомостей про важливість продукційних правил в даній предметній області. Це зазвичай інформація, що отримується від експертів. Динамічні пріоритети виробляються в процесі функціонування системи продукції.

5. Управління за іменами. Для задання імен продукцій, що входять в деяку систему, використовується певна формальна граматики або інша процедура, яка дозволяє формувати множину готових продукцій і вибирати з неї чергову продукцію.

Переваги продукційних систем:

- простота створення і розуміння окремих правил;
- простота поповнення та модифікації;
- простота механізму логічного виведення.

Недоліки продукційних систем:

- незрозумілість взаємних відношень між правилами;
- складність оцінки цілісного образу знань;
- низька ефективність обробки БП при виконанні виведення;
- відсутність гнучкості в логічному виведенні.

4.6.2 Модель семантичних мереж

Семантична мережа - структура для подання знань у вигляді вузлів, з'єднаних дугами. Найперші семантичні мережі були розроблені в якості мови-посередника для систем машинного перекладу, а багато сучасних версії і тепер подібні за своїми характеристиками до природної мови. Однак останні версії семантичних мереж стали більш потужними і гнучкими і складають конкуренцію фреймовим системам, логічному програмуванню та іншим моделям подання знань.

Термін «семантична (сміслова) мережа» (або «концепція слів») був введений в 1968-69 рр. в роботах Р. Куїлліана. Він охоплює цілий клас загальних підходів, для яких характерне використання графічних схем з вузлами, з'єднаними дугами (графів).

Спосіб представлення знань у моделях семантичних мереж найбільш близький до того, як вони представлені в текстах на природній мові. У його основі лежить ідея про те, що вся необхідна інформація може бути описана як сукупність трійок $\langle a \ r \ b \rangle$, де a і b -об'єкти, а r -бінарне відношення між ними.

Формально модель семантичної мережі може бути задана у вигляді

$$H = \langle I, C_1, \dots, C_n, R \rangle,$$

де I – множина інформаційних одиниць (ІО);

$C_1, \dots, C_n$ – множина типів зв'язків (відношень) між елементами I ;
 R – відображення I у C , що задає відношення між ІО, що входять в I , використовуючи зв'язки із заданого набору типів зв'язків (C_i).

Особливість семантичних мереж полягає в тому, що в теорії семантичних мереж розглядаються не тільки графи-дерева, але циклічні і повнозв'язні графи і зв'язки різного типу. Крім того окремо і особливо досліджуються можливі типи відношень (фундаментальних та нефундаментальних).

Фундаментальними відношеннями є:

АКО (*a kind of*) – є видом;
 ISA (*is a*) – є представником;
 HP (*has part*) – має частину;
 ISPART (*is part*) – є частиною.

Спочатку семантичні мережі були розроблені для аналізу природних мов і побудови психологічних моделей людської пам'яті (задача автоматичного перекладу - положення Аристотеля про те, що «людина мислить мовою», завдання підбору синонімів до заданого слова та ін.). На цьому етапі вважалося, що в реченні є певна «центральна тема», «розкрутивши» яку, машина може «зрозуміти» сенс (семантику) речення (рис. 4.17).

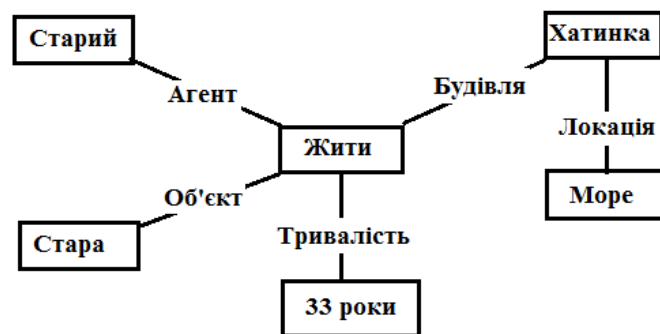


Рисунок 4.17 - СС для речення «Старий жив зі старою 33 роки у хатинці біля моря»

Семантика - наука, що встановлює відношення між символами і об'єктами, які вони позначають, тобто наука, яка визначає сенс знаків.

Семантична мережа - це орієнтований граф, вершинами якого є поняття, а дугами - відношення між ними. Об'єктами можуть бути поняття, події, властивості, процеси. Таким чином, семантична мережа є одним із способів представлення знань. У назві «семантична мережа» сполучені терміни з двох наук - термін семантичності означає смислова, семантика - це наука, яка встановлює відношення між символами і об'єктами, які вони позначають, тобто наука, що визначає сенс знаків. А мережа в математиці є різновидом графа - набору вершин, сполучених дугами (ребрами).

У семантичній мережі роль вершин виконують поняття бази знань, а дуги (направлені лінії) задають відношення між ними. Таким чином, семантична мережа відображає семантику предметної області у вигляді понять і відношень. Це найбільш загальна модель представлення знань, тому що в ній є засоби реалізації всіх, властивостей, характерних для знань: внутрішньої інтерпретації, структурованості, семантичної метрики і активності.

Експериментальну перевірку теоретичних положень про організацію людської пам'яті та організацію логічного виведення Р.Куїлліан провів спільно з психологом А. Колінзом (Allan Collins) в 1969-70 рр. на основі ієрархічної мережі. Для такої мережі характерні відношення виду клас - елемент класу; властивість – значення. Приклад елемента класу зображено на рис. 4.18.



Рисунок 4.18 - Ієрархічна мережа з використанням відношення «є»

Вузли в семантичній мережі зазвичай відповідають об'єктам, концепціям, подіям або поняттями.

Дуги можуть бути визначені різними методами, що залежать від методу представлення знань.

Будь-який фрагмент мережі, наприклад, одна вершина, дві вершини і з'єднують їх дуги, називають підмережою.

Логічний виведення (пошук рішення) на семантичній мережі полягає саме в тому, щоб знайти або сконструювати підмережу, що задовольняє деяким умовам. Для того щоб формалізувати цей процес, вводять типізацію семантичних мереж і потім розробляють методи рішення для мереж конкретного виду (на основі аналізу математичних властивостей відношень, що входять у мережу).

Мережі різного виду отримують і в залежності від того, які обмеження накладають на вершини та дуги.

Якщо вершини мережі не мають внутрішньої структури, то їх називають простими, якщо вершини мають внутрішню структуру, то їх називають ієрархічними.

Динамічні семантичні мережі (сценарії) - мережі з подіями. Одна з основних відмінностей ієрархічної мережі від простої мережі - у можливості розділяти мережу на підмережі і встановлювати відношення не тільки між вершинами, але і між підмережами. Різні підмережі, що існують у мережі, можуть бути впорядковані у вигляді дерева підмереж, вершини якого -

підмережі, а дуги - відношення видимості. Поняття підмережі аналогічно поняттю дужок в математичній нотації, поняття видимості - поняттю змінних, що знаходяться всередині і зовні дужок.

Опишемо за допомогою СС технічний об'єкт «Арка» (рис. 4.19) – перекладина, що спирається на 2 опори, які не повинні стикатися один з одним. Формально цю мережу можна описати так:

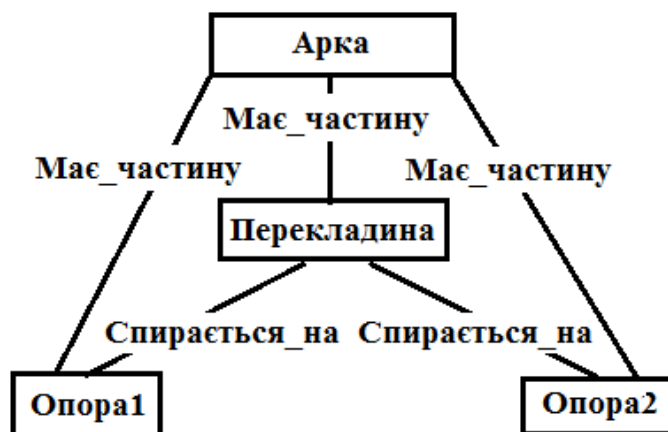


Рисунок 4.19 - Семантична мережа поняття арка

Арка Має_частину Опора1;
 Арка Має_частину Опора2;
 Арка Має_частину Перекладина;
 Перекладина Спирається_на Опора 1;
 Перекладина Спирається_на Опора 2.

Фактично для опису семантичних мереж і продукційних правил використовуються одні і ті ж машинні мови. Хоча перевагою мереж є наочність представлення знань, виведення на мережах є, безумовно, складнішим. Більш того, багато типів відношень досі не досліджені і формального алгоритму виведення не існує.

Найпростіші семантичні мережі, які використовуються в системах штучного інтелекту, – реляційні граfi. Вони складаються з вузлів, з'єднаних дугами. Кожен вузол являє собою поняття, а кожна дуга - відношення між різними поняттями.

При роботі з реляційними граfiами виникають проблеми з передачею всього різноманіття часових відношень і відношень модальності. Незважаючи на те, що ці граfiи широко використовують для вирішення складних проблем, до цих пір так і не розроблено загального методу для їх вирішення.

У пропозиційних мережах вузли представляють цілі речення. Ці вузли є точками дотику для відношень між окремими реченнями пов'язаного тексту. З іншого боку, вони визначають час і модальність для всього контексту.

Класифікація семантичних мереж

Для всіх семантичних мереж справедливим є поділ за арністю і кількістю типів відношень.

За кількістю типів відношень мережі можуть бути однорідними і неоднорідними. Однорідні мережі мають тільки один тип відношень (стрілок), наприклад, такою є класифікація біологічних видів (з єдиним відношенням АКО – є видом). У неоднорідних мережах кількість типів відношень більше двох. У класичних прикладах представлення знань за моделлю семантичних мереж застосовуються саме такі мережі. Неоднорідні мережі представляють більший інтерес для практичних цілей, але і велику складність для дослідження.

За арністю відношень типовими є мережі з бінарними відношеннями (що зв'язують рівно два поняття). Бінарні відношення є простими і зображаються на графі у вигляді стрілки між двох концептів. Крім того, вони грають виняткову роль в математиці. Однак на практиці можуть знадобитися відношення, що зв'язують більше двох об'єктів, тобто, N -арні.

Залежно від типів зв'язків (C_i) розрізняють класифікуючі, функціональні мережі та мережі-сценарії.

Класифікуючі мережі дозволяють вводити в бази знань різні ієрархічні відношення між елементами множини інформаційних одиниць (об'єктів). У класифікуючих мережах використовуються відношення структуризації.

Функціональні мережі – це обчислювальні моделі, що характеризуються наявністю функціональних відношень. Вони використовуються для опису процедур обчислень одних інформаційних одиниць через інші.

Мережі сценарії передбачають використання каузальних відношень (причинно-наслідкових відношень або таких, що встановлюють вплив одних явищ або фактів на інші) або відношень типу "засіб-результат", "знаряддя-дія" і т.д.

Переваги мережевих моделей:

- великі виразні можливості;
- наочність системи знань, представленої графічно;
- близькість структури мережі, що представляє систему знань, семантичній структурі фраз на природній мові;
- відповідність сучасним уявленням про організацію довготривалої пам'яті людини.

Недоліки мережевих моделей:

– мережева модель не дає (точніше не містить) зрозумілого представлення структури предметної області, яка їй відповідає, тому формування і модифікація такої моделі є складними (існує точка зору, що СМ – це формалізм, який існує для представлення висловлювань про зовнішній світ, а не для представлення проблемно-орієнтованих моделей зовнішнього світу);

– складність пошуку виведення на СМ: мережеві моделі являють собою пасивні структури для обробки яких необхідний спеціальний апарат формального виведення і планування – проблема пошуку рішення в БЗ типу

СМ зводиться до задачі пошуку фрагмента мережі, який відповідає деякій підмережі поставленого завдання (запиту).

Для реалізації СМ існують спеціальні мережеві мови (наприклад, OWL).

Ще раз підкреслимо, що мережеві моделі є дуже наочним і досить універсальним засобом представлення знань. Проте їх формалізація в конкретних моделях подання, використання і модифікації знань виявляється досить трудомісткою, особливо за наявності множинних відношень між її елементами.

Від простих семантичних мереж в даний час перейшли до ієрархічних мереж, у яких вершини мають власну внутрішню структуру. Це дозволяє розділити мережу на підмережі (простори) і встановлювати відношення не тільки між вершинами, але і між просторами.

Крім того, використовуються розширені семантичні мережі, які забезпечують подання складних видів інформації, наприклад, коли об'єкти, пов'язані відношеннями, утворюють сукупності, коли відношення у свою чергу рекурсивно зв'язуються відношеннями. Такі мережі є розвитком в напрямку підвищення образотворчих можливостей при збереженні властивості однорідності подання різних видів інформації, представленої у формах природної мови, в тому числі типових логічних конструкцій зі словами - кванторами, з нечіткими категоріями, різного роду невизначеностями.

Семантичні відношення

Кількість типів відношень в семантичній мережі визначається її розробником, який виходить з конкретних цілей. У реальному світі їх число прямує до нескінченності. Кожне відношення є, по суті, предикатом, простим або складним. Швидкість роботи з базою знань залежить від того, наскільки ефективними є компоненти ЕС, призначені для обробки відповідних відношень.

Ієрархічні (фундаментальні) відношення.

Найбільш часто виникає потреба в описі відношень між елементами, множинами і частинами об'єктів.

Відношення між об'єктом і множиною, що позначає, що об'єкт належить цій множині, називається відношенням класифікації (ISA). Кажуть, що множина (клас) класифікує свої екземпляри. Назва походить від англійського «IS A» (найбільш точний переклад, що використовується в основному в наукових колах – «суть», наприклад, «всі зайці суть ссавці»). Іноді це відношення іменують також *MemberOf* або подібним чином. Зв'язок ISA припускає, що властивості об'єкта успадковуються від множини. Зворотне до ISA відношення використовується для позначення прикладів, тому так і називається «*Example*», або українською «*Наприклад*».

Відношення між надмножиною і підмножиною називається АКО - «*A Kind Of*» («*Різновид*»). Елемент надмножини називається гіперонімом, а підмножини – гіпонімом, а саме відношення називається відношенням гіпоніму. Альтернативні назви - «*SubsetOf*» і «*Підмножина*». Це відношення визначає, що кожен елемент другої множини входить в першу (виконується ISA

для кожного елементу), а також що є логічний зв'язок між самими підмножинами: що друга не більше першої і властивості першої множини успадковуються другою.

Об'єкт, як правило, складається з декількох частин, або елементів. Наприклад, комп'ютер складається з системного блоку, монітора, клавіатури, миші і т. д. У цьому випадку важливим відношенням є *HasPart* (або АРО - *A Part Of*), що описує частини/цілі об'єкти (відношення меронімії).

Гіперонім-гіпонім (загальне-приватне). Цей тип відношень представляє зв'язок між абстрактним поняттям і конкретним. Так, поняття «зелений», буде гіпонімом до поняття «колір», так як «зелений» – окремий випадок кольору. Або до поняття «трамвай» гіперонімом буде «міський транспорт», до поняття «хімія» гіперонімом буде поняття «наука». Гіперонім часто використовується як головний член визначення, наприклад, «метан – органічна сполука ...»

Холонім-Меронім (ціле-частина). Ці відношення представляють зв'язок між об'єктом і його компонентами. У дерева є гілки, корінь, листя і т.д., і поняття «гілка» (і решта) буде меронімом до поняття «дерево», а саме «дерево» поняттям-холонімом до «гілка», «лист» і т. д. Меронім – це об'єкт, що є частиною іншого. Двигун – це меронім для автомобіля. Холонім – це об'єкт, який включає в себе інші об'єкти, що є його компонентами. Наприклад, будинок має дах. "Будинок" – холонім для поняття "дах". "Комп'ютер" – холонім для поняття "монітор". Меронім і холонім - протилежні поняття.

Допоміжні відношення

У семантичних мережах часто використовуються також наступні відношення:

- родо-видові відношення;
- відношення «ціле-частина»;
- еквівалентність:
 - еквівалентність – поняття означають одну й ту ж сутність, наприклад, «бегемот» і «гіпопотам»;
 - синонімія – розширення відношення еквівалентності, коли два поняття відповідають більшою частиною одній сутності, але можуть мати незначні відмінності;
 - несумісність – одна і та ж сутність не може відповідати обома поняттями одночасно, наприклад, «червоний» і «синій»;
 - антонімія – окремий випадок несумісності, коли два поняття позначають повні протилежності, наприклад, «широкий» і «вузький»;
- логічні зв'язки (І, АБО, НЕ);
- функціональні відношення (визначені зазвичай дієсловами «виробляє», «впливає» ...);
- атрибутивні відношення (мати властивість, мати значення);
- кількісні відношення (більше, менше, так само, ширше, вже, яскравіше, тьмяніше, швидше, повільніше, і т. п. ...);
- просторові відношення (далеко від, близько від, за, під, над ...);

- часові відношення (раніше, пізніше, протягом, одночасно ...);
- лінгвістичні відношення.

Цей список може скільки завгодно продовжуватися: в реальному світі кількість відношень величезна. Між поняттями може використовуватися відношення "абсолютно різні речі" або "не мають відношення один до одного" (Сонце, Кухонний чайник).

Особливості використання деяких типів відношень

У семантичній мережі поняттями можуть бути як екземпляри об'єктів, так і їхні множини. Використання одних і тих же відношень і для елементів, і для множин елементів може призвести до непорозумінь. Подібні помилки в роботі деяких перших систем були описані в статті Д.Макдермотта «Штучний інтелект стикається з природною дурістю».

Розглянемо приклад – чотири речення:

- I. У Павла є батько на ім'я Олексій.
- Па. Для Павла знайдеться батько з множини чоловіків.
- Пб. Знайдеться людина, для якого Олексій є батьком.
- ІІІ. У кожної людини є батько з множини чоловіків.

Для людини сенс цих фраз зрозумілий - є батько. Проте у формальному сенсі це є помилкою: В одному випадку, дійсно, описується відношення між двома екземплярами, але в другому і третьому - між екземпляром та множиною, а в четвертому - відношення між представниками з двох множин. У математичного записі відповідно для пропозицій 1-4 це має такий вигляд:

- I. $\exists \text{Павло} \ \& \ \exists \text{Олексій: батько(Олексій, Павло);}$
- Па. $\exists \text{Павло} \ \rightarrow \exists x \in \text{чоловіки: батько}(x, \text{Павло});$
- Пб. $\exists \text{Олексій} \rightarrow \exists y \in \text{люди: (батько(Олексій, } y));$
- ІІІ. $\forall y \in \text{люди} \rightarrow \exists x \in \text{чоловіки: батько}(x, y).$

Ми бачимо, що випадки відрізняються тільки порядком входження змінних у предикат, однак при побудові правильної мережі це може зіграти важливу роль. У прикладі перераховані лише 4 роди відношень, всього ж для бінарної мережі їх існує дев'ять. Вони розрізняються кванторами, а також порядком змінних.

Графічно для відмінності всіх цих випадків застосовують спеціальні форми позначок відношень на графі: наприклад, відношення першого роду залишають без змін, другого – обводять прямокутною рамкою з точок, третього - тире, а четвертого – тире-крапка, або можна просто написати поруч індекс типу відношення.

Найбільш часто плутанина виникає щодо відношення ISA. Тому в багатьох сучасних роботах приймається, що ISA позначає зв'язок між

екземпляром та множиною: "Мурка ISA кішка". Одиночна рамка при цьому не використовується. Якщо потрібно визначити відношення еквівалентності (випадок I), то для цього може вводитися спеціальне відношення (хоча для семантичної мережі потреба в ньому невелика). ISA можна використовувати для позначення входження елементів однієї множини в іншу, проте так робити не рекомендується. Для позначення підмножин застосовується ще одне спеціальне відношення – АКО. Різниця між "ISA в рамці" і АКО полягає в тому, що останнє передбачає ще й за успадкування властивостей самих множин, а не тільки елементів.

Застосування семантичних мереж

Семантичні мережі можуть бути програмно реалізовані практично на будь-якій мові програмування. Найпопулярнішими в цьому відношенні є мови LISP і PROLOG. Однак багато версії були створені і на C++, Java, Python та інших мовах програмування. Для зберігання всіх вузлів і дуг необхідна велика пам'ять, хоча перші системи були виконані в 60-х роках на машинах, які були набагато менше і повільніше сучасних комп'ютерів.

Одна з найпоширеніших мов, розроблених для запису природної мови у вигляді мереж, – мова PLNLP (Programming Language for Natural Language Processing - Мова програмування для Обробки Природної Мови), створена Хайдерном. Ця мова використовується для роботи з великими граматами з великим покриттям.

PLNLP працює з двома видами правил:

- 1) за допомогою правил декодування проводиться синтаксичний аналіз лінійного мовного ланцюжка і будується мережа;
- 2) за допомогою правил кодування мережа сканується і породжується мовний ланцюжок або інша трансформована мережа.

Крім спеціальних мов для семантичних мереж було також розроблено спеціальне апаратне забезпечення. На звичайних комп'ютерах можуть бути успішно виконані операції з мовами синтаксичного аналізу та операції сканування мереж. Однак для великих баз знань знаходження потрібних правил або доступ до передзнання може знадобитися дуже багато часу. Щоб дозволити різним процесам пошуку проходити одночасно, Фальман розробив систему NETL, яка представляє собою семантичну мережу, яка може використовуватися з паралельним апаратним забезпеченням. Таким чином він хотів створити модель людського мозку, в якому сигнали можуть рухатися по різних каналах одночасно. Інші вчені розробили паралельне програмне забезпечення для пошуку найбільш вірогідною інтерпретації двозначних фраз природної мови.

Семантична павутина

Концепція організації гіпертексту нагадує однорідну бінарну семантичну мережу, проте тут є суттєва відмінність:

1) зв'язок, що є гіперпосиланням, не має семантики; призначення семантичної мережі полягає в тому, щоб описати взаємозв'язки об'єктів, а не додаткову інформацію про предметну область; людина може розібратися, навіщо потрібне те чи інше гіперпосилання, але для комп'ютера цей зв'язок не зрозумілий;

2) сторінки, що зв'язуються гіперпосиланнями, є документами, що описують, як правило, проблемну ситуацію в цілому; у семантичній мережі вершини, зв'язані відношеннями, представляють собою поняття або об'єкти реального світу.

Спроба створення семантичної мережі на основі Всесвітньої павутини отримала назву семантичної павутини. Ця концепція припускає використання мови RDF (Resource Description Framework-мови розмітки на основі XML) і покликана надати посиланнями якийсь сенс, зрозумілий комп'ютерним системам. Це дозволить перетворити Інтернет у розподілену базу знань глобального масштабу

Приклади семантичних мереж

Приклад 1. Семантична мережа (рис. 4.20) описує відомості про те, що людина Ткачук захоплюється автомобілями і йому належить автомобіль Ford білого кольору з номерним знаком AP0007AX.

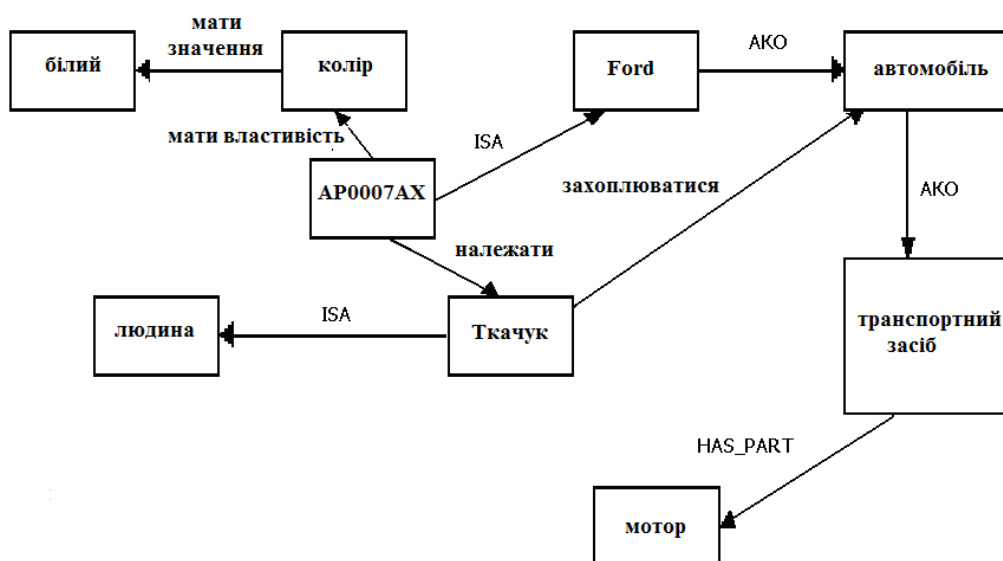


Рисунок 4.20 - Приклад семантичної мережі

Приклад 3. У прикладі рис. 4.21 описані відомості про подружні пари. Подвійна стрілка позначає відношення ISA (є представником), потрійна - АКО (є видом), пунктиром - нефундаментальное відношення, характерне для даної предметної області

Можна скласти мережу-запит (рис. 4.22). Така мережа містить як відомі вершини і дуги, так і невідомі. Мета запиту - визначити невідомі вершини і дуги шляхом зіставлення основної мережі із зразком (мережею-запитом).

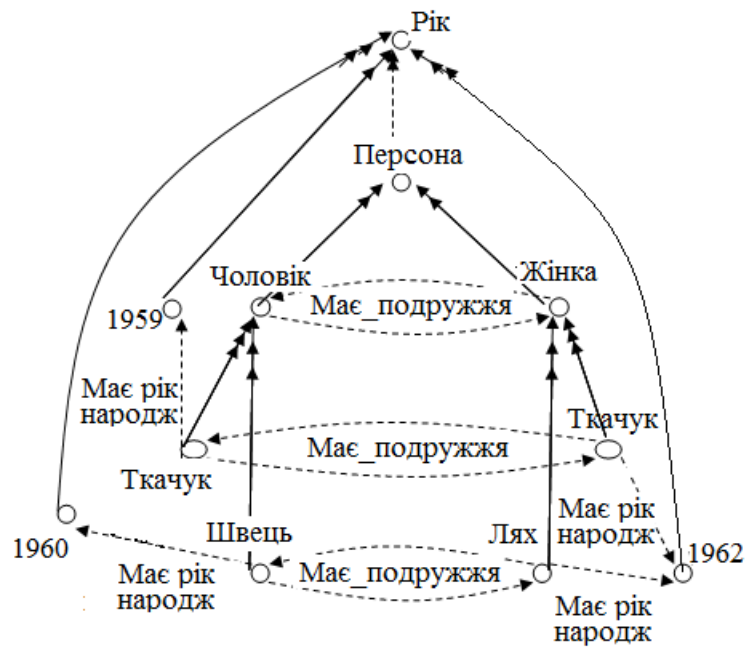


Рисунок 4.21 - СМ для опису відомостей про подружні парах

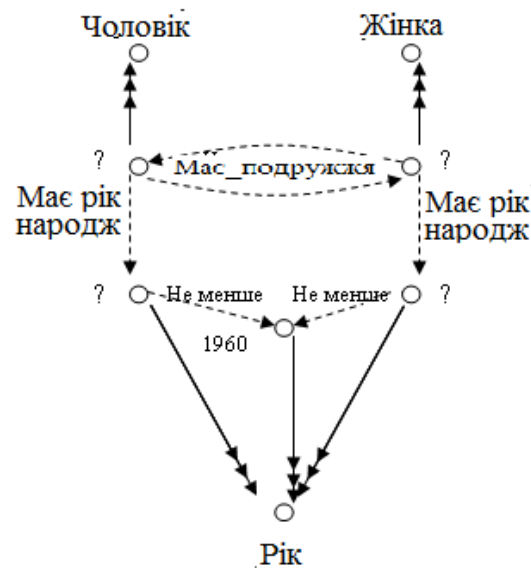


Рисунок 4.22 – Приклад мережі-запиту

Приклад 4. Семантична мережа рис. 4.23 описує сенс англійської фрази On Wednesday morning John hit Mary in the park by the fist у вигляді набору вузлів, відповідних об'єктів або концептуальних понять, які пов'язані спрямованими зв'язками.

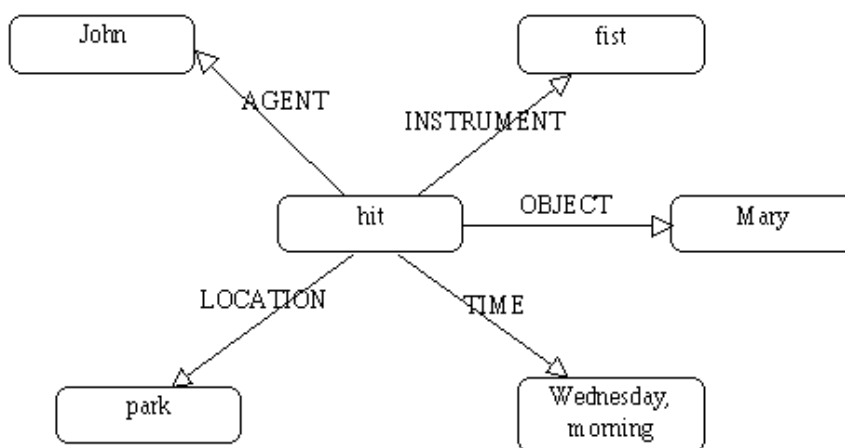


Рисунок 4.23 - СМ для фрази «On Wednesday morning John hit Mary in the park by the fist»

Приклад 5. При побудові семантичної мережі для деякого освітнього розділу виникає ієрархічна структура з трьох рівнів. Структура моделі показана на рис. 4.24

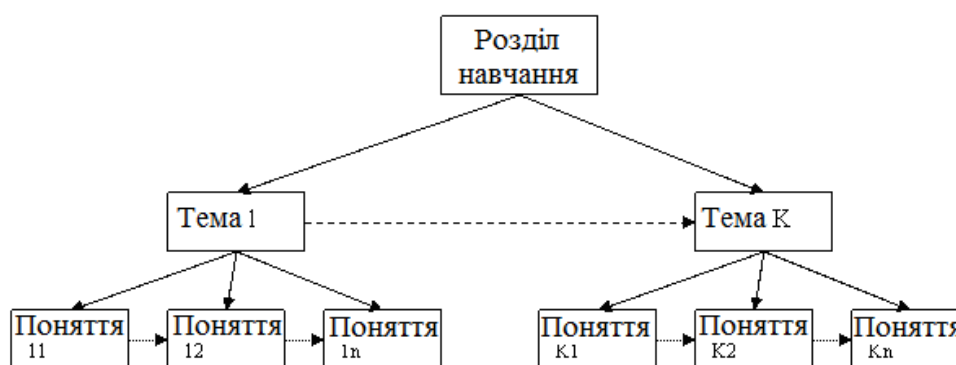


Рисунок 4.24 - Семантична мережа для деякого освітнього розділу

Верхній рівень - рівень самого розділу. Для цього рівня вводяться наступні характеристики: Оцінка часу, необхідного для вивчення, форма контролю.

Наступний рівень - рівень тем. На цьому рівні вводиться відношення порядку, що відображає послідовність вивчення тем.

Нижній рівень моделі - рівень понять. Цей рівень відображає мінімальні одиниці розділу (терміни, аксіоми, теореми і т.д.). На цьому рівні вводиться відношення часткового порядку.

4.6.3 Фреймова модель представлення знань

Велика частина нашої повсякденної діяльності відбувається за стереотипними ситуаціями (шлях на роботу, готування їжі, здійснення покупки і т.д.). Стереотипні ситуації мають місце у процесі функціонування складних об'єктів, в тому числі і виробничих. Для представлення та опису стереотипних

об'єктів, подій або ситуацій були введені поняття "фрейми", які є складними структурами даних.

Фрейми були вперше запропоновані в якості апарату для представлення знань М. Мінським в 1975 р. Згідно з його визначенням фрейми – це мінімальні структури інформації, необхідні для подання класу об'єктів, явищ або процесів. У загальному вигляді фрейм формально може описаний рядком:

$$\{FN, (SN, SV, CP), \dots, (SN, SV, CP)\},$$

де: *FN* - ім'я фрейма;

SN - ім'я слота;

SV - значення слота;

CP - ім'я приєднаної процедури (необов'язковий параметр).

Слоти - це деякі незаповнені підструктури фрейма, заповнення яких призводить до того, що даний фрейм ставиться у відповідність деякій ситуації, явищу чи об'єкту.

З кожним фреймом пов'язана інформація:

- як використовувати фрейм;
- що робити в непередбаченій ситуації;
- як отримати невідомі значення слотів.

Фрейм із заповненими слотами називається екземпляром фрейму. Для організації зв'язку між об'єктами предметної області будується мережа фреймів. Зв'язок може бути організований шляхом задання в якості значень деяких слотів одного фрейму імен інших фреймів.

В якості даних фрейм може містити звернення до процедур (так звані приєднані процедури). Виділяють два види процедур: процедури-демони і процедури-слуги. Процедури-демони активізуються при кожній спробі додавання або видалення даних за замовчуванням. Процедури-слуги активізуються тільки при виконанні умов, визначених користувачем при створенні фрейму.

Для зменшення інформаційної надлишковості у фреймових системах реалізують принцип успадкування інформації, що дозволяє спільну (глобальну) для системи інформацію зберігати в окремому фреймі, а у всіх інших фреймах вказувати лише посилання на місце зберігання цієї інформації.

У рамках фреймового підходу передбачається, що знання в системі представляються у вигляді окремих кластерів знань, або підструктур, що містять відомості про стереотипи (тобто про деякі загальні характеристики даного класу об'єктів або ситуацій). Відповідно до даного припущення розуміння ситуації для системи означає пошук в переліку накопичених структур такої, що найкращим чином описувала б розглянуту ситуацію. При цьому слоти заповнюються деякою інформацією і заповнений фрейм перевіряється на адекватність даній ситуації. У випадку розбіжності шукається новий фрейм і процес триває.

Таким чином, можна виділити три основні процеси, що відбуваються у фреймових системах.

1. Створення екземпляра фрейму. Для створення екземпляра фрейму необхідно знайти відповідний фрейм і заповнити його слоти інформацією, що описує специфіку ситуації, що розглядається.

Для заповнення слотів використовується спеціальна інформація про те, як знайти потенційні "наповнювачі" слотів. Ця інформація часто зберігається в процедуральній формі.

2. Активація фреймів. Коли фрейм вважається придатним для опису певної ситуації, здійснюється його активація глобальним процесом. Якщо виявляється занадто багато відмінностей значень слотів фрейму від специфічних особливостей ситуації, що розглядається або вони мають досить серйозний характер, організується пошук іншого, більш відповідного фрейму. При цьому "знехтуваний" фрейм може містити вказівки на те, які саме фрейми потрібно досліджувати замість даного (наприклад, більш загальні або, навпаки, більш спеціалізовані). Частина даних, що використовуються для заповнення слотів "відкинутого" фрейма, може бути використана при розгляді нових кандидатів.

3. Організація вводу-виводу, що полягає в послідовному пошуку та активації в мережі фреймів до знаходження найбільш відповідного і побудови на його основі екземпляра фрейму.

У фреймах об'єднуються переваги декларативного і процедурального подання знань. Знання, що стосуються функцій безпосереднього представлення їх за допомогою фреймів, повинні зберігатися в декларативній формі, а знання про використання фреймів - в процедуральній.

Зокрема, процедури можуть зберігати знання, що дозволяють дати відповідь на питання:

1) "Коли активувати фрейм?" – подібно "демонам" фрейми можуть активувати самі себе у випадку, якщо розпізнана відповідна ситуація;

2) "У якому випадку вважати, що даний фрейм неадекватний ситуації і що в цьому випадку робити?" – фрейм міг би, наприклад, автоматично передати управління іншому фрейму або деактивувати себе;

3) "Коли здійснювати заповнення слотів - у момент дзвінка або пізніше, в міру необхідності?" – реалізація цих функцій може бути покладена на приєднані процедури. Процедури можуть також реалізовувати евристики, спрямовані на пошук інформації, необхідної для заповнення слотів.

Характерні особливості фрейм-підходу до проблеми подання знань

Одним з можливих нових шляхів організації машинної моделі реального світу є підхід, запропонований М. Мінським. Відповідно до цього підходу знання про світ – машинна модель реального світу – повинні бути представлені в пам'яті комп'ютера у вигляді достатньо великої сукупності певним чином структурованих даних, що представляють собою стереотипні ситуації. Ці структури мають назву "фрейми".

У разі виникнення конкретної ситуації, наприклад, необхідності здійснити роботом, керованим комп'ютером, певну дію, необхідно сприйняти за допомогою сенсорів, пов'язаних з комп'ютером, певний візуальний образ.

Далі з пам'яті комп'ютера повинен бути вибраний фрейм, який відповідає даному класу ситуацій. Цей фрейм має пройти узгодження з розглянутою конкретною ситуацією шляхом конкретизації слотів та зіставлення їх, з даними, що містяться у фреймі-ситуації.

Так як фрейм можна уявити собі у вигляді мережі, що складається з вузлів і зв'язків між ними, то кожен вузол повинен бути заповнений своїм "завданням", що представляє собою ті чи інші характерні риси ситуації, якій він відповідає. У загальному випадку у фреймі можна виділити кілька рівнів, ієрархічно пов'язаних один з одним. Вузли фрейму, що належать до верхніх рівнів, являють собою більш загальні речі, які завжди справедливі відносно ситуації, що розглядається. Ці вузли вже заповнені своїми завданнями. Наприклад, вузол фрейму самого верхнього рівня зазвичай заповнений назвою ситуації, тобто. назвою візуального образу (це може бути, наприклад, "куб"), назвою дії (наприклад, "прибирання кімнати"). Вузли нижніх рівнів здебільшого не заповнені своїми завданнями. Такі незаповнені вузли називають терміналами. Вони повинні бути заповнені конкретними даними, що представляють собою їх можливі завдання в процесі пристосування фрейму до конкретної ситуації, з того класу ситуацій, який надає даний фрейм. Кожен термінал може встановлювати умови, яким повинні відповідати його завдання. Прості умови встановлюються "маркерами", які можуть вимагати, наприклад, щоб завданням терміналу було обличчя, предмет достатньої величини, елементарна дія або вказівник на інший фрейм, що представляє собою іншу, зазвичай більше детальну ситуацію, і називається субфреймом. Більш складні умови можуть встановлювати зв'язки між завданнями для декількох терміналів.

Група фреймів може об'єднуватися в систему фреймів. Результати характерних дій відображаються за допомогою трансформацій між фреймами системи. Вони використовуються, щоб прискорити обчислення певних видів при поданні типових змін одній і тій же ситуації.

У разі візуального образу різні фрейм-системи описують картину з різних точок спостереження, а трансформація одного фрейму в інший відображає результати переміщення спостерігача з одного місця в інше.

Для фреймів невізуальних видів відмінності між фрейм-системами можуть відображати дії, причинно-наслідкові зв'язки і зміни понятійної точки зору. Різні фрейм-системи можуть використовувати одні й ті ж термінали. Це важлива обставина, завдяки якій, зокрема, економиться обсяг пам'яті комп'ютера, що використовується для побудови моделі реального світу.

Характерною рисою описаного підходу є можливість використання різних видів прогнозів, очікувань, припущень. Відповідно до цього термінали фрейму, що обирається для представлення ситуації, звичайно вже заповнені завданнями, які є найбільш вірогідними в даній ситуації. Ці завдання називаються "завданнями відсутності".

Таким чином, фрейм може містити велику кількість деталей, які можуть і не підтвердитися даною ситуацією. Завдання відсутності "неміцно" пов'язані зі своїми терміналами, тому вони можуть бути легко "витиснуті" іншими завданнями, які краще підходять до поточної ситуації.

Після того як обрано фрейм для представлення ситуації, процес узгодження фрейму з даною конкретною ситуацією полягає у знаходженні таких завдань для терміналів фрейма, які сумісні з маркерами терміналів. Процес узгодження частково контролюється інформацією, пов'язаною з фреймом (до якої входить і інформація щодо того, як діяти у випадку появи незвичайних ситуацій, "сюрпризів"), а частково знанням поточних цілей.

Якщо вибраний фрейм не вдається узгодити з реальністю, тобто. якщо неможливо знайти завдання для терміналів, які відповідним чином узгоджуються з умовами маркера, то відбувається звернення до так званої мережі пошуку інформації, за допомогою якої з'єднуються між собою системи фреймів. Ця мережа дозволяє знайти інші способи представлення знань про факти, аналогії та іншої інформації, яку можна використати для узгодження з реальністю.

Теорія представлення знань за допомогою фреймів претендує на пояснення ряду характерних особливостей людського мислення. Вважають, що вона дозволяє охопити єдиною концепцією такі, здавалося б, різні теорії, як розуміння природної мови, машинного "сприйняття" візуальних образів, пошуку рішень, планування, у тому числі стосовно до задач керування роботами. Вона об'єднує багато класичних та сучасних ідей психології, лінгвістики, а також штучного інтелекту. Зокрема, ця теорія узагальнює ідеї, висловлені в ряді відомих робіт з штучного інтелекту, у яких знання про світ представлені за допомогою просторів підзадач, де модель світу представляється простором "сценаріїв", та в яких пропонується підрозділити знання на "мікросвіту".

У своїй теорії М.Мінський не проводить межу між теорією людського мислення і теорією побудови "думаючої" машини (штучного інтелекту). Він вважає, що і процес мислення людини заснований на наявності в його пам'яті якимось чином матеріалізованого величезного набору різноманітних фреймів, за допомогою яких людина усвідомлює зорові образи (фрейми візуальних образів), розуміє слова (семантичні фрейми), міркування, дії (фрейми-сценарії), оповідання (фрейми-розповіді) і т.д. Процес розуміння при цьому супроводжується вибором з пам'яті відповідного фрейму, у якого термінали вже заповнені завданнями відсутності, і пристосуванням його до поточної ситуації. Якщо це не вдається, то з пам'яті вибирається новий більш відповідний до ситуації фрейм. У випадку, коли і цей фрейм не досить добре узгоджується з ситуацією і пошуки нового не приводять до успіху, відбувається пристосування найбільш відповідає ситуації фрейму, який було виявлено в процесі пошуку.

Фрейм-візуальний образ

В якості найпростішого прикладу, що ілюструє подання знань за допомогою фреймів, розглянемо можливу систему фреймів для елементарного зорового образу – куба. Відповідно до подання тіл правильної форми з допомогою "областей" і "зв'язків" між ними можна допустити, що результатом розглядання куба є структура, подібна наведеній на рис. 4.25 (а). Цю структуру можна ідентифікувати з фреймом куба при погляді на нього з відповідної

позиції. Области A , E і B є терміналами фрейму, завдання для яких відповідають можливим деталям або позначенням на видимих з даної позиції гранях куба. Якщо позиція погляду на куб переміщується вправо, то грань A зникає з поля зору і стає видимою грань C .

Якби потрібно було провести повний аналіз цього нового візуального образу, необхідно було б:

- 1) втратити знання про межі A ;
- 2) повторно сприйняти образ грані B ;
- 3) сприйняти образ нової грані C .

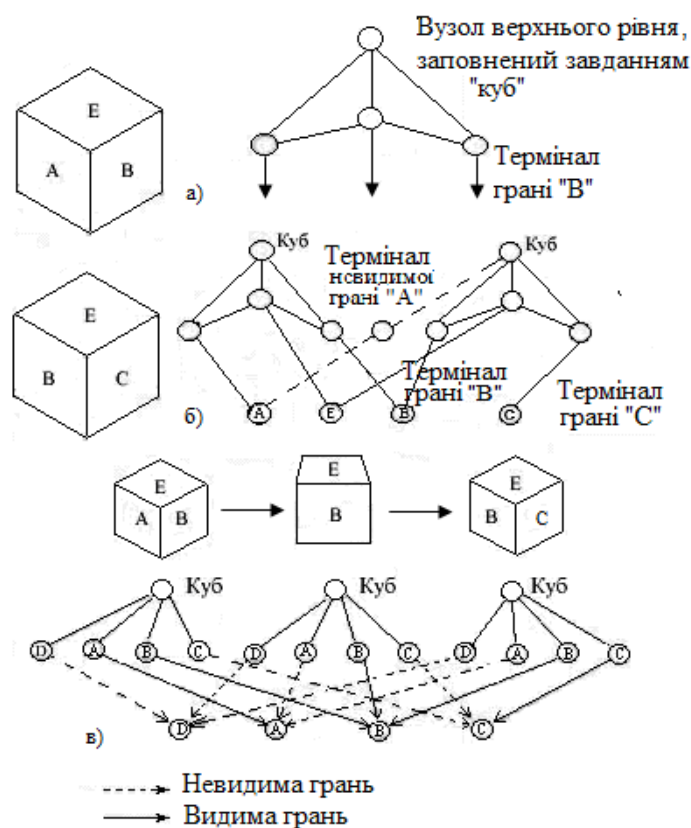


Рисунок 4.25 – Фрейм «Куб»

Однак, оскільки відомо, що відбулося переміщення позиції спостереження вправо, можна зберегти знання про грань B у вигляді задання терміналу лівій грані нового фрейму куба, відповідного нової позиції спостереження. Крім того, щоб зберегти знання і про грань A , можна ввести додатковий термінал невидимої грані, що відноситься до цього нового фрейму.

При поверненні на початкову позицію спостереження виявляється можливим відновити візуальний образ куба без будь-яких нових "обчислень". Для цього достатньо "викликати" з пам'яті перший фрейм. Очевидно, корисно зберегти знання і про грань C , для чого можна ввести додатковий термінал цієї невидимої межі в першому фреймі.

Можна продовжити цю процедуру побудови системи фреймів, відповідну переміщенню точки спостереження навколо куба. Це призвело б до отримання

більш широкої системи фреймів, в якій кожен фрейм відповідає своїй позиції спостереження куба. На рис. 4.25(в) показана система фреймів, що складається з трьох фреймів, кожен з яких представляє візуальний образ, що виникає в одній з трьох позицій спостереження. Дві з цих позицій відповідають переміщенню вправо і вліво на 45° щодо третьої позиції; покажчики між фреймами відповідають переміщенням точки спостереження. Важливо звернути увагу на виявлену в цьому прикладі важливу властивість подання ситуації за допомогою системи фреймів. Вона полягає в тому, що різні фрейми, що входять в систему, використовують один і той же термінал, відповідний однієї і тій же фізичній межі, яку видно з різних позицій спостереження. Це дає можливість зосередити в одному місці інформацію про властивості відомих об'єктів незалежно від позицій спостереження, яких, особливо для предметів складної форми, може бути дуже багато. В результаті заощаджується пам'ять і скорочується процес сприйняття при зміні позицій спостереження, так як пам'ять вже має в своєму розпорядженні необхідною інформацією і час витрачається лише на "вилучення" її з пам'яті.

Самі ж системи фреймів, мабуть, сформовані в пам'яті не для візуальних образів кожного можливого предмета, а для "основних форм", які, вступаючи в різні комбінації, утворюють системи фреймів для нових випадків. Це створює додаткові можливості економії пам'яті. Так само, як і у випадку окремих заздалегідь сформованих терміналів, що належать фрейму, наявність у пам'яті заздалегідь заготовленого набору систем фреймів прискорює процес сприйняття, так як новий образ не доводиться будувати заново, а тільки отримувати його з пам'яті і "приспосовувати" до дійсності.

Фрейм-сценарій

Зазначені властивості, що дозволяють економити пам'ять і час сприйняття, очевидно, можуть проявлятися і при поданні невізуальних знань про світ. Дійсно, системи невізуальних фреймів, наприклад описуваних нижче фреймів-сценаріїв, можна представити як одне з можливих розумінь предмета обговорення. Наприклад, електрогенератор можна представити як механічну і як електричну системи. Будь-які знання про світ можна уявити собі у вигляді деяких загальних універсальних фрагментів, яким відповідають свої системи фреймів і з яких можна формувати нові системи фреймів, відповідні деяким новим уявленням.

Представлення знань про світ за допомогою фреймів виявляється досить плідним при поясненні механізмів розуміння людиною природної мови, міркувань, оповідань, спостережуваних дій іншої особи і т. д. Воно може бути корисним і при розробці штучних механізмів реалізації перерахованих вище функцій за допомогою ЕОМ.

У роботі М.Мінського в цьому випадку пропонується будувати знання про світ у вигляді фреймів-сценаріїв. Фрейм-сценарій представляє собою типову структуру для певної дії, поняття події і т.п., що включає характерні елементи цієї дії, поняття, події. Наприклад, фрейм-сценарій для події, що

складається з святкування дня народження дитини, включає наступні елементи, які можна трактувати як вузли фрейму, заповнені завданнями відсутності:

Одяг: святковий, найкращий;

Подарунок: повинен сподобатися.

Для пояснення швидкого розуміння людиною ситуації, яка надається сценарієм, пропонується ототожнювати термінали фрейма-сценарія з найбільш характерними питаннями, звичайно пов'язаними з цією ситуацією. Відповіді на ці питання корисно отримати для розуміння даної ситуації. По суті фрейм-сценарій в цьому випадку є набором питань, які потрібно поставити щодо деякої гіпотетичної ситуації, і способів відповіді на них.

Для того щоб збагнути дію, про яку йде мова або яка спостерігається, людина часто змушена задати такі запитання:

"Хто здійснює дію (*агент*)?",

"Яка мета дії (*намір*)?",

"Які наслідки (*ефект*)?",

"На кого ця дія впливає (*одержувач*)?",

"Яким чином дія виконується (*інструмент*)?".

Щодо розуміння речей, що відрізняються від дій, задають дещо інші запитання, і ці запитання можуть бути значно менше локалізовані, ніж у разі розуміння дій, наприклад: "Чому вони говорять це мені?", "Яким чином я можу з'ясувати більше про X?", "Яким чином це допоможе у вирішенні проблеми?" і т.д. При розгляді оповідання запитують, яка тема, яке ставлення автора, яка головна подія, хто є головним героєм і т. д. По мірі того, як на кожне запитання дається попередня відповідь, з пам'яті можуть викликатися нові фрейми, відповідні ситуації, які виникають в результаті відповідей на запитання. Запитання-термінали цих нових фреймів стають у свою чергу активними.

Слід зазначити, що кількість питань, пов'язаних з фреймом, є невизначеною, і на перший погляд здається, що для розуміння ситуації їх може бути дуже багато. Однак на практиці виявляється достатнім задати дуже мало питань, щоб розібратися в ситуації.

Різні люди можуть задавати різне число питань щодо однієї і тієї ж ситуації. Число і характер цих питань у великій мірі залежать від бази знань щодо обговорюваного об'єкта у того чи іншого індивідуума. Може виявитися, що повне розуміння не буде досягнуто через відсутність у людини необхідної системи фреймів, що об'єднують знання про обговорюваний предмет.

Проте, коли необхідна база знань існує, а щодо повсякденних понять, дій, міркувань вона є в будь-якої людини, то для розуміння ситуації звичайно достатньо не дуже великої кількості питань. Це дуже принципова обставина, завдяки якій на інтуїтивному рівні ми приходимо до досить обнадійливого для практики положення, що полягає в тому, що велика кількість реальних ситуацій, що виникають при розумінні мови, міркування, дії можна зрозуміти, отримавши відповіді на обмежене коло питань. Це побічно підтверджує можливість побудови відносно простих моделей світу в даному випадку за допомогою фреймів, достатніх для прийняття рішень в цих ситуаціях.

У разі фреймів-сценаріїв маркери терміналів фрейму стають більш складними, ніж це було у випадку фреймів візуальних образів, і визначають рекомендації щодо того, яким чином треба відповідати на запитання, тобто, заповнювати термінал завданням. Кожен термінал повинен містити рекомендації щодо того, яким чином знайти його завдання – відповідь на запитання. Завдання відсутності або перелік можливих відповідей на питання є найпростішими особливими випадками таких рекомендацій. Мабуть, людина може мати ієрархічний набір таких рекомендацій подібно схемами переваги.

Повне і всебічне представлення кожної ситуації типу подія, дія, міркування і т.п. подібно поданням візуального образу і здійснюється за допомогою не одного, а системи фреймів. Кожен фрейм, системи відповідає одній з можливих точок зору на ситуацію, що представляється системою фреймів, подібно до того, як один з візуальних фреймів системи представляє візуальний образ з деякою однієї точки спостереження. Різні фрейми системи представляють різні шляхи використання однієї і тієї ж інформації, що знаходиться на загальних терміналах. Як і в "візуальній ситуації", людина, розуміючи або передаючи думку, "вибирає" один з фреймів. Цей вибір "по суті" полягає у виборі питань, які, треба задати щодо ситуації, що розглядається.

В одній з інтерпретацій сценарій являє собою послідовність дій, які описують ситуації, що часто зустрічаються. У цій послідовності дій використовується принцип каузальних зв'язків, тобто результатом кожної дії є умови, за яких може відбутися наступна дія.

Кожен сценарій має виконавців ролей. Сценарій має різні інтерпретації, що відображають точки зору різних виконавців. Таким чином, його можна розглядати як систему фреймів.

Кількість сценаріїв, що відбивають можливі ситуації, які зустрічаються в реальному житті, величезна. Сюди належать і день народження дитини, і гра в футбол, заняття в школі і т. д.

Оскільки відповідно з висловленою раніше положенням процес розуміння людиною реальної ситуації є вибором з пам'яті і пристосуванням до цієї ситуації відповідного фрейма, виникає питання про можливий механізм цього вибору.

Спосіб формалізації фреймів

Один з можливих варіантів формалізації фрейму-сценарію припускає подання його у вигляді мережі ієрархічної структури (рис. 4.26)

Вузол самого верхнього рівня мережі (на рисунку йому відповідає коло, обведене жирною лінією) ототожнюється із заголовком сценарію. Дочірні вершини цього вузла, позначені на рисунку прямокутниками, є терміналами фрейму. Вони ототожнюються з набором питань типу "а що якщо ..." або в загальному випадку з набором будь-яких інших питань - тестом. Кожне з можливих завдань терміналу, що є елементарною дією і позначене на рисунку колом усередині прямокутника-терміналу, відповідає значенню тесту, тобто. сукупності відповідей на набір питань. У найпростішому випадку є кінцевий перелік можливих відповідей, який і визначає маркер терміналу.

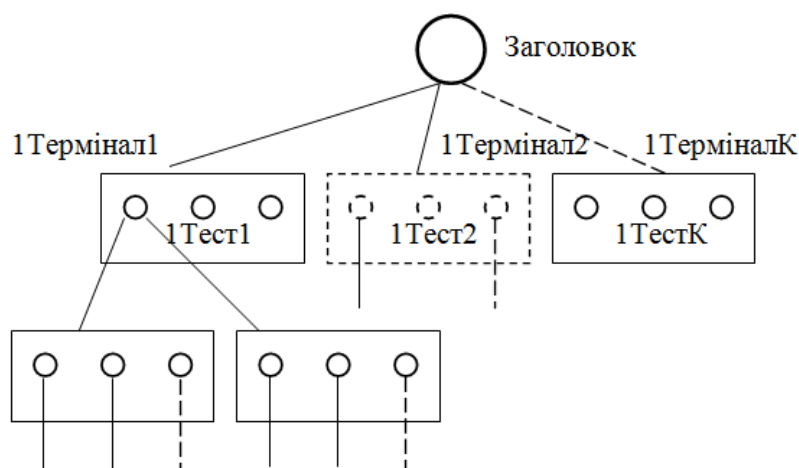


Рисунок 4.26- Формалізація фрейму-сценарію

Кожне з можливих завдань терміналу розглядається в свою чергу як фрейм наступного за рангом рівня (субфрейм) зі своїми терміналами, які представляються на рисунку дочірніми вершинами завдань - субфреймами та позначаються прямокутниками. Кожен з цих терміналів нижчого рангу так само, як і термінали фрейму верхнього рангу, ототожнюється зі своїм тестом. Завдання терміналів наступного за рангом рівня, також позначені кружками, є більш дрібними і більш конкретизованими елементарним "діями". Кожне з них відповідає значенню тесту. Якщо й ці завдання ще не є достатньо конкретними, то фрейм може мати ще один більш низький за ієрархією рівень.

Неважко переконатися, що описана структура зводиться до звичайного графа *I/АБО*, якщо ототожнити термінали фреймів і субфреймів, позначені на рис. 4.26 прямокутниками, з вершинами типу *I*, завдання цих терміналів - з вершинами *АБО*, а кожній дузі, що йде від вершини *I* до *АБО*, поставити у відповідність значення тесту. На рис. 4.27 дано подання раніше представленого фрейму (рис. 4.26) у вигляді графа *I/АБО*. Зауважимо, що при такій формалізації дочірніми вершинами для верхнього вузла будуть вершини типу *I*.

Термінали фрейму і субфреймів у пропонованій формалізації можна в більшості випадків ототожнювати не тільки з тестом, але і з дією, яка є узагальненням всіх дій, що визначаються можливими завданнями терміналі. При такому поданні всі вершини *I/АБО* графа, що формалізують фрейм, відповідають діям, причому тим більше конкретизованим, чим нижче за ієрархією вершина. Часто корисним виявляється використання узагальнення графа *I/АБО*, що описує фрейм-сценарій, яке виходить за рахунок подання вершинами графа не дій тієї чи іншою мірою спільності, а так званих "схем дій". Ці схеми визначають лише загальну структуру дії, тобто. в них можуть бути не визначені конкретно або діючі особи, або засоби дії, або місце дії і т. п., або, нарешті, і те й інше разом. Наприклад, можлива наступна схема дії: *X* прибути в *V*, де *X* може бути будь-яким конкретною особою, а *V* - будь-яким конкретним місцем. Чим нижче по ієрархії вершина, тим більш конкретно визначає вона дію.

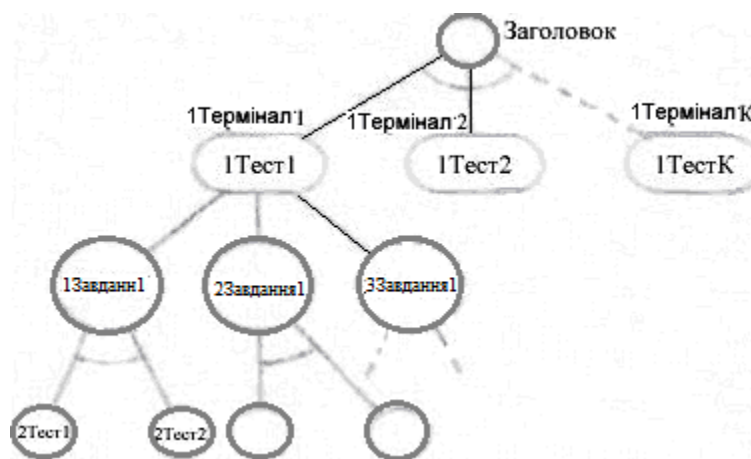


Рисунок 4.27 - Подання фрейму у вигляді I/O графа

Важливо підкреслити, що, якщо база знань про світ утворена сукупністю формалізованих фреймів-сценаріїв, то дуже ймовірна можливість багаторазового входження одних і тих же схем дій в різні фрейми-сценарії. Ця ймовірність тим більше, чим більша кількість фреймів входить до бази знань. Для використання цієї обставини з метою економії пам'яті корисно зв'язати кожен схему дії з кожним своїм входженням у фрейми, що утворюють базу знань, за допомогою "множини входжень", яке представляє вказівники на всі ті місця в базі знань, де є посилання на дану схему. Крім того, кожна схема дії пов'язана з вищою за рівнем схемою.

Механізми "пристосування" фрейму до реальної ситуації

Процес вибору фрейму з пам'яті і пристосування його до реальної ситуації лежить в основі розуміння людиною реальної ситуації. У разі ж машинної бази знань цей процес відкриває доступ до знань, матеріалізованих в пам'яті комп'ютера у вигляді сукупності систем фреймів.

Цей механізм приводиться в дію двома цілями, що доповнюють одна одну:

1) необхідність знаходження завдань терміналів фрейму, який задовольняє маркерами цих терміналів;

2) розглянутий фрейм має задовольняти маркерами терміналу більш загального фрейму, куди входить перший; кожен фрейм вважається пристосованим до ситуації, якщо він включений у більший фрейм як завдання його терміналу і якщо його термінали заповнені завданнями, що задовольняють маркерами.

У процесі розуміння міркування, розповіді, так само, як і сприйняття образу, ключові слова, ідеї міркування, елементи образу викликають з пам'яті різні сукупності фреймів, більшість терміналів яких ще не є узгодженими з реальністю, а заповнені завданнями відсутності. У міру надходження нової інформації з'ясовується, що деякі попередньо вибрані з пам'яті субфрейми не узгоджуються з реальністю. Вони замінюються іншими більш придатними субфреймами, що задовольняють двом вищенаведеним умовам.

У найпростішому випадку така заміна здійснюється шляхом так званої операції узгодження. Вона має місце, коли відсутні особливі знання щодо того, як чинити при зміні фрейму, крім наявної деякої загальної стратегії. Ця стратегія полягає у виборі після надходження чергової порції вхідної інформації такого фрейму, для якого вся інформація, що надійшла раніше, наприклад текст у разі розуміння мови, і порція нової інформації задовольняють маркерами його терміналів.

Приклади фреймів

На рисунках 4.29–4.32 наведено приклади фрейм-структур.

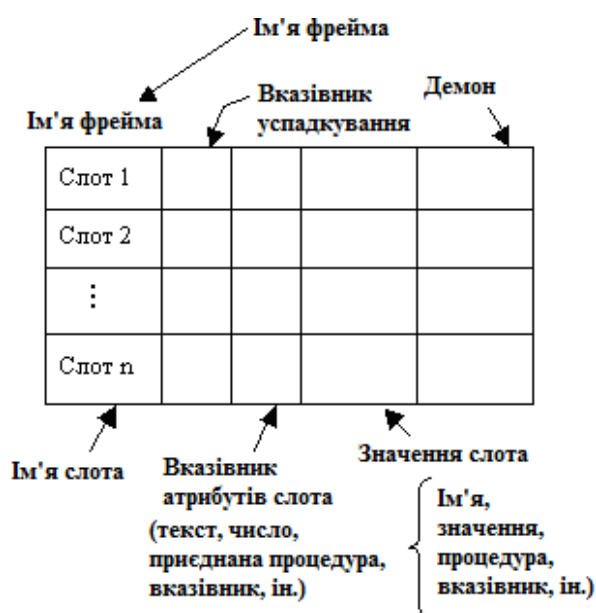


Рисунок 4.29 – Структура даних "фрейм"

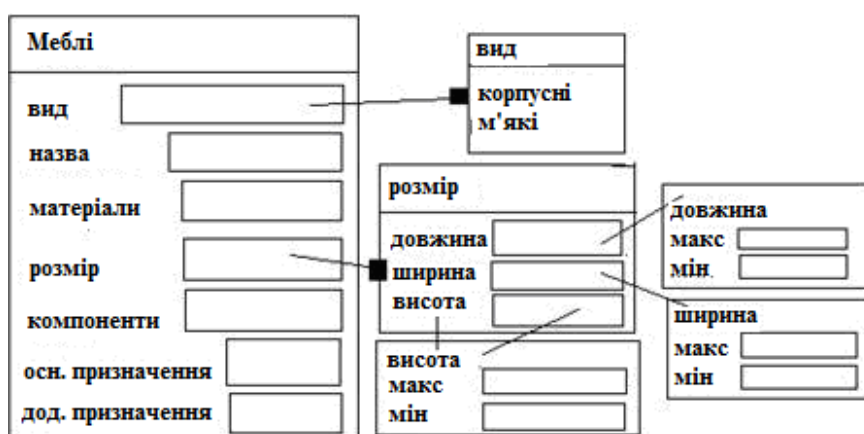


Рисунок 4.30 – Фреймове подання меблів

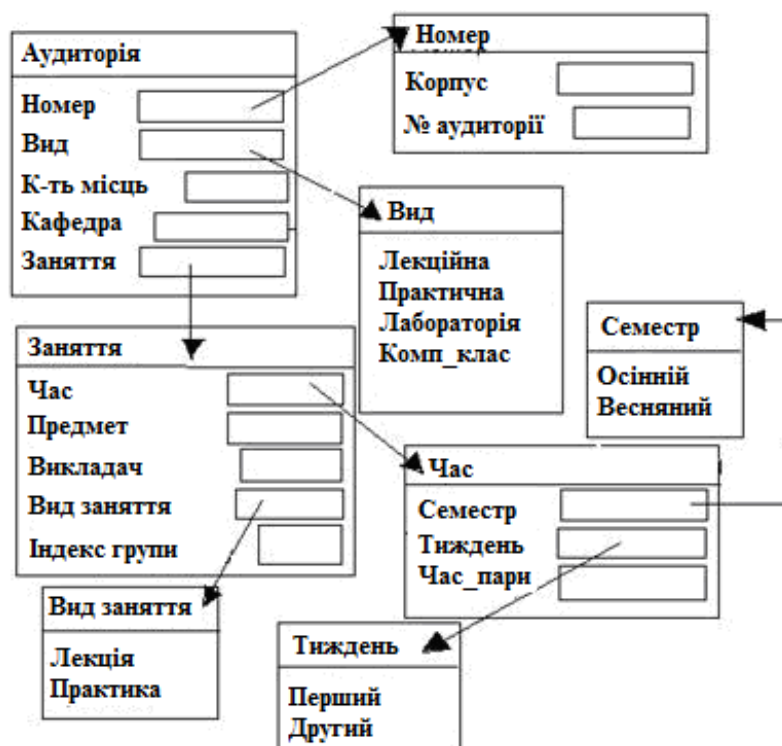


Рисунок 4.31 – Фреймова структура "Проведення занять"

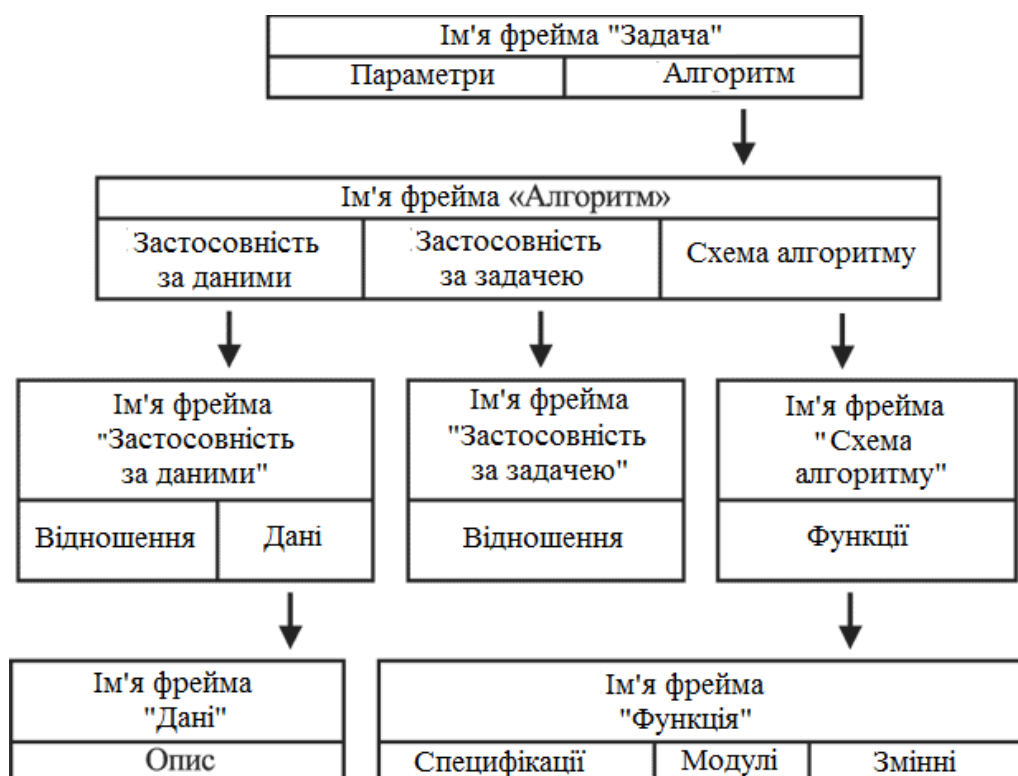


Рисунок 4.32 – Фреймова структура "Задача обчислювального типу".

4.6.4 Логічна модель представлення знань

Поняття логічної моделі знань

В основі логічної моделі знань лежить формальна теорія і поняття формальної системи. Відношення, які існують між одиницями знань, можна описувати тільки за допомогою синтаксичних правил, допустимих в рамках цієї теорії.

Формальна система задається завжди четвіркою символів

$$S = \langle B, F, A, R \rangle,$$

- де
- B – кінцева множина базових символів, інакше – алфавіт системи S ;
 - F – підмножина виразів системи S , званих формулами системи. Зазвичай є ефективна процедура, яка являє собою сукупність правил, які дозволяють з елементів множини U будувати синтаксично правильні вирази (тобто. ППФ - правильно побудовані формули);
 - A – виділена множина правил, аксіом системи, тобто множина апіорно істинних формул;
 - R – кінцева множина відношень $(r_1, r_2, \dots, r_n)$ між формулами, правилами виведення. Для будь-якого r_i існує ціле позитивне число j , таке, що для кожного безлічі, що складається з j формул, і для кожної формули F ефективно вирішується питання про те, чи знаходяться ці j -формули відносно r_i з формулою F . Якщо r_i виконується, то F називають безпосереднім наслідком F -формул за правилом r_i .

Наслідком (висновком) формули в теорії S називається така послідовність правил, що для будь-якого з них представлена формула є або аксіомою теорії S , або безпосереднім наслідком.

Правила виведення, які розробляються проектувальниками, дозволяють розширити множину формул, які є аксіомами теорії.

Формальна теорія називається розв'язуваною, якщо існує ефективна процедура, що дозволяє дізнатися для будь-якої заданої формули, чи існує її висновок в теорії S .

Формальна теорія S називається несуперечливою, якщо не існує такої формули A , що і A , і не A що виводяться в цій теорії.

Найбільш поширеною формальною теорією, що використовується в системах штучного інтелекту, є числення предикатів, тобто функцій, які можуть приймати тільки 2 значення – "істина" або "хиба".

До переваг логічної моделі належать:

- наявність стандартної типової процедури логічного висновку (докази теорем);
- гарантія несуперечності висновку;
- гарантія алгоритмічної розв'язності (для обчислення висловлювань);
- гарантія напіврозв'язуваності (для числень предикатів першого порядку).

Проте така уніфікація тягне за собою основний недолік моделі - складність використання в процесі логічного виведення евристик, що відбивають специфіку предметної області (ПО).

До інших недоліків логічної моделі відносять:

- "Монотонність";
- "Комбінаторний вибух";
- Слабкість структурованості описів.

Характеристика мови предикатів першого порядку. Особливості подання знань у логічній моделі

В основі мови предикатів першого порядку лежить поняття предикатів, тобто логічна функція від однієї або декількох атомарних змінних. Функція може приймати значення істина (t) або хиба (f). У рамках логіки твердження вважається дійсним, якщо припущення, яке воно висловлює, вважається істинним і висновок самого твердження теж є істиною.

Синтаксис мови предикатів включає: предикативні символи, символи змінних, константи, а також роздільники $()$, $[]$, $;$, $.$

Предикативні символи використовуються для позначення відношень. Об'єкти відношень записуються в круглих дужках після предикативного символу і називається аргументами. Повний запис відношення називається атомарною формулою:

Атомарна формула:

предикативний_символ(терм_1, терм_2)

Терми можуть представлятися константами та змінними. Дозволено також як терми використовувати функції, які обов'язково повинні бути визначені в рамках предметної області. Проектувальник ЕС заздалегідь визначає, як інтерпретувати порядок термів у відношенні.

Допустимі вирази в численні предикатів, зокрема атомарні формули, називається правильно побудованими формулами (ППФ). У мові предикатів для кожної ППФ обов'язково визначається конкретна інтерпретація. Як тільки для ППФ визначена інтерпретація, кажуть, що формула має значення "істина", якщо відповідне затвердження предметної області істинно, в іншому випадку ППФ має значення "хиба".

З формул можна скласти речення за допомогою логічних зв'язок: кон'юнкція, диз'юнкція, імплікація, заперечення.

Кон'юнкція використовується для утворення складених фраз.

ППФ, побудовані за допомогою зв'язки "кон'юнкція", називаються просто *кон'юнкція*.

Диз'юнкція реалізує функцію невиключаючого "або".

ППФ, побудовані за допомогою зв'язки *диз'юнкція*, називаються *диз'юнкція*.

Зв'язка імплікація " $\rightarrow$ " використовується для представлення твердження типу "якщо, то".

ППФ, побудована шляхом з'єднання формул за допомогою зв'язки $\rightarrow$, називається *імплікацією*.

Ліва сторона імплікації називається *антецедентом*, права - *консеквентом*. Імплікація має значення "істина", якщо антецедент і консеквент мають значення "істина", або антецедент має значення "хиба" незалежно від консеквента. В інших випадках імплікація має значення "хиба".

ППФ зі знаком заперечення ($\sim$) перед нею називається *запереченням*.

У мові предикатів атомарна формула може приймати тільки істинні значення, тільки помилкові значення, а також у залежності від значень змінних, які до неї входять, або "істина", або "хиба". Для того, щоб при численні предикатів можна було маніпулювати значеннями змінних, введено поняття "квантор".

Квантор – логічний оператор, що перетворює всякий предикат на предикат меншої арності, зв'язуючи деякі змінні початкового предиката.

Розрізняють квантор загальності і квантор існування:

– універсальний квантор або квантор загальності; у природній мові читається як «всі», «кожен», «всякий», «який би не був»;

$\exists$ – екзистенціальний квантор або квантор існування. У природній мові читається як «існує (бодай один)», «існують», «знайдеться (бодай один)», «знайдуться», «деякі».

Якщо предикат $F(x)$ не містить інших змінних, окрім x , вирази та є реченнями, що виражають істинні або хибні висловлювання. Перше називається універсальним висловлюванням (або висловлюванням загальності), а друге – екзистенціальним висловлюванням (або висловлюванням існування). Універсальне висловлювання означає, що властивість $F(x)$ мають всі індивіди (предмети) з обраної індивідної області; екзистенціальне висловлювання означає, що властивість $F(x)$ має бодай один індивід з розглядуваної індивідної області. Універсальне висловлювання говорить, що область істинності предиката $F(x)$ є універсальною (збігається з індивідною областю), а екзистенціальне висловлювання говорить, що область істинності предиката $F(x)$ є не порожньою.

В рамках одного предиката можна використовувати і квантори загальності, і квантори існування, але для різних змінних.

Наприклад, якщо – двомісний предикат « любить », то застосування квантора загальності по породить предикат – «всі люблять », а застосування на цей останній предикат квантора існування по породить висловлювання (нульмісний предикат) – «існує той, кого люблять усі».

Якщо деяка змінна в ППФ є проквантифікованою, то вона називається зв'язаною. В іншому випадку змінна називається вільною. Будь-який вираз, який отримують шляхом квантифікування правильної формули, є також ППФ.

Предикатами першого порядку називаються предикати, в яких не допускається квантифікація по предикатних або функціональних символах, квантифікувати можна тільки змінні.

Апарат логічного виведення

У мові предикатів процедури логічного виведення виконуються над знаннями, представленими у внутрішній формі по відношенню до тих описів, які виконав проектувальник, відображаючи специфіку предметної області, таким чином проектувальник працює з зовнішньою формою представлення знань, а процедури логічного виведення – з внутрішньою.

Перетворення зовнішньої форми у внутрішню виробляється в системах, які реалізують мовою предикатів, автоматично на основі таблиць істинності для обчислення окремих предикатів і логічних операцій, а також на підставі цілого ряду еквівалентності (закони де Моргана, дистрибутивні закони, асоціативні закони). У процесі логічного виведення мови предикатів використовують операції, які застосовуються до існуючих ППФ з метою побудови нових ППФ.

"*Modus ponens*" - використовується для створення з ППФ виду A ППФ виду Y .

Операція спеціалізації дозволяє довести, що якщо деякому класу об'єктів притаманна певна властивість, то будь-який об'єкт даного класу буде мати цю ж властивість.

Операція уніфікації використовується для доказу теорії. Кванторизовані формули приводять до відповідної форми шляхом знаходження підстановок певних підвиразів.

Операція резолюції використовується для породження нових припущень. В основі методу лежить резолюція спростування гіпотези і доказ того, що ця гіпотеза є хибною. У процесі реалізації методу використовується операція виключення висловлювання, якщо ці висловлювання в даних припущеннях заперечуються, а в інших – ні. У результаті доказу, якщо спростування неправдиво, формують порожню резольвенту.

Для застосування резолюції ППФ повинні бути переведені в клаузальну форму шляхом спрощення, а потім представлені у формі диз'юнкції. Процес перетворення зводиться до наступних основних етапів:

1 - виключення символів імплікації з формул і обмеження області дії символу заперечення;

2 - розділення змінних, тобто заміна однієї пов'язаної квантором змінної, котра зустрічається у виразі кілька разів, різними іменами;

3 - виключення кванторів існування шляхом їх заміни функціями, аргументами яких є змінні, пов'язані кванторами спільності, область дії яких включає область дії виключеного квантору існування;

4 - перетворення припущень у префіксну форму, тобто в ППФ не залишається кванторів існування. Кожен квантор загальності має свою змінну, тому всі квантори загальності можна перемістити в початок ППФ і вважати, що область дії кожного квантор включає всю ППФ;

5 - приведення матриці до кон'юнктивної нормальної форми, тобто кон'юнкції кінцевої множини диз'юнкцій;

6 - виключення кванторів загальності; це можливо, тому що всі змінні, що залишилися на цьому етапі, знаходяться під дією квантора спільності;

7 - виключення символів кон'юнкції; в результаті матриця залишається тільки у вигляді диз'юнкцій, над якими можливе проведення операцій резолюції.

Особливості реалізації мови предикатів першого порядку

Машинна реалізація мови предикатів першого порядку має ряд серйозних проблем, які пов'язані з універсальністю апарату логічного виведення. Перша проблема - монотонність міркувань (у процесі логічного виведення не можна відмовитися від проміжного висновку, якщо стають відомими додаткові факти, які свідчать про те, що отримані на основі цього висновку рішення не призводять до бажаного результату. Друга проблема – комбінаторний вибух (у процесі логічного виведення неможливо застосовувати оціночні критерії для вибору чергового правила). Безсистемне застосування правил з розрахунку на випадковий доказ призводить до того, що виникає багато зайвих ланцюжків ППФ, активних в певний момент часу. Це найчастіше приводить до переповнення робочої пам'яті.

У процесі досліджень з відшукування ефективних процедур машинної реалізації мови предикатів є два основних підходи.

1. Відкидається принцип універсальності мови предикатів і проводиться пошук конкретних процедур, ефективних для конкретної предметної області. У цьому випадку в БЗ вводяться великі знання предметної області. Найбільш типовий представник – LISP.

2. Підхід в рамках традиційної логіки, направлений на збереження універсальності, властивій мові предикатів шляхом розробки ефективних процедур логічного виведення універсальних за своїм характером, але які дозволяють нейтралізувати монотонність і комбінаторний вибух.

Найбільш ефективною розробкою логічного підходу є мова PROLOG. У ньому прийнята зворотна стратегія виведення. Повністю реалізовані всі засоби опису знань мовою предикатів, в т.ч. і кванторами. Для породження нових висловлювань використовується операція резолюції. В якості процедури пошуку рішення, що дозволяє усунути монотонність і комбінаторний вибух використовують пошук в ієрархічно впорядкованому просторі станів.

Контрольні запитання

1. Що таке інженерія знань?
2. Що називають полем знань?
3. Що таке семіотика? Яка структура структури семіотики?
4. Поясніть семіотичну модель знань.
5. Що таке функціональна структура предметної області?
6. Назвіть та поясніть етапи трансляції знань при формуванні поля знань.

7. Що називають пірамідою знань?
8. Які основні стратегії здобуття знань?
9. Опишіть процедуру отримання знань.
10. Опишіть процедуру придбання знань.
11. Опишіть процедуру формування знань.
12. Назвіть теоретичні аспекти отримання знань.
13. Охарактеризуйте психологічний аспект отримання знань.
14. Охарактеризуйте гносеологічний аспект отримання знань.
15. Охарактеризуйте лінгвістичний аспект отримання знань.
16. Назвіть і поясніть шари лінгвістичних проблем знань.
17. Поясніть особливості формування понятійної структури.
18. Які особливості гносеологічного аспекту отримання знань?
19. Як виконати побудову моделі предметної галузі?
20. Назвіть підходи до структурування знань.
21. Охарактеризуйте ієрархічний підхід до структурування знань.
22. Охарактеризуйте структурний (системний) підхід до структурування знань.
23. Охарактеризуйте об'єктно-орієнтований підхід до структурування знань.
24. Охарактеризуйте об'єктно-структурний підхід до структурування знань.
25. Що таке стратифікація знань?
26. Поясніть алгоритм об'єктно-структурного аналізу.
27. Поясніть призначення продукційної моделі знань.
28. Поясніть призначення моделі семантичних мереж для представлення знань.
29. Поясніть призначення фреймової моделі знань.
30. Поясніть призначення логічної моделі знання.
31. Поясніть формальний опис продукційної моделі представлення знань.
32. Поясніть формальний опис моделі семантичних мереж.
33. Поясніть формальний опис кадрової моделі представлення знань.
34. Поясніть формальний опис логічної моделі уявлення знань.
35. Назвіть переваги та недоліки моделей представлення знань.
36. Наведіть приклади знань, наведених у вигляді розглянутих класичних моделей.

5 РОБОТА З НЕЧІТКИМИ ЗНАННЯМИ

У прикладах, розглянутих у попередніх розділах, усі знання були визначеними. Твердженнями були або "істина", або "хиба". Однак у реальному житті у наших знаннях існує "нечіткість". Тим не менш, на підставі неточних даних часто можна робити цілком певні висновки. Для цього доводиться розглядати комбінацію елементів знань, а також значення їхньої визначеності та міру їхньої визначеності і в результаті виводити нові знання та давати оцінку їхньої визначеності.

Людина робить необхідні висновки в подібних умовах щодня: ставить медичні діагнози та призначає лікування, керуючись симптомами; з'ясовує причини поганої роботи двигуна з акустичного шуму; правильно розуміє уривки фраз природної мови; впізнає друзів за їхніми голосами та ін. Знанням, якими оперує людина у зазначених випадках, властива невизначеність.

5.1 Види невизначеності

Невизначеність може породжуватися неповнотою опису ситуації, ймовірнісним характером подій, що спостерігаються, неточністю подання даних, багатозначністю слів природної мови, використанням евристичних правил виведення та ін. Найбільш важливі види невизначеності можна представити у вигляді дерева (рис. 5.1).

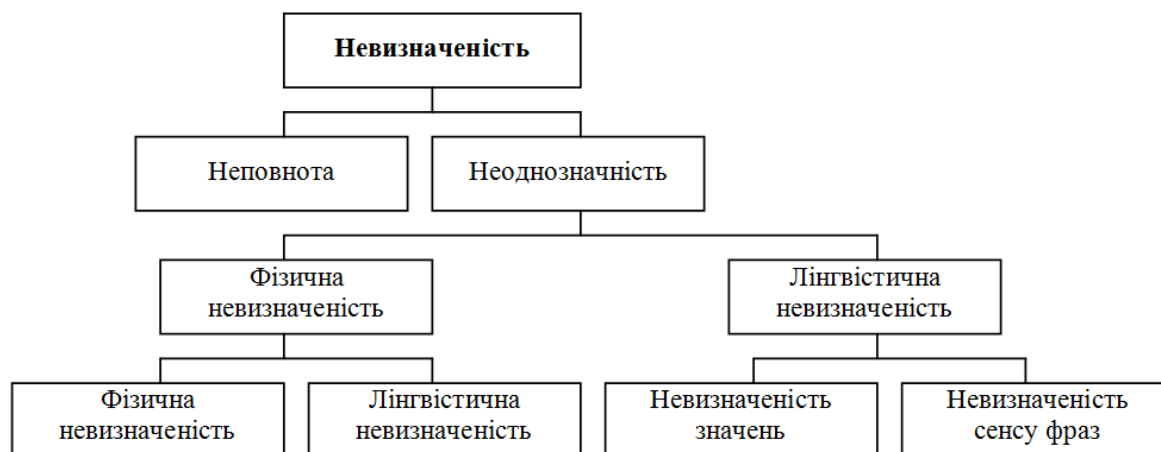


Рисунок 0.1 - Види невизначеності

На першому рівні дерева зображені терміни, які мають якісну оцінку характеру невизначеності. Невизначеність може бути пов'язана або з неповнотою знань, або з їх неоднозначністю.

Неповнота знань виникає, коли зібрана не вся інформація, необхідна побудови висновків.

Неоднозначність означає, що істинність тих чи інших висловлювань не може бути встановлена з абсолютною достовірністю. Вона породжується або фізичними причинами (фізична невизначеність), або лінгвістичними (лінгвістична невизначеність).

Фізична невизначеність може бути пов'язана з випадковістю подій, ситуацій, станів об'єкта чи неточністю представлення даних.

Лінгвістична невизначеність пов'язана з використанням природної мови для подання знань, що мають якісний характер, і виникає через множинність значень слів (полісемія) та сенсу фраз. Наприклад, "Двигун часто перегрівается" або "Петро вже великий". У цих прикладах неоднозначність обумовлена нечіткістю понять "часто" та "великий". Або, наприклад, "Він зустрів її на галявині з квітами". Як він її зустрів: на галявині, де росли квіти або зустрів з квітами на галявині? Такі невизначеності присутні у системах, на поведінку яких значною мірою впливають судження людини.

При аналізі невизначеності сенсу фраз виділяють синтаксичну, семантичну та прагматичну невизначеності.

Розглянута схема видів невизначеності є до певної міри умовною. У реальних системах зазначені види невизначеності можуть накладатися одна на одну. Наприклад, фізична невизначеність може ускладнюватись лінгвістичною невизначеністю.

Часто зазначені вище види неоднозначності не поділяють на окремі групи, а розглядають у межах терміна "ненадійні знання". Основним поняттям, використовуваним при побудові моделей виведення таких знань, є поняття достовірності. Достовірність висновків з урахуванням ненадійних знань можна визначити з допомогою різних підходів. Найчастіше використовуються: коефіцієнти впевненості, ймовірнісна байєсовська логіка, нечітка логіка, теорія Демпстера-Шефера.

Неповнота знань призводить до необхідності здійснення немонотонного виведення. Для формальної обробки знань, що характеризуються неповнотою, використовуються логіка замовчування Рейтера, немонотонна логіка Мак-Дермотта та Дойла, системи підтримки значень істинності.

5.2 Коефіцієнти упевненості

Коефіцієнти впевненості (КУ) застосовуються в галузі математики, яка називається нечіткою логікою. Оскільки евристичні правила ЯКЩО-ТО ґрунтуються виключно на людському досвіді, то неможна з певністю вважати, що вони мають рацію. Користувач ЕС також не може бути впевнений, що значення, які він надає змінним, є абсолютно правильними. Наприклад, правило

ЯКЩО	процентні_ставки=падають
І	податки=зменшуються
ТО	ціни=ростуть

є справедливим не завжди, тому йому можна приписати деяке значення коефіцієнта упевненості.

Зазвичай КУ може мати значення від -1 до 1 . Негативне значення КУ показує впевненість у тому, що правило не є істинним, а $KU > 0$ – що правило є істинним. Якщо $KU=1$, маємо повну впевненість у тому, що правило є істинним, якщо $KU=-1$, то маємо повну впевненість у хибності правила. Якщо $KU=0$, то немає сенсу розглядати правило.

Нехай наведене вище правило має $KU=0.9$, і неможна стверджувати, що відсоткові ставки падають, тобто, першій умові правила призначено $KU=0.6$. Крім того, припустимо, що податки коливаються (тобто збільшуються, то зменшуються), тому припустити зменшення податків можна з упевненістю $KU = 0.8$. Тоді правило можна записати так:

ЯКЩО процентні_ставки=падають ($KU=0.6$)
 І податки=зменшуються ($KU=0.8$)
 ТО ціни=ростуть ($KU \text{ правила}=0.9$)

Упевненість у тому, що рівень цін на зростатиме, може бути підрахована так:

- 1) якщо умови правила зв'язані зв'язкою *І*, то вибирається **мінімальний** КУ для умов;
- 2) якщо умови правила зв'язані зв'язкою *АБО*, то вибирається **максимальний** КУ для умов;
- 3) вибраний КУ множиться на КУ всього правила.

Для нашого прикладу:

$$(\min \{0.6, 0.8\}) * 0.9 = 0.6 * 0.9 = 0.54$$

Отже, можна сказати з упевненістю 0.54 , що рівень цін зростатиме.

Приклади.

1. Маємо правила:

ЯКЩО А ($KU=0.3$)
 І В ($KU=0.6$)
 ТО С ($KU=0.5$)

ЯКЩО D ($KU=0.4$)
 І Е ($KU=0.7$)
 ТО С ($KU=0.9$)

КУ для логічного висновку С підраховується так:

$$\max\{\{\min\{0.3, 0.6\} * 0.5\}, \{\min\{0.4, 0.7\} * 0.9\}\} = \\ = \max\{\{0.3 * 0.5\}, \{0.4 * 0.9\}\} = \max\{0.15, 0.36\} = 0.36$$

2. Маємо правило:

ЯКЩО A (КУ=0.3)
І В (КУ=0.6)
АБО D (КУ=0.5)
ТО C (КУ=0.4)

КУ для логічного висновку C підраховується так:

$$\max\{\min\{0.3, 0.6\}, 0.5\} * 0.4 = \max\{0.3, 0.5\} * 0.4 = 0.5 * 0.4 = 0.2$$

Метод використання КУ в ЕС MYCIN

В ЕС MYCIN коефіцієнт впевненості позначається CF і відображає різницю між мірою довіри та мірою недовіри до факту чи правила.

Для отриманих висновків застосовуються коефіцієнти впевненості, які можуть мати значення з діапазону $-1 \div 1$. При цьому застосовуються такі позначення:

$$CF[A, X] = \begin{cases} \frac{P(A | X) - P(A)}{1 - P(A)}, & \text{якщо } P(A | X) \geq P(A) \\ \frac{P(A | X) - P(A)}{P(A)}, & \text{якщо } P(A | X) < P(A) \end{cases},$$

$$(P(A) < > 1),$$

де CF[A,X] – коефіцієнт упевненості висновку A, якщо задовольняється умова X;

P(A) – апріорна ймовірність;

P(A | X) – апостеріорна ймовірність.

Умови правил можуть бути пов'язані зв'язками І та АБО. Крім того, у ЕС MYCIN враховується, що один і той же висновок можуть давати кілька правил. У цьому випадку маємо зв'язку КОМБ.

Приклад.

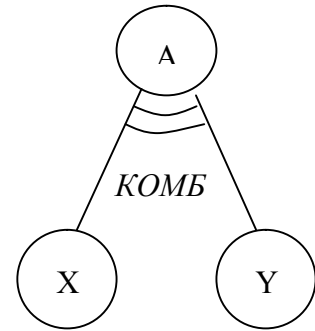
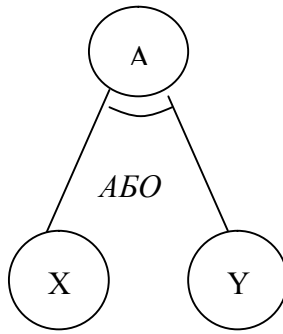
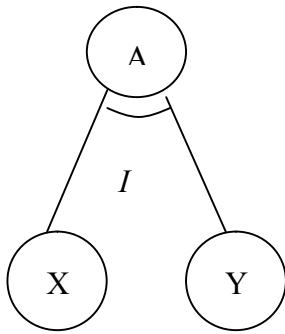
Розглянемо випадок визначення, чи хворий має застуду.

Свідокство 1 – кашель у хворого – надійне з упевненістю 0.6.

Свідокство 2 - температура 39-40° – надійне з упевненістю 0.5.

Простудний стан при спостереженні лише одного з цих симптомів можна підтвердити з упевненістю 0,6 або 0,5. Але обидва симптоми роблять застуду достовірнішою. В результаті з'єднання КОМБ ми маємо отримати більше значення КУ.

Опис правил:



ЯКЩО $X \text{ I } Y$ TO A $KY=C_1$
 $KY=C_2$

ЯКЩО X TO A $KY=C_{31}$



ЯКЩО X TO A $KY=C_{21}$
 ЯКЩО Y TO A $KY=C_{22}$

ЯКЩО $X \text{ АБО } Y$ TO A

ЯКЩО Y TO A $KY=C_{32}$

$C_1: CF[A, X \text{ I } Y]$ $C_2: CF[A, X \text{ АБО } Y]$ $C_{31}: CF[A, X]$ $C_{32}: CF[A, Y]$	$CF_{\text{правила}}$
--	-----------------------

$CF[X, \bullet]$ • визначається залежно від висновку та додається для подання
 $CF[X, \bullet]$ коефіцієнта впевненості при деякій умові.

Для зв'язки I :

$$CF_{\text{передумови}} = CF[X \text{ I } Y, \bullet] = \min\{CF[X, \bullet], CF[Y, \bullet]\}$$

Для зв'язки АБО :

$$CF_{\text{передумови}} = CF[X \text{ АБО } Y, \bullet] = \max\{CF[X, \bullet], CF[Y, \bullet]\}$$

Якщо $CF_{\text{передумови}} < 0$ то дії у висновку правила не виконуються.

Висновок є достовірним з коефіцієнтом $CF_{\text{правила}} * CF_{\text{передумови}}$, це значення переноситься на висновок A :

$$CF[A, \bullet] = CF_{\text{правила}} * CF_{\text{передумови}}$$

Для зв'язки КОМБ зазвичай окремо отримують $CF[A, X]$ і $CF[A, Y]$.
В ЕС MYCIN для зв'язки КОМБ коефіцієнт впевненості обчислюють за такими правилами:

$$CF[A, (X,Y)] = \begin{cases} 1, & \text{якщо } CF[A, X]=1 \text{ АБО } CF[A, Y]=1; & (5.1) \\ CF[A, X] + CF[A, Y] - CF[A, X] * CF[A, Y], & (5.2) \\ & \text{якщо } CF[A, X]>0 \text{ І } CF[A, Y]>0; \\ CF[A, X] + CF[A, Y], & (5.3) \\ & \text{якщо } CF[A, X] * CF[A, Y] \leq 0, \\ & CF[A, X] < \pm 1, CF[A, Y] < \pm 1; \\ CF[A, X] + CF[A, Y] + CF[A, X] * CF[A, Y], & (5.4) \\ & \text{якщо } CF[A, X] < 0 \text{ І } CF[A, Y] < 0; \\ -1, & \text{якщо } CF[A, X]=-1 \text{ АБО } CF[A, Y]=-1. & (5.5) \end{cases}$$

Коефіцієнт впевненості, отриманий із трьох і більше незалежних доказів, можна вивести, послідовно використовуючи зазначені вище формули. Але при отриманні $CF>0$ і $CF<0$ насамперед потрібно уточнити, який знак мають CF у формулах (5.2) та (5.4), а потім застосовувати формулу (5.3) для різнознакових CF . Якщо не дотримуватись цього правила, то залежно від порядку формул будуть отримані різні CF .

В MYCIN $CF>0$ визначає міру довіри, а $CF<0$ – міру недовіри. $CF=0$ визначає повну невизначеність.

В ЕС EMYCIN замість формули (5.2) застосовують формулу:

$$CF[A, (X,Y)] = \frac{CF[A, X] + CF[A, Y]}{1 - \min\{|CF[A, X]|, |CF[A, Y]|\}},$$

якщо $CF[A, X] \leq 0$ и $CF[A, Y] \leq 0$
 $CF[A, X] < \pm 1$ и $CF[A, Y] < \pm 1$.

У системі MYCIN виконується виведення від мети. Граничне значення коефіцієнта впевненості $CF_{\text{гран}}=0.2$. Якщо значення коефіцієнта впевненості менше граничного, правило не використовується.

Граничні коефіцієнти упевненості

У багатьох випадках граничні КУ задано заздалегідь. Логічне виведення вважається приєднаним лише в тому випадку, якщо його КУ перевищує заздалегідь задані граничні значення. Робота з БЗ триває доти, доки значення КУ логічного висновку не більше граничного значення. У процесі роботи виконуються певні обчислення.

Наприклад, нехай маємо логічний висновок з $KY=0.4$. Це значення запам'ятовується. Потім воно порівнюється з граничним значенням КУ

(припустимо, воно дорівнює 0.8). Запам'ятоване значення виявилось меншим за граничне, і, отже, робота з базою знань триває.

Якщо під час роботи з БЗ зустрівся той самий логічний висновок, КУ нового правила множиться на 1 мінус значення запам'ятованого раніше КУ і результат додається до запам'ятованого раніше КУ. Якщо КУ=1, то маємо абсолютну впевненість у правильності виведення. Потім знову запам'ятоване значення КУ порівнюється з граничним і якщо воно більше, виконується логічний виведення, інакше робота з БЗ триває.

Таким чином, використовуються формула

$$\text{Запам'ятований_КУ} = \text{Раніше_запам'ятований_КУ} + \\ + (1 - \text{Раніше_запам'ятований_КУ}) * \text{КУ_нового_правила}$$

Приклад.

Граничне значення КУ = 0.8

Правило:

ЯКЩО А ТО В (КУ=0,6)

Запам'ятований КУ=0.6

Нове правило: ЯКЩО С ТО В (КУ=0.7)

Запам'ятований КУ=0.6+(1-0.6)*0.7=0.88 (граничні значення перевищені, виконується виведення рішення).

5.3 Системи з дошкою оголошень

В останні роки у розробці архітектури експертних систем з'явився новий напрямок, який отримав назву системи з дошкою оголошень (*blackboard systems*). Системи з такою архітектурою можуть емулювати режим побудови як прямого ланцюжка логічного виведення, так і зворотного, а також поперемінно застосовувати ці режими в процесі роботи. Крім того, застосування систем з дошкою оголошень спонукає інженерів зі знань до ієрархічної організації знань щодо предметної області, та простору часткових та повних рішень. Таким чином, ця архітектура дуже добре підходить для вирішення завдань проектування, для яких характерно великий, але багатовимірний простір рішень, що факторизується. Системи з подібною архітектурою застосовують для інтерпретації даних, наприклад, розпізнавання графічних зображень та мовлення, аналізу та синтезу багатовимірних структур протеїнів та планування.

Недоліком систем з дошкою оголошень є те, що вони потребують значних обчислювальних ресурсів.

Принцип організації систем з дошкою оголошень

В основу організації систем цього типу покладено наступну ідею.

1. Група складається з експертів, які сидять біля класної дошки (або великої дошки оголошень) і намагаються вирішити певну проблему.
2. Кожен експерт є фахівцем у певній галузі, що має відношення до вирішення проблеми.
3. Формулювання проблеми та вихідні дані записані на дошці.
4. Експерти бачать те, що написано на дошці, і кожен із них думає над тим, чим він може допомогти у вирішенні проблеми.
5. Якщо хтось із експертів має щось сказати, він виконує відповідні обчислення та записує результат на тій же дошці.
6. Новий результат може допомогти іншим експертам у вирішенні проблеми.
7. Процес припиняється, коли проблему буде вирішено.

Така методика спільного вирішення проблем буде ефективна, якщо дотримуються певних правил, а саме:

- всі експерти повинні говорити тією самою мовою, хоча при записі результатів на дошці можуть використовуватись і різні схеми позначень;
- повинен існувати певний протокол визначення черговості "виступів", який набирає чинності в ситуації, коли одразу кілька експертів бажають розмістити свої результати на дошці.

Якщо розглядати працюючу за таким принципом систему з погляду організації процесу обчислень, то системі можна виділити такі структурні компоненти:

- знання предметної області розділені між *незалежними джерелами знань* (*KS – knowledge sources*), які працюють під управлінням *планувальника* (*scheduler*);
- рішення формується в деякій глобально доступній структурі - *дошці оголошень* (*blackboard*).

Таким чином, у цій системі всі знання "як вчинити" будуть представлені не у вигляді єдиного набору правил, а у вигляді набору програм. Кожен із компонентів цього набору може мати власний набір правил чи комбінацію правил і процедур.

Функції дошки оголошень багато в чому подібні до функцій робочої пам'яті в продукційних системах, але її організаційна структура значно складніша. Як правило, дошка оголошень поділяється на кілька рівнів опису, причому кожен рівень відповідає певному ступеню деталізації. Дані у межах окремих рівнів дошки оголошень представляють ієрархії об'єктів чи графи, тобто, структури більш складні, ніж вказівники, які використовувалися у робочій пам'яті продукційних систем. У найсучасніших системах може бути кілька дощок оголошень.

Джерела знань формують об'єкти на дошці оголошень під управлінням планувальника. Зазвичай записи активізації джерел знань розміщуються у спеціальному списку вибору, звідки їх витягує планувальник. Джерела знань "спілкуються" між собою лише через дошку оголошень та не можуть безпосередньо передавати дані один одному або запускати на виконання будь-які процедури. Тут є

певна аналогія з організацією роботи продукційних систем, у яких правила також можуть безпосередньо активізувати одне: все має проходити через робочу пам'ять.

Система HEARSAY

Архітектура на основі дошки оголошень виросла з розробленої наприкінці 70 років системи розпізнавання мови HEARSAY-II та HEARSAY-III. Програмування комп'ютера з метою розпізнавання мови – це одне з найскладніших завдань на той час, за які будь-коли бралися фахівці в галузі штучного інтелекту. Її рішення вимагає:

- складної обробки сигналів;
- зіставлення фізичних характеристик звукових сигналів із символічними елементами природної мови;
- виконання пошуку у великому просторі можливих інтерпретацій, в якому об'єднані ці різні за своєю природою елементи.

Для вирішення цієї проблеми було обрано методику, засновану на виділенні кількох рівнів абстракції опису аналізованих даних. Найнижчим є рівень фізичних акустичних сигналів, у якому формується звуковий спектр аналізованих сигналів. На наступних рівнях інформація проходить через напластування лінгвістичних абстракцій з дедалі більшим рівнем спільності понять - фонема, силаби (співзвучності), морфеми, слова, висловлювання та речення.

Починаючи з розробки системи, її творці розуміли, що з кожним рівнем аналізу пов'язана окрема галузь знань – аналіз звукових сигналів фонетика, лексичний аналіз, граматика, семантика, ораторське мистецтво. Жодна галузь окремо не здатна надати достатньо інформації для того, щоб вирішити проблему. Припустимо, наприклад, що, користуючись методами обробки акустичних сигналів, змогли розкласти вихідний звук на фонему. Але без додаткової інформації все одно не вдасться виділити сенс виразів, подібно до наступних: "I scream" (я вигукую) і "ice cream" (морозиво) або "please let us know" (будь ласка, дайте нам знати) і "please lettuce no" (будь ласка, без салату). Таким чином, хоча кожен окремих вид (набір) знань відіграє істотну роль у вирішенні проблеми і кожен з них може бути представлений у програмі більш менш незалежно від інших, автоматичне розпізнавання мови вимагає використання всіх цих знань спільно.

При розпізнаванні мови дослідникам доводиться стикатися ще з однією проблемою, яку також можна віднести до ключових, — проблемою невизначеності. Вона проявляється на всіх рівнях подання інформації:

- дані неповні та зашумлені;
- відсутня однозначна відповідність між даними на сусідніх рівнях: прикладом може бути відповідність між рівнями фонем і лексичних з при аналізі фраз "I scream" та "ice cream";
- важливу роль відіграють лінгвістичний та смисловий контексти; інтерпретація сусідніх елементів робить більш менш ймовірними різні варіанти інтерпретації поточного сегмента.

Більш традиційні підходи до розпізнавання мовлення ґрунтуються на використанні статистичних моделей з теорії передачі інформації для визначення кореляційного зв'язку між сегментами. Підхід, що базується на знаннях, вимагає значного перегляду методів обробки невизначеності.

Перелічимо вимоги, яким має задовольняти ефективно діюча система розпізнавання мови, заснована на знаннях.

1. З усіх можливих послідовностей операцій (часткових рішень) хоча б одна повинна призводити до коректної інтерпретації.

2. Процедура аналізу наявних варіантів інтерпретації має давати коректному варіанту більш високу оцінку, ніж іншим конкуруючим варіантам. Інакше кажучи, правильна інтерпретація з урахуванням вимови має бути оцінена вище, ніж інші варіанти інтерпретації, які враховують особливостей індивідуальної дикції.

3. Обчислювальні ресурси (пам'ять і час обчислень), необхідні пошуку правильної інтерпретації, не повинні перевищувати певний поріг. Система розпізнавання, яка через пару днів видасть результат, нехай і правильний, і вимагатиме пам'яті об'ємом кілька гігабайт, навряд чи комусь буде потрібна.

У наведеному списку перша і третя вимоги певною мірою суперечать одна одній. Для того щоб коректне рішення спочатку було присутнє в просторі гіпотез, на стадії формування гіпотез мимоволі доводиться бути марнотратним, що при великому словнику може призвести до комбінаторного вибуху елементів рішень. Вихід може бути знайдений лише при використанні надзвичайно ефективних евристик. Таким чином, найважливішою передумовою досягнення успіху у створенні такої системи є розробка відповідної процедури оцінки варіантів (друга з перерахованих вище вимог).

Використання джерел знань у системі HEARSAY-II

Для генерації, комбінування та розвитку гіпотез інтерпретації у системі HEARSAY-II використовується кілька джерел знань. Створені гіпотези різного рівня абстракції зберігаються на дошці оголошень.

Кожне джерело знань можна вважати в першому наближенні набором пар "умова-дія", хоча вони можуть бути реалізовані і у формі, відмінній від правил, що породжують (наприклад, умови і дії можуть бути насправді довільними процедурами). Потік управління у цій системі також відрізняється від потоку управління у продукційних системах. Замість того, щоб у кожному циклі інтерпретатор аналізував виконання умов, специфікованих у джерелах знань, джерела знань заздалегідь оголошують про активізовані в них умови, повідомляючи, який вид модифікації даних впливатиме на виконання цих умов. У результаті система керується перериваннями, а цей режим управління значно ефективніший, ніж режим циклічного перегляду стану, що є основним для продукційних експертних систем. Такий режим нагадує використання демонів у фреймових системах, де потік управління регулюється оновленням даних.

Джерела знань пов'язані з рівнями дошки оголошень наступним чином. Умови, специфіковані в джерелі знань, задовольнятимуться внаслідок оновлення даних на певному рівні дошки оголошень. Джерело знань також може записувати дані в певний рівень, причому не обов'язково в той самий, що впливає на виконання умов. Більшість джерел знань у системі HEARSAY-II організовано так, що вони розпізнають дані на певному рівні лінгвістичного аналізу, а операції, що виконуються ними, належать до наступного за порядком рівня. Наприклад, деяке джерело активізується даними на силабічному рівні і формує лексичну гіпотезу на рівні слів.

HEARSAY-II має сім рівень розуміння від акустичних параметрів звукових хвиль до розуміння сенсу питання.

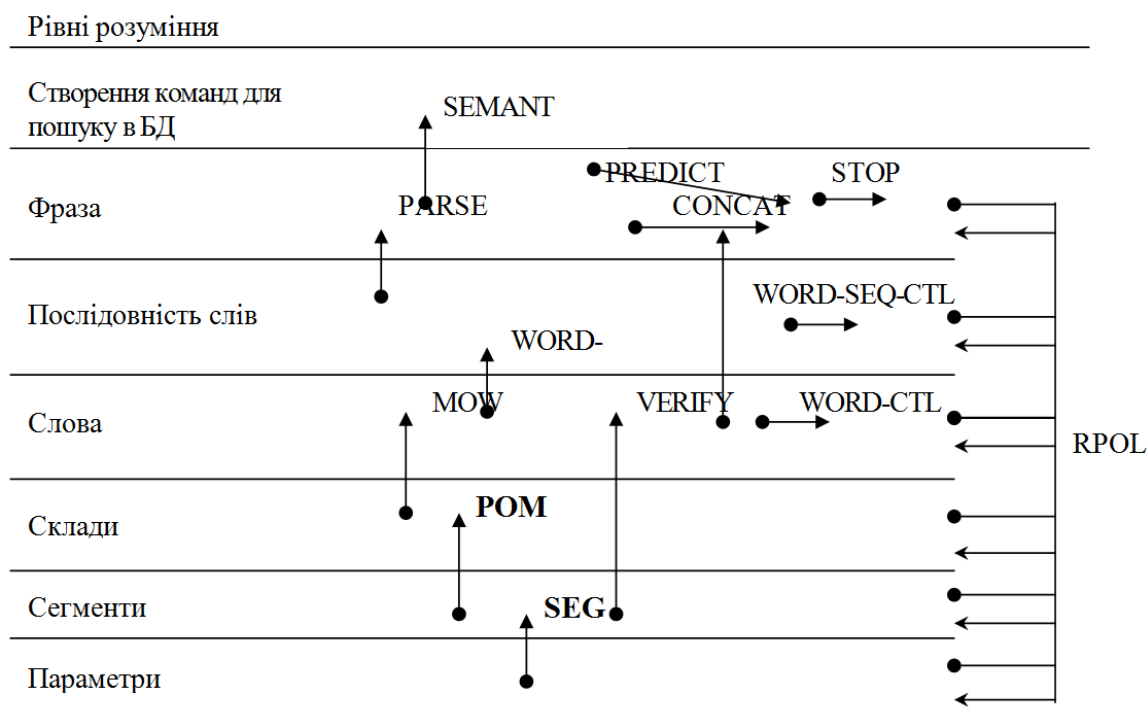


Рисунок 5.2 - Структура знань системи HEARSAY-II

Знання розподілені за джерелами знань, ● - місце зіставлення даних в умовах, ← місце занесення дій. Джерело знань - набір модулів типу "умова - дія":

SEG – перетворює мовні сигнали на дискретну форму, вимірює параметри, утворює сегменти;

POW – на основі сегментів створює гіпотези про склади;

MOW – на основі складів створює гіпотези про прості слова;

WORD-CTL – управляє числом гіпотез, створених MOW;

WORD-SEQ – на основі гіпотези про слова та граматичні правила створює гіпотези про послідовність слів;

WORD-SEQ-CTL – управляє числом гіпотез;

PARSE – виконує граматичний аналіз послідовності слів, якщо все правильно, створює гіпотези про фразу;

PREDICT – передбачає слова, які повинні передувати фразі чи йти за нею.

VERIFY – оцінює ступінь відповідності між гіпотезою про сегменти та парою пов'язаних слів;

CONCAT – на основі перевірених пар пов'язаних слів створює гіпотезу про фразу;

RPOL – оцінює ступінь довіри до інших гіпотез на основі інформації в гіпотезах, створених іншими джерелами;

STOP – оцінює необхідність зупинення процесу (чи закінчено речення) і вибирає гіпотезу, яка вважається найбільш вірною;

SEMANT – інтерпретує сенс для системи пошуку інформації.

Гіпотези оцінюються за шкалою від 0 до 100. Оцінка діє цьому рівні.

У дещо спрощеному вигляді архітектура системи HEARSAY-II представлена на рис.5.3.

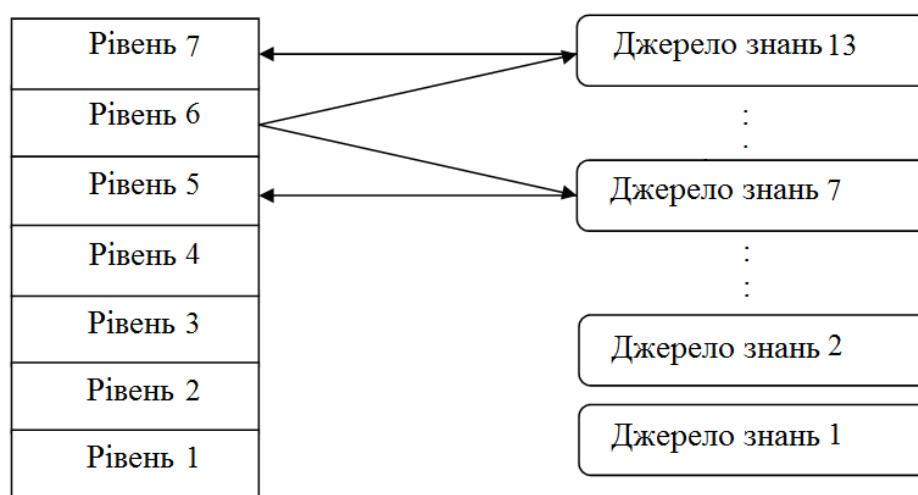


Рисунок 5.3 - Архітектура системи HEARSAY-II

Стрілки, спрямовані від рівнів дошки оголошень до джерел знань, вказують, як дані якого рівня змінюють виконання умов, специфікованих у джерелі знань. Стрілки у зворотному напрямку вказують, який рівень поміщає дані те чи інше джерело знань. Відгалуження від стрілки "Дія" джерела знань до монітора дошки оголошень означає, що зміна даних, виконана одним джерелом знань, фіксується в моніторі і потім використовується планувальником для активізації іншого джерела знань.

5.4 Ймовірнісна логіка у експертних системах

У ймовірнісній логіці висловлюванням надаються не тільки значення з безперервної шкали значень істинності від 0 до 1, і нуль відповідає неможливій події, одиниця – практично істинній. Значення істинності в ймовірнісній логіці

відповідає можливій істинності висловлювання, ступеню правдоподібності або ступеню підтвердження.

Висловлювання зі значенням істинності, яке знаходиться між істиною і хибною, називається в ймовірнісній логіці гіпотезою. Щодо кожного досліджуваного об'єкта можна висунути кілька гіпотез. Гіпотези, що висувуються, можуть відрізнятися одна від одної ступенем ймовірності, тобто ступенем наближення до достовірності. Достовірне знання не має ступенів: воно є або істинним, або хибним. Ймовірне знання характеризується ступенем наближення до достовірності: від повної неймовірності до повної достовірності.

Метою ймовірнісної логіки є поєднання здатності теорії ймовірностей обробляти невизначеність зі здатністю дедуктивної логіки розробляти структури.

У традиційній логіці дедуктивні висновки є цілком достовірними, якщо істинними є всі їхні вхідні посилки, і якщо в процесі міркування не порушені закони логіки. Близькими до достовірності можуть бути висновки неповної індукції. Достовірність узагальнення може бути спростована першим прикладом, який суперечить цьому узагальненню. Остаточна достовірність завжди досягається єдністю індукції і дедукції. Ймовірнісна логіка досліджує процес виведення загальних положень з одиничних даних спостереження і експерименту та використовує правила індуктивної логіки. У ймовірнісній логіці використовуються методи дослідження причинно-наслідкових зв'язків, тому її називають формою індуктивної логіки. Ступінь ймовірності гіпотези залежить від стану накопичених знань. У ймовірнісній логіці ймовірність розглядається як функція двох аргументів - самої гіпотези і наявного знання, причому співвідношення гіпотези до дійсності розглядається не безпосередньо, а через інші висловлювання, що виражають наявні знання.

Ймовірнісна логіка намагається знайти природне розширення традиційної логічної таблиці істинності: визначені в них результати отримуються за допомогою ймовірнісних виразів. Складність ймовірнісних логік полягає у тенденції поєднання обчислювальної складності їх ймовірнісних та логічних складових. Інші складності включають можливість отримання нелогічних результатів, таких, які зустрічаються у теорії Демпстера-Шефера. Необхідність мати справу з широким розмаїттям можливих ситуацій і проблем призвело до появи різноманітних напрямків ймовірнісної логіки.

Історично, спроби чисельного уявлення ймовірнісних міркувань бере початок з часів античності. Особливо сильний інтерес спалахнув, починаючи з 12-го століття, з роботою схоластів з винаходом половини доказу (так, два напівдокази – докази є достатніми, щоб довести винуватість), пояснення внутрішньої впевненості (достатньою впевненістю у скоєнні дії, але не абсолютною впевненістю), розвиток католицького пробабілізму (ідея, що завжди краще покладатися на встановлені доктрини чи міркування експертів, навіть якщо вони є менш ймовірними), появою міркування казуїстики, і вчення Лаксизму (в результаті чого імовірність застосовувалася для надання підтримки практично будь-якому судженню взагалі, причому можливо знайти експертну думку в підтримку майже будь-якого припущення).

Ймовірність може бути мірою суб'єктивної впевненості. Наприклад, в настанні тієї чи іншої події ("ймовірно завтра буде хмарно"), заснованої на знанні суб'єктом деяких прикмет (за кольором хмар при заході сонця, по вологості повітря і т.п.). У таких випадках дати якусь кількісну оцінку ступеня ймовірності дуже важко.

При застосуванні у штучному інтелекті ймовірність може виступати як міра об'єктивної впевненості (математична ймовірність) та як міра суб'єктивної впевненості.

Математична ймовірність є "об'єктивною характеристикою ступеня можливості появи певної події в якихось заздалегідь заданих умовах, які можуть повторюватися необмежену кількість разів". У математичній ймовірності діють статистичні закономірності, коли стан тієї чи іншої системи визначається не однозначно, а лише з певною ймовірністю.

Суб'єктивна ймовірність дозволяє аналізувати особливості суб'єктивної пізнавальної діяльності людей в умовах невизначеності. Наприклад, людина стверджує: "Досить імовірно, що в найближчі роки значно більшого поширення в промисловому виробництві отримають автоматичні маніпулятори (промислові роботи)". Тут ймовірність виступає як міра суб'єктивної впевненості. Остання визначається, по-перше, наявною (або відсутньою) у людини інформацією; по-друге, психологічними особливостями людини, які відіграють важливу роль при оцінці людиною ступеня ймовірності настання тієї чи іншої події. У висловлюваннях для характеристики явищ ми використовуємо різні слова: "дуже ймовірно", "малоймовірно", "неймовірно", "неправдоподібно" і ін.

Експерт робить обґрунтовані припущення у своїй предметній області. Це означає, що існують якісь статистичні дані, що дозволяють експерту робити певні припущення.

Британський математик Томас Байєс розробив ймовірнісну методику, засновану на твердженні, що певна подія відбудеться, тому що раніше вже відбулася якась інша подія. Такий підхід ґрунтується на обчисленні умовної ймовірності. У ЕС широко застосовуються статистичні рішення, засновані на теорії Байєса.

5.4.1 Основні поняття теорії ймовірностей

Теорія ймовірностей вивчає випадкові події. Часто людина, сама того не помічаючи, висловлює припущення або робить висновок, користуючись теорією ймовірностей. Наприклад, "Я на 80% упевнений, що все в порядку".

Для визначення ймовірності деякого події використовується формула:

$$P = \frac{\text{Число експериментів, результатом яких є якась подія}}{\text{Загальне число експериментів}}$$

Приклад. Нехай 2 рази кидають монету. Можливі наслідки:

№ експерименту	Перший кидок	Другий кидок
1.	Орел	Орел
2.	Орел	Решка
3.	Решка	Орел
4.	Решка	Решка

Ймовірність випадання Решки:

$$P = \frac{\text{Число експериментів, в яких випадає решка}}{\text{Загальне число експериментів}} = \frac{3}{4},$$

тобто решка випадає в другому, третьому і четвертому експерименті.

Приклад. Ймовірність того, що при першому кидку випаде решка, дорівнює 2/4 (3 і 4 експеримент). Ймовірність того, що решка взагалі не випаде, рівна 1/4 (перший експеримент).

Ймовірність появи будь-якої події ≥ 0 і ≤ 1 .

5.4.3 Ймовірність Байєса

Байєсіанізм (байєсіанство) - формальний підхід, заснований на розумінні ймовірності як міри впевненості. Такий підхід базується на теоремі Байєса. Відіграє важливу роль в теорії підтвердження гіпотез експериментальними даними. Суть підходу полягає у тому, що ступінь нашої раціональної впевненості в певній теорії змінюється в залежності від отримання нових емпіричних даних, що стосуються досліджуваного явища. Тому для байєсіанських теорій велике значення має поняття апостеріорної ймовірності.

В основі байєсіанського підходу лежать концепції *апріорної* (безумовної) і *апостеріорної* (умовної) ймовірностей.

Апріорна ймовірність - це явна міра впевненості суб'єкта в істинності; *апостеріорна ймовірність* - ступінь впевненості суб'єкта після отримання нових даних у результаті експериментів.

Теорема Байєса

Теорема Байєса (формула Байєса) – одна з основних теорем теорії ймовірностей, яка дозволяє визначити ймовірність якої-небудь події за умови, що сталася інша статистична взаємозалежна з нею подія.

Якщо подія A може відбутися тільки при виконанні однієї з подій

$$B_1, B_2, \dots, B_n,$$

які утворюють повну групу несумісні подій, то ймовірність події A обчислюється за формулою повної ймовірності:

$$P(A) = P(B_1)P(A|B_1) + P(B_2)P(A|B_2) + \dots + P(B_n)P(A|B_n).$$

Події $B_1, B_2, \dots, B_n$ називаються гіпотезами.

Якщо подія A сталася, то це може змінити ймовірності гіпотез

$$P(B_1), P(B_2), \dots, P(B_n)$$

За теоремою множення ймовірностей

$$P(AB_1) = P(B_1)P(A|B_1) = P(A)P(B_1|A),$$

звідси

$$P(B_1|A) = \frac{P(B_1)P(A|B_1)}{P(A)}.$$

Аналогічно, для решти гіпотез

$$P(B_i|A) = \frac{P(B_i)P(A|B_i)}{P(A)}, i = 1, \dots, n.$$

Отримана формула називається формулою Байєса.

Ймовірності гіпотез $P(B_i|A)$ називаються апостеріорними ймовірностями.

Ймовірності подій $P(B_i)$ – апіорні ймовірності.

Таким чином, умовна ймовірність враховує вже відомі результати експериментів.

Приклад.

Розглянемо той же експеримент дворазового кидання монети. Визначимо ймовірність випадання решка у другому кидку з урахуванням того, що в першому вже випав орел.

Решка випадає після орла тільки в другому експерименті, і ймовірність цієї події дорівнює $1/4$. Але ймовірність випадання решка у другому кидку з урахуванням того, що в першому вже випав орел, рівна $1/2$.

Ймовірність в цьому випадку вважається з урахуванням іншого набору експериментів:

№ експерименту	Перший кидок	Другий кидок
1.	Орел	Орел
2.	Орел	Решка

З розглянутих раніше експериментів видалені ті, в яких при першому кидку випадає решка, тому що відомо випадання орла при першому кидку. З двох, що залишилися експериментів тільки в другому після випадання орла

випадає решка. Отже, умовна ймовірність випадання орла у другому кидку з урахуванням випадіння орла у першому дорівнює $1/2$.

Умовна ймовірність - це ймовірність настання якоїсь події A за умови, що вже настала певна інша подія B .

Умовна ймовірність позначається $P(A|B)$.

Ймовірність настання двох подій обчислюється так:

$$P(B \text{ і } A) = P(A | B) * P(B),$$

це ймовірність того, що відбудуться дві події A і B , причому U станеться першим, дорівнює ймовірності настання події A , якщо відомо, що сталося подія B , помноженої на ймовірність появи події B .

Приклад.

З набору літер ААОО випадковим чином вибирається або літера A , або O . Застосовуючи рівняння умовної ймовірності, можна обчислити вірогідність того, що в двох спробах спочатку попадеться буква O , а потім A :

$$P(O \text{ і } A) = P(A | O) * P(O) \quad P(O) = 2/4,$$

тому що в наборі дві літери A і дві букви O .

Ймовірність того, що після букви O буде обрана A : $P(A|O) = 2/3$. Щоб зрозуміти, чому це так, розглянемо випадок, коли буква O вже вибрана і залишилися три літери: AAO .

Ймовірність вибору літери A складає $2/3$, оскільки з трьох літер дві A . Ймовірність $P(O \text{ і } A)$ дорівнює:

$$P(A | O) * P(O) = 2/3 * 2/4 = 1/3.$$

У США використовується ще одне рівняння умовної ймовірності:

$$P(A) = P(A|B) * P(B) + P(A|\text{not } B) * P(\text{not } B) \quad (5.6)$$

Це треба розуміти так: ймовірність появи події A , $P(A)$, дорівнює ймовірності появи події A за умови появи події B , $P(A|B)$, помноженій на ймовірність появи події B , $P(B)$, плюс ймовірність появи події A за умови, що подія B не відбулася, $P(A | \text{not } B)$, помножена на ймовірність, що подія B не відбулася, $P(\text{not } B)$.

Приклад.

Розглянемо використання умовної ймовірності на прикладі правил ринкової ціни:

П1. ЯКЩО ціни_на_ринку = падають
ТО рівень_життя = зростає

- П2. ЯКЩО $\text{ціни_на_ринку} = \text{зростають}$
ТО $\text{рівень_життя} = \text{падає}$
- П3. ЯКЩО $\text{курс_долара} = \text{падає}$
ТО $\text{ціни_на_ринку} = \text{падають}$
- П4. ЯКЩО $\text{курс_долара} = \text{зростає}$
ТО $\text{ціни_на_ринку} = \text{зростають}$

Визначимо імовірність підвищення рівня життя.

Система, що реалізує зворотне виведення, буде шукати висновок " $\text{ціни_на_ринку} = \text{зростають}$ ". Підійде П1: " $\text{рівень_життя} = \text{зростає}$ " за умови, що " $\text{ціни_на_ринку} = \text{падають}$ ".

Використовуючи рівняння (5.6), можна оцінити ці умови.

$$\begin{aligned} P(\text{level} = \text{зростає}) &= P(\text{level} = \text{зростає} | \text{rynok} = \text{падає}) * \\ &P(\text{rynok} = \text{падає}) + P(\text{level} = \text{зростає} | \text{rynok} = \text{не_падає}) * \\ &P(\text{rynok} = \text{не_падає}) \end{aligned} \quad (5.7)$$

Для того, щоб визначити, чи присвоєно змінній *rynok* значення *падає*, треба повернутися до правила П3. П3 перепишемо у вигляді рівняння 5.8:

$$\begin{aligned} P(\text{rynok} = \text{падає}) &= P(\text{rynok} = \text{падає} | \text{dollar} = \text{падає}) * \\ &P(\text{dollar} = \text{падає}) + \\ &P(\text{rynok} = \text{падає} | \text{dollar} = \text{не_падає}) * P(\text{dollar} = \text{не_падає}) \end{aligned} \quad (5.8)$$

Оскільки в жодному з правил в частині ТО немає змінної *dollar*, тобто значення ймовірності для неї визначити не можна, це значення має бути введено користувачем. З цієї ж причини умовну ймовірність також повинен задати користувач (вона не входить в частини ТО правил).

$$P(\text{dollar} = \text{падає}) = 0.6 \quad (\text{задав користувач})$$

Сума ймовірностей появи і не появи події дорівнює 1.

Отже

$$P(\text{dollar} = \text{не_падає}) = 1 - P(\text{dollar} = \text{падає}) = 1 - 0.6 = 0.4$$

Задамо значення всім умовним ймовірностям:

$$\begin{aligned} P(\text{rynok} = \text{падає} | \text{dollar} = \text{падає}) &= 0.8 & (\text{задав користувач}) \\ P(\text{rynok} = \text{падає} | \text{dollar} = \text{не_падає}) &= 0.1 \end{aligned}$$

Сума умовних ймовірностей для протилежних подій ≤ 1 .

Протилежними в умовних ймовірностях будуть події

dollar = *падає*

i *dollar* = *не_падає*.

Підставимо в рівняння числові значення:

$$P(\text{rynok} = \text{падає}) = 0.8 * 0.6 + 0.1 * 0.4 = 0.52$$

$$P(\text{rynok} = \text{не_падає}) = 1 - 0.52 = 0.48$$

Задамо значення умовних ймовірностей:

$$P(\text{level} = \text{зростає} | \text{rynok} = \text{падає}) = 0.85 \quad (\text{задав користувач})$$

$$P(\text{level} = \text{зростає} | \text{rynok} = \text{не_падає}) = 0.1$$

За рівнянням 5.6:

$$P(\text{level} = \text{зростає}) = 0.85 * 0.52 + 0.1 * 0.48 = 0.49 \text{ (або 49\%)}$$

5.4.4 Таблиці ймовірностей

Для оцінки підвищення рівня цін на ринку будемо використовувати два правила:

ЯКЩО RYNOK = *падає*

TO LEVEL = *зростає*

ЯКЩО DOLLAR = *падає*

TO RYNOK = *падає*

і відповідні їм рівняннями ймовірностей:

$$P(\text{LEVEL} = \text{зростає}) = P(\text{LEVEL} = \text{росте} | \text{RYNOK} = \text{падає}) * P(\text{RYNOK} = \text{падає}) + P(\text{LEVEL} = \text{росте} | \text{RYNOK} = \text{не_падає}) * P(\text{RYNOK} = \text{не_падає})$$

$$P(\text{RYNOK} = \text{падає}) = P(\text{RYNOK} = \text{падає} | \text{DOLLAR} = \text{падає}) * P(\text{DOLLAR} = \text{падає}) + P(\text{RYNOK} = \text{падає} | \text{DOLLAR} = \text{не_падає}) * P(\text{DOLLAR} = \text{не_падає})$$

$$P(\text{RYNOK} = \text{не_падає})$$

$$P(\text{RYNOK} = \text{падає}) = P(\text{RYNOK} = \text{падає} | \text{DOLLAR} = \text{падає}) * P(\text{DOLLAR} = \text{падає}) + P(\text{RYNOK} = \text{падає} | \text{DOLLAR} = \text{не_падає}) * P(\text{DOLLAR} = \text{не_падає})$$

$$P(\text{DOLLAR} = \text{падає})$$

$$P(\text{DOLLAR} = \text{не_падає})$$

У вирішенні цих рівнянь обов'язковою є участь експерта, який повинен побудувати для користувача таблицю ймовірностей. Ця таблиця буде містити уточнюючі слова, які використовуються для конкретизації необхідних користувачеві понять (наприклад, "середнє зростання"). В одному рядку з уточнюючими словами записується відсоток зміни лінгвістичної змінної, а також імовірність і умовна ймовірність, що ця зміна відбудеться. Нижче наводиться така таблиця з поясненнями.

Таблиця 5.2

Стовпчик 1 Лінгв. змінна	Стовпчик 2 Значення падіння курсу долара	Стовпчик 3
Падіння долара	$P(\text{RYNOK} = \text{падає} \text{DOLLAR} = \text{падає})$	$P(\text{DOLLAR} = \text{падає})$
Маленьке 1-2%	0.2	0.5
Середнє 3-4%	0.5	0.3
Велике >4%	0.8	0.1

Припустимо, що експерт вважає, що курс долара або падає відповідно з числами стовпця 1, або не падає взагалі. Ніяких інших варіантів немає. Отже, експерт повинен визначити ймовірність для двох випадків. У першому випадку, коли курс долара падає, ймовірність записується прямо в таблицю. У другому випадку за умови, що курс долара не падає, експерт записує умовну ймовірність падіння цін на ринку:

$$P(RYNOK = \text{падає} \mid DOLLAR = \text{не_падає}) = 0.7$$

Ймовірність зростання рівня життя при збереженні ринкових цін (ринкові ціни падають) експерт записує так:

$$P(LEVEL = \text{росте} \mid RYNOK = \text{падає}) = 0.2$$

Якщо ринкові ціни не падають, то ця ймовірність записується так:

$$P(LEVEL = \text{росте} \mid RYNOK = \text{не_падає}) = 0.1$$

Такі таблиці створюються і заповнюються експертом. Вони стають частиною БЗ і час від часу можуть змінюватися. Зміни до таблиці вносить або експерт, або інша ЕС.

Коли користувач ЕС намагається дізнатися ймовірність підвищення цін на ринку, система ставить запит:

"Визначити результат, якщо падіння курсу долара"

Система пропонує зробити вибір одного з варіантів падіння курсу долара:

- 1 – маленьке
- 2 – середнє
- 3 – велике

Нехай був обраний номер 2. З таблиці видно, що користувач очікує зниження курсу долара на 3-4%. Всі дані, необхідні для вирішення, можна знайти з таблиці:

Таблиця 5.3

Ймовірність	Джерело
$P(DOLLAR=\text{падає})=0.3$	Рядок 2, стовпчик 3 табл. 5.2
$P(DOLLAR=\text{не падає})=1-0.3=0.7$	Основне правило
$P(RYNOK=\text{падає} \mid DOLLAR=\text{падає})=0.5$	Рядок 2, стовпчик 2 табл. 5.2
$P(RYNOK=\text{падає} \mid DOLLAR=\text{не падає})=0.07$	Експерт
$P(RYNOK=\text{падає})=0.5*0.3+0.07*0.7= 0.199$	Рівняння 5.6

Тепер можна порахувати $P(LEVEL=\text{зростає})$. Для цього потрібно скористатися попереднім результатом:

Таблиця 5.4

Ймовірність	Джерело
$P(\text{RYNOK}=\text{падає})=0.199$	З таблиці 5.3
$P(\text{RYNOK}=\text{не падає})=1-0.199=0.811$	Основне правило
$P(\text{LEVEL}=\text{зростає} \mid \text{RYNOK}=\text{падає})=0.2$	Експерт
$P(\text{LEVEL}=\text{зростає} \mid \text{RYNOK}=\text{не падає})=0.1$	Експерт

З рівняння 3 можна обчислити ймовірність підвищення рівня життя:

$$P(\text{LEVEL}=\text{зростає}) = 0.2 * 0.199 + 0.1 * 0.811 = 0.121,$$

тобто ймовірність дорівнює 12,1%.

Таблицю заповнює експерт, а не користувач.

Таким чином, при використанні ймовірностей у БЗ слід пам'ятати наступне.

1. Ймовірність появи деякої події обчислюється за формулою:

число експериментів з бажаним результатом, поділене на число всіх експериментів.

Ймовірність Байєса - це умовна ймовірність, тобто ймовірність появи однієї події за умови, що інша подія вже відбулася. Системи, що реалізують зворотний ланцюжок міркувань, використовують основні співвідношення теорії ймовірностей при роботі з правилами БЗ.

2. При імовірнісному аналізі цікавих для користувача величин може застосовуватися нечітка логіка. При цьому аналізуються лінгвістичні змінні, слова типу "росте", "великий" і т.д. Лінгвістичні змінні використовуються в БЗ для виконання логічних висновків. Для їх аналізу застосовуються формули теорії ймовірностей. При цьому в оцінці значень змінних присутні такі поняття як маленький, середній, великий.

Слід пам'ятати, що значення ймовірностей, що беруть участь в обчисленнях, визначаються згідно з конкретної ситуації і працюють тільки в її рамках.

5.4.5 Подання і обробка нечітких знань: підхід на основі умовних ймовірностей

У реальних умовах знання, якими має в своєму розпорядженні людей, завжди в якоюсь мірою неповні, приближені, ненадійні. Тим не менш, людям на основі таких знань все-таки вдається робити досить обґрунтовані висновки і приймати розумні рішення. Отже, щоб інтелектуальні системи були справді корисні, вони повинні бути здатні враховувати неповну визначеність знань і успішно діяти в таких умовах.

Джерела невизначеності в знаннях

Невизначеність може мати різну природу. Найбільш поширений тип недостатньої визначеності знань обумовлений об'єктивними причинами:

- дією випадкових і неврахованих обставин,
- неточністю вимірювальних приладів,
- обмеженими здібностями органів почуттів людини,
- відсутністю можливості отримання необхідних свідчень.

У таких випадках в оцінках і міркуваннях люди використовують імовірності, допуски і шанси (наприклад, шанси виграти в лотереї).

Другий тип невизначеності обумовлений суб'єктивними причинами:

- нечіткістю змісту використовуваних людиною понять (наприклад, "багато"),
- неоднозначністю сенсу слів і висловів (наприклад, "замок" чи "студента відрахувати не можна залишити").

Неоднозначність сенсу слів і висловів часто вдається усунути, прийнявши до уваги контекст, в якому вони вживаються, але це теж виходить не завжди або не повністю.

Таким чином, неповна визначеність і нечіткість наявних знань - типова картина при аналізі та оцінці ситуацій, при побудові висновків і рекомендацій. У процесі досліджень зі штучного інтелекту для вирішення цієї проблеми вироблено декілька підходів.

Найпершим можна вважати використання евристик у вирішенні завдань, в яких достатньо віддалений прогноз розвитку подій неможливий (як, наприклад, в шаховій грі).

При створенні експертних систем був застосований імовірнісний підхід. Однак, що виникли проблеми змусили звернутися до розробки особливих підходів до обліку невизначеності в знаннях безпосередньо для експертних систем (коефіцієнти впевненості в системах MYCIN і EMYCIN).

Наступним кроком дослідження в цій області стала розробка нечіткої логіки, основи якої були закладені Лотфі Заде.

У рішенні проблеми, що розглядається стосовно до США, побудованих на основі правил (систем продукцій), виділяються такі основні питання:

- а) як кількісно виразити достовірність, Надійність посилок?
- б) як виразити ступінь підтримки укладення конкретної посилкою?
- в) як врахувати спільне вплив декількох посилок на укладення?
- г) як будувати ланцюжка умовиводів в умовах невизначеності?

Одним з варіантів рішення цієї проблеми є підхід до побудови логічного висновку на основі умовних ймовірностей (байєсівський підхід). Байєсівський підхід не є єдиним підходом до побудови висновків на основі використання ймовірностей, але він представляється зручним в умовах, коли рішення доводиться приймати на основі частини свідочств та уточнювати у міру надходження нових даних.

5.4.6 Байєсівський підхід

Байєсівський підхід виходить з того, що будь-якому припущенням може бути приписана якась ненульова апріорна (від лат. *A priori* - з попереднього) ймовірність того, що воно істинно, щоб потім шляхом залучення нових свідочств отримати апостеріорної (від лат. *A posteriori* - з наступного) ймовірність істинності цього припущення. Якщо висунута гіпотеза справді вірна, нові свідчення повинні сприяти збільшенню цієї ймовірності, в іншому ж випадку повинні її зменшувати.

Прийmemo для подальших міркувань наступні позначення:

$P(H)$ - апріорна ймовірність істинності гіпотези H (від англ. Hypothesis - гіпотеза);

$P(H | E)$ - апостеріорна ймовірність істинності гіпотези H за умови, що отримано свідочство E (від англ. Evidence - свідочство);

$P(E | H)$ - ймовірність отримання свідочства E за умови, що гіпотеза H вірна;

$P(\sim E | \text{не}_H)$ - ймовірність отримання свідочства E за умови, що гіпотеза H невірна.

За визначенням умовних ймовірностей маємо:

$$P(H : E) = \frac{P(H \text{ и } E)}{P(E)} \quad \text{і} \quad P(E : H) = \frac{P(E \text{ и } H)}{P(H)}.$$

Враховуючи, що $P(H \text{ и } E) = P(E \text{ и } H)$, отримуємо теорему Байєса:

$$P(H : E) = \frac{P(E : H)P(H)}{P(E)}.$$

Враховуючи, що

$$P(E) = P(E|H) * P(H) + P(E|\text{не}_H) * P(\text{не}_H)$$

і

$$P(\text{не}_H) = 1 - P(H),$$

отримуємо формулу, що дозволяє уточнювати ймовірність істинності гіпотези H , що перевіряється, з урахуванням отриманого свідочства E :

$$P(H | E) = \frac{P(E | H) P(H)}{P(E | H) P(H) + P(E | \text{не}_H) (1 - P(H))}.$$

Переваги байєсівського підходу.

Первісна (апріорна) оцінка ймовірності істинності гіпотези $P(H)$ може бути досить наближеною, але вона дозволяє шляхом врахування свідочства E

одержати більш точну оцінку $P(H|E)$, яку можна тепер використовувати як оновлене значення $P(H)$ для нового уточнення з залученням нового свідчення.

Інакше кажучи, процес уточнення ймовірності $P(H)$ можна повторювати рекурсивно з залученням все нових і нових свідчень, кожен раз звертаючись до однієї і тієї ж формули. У кінцевому рахунку, якщо свідочтв виявиться достатньо, можна отримати остаточний висновок про істинність (якщо виявиться, що $P(H)$ близьке до 1) або хибність (якщо виявиться, що $P(H)$ близьке до 0) гіпотези M .

Шанси та ймовірності пов'язані між собою наступною формулою:

$$O(H) = \frac{P(H)}{1 - P(H)}$$

У деяких випадках використання шансів більш поширене, ніж використання ймовірностей. Крім того, використання шансів замість ймовірностей може бути більш зручним з точки зору обчислень.

Переходячи до шансів у розглянутих формулах, отримаємо:

$$O(H : E) = \frac{P(E : H)}{P(E : neH)} O(H)$$

Якщо ж перейти до логарифмів величин, а в базі знань зберігати логарифми відношень $P(E|H)/P(E|ne_H)$, то всі обчислення зводяться просто до підсумовування, оскільки

$$\ln|O(H|E)| = \ln |P(E|H)/P(E|ne_H)| + \ln|O(H)|.$$

Проти використання шансів є кілька заперечень, головне з яких полягає в тому, що крайні значення шансів рівні "плюс" і "мінус" нескінченності, тоді як для ймовірностей це 0 і 1. Тому шанси зручно використовувати в тих випадках, коли жодна з гіпотез не може бути ні свідомо достовірною, ні свідомо неможливою.

Проблеми байєсівського підходу.

1. Суб'єктивний чинник. Очікування суб'єктивні і залежать від апріорної, індивідуальної інформації.

2. Проблематичність використання частотних розподілів, оскільки вони являють собою окремих випадок.

3. Можливість обчислення величини $P(E)$. Цю величину легко визначити, якщо є можливість обчислити $P(E|ne_H)$, а це не завжди можна зробити. Одна з можливостей обійти це ускладнення полягає в переході до повної групи подій, проте це не рятує становище, якщо склад повної групи подій невідомий. Можна користуватися й грубими оцінками, якщо зберігається точність діагнозу. Крім того, якщо $P(H)$ уточнюється в ході роботи, то $P(E)$ можна теж уточнювати.

4. Припущення про незалежність свідочств. З теоретичної точки зору це зауваження є дуже серйозним, але оскільки в кінці процесу аналізу нас цікавлять не стільки точні значення ймовірностей (це більше турбує статистиків), скільки співвідношення ймовірностей, то при однаковому порядку помилковості оцінок ймовірностей гіпотез для практики більш важливою виявляється правильність загальної картини, створюваної СШ.

5.4.7 Байєсові мережі довіри

Для виконання логічного висновку в рамках байєсівського підходу виконують побудову байєсової мережі.

Байєсова мережа довіри – це графова модель, що поєднує ймовірнісні та причинно-наслідкові відношення між змінними.

У байєсових мережах поєднуються:

- емпіричні частоти появи значень змінних;
- суб'єктивні оцінки очікувань;
- теоретичні уявлення про ймовірність наслідків на основі апріорної інформації.

Суть байєсової логіки:

- базується на попередньому досвіді і уточнюється на основі наявної оперативної інформації;
- вузлами байєсової мережі є випадкові змінні;
- ребрами графа є причинно-наслідкові відносини;
- кожна змінна може залежати тільки від безлічі своїх безпосередніх сусідів;
- щільність спільної ймовірності значень всіх змінних спрощується до твору локальних умовних густин.

Байєсові мережі довіри використовуються в тих областях, які характеризуються успадкованою невизначеністю.

Причини успадкованої невизначеності:

- неповне розуміння предметної області;
- неповні знання;
- знання характеризуються випадковістю.

Таким чином, байєсові мережі довіри застосовують для моделювання ситуацій, що містять невизначеність в деякому сенсі.

Для байєсових мереж довіри іноді використовується ще одна назва "причинно-наслідкові мережі", в яких випадкові події з'єднані причинно-наслідковими зв'язками.

Як в умовах визначеності, так і в умовах невизначеності генеруються нові знання.

Суть знання, яке генерується в умовах невизначеності, полягає в розумінні того, чи впливає отримана інформація на наші очікування щодо інших подій.

Для створення байесової мережі необхідно виконати наступні кроки.

1. Вибрати множину змінних, які є значущими для вирішення завдань в предметній області. Це слід виконувати, консультуючись з експертом в предметній області.

2. Упорядкувати змінні. Кореневі (вихідні) причини повинні бути обрані першими. Потім змінні, залежні від кореневих, потім від тих, які вже вибрали.

3. Побудувати граф залежних змінних (тобто визначити правила):

1) вибрати змінну X і додати вузол для неї;

2) встановити батьків X , кожен батьківський вузол повинен мати прямий вплив на X ;

3) повторити цю процедуру для всіх вузлів.

4. Побудований граф повинен бути ациклічним.

5. Кожна змінна є випадковою і може приймати одне з взаємовиключаючих значень.

6. Кожній змінній-нащадку A зі змінними-предками $B_1, \dots, B_n$ приписати таблицю умовних ймовірностей (ТУЙ) $P(A|B_1, \dots, B_n)$.

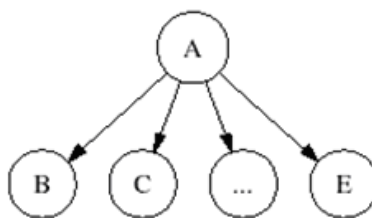
Типи з'єднань в байесовій мережі

1. Послідовне з'єднання:



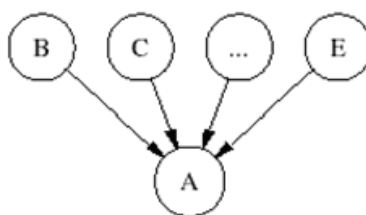
Вплив може поширюватися від A до C і назад, поки U не конкретизовано.

2. Дивергентне з'єднання:



Вплив може розповсюджуватися між нащадками вузла A , поки його значення не конкретизовано.

3. Конвергентне з'єднання:



Якщо про A нічого не відомо, крім того, що може бути виведено з інформації про його предків $B, C, \dots, E$, то ці змінні-предки є розділеними. При уточненні A можна оцінити взаємний вплив між його предками.

d-розділеність: Дві змінні в байєсовій мережі є *d*-розділеними, якщо на кожному шляху, що з'єднує ці вершини, знайдеться проміжна вершина X , така що:

з'єднання з X є послідовним або дивергентним, і значення X відомо,
АБО

з'єднання конвергентне, і немає свідочств ні про значення X , ні про кожного з її нащадків.

Редукція спільної ймовірності розподілу кількох випадкових змінних в байєсовій мережі (ланцюгове правило повної ймовірності).

У теорії ймовірностей ланцюгове правило (загальне правило добутку) дає можливість обчислювати будь-який член спільного розподілу набору випадкових змінних із застосуванням лише умовних імовірностей. Це правило є корисним у дослідженні байєсових мереж, що описують розподіл імовірності в термінах умовних імовірностей.

Розгляньмо пронумерований набір наборів. Щоби знайти значення одного члена спільного розподілу, ми можемо застосувати визначення умовної ймовірності для отримання

$$P(A_n, \dots, A_1) = P(A_n | A_{n-1}, \dots, A_1) \cdot P(A_{n-1}, \dots, A_1)$$

Повторення цього процесу з кожним кінцевим елементом створює добуток

$$P\left(\bigcap_{k=1}^n A_k\right) = \prod_{k=1}^n P\left(A_k \mid \bigcap_{j=1}^{k-1} A_j\right)$$

Для чотирьох змінних ланцюгове правило дає добуток умовних імовірностей:

$$P(A_4, A_3, A_2, A_1) = P(A_4 | A_3, A_2, A_1) \cdot P(A_3 | A_2, A_1) \cdot P(A_2 | A_1) \cdot P(A_1)$$

Це правило ілюструється таким прикладом:

Урна1 містить 1 чорну кулю та 2 білих кулі;

Урна2 містить 1 чорну кулю та 3 білих кулі.

Припустімо, що ми обираємо урну навмання, і потім беремо кулю з цієї урни.

Нехай подією A буде обрання Урни1: $P(A) = P(\text{не } A) = 1/2$.

Нехай подією B буде шанс взяти білу кулю.

Шанс взяти білу кулю за умови, що ми обрали першу урну, становить $P(B|A) = 2/3$. Подія A, B буде їхнім перетином: обрання першої урни та взяття білої кулі з неї. Цю ймовірність може бути знайдено за ланцюговим правилом для ймовірності:

$$P(A, B) = P(B|A) \cdot P(A) = (2/3) \cdot (1/2) = 1/3.$$

5.4.8 Застосування байєсових мереж

Медицина – системи діагностики захворювань (PathFinder Child MUNIN, Painulim, SWAN).

Космос і військові розробки – СППР Vista (центр управління польотами NASA), "Operation Dardanelles" (дослідження сценаріїв проведення військових операцій).

Комп'ютери, ПЗ – управління інтерфейсними агентами в MS Office, діагностика проблем принтерів, wizard-підсистеми, анти-спамові програми.

Обробка зображень і відео - відновлення 3D-сцен з динамічної 2D-інформації, синтез статичних зображень високої чіткості з відеосигналу.

Фінанси і економіка - оцінка ризику і прогноз дохідності портфелів фінансових інструментів.

5.5 Теорія свідоцтв Демпстера-Шефера

У теорії Демпстера-Шафера (ТДШ) для міри істинності твердження A використовується інтервальна оцінка $[a, 1 - \bar{a}]$, де нижня межа a називається довірою і дорівнює сумі фактів, що підтримують твердження, а верхня межа, яка називається правдоподібністю, дорівнює $1 - \text{сума фактів, що суперечать твердженню } A$, тобто, підтримують заперечення A . Якщо немає фактів, що підтримують твердження, то нижня межа (довіра) дорівнює нулю, а якщо немає і фактів, що суперечать твердженню, то верхня межа (правдоподібність) дорівнює одиниці. Така ситуація $[0, 1]$ відповідає повній невизначеності.

Можливість розрізнити невизначеність $[0, 1]$ та рівноймовірність $[0.5, 0.5]$ є важливою при проведенні міркувань. Наприклад, у методі Байєса, якщо апіорна ймовірність події A невідома, вважають, що $P(A) = 0.5$. Цей вибір ми змушені робити для "істинно рівноймовірної" події (наприклад, підкидання монети). Теорія Демпстера-Шефера дозволяє розрізнити ці дві ситуації.

Припустимо, що є дві конкуруючі гіпотези h_1 і h_2 . За відсутності інформації, що підтримує ці гіпотези, міра довіри та правдоподібності кожної з них належать відрізка $[0; 1]$. У міру накопичення свідоцтв ці інтервали зменшуватимуться, а довіра до гіпотез – збільшуватиметься.

У теорії Демпстера-Шефера (ТДШ) невизначеність знань представляється з допомогою деякої множини X . Елементи цієї множини відповідають можливим фактам чи висновкам. Невизначеність полягає в тому, що заздалегідь невідомо, яке з можливих значень прийме факт або висновок $x \in X$. Для характеристики ступеня визначеності в ТДШ вводиться певна одинична міра упевненості (її називають також одиничною масою впевненості), яка розподіляється між елементами X . При цьому, якщо вся маса (ступінь) впевненості припадає на один елемент $x \in X$, то ніякої невизначеності немає. Невизначеність виникає, коли маса впевненості розподіляється між декількома

елементами $x \in X$. Розподіл мас упевненості (рис. 5.4) між елементами множини X представлено у вигляді точок. Тут $X = \{x_1, x_2, x_3\}$.

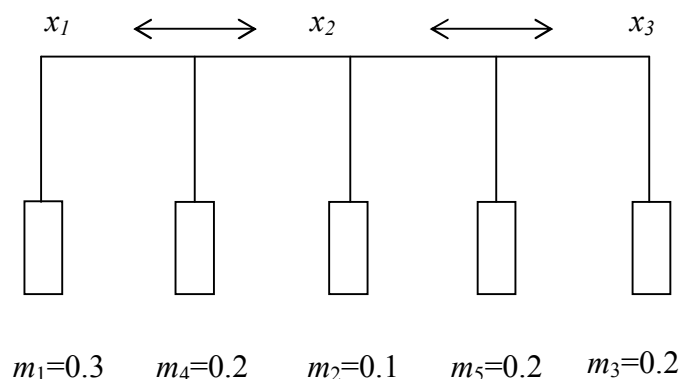


Рисунок 0.4 Розподіл мас упевненості

З кожним елементом множини X жорстко пов'язана відповідна маса впевненості. Так, x_1 відповідає $m_1=0,3$, x_2 відповідає $m_2=0,1$, x_3 відповідає $m_3=0,2$. Є також вільні маси впевненості $m_4=0,2$, $m_5=0,2$, які відносяться відразу до кількох елементів. Маса m_4 вільно переміщується між елементами x_1 і x_2 , а маса m_5 – між елементами x_2 і x_3 , тобто, m_4 закріплена за підмножиною $\{x_1, x_2\}$, а m_5 – за підмножиною $\{x_2, x_3\}$. Маси виражають ступінь впевненості у можливих значеннях фактів чи висновків. Так, ступінь впевненості у значенні x_1 може змінюватись від 0,3 до 0,5. Таким чином, ступінь незнання відповідає масі, розташування якої не визначено.

У загальному випадку розподіл мас впевненості задається функцією $m(A)$, що має такі властивості:

$$\begin{aligned} m(\emptyset) &= 0, \\ \sum m(A) &= 1. \end{aligned}$$

Тут A – множина, утворена з підмножин X , яким призначені відповідні маси (ступені) впевненості; $m(A)$ – функція, котра задає відображення A на інтервал $[0, 1]$. Для прикладу рис. 5.4 маємо:

$$A = \{\emptyset, \{x_1\}, \{x_2\}, \{x_3\}, \{x_1, x_2\}, \{x_2, x_3\}, \{x_1, x_3\}\},$$

а розподіл мас впевненості задається функцією $m(A)$, яка характеризується множиною значень

$$m(A) = \{0; 0,3; 0,1; 0,2; 0,2; 0,2; 0\}.$$

Зазначимо, що A складається з підмножин. Позначимо кожну таку підмножину через A_i . Ступінь довіри до висловлювань, що відповідають підмножині A_i , може бути обчислена за формулою

$$Bl(\{A_i\}) = \sum_{A_j \subseteq A_i} m(A)$$

Тут підсумовування виконується за всіма іншими підмножинами A_j , що входять до A_i . Наприклад:

$$Bl(\{x_1, x_2\}) = m(A_1) + m(A_2) + m(A_3) = m(\{x_1\}) + m(\{x_2\}) + m(\{x_1, x_2\}) = 0.3 + 0.1 + 0.2 = 0.6$$

Результати обчислень ступенів правдоподібності наведені нижче у табл. 5.5.

Таблиця 0.5 - Значення $Bl(A_i)$ і $Pl(A_i)$

A_i	$\emptyset$	$\{x_1\}$	$\{x_2\}$	$\{x_3\}$	$\{x_1, x_2\}$	$\{x_2, x_3\}$	$\{x_1, x_3\}$	X
$Bl(A_i)$	0	0.3	0.1	0.2	0.6	0.5	0.5	1
$Pl(A_i)$	0	0.5	0.5	0.4	0.8	0.7	0.9	1

Ступінь правдоподібності підмножини A_i визначається за формулою:

$$Pl(A_i) = 1 - Bl(not A_i) = 1 - \sum_{A_j \in A_i} m(A_j)$$

Величини $Bl(A_i)$ і $Pl(A_i)$ мають просту інтерпретацію: $Bl(A_i)$ представляє загальну масу впевненості, яка залишається, якщо з X видалити усі елементи, які не асоціюються з A_i , $Pl(A_i)$ представляє максимальну масу упевненості, котру можна отримати, якщо зсунути вільні маси до елементів множини A_i . При цьому $Bl(A_i) \leq Pl(A_i)$. Іншими словами, $Bl(A_i)$ представляє нижню межу довіри до A_i , а $Pl(A_i)$ - верхню.

Важливим елементом ТДШ є правило комбінації свідочств:

$$m(A_k) = \frac{\sum_{A_i \cap A_j = A_k} m_1(A_i) \cdot m_2(A_j)}{1 - \sum_{A_i \cap A_j = \emptyset} m_1(A_i) \cdot m_2(A_j)}$$

Сума в чисельнику правила розповсюджується на множину $A_k = A_{1i} \cap A_{2j}$. Правило є евристичним і дозволяє здійснювати розподіл ступенів довіри під час виведення.

5.6 Нечіткі множини і нечітка логіка

Прикладні завдання багатьох предметних областей неможливо вирішити, якщо для них неможливо отримати повну інформацію або якщо їх визначення недостатньо повно.

Такі завдання зустрічаються:

- в складних технічних системах;
- в системах економічного планування;
- в соціальних системах великої розмірності;

- в системах прийняття рішень і т.п.

Для вирішення таких завдань використовуються інтелектуальні системи, засновані на м'яких обчисленнях.

Під м'якими обчисленнями розуміють механізми, які включають в себе:

- нечітку логіку і імовірнісні обчислення;
- нейрокомп'ютинг з функціями навчання, адаптації, класифікації, системного моделювання та ідентифікації;
- генетичні обчислення з функціями синтезу, налаштування і оптимізації за допомогою систематизованого випадкового пошуку та еволюції.

Ці механізми можуть використовуватися як окремо, так і спільно, утворюючи гібридні системи, що створює ефект взаємного посилення.

Поряд з терміном "м'які обчислення" використовується термін "обчислювальний інтелект" – науковий напрямок, де вирішуються завдання штучного інтелекту на основі теорії нечітких систем, нейронних мереж і еволюційних (генетичних) обчислень.

Нечіткі нейронні мережі з генетичним налаштуванням параметрів (гібридні системи) демонструють взаємне посилення достоїнств і недоліків нівелювання окремих методів, наведених нижче.

1. Подання знань у нейронних мережах у вигляді матриць ваг не дозволяє пояснити результати проведеного розпізнавання або прогнозування, тоді як в системах виводу на базі нечітких правил результати сприймаються як відповіді на питання "чому?".

2. Нейронні мережі навчаються за допомогою універсального алгоритму, тобто трудомісткий витяг знань замінюється збором достатньої за обсягом навчальної вибірки. Для нечітких систем виведення витяг знань включає в себе складні процеси формалізації понять, визначення функцій приналежності, формування правил виведення.

3. Нечіткі нейронні мережі навчаються як нейронні мережі, але їх результати пояснюються як в системах нечіткого виведення.

Термін fuzzy (англ. нечіткий, розмитий - вимовляється "фазі") став ключовим словом на ІТ-ринку. Нечітка логіка з'явилася в 1965 в роботах Лотфі А. Заде (Lotfi A. Zadeh), професора технічних наук Каліфорнійського університету в Берклі.

Нечітка логіка є багатозначною логікою, що дозволяє визначити проміжні значення для таких загальноприйнятих оцінок, як так|ні, істинно|помилково, чорне|біле і т.п. Вирази подібні таким, як трохи тепло або досить холодно можна формувати математично і обробляти на комп'ютерах.

5.6.1 Нечіткі множини

Поняття нечітких множин (fuzzy sets) як узагальнення звичайних (чітких) множин було введено Л. Заде в 1965 р.

Нечіткою підмножиною A множини X називається пара

$$(X, \mu_A),$$

де $\mu_A(x) \in [0, 1]$ – функція, кожне значення якої інтерпретується як ступінь приналежності елемента x множини X підмножині A .

Функція $\mu_A(x)$ називається **функцією приналежності**

$$\mu_A(x) = \begin{cases} 1, & \text{якщо елемент } x \text{ належить множині } A, \\ 0, & \text{якщо елемент } x \text{ не належить множині } A \end{cases}$$

Конкретне значення функції приналежності називається ступенем або коефіцієнтом приналежності.

5.6.2 Нечіткі і та лінгвістичні змінні

Нечітка змінна описується набором

$$(N, X, A),$$

де N - назва змінної,

X - універсальна множина (область міркувань),

A - нечітка підмножина множини X .

Лінгвістична змінна має значення у вигляді нечітких змінних, тобто лінгвістична мінлива знаходиться на більш високому рівні, ніж нечітка змінна.

До складу лінгвістичної змінної входять:

- назва;
- множина значень, яка називається базовою терм-множиною T ; елементи базової терм-множини являють собою назви нечітких змінних;
- універсальна множина X ;
- синтаксичне правило G , за яким генеруються нові терми із застосуванням слів природної або формальної мови;
- семантичне правило P , яке кожному значенню лінгвістичної змінної ставить у відповідність нечітку підмножину A множини X .

Приклад.

Розглянемо таке нечітке поняття як «Вік» людини.

Це і є назва лінгвістичної змінної.

Сформуємо для лінгвістичної змінної базове терм-множина, що буде складатися з трьох нечітких змінних:

«Молодий», «Середній», «Старий».

Задамо область міркувань у вигляді $X = [0; 100]$ (років).

Побудуємо функції приналежності для кожного лінгвістичного терма з базового терм-множини T .

Сформуємо множину віку з кроком 10 років:

Вік = Молодий: $\mu_{\text{молодий}}(X) = \{1/0; 1/10; 0.8/20; 0.3/30; 0/40; \dots; 0/100\}$

Вік = Середній: $\mu_{\text{середній}}(X) = \{0/0; \dots; 0/20; 0.5/30; 1/40; 0.5/50; 0/60; \dots; 0/100\}$

Вік = Старий: $\mu_{\text{старий}}(X) = \{0/0; \dots; 0/30; 0.4/50; 0.8/60; 1/70; 1/80; 1/90\}$

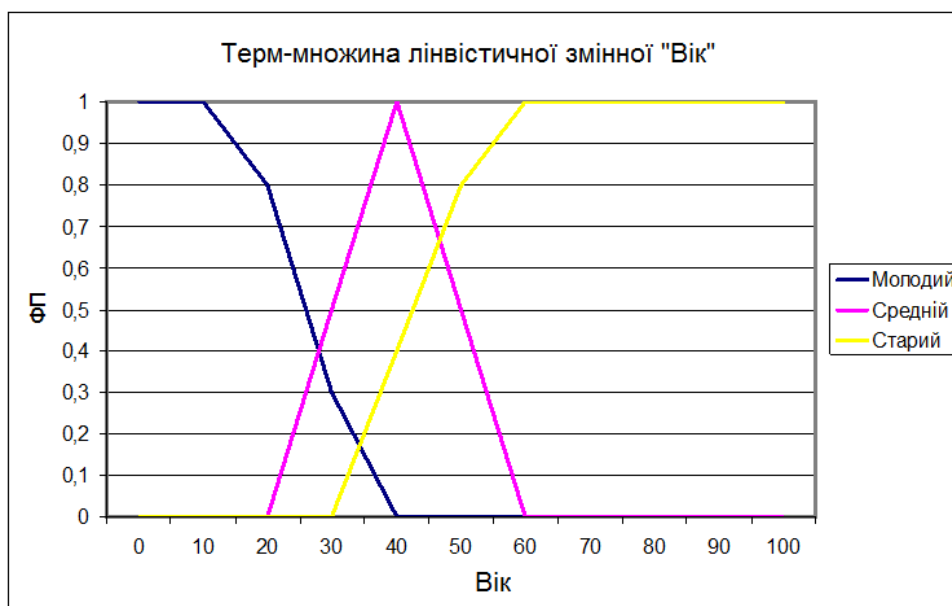


Рисунок 5.5 - Нечіткі множини і функції приналежності ЛЗ Вік

Таким чином, були побудовані функції приналежності для кожного лінгвістичного терма з базової терм-множини T .

Приклад.

Нехай x позначає температуру повітря навесні.

Назва лінгвістичної змінної «Температура».

Задамо область міркувань у вигляді:

$X = [-5; 30]$ (градусів за Цельсієм).

Сформуємо для лінгвістичної змінної базову терм-множину, що буде складатися з нечітких змінних «Низька», «Середня», «Висока», «Дуже_висока», записавши значення температури через 5 градусів:

Таблиця 5.6

Температура	ФП							
Низька	1	0,7	0,2	0	0	0	0	0
Середня	0	0,1	0,8	1	0,4	0	0	0
Висока	0	0	0,1	0,6	1	0,2	0	0
Дуже_висока	0	0	0	0	0,2	0,6	1	1
Множина X	-5	0	5	10	15	20	25	30

За даними табл. 5.6 побудуємо функції приналежності (рис. 5.6).

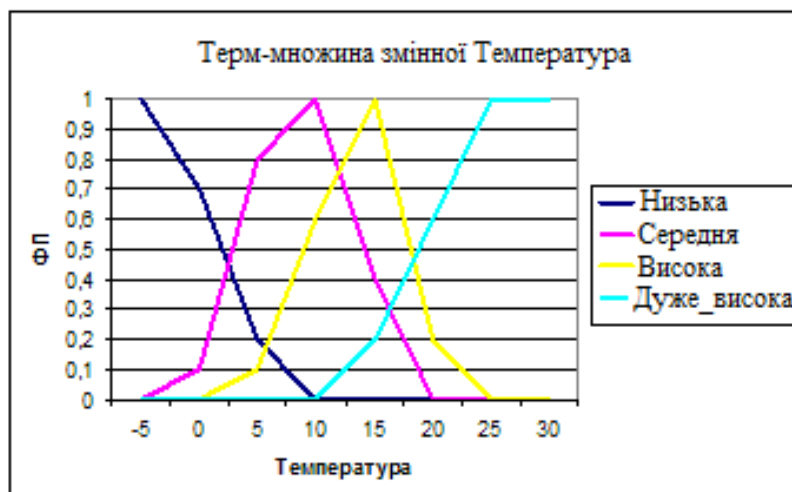


Рисунок 5.6 - Нечіткі множини і функції приналежності ЛЗ Температура

Існує понад десяток типових форм кривих для задання функцій приналежності. Найбільше розповсюдження отримали: трикутна, трапецеїдальних і гаусової функції приналежності.

Трикутна функція належності визначається трійкою чисел (a, b, c) , та її значення в точці x обчислюється згідно виразу:

$$\mu_A(x) = \begin{cases} 1 - \frac{b-x}{b-a}, & a \leq x \leq b \\ 1 - \frac{x-c}{c-b}, & b \leq x \leq c \\ 0, & \text{у решті випадків} \end{cases}$$

При $(b-a)=(c-b)$ маємо випадок симетричної трикутної функції приналежності, яка може бути однозначно задана двома параметрами з трійки (a, b, c) .

Аналогічно для завдання трапецеїдальної функції приналежності необхідна четвірка чисел (a, b, c, d) :

$$\mu_A(x) = \begin{cases} 1 - \frac{b-x}{b-a}, & a \leq x \leq b \\ 1, & b \leq x \leq c \\ 1 - \frac{x-c}{d-c}, & c \leq x \leq d \\ 0, & \text{у решті випадків} \end{cases}$$

При $(b-a)=(d-c)$ трапецеїдальних функція належності приймає симетричний вид.

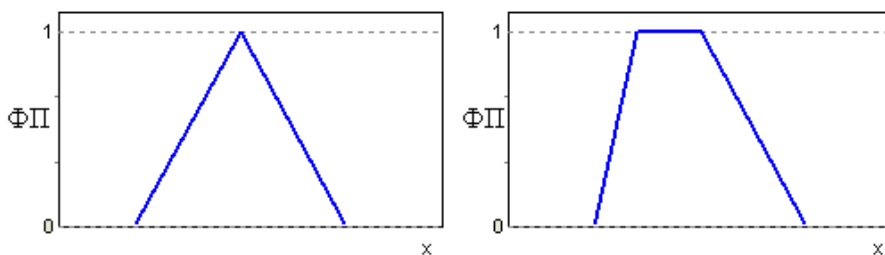


Рисунок 5.7 - Типові кусково-лінійні функції приналежності

Функція приналежності гаусового типу описується формулою

$$\mu_A(x) = \exp \left[- \left(\frac{x - c}{\sigma} \right)^2 \right]$$

і оперує двома параметрами. Параметр c позначає центр нечіткої множини, а параметр σ відповідає за крутизну функції

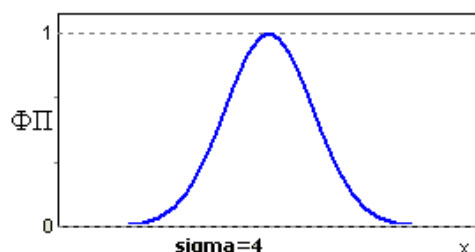


Рисунок 5.8 - Гаусова функція приналежності

Сукупність функцій приналежності для кожного терма з базового терм-множини T зазвичай зображуються разом на одному графіку.

Кількість термів в лінгвістичній змінній рідко перевищує 7.

5.6.3 Операції над нечіткими множинами

Визначимо поняття доповнення, об'єднання і перетину для нечітких множин.

Перетин ($C=A \cap B$); $\mu_C = \mu_{A \cap B}(X) = \mu_A(X) \cap \mu_B(X)$.

$\cap$ – знак операції взяття мінімуму.

Лінгвістичний сенс: операція кон'юнкції I .

Функція приналежності: $\mu_C = \min\{\mu_A; \mu_B\}$.

Графік функції приналежності:

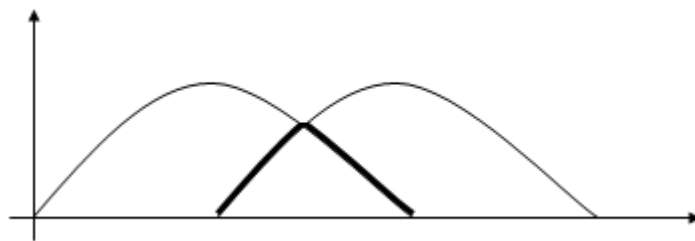


Рисунок 5.9 - Перетин множин

Об'єднання множин ($C=A \cup B$); $\mu_C = \mu_{A \cup B}(X) = \mu_A(X) \cup \mu_B(X)$

$\cup$ – знак операції взяття мінімуму

Лінгвістичний сенс: операція диз'юнкції *АБО*.

Функція приналежності: $\mu_C = \max\{\mu_A, \mu_B\}$.

Графік функції приналежності:

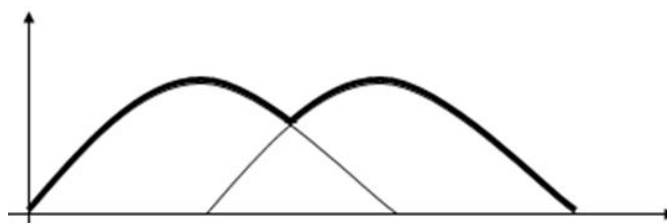


Рисунок 5.10 - Об'єднання множин

Доповнення ($C=A'$); $\mu_{A'}(X) = 1 - \mu_A(X)$

Лінгвістичний сенс: операція заперечення *НІ*.

Функція приналежності: $\mu_C = 1 - \mu_A$.

Графік функції приналежності:

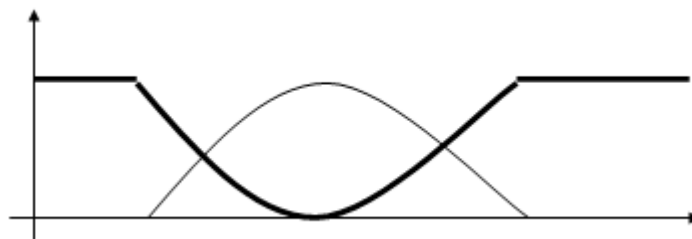


Рисунок 5.11 - Доповнення множин

Концентрація

Лінгвістичний сенс: *ДУЖЕ*.

Функція приналежності: $\mu_C = (\mu_A)^2$.

Графік функції приналежності:

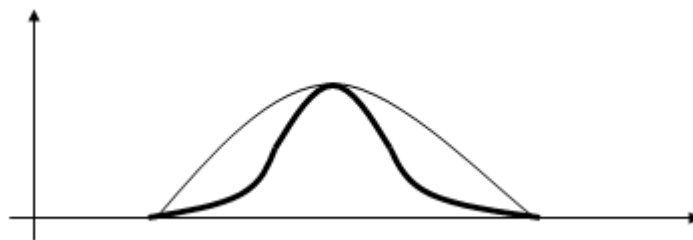


Рисунок 5.12 - Концентрація множини

Розмивання

Лінгвістичний сенс: СЛАБКІШЕ.

Функція приналежності: $\mu_C = (\mu_A)^{1/2}$.

Графік функції приналежності:

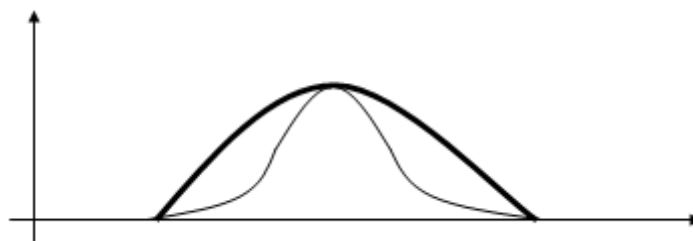


Рисунок 5.13 - Розмивання множини

Найбільш часто використовуються операції об'єднання, перетину і доповнення.

Приклади застосування операцій над нечіткими множинами

Маємо нечіткі множини "молодий" і "середній":

$$\mu_{\text{молодий}}(X) = 1/0 + 1/10 + 0.8/20 + 0.3/30 + 0/40 + 0/50 + 0/60 + 0/70 + 0/80 + 0/90 + 0/100$$

$$\mu_{\text{середній}}(X) = 0/0 + 0/10 + 0/20 + 0.5/30 + 1/40 + 0.5/50 + 0/60 + 0/70 + 0/80 + 0/90 + 0/100$$

Визначимо множину «не_молодий», виконавши операцію доповнення:

$$\text{не_молодий} = \mu_{\text{не_молодий}}(X) = \frac{0.2}{20} + \frac{0.7}{30} + \frac{1}{40} + \frac{1}{50} + \frac{1}{60} + \frac{1}{70} + \frac{1}{80} + \frac{1}{90}$$

Визначимо множину «Молодий_або_середній», виконавши операцію об'єднання:

$$(\text{молодий} \cup \text{середній}) = \mu_{\text{молодий} \cup \text{середній}}(X) = \frac{1}{0} + \frac{1}{10} + \frac{0.8}{20} + \frac{0.5}{30} + \frac{1}{40} + \frac{0.5}{50}$$

Визначимо множину «Молодой_і_середній» виконавши операцію перетину:

$$(\text{молодий} \cap \text{середній}) = \mu_{\text{молодий} \cap \text{середній}}(X) = \frac{0.3}{30}$$

Приклади реального застосування нечіткої логіки

- Автоматичне управління воротами греблі на гідроелектростанціях (Tokio Electric Power);
- Спрощене керування роботами (Hirota, Fuji Electric, Toshiba, Omron);
- Наведення телекамер при трансляції спортивних подій (Omron);
- Заміна експертів при аналізі роботи біржі (Yamaichi, Hitachi);
- Запобігання небажаних температурних флуктуацій в системах кондиціонування повітря (Mitsubishi, Sharp);
- Ефективне і стабільне управління автомобільними двигунами (Nissan);
- Управління економічною швидкістю автомобілів (Nissan, Subaru);
- Покращення ефективності та оптимізація промислових систем управління (Aptronix, Omron, Meiden, Sha, Micom, Mitsubishi, Nisshin-Denki, Oku-Electronics);
- Позиціонування приводів у виробництві напівпровідників wafer-steppers (Canon);
- Оптимізоване планування автобусних розкладів (Toshiba, Nippon-System, Keihan-Express);
- Системи архівації документів (Mitsubishi Eleceric);
- Системи прогнозування землетрусів (Inst. of Seismology Bureau of Metrology, Japan);
- Медицина: Діагностика раку (Kawasaki Medical School);
- Поєднання методів нечіткої логіки і нейронних мереж (Matsushita);
- Розпізнавання рукописних символів у кишенькових комп'ютерах (записних книжках) (Sony);
- Розпізнавання руху зображення у відеокамерах (Canon, Minolta);
- Автоматичне управління двигуном пилососів з автоматичним визначенням типу поверхні і ступеня засміченості (Matsushita);
- Управління освітленістю в камкодерах (Sanyo);
- Компенсація вібрацій в камкодерах (Matsushita);
- Однокноповий управління пральними машинами (Matsushita, Hitachi);
- Розпізнавання рукописних текстів, об'єктів, голосів (CSK, Hitachi, Hosai Univ., Ricoh);
- Допоміжні засоби польоту вертольотів (Sugeno);
- Моделювання судових процесів (Meihi Gakuin University, Nagoy University);
- САПР виробничих процесів (Aptronix, Harima, Ishikawajima-OC Engeneering);
- Управління швидкістю ліній і температурою при виробництві сталі (Kawasaki Steel, New-Nippon Steel, NKK);

- Управління метрополітенами для підвищення зручності водіння, точності зупинки та економії енергії (Hitachi);
- Оптимізація споживання бензину в автомобілях (NOK, Nippon Denki Tools);
- Підвищення чутливості та ефективності управління ліфтами (Fujites, Hitachi, Toshiba);
- Підвищення безпеки ядерних реакторів (Hitachi, Bernard, Nuclear Fuel).
-

Приклад: Система кондиціонування повітря Mitsubishi

Промислова система кондиціонування повітря, що забезпечує гнучку реакцію на зміни навколишніх умов

Реалізація:

- 50 правил
- 6 лінгвістичних змінних
- Дозвіл: 8 біт
- Вхідні значення: температура в кімнаті, температура стіни і миттєві значення цих сигналів

Розробка:

- 4 дні на створення прототипу
- 20 днів на тестування та інтеграцію
- 80 днів на оптимізацію на реальних тестових об'єктах
- Реалізація у вигляді чисто програмного комплексу на стандартному мікроконтролері.

Результати:

- Зменшення часу початку роботи до 40 відсотків до стандартного рішенням
- Підтримка температури при наявності збурюючих факторів (відкриті вікна і тому подібне) істотно поліпшена
- Невелике необхідне число датчиків
- Економія енергії - 24 відсотки

Приклад: Камкодер зі стабілізатором зображення, Matsushita

Послідовність дій системи:

- 1) збереження моментального кадру;
- 2) поділ картинки на 4 частини по 30 точок у кожній. Збереження сигналів від цих точок;
- 3) отримання наступного кадру і порівняння;
- 4) порівняння сигналів від нового кадру з сигналами збереженого; стабілізація або оновлення відповідно;
- 5) якщо є невеликі одно орієнтовані відхилення (відповідає вібрації), тоді передати збережений кадр;

б) якщо є великі (= передбачувані) або не дорівнює орієнтовані відхилення (відповідає руху), тоді передати і зберегти новий кадр.

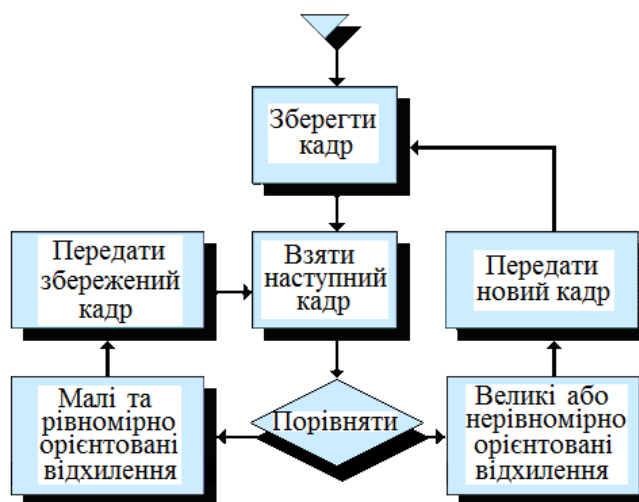


Рисунок 5.14 –Схема алгоритму роботи системи

5.6.4 Нечітке логічне виведення

Основою для проведення операції нечіткого логічного виведення є база правил, яка містить нечіткі висловлювання у формі "Якщо-То" і функції приналежності для відповідних лінгвістичних термів. При цьому повинні дотримуватися наступні умови:

1) існує хоча б одне правило для кожного лінгвістичного терма вихідної змінної;

2) для будь-якого терма вхідної змінної є хоча б одне правило, в якому цей терм використовується як передумови (ліва частина правила).

В іншому випадку має місце неповна база нечітких правил.

Формальне подання правила нечіткої продукції

$$(i): Q; P; A \longrightarrow B; S; F; N,$$

де (I) – ім'я нечіткої продукції;

Q – сфера застосування нечіткої продукції;

P – умова застосовності ядра нечіткої продукції;

$A \longrightarrow B$ - ядро нечіткої продукції, A - умова ядра,

B – висновок ядра;

$\longrightarrow$ - Логічна секвенція;

S – метод визначення кількісного значення ступеня істинності виводу ядра;

F – коефіцієнт визначеності або впевненості нечіткої продукції;

N – постумова продукції.

5.6.4.1 Механізм логічного виведення

У загальному випадку механізм логічного виведення (рис. 5.15) включає три етапи:

- 1) введення нечіткості (фаззифікація);
- 2) нечіткий висновок;
- 3) приведення до чіткості, або дефаззифікація.

Алгоритми нечіткого виведення розрізняються головним чином видом використовуваних правил, логічних операцій і різновидом методу дефаззифікації. Розроблено моделі нечіткого висновку Мамдані, Сугено, Ларсена, Цукамото.

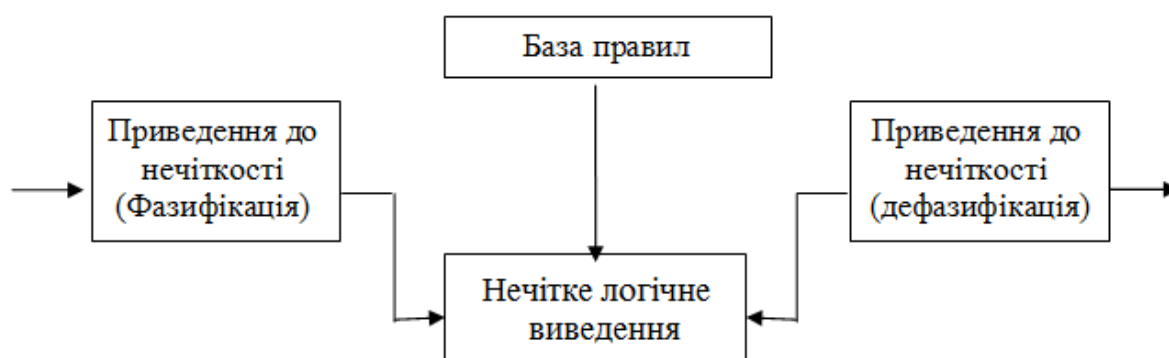


Рисунок 5.15 - Етапи нечіткого логічного виведення

Суть етапів логічного виведення

Фаззифікація - процедура знаходження значень функцій приналежності нечітких множин (термів) на основі звичайних (не нечітких) даних.

Мета етапу фаззифікації - встановлення співвідношення між конкретним (числовим) значенням окремої вхідної змінної системи нечіткого виводу і значенням функції приналежності відповідного їй терма вхідний лінгвістичної змінної. Після завершення цього етапу для всіх вхідних змінних повинні бути визначені конкретні значення функцій приналежності по кожному з лінгвістичних термів, які використовуються в підумовах бази правил.

Нечітке виведення - побудова нечіткого відношення на основі правила продукції.

Етапи нечіткого логічного виведення:

- агрегування - знаходження ступеня істинності умов кожного з правил із застосуванням парних нечітких логічних операцій;
- активізація - знаходження ступеня істинності кожного з підвисновків нечітких правил;
- акумуляція - знаходження функцій приналежності для вихідних лінгвістичних змінних.

Дефаззифікація - процедура знаходження звичайних (не нечітких) значень для кожної з вихідних лінгвістичних змінних.

Мета дефаззифікації - отримання звичайних кількісних значень кожної з вихідних змінних, використовуючи результати акумуляції всіх вихідних лінгвістичних змінних. Кількісні значення можуть бути використані спеціальними пристроями, зовнішніми по відношенню до системи нечіткого виведення.

5.6.4.2 Алгоритми нечіткого логічного виведення

Алгоритм Мамдані (Mamdani)

Алгоритм Мамдані (рис. 5.16) є найбільш поширеним способом логічного виведення в нечітких системах. У ньому використовується максимінна композиція нечітких множин.

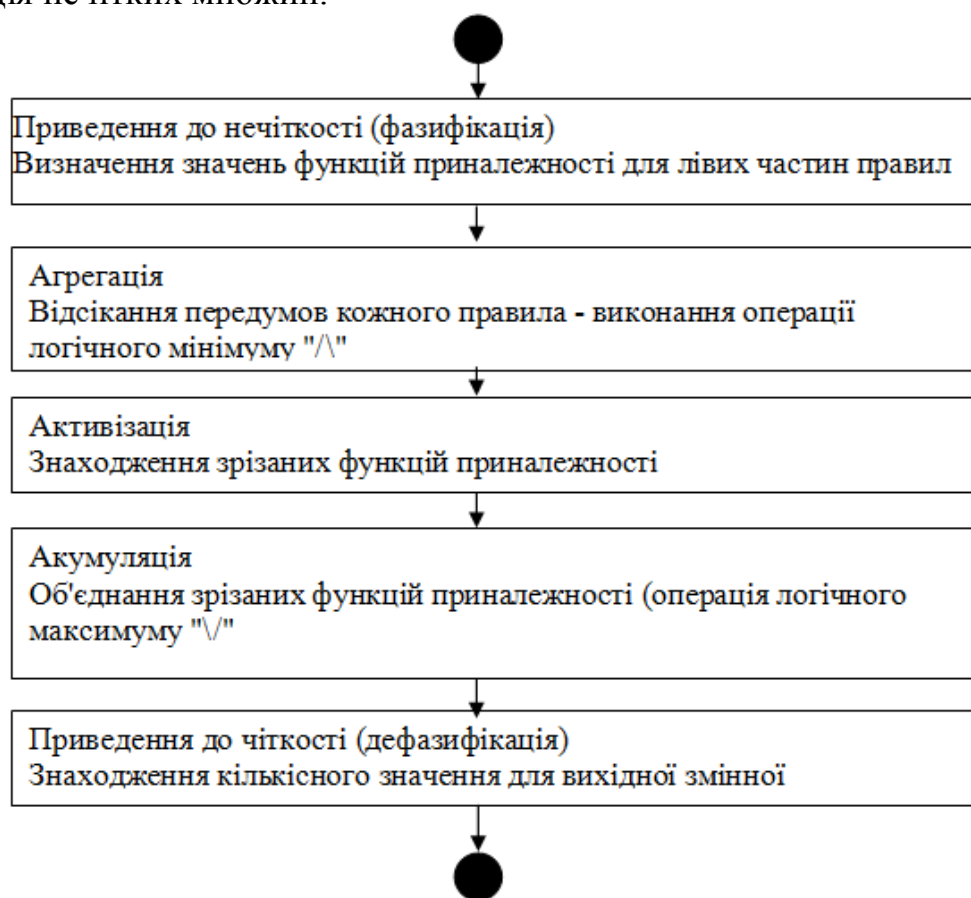


Рисунок 5.16 – Алгоритм нечіткого логічного виведення

На основі нечіткого правила формується нечітке відношення.

При виконанні нечіткого логічного виведення логічна операція згортки виконується за максимінним принципом.

Приклад

Розберемо основні особливості кожного з етапів нечіткого виведення, відображених на прикладі (рис. 5.17).

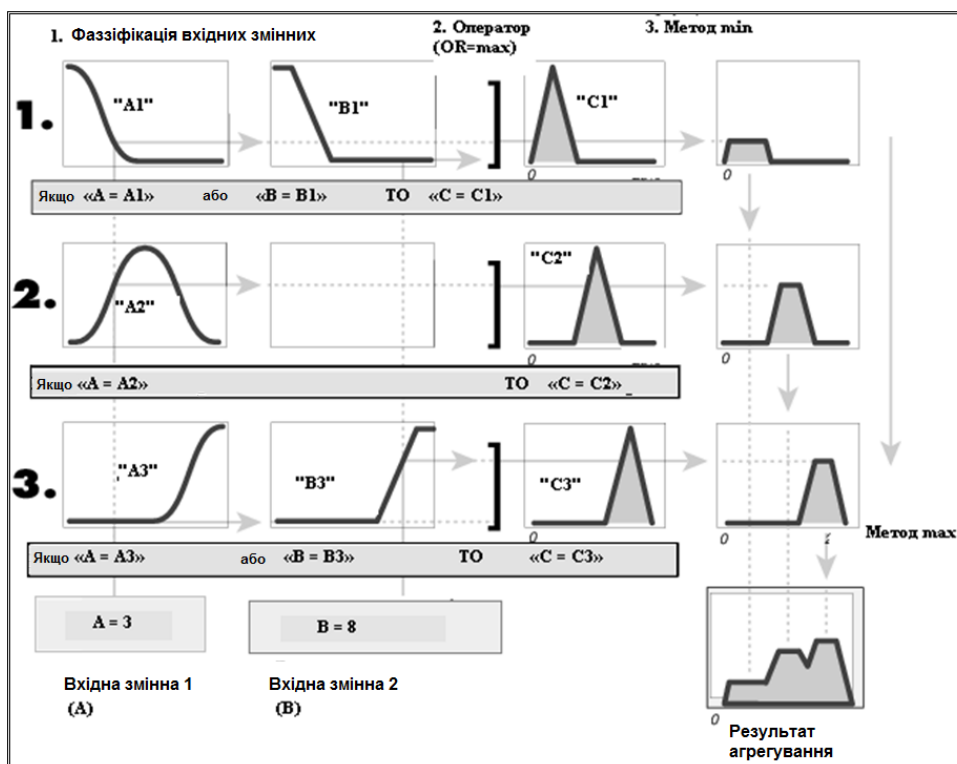


Рисунок 5.17 - Етап нечіткого виведення за алгоритмом Мамдані

Розглянемо модель, що складається з правила:

$$A, B \Rightarrow C,$$

де «A» і «B» - вхідні змінні, а «C» - вихідна.

Кожна із змінних може приймати відповідні значення зі своєї терм-множини.

У свою чергу, для кожного терму з терм-множин задається функція приналежності.

Завдання нечіткого виводу для даного прикладу є визначення числового значення для вихідної змінної C.

5.6.4.3 Формування бази правил системи нечіткого виведення

Процес формування бази правил нечіткого виведення є формальним поданням емпіричним знань експерта в тій чи іншій проблемній області. Найчастіше часто база правил має вигляд структурованого тексту:

Правило_1: Якщо «Умова_A1» або «Умова_B1» ТО «Висновок_C1»

Правило_2: Якщо «Умова_A2» або «Умова_B2» ТО «Висновок_C2»

...

Правило_n: Якщо «Умова_An» або «Умова_Bn» ТО «Висновок_Cn»,

де «Умова_ A1», «Умова_ A2», ..., «Умова_ An» і «Умова_ B1», «Умова_ B2», ..., «Умова_ Bn» - вхідні лінгвістичні змінні,

«Висновок_ C1», «Висновок_ C2», ..., «Висновок_ Cn» - вихідні лінгвістичні змінні.

Вхідні і вихідні лінгвістичні змінні вважаються визначеними, якщо для них задані функції приналежності. Так, на рис. 5.17 зображено етап формування трьох груп правил (етапи 1-3), де для кожної із змінних задані функції приналежності.

У системах нечіткого логічного виведення використовуються загальноприйняті скорочення для значень основних термів лінгвістичних змінних (табл. 5.7)

Таблиця 5.7 – Основні терми лінгвістичних змінних

Означення	Англомовна нотація	Україномовна нотація
NB	Negative Big	Негативне велике
NM	Negative Middle	Негативне середнє
NS	Negative Small	Негативне мале
ZN	Zero Negative	Негативне, близьке до нуля
Z	Zero	Нуль
ZP	Zero Positive	Позитивне, близьке до нуля
PS	Positive Small	Позитивне мале
PM	Positive Middle	Позитивне середнє
PB	Positive Big	Позитивне велике

5.6.4.4 Дефазифікація нечіткої множини

Результат дефазифікації для вихідної лінгвістичної змінної визначається у вигляді кількісного значення, одержуваного по одному з методів:

- 1) центру ваги;
- 2) центру тяжіння для одиночних множин;
3. центру площі;
- 4) лівого модального значення;
- 5) правого модального значення.

Метод центра ваги

Дефазифікація нечіткої множини за методом центру ваги здійснюється за формулою

$$y = \frac{\int_{Min}^{Max} x \times \mu(x) dx}{\int_{Min}^{Max} \mu(x) dx}.$$

де y - результат дефазифікації;

x - змінна, відповідна вихідній лінгвістичній змінній;

(x) - функція належності нечіткої множини, відповідного вихідної змінної після етапу акумуляції;

Min і Max - ліва і права точки інтервалу носія нечіткої множини вихідної змінної.

При дефазифікації за методом центру ваги звичайне (не нечітке) значення вихідної змінної дорівнює абсцисі центру тяжіння площі, обмеженої графіком кривий функції приналежності відповідної вихідної змінної.

Фізичним аналогом цієї формули є знаходження центра ваги плоскої фігури, обмеженої осями координат і графіком функції приналежності нечіткої множини.

Метод центра ваги для одиночних множин

$$y = \frac{\sum_{i=1}^n x_i \mu(x_i)}{\sum_{i=1}^n \mu(x_i)}$$

де n - число одноточкових (одноеlementних) нечітких множини, кожне з яких характеризує одне значення розглянутої вихідної лінгвістичної змінної.

Метод центра площі

Центр площі дорівнює абсцисі, яка ділить площу, обмежену графіком кривої функції приналежності відповідної вихідної змінної, на дві рівні частини. Іноді центр площі називають бісектрисою площі. Цей метод не може бути використаний у випадку одноточкових множин.

$$\int_{Min}^u \mu(x) dx = \int_u^{Max} \mu(x) dx ,$$

Метод лівого модального значення

Обчислюється за формулою

$$y = \min\{x_m\}$$

де x_m - модальне значення (мода) нечіткої множини, відповідного вихідної змінної після акумуляції.

Значення вихідної змінної визначається як мода нечіткої множини для відповідної вихідної змінної або найменша з мод (найлівіша), якщо нечітка множина має кілька модальних значень.

Метод правого модального значення

Обчислюється за формулою

$$y = \max\{x_m\}$$

де x_m - модальне значення (мода) нечіткої множини для вихідної змінної після акумуляції. У цьому випадку значення вихідної змінної також визначається як мода нечіткої множини для відповідної вихідної змінної або найбільша з мод (найправіша), якщо нечітка множина має кілька модальних значень.

5.6.4.5 Побудова функцій приналежності

Функції приналежності можуть бути побудовані на основі статистичної інформації (ступінь приналежності адекватний поняттю ймовірності події) або на основі експертних знань (ступінь приналежності розглядається, як інтенсивність прояву деякої властивості - відчуття). Методи побудови функції приналежності діляться на *прямі* і *непрямі*.

Прямі методи характеризуються тим, що експерт безпосередньо задає правила визначення значень функції приналежності $\mu_A(x)$, що характеризує елемент x . Прямі методи використовуються для вимірних понять, таких як швидкість, час, відстань, тиск, температура і т.д. прямі методи в основному використовуються в якості допоміжних, так як характеризуються великою часткою суб'єктивізму.

До числа прямих методів належать прямі групові методи, табличний, графічний, формульний методи. Табличний, графічний, формульний методи передбачають безпосереднє задання функції приналежності відповідно таблицею, графіком або формулою.

При використанні прямих групових методів, наприклад, групі експертів пред'являють конкретний об'єкт, і кожен повинен дати одну з двох відповідей: належить чи ні цей об'єкт до заданої множини. Число позитивних відповідей, поділене на загальне число експертів, дає значення функції приналежності об'єкта до даної нечіткої множини.

Непрямі методи побудови значень функції приналежності використовуються у випадках, коли немає елементарних вимірних властивостей, через які визначаються нечіткі множини. У непрямих методах значення функції приналежності вибираються таким чином, щоб задовольнити заздалегідь сформульованим умовам. Експертна інформація є тільки вихідною інформацією для подальшої обробки. Додаткові умови можуть накладатися як на вигляд одержуваної інформації, так і на процедуру її обробки. До таких методів належать статистичний метод, метод парних порівнянь, метод експертних оцінок.

За методом статистичних даних експерт реєструє частоту $p = k/n$ появи факту A і оцінює її за допомогою слів «часто», «рідко» і т.п. На універсальній шкалі $[0,1]$ він розміщує значення лінгвістичної змінної: «дуже рідко», «більш-менш рідко», «більш менш часто», «дуже часто». Ступінь приналежності деякого значення обчислюється як відношення числа експериментів, в яких воно зустрічалось в певному інтервалі шкали, до максимального для цього значення числа експериментів по всіх інтервалах. В кожен інтервал шкали повинно потрапляти однакове число експериментів.

Метод парних порівнянь заснований на обробці матриці оцінок, які відображають думку експерта про відносну приналежності елементів множини або ступеня вираженості у них деякої оцінюваної властивості. Ступінь приналежності елементів множини A буде визначатися за допомогою парних порівнянь.

Метод експертних оцінок застосовується для порівняння параметрів об'єктів, що знаходяться в одному "класі", однаковій категорії, і належить до різновиду мозкового штурму.

Параметричний метод заснований на припущенні, що експерт, який характеризує лінгвістичне значення якої-небудь ознаки, легко може вказати три точки шкали: A , B , C , з яких B і C - точки, на його думку, які ще (або вже) не належать даному лінгвістичному значенню, A - точка, безумовно належить йому.

Метод інтервальних оцінок застосовується для формалізованого подання завдань вибору, в яких відсутня чітка грань між допустимим і недопустимим (в просторі некерованих параметрів) і між ідеальним і незадовільним станами (в просторі критеріїв). Інтервальне оцінювання - один з видів статистичного оцінювання, що передбачає побудову інтервалу, в якому з певною ймовірністю знаходиться істинне значення оцінюваного параметра.

Метод рангових оцінок базується на ідеї розподілу ступеня приналежності елементів універсальної множини згідно з їх рангами. Рангом елемента є число, яке характеризує значимість цього елемента в формуванні властивості, яке описується нечітким термом. Чим більший ранг елемента, тим більше ступінь приналежності.

Переваги та недоліки методу нечітких множин

Використання методу нечітких множин дає ряд переваг, тому що дозволяє:

- включати в аналіз якісні змінні;
- оперувати нечіткими вхідними даними;
- оперувати лінгвістичними критеріями;
- швидко моделювати складні динамічні системи і порівнювати їх із заданим ступенем точності;
- долати недоліки і обмеження існуючих методів оцінки проектних ризиків.

Недоліки методу:

- існує суб'єктивність у виборі функцій приналежності і формуванні правил нечіткого введення;
- відсутність інформованості про метод, а також незначна увага до застосування методу професійними установами;
- необхідність спеціального програмного забезпечення, а також фахівців, що вміють з ним працювати.

Незважаючи на недоліки та обмеження теорії, метод нечітких множин отримав визнання як перспективний з точними результатами рядом найбільших міжнародних компаній (Motorola, General Electric, Otis Elevator, Pacific Gas Electric, Ford). Аналіз ризиків на основі статистичних методів для більшої частини нових компаній не застосовується, оскільки немає накопиченої статистичної інформації для отримання об'єктивних оцінок.

Контрольні запитання

1. Назвіть та охарактеризуйте види невизначеності, що розглядаються в ЕС.
2. Що називають коефіцієнтами впевненості? Де і як вони використовуються?
3. Поясніть метод використання нечітких знань, який застосовується в ЕС MYCIN.
4. Що називають граничними коефіцієнтами впевненості? Як вони використовуються?
5. Поясніть модель системи із дошкою оголошень.
6. Поясніть архітектуру та принцип роботи системи HEARSAY.
7. Охарактеризуйте Байєсовський підхід.
8. У чому суть теорії свідчень Демпстера-Шефера?
9. Що називають нечіткими множинами?
10. Що називають функцією приналежності?
11. Як можна побудувати функцію приналежності?
12. Що таке нечітка логіка?
13. Як виконується нечітке логічне виведення?
14. Що називають фаззифікацією?
15. Поясніть суть дефаззифікації.
16. Назвіть методи дефаззифікації.
17. Як працює агрегація?
18. У яких сферах застосовується нечітка логіка?
19. У яких сферах застосовується ймовірнісна логіка?
20. Що таке багатоступінчасте міркування?
21. Що таке коефіцієнти впевненості?
22. Як відбувається поширення коефіцієнтів впевненості у мережі?
23. Що таке граничні коефіцієнти впевненості?

6 МЕТОДИ ВИЛУЧЕННЯ ЗНАНЬ

6.1 Комунікативні методи

Комунікативні методи вилучення знань охоплюють методи та процедури контактів інженера зі знань з безпосереднім джерелом знань – експертом.

У свою чергу комунікативні методи можна поділити на активні та пасивні.

Пасивні методи мають на увазі, що провідну роль у процедурі вилучення знань грає експерт, а інженер зі знань лише протоколює висловлювання експерта під час його роботи.

В активних методах ініціативою володіє інженер зі знань, який контактує з експертом у різний спосіб: в іграх, діалогах, бесідах за круглим столом. Слід зазначити, що активні та пасивні методи можуть чергуватись навіть у рамках одного сеансу вилучення знань.

6.1.1 Пасивні методи

Термін «пасивний метод» є протилежністю терміну «активний метод», але насправді пасивні методи вимагають від інженера зі знань не меншої віддачі, ніж такі активні методи, як ігри та діалог.

У пасивних методах вилучення знань провідна роль у процедурі вилучення фактично передається експерту, а інженер зі знань лише фіксує міркування експерта під час роботи з прийняття рішень.

До групи пасивних методів належать спостереження; аналіз протоколів «думок вголос»; лекції.

Спостереження

У процесі спостережень інженер зі знань перебуває безпосередньо поруч із експертом під час його професійної діяльності чи імітації цієї діяльності. Під час підготовки до сеансу вилучення знань експерту необхідно пояснити мету спостережень і попросити максимально коментувати свої дії. Під час сеансу аналітик записує всі дії експерта, його репліки та пояснення. Може бути зроблений відеозапис у реальному масштабі часу. Неодмінна умова цього — невтручання аналітика в роботу експерта хоча б спочатку. Саме метод спостережень є єдино «чистим» методом, що виключає втручання інженера зі знань та нав'язування ним якихось своїх структур уявлень.

Інженер зі знань застосовує такі види спостереження:х

- спостереження за реальним процесом;
- спостереження за імітацією процесу.

Зазвичай використовуються обидва різновиди.

Спочатку інженеру зі знань корисно спостерігати за реальним процесом, щоб глибше зрозуміти предметну область та відзначити всі зовнішні особливості процесу ухвалення рішення. Це необхідно для проектування ефективного користувача інтерфейсу. Адже майбутня система має працювати саме у контексті такого реального виробничого процесу. Крім того, лише спостереження дозволить аналітику побачити предметну область.

Спостереження за імітацією процесу проводять зазвичай також на робочому місці експерта, але процес діяльності запускається спеціально для аналітика.

Перевага цього різновиду в тому, що експерт менш напружений, ніж у першому варіанті, коли він працює на два фронти — і веде професійну діяльність, і демонструє її. Недоліки збігаються з перевагою — саме менша напруженість експерта може вплинути на результат — оскільки робота несправжня, то й рішення може відрізнятись від дійсного.

Спостереження за імітацією проводять також у тих випадках, коли спостереження за реальним процесом з будь-яких причин неможливі (наприклад, професійна етика лікаря-психіатра може не допускати присутності стороннього на прийомі).

Сеанси спостережень можуть вимагати від інженера зі знань:

- оволодіння технікою стенографії для фіксації дій експерта у реальному масштабі часу;
- ознайомлення з методиками хронометражу для чіткого структурування виробничого процесу у часі;
- розвитку навичок «читання по очах», тобто спостережливості до жестів, міміки та інших невербальних компонентів спілкування;
- серйозного попереднього знайомства з предметною областю, оскільки через відсутність «зворотного зв'язку» іноді багато незрозуміло у діях експертів.

Протоколи спостережень після сеансів під час домашньої роботи ретельно розшифровуються, потім обговорюються з експертом. Отже, спостереження — один із найпоширеніших методів вилучення знань на початкових етапах розробки. Зазвичай він застосовується не самотійно, а разом із іншими методами.

Аналіз протоколів «думки вголос»

Протоколювання «думок вголос» відрізняється від спостережень тим, що експерта просять не просто прокоментувати свої дії та рішення, а й пояснити, як це рішення було знайдено, тобто продемонструвати весь ланцюжок своїх міркувань. Під час міркувань експерта всі його слова, весь «потік свідомості» протоколюється інженером зі знань, при цьому корисно відзначити навіть паузи та вигуки. Іноді цей метод називають "вербальні звіти".

Питання використання для цієї мети диктофонів є дискусійним, оскільки диктофон іноді паралізує діє експерта, руйнуючи атмосферу довірливості, що може і має виникати при безпосередньому спілкуванні.

Основною проблемою під час протоколювання «думок вголос» є складність для будь-якої людини пояснити, як вона думає. При цьому існують експериментальні психологічні докази того факту, що люди не завжди можуть достовірно описувати розумові процеси. Крім того, частина знань, що зберігаються в невербальній формі (наприклад, різні процедурні знання типу «як зав'язувати шнурки») взагалі слабо корелюють з їх словесним описом. Автор теорії фреймів М. Мінський вважає, що «тільки як виняток, а не як правило, людина може пояснити те, що вона знає». Проте існують люди, схильні до рефлексії, котрим ця робота є цілком доступною. Отже, така характеристика когнітивного стилю як рефлексивність є для експерта більш ніж бажаною.

Рефлексія – здатність спрямовувати розумовий процес на власну свідомість, поведінку, накопичені знання, здійснені та майбутні дії. Говорячи простою мовою, рефлексія є здатністю людини зазирнути у себе. Причому заглянути можна не лише у свою, а й у чужу свідомість.

Рефлексувати означає фокусуватися і осмислювати зміст власної свідомості.

Рефлексивність - це здатність індивіда вийти за рамки свого "я", розмірковувати, аналізувати і робити висновки, порівнювати своє "я" з іншими. Це унікальна можливість щоб критично подивитись на себе збоку.

Розшифрування отриманих протоколів проводиться інженером зі знань самостійно з корекціями на наступних сеансах отримання знань. Вдало проведене протоколювання «думок вголос» один з найефективніших методів вилучення, оскільки у ньому експерт може проявити себе максимально яскраво, він нічим не скований, ніхто йому не заважає, він ніби вільно рухається в потоці власних міркувань і міркувань. Він може тут блиснути ерудицією, продемонструвати глибину своїх знань. Для великої кількості експертів це найприємніший спосіб вилучення знань. Від інженера зі знань метод «думки вголос» вимагає тих самих умінь, як і метод спостережень. Зазвичай «думки вголос» доповнюються потім одним із активних методів для реалізації зворотного зв'язку між інтерпретацією інженера зі знань та уявленнями експерта.

Лекції

Лекція є найдавнішим способом передачі знань. Лекторське мистецтво з давніх-давен дуже високо цінувалося у всіх галузях науки і культури. Але при вилученні знань важливою є не так здатність до підготовки та читання лекцій, скільки здатність цю лекцію слухати, конспектувати та засвоювати.

Найчастіше експертів не обирають, і тому вчити експерта читання лекції інженер із знань не зможе. Але якщо експерт має досвід викладача (наприклад, професор клініки чи досвідчений керівник виробництва), можна скористатися таким концентрованим фрагментом знань, як лекція.

У лекції експерту також надано багато ступенів свободи для самовираження; при цьому необхідно сформулювати експерту тему та завдання лекції. Наприклад, тема циклу лекцій «Постановка діагнозу — запалення

легень», тема конкретної лекції «Міркування з аналізу рентгенограм», завдання — навчити слухачів за перерахованими експертами ознаками ставити діагноз запалення легень і ставити прогноз. При такій постановці досвідчений лектор може заздалегідь структурувати свої знання та перебіг міркувань. Від інженера зі знань у цій ситуації потрібно лише грамотно законспектувати лекцію і насамкінець поставити необхідні питання.

Студенти добре знають, що конспекти лекцій одного й того ж лектора у різних студентів суттєво відрізняються. Списати конспект лекцій просять, як правило, в одного-двох студентів із групи. Люди, які вміло ведуть конспект, зазвичай є сильними студентами. Зворотне не є вірним.

Мистецтво ведення конспекту полягає у «завадо стійності». Записувати головне, опускаючи другорядне, виділяти фрагменти знань (параграфи, підпараграфи), записувати лише осмислені речення, вміти узагальнювати. Добре питання по ходу лекції допомагає і лектору, і слухачеві. Серйозні та глибокі питання можуть суттєво підняти авторитет інженера зі знань в очах експерта.

Досвідчений лектор знає, що всі питання можна умовно розбити на три групи:

- 1) розумні питання, які поглиблюють лекцію;
- 2) дурні питання чи питання не по суті;
- 3) питання «на засипку» чи провокаційні.

Якщо інженер зі знань ставить питання другого типу, то можливі дві реакції. Ввічливий експерт розмовлятиме з таким аналітиком як з дитиною, яка зараз не розуміє і все одно нічого вже не зрозуміє. Зарозумілий експерт просто вийде з контакту, не бажаючи гаяти час.

Якщо ж інженер зі знань захоче продемонструвати свою ерудицію питаннями третього типу, то нічого, крім роздратування та відчуження, він, мабуть, у відповідь не отримає.

Тривалість лекції рекомендується стандартна – від 40 до 50 хвилин та через 5-10 хвилин – ще стільки ж.

Курс зазвичай від двох до п'яти лекцій.

Метод отримання знань у формі лекцій, як і всі пасивні методи, використовують на початку розробки як ефективний спосіб швидкого занурення інженера з знань у предметну область.

Поради про те, як слухати лекції.

1. До лекції підготуйтеся, тобто ознайомтеся з предметною областю.
2. Слухайте з максимальною увагою, для цього: усуньте фактори, що заважають (скрип дверей, шарудіння і т. д.); зручно влаштуйтеся; менше рухайтеся.
3. Навчіться відпочивати під час слухання (наприклад, коли лектор наводить цифри, які можна взяти з довідника).
4. Слухайте одночасно і лектора, і самого себе (паралельно з думками лектора щодо асоціації виникають власні думки).
5. Слухайте і одночасно записуйте, але записуйте текст скорочено, використовуючи умовні значки (для цього зовсім не слід неодмінно

бути стенографом, достатньо встановити собі ряд умовних значків і ними незмінно користуватися).

6. Розшифруйте записи лекції того ж дня.
7. Не сперечайтесь з лектором у процесі лекції.
8. Раціонально використовуйте перерви у лекції для підбиття підсумків прослуханого.

Характеристики пасивних методів вилучення знань дано в табл. 6.1.

Таблиця 6.1 – Порівняльні характеристики пасивних методів вилучення знань

Метод	Спостереження	«Думки вголос»	Лекції
Переваги спостереження	Відсутність впливу аналітика та його суб'єктивної позиції. Максимальне наближення аналітика до предметної області	Свобода самовираження для експерта Відкритість структур міркувань Відсутність впливу аналітика та його суб'єктивної позиції	Свобода самовираження для експерта Структурований виклад. Висока концентрація Відсутність впливу аналітика та його суб'єктивної позиції
Недоліки	Відсутність зворотного зв'язку Фрагментарність отриманих коментарів	Відсутність зворотного зв'язку Можливість відходу «убік» у міркуваннях експерта	«Зашумленість» деталями Слабкий зворотний зв'язок Недолік хороших лекторів серед експертів-практиків
Вимоги до експерта (типи та основні якості)	Співрозмовник чи мислитель: здатність до вербалізації думок, аналітичність, відкритість, рефлексивність	Співрозмовник чи мислитель: здатність до вербалізації думок, аналітичність, відкритість, рефлексивність	Мислитель: лекторські здібності
Вимоги до аналітика (типи та основні якості)	Мислитель: спостережливість, полenezалежність	Мислитель чи співрозмовник: контактність, полenezалежність	Мислитель: полenezалежність, здатність до спілкування
Характеристика предметної області	Слабо та середньо структуровані; слабо та середньо документовані	Слабо та середньо структуровані; слабо та середньо документовані	Слабо та середньо структуровані; слабо та середньо документовані

6.1.2 Активні індивідуальні методи

Активні індивідуальні методи вилучення знань на сьогоднішній день є найпоширенішими. Тією чи іншою мірою до них вдаються розробники практично будь-якої системи.

До основних активних методів можна віднести анкетування, інтерв'ю, вільний діалог, ігри з експертом.

У всіх цих методах активну функцію виконує інженер зі знань, який пише сценарій та режисує сеанси отримання знань. Ігри з експертом суттєво відрізняються від трьох інших методів. Три методи, що залишилися, дуже схожі між собою і відрізняються лише за ступенем свободи, яку може собі дозволити інженер зі знань при проведенні сеансів отримання знань. Їх можна назвати методами пошуку знань, головну роль у яких відіграють запитання.

Анкетування

Анкетування — це найбільш стандартизований метод. Застосовуючи його інженер зі знань заздалегідь складає опитувальник або анкету, розмножує її та використовує для опитування кількох експертів. Це основна перевага анкетування.

Процедура анкетування може проводитись двома способами:

- 1) аналітик вголос ставить запитання і сам заповнює анкету за відповідями експерта;
- 2) експерт самостійно заповнює анкету після попереднього інструктування.

Вибір способу залежить від конкретних умов (наприклад, від оформлення анкети, її зрозумілості, готовності експерта).

Другий спосіб є кращим, оскільки в експерта з'являється необмежений час на обмірковування відповідей.

У схемі спілкування (рис. 4.7) основними чинниками, на які можна суттєво вплинути під час анкетування, є засоби спілкування (у даному разі це опитувальник) і ситуація спілкування. Перелік запитань (анкета) заслуговує на особливу увагу.

Рекомендації для складання анкет є універсальними, тобто не залежить від предметної області. Найбільший досвід роботи з анкетами накопичено у соціології та психології.

Існує кілька загальних рекомендацій для складання анкет.

1. Анкета не повинна бути монотонною та одноманітною, тобто викликати нудьгу чи втому. Це досягається варіаціями форми питань, зміною тематики, вставкою питань-жартів та ігрових питань.
2. Анкета має бути пристосована до мови експертів.
3. Слід враховувати, що питання впливають одне на одного і тому послідовність питань має бути чітко продумана.
4. Бажано прагнути оптимальної надмірності. Відомо, що в анкеті завжди багато зайвих питань, частина з них необхідна (контрольні запитання), а інша частина має бути мінімізована.

5. Анкета повинна мати «хороші манери», тобто її мова має бути чіткою, зрозумілою, гранично ввічливою. Методичною майстерністю складання анкети можна опанувати лише на практиці.

Інтерв'ю

Під інтерв'ю розуміють специфічну форму спілкування інженера зі знань та експерта, в якій інженер зі знань задає експерту серію заздалегідь підготовлених запитань з метою отримання знань про предметну область.

Інтерв'ю стоїть дуже близько до того способу анкетування, коли аналітик сам заповнює анкету, заносючи туди відповіді експерта. Основна відмінність інтерв'ю в тому, що воно дозволяє аналітику опускати низку запитань залежно від ситуації, вставляти нові запитання в анкету, змінювати темп, урізноманітнити ситуацію спілкування. Крім цього, у аналітика з'являється можливість позитивно налаштувати експерта, зацікавити його процедурою і тим самим збільшити ефективність сеансу вилучення знань.

Запитання для інтерв'ю

Запитання є центральною частиною активних індивідуальних методів. Інженери зі знань рідко сумніваються у своїй здатності ставити запитання, тоді як і у філософії та в математиці ця проблема обговорюється з давніх-давен. Існує навіть спеціальна галузь математичної логіки – еротетична логіка (логіка ведення діалогу у формі запитань і відповідей).

Всі запитальні речення можна розбити на два типи:

- запитання з невизначеністю, що стосується всієї пропозиції («Справді, введення великих доз антибіотиків може спричинити анафілактичний шок?»);
- запитання з неповною інформацією («За яких умов необхідно включати кнопку?»), які часто починаються зі слів «хто», «що», «де», «коли» тощо.

Цей поділ можна доповнити класифікацією, наведеною на рис. 6.1.

Відкрите запитання називає тему чи предмет, залишаючи повну свободу експерту за формою та змістом відповіді («Не могли б ви розповісти, як краще збити високу температуру у хворого із запаленням легень?»).

У закритому питанні експерт вибирає відповідь із набору запропонованих («Вкажіть, будь ласка, що ви рекомендуєте при ангіні:

- а) антибіотики,
- б) полоскання,
- в) компреси,
- г) інгаляції.

Закриті запитання легше обробляти при подальшому аналізі, але вони «закривають» перебіг міркувань експерта та «програмують» його відповідь у певному напрямку.

При складанні сценарію інтерв'ю корисно чергувати відкриті і закриті запитання, особливо ретельно продумувати закриті, оскільки їх для складання потрібна певна ерудиція у предметній області.

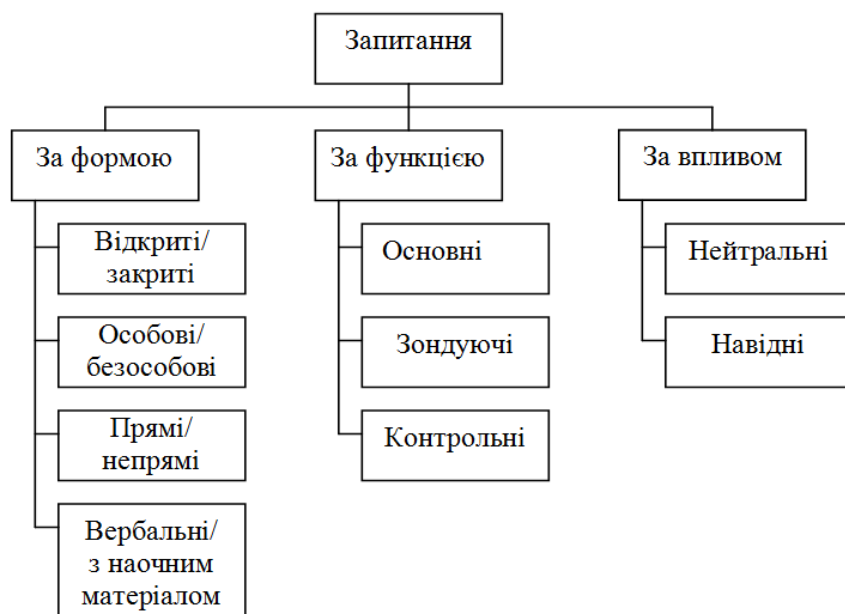


Рисунок 6.1 – Класифікація запитань

Особисте запитання стосується безпосередньо особистого індивідуального досвіду експерта («Скажіть, будь ласка, Іване Даниловичу, у вашій практиці ви застосовуєте вулнузан при фурункульозах?»).

Особисті запитання зазвичай активізують мислення експерта, "грають" на його самолюбстві, вони завжди прикрашають інтерв'ю. Безособове запитання спрямовано на виявлення найпоширеніших і загальноприйнятих закономірностей предметної області («Що впливає швидкість процесу ферментації лізину?»).

При складанні запитань слід враховувати, що мовні здібності експерта, як правило, обмежені і внаслідок скутості, замкнутості, боязкості він не може відразу висловити свою думку та надати знання, які від нього вимагаються (навіть якщо припустити, що він їх чітко для себе формулює). Тому часто при «затисненості» експерта використовують не прямі запитання, які безпосередньо вказують на предмет або тему («Як ви ставитеся до методики доктора Сухарєва?»), а непрямі, які лише побічно вказують на предмет, що цікавить («Чи застосовуєте ви методику доктора Сухарєва? Опишіть, будь ласка, результати лікування»).

Іноді доводиться ставити кілька десятків непрямих запитань замість прямого.

Вербальні запитання – це традиційні усні запитання.

Запитання з використанням наочного матеріалу урізноманітнюють інтерв'ю та знижують стомлюваність експерта. У цих питаннях використовують фотографії, малюнки та картки. Наприклад, експерту пропонуються кольорові картонні картки, на яких виписано ознаки захворювання. Потім аналітик вимагає розкласти ці картки в порядку зменшення значущості для постановки діагнозу.

Розподіл запитань за функцією на основні, зондуючі, контрольні пов'язаний з тим, що часто основні запитання інтерв'ю, спрямовані на виявлення знань, не спрацьовують — експерт з якихось причин уникає запитання, відповідає нечітко. Тоді аналітик використовує зондуючі запитання, які спрямовують міркування експерта у потрібний бік. Наприклад, якщо не спрацювало основне запитання: "Які параметри визначають момент закінчення процесу ферментації лізину?" — аналітик починає ставити зондуючі запитання: «Чи завжди процес ферментації триває 72 години? А якщо він закінчується раніше, як це дізнатися? Якщо він триватиме більше, то що змусить мікробіолога не закінчити процес на 72-й годині?» і т.д.

Контрольні запитання застосовують для перевірки достовірності та об'єктивності інформації, отриманої в інтерв'ю раніше («Скажіть, будь ласка, а київська школа психологів так само як ви трактує шкалу опитувальника?» або «Чи рекомендуєте ви ін'єкції АТФ?» (АТФ — препарат, знятий з виробництва).

Контрольні запитання мають бути ретельно складені, щоб не образити експерта недовірою (для цього використовують повторення запитань в іншій формі, уточнення, посилення на інші джерела). «Краще двічі запитати, ніж один раз наплутати».

Нейтральні та навідні запитання. У принципі інтерв'юєру (у нашому випадку інженеру зі знань) рекомендують бути неупередженим, звідси і його запитання мають носити нейтральний характер, тобто не повинні вказувати на ставлення інтерв'юєра до цієї теми. Навпаки, навідні питання змушують респондента (експерта) прислухатися і навіть взяти до уваги позицію інтерв'юєра. Нейтральне запитання: «Чи збігаються симптоми крововиливу в мозок і струсу мозку?» Навідне запитання: «Дуже важко диференціювати симптоми крововиливу в мозок, чи не так?».

Крім наведених вище, корисно розрізняти та включати в інтерв'ю такі запитання:

- контактні (що «ламають лід» між аналітиком та експертом);
- буферні (для розмежування окремих тем інтерв'ю);
- оживляючі пам'ять експертів (для реконструкції окремих випадків із практики);
- «провокуючі» (для отримання спонтанних, непередбачених відповідей).

Основні характеристики запитань, що впливають на якість інтерв'ю:

- мова питання (зрозумілість, лаконічність, термінологія);
- порядок питань (логічна послідовність та немонотонність);
- доречність питань (етика, ввічливість).

Запитання в інтерв'ю — це не просто засіб спілкування, а й спосіб передачі думок та позиції аналітика. Запитання є формою руху думки, у ньому яскраво виражений момент переходу від незнання до знання, від неповного,

неточного знання до більш повного і більш точного. Звідси необхідність у протоколах фіксувати як відповіді, так і запитання, попередньо ретельно відпрацьовуючи їхню форму і зміст.

Очевидно, що будь-яке питання має сенс лише у контексті. Тому запитання може готувати інженер зі знань, що вже опанував ключовий набір знань. Запитання мають для експерта діагностичне значення — кілька відвертих «дурних» питань можуть розчарувати експерта і відбити в нього бажання до подальшої співпраці.

Вільний діалог

Вільний діалог — метод отримання знань у формі розмови інженера зі знань та експерта. У вільному діалозі немає жорстко регламентованого плану та опитувальника. Проведення вільного діалогу потребує найвищої професійної та психологічної підготовки за схемою рис. 6.2, що дозволяє уникнути помилок.

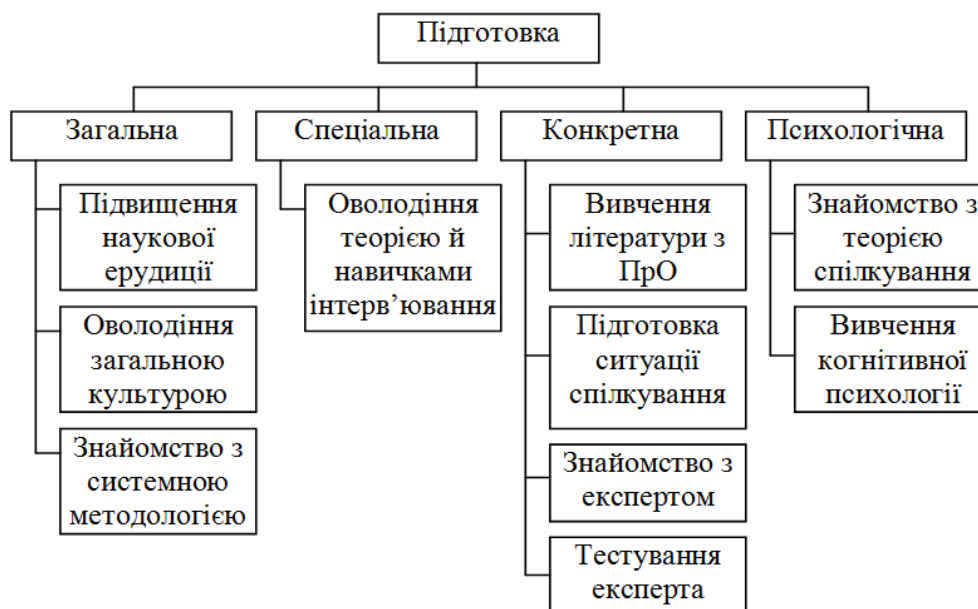


Рисунок 6.2 – Підготовка до вилучення знань

Аналітик планує гладкий перебіг процедури отримання знань: формування приємного враження на початку бесіди, перехід до професійного контакту через пробудження інтересу та завоювання довіри експерта, підтвердження впевненості експерта у власній компетентності. Інженер зі знань повинен мати здоровий вигляд, бути спокійним, впевненим, вселяти довіру, бути щирим, веселим, виявляти інтерес до розмови, охайно одягненим, доглянутим.

Важливим є темп бесіди: без великих пауз, інакше експерт може відволіктися, без гонки, бо гонка втомлює обох учасників та викликає напруженість, особливо коли експерт думає повільно. Слід чергувати різні темпи, оскільки це значно покращує результат.

Під час підготовки до вільного діалогу слід враховувати такі стадії його проведення:

- 1) початок розмови – знайомство, створення позитивного образу аналітика, пояснення цілей та завдань роботи;
- 2) діалог із вилучення знань;
- 3) заключна стадія – подяка експерту, підбиття підсумків, домовленість про наступні зустрічі.

Діалог передбачає:

- 1) унікальність кожного партнера, їхню принципову рівність один одному;
- 2) відмінність та оригінальність точок зору;
- 3) орієнтацію кожного на розуміння та активну інтерпретацію його погляду партнером;
- 4) очікування відповіді та її попередження у своєму висловлюванні;
- 5) взаємну доповнюваність позицій учасників діалогу, співвіднесення яких є метою діалогу.

Порівняльні характеристики активних індивідуальних методів вилучення знань наведені в табл. 6.2.

Таблиця 6.2 - Характеристики активних індивідуальних методів вилучення знань

Метод	Анкетування	Інтерв'ю	Вільний діалог
Переваги	Можливість стандартизованого опитування кількох експертів Не потребує особливого напруження від аналітика під час анкетування	Наявність зворотного зв'язку (уточнення та усунення протиріч) для експерта	Гнучкість Сильний зворотний зв'язок Можливість зміни сценарію та форми сеансу
Недоліки	Потребує вміння та досвіду у складанні анкет Відсутність зворотного зв'язку Можливість неправильного розуміння запитань експертом	Значний час на підготовку запитань інтерв'ю	Високе напруження аналітика Відсутність формальних методик проведення Складність протоколювання результатів
Вимоги до експерта	Практик та мислитель	Співбесідник або мислитель	Співбесідник або мислитель
Вимоги до аналітика	Мислитель, уважний, педант у складанні анкет	Співбесідник, вміння слухати, задавати запитання	Співбесідник, спостережливий, вміння слухати
Характеристики предметної області	Слабо структурована, слабо та середньо документована	Слабо структурована, слабо та середньо документована	Слабо структурована, слабо та середньо документована

6.1.3 Активні групові методи

Активні групові методи самі по собі не можуть бути джерелом повних знань, вони застосовуються разом з індивідуальними методами.

Перевагою активних групових методів є можливість вилучення знань одночасно від кількох експертів. У процесі взаємодії експертів виявляються нові аспекти знань у результаті викладу та обговорення різних поглядів та позицій.

До активних групових методів належать «круглий стіл» за участю кількох експертів, мозковий штурм, рольові ігри.

«Круглий стіл»

Метод круглого столу передбачає обговорення будь-якої проблеми з обраної предметної області, у якому беруть участь кілька експертів із рівними правами.

Спочатку учасники (3-7 осіб) висловлюються у певному порядку, а потім переходять до живої вільної дискусії.

Під час підготовки та проведення круглого столу слід враховувати специфіку поведінки людини у групі.

1. Інженеру зі знань необхідно враховувати як організаційні моменти (правильно вибрати місце, час, обстановку, забезпечити мінеральну воду, чай, кворум), і психологічні (уміння вставляти доречні репліки, виявляти почуття гумору, пам'ять на імена, здатність гасити конфліктні ситуації) .

2. Багато учасників будуть говорити, бажаючи справити враження на інших експертів, говорити не про те, що дійсно цікаво їм, а те, що демонструє їхню власну компетентність. Хід розмови доцільно фіксувати як звуковий запис, а при розшифровці та аналізі результатів враховувати взаємні відносини учасників.

Завдання круглого столу – колективно, з різних поглядів досліджувати спірні гіпотези предметної області. Для гостроти та уникнення односторонності запрошують представників різних наукових шкіл та поглядів, різних поколінь. Слід розрізняти дискусію (метод наукового пізнання, уміння розвивати теорію, а не просто суперечка) і полеміку (ворожу і войовничу суперечку).

Перед початком круглого столу слід:

- переконатися, що всі правильно розуміють завдання – вилучення знань;
- встановити регламент;
- чітко сформулювати тему.

Під час дискусії важливо простежити, щоб надто емоційні та балакучі експерти не підмінили тему, і щоб критика один одного була обґрунтованою.

«Мозковий штурм»

Порівняльні характеристики методів круглого столу та мозкового штурму наведено в табл. 6.3.

«Мозковий штурм» - один із найпоширеніших методів розкріпачення та активізації творчого мислення.

Як правило, штурм триває близько 40 хвилин. Число учасників – до 10 осіб. Учасникам пропонується висловлювати будь-які ідеї на задану тему – жартівливі, фантастичні, помилкові, критика заборонена. Зазвичай висловлюється до 50 ідей. Регламент виступу – до 2 хвилин. У процесі штурму настає піковий момент, коли відбувається мимовільна генерація ідей учасниками, 10-15% з яких є розумними. Результати зазвичай оцінює група експертів, які брали участь у генерації. Ведучий штурму (інженер зі знань) повинен вільно володіти аудиторією, підібрати активну групу експертів-«генераторів», не «затискати» погані ідеї, вміти ставити запитання, зупиняти багатослівних експертів.

Основне правило штурму – що більше ідей, то краще. Фіксація ходу штурму – звуковий запис.

Порівняльні характеристики методів круглого столу та мозкового штурму наведені у табл. 6.3.

Таблиця 6.3 - Порівняльні характеристики методів круглого столу та мозкового штурму

Метод	Круглий стіл	Мозковий штурм
Переваги	Отримання більш об'єктивних фрагментів знань Пожвавлення процедури вилучення знань Обмін знаннями між учасниками	Виявлення глибинних знань на рівні підсвідомого Активізація експертів Отримання нових знань, гіпотез
Недоліки	Великі організаційні витрати Складність проведення	Можливий тільки для нових цікавих проблем Низький відсоток продуктивних ідей
Вимоги до експерта	Співбесідник або мислитель, що володіє мистецтвом полеміки	Мислитель, який має здатність до творчості
Вимоги до аналітика	Співбесідник з дипломатичними здібностями	Співбесідник або мислитель зі швидкою реакцією та почуттям гумору
Характеристики предметної області	Слабо структурована, слабо документована з наявними спірними проблемами	Слабо структурована, слабо документована з наявними перспективними моментами

Експертні ігри

Гра – вид людської діяльності, що відбиває та відтворює інші види діяльності. Для гри характерні одночасно і умовність та серйозність.

Експертні ігри можна умовно поділити на

- ділові ігри – використовуються при підготовці фахівців та моделюванні;
- діагностичні ігри;
- комп'ютерні ігри – застосовуються у навчанні.

Ділова гра – експеримент, в якому учасникам пропонується реальна ситуація, а вони на основі свого життєвого досвіду, загальних та спеціальних

знань та уявлень приймають рішення. У результаті наступного аналізу рішень знаходять закономірності мислення учасників гри, в такий спосіб виявляються знання.

Якщо учасниками ділової гри є експерти, то така гра називається експертною.

Ділові ігри поділяються на навчальні, планово-виробничі та дослідні. До експертів найближче дослідницькі ігри, що використовуються для аналізу систем, перевірки правил ухвалення рішень.

Діагностична гра – ділова гра із жорстко закріпленою предметною областю, наприклад, медициною. Вона застосовується для діагностики методів прийняття рішень у предметній області. У ході гри досвід передається від досвідчених учасників новачкам.

Моделювання реальних ситуацій в іграх є актуальним для всіх областей науки і техніки. Вони розвивають логічне мислення, уміння швидко ухвалювати рішення, викликають інтерес у експертів.

Експертні ігри можна розділити на індивідуальні та групові.

Крім того, експертні ігри можна розділити і за іншими критеріями на ігри з використанням спеціального обладнання та гри із застосуванням обчислювальної техніки (рис. 6.3).

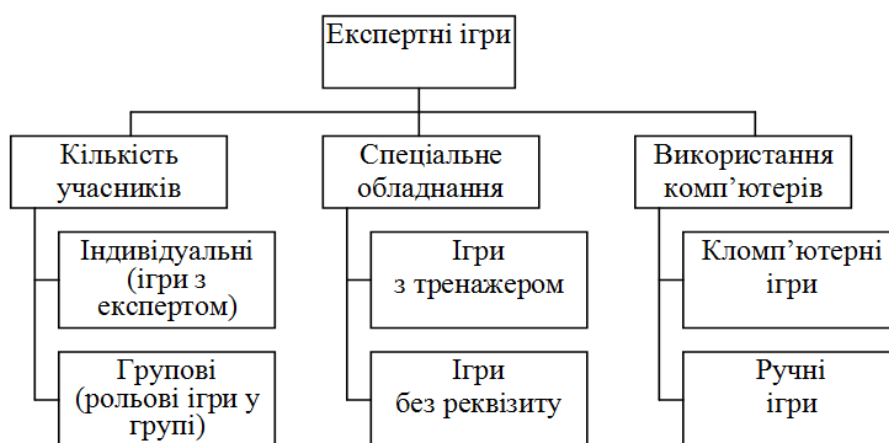


Рисунок 6.3 – Класифікація експертних ігор

Ігри з експертом

З експертом грає інженер зі знань, який бере на себе певну роль у ситуації, що моделюється.

Приклади:

- вчитель-учень – інженер зі знань грає роль учня і під наглядом експерта виконує його роботу, а експерт оцінює результати; доцільно застосовувати під час роботи з неговірким експертом;
- консультант – інженер зі знань грає роль особи, що приймає рішення, яка добре знає об'єкт експертизи; експерт відіграє роль консультанта,

ставити запитання та висловлюється про доцільність застосування призначеного методу виправлення несправності, лікування;

- експерта просять написати обґрунтування прогнозу для об'єкту експертизи, збирають кілька таких обґрунтувань від різних експертів, а через деякий час експерту зачитують його власне обґрунтування та просять зробити прогноз; зазвичай він цього зробити не може через неповноту свого обґрунтування; експерт доповнює своє обґрунтування, при цьому виявляються приховані пласти знань;
- фокусування на контексті – експерт відіграє роль ЕС, а інженер із знань – роль користувача; розігрується ситуація консультації, і перші запитання експерта виявляють найбільш значущі поняття та важливі аспекти проблеми.

Поради щодо проведення індивідуальних ігор:

- 1) грайте сміливіше, вигадуйте ігри самі;
- 2) не нав'язуйте гру експерту, якщо він не налаштований;
- 3) у грі не «тисніть» на експерта і не забувайте мету гри;
- 4) грайте вільно, нешаблонно;
- 5) не забувайте про час – гра стомлює експерта.

Рольові ігри у групі

Роль - комплекс зразків поведінки. Роль пов'язані з іншими ролями.

Групові ігри передбачають участь кількох експертів. Для гри складається сценарій, розподіляються ролі, до кожної ролі готується портрет-опис та розробляється система оцінювання гравців.

Інженер зі знань вибирає форму проведення гри: гравці можуть самі придумати собі імена для гри; усі гравці можуть переходити на «ти»; гравці можуть самі обирати ролі; гравці ролі тягнуть за жеребом.

Зазвичай у грі 3-6 експертів, якщо більше, їх можна розбити на конкуруючі бригади і оцінювати результати кожної.

Для підготовки гри слід надрукувати таблички із назвами ролей, імен, пам'ятки з правилами гри тощо. Мета гри – розкріпачити експертів для передачі знань.

Ігри з тренажерами

Ігри з тренажерами близькі до імітаційних вправ у ситуації, наближеній до дійсності.

Наявність тренажера дозволяє максимально відтворити реальну ситуацію та спостерігати за експертом. Використання тренажерів дозволяє зафіксувати фрагменти «летючих» знань, що виникають під час та на місці реальних ситуацій та випадають при виході за межі ситуації.

Комп'ютерні експертні ігри

Комп'ютерні експертні ігри різних жанрів можуть застосовуватися для отримання знань. Однак, розробка таких ігор є дорогою та витратною за часом.

Порівняльні характеристики експертних ігор наведено у табл. 6.4.

Таблиця 6.4 - Порівняльні характеристики експертних ігор

Види ігор	Індивідуальні	Групові	Комп'ютерні
Переваги	Швидке отримання якісної картини прийняття рішень Виявлення інформації, використовуваної експертом	Створена ситуація реально відповідає задачі Розкріпачення експертів Групові знання є більш об'єктивними Виявлення логіки та аргументів експертів	Викликають інтерес експерта Приваблюють експерта дизайном і динамікою
Недоліки	Відсутні методики стандартного набору ігор Високі професійні вимоги до аналітика	Відсутні методики стандартного набору ігор Високі професійні вимоги до аналітика	Складність, висока ціна створення спеціалізованих ігор
Вимоги до експерта	Співбесідник або практик, розкутий, з акторською майстерністю	Співбесідник або практик, розкутий, з акторською майстерністю	Практик з досвідом роботи з комп'ютером
Вимоги до аналітика	Співбесідник, режисер зі вмінням складати сценарії, з акторською майстерністю	Співбесідник зі вмінням конференсу, режисер зі вмінням складати сценарії, з акторською майстерністю	Мислитель, контактує з програмістом
Характеристики предметної області	Середньо і слабо структурована, слабо документована	Середньо і слабо структурована, слабо документована	Середньо і слабо структурована, слабо документована

6.2 Текстологічні методи

Текстологічні методи поєднують методи отримання знань, засновані на вивченні спеціальних текстів з підручників, монографій, статей, методик та інших носіїв спеціальних знань. Вивчення текстів має жорстку прагматичну спрямованість отримання професійних знань.

Завдання отримання знань з текстів – завдання розуміння та виділення сенсу тексту (рис. 6.4).

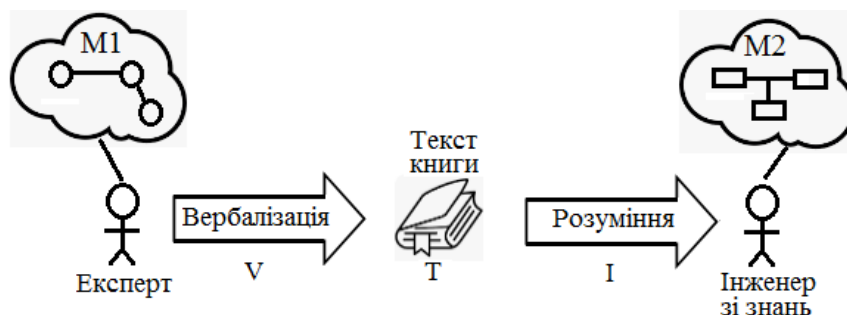


Рисунок 6.4 – Схема вилучення знань зі спеціальних текстів

Можна виділити такі смислові структури:

$M1$ - сенс, який намагався закласти автор, це його модель світу;

$M2$ – сенс, який осягає читач (інженер зі знань) у процесі інтерпретації I ;

T – словесне формулювання $M1$, тобто, результат вербалізації V .

Збіг знань, що утворюють $M1$ та $M2$, є принципово неможливим. Два інженери знання витягнуть з одного тексту T різні моделі $M2^i$.

Виникає завдання досягнення максимальної адекватності $M1$ та $M2$.

Будь-який науковий текст ґрунтується на матеріалах первинних спостережень α та системі наукових понять β . Крім того, в тексті є суб'єктивні погляди автора γ , результат його особистого досвіду та «загальні місця» δ , запозичення з інших джерел θ . Усі названі елементи тексту висловлені у мовному середовищі L (рис. 6.5):

$$T = (\alpha, \beta, \gamma, \delta, \theta)_L$$

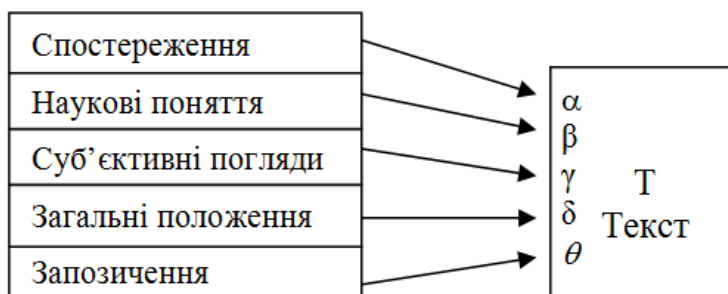


Рисунок 6.5 – Компоненти наукового тексту

До моделі $M1$ входять компоненти наукового тексту β , суб'єктивні погляди автора γ , частина первинних спостережень α .

При вилученні знань аналітику доводиться виділяти у тексті його компоненти, значимі для реалізації фрагментів БЗ. Крім того, будь-який текст набуває сенсу лише в контексті – оточенні, в яке «занурений» текст. Мікроконтекст - найближче оточення тексту (речення набуває сенсу наприкінці абзацу, абзац – наприкінці розділу тощо). Макроконтекст - вся система знань, пов'язана з предметною областю, тобто. знання про особливості та властивості,

явно не вказані в тексті. Будь-яке знання знаходить сенс у контексті деякого метазнання.

Центральним аспектом процедури вилучення знань є розуміння тексту.

Розуміння – це відтворення семантичної структури $M1$ у процесі формування моделі $M2$, тобто, перший крок у структуруванні знань.

Основні моменти розуміння тексту:

- 1) висування попередньої гіпотези про сенс всього тексту - передбачення;
- 2) визначення значень незрозумілих слів – спеціальної термінології;
- 3) виникнення загальної гіпотези про зміст тексту – знання;
- 4) уточнення значення термінів та інтерпретація окремих фрагментів тексту відповідно до загальної гіпотези – від цілого до частин;
- 5) формування деякої смислової структури тексту за рахунок встановлення внутрішніх зв'язків між окремими важливими (ключовими) словами та фрагментами, а також за рахунок утворення абстрактних понять, що узагальнюють конкретні фрагменти знань;
- 6) коригування загальної гіпотези щодо фрагментів знань, які містяться у тексті – від частин до цілого;
- 7) прийняття основної гіпотези – формування $M2$, зв'язування смислових одиниць у єдину семантичну структуру.

Центральними моментами процесу розуміння тексту I є кроки 5 та 7.

При виявленні внутрішніх зв'язків між окремими елементами тексту та поняттями виділяють експліцитні (явні) зв'язки, що виражаються у зовнішньому дробленні тексту, та імпліцитні (приховані) зв'язки. Експліцитні зв'язки поділяють текст на параграфи з допомогою перерахування компонентів, вступних слів тощо. Імпліцитні зв'язки викликають основну проблему при розумінні тексту.

Семантична структура тексту утворюється у свідомості суб'єкта, що пізнає, за допомогою знань про мову, знань про світ, загальних (фонових) знань у предметній області. Для розуміння наукового тексту потрібна певна підготовка.

Підготовкою до прочитання спеціальних текстів є вибір разом із експертом базового списку літератури, який занурить аналітика у предметну область.

На процес розуміння I та формування моделі $M2$ впливають (рис. 6.6):

- вилучення компонентів $(\alpha, \beta, \gamma, \theta)'$ з тексту T ;
- попередні знання аналітика про предметну область ω ;
- загальнонаукова ерудиція аналітика ε ;
- особистий досвід аналітика φ .

$$M2 = [(\alpha, \beta, \gamma, \theta)', \omega, \varepsilon, \varphi]$$

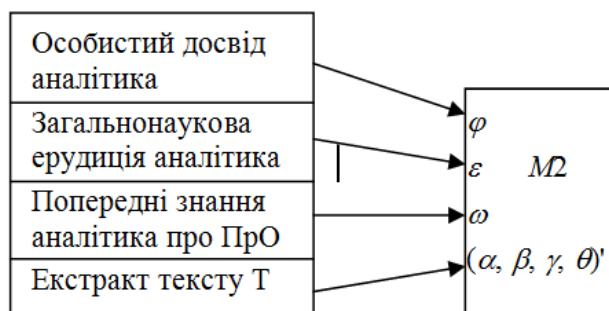


Рисунок 6.6 – Компоненти формування сенсу тексту

Процес I – складний неформальний процес, на який впливають індивідуальні особливості – когнітивний стиль пізнання, інтелектуальні характеристики та ін.

Процедура розбиття тексту на смислові групи та подальше формування смислового фрагмента є основою будь-якого індивідуального процесу розуміння. Стиснення тексту як набору ключових слів може бути методологічною основою для проведення текстологічних процедур вилучення знань.

Ключовим словом може бути будь-яка частина мови чи словосполучення. Набір ключових слів (НКС) – набір опорних точок, якими текст кодується у пам'яті і усвідомлюється при декодуванні, це семантичне ядро тексту.

Текстологічні методи можна поділити на

- аналіз спеціальної літератури;
- аналіз підручників;
- аналіз методик.

Перелічені методи відрізняються за рівнем концентрованості спеціальних знань та за співвідношенням спеціальних та фонових знань.

Найпростішим є аналіз підручників, у яких логіка викладу відповідає логіці предмета і тому макроструктура тексту значиміша, ніж структура тексту спеціальної статті. У методиках виклад є стислим, немає коментарів, тобто, фонових знань, що полегшують розуміння для нефахівців. Для практичної роботи доцільно поєднувати всі ці методи.

Алгоритм отримання знань з тексту.

1. Складання базового списку літератури для ознайомлення з предметною областю та вивчення за списком.
2. Вибір тексту для отримання знань.
3. Перше знайомство з текстом (швидке прочитання). Значення незнайомих слів шукати у довідниках чи запитувати у фахівців.
4. Формування першої гіпотези про макроструктуру тексту.
5. Уважне прочитання тексту з виписуванням ключових слів і виразів, тобто, виділення смислових віх (компресія тексту).

6. Визначення зв'язків між ключовими словами, розробка макроструктури тексту у вигляді графа чи реферату.

7. Формування поля знань з урахуванням макроструктури тексту.

6.3 Методи структурування

Найпростіший алгоритм структурування знань наведено нижче.

1. Визначення вхідних $\{X\}$ та вихідних $\{Y\}$ даних. Структура вхідних та вихідних даних істотно впливає на зміст знань у майбутній системі.

2. Складання словника термінів та набору ключових слів N , що позначають поняття, явища, процеси, предмети, дії, ознаки. Словник складається з урахуванням аналізу всіх протоколів сеансів здобуття знань.

3. Виявлення у словнику об'єктів і понять $\{A\}$, значимих на вирішення проблем предметної області, та їхніх ознак.

4. Виявлення зв'язків між поняттями. Будується мережа асоціацій, у якій присутні непомічені зв'язки.

5. Виявлення метапонять та деталізація понять, структурування понять.

6. Побудова піраміди знань, що містить рівні деталізації понять. Нижній рівень містить детальні описи понять, верхній узагальнення (абстракції) понять. Кількість рівнів залежить від особливостей предметної області, професіоналізму аналітика та експертів.

7. Визначення відношень між поняттями $\{RA\}$ для кожного рівня піраміди та між рівнями. Найменування зв'язків та відносин.

8. Визначення стратегій прийняття рішень $\{Si\}$, ланцюжків міркувань. Стратегії роблять систему динамічною, а знання активними.

6.3.1 Спеціальні методи структурування знань

На практиці при застосуванні найпростішого алгоритму можуть виникнути помилки на стадії отримання знань, тоді доцільно застосовувати спеціальні методи структурування знань на різних кроках найпростішого алгоритму.

Методи виявлення об'єктів, понять та їхніх атрибутів

Поняття чи концепт – це узагальнення предметів деякого класу за їхніми специфічними ознаками. Наприклад, поняття "автомобіль" поєднує множину об'єктів, але всі вони мають колеса, двигун та інші деталі, що дозволяють їм перевозити вантажі та людей. Існує ряд методів виявлення як понять предметної області, так і їхніх ознак під час вилучення знань.

На етапі виявлення об'єктів, понять та їхніх атрибутів визначаються також інтенціонали і екстенціонали предметної області. Іntenціонал визначає поняття через взаємозв'язок значимих ознак, а екстенціонал – через перерахування конкретних екземплярів об'єкта. Виділення реальних об'єктів A

засноване на спостереженнях та лінгвістичних здібностях аналітика. Визначення метапонять вимагає інтелектуального вміння виконувати узагальнення і класифікацію.

Методи виявлення понять можна розділити на:

- традиційні, засновані на математичному апараті розпізнавання образів та класифікації;
- нетрадиційні, засновані на методології інженерії знань.

Дані концептуалізації за наслідками експериментів можна подати у вигляді таблиці (табл. 6.5).

Таблиця 6.5 - Дані концептуалізації

Категорії	Процент від загальної кількості термінів	Процент від загальної кількості термінів, отриманих за методом			
		Перелік понять	Інтерв'ювання	Список операцій	Складання змісту
Пояснення	6,0	5,5	7,2	7,0	4,9
Загальні правила	22,0	43,6	18,9	36,8	4,9
Режимні правила	9,0	9,8	8,4	11,6	6,6
Поняття	42,0	18,4	38,9	8,5	77,7
Процедури	9,0	5,1	9,5	25,6	1,1
Факти	9,0	15,1	12,5	8,9	1,2
Іні поняття	3,0	2,6	4,6	1,6	3,5

Методи інтерв'ювання та складання змісту підручника для представлення знань предметної галузі виявилися найрезультативнішими. Але найбільше правил було породжене у способі списку дій. Таким чином, немає «найкращого» методу, є методи, які підходять для ситуацій та типів знань.

У розглянутих методах число правил продукцій типу «якщо» склало невеликий відсоток у всіх представлених методах.

Методи виявлення зв'язків між об'єктами

Концепти входять до структури понять через відношення. Процес виявлення зв'язків між поняттями у розробці БЗ є досить складним. У пам'яті людини знання являють собою певні зв'язні структури, а не окремі фрагменти. Проте, при розробці БЗ в існуючих моделях основою є поняття, а зв'язки використовуються спрощені, переважно причинно-наслідкові.

В останніх роботах з ІІІ все більше уваги приділяється взаємозв'язкам структур знань. Запроваджується поняття сценарію як структури представлення знань. Сценарій будується на основі КОП (концептуальної організації пам'яті) та мета-КОП (узагальнюючих структур). Сценарії поділяються на фрагменти чи сцени. Зв'язки між фрагментами – часові та просторові, а всередині фрагмента – різні: ситуативні, асоціативні, функціональні та ін.

Методи виявлення зв'язків можна розділити на групи:

- формальні;
- неформальні, засновані на додатковій роботі з експертом.

Неформальні методи виявлення знань винаходить інженер зі знань для отримання від експерта відомостей про явні та неявні зв'язки між поняттями.

Найбільш поширеним методом формування понять є метод «сортування карток» за групами.

При застосуванні методу «замкнених кривих» експерта просять обвести замкнутою кривою пов'язані поняття. У цьому випадку використовуються елементи когнітивної графіки.

Після визначення зв'язку між поняттями всі поняття розпадаються на групи, що є метاپоняттями. Метапоняттям присвоюються імена на наступній стадії процесу структурування.

Методи виділення метاپонять і деталізація понять (піраміда знань)

Суть процесу виділення метاپонять є інтерпретацію груп понять, отриманих на попередній стадії. Зворотним до цього є процес деталізації понять (рис. 6.8).

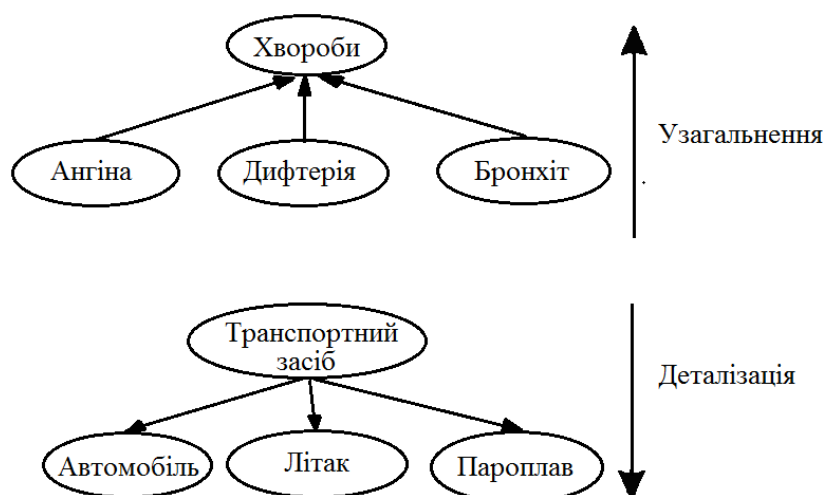


Рисунок 6.8 – Узагальнення та деталізація понять.

Обидва процеси не піддаються формалізації та вимагають від експерта високої кваліфікації та лінгвістичних здібностей для присвоєння імен групам понять чи окремим ознакам. У реальних предметних областях узагальнення та деталізація є дуже складними завданнями. Цей процес є трудомістким та витратним за часом.

При побудові піраміди знань використовують наочний матеріал – малюнки, схеми, кубики тощо. Рівні піраміди виникають у свідомості інженера зі знань як деякі образи. Побудови піраміди знань може ґрунтуватись і на природній ієрархії предметної області, наприклад, з організаційною структурою підприємства або з рівнем компетентності спеціалістів (рис. 6.9).

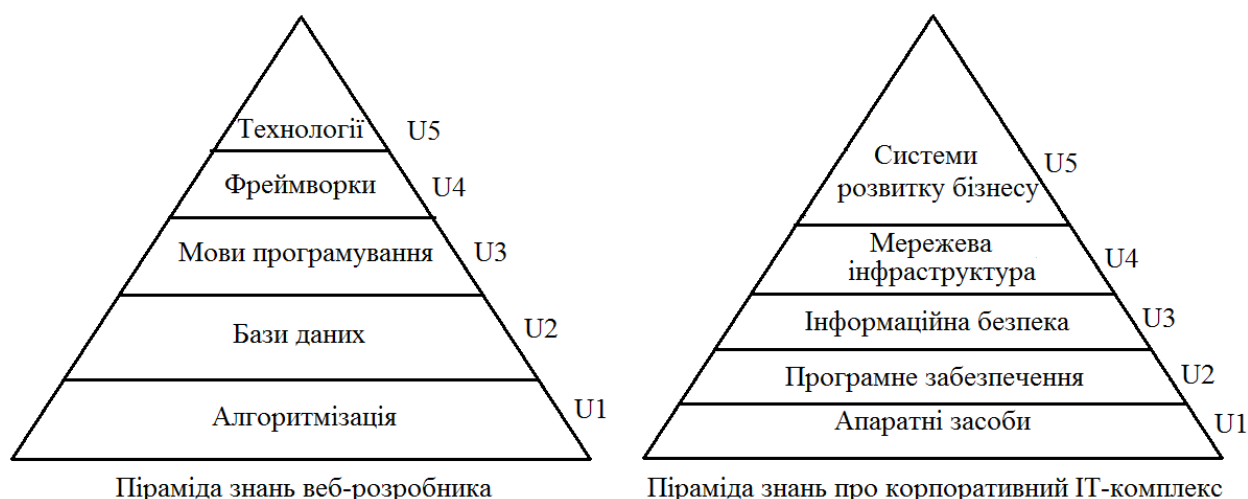


Рисунок 6.9 – Приклади пірамід знань

Методи визначення відношень

На стадії 4 (рис. 6.7) було виявлено зв'язок між поняттями. Ці зв'язки були використані на стадіях 5, 6 для отримання знань піраміди. На стадії 7 слід надати імена зв'язкам, тобто, перетворити їх на відношення.

Між поняттями може існувати понад 200 базових видів відношень, серед яких є універсальні, але можна вводити і нові відношення, притаманні конкретній предметній області.

Використання можливостей когнітивної графіки дозволяє візуалізувати у вигляді схем найпростіші поняття та відношення та будувати схеми вирішення проблем.

В даний час проблемі структурування знань приділено ще мало уваги в наукових розробках та публікаціях, і інженери зі знань виконують концептуалізацію шляхом проб та помилок.

6.4 Методи експертних оцінок

Досвід, інтуїція, почуття перспективи у поєднанні з інформацією допомагають фахівцям точніше вибирати найбільш важливі цілі та напрямки розвитку. Для знаходження найкращих варіантів рішення складних науково-технічних та соціально-економічних задач залучаються експерти, які на основі свого досвіду та знань та експертних методів допомагають знайти найкращі рішення.

Експертні методи безперервно розвиваються та вдосконалюються. Сучасні методи мають широкий спектр сфер застосування і спираються на математичний апарат і використання математичних методів та електронно-обчислювальної техніки.

6.4.1 Експертне оцінювання

Експертне оцінювання – процедура отримання оцінки проблеми на основі думки фахівців (експертів) з метою подальшого ухвалення рішення (вибору).

Експерти мають знання і здатні висловити аргументовану думку щодо явища, що вивчається.

Методи експертних оцінок - це методи організації роботи зі спеціалістами-експертами та опрацювання думок експертів.

Сутність методів експертних оцінок полягає в тому, що в основу прогнозу закладається думка спеціаліста або колективу фахівців, яка базується на професійному, науковому та практичному досвіді. Розрізняють індивідуальні та колективні експертні оцінки.

Приклади застосування експертних оцінок:

- 1) вибір одного варіанта технічного пристрою з кількох зразків для запуску в серію;
- 2) вибір групи розробників програмного забезпечення з множини претендентів;
- 3) вибір науково-дослідного проекту для фінансування;
- 4) вибір одержувачів кредитів із заявлених бажаючих;
- 5) вибір інвестиційних проектів для реалізації серед поданих на розгляд і т.д.

6.4.2 Етапи експертного оцінювання

Експертне оцінювання виконується за рядом етапів:

- 1) формулювання мети та завдань експертної оцінки;
- 2) вибір форми та визначення бюджету експертного дослідження;
- 3) підготовка інформаційних матеріалів, бланків анкет, модератора процедури експертного оцінювання, вибір методу експертних оцінок, формування правил та методів обробки думок експертів;
- 4) підбір експертів;
- 5) проведення експертної оцінки;
- 6) статистичний аналіз результатів.

Формулювання мети та завдань експертної оцінки.

Експертне оцінювання передбачає використання колективного розуму, що має великі можливості в порівнянні з можливостями окремої людини. Експертний підхід дозволяє вирішувати задачі, що не піддаються вирішенню звичайним аналітичним способом.

Перед початком експертного дослідження необхідно чітко визначити його мету (проблему) та сформулювати відповідне задачу для експертів. При цьому рекомендується дотримуватись таких правил:

- чітке визначення умов, часу, зовнішніх та внутрішніх обмежень проблеми; можливість давати чіткі відповіді на запитання;
- питання краще формулювати для отримання якісного твердження, ніж чисельної оцінки; для чисельних оцінок не рекомендується ставити понад п'ять градацій;
- експерти оцінюють можливі варіанти, і не слід очікувати від них побудови закінченого плану дій, розгорнутого опису можливих рішень.

Вибір форми та визначення бюджету експертного дослідження.

Існуючі методи експертних оцінок можна класифікувати за ознаками:

- 1) за формою участі експертів – очні/заочні; очні методи дозволяють зосередити увагу експертів на вирішуваній проблемі, що підвищує якість результату, проте заочні методи можуть бути дешевшими;
- 2) за кількістю ітерацій (повторів процедури для підвищення точності) – однокрокові/багатокрокові;
- 3) за розв'язуваними завданнями – генеруючі рішення/оцінюючі варіанти;
- 4) за типом відповіді – ідейні/ранжируючі, що оцінюють об'єкт у відносній чи абсолютній (чисельній) шкалі;
- 5) за способом обробки думок експертів – безпосередні/аналітичні;
- 6) за кількістю експертів, що залучаються – без обмеження/обмежені; зазвичай використовують 5-12 осіб експертів.

Найбільш відомі методи експертних оцінок є метод Дельфі, мозковий штурм та метод аналізу ієрархій. Кожному методу відповідають свої терміни проведення та кількість експертів.

Після вибору методу експертного оцінювання можна визначити витрати на процедуру оцінювання, які включають оплату експертів, оренду приміщення, придбання канцтоварів, оплату фахівця з проведення експертизи та аналізу її результатів.

Підготовка інформаційних матеріалів, бланків анкет, модератора процедури експертного оцінювання.

Експерти перед винесенням судження мають різнобічно розглянути подану проблему. Для проведення цієї процедури необхідно підготувати інформаційні матеріали з описом проблеми, статистичні дані, довідкові матеріали, бланки анкет, інвентар.

Слід уникати таких помилок: згадувати розробників матеріалів, виділяти той чи інший варіант рішення, висловлювати ставлення керівництва до очікуваних результатів. Дані мають бути різнобічними та нейтральними.

Саму процедуру проводить незалежний модератор процедури, який контролює дотримання регламенту, роздає матеріали та анкети, але сам не висловлює свою думку.

Підбір експертів.

Проблема підбору експертів є однією з найскладніших у теорії та практиці експертних досліджень. Як експертів необхідно використовувати тих

людей, чий міркування найбільше допоможуть прийняттю адекватного рішення. Але проблемою є виділити, знайти, підібрати таких людей. Наразі немає методів підбору експертів, які, гарантовано забезпечать успіх експертизи.

У проблемі підбору експертів можна виділити дві складові - складання списку можливих експертів та вибір з них експертної комісії відповідно до компетентності кандидатів.

Експерти повинні мати досвід в областях, що відповідають вирішуваним завданням. При підборі експертів слід враховувати момент особистої зацікавленості, який може стати суттєвою перешкодою для одержання об'єктивного судження.

Проведення експертної оцінки.

Проведення експертизи залежить від методу, що використовується. Загальні рекомендації:

- перешкоджати тиску авторитетів (експерт часто боїться суперечити думці більшості експертів або найбільш шанованого фахівця);
- встановити регламент та дотримуватись його; збільшення часу прийняття рішення не підвищує точність відповіді.

Статистичний аналіз результатів.

Після отримання результатів від експертів необхідно провести їхню оцінку. Це дозволяє:

- 1) оцінити узгодженість думок експертів; за відсутності значної узгодженості експертів необхідно виявити причини неузгодженості (наявність груп) та визнати відсутність узгодженої думки (нікчемні результати);
- 2) оцінити помилку дослідження;
- 3) побудувати модель властивостей об'єкта з урахуванням відповідей експертів (для аналітичної експертизи); результати експертного оцінювання оформлюються як звіт. У звіті вказується мета дослідження, склад експертів, отримана оцінка та статистичний аналіз результатів.

6.4.3 Характеристики методів експертних оцінок

Методи експертних оцінок можна розділити на дві групи:

- методи колективної роботи експертної групи;
- методи одержання індивідуальної думки членів експертної групи.

Методи групової роботи експертів — методи, засновані на проведенні роботи групи експертів з подальшим аналізом і обробкою отриманої від них інформації.

Метою цих методів є отримання групової експертної оцінки для приймання остаточних рішень.

Підставою для вибору є необхідність ухвалення складних рішень в ситуації невизначеності або складання науково обґрунтованого прогнозу, що вимагає участі групи незалежних і компетентних фахівців у вузькій області або

багатьох галузях знань (наприклад, знань однорідною групи товарів або всіх продовольчих товарів).

Методи колективної роботи експертної групи передбачають отримання спільної думки під час спільного обговорення проблеми, що вирішується. Іноді ці методи називають методами прямого здобуття колективної думки. Методи колективної роботи включають методи «мозкового штурму», «сценаріїв», «ділових ігор», «нарад» та «суду». Кожен з цих видів експертного оцінювання має свої переваги і недоліки, що визначають раціональну сферу застосування. У багатьох випадках найбільший ефект дає комплексне застосування кількох видів експертизи.

Основні переваги групової експертної оцінки полягають в можливості різностороннього аналізу кількісних і якісних аспектів проблем визначення і/або прогнозування окремих характеристик товарів або їх сукупності. Взаємодія між експертами дозволяє значно збільшити обсяг сумарної інформації, якою володіє група експертів, в порівнянні з інформацією будь-якого члена групи. До того ж кількість чинників, врахованих при груповій оцінці і що впливають на результативність ухвалюваного рішення, більше, ніж сума чинників, врахованих одним експертом. При груповій оцінці менша помилка прийняття основних рішень і показників, що не мають суттєвого значення для вирішення проблеми. Тому важливою перевагою групової оцінки є можливість отримання узагальненого результату.

До недоліків групових оцінок належать:

труднощі в отриманні надійної і узгодженої оцінки;

- отримання неоднакових відповідей на одне і те ж питання з великою різницею думок через різний ступінь компетентності експертів;
- отримання однозначних відповідей не гарантує їх обґрунтованість і достовірності, причому при проведенні експертизи це неможливо перевірити;
- більша кількість невірної інформації в групі експертів, ніж у окремого експерта, може призвести до значних помилок в кінцевих результатах;
- можливість конфронтації, коли окремі експерти через невпевненість або інші причини можуть погоджуватися з думкою більшості.

Незважаючи на вказані недоліки, при дотриманні певних вимог групова оцінка надійніша, ніж індивідуальна. До таких вимог належать: обґрунтований розподіл оцінок; групова надійність; підготовка експертизи.

Методи отримання індивідуальної думки членів експертної групи ґрунтуються на попередньому отриманні інформації від експертів, опитуваних незалежно один від одного, з подальшою обробкою отриманих даних. До цих методів можна віднести методи анкетного опитування, інтерв'ю та методи «Дельфі». Основні переваги методу індивідуального експертного оцінювання полягають у їх оперативності, можливості повною мірою використовувати індивідуальні здібності експерта, відсутність тиску з боку авторитетів та низькі витрати на експертизу. Головним їх недоліком є високий рівень суб'єктивності одержуваних оцінок через обмеженість знань одного експерта.

6.4.4 Обробка експертних оцінок

Нехай m експертів провели оцінку n об'єктів за l показниками. Результати оцінки представлені у вигляді величин x_{ij}^h , де j – номер експерта, i – номер об'єкта, h – номер показника (ознаки) порівняння. Якщо оцінку об'єктів здійснено методом ранжирування, то величини x_{ij}^h являють собою ранги. Якщо оцінка об'єктів виконана методом безпосередньої оцінки або методом послідовного порівняння, то величини x_{ij}^h є числами з деякого відрізка числової осі, або бали. Обробка результатів оцінки залежить від розглянутих методів виміру.

На базі оцінок експертів отримують узагальнену інформацію про досліджуваний об'єкт (явище) і формують рішення, яке є метою експертизи. При обробці індивідуальних оцінок експертів використовують різноманітні кількісні та якісні методи. Вибір того чи іншого методу залежить від складності розв'язуваної проблеми, форми представлення думки експертів, цілей експертизи.

Залежно від цілей експертного оцінювання та обраного методу вимірювання при обробці результатів опитування виникають такі основні завдання:

- 1) побудова узагальненої оцінки об'єктів з урахуванням індивідуальних оцінок експертів;
- 2) побудова узагальненої оцінки з урахуванням парного порівняння об'єктів кожним експертом;
- 3) визначення відносних ваг об'єктів;
- 4) визначення узгодженості думок експертів;
- 5) визначення залежностей між ранжируваннями;
- 6) оцінка надійності результатів обробки.

Завдання побудови узагальненої оцінки об'єктів за індивідуальними оцінками експертів виникає за групового експертного оцінювання. Розв'язання цього завдання залежить від використаного експертами методу виміру.

Визначення узгодженості думок експертів здійснюється шляхом обчислення числового заходу, що характеризує міру близькості індивідуальних думок. Аналіз значення міри узгодженості сприяє виробленню правильного судження про загальний рівень знань з проблеми і виявлення угруповань думок експертів. Якісний аналіз причин угруповання думок дозволяє встановити існування різних поглядів, концепцій, виявити наукові школи, визначити характер професійної діяльності тощо. Усі ці фактори дають можливість глибше осмислити результати опитування експертів.

Найчастіше під час обробки результатів опитування використовуються методи математичної статистики.

Отримання групової експертної оцінки шляхом підсумовування індивідуальних оцінок з вагами компетентності експертів та важливості показників при вимірі властивостей об'єктів у кардинальних шкалах

ґрунтується на припущенні про виконання аксіом теорії корисності фон Неймана-Моргенштерна як для індивідуальної, так і для групової оцінки та умов нерозрізненості об'єктів у груповому відношенні, якщо вони нерозрізнені у всіх індивідуальних оцінках (частковий принцип Парето). У реальних завданнях ці умови, як правило, виконуються, тому отримання групової оцінки об'єктів шляхом підсумовування з вагами індивідуальних оцінок експертів широко застосовується на практиці.

Коефіцієнти компетентності експертів можна обчислити за апостеріорними даними, тобто за результатами оцінки об'єктів. Основною ідеєю цього обчислення є припущення, що компетентність експертів повинна оцінюватися за ступенем узгодженості їх оцінок з груповою оцінкою об'єктів.

Методи групових оцінок дозволяють отримати достовірні результати у разі добре підібраної групи експертів та узгодженості їхніх думок. Якщо це не так, то постає завдання визначення кількісної оцінки ступеня узгодженості експертів. Отримання кількісного заходу дозволяє більш обґрунтовано інтерпретувати причини розбіжності думок.

Для оцінки міри узгодженості думок групи експертів використовують, зокрема, дисперсійний та ентропійний коефіцієнти конкордації. Крім цього, під час обробки результатів ранжирування можуть виникати завдання:

- визначення залежності між ранжируваннями двох експертів;
- зв'язки між досягненням двох різних цілей при вирішенні однієї й тієї самої сукупності проблем;
- взаємозв'язки між ознаками (об'єктами).

У цих випадках мірою взаємозв'язку може бути коефіцієнт рангової кореляції. Характеристикою взаємозв'язку множини ранжирувань буде матриця коефіцієнтів рангової кореляції. Відомі коефіцієнти рангової кореляції Спірмена та Кендала.

Контрольні запитання

1. Назвіть методи отримання знань.
2. Охарактеризуйте комунікативні методи отримання знань.
3. Охарактеризуйте пасивні методи отримання знань.
4. Наведіть порівняльні характеристики пасивних методів отримання.
5. Охарактеризуйте активні індивідуальні методи отримання знань.
6. Порівняйте активні індивідуальні методи отримання знань.
7. Охарактеризуйте активні групові методи отримання знань.
8. Порівняйте методи круглого столу та мозкового штурму.
9. Наведіть класифікацію експертних ігор.
10. Порівняйте характеристики експертних ігор.
11. Охарактеризуйте текстологічні методи отримання знань.
12. Назвіть компоненти формування змісту тексту.
13. Поясніть найпростіший алгоритм структурування знань.
14. Назвіть та охарактеризуйте методи структурування знань.

15. Назвіть та охарактеризуйте завдання обробки експертних оцінок.
16. Поясніть отримання групової експертної оцінки об'єктів під час безпосереднього оцінювання.
17. Як здійснюється обробка результатів парних порівнянь?
18. Поясніть процес визначення узагальнених ранжирувань.

7 НЕЙРОМЕРЕЖЕВА ТЕХНОЛОГІЯ У ЕКСПЕРТНИХ СИСТЕМАХ

Більшість експертних систем було створено у рамках символного підходу, у таких ЕС базу знань побудовано на моделях: продукційній, семантичних мереж, фреймів, логічній. На противагу символному розвинувся конекціоністський підхід до побудови систем штучного інтелекту. В основі конекціоністського підходу лежить спроба безпосереднього моделювання розумової діяльності людського мозку. Мозок людини складається з величезної кількості нервових клітин (нейронів), що взаємодіють між собою і складають мережу нейронів.

Штучні нейромережі – це перспективна обчислювальна технологія, яка спирається на нейрофізіологію, математику, статистику, фізику, комп'ютерні науки.

Основна особливість нейромереж - здатність навчатися (з учителем чи без учителя) і адаптивність.

Нейромережі знаходять застосування у вирішенні таких завдань:

- розпізнавання візуальних образів;
- доповнення образів;
- розпізнавання мовлення;
- асоціативний пошук;
- класифікація;
- оптимізація;
- діагностика;
- аналіз часових рядів;
- прогнозування;
- моделювання нелінійних процесів;
- робота з зашумленими даними;
- обробка сигналів;
- сегментація даних та ін.

7.1 Біологічна нейронна мережа

Нервова система і мозок людини складаються з нейронів, з'єднаних між собою нервовими волокнами. Нервові волокна здатні передавати електричні імпульси між нейронами. Всі процеси передачі подразнень від нашої шкіри, вух і очей до мозку, процеси мислення і управління діями - все це реалізовано в живому організмі як передача електричних імпульсів між нейронами.

Загальне число нейронів у центральній нервовій системі людини досягає $10^{10} - 10^{11}$, при цьому кожна нервова клітина пов'язана в середньому з $10^3 - 10^4$ інших нейронів. Встановлено, що в головному мозку сукупність нейронів в обсязі масштабу 1 мм^3 формує відносно незалежну локальну мережу, що несе певну функціональне навантаження.

Біологічний нейрон має наступну будову. Кожен нейрон має відростки нервових волокон двох типів:

- **дендрити**, за якими приймаються імпульси;
- єдиний **аксон**, за яким нейрон може передавати імпульс.

Аксон контактує з дендритами інших нейронів через спеціальні утворення – **синапси**, які впливають на силу імпульсу (рис. 7.1).

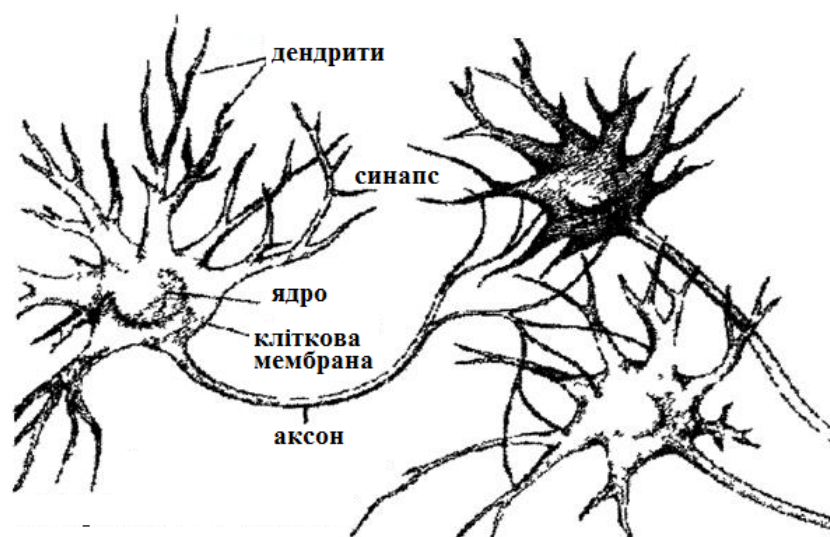


Рисунок 7.1 – Біологічний нейрон

При проходженні синапсу сила імпульсу змінюється згідно з **вагою синапсу**. Імпульси, що надійшли до нейрону одночасно за кількома дендритами, додаються. Якщо сумарний імпульс перевищує деякий поріг, нейрон збуджується, формує власний імпульс і передає його далі по аксону. Ваги синапсів можуть змінюватися з часом, а значить, змінюється і поведінка відповідного нейрона.

Модель біологічного нейрона – математичний опис властивостей нервових клітин, або нейронів. Модель створена для того, щоб якомога точно передати і передбачити поведінку біологічних процесів. Це відрізняє дані моделі від штучного нейрона, який застосовується в задачах де ефективність обчислення має більш важливе значення.

Математична модель нейрона (рис. 7.2) має кілька входів (дендритів).

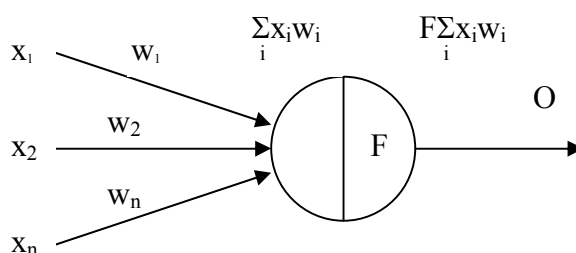


Рисунок 7.2 – Математична модель нейрона

Синапси дендритів мають ваги w_1 , w_2 , w_n . Якщо до синапсу надходять імпульси силою x_1 , x_2 , x_n відповідно, тоді після проходження синапсів і дендритів до нейрону надходять імпульси w_1x_1 , w_2x_2 , w_nx_n , де w_i - провідність синапсу. Нейрон перетворює отриманий сумарний імпульс

$$x = w_1x_1 + w_2x_2 + w_nx_n$$

відповідно до деякої передатної функції $F(x)$ (функції активації). Сила вихідного імпульсу дорівнює

$$O = F(x) = F(w_1x_1 + w_2x_2 + w_nx_n).$$

Таким чином, нейрон повністю описується своїми вагами w_i і передатною функцією $F(x)$. Отримавши набір чисел (вектор) x_i як входів, нейрон видає деяке число o на виході.

7.2. Штучна нейронна мережа

Штучна нейронна мережа (ШНМ) – це обчислювальна система, створена за прикладом біологічних нейронних мереж. Такі системи навчаються вирішувати задачі (поступово покращують якість рішень), розглядаючи приклади, без спеціального програмування під конкретну задачу.

ШНМ можна зобразити як набір нейронів, сполучених між собою (рис. 7.3). Як правило, передавальні функції всіх нейронів у нейронній мережі є фіксованими, а ваги є параметрами нейронної мережі і можуть змінюватися. Деякі входи нейронів помічені як зовнішні входи нейронної мережі, а деякі виходи - як зовнішні виходи нейронної мережі.

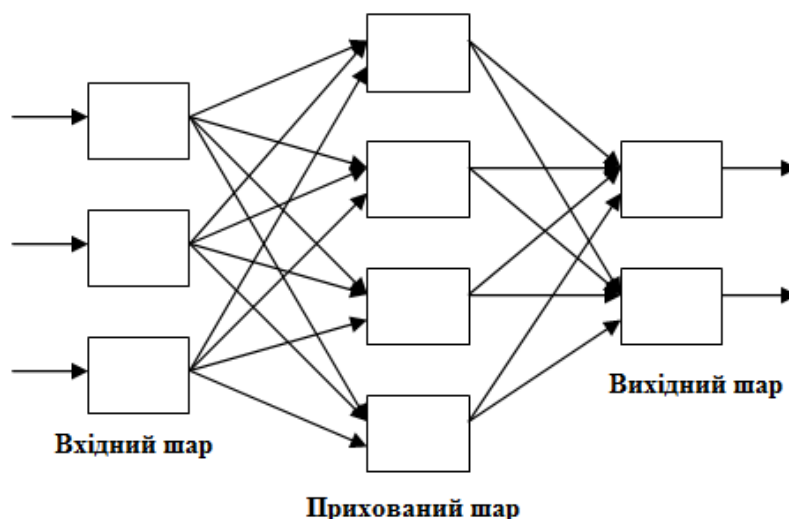


Рисунок 7.3 – Багатошарова нейронна мережа

Вхідний – шар нейронів, який одержує інформацію.

Прихований – кілька шарів, що обробляють інформацію.

Вихідний – шар нейронів, що представляє результати обчислень.

Подаючи будь-які числа на входи нейронної мережі, ми отримуємо певний набір чисел на виходах нейронної мережі. Таким чином, робота нейронної мережі полягає в перетворенні вхідного вектора у вихідний вектор, причому це перетворення задається вагами нейронної мережі.

Практично будь-яке завдання можна звести до задачі, розв'язуваної нейронною мережею.

7.3 Робота з нейронною мережею

Робота з нейронною мережею складається з трьох етапів (рис. 7.4):

1. Вибір типу мережі.
2. Навчання мережі.
3. Застосування мережі.

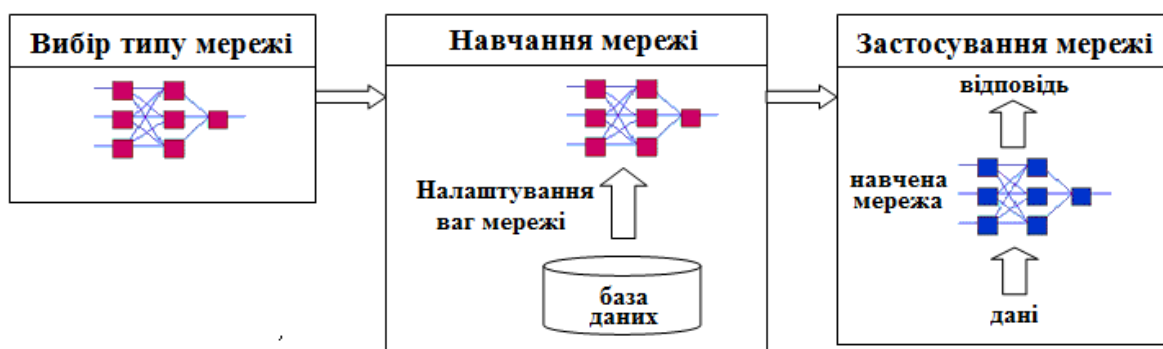


Рисунок 7.4 – Етапи роботи з нейронною мережею

Побудова нейронної мережі

Питання побудови нейронної мережі вирішується в два етапи:

1. Вибір типу (архітектури) нейронної мережі.
2. Підбір ваг (навчання) нейронної мережі.

На першому етапі слід визначити наступне:

- які нейрони ми хочемо використовувати (число входів, передавальні функції);
- яким чином слід з'єднати їх між собою;
- що взяти в якості входів і виходів нейронної мережі.

Існує кілька десятків різних нейромережних архітектур, і ефективність багатьох з них доведено математично. Найбільш популярні й вивчені такі архітектури:

- багатошаровий перцептрон;
- нейронна мережа із загальною регресією;
- нейронні мережі Кохонена та інші.

На другому етапі слід "навчити" обрану нейронну мережу, тобто підібрати такі значення її ваги, щоб вона працювала належним чином.

Навчання нейронної мережі

Для рішення задач треба використовувати навчену нейронну мережу. Навчання нейронної мережі - це пошук найкращого набору вагових коефіцієнтів зв'язків між нейронами для отримання рішення задачі з максимальною точністю (рис. 7.5).

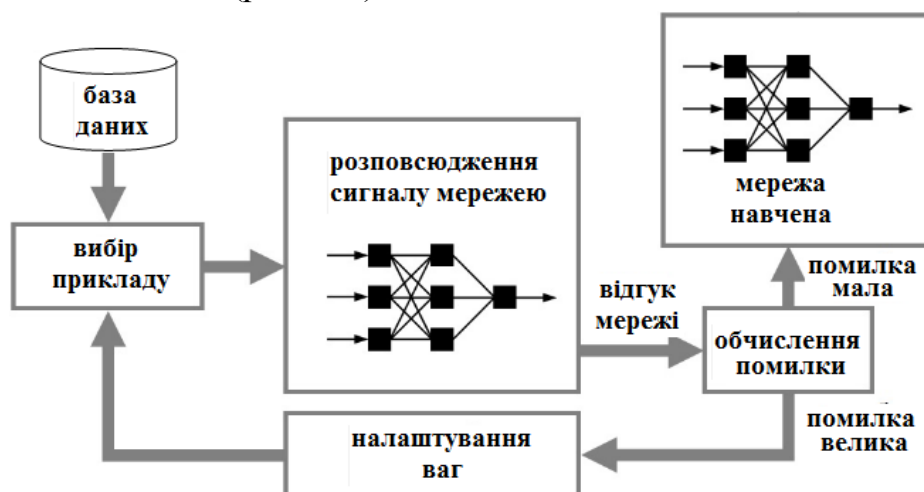


Рисунок 7.5 – Процес навчання нейронної мережі

Для навчання мережі застосовують алгоритми з учителем і без.

Навчання з учителем: нейронній мережі надають деяку вибірку навчальних прикладів. Зразок подається на вхід, після цього відбувається його обробка прихованими шарами і розраховується вихідний сигнал, який порівнюється з відповідним значенням цільового вектора (відомим нам «правильною відповіддю» для кожного з навчальних прикладів). Якщо відповідь мережі не збігається з очікуваним значенням, проводиться корекція ваг мережі за алгоритмом зворотного розповсюдження помилки.

Алгоритм зворотного розповсюдження помилки дозволяє за вектором помилки обчислити необхідні поправки для ваг нейронної мережі.

Після багаторазового пред'явлення прикладів ваги нейронної мережі стабілізуються, причому нейронна мережа дає правильні відповіді на всі (або майже всі) приклади з бази даних. У такому випадку говорять, що нейронна мережа навчена. В процесі навчання величина помилки (сума квадратів помилок з усіх виходів) поступово зменшується. Коли величина помилки досягає нуля або прийняттого малого рівня, тренування зупиняють, а отриману нейронну мережу вважають натренованою і готовою до застосування на нових даних.

При навчанні без вчителя алгоритм готує ваги мережі таким чином, щоб можна було отримати узгоджені вихідні вектори, тобто надання досить близьких векторів буде давати схожі виходи. Одним з таких алгоритмів навчання є правило Хебба, за яким налаштовується перед роботою матриця ваг, наприклад, рекурентної мережі Хопфілда.

Якість навчання нейронної мережі безпосередньо залежить від кількості прикладів в навчальній вибірці, а також від того, наскільки повно ці приклади описують дану задачу. Вважається, що для повноцінної тренування нейронної мережі потрібно хоча б кілька десятків (а краще сотень) прикладів. Навчання нейронних мереж - складний і наукомісткий процес. Алгоритми навчання нейронних мереж мають різні параметри і настройки, для управління якими потрібно розуміння їх впливу.

Навчену нейронну мережу можна застосовувати для вирішення поставлених завдань. Нейронна мережа, навчена коректно, може з великою ймовірністю правильно реагувати на нові, не пред'явлені їй раніше дані.

7.4 Деякі області застосування нейронних мереж

Нейромережева класифікація

Задачі класифікації (типу розпізнавання букв) досить погано алгоритмізуються. В складних практичних завданнях навчена нейронна мережа

виступає як експерт, що володіє великим досвідом і здатний дати відповідь на важке запитання.

Прикладом завдання класифікації служить медична діагностика, де нейронна мережа може враховувати велику кількість числових параметрів (енцефалограма, тиск, вага і т.д.). Звичайно, "думка" нейронної мережі в цьому випадку не може вважатися остаточною.

Класифікація підприємств за ступенем їх перспективності – це звичний спосіб використання нейронних мереж у практиці західних компаній. При цьому нейронна мережа також використовує багато економічних показників, складним чином пов'язаних між собою.

Нейромережевий підхід особливо ефективний у задачах експертної оцінки з тієї причини, що він поєднує в собі здатність комп'ютера до обробки чисел і здатність мозку до узагальнення і розпізнавання. Кажуть, що у хорошого лікаря здатність до розпізнавання у своїй області настільки велика, що він може провести приблизну діагностику вже за зовнішнім виглядом пацієнта. Можна погодитися також, що досвідчений трейдер почуває напрямок руху ринку за видом графіка. Однак у першому випадку всі фактори наочні, тобто характеристики пацієнта миттєво сприймаються мозком як "бліде обличчя", "блиск в очах" і т.д. У другому ж випадку враховується лише один чинник, показаний на графіку - курс за певний період часу. Нейронна мережа дозволяє обробляти величезну кількість чинників (до декількох тисяч), незалежно від їх наочності – це універсальний "хороший лікар", який може поставити свій діагноз у будь-якій області.

Кластеризація за допомогою нейронних мереж і пошук залежностей

Кластеризація – це розбиття набору прикладів на кілька компактних областей (кластерів), причому число кластерів заздалегідь невідомо. Кластеризація дозволяє представити неоднорідні дані в більш наочному вигляді та використовувати далі для дослідження кожного кластера різні методи. Наприклад, таким чином можна швидко виявити фальсифіковані страхові випадки та несумлінні підприємства.

Нейронні мережі широко використовуються для пошуку залежностей в даних і кластеризації.

Наприклад, нейронна мережа на основі методики МГВА (метод групового врахування аргументів) дозволяє на основі навчальної вибірки побудувати залежність одного параметра від інших у вигляді полінома. Така нейронна мережа може не тільки миттєво вивчити таблицю множення, а й

знайти складні приховані залежності в даних (наприклад, фінансових), які не виявляються стандартними статистичними методами.

Застосування нейронних мереж у задачах прогнозування

Здатність нейронної мережі до прогнозування безпосередньо впливає з її здатності до узагальнення і виділення прихованих залежностей між вхідними та вихідними даними. Після навчання мережа здатна передбачити майбутнє значення деякої послідовності на основі декількох попередніх значень і (або) певних існуючих у поточний момент чинників. Прогнозування можливо тільки тоді, коли попередні зміни дійсно в деякій мірі визначають майбутні. Наприклад, прогнозування котирувань акцій на основі котирувань за минулий тиждень може виявитися успішним (а може і не виявитися), тоді як прогнозування результатів завтрашньої лотереї на основі даних за останні 50 років майже напевно не дасть ніяких результатів.

7.5 Проблеми нейромереж

Повільна збіжність процесу навчання. Строго збіжність доведена для диференціальних рівнянь, тобто для нескінченно малих кроків у просторі ваг. Але нескінченно малі кроки означають нескінченно великий час навчання. При кінцевих кроках збіжність не гарантується, але навіть якщо вона має місце, то потрібний для цього час може бути занадто великим, порівнянним з часом життя користувача.

"Пастки", утворювані локальними мінімумами. Детермінований алгоритм навчання не може виявити глобальний мінімум або покинути локальний мінімум. Одним з прийомів, що дозволяють обходити пастки, є розширення розмірності простору ваг за рахунок збільшення числа нейронів другого шару. Деякі нові можливості відкривають стохастичні методи. Але все це досягається ціною додаткових витрат часу навчання.

"Параліч" мережі. Сигмоїдальний характер передатної функції нейрона призводить до того, що якщо в процесі навчання декілька ваг стали занадто великими, то нейрон потрапляє на горизонтальний ділянку функції в область насичення. При цьому, зміни інших ваг, навіть досить великі, практично не позначаються на величині виходу нейрона, а значить, і на величині цільової функції. Це означає, що зв'язок між сусідніми шарами практично розривається, і процес навчання блокується.

Невдалий вибір діапазону вхідних змінних. Якщо x_i - двійкова змінна зі значеннями 0 і 1, то приблизно в половині випадків вона буде мати нульове значення. Оскільки x входить співмножником у вираз для модифікації ваги, то ефект буде той же, що при насиченні: модифікація відповідних ваг припиниться, і навчання буде блоковано. Правильний діапазон для вхідних змінних повинен бути симетричним, наприклад від 1 до -1.

"Перетренування". Зависока точність, отримана на навчальній вибірці, може обернутися нестійкістю результатів на тестовій вибірці. Тут діє загальний закон: чим краще система адаптована до даних конкретних умов, тим менше вона здатна до узагальнення та екстраполяції, тим швидше вона може виявитися непрацездатною при зміні цих умов. А такі зміни від вибірки до вибірки неминучі, особливо якщо вибірки мають невеликі розміри. Розширення обсягу навчальної вибірки дозволяє домогтися більшої стійкості, але за рахунок збільшення часу навчання.

Проблема обсягу пам'яті. Ємність пам'яті нейромережі, її здатність зберігати і відтворювати інформацію є однією з найважливіших характеристик нейромережі. Однак якщо в традиційних-послідовних машинах характеристики пам'яті достатньо зрозумілі і доступні для оцінки, то в нейромережах це набагато складніше.

7.6 Нейромережеві експертні системи

Гібридні інтелектуальні системи — це системи, які поєднують декілька інтелектуальних технологій, щоб створити нові системи, які мають силу кількох систем і покривають слабкі сторони інших.

Комбінація нейронної мережі та експертної системи є нейронною експертною системою. Поєднання нейронної мережі та експертної системи стає новою системою, яка має здатність до навчання та пояснює користувачеві зроблені висновки. Існує кілька підходів, які об'єднують нейронні мережі та символічні правила, які є основою традиційних ЕС.

В даний час відомо багато вдалих прикладів застосування нейромережевого підходу для побудови інтелектуальних інформаційних систем та, зокрема, експертних систем.

Комбіноване використання експертної системи та апарату штучних нейронних мереж забезпечує необхідну гнучкість та самонавчання на основі знань, водночас отримані від експертів знання дозволяють суттєво спростити структуру нейронних мереж, зменшити кількість нейронів та зв'язків у мережі.

Наприклад, медичні нейромережеві експертні системи проявили себе серйозним суперником традиційних експертних систем, складаючи конкуренцію кваліфікованим експертам.

Існують різні інтелектуальні технології, такі як ймовірнісна логіка, нечітка логіка, нейронні мережі і еволюційні обчислення. Кожна технологія має свої переваги і недоліки. У системах зі штучним інтелектом необхідно не тільки вилучити знання з різних джерел, але й комбінувати застосування різних технологій.

У експертних системах, заснованих на правилах, правила продукцій отримують від людей-експертів. Це завдання, яке називається вилученням знань, є складним і дорогим. Крім того, коли правила зберігаються в базі знань, вони не можуть бути змінені самою експертною системою. Більшість експертних систем не можуть вчитися на досвіді або адаптуватися до нового середовища.

Знання в нейронних мережах зберігаються як синаптичні зв'язки між нейронами. Ці знання отримують на етапі навчання, коли навчальний набір даних подається в мережу. Мережа поширює вхідні дані від рівня до рівня у мережі, поки не будуть згенеровані вихідні дані. Якщо вони відрізняються від бажаного результату, помилка обчислюється та поширюється у зворотному напрямку по мережі. На відміну від експертних систем, нейронні мережі навчаються без втручання людини.

В експертних системах знання можна розділити на окремі правила, і користувач може бачити та розуміти частину знань, застосовану системою. У нейронних мережах неможливо вибрати одну синаптичну вагу як окрему частину знання. Тут знання вбудовані у всю мережу. Будь-яка зміна синаптичної ваги може призвести до непередбачуваних результатів.

Експертна система не може навчатися, але може пояснити, як вона приходить до конкретного рішення. Нейронна мережа може навчатися, але діє як чорний ящик. Таким чином, поєднуючи переваги кожної технології, можна створити більш потужну та ефективну експертну систему. Навчання, узагальнення, стійкість і паралельна обробка інформації роблять нейронні мережі перспективним компонентом для побудови нового покоління експертних систем.

Нейронна експертна система може витягувати правила ЯКЩО-ТО з нейронної мережі, які дозволяють їй обґрунтовувати та пояснювати свій висновок. Тільки нейронні мережі можуть виробляти знання як налаштування ваги дендритів у процесі навчання. Але вона не може пояснити рішення користувачеві. Таким чином у такій ЕС мають бути створені правила з обчислення ваг.

На сьогоднішній день відомі кілька шляхів технологій інтеграції експертних систем (ЕС) і нейронних мереж:

- інтеграція в гібридну інформаційну систему (ГІС): використання нейромереж як машини виведення в ЕС або використання ЕС в якості препроцесора ініціалізації нейромережі, вибору топології нейромережі, алгоритму навчання;

– розробка конекціоністських експертних мереж і уніфікованих нейромереж, тобто, спеціальні програми, що імітують навчання та обчислення в нейромережі, в яких кожному нейрону зіставлено символічний еквівалент – поняття природної мови;

- підвищення якості знань при використанні нейромереж;

– інтеграція знань про предметну область у нейромережах, тобто, включення знань про предметну область у нейрообчислення з метою вдосконалення останніх;

- вилучення правил з навчених нейромереж для використання в ЕС або трансляція правил предметної області в нейромережу;

– конвертування нейромереж у дерева рішень чи навпаки.

У технології експертних систем та нейронних мереж використовуються різні моделі обчислень.

Модель обчислень в експертних системах має вигляд:

$$ES = \langle KB, FB, RB, I^e, R^e \rangle,$$

де KB – база знань у формі продукційних правил;

FB – база фактів про систему ES ;

RB – база висновків, яка формується інтерпретатором у ході роботи і містить інформацію про причини змін у базі висновків та коментарі для пояснень, введені експертом;

R^e – системоутворюючі відношення;

I^e – інтерпретатор, представлений четвіркою послідовних процесів, виконуваних циклічно.

У цикл роботи інтерпретаторі входять:

– процес вибору з бази знань підмножини активних правил;

– процес зіставлення зі зразком для активних правил;

– процес вирішення конфліктів правил;

– процес виконання правила, тобто, використання його у міркуваннях.

Модель нейрообчислень може бути представлена таким чином:

$$NN = \langle Ar, X, Y, M^n, Ed, I^{n1}, I^{n2} \rangle$$

де Ar – архітектура (топологія) нейромережі;

X, Y – множини входів та виходів нейромережі відповідно;

M^n – множина моделей штучних нейронів;

Ed – навчальна та тестуюча послідовності;

I^{n1} – інтерпретатор навчання,

I^{n2} – інтерпретатор нейрообчислень.

Таким чином, інтегрована нейро-експертна модель обчислень має такий вигляд:

$$NES = \langle KB, X, Y, Ed, R', I^{n1}, I^{n2} \rangle$$

У нейромережевій експертній системі зберігається база правил ЕС, на основі якої будується нейромережа зі входами X та виходами Y ; нова система отримує новий набір системостворчих відношень R' . Така нейромережева модель експертної системи здатна зберігати досвід системи в цілому у неявному вигляді. Для отримання відповіді необхідно проводити опитування системи і проводити нейрообчислення. Це не завжди є найкращим рішенням для деяких задач. Зберігати досвід інтелектуальної системи явно здатні системи, засновані на прецедентах.

Прецедент включає:

- опис проблемної ситуації, що визначає стан досліджуваного процесу, коли відбувся прецедент;
- спосіб рішення цієї проблеми;
- результат, який описує стан досліджуваного процесу після проблемної ситуації, що відбулася.

Модель обчислень з використанням методів міркувань на основі прецедентів має такий вигляд:

$$PS = \langle KB_1, KB_2, A(p), I^p \rangle$$

де KB_1 – база одиниць прецедентів,
 KB_2 – база одиниць загальних знань про предметну область;
 $A(p)$ – алгоритм визначення схожих прецедентів p .

Інтерпретатор I^p , застосовуючи $A(p)$ і KB_2 , обробляє інформацію, що зберігається в базі прецедентів KB_1 , і являє собою сукупність процесів I^p , до складу яких входять процеси виявлення, адаптації, перегляду, збереження.

Після включення в модель нейромережевої експертної системи обчислення на основі прецедентів то отримана гібридна експерта система виявиться більш функціональною, зберігши свою гнучкість:

$$NES = \langle KB, KB_p, A(p), R'', I^{np}, I^{n2}, I^p \rangle,$$

де KB – база знань, яка зберігає знання у вигляді продукцій і у вигляді прецедентів,
 KB_p – системоутворюючі відношення системи,
 $A(p)$ – алгоритм визначення схожих прецедентів,
 R'' – системоутворюючі відношення нейромережевої експертної системи,
 I^{np} – навчання нейромережі,

I^{n2} – нейромережевий пошук рішень,
 I^p – прецедентний пошук рішень.

Пошук рішень у нейромережевій експертній системі з обчисленнями на основі прецедентів розбивається на нейромережевий I^{n2} та прецедентний I^p за алгоритмом визначення схожих прецедентів $A(p)$. Навчання нейромережі проводиться шляхом пред'явлення прецедентів як тестових прикладів.

Нейромережева експертна система (рис. 7.6) складається з бази знань прецедентів та продукцій (БЗП), блоку отримання знань (БОЗ), механізму пошуку прецедента (МПП), нейро-експертна машини отримання рішення (НЕМ), блоків адаптації даних (АД)) та пояснення рішення (БП).

Система працює наступним чином. Користувач вводить опис прецеденту. Спочатку активізується алгоритм пошуку прецеденту у базі знань прецедентів. Кожен факт-ознака у прецеденті має свою вагу, що враховується як його відносна цінність. Опис прецедента зіставляється з прецедентами з БЗП і обчислюється ступінь близькості прецеденту за всіма ознаками за метрикою Хеммінга як відношення кількості ознак, що співпали зі зразком, до загальної кількості ознак.

Якщо подібна проблема раніше вирішувалася і знайдено спів падіння з прецедентом-зразком, користувачеві виводиться рішення з рекомендацією.

Якщо подібна проблема раніше не вирішувалася і прецедент відсутній у базі знань, то завдання передається на вхід нейро-експертного механізму. Нейро-експертний механізм є нейромережею правил. Усі правила виконуються одночасно. Вузлами мережі є нейрони - окремі факти, що витягуються з прецедентів, що входять до умов та наслідків правил експертної системи, а зв'язки між вузлами мережі реалізують правила. У такий спосіб організується слабопов'язана багат шарова нейромережа фактів та правил.

Умови правил мають обмеження ймовірності факту, тобто, мінімально та максимально необхідні ймовірності для того, щоб умова виконувалася.

Вихідний шар нейромережі складається з понять основи знань, тобто, відповідей ЕС. На етапі навчання – це значення висновків системи, вилучені з прецедентів навчальної вибірки. Фактам висновків автоматично призначається ймовірність з урахуванням вихідних даних як усереднене значення сукупності ймовірностей вхідних фактів.

Вхідний вектор нейронної мережі формується на початковому етапі роботи з ЕС з урахуванням попереднього опитування. Кожен нейрон вхідного шару може набувати значення, наприклад, від 0 до 100, що визначає вагу кожного факту.

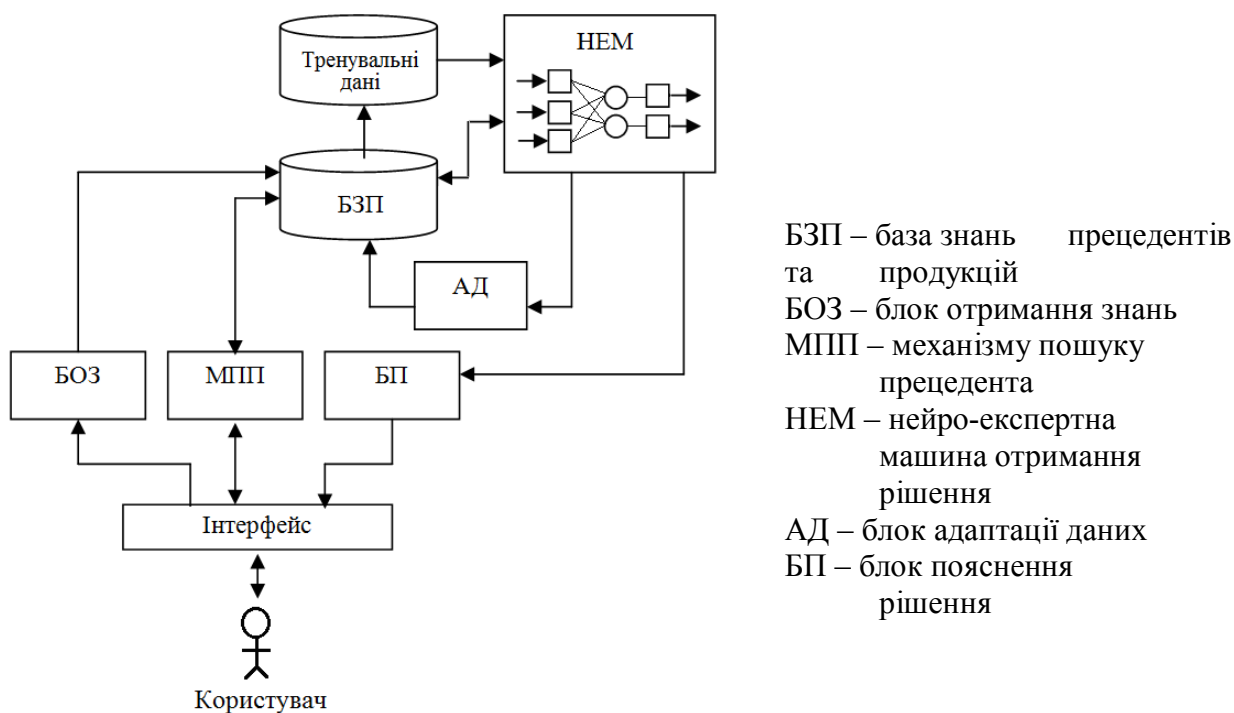


Рисунок 7.6 – Архітектура нейромережевої експертної системи

Навчання нейронної мережі проводиться за методом зворотного поширення помилки з урахуванням бази прецедентів, що зберігаються. Нейромережі пред'являються вхідні образи-факти, для яких мають бути отримані задані значення виходів. У процесі навчання відбувається підстроювання ваги зв'язків між нейронами. Активність нейронів з проміжних шарів обчислюється шляхом підсумовування вагових коефіцієнтів всіх зв'язків нейрона.

Опитування нейромережевої ЕС починається із задання ймовірностей виконання того чи іншого факту ЕС. Нейромережева ЕС може працювати у двох режимах: інтерактивний експертний та неінтерактивний нейромережевий.

В інтерактивному режимі система працює як експертна та ймовірність факту визначається користувачем. У неінтерактивному режимі ймовірність факту формується на основі даних прецедентів і вибирається значення, що найчастіше зустрічається.

Для пояснення отриманого ЕС рішення щодо кожної залежності, що бере участь у висновку, генерується на основі вхідних параметрів вирішальне правило кон'юнкцій посилок та наслідків продукції експертної системи.

Готове рішення формується у вигляді прецеденту та зберігається в базі знань прецедентів експертної системи.

Загальний алгоритм функціонування системи складається з наступних кроків:

- надходження проблеми;
- формування запиту системи;
- пошук рішення на основі знань прецедентів;

- якщо рішення не знайдено, то отримане рішення передається в блок пояснень;
- якщо рішення не знайдено, то виконується передача завдання до нейро-експертного механізму;
- пошук рішення за допомогою нейро-експертного механізму;
- рішення передається в блок адаптації даних;
- блок адаптації даних перетворює рішення нейро-експертної системи на прецедент, який додається до БЗ прецедентів;
- отримане рішення передається в блок пояснень.

Пропонований підхід до інтеграції інтелектуальних технологій має безліч переваг, є ефективним для вирішення інтелектуальних завдань.

Контрольні запитання

1. Прокоментуйте аналогію нейромережі і мозку людини.
2. Що є нейромережевими експертними системами?
3. Поясніть схему біологічного нейрона.
4. Поясніть схему та принцип роботи базового штучного нейрона.
5. Що таке передатна функція нейрона?
6. Поясніть діаграму простої нейронної мережі.
7. Поясніть принцип роботи нейромережі із зворотним зв'язком.
8. У чому полягає процес навчання нейромережі?
9. Які особливості застосування нейромереж?
10. Прокоментуйте застосування нейромереж для вирішення практичних завдань.
11. З чим пов'язані проблеми застосування нейромереж?
12. Що є нейромережевими експертними системами?

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. 5 полезных приемов при проектировании искусственного интеллекта [Электронный ресурс] : учебный курс Интернет-университета информационных технологий. – Режим доступа : <http://gamesmaker.ru/programming/ai/5-poleznyh-priemov-pri-proektirovanii-iskusstvennogo-intellekta/>
2. Introduction to CLIPS. [Электронный ресурс]. – Режим доступа: http://ir.nuk.edu.tw:8080/ir/bitstream/310360000Q/11342/2/CLIPS_IntroCLIPS.pdf
3. Алиев Р.А. Производственные системы с искусственным интеллектом. [Текст] / М.: Радио и связь, 1990.
4. Базы знаний интеллектуальных систем. [Текст] / Т.А.Гаврилова, В.Ф.Хорошевский – СПб: Питер, 2000. – 384с.
5. Башмаков А.И., Башмаков И.А. Интеллектуальные информационные технологии: Учеб. пособие. [Текст] / – М.: Изд-во МГТУ им. Н.Э. Баумана, 2005. – 304 с.
6. Вступ до експертних систем: Навч. посіб. / Кравець В.О., Хавіна І.П. та ін. — Харків: НТУ “ХПІ”, 2006. — 232 с.
7. Гаврилова Т.А., Хорошевский В.Ф. Базы знаний интеллектуальных систем. СПб., 2000.
8. Джозеф Джарратано Экспертные системы. Принципы разработки и программирование, 4-е издание [Текст] / Джозеф Джарратано, Гари Райли. Вильямс 2007. 1152с.
9. Джон Серль. Разум мозга – компьютерная программа. 2-е изд. [Текст] / - М «Вильямс» 2006 – с. 1296.
10. Експертні системи в медицині: Навчальний посібник / Продеус А. М., Синєкоп Ю. С., Швець Є. Я., Кісельов Є. М., Баран М. М. – Запоріжжя: Видавництво ЗДІА, 2014.
11. Жернаков С.В. Нейросетевая база знаний прецедентов активной экспертной системы для комплексного контроля и диагностики параметров авиационного двигателя // Информационные технологии. 2002. № 5. С. 45.
12. Искусственный интеллект и универсальное мышление. [Электронный ресурс] : учебный курс Интернет-университета информационных технологий. – Режим доступа : <http://gamesmaker.ru/books/iskusstvennyy-intellekt-i-universalnoe-myshlenie/>
13. Искусственный интеллект: В 3-х кн. Кн.2. Модели и методы: Справочник. [Текст] / / Под ред. Д.А. Поспелова. – М.: Радио и связь, 1990. – 304 с.
14. Искусственный интеллект: Учеб. пособие для вузов [Текст] / В.Н.Бондарев, Ф.Г.Аде. – Севастополь:Изд-во СевНТУ,2002.
15. Карпов Л.Е., Юдин В.Н. Методы добычи данных при построении локальной метрики в системах вывода по прецедентам / ИСП РАН, препринт № 18. М., 2006. 5. Савушкин С.А. Нейросетевые экспертные системы // Нейрокомпьютер. 1992. № 2. С. 29.

- 16.Классификация моделей представления знаний [Электронный ресурс]. – Режим доступа:.. – <http://www.aiportal.ru/articles/knowledge-models/classification.html>
- 17.Колесников А.В., Кириков И.А. Методология и технология решения сложных задач методами функциональных гибридных интеллектуальных систем. М., 2007.
- 18.Кофман А. Введение в теорию нечетких множеств. [Текст] / – М.: Радио и Связь, 1982. – 431 с.
- 19.Кургаев А.Ф. Анализ моделей представления знаний // Нові комп'ютерні засоби, обчислювальні машини та мережі. – К.: Ін-т кібернетики ім. В.М. Глушкова НАН України, 2001. – Т.1. – С. 129–135.
- 20.Люгер, Д.Ф. Искусственный интеллект: стратегии и методы решения сложных проблем [Текст] / Д.Ф.Люгер. – М.: Издательский дом «Вильямс», 2003.
- 21.Месюра В. І., Яровий А. А., Арсенюк І. Р. М53 Експертні системи. Частина 1. Навчальний посібник. – Вінниця: ВНТУ, 2006.– 114 с.
- 22.Мерedit Бруссард. Искусственный интеллект. Пределы возможного. [Текст] / - Альпина Паблишер, 2020. – 362с.
- 23.Разработка экспертных систем. Среда CLIPS [Текст] / А. П. Частиков, Д. Л. Белов, Т. А. Гаврилова - СПб: Питер, 2003. 396с.
- 24.Соловьев В.Д. Учебный курс Интернет-университета информационных технологий. [Электронный ресурс]. – Режим доступа: <http://www.intuit.ru/department/expert/ontoth/2/>
- 25.Т.Таулли Основы искусственного интеллекта. Нетехническое введение [Текст] / - БХВ-Петербург2021- 288с.
- 26.Томас Х. Кормен, Чарльз И. Рональд Л. Ливест Клиффорд Штайн. - Алгоритмы и анализ. 2-е изд. [Текст] / - М «Вильямс» 2006 – с. 1296
- 27.Уотермен, Д. Экспертные системы [Текст] / Д.Уотермен. - М.: Мир, 1990.
- 28.Russel, Stuart J.; Norving, Peter (2003), Artificial Intelligence: A Modern Approach (2nd ed.) Upper Saddle River, New Jersey: Prentice Hall, pp. 375-459, ISBN 0-13-790395-2.
- 29.Bostrom N. Superintelligence. Paths, Dangers, Strategies [Текст] / N.Bostrom. - Oxford University Press, 2016 – 432pp.
- 30.Bryan S. Todd. AN INTRODUCTION TO EXPERT SYSTEMS. [Электронный ресурс]. – Режим доступа: <https://www.cs.ox.ac.uk/files/3425/PRG95.pdf>
- 31.Bruce G. Buchanan & Reid Q. Smith. FUNDAMENTALS OF EXPERT SYSTEMS. [Электронный ресурс]. – Режим доступа: <https://stacks.stanford.edu/file/druid:gh880nv8408/gh880nv8408.pdf>
- 32.Champanand A.J. Getting started with decision making and control systems [Текст] / A.J. Champanand. –AI Game Programming Widsom, Vol.4, 2008 – pp. 257-263.

33. Champandard A.J. The behavior tree starter kit [Текст] / A.J. Champandard, P. Dustan. – In Game AI Pro: Collected Widsom for Game AI Professionals. Boca Raton, FL: A K Peters/CRC Press, 2013.
34. Introduction to CLIPS. [Электронный ресурс]. – Режим доступа: http://ir.nuk.edu.tw:8080/ir/bitstream/310360000Q/11342/2/CLIPS_IntroCLIPS.pdf
35. Introduction to Expert Systems. [Электронный ресурс]. – Режим доступа: <https://nou.edu.ng/coursewarecontent/CIT474%20Expert%20systems.pdf>
36. Introduction to Expert Systems. [Электронный ресурс]. – Режим доступа: <http://www.sci.brooklyn.cuny.edu/~dzhu/cis718/preview01.pdf>
37. Nilsson N.J. The quest for artificial intelligence. // [Электронный ресурс]: Stanford University. – Режим доступа: <http://ai.stanford.edu/~nilsson/QAI/qai.pdf>
38. Poole D. Artificial Intelligence // [Электронный ресурс]: D. Poole, A. Mackworth – Режим доступа: <http://artint.info/html/ArtInt.html>.
39. Ray Barrera, Aung Sithu Kyaw, and Thet Naing Swe titled Unity 2017 Game AI Programming – Third Edition.
40. Smart Computing and Self-Adaptive Systems. Edited By Simar Preet Singh, Arun Solanki, Anju Sharma, Zdzislaw Polkowski, Rajesh Kumar CRC Press, 2022 – 288pp.

Наукове видання

Людмила НІКІТІНА

ЕКСПЕРТНІ СИСТЕМИ

Навчальний посібник
для студентів спеціальності 122 – Комп'ютерні науки

Укладач НІКІТІНА Людмила Олексіївна

За авторською редакцією