Тут буде тітулка

Тут буде лист завдання

Тут буде реферат

Тут буде зміст

# Вступ

# 1 ЗАГАЛЬНИЙ РОЗДІЛ

## Експертні системи

Експертні системи (ЕС) є галуззю прикладних штучних інтелект (ІІ) та були розроблені спільнотою ІІ у середині 1960-х років. Основна ідея ES полягає в тому, що експертиза, яка є великою сукупністю знань з конкретних завдань, передається від людини до комп'ютера. Це знання потім зберігається на комп'ютері, і користувачі викликають комп'ютер для конкретних порад при необхідності. Комп'ютер може зробити висновки та дійти конкретного висновку. Потім як людина-консультант, дає поради та пояснює, якщо необхідно, логіка порад [1]. ES надає потужні та гнучкі засоби для отримання рішень різних проблем, які часто не можуть бути вирішені іншими, більш традиційними та ортодоксальними методи. Таким чином, їх використання поширюється на багато секторів. наше соціальне та технологічне життя, де їх застосування бути критично важливим у процесі підтримки прийняття рішень та вирішення проблем [2].

## Класифікація експертних систем

Існує багато способів як можна класифікувати ЕС системи, ось деякі із цих способів класифікації:

Класифікація експертних систем, що відображає специфіку задач, які вирішуються за допомогою цієї технології [3]:

1. Інтерпретуючі системи призначені для формування опису ситуацій по результатам спостережень чи даним, одержуваним від різноманітних сенсорів [3].
2. Прогнозуючі системи призначені для логічного аналізу можливих наслідків заданих ситуацій чи подій [3].
3. Діагностичні системи призначені для виявлення джерел несправностей по результатів спостережень за поведінкою контрольованої системи [3].
4. Системи проектування призначені для структурного синтезу конфігурації об'єктів за заданих обмежень [3].
5. Системи планування призначені для підготовки планів проведення послідовності операцій, що веде до заданої мети [3].
6. Системи моніторингу аналізують поведінку контрольованої системи та, порівнюючи отримані дані з критичними точками заздалегідь складеного плану, прогнозують ймовірність досягнення поставленої мети [3].
7. Налагоджувальні системи призначені для вироблення рекомендацій щодо усунення несправностей у контрольованій системі [3].
8. Системи надання допомоги під час ремонту обладнання виконують планування процесу усунення несправностей у складних об'єктах, наприклад у інженерних мережах комунікацій [3].
9. Навчальні системи проводять аналіз знань студентів з певного предмета, знаходять прогалини в знаннях і пропонують засоби для їх ліквідації [3].
10. Системи контролю забезпечують адаптивне управління поведінкою складних людино-машинних систем, прогнозуючи появу можливих збоїв та плануючи дії, необхідні для їх попередження [3].

За типом використання розрізняють:

1. Ізольовані ЕС [4].
2. Гібридні ЕС інтегровані з базами даних та іншими програмними продуктами (додатками) [4].

За рівнем залежності від зовнішнього середовища виділяють:

1. Статичні ЕС, які залежать від зовнішнього середовища [4].
2. Динамічні, що враховують динаміку зовнішнього середовища, та призначені для вирішення завдань у реальному часі [4].

## 1.1.2 Склад експертної системи

Експертні системи відрізняються між собою своє складністю, та кількістю компонентів, але у кожної ЕС є чотири базові компоненти:

1. База знань.
2. Механізм виведення.
3. Модуль набуття знань.
4. Пояснювальний інтерфейс.

База знань (БЗ) - містить факти (або затвердження) та правила. Факти – це короткострокова інформація, яка можна швидко змінюватись. Правила – це довгострокова інформація про те як генерувати нові факти або гіпотези з того, що зараз відомо. Основна відмінність що основа знань більш креативна. Факти в базі даних зазвичай пасивні: вони або там є або їх там. З іншого боку, база знань активно намагається заповнити інформацію, що бракує [5].

Механізм виведення – це програма, яка модулює те, яка саме інформація буде виводитись ґрунтуючись на фактах, та правилах, які знаходяться у БЗ. Існують два механізми виведення інформації: прямий ланцюжок - включає міркування від даних до гіпотез, та зворотний ланцюжок - намагається знайти дані для підтвердження або спростування гіпотези. Чистий прямий ланцюжок призводить до несфокусованих питань у системі діалогового режиму, тоді як чистий зворотний ланцюжок має тенденцію бути досить невтомним у своїх цілеспрямованих питаннях [5].

Модуль набуття знань – модуль до якого залучається експерт зі знань необхідний для деякої ЕС, та інженер зі знань який має навички у роботі із програмою, через співпрацю цих людей йде систематизація знань та рішень експерта та доведення їх до мови, яка розуміє програма [5].

Пояснювальний інтерфейс – програма, або їх комплекс, за допомогою якої можливий діалог між користувачем та експертною системою [5].

## 1.1.3 Етапи створення експертної системи

Технологія розробки експертних систем включає шість наступних етапів:

1. Ідентифікація.
2. Концептуалізація.
3. Формалізація.
4. Впровадження.
5. Тестування.
6. Дослідна експлуатація [6].

На етапі ідентифікації процес розвитку планується прототип системи [7], джерела знання (книги, експерти та методології), цілі (розподіл досвіду, автоматизація рутинних операції), класи розв'язуваних завдань та багато іншого. Результатом ідентифікації є відповідь на питання, що потрібно зробити і які ресурси потрібно використовувати.

На етапі концептуалізації проводиться поглиблений аналіз проблемної галузі, визначаються методи вирішення проблеми. Цей етап завершується створенням моделі предметної області (програмного забезпечення), що включає основні поняття та відносини [8]. На етапі концептуалізації визначаються такі характеристики проблеми:

1. Доступні типи даних;
2. Вхідні та вихідні дані, під задачі загального проблема;
3. Застосовувані стратегії та гіпотези;
4. Типи відносин між програмними об'єктами, типи використовуваних відносин;
5. Процеси, що використовуються у рішеннях;
6. Зміст знань, що використовуються під час вирішення проблем;
7. Типи обмежень, що накладаються на процеси, що використовуються в рішеннях;
8. Зміст знань, що використовується у поясненнях рішень [8]

На етапі формалізації вибирається і визначаються шляхи уявлення всіх видів знань, формалізуються основні поняття, способи пізнання визначеної інтерпретації, моделювання роботи системи, адекватність цілей, що стосуються системи фіксованого визначення, оцінюється методи вирішення, способи презентації [9]. Вихід стадії формалізації є описом того, як проблема можуть бути представлені у вибраному або розвиненому формалізмі (фрейми, сценарії, семантичні мережі та ін.) та розуміння методів маніпулювання цим знанням [10].

На етапі впровадження, коли наповнюється база даних з експертом. Процес набуття знань виконується інженером з знань на основі дії експертів щодо реальних рішень проблем. Мета цього етапу – створення одного прототипу ЕС. В подальшому за результатами тестування та дослідної експлуатації кінцевий продукт, придатний для промислового використання, створюється за цей етап [11]. Розробка прототипу складається з програмування компонентів або їх вибір серед відомих програмних інструментів та наповнення бази знань [12]. Таким чином, в даний час процес оцінки ситуації та прийняття рішень є одним з найбільш трудомістких, тому методологія розробки експертних систем вважає наступні потреби найважливішими:

1) Правильна постановка задачі;

2) Систематизація знань для передачі їх в комп'ютерну систему;

3) Розробка інструментів управління знаннями бази даних, індуктивний висновок та спрощений діалог.

## 1.1.4 Переваги та недоліки експертних систем

Експертні системи найбільш поширені у складній проблемній галузі і вважаються альтернативами, що широко використовуються, при пошуку рішень, що вимагають наявності спеціальних людських знань. Експертна система також здатна обґрунтовувати запропоновані нею рішення на основі знань та даних, отриманих від попередніх користувачів [13].

Відмінність експертної системи від звичайної системи вирішення завдань у тому, що остання є системою, у якій кодуються програми, і структури даних, у той час як для експертної системи жорстко закодовані лише структури даних, і жодна специфічна для проблеми інформація не кодується у структурі програми. Натомість знання про людський досвід фіксуються та систематизуються в процесі, відомому як інженерія знань. Отже, щоразу, коли конкретна проблема вимагає допомоги певного людського досвіду на вирішення, систематизований людський досвід використовуватиметься і оброблятиметься для забезпечення раціонального та логічного рішення. Ця експертна система, що базується на знаннях, дозволяє системі часто додавати нові знання; і відповідним чином адаптуватися до нових вимог постійно мінливого та непередбачуваного середовища [13].

Переваги використання експертної системи:

1. Надання послідовних рішень: Він може давати послідовні відповіді на повторювані рішення, процеси та завдання. Поки основа правил у системі залишається постійною, незалежно від цього, скільки разів тестуються подібні завдання, зроблені висновки залишаться тими самими [13].
2. Дає розумні пояснення: може прояснити причини, з яких було зроблено висновок і чому він вважає найбільш логічним вибір серед інших альтернатив. Якщо є якісь сумніви у укладанні певного завдання; він запропонує користувачам відповісти на деякі питання, щоб опрацювати логічний висновок [13].
3. Долає людські обмеження: ЕС не має людських обмежень і може працювати цілодобово безперервно. Користувачі зможуть часто використовувати його для пошуку рішень. Знання експертів є безцінним активом для компанії. Віна може зберігати знання і використовувати їх, доки вони потрібні організації [13].
4. Легко адаптувається до нових умов: На відміну від людей, у яких часто виникають проблеми з адаптацією до нових умов, експертна система має високу адаптивність і може відповідати новим вимогам за короткий період часу. Він також може отримувати нові знання від експерта та використовувати їх як правила виводу для вирішення нових проблем [13].

Недоліки використання експертної системи:

1. Бракує здорового глузду: у ЕС відсутній здоровий глузд, необхідний для прийняття деяких рішень, оскільки всі прийняті рішення засновані на правилах виведення, встановлених у системі. Він також не може давати творчі та новаторські відповіді, як люди-експерти у незвичайних обставинах [13].
2. Висока вартість впровадження та обслуговування: впровадження експертної системи в бізнесі буде фінансовим тягарем для невеликих організацій, оскільки воно пов'язане з високими витратами на розробку, а також подальшими періодичними витратами на модернізацію системи для адаптації до нового середовища [13].
3. Складність створення правил виведення: експерти предметної галузі не завжди зможуть пояснити свою логіку та міркування, необхідні для процесу інженерії знань. Таким чином, завдання кодування знань дуже складне і може вимагати великих зусиль [13].
4. Може надати неправильні рішення: ЕС не безпомилкова. При обробці можуть виникнути заблудження через деякі логічні помилки, допущені в базі знань, які потім будуть давати неправильні рішення [13].

## 1.1.5 Моделі представлення знань у експертних системах.

Уявлення знань — це, власне, сурогат, замінник самої речі, використовуваний у тому, щоб дозволити сутності визначати наслідки, думаючи, міркуючи про світ , а не роблячи у ньому дії. Це фрагментарна теорія розумного мислення, виражена у термінах трьох компонентів [14]:

1. Фундаментальна концепція уявлення розумного мислення;
2. Набір висновків, на які накладаються репрезентативні санкції;
3. Набір висновків, які він рекомендує.

Це середовище для прагматично ефективних обчислень, в якому здійснюється мислення. Один внесок у цю прагматичну ефективність робить керівництво, яке дає уявлення для організації інформації, щоб полегшити отримання рекомендованих висновків. Це засіб людського вираження, тобто мова, якою ми говоримо про світ. Розуміння ролей та визнання їхньої різноманітності має кілька корисних наслідків. По-перше, кожна роль вимагає чогось трохи відмінного від уявлення. Кожна роль відповідно призводить до цікавого та різного набору властивостей, у яких ми хочемо мати уявленні. По-друге, ми вважаємо, що ролі забезпечують основу, корисну для характеристики найрізноманітніших репрезентацій. Ми припускаємо, що фундаментальне «мислення» репрезентації можна вловити, зрозумівши, як воно розглядає кожну з ролей, і це виявить суттєві подібності та відмінності. По-третє, ми вважаємо, що деякі попередні розбіжності з приводу представництва можна з користю вирішити, якщо належним чином розглянути всі п'ять ролей. Ми демонструємо це, переглядаючи та аналізуючи ранні аргументи, що стосуються фреймів та логіки. Нарешті ми вважаємо, що такий погляд на репрезентації має наслідки як для досліджень, так і для практики. Для дослідження ця точка зору дає одну пряму відповідь на питання, що має фундаментальне значення у цій галузі. Він також пропонує прийняти широкий погляд на те, що важливо у репрезентації, і доводить, що однієї важливої ​​частини зусиль з репрезентації — захоплення та уявлення багатства світу природи — приділяється недостатньо уваги [14].

## 1.2 Засоби розробки експертних систем

## Засоби розробки експертної системи - це системи програмування, які спрощують роботу з побудови експертної системи. Вони варіюються від мови програмування дуже високого рівня засобів підтримки низького рівня [15].

## 1.2.1 Мови програмування

Найбільш важливими мовами програмування, що використовуються для експертних систем, зазвичай є або проблемно-орієнтовані мови, такі як FORTRAN і PASCAL, або мови обробки символів, такі як LISP і PROLOG. Проблемно-орієнтовані мови призначені на вирішення певних класів завдань: наприклад, FORTRAN має зручні функції для виконання алгебраїчних обчислень і найбільш застосовний до наукових, математичних та статистичних проблем. Мови маніпулювання символами призначені додатків штучного інтелекту: наприклад, LISP має механізм маніпулювання символами у вигляді структур списку. Список — це просто набір елементів, укладених у круглі дужки, де кожен елемент може бути символом або іншим списком. Найбільш популярною і широко використовуваною мовою програмування для програм експертних систем є LISP. Такі мови маніпулювання символами більше підходять для роботи зі штучним інтелектом, хоча кілька експертних систем були написані проблемно-орієнтованими мовами, такими як FORTRAN та PASCAL. ЕС зазвичай пишуться такими мовами, як LISP і PROLOG або навіть CLIPS (інтегрована виробнича система мовою C, розроблена в середині 1980-х років). Використання цих мов для розробки е.с. спрощує процес кодування. Основними перевагами цих мов у порівнянні зі звичайними мовами програмування є простота додавання, виключення або заміни нових правил та можливості керування пам'яттю [15].

Деякі з характерних рис мов програмування, необхідні для розробки ЕС є:

1. Ефективне поєднання цілих та речових змінних.
2. Хороші процедури керування пам'яттю.
3. Великі процедури маніпулювання даними.
4. Інкрементна компіляція.
5. Архітектура пам'яті з тегами.
6. Оптимізація системного середовища
7. Ефективні процедури пошуку.

Мови програмування, такі як LISP, CLIP і т. д., пропонують найбільшу гнучкість розробнику експертної системи, але можуть дати вказівок у тому, як представляти знання чи механізми доступу до бази знань. З іншого боку, мови інженерії знань, такі як KAS пропонують невелику гнучкість, оскільки розробник системи повинен використовувати схему управління, визначену готовим механізмом виведення. Однак вони надають рекомендації щодо подання та готові механізми виведення для управління використанням бази знань [15].

## 1.2.2 Мови інженерії знань

Мова інженерії знань - це складний інструмент для розробки експертних систем, що складається з мови побудови експертних систем, інтегрованої в середовище підтримки. Мова програмування - це розробка штучної мови для отримання знань, прийняття/відхилення знань контролю та управління роботою комп'ютера [15].

Мова інженерії знань - це тип мови програмування, призначений для побудови та налагодження експертної системи, мова надає певні можливості для побудови експертної системи; вони гнучкіші, ніж інші мови програмування, у тому, що стосується представлення знань та управління ними. Мови інженерії знань можна віднести або до скелетних систем, або до систем загального призначення. Дослідники ІІ розробили ці мови спеціально для створення експертних систем. Скелетна мова інженерії знань - це просто урізана експертна система, тобто експертна система, з якої видалено її предметно-орієнтовані знання (оболонка), залишивши лише механізм виведення та допоміжні засоби. Мова інженерії знань загального призначення може обробляти безліч різних проблемних областей, таких як вилучення знань, створення логічних висновків або створення інтерфейсу користувача, хоча його використання досить стомлююче. Ці мови відрізняються універсальністю та гнучкістю [15].

Розробники PROSPECTOR позбавили його знань про геологію, щоб перетворити його на KAS, скелетну систему для діагностики та класифікації. Так само система MYCIN для діагностики та лікування бактеріальних інфекцій стала скелетною системою EMYCIN, а консультаційна система CASNET по глаукомі стала скелетною системою EXPERT. Каркасні системи забезпечують структуру та вбудовані засоби, які спрощують та прискорюють розробку системи. Але їм не вистачає спільності та гнучкості; вони застосовні лише до обмеженого класу завдань та значно скорочують можливості проектування експертних систем [15].

EMYCIN

Ця каркасна мова інженерії знань по суті є MYCIN з віддаленими знаннями предметної області. EMYCIN використовує схему представлення знань на основі правил із жорстким механізмом управління зі зворотним ланцюжком, який обмежує його застосування проблемами діагностики та класифікації. Тим не менш, система надає складні засоби пояснення та отримання знань, що явно прискорюють розробку експертної системи. EMYCIN корисна інженерна мова для діагностичних властивостей, тому він використовувався для створення експертних систем у медицині, геології, техніці, сільському господарстві та інших галузях [15].

EXPERT

Ця скелетна мова інженерії знань використовує схему представлення знань, засновану на правилах, і має обмежений механізм керування прямим ланцюжком, що робить його придатним для задач діагностики та класифікації. EXPERT має вбудовані засоби пояснення, збирання знань та перевірки несуперечності для прискорення розробки системи. Модуль перевірки несуперечності працює, зберігаючи базу даних репрезентативних прикладів з відомими висновками та використовуючи її для тестування експертної системи після того, як інженер за знаннями додасть нових правил [15].

Якщо випадок не призводить до правильного висновку, EXPERT відображає обґрунтування для цього випадку, щоб інженер знань міг зрозуміти, як нові правила привели до несподіваних результатів. EXPERT використовувався для побудови діагностичних програм у медицині, геології та інших галузях [15].

OPS5

Ця мова інженерії знань загального призначення використовує схему представлення на основі правил, що працює через прямий ланцюжок. Універсальність системи підтримує різноманітні структури представлення даних та управління в рамках однієї програми. OPS5 має потужний засіб зіставлення зі зразком та ефективний інтерпретатор для зіставлення правил із даними, але не має складного середовища підтримки. Вона не має вбудованих механізмів пояснення чи отримання, та має лише мінімальні засоби для редагування та налагодження програм. OPS5 є останнім серед подібних мов, заснованих на правилах, які виникли в результаті роботи в Університеті Карнегі-Меллона з розробки мов програмування для моделювання людського пізнання та пам'яті. OPS5 та ранні мови серії OPS використовувалися для багатьох додатків когнітивної психології, штучного інтелекту та експертних систем [15].

## 1.2.3 Допоміжні засоби для побудови систем

Допоміжні засоби для побудови системи складаються з програм, які допомагають набувати та представляти знання експерта в предметній галузі, та програм, які допомагають проектувати експертну систему, що будується. Ці програми вирішують складні завдання. Багато з них є дослідницькими інструментами, які тільки починають перетворюватися на практичні та корисні допоміжні засоби, хоча деякі з них пропонуються як повноцінні комерційні системи. Порівняно з мовами програмування та мови інженерних знань було розроблено відносно небагато допоміжних засобів для побудови систем. Ті, що існують, поділяються на дві основні категорії; засоби проектування та засоби придбання знань [15].

Система AGE служить прикладом засобів проектування, а TEIRSIAS, MOLE і SALT є прикладом засобів отримання знань, побудови системи TIMM і засобів уточнення знань SEEK [15].

AGE

Цей програмний інструмент допомагає інженеру зі знань проектувати та створювати експертну систему. AGE надає користувачеві набір компонентів, які, подібно до будівельних блоків, можуть бути зібрані для формування частин експертної системи. Кожен компонент, що є набором функцій INTERLISP, підтримує каркас експертної системи, такий як прямий ланцюжок, зворотний ланцюжок або архітектура Blackboard. Термін Blackboard відноситься до центральної бази даних, що користуються системами для координації та контролю роботи незалежних груп та правил, які називають джерелами знань [15].

Джерела знань спілкуються, записуючи повідомлення на дошці та читаючи повідомлення з інших джерел знань. Ця архітектура була вперше використана в HEARSAY-II, системі розуміння мови, розробленої в середині 1970-х років, і привела до HEARSAY-III, мови інженерії знань для управління кількома джерелами знань. Системи Blackboard є основою сучасної архітектури інтерфейсу користувача. Інженери знань використовували AGE для розробки та створення HANNIBAL, експертної системи, яка виконує оцінку ситуації, інтерпретуючи дані радіозв'язку супротивника. Система використовує інформацію про місцезнаходження та сигнальні характеристики даних для ідентифікації організаційних підрозділів противника та їх зв'язку, бойового порядку [15].

MOLE

Це система набуття знань для евристичного завдання класифікації, такої як діагностика захворювань. Зокрема, він використовується у поєднанні з методом покриття та диференціації завдань. Експертна система MOLE приймає вхідні дані, пропонує набір можливих пояснень або класифікацій, які охоплюють дані, а потім використовує знання, що диференціюють, щоб визначити, яке з них краще. Процес ітеративний, тому що пояснення самі мають бути обґрунтовані доти, доки не будуть встановлені кінцеві причини. MOLE взаємодіє з експертом у предметній області для створення бази знань, яку система, звана MOLE-p, використовує на вирішення проблем [15].

Придбання знаннь відбувається у кілька етапів:

1. Побудова вихідної бази знань. MOLE просить експерта перерахувати загальні симптоми чи скарги, які можуть вимагати діагностики. Для кожного симптому MOLE запитує перелік можливих пояснень. Потім MOLE в інтерактивному режимі шукає пояснення вищого рівня, доки знайде набір кінцевих причин. У ході цього процесу MOLE створює мережу впливу, аналогічну до мережі переконань. Щоразу, коли подія має кілька пояснень, MOLE намагається визначити умови, за яких одне з пояснень є правильним. Експерт надає знання про те, що гіпотетична подія може бути причиною певного симптому. MOLE намагається вивести попереднє знання, яке говорить, що якщо гіпотетична подія не відбудеться, та симптом обов'язково з'явиться. Це знання дозволяє системі виключити певну гіпотезу на підставі, що певні системи відсутні [15].

2. Уточнення основи знань. Тепер MOLE намагається виявити слабкі місця у базі знань. Один із підходів полягає в тому, щоб знайти дірки та запропонувати експерту їх заповнити. Як правило, важко дізнатися, чи є база знань повною, тому натомість MOLE дозволяє експертам спостерігати за тим, як MOLE-p вирішує приклади завдань. Щоразу, коли MOLE-p ставить невірний діагноз, експерт додає нові знання. Є кілька способів, якими MOLE-p може дійти неправильних висновків. Він може помилково відкинути гіпотезу, оскільки вважає, що гіпотеза необхідна пояснення будь-якого симптому. Він може висувати гіпотезу, тому що він необхідний для пояснення якоїсь інакше незрозумілої гіпотези. Або йому може бракувати диференційного знання для вибору між альтернативними гіпотезами. Наприклад, у лікаря є пацієнт із симптомами A та B. Далі, припустимо, що симптом A може бути викликаний подіями X та Y, а симптом B може бути викликаний Y та Z. MOLE-p може укласти Y, оскільки він пояснює як A , так і B. Якщо експерт вказує, що це рішення було неправильним, то MOLE запитає, які докази слід використовувати, щоб віддати перевагу X та/або Z Y. MOLE використовувався для створення систем, що діагностують проблеми з автомобільними двигунами, проблеми на сталеливарних заводах та неефективність вугільних електростанцій. Щоб MOLE була застосовна, кодування знань з погляду охоплення та диференціації має бути виконане заздалегідь [15].

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

Програма SALT надає механізми для роз'яснення цього знань від експерта. Ця система, заснована на методі «запропонуй і переглянь».

По-перше, система пропонує розширення поточного проекту. Потім вона перевіряє, чи не порушує розширення будь-які глобальні чи локальні обмеження. Потім порушення обмежень фіксуються і процес повторюється. Виявляється, експерти у предметній галузі добре вміють перераховувати загальні проектні обмеження та встановлювати локальні обмеження для окремих параметрів, але не так добре пояснюють, як дійти глобальних рішень. SALT будує мережу, орієнтовану залежність, під час спілкування з експертом. Кожен вузол означає значення параметра, який повинен бути отриманий або згенерований.

Существует три вида ссылок: Способствует Ограничения Suggested\_revision of Узел A вносит вклад в узел B, если значение в A появляется при вычислении значения в B, в то время как узел A ограничивает B, если значение A запрещает B принимать определенные значения. С первым типом связи связаны процедуры, позволяющие SALT генерировать значение одного параметра на основе значения другого. Второй тип связи, ограничения, исключает определенные значения параметров. Третья ссылка, предлагаемая-пересмотр-из, указывает на способы исправления нарушения ограничения. SALT использует следующие эвристики в процессе выявления: я. Каждому невходному узлу в сети требуется по крайней мере одна входящая в него ссылка. Если ссылки отсутствуют, эксперту предлагается их заполнить. II. В сети не допускаются циклы участия, без значения хотя бы одного параметра в цикле невозможно вычислить значения для любого параметра в этом цикле. Если петля существует, SALT пытается преобразовать одну из взаимодействующих ссылок в ограничения. III. Ограничивающие ссылки должны иметь связанные с ними ссылки с предложением пересмотра. К ним относятся ссылки ограничений, которые создаются при разрыве циклов зависимости. Знания управления также важны. Крайне важно, чтобы система предлагала расширения и изменения, ведущие к проектному решению. SALT позволяет эксперту оценивать исправления с точки зрения того, сколько проблем они могут вызвать. SALT компилирует свою сеть зависимостей в набор производственных правил. Эксперт может наблюдать, как производственная система решает проблемы, и может отменить решение системы. В этот момент база знаний может быть изменена или переопределение может быть зарегистрировано для будущей проверки. в. ТИРЕСИЙ: TEIRESIAS был разработан Дэвисом в середине 1970-х годов в Стэнфордском университете (США) как инструмент для исследования новых идей в области сбора знаний и обслуживания баз данных, а не как инструмент для создания экспертных систем. TEIRESIAS получает знания в интерактивном режиме от эксперта. Если MYCIN поставил неверный диагноз, то TEIRESIAS проведет эксперта по цепочке неверных рассуждений до тех пор, пока эксперт не укажет, с чего началось неправильное рассуждение. Возвращаясь к цепочке рассуждений, TEIRESIAS также будет взаимодействовать с экспертом, чтобы изменить неверные правила или получить новые правила. Информация о новых правилах не сразу заносится в MYCIN. Вместо этого TEIRSIAS проверяет, совместимо ли новое правило с аналогичными правилами. Например, если новые правила описывают, как инфекция попадает в организм, а в других принятых правилах есть условный элемент, указывающий портал входа в организм, то новое правило также должно. Если в новом правиле не указан входной портал, то TEIRESIAS запросит у пользователя это несоответствие. TEIRESIAS имеет шаблон модели правил, состоящий из похожих правил, которые он знает, и пытается вписать новое правило в свою модель правил. Другими словами, модель правила — это знание, которое ТИРЕСИАС имеет о своем знании. TEIRESIAS служил интерфейсом для экспертной системы MYCIN. Фрагмент беседы TEIRSIAS-MYCIN с пользователем (врачом) приведен на рис. 12.19. Программа — это информация, необходимая ей для продолжения рассуждений. Врач хочет знать, почему программе нужна информация, а также спрашивает, как программа пришла к заключению, к которому, как она утверждает, пришла.

## 1.2.4 Середовище підтримки інструментів

Це просто програмні пакети, які постачаються з кожним інструментом, щоб зробити його зручнішим для користувача та ефективнішим. Середовища, які можна поділити на: (i) Ті, які потрібні для розробки програм експертних систем, такі як засоби налагодження та редактори баз знань. (ii) Ті, які необхідні для розширення можливостей розроблених програм, такі як засоби введення/виводу та механізм пояснення. Хоча кілька експертних системних інструментів підтримують ці інструменти, всі вони підтримують деякі з них. Ці кошти зазвичай доступні як частина K.E. мові та призначені для роботи спеціально з цією мовою. Вони описані нижче: а. Допоміжні засоби налагодження: Більшість мов програмування та інженерії знань містять засоби трасування та розривні пакети. Трасування надає користувачеві звичайне трасування або відображення роботи системи шляхом перерахування імен (або номерів) всіх запущених правил або відображення імен усіх підпрограм. Пакет break дозволяє користувачеві заздалегідь повідомити програму, де зупинитися, щоб користувач міг зупинити виконання програми безпосередньо перед деякою помилкою, що повторюється, і перевірити поточні значення в базі даних. Усі інструменти експертної системи повинні мати ці кошти. Декілька експертних системних інструментів включають автоматичне тестування, дещо екзотичніший засіб налагодження, ніж засоби трасування або зламування пакетів. Ця допомога дозволяє користувачеві автоматично тестувати програму на великій кількості контрольних завдань, щоб виявити помилки чи невідповідності у рішеннях. б. Редактори бази знань: Більшість інструментів експертних систем надають механізм редагування бази знань. У найпростішому випадку це звичайний текстовий редактор для зміни правил та даних вручну. Але багато інструментів включають у своє середовище підтримки інші засоби. Наприклад, EMYCIN використовує автоматичний облік. Редактор EMYCIN відстежує зміни, внесені користувачем, та записує відповідну інформацію про подію. Якщо користувач додає або змінює правило, редактор автоматично зберігає дату зміни та ім'я користувача разом із правилом для подальшого використання. Інженери знань це знають особливо корисним, коли кілька різних експертів модифікують або уточнюють базу знань. Ще одним поширеним засобом у редакторах баз знань є перевірка синтаксису, коли редактор використовує знання про граматичну структуру мови експертної системи, щоб допомогти користувачам запровадити правила з правильним написанням та форматом. Коли користувач вводить неписьменне правило або команду, редактор вловлює це і пояснює, що не так. Виправлення подібних помилок під час редагування, а не під час тестування системи може значно скоротити час розробки. Надзвичайно корисним, але зазвичай недоступним засобом для редакторів баз знань є перевірка узгодженості, коли система перевіряє семантику або значення правил та даних, щоб побачити, чи не суперечать вони існуючим знанням у системі. У разі конфлікту редактор допомагає користувачеві вирішити конфлікт, пояснюючи, що його викликало, і описуючи способи його усунення. Спочатку дослідники ІІ мали труднощі з перевіркою узгодженості, але деякі засоби побудови дослідницьких систем (наприклад, TEIRESIAS) і деякі комерційні засоби побудови систем (наприклад, TIMM) вирішили цю проблему. Щоб перевірити несуперечність більш ніж поверхнево, редактор повинен розуміти, що насправді означають різні форми правил і даних.

# ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Turban, E., & Aronson, J. E. (2001). Decision support systems and intelligent systems, sixth Edition (6th ed). Hong Kong: Prentice International Hall.
2. Shu-Hsien Liao. (2005). Expert system methodologies and applications—a decade review from 1995 to 2004. Expert Systems with Applications 28 (2005) 93–103.
3. Питер Джексон (2001). Введение в экспертные системы = Introduction to Expert Systems. — 3-е изд. — М.: Вильямс — С. 624.
4. Страхова О.П. (2014). Экспертные системы в фармации: Методическое пособие для студентов фармацевтического факультета - Запорожье: ЗГМУ, - 36с.
5. Forsyth, R.S. (1984). The Architecture of Expert Systems. In: Forsyth, R.S. (ed.) Expert Systems: Principles & Case Studies. London: Chapman & Hall Ltd., pp. 9-17.
6. Kantureeva M. Zakirova A. (2014) The Methodology of Expert Systems IJCSNS International Journal of Computer Science and Network Security, VOL.14 No.2,
7. Chastikov A.P. Gavrilova T.A. Belov D. L. Development of expert systems. CLIPS environment. – SPb. : BHVPetersburg, 2003. – 608 p.
8. Informatics: The textbook / Under the editorship of the prof. of N. V. Makarova – M: Finance and statistics, 1997.
9. Artificial intelligence: Application in the integrated production systems / Under the editorship of E.Kyyusiak; The Lane with English – M: Mechanical engineering, 1991
10. Artificial intelligence: in 3 books of Book 1. Systems of communication and expert systems. The directory / Under the editorship of E.V.Popova – M: Radio and communication, 1990
11. Page Osuga. Processing of knowledge: The lane with M: World, 1989
12. Priests E.V.Ekspertnye systems: The solution of unformalized tasks in dialogue with the COMPUTER. – M: Science. Hl. edition physical. 1987
13. Advantages and disadvantages of Expert Systems. ilearnlot. URL: <https://www.ilearnlot.com/expert-system-advantages-disadvantages/34332/>
14. Bahramі M., Kaviani S. A New Method for Knowledge Representation in Expert System's. First International Conference on Emerging Trends in Engineering and Technology. 2008.
15. Subarna D. Construction of an Expert System: 4 Tools | Artificial Intelligence. URL:<https://www.engineeringenotes.com/artificial-intelligence-2/expert-systems/construction-of-an-expert-system-4-tools-artificial-intelligence/35582>.