

ЛЕКЦІЯ №3

МЕТОДОЛОГІЇ СТВОРЕННЯ ЕКСПЕРТНИХ СИСТЕМ

ПЛАН

3.1 Методи створення експертних систем, які засновані на правилах.

3.2 Методи створення експертних систем, які базуються на логіці.

3.3 Принципи та базові етапи створення ЕС.

3.4 Етапи розробки експертних систем.

3.5 Взаємодія людей-розробників при створенні ЕС.

3.6 Труднощі при розробці експертних систем.

3.7 Методологія побудови експертних систем. Приклади.

Час: 4 години.

Література: [1-23]

3.1 Методи створення експертних систем, які засновані на правилах.

Розглянемо докладно базові принципи та методи за допомогою яких проводиться проектування всіх експертних систем.

У всіх експертних системах існує залежність між вхідним потоком даних і даними в базі знань. Під час консультації вхідні дані зіставляються з даними в базі знань. Результатом зіставлення є негативна або позитивна відповідь. У системі, що базується на правилах ствердний результат є дією одного з продукційних правил.

Ці продукційні правила визначаються вхідними даними. Таким чином, експертна система, що базується на правилах (наприклад у мові Пролог) містить множину правил, які викликаються за допомогою вхідних даних у момент зіставлення. Експертна система також містить інтерпретатор в механізмі виведення, який вибирає і активізує різні модулі системи.

Роботу такого інтерпретатора можна описати послідовністю з 3 кроків:

Крок 1. Інтерпретатор зіставляє зразок правила з елементами даних в базі знань.

Крок 2. Якщо можна викликати більш ніж одне правило, то інтерпретатор використовує механізм вирішення конфлікту для вибору потрібного правила.

Крок 3. Інтерпретатор застосовує вибране правило, щоб знайти відповідь на питання.

Цей трьохкроковий процес інтерпретації є циклічним і називається циклом „розпізнавання-дія”. У системі, яка базується на правилах, кількість продукційних правил визначає розмір бази знань. Деякі найбільш складні системи мають бази знань з більш ніж 5000 продукційних правил.

При проектуванні таких ЕС існують певні вимоги, а саме:

- використовувати мінімальну достатню безліч умов при визначенні продукційного правила;
- уникати продукційних правил, які суперечать один одному;
- конструювати правила, спираючись на структуру властиву наочній області.

3.2 Методи створення експертних систем, які базуються на логіці.

У експертних системах, що базуються на логіці, база знань складається з тверджень у вигляді пропозицій логіки предикатів. Такі пропозиції можуть групуватися, утворюючи базу даних (наприклад у мові Пролог). Правила можуть або описувати дані або керують процесом внутрішньої уніфікації Прологу.

Так само як і в системі на правилах експертна система, що базується на логіці, має множину правил, які можуть викликатися за допомогою даних з вхідного потоку. Система має також інтерпретатор, який може вибирати і активізувати модулі, що включаються в роботу системи.

Інтерпретатор виконує певні функції у системі.

А саме він робить наступне:

Крок 1. Система має пропозиції в базі знань, які управляють пошуком і зіставленням. Інтерпретатор зіставляє ці пропозиції з елементами даних в базі даних.

Крок 2. Якщо може бути викликано більш за одне правило, то система використовує можливості системи (наприклад мови Пролог) для вирішення конфлікту. Отже користувачу або програмісту не потрібно розглядати потенційно можливі конфлікти.

Крок 3. Система отримує результати процесу уніфікації автоматично, тому вони можуть зразу направлятися на потрібний пристрій виведення інформації (наприклад у файл, на друк, у вікно виведення, тощо).

Так само як і в системі, що базується на правилах, даний циклічний процес є процесом „розпізнавання-дія”. Великі можливості системи, яка заснована на логіці, полягають в тому, що вона відображає структуру системи. Цим пояснюється тим фактом, що дана система дуже ефективна в роботі та користуванні. Найбільш важливим аспектом для бази знань в такій системі, яка заснована на логіці, є проектування бази знань, її тверджень і їх структури. База знань повинна мати недвозначну логічну організацію, і вона повинна містити мінімум надмірної інформації. Так само як і в системі, що базується на правилах, мінімальна достатня кількість даних утворюють найбільш ефективну систему.

3.3 Принципи та базові етапи створення ЕС.

Розглянемо докладно принципи та базові етапи створення ЕС.

Етап 1. Етап ідентифікації: ідентифікується завдання, визначаються учасники проектування, ресурси, цілі.

Етап 2. Етап концептуалізації: визначення ключових понять, відносин, характеристик, необхідних для опису процесу рішення задачі. А саме: типів даних; початкових даних, що виводяться; підзадач загального завдання; визначаються стратегії і гіпотези; визначаються види взаємозв'язків між об'єктами; визначаються типи відносин; визначаються процеси і типи обмежень; визначаються склад знань для вирішення проблеми; визначається склад знань для пояснення.

Етап 3. Етап формалізації: визначаються ключові поняття, які потім виражаються на деякій формальній мові; визначається, чи підходять ця система для вирішення вибраної проблеми.

Етап 4. Етап виконання: проводиться створення одного або декількох прототипів експертної системи, за наслідками етапів тестування і дослідної експлуатації – створюється закінчений програмний продукт.

Етап 5. Етап тестування: проводиться оцінка вибраного способу представлення знань і експертної системи в цілому; проводиться перевірка системи на прикладах; виявляються недоліки бази знань і механізму керування.

Етап 6. Етап дослідної експлуатації: перевірка придатності системи для кінцевого користувача та апробація зручності системи для користувача.

Етап 7. Модифікація системи: переформулювання понять і вимог; переконструювання уявлення та удосконалення прототипу.

Експертна система працює в двох режимах: в режимі придбання знань і в режимі рішення задачі (так званому режимі консультації або режимі використання ЕС).

У режимі придбання знань спілкування з ЕС здійснює (через посередництво інженера по знаннях) сам експерт. У цьому режимі експерт, використовуючи компонент придбання знань, наповнює систему знаннями, які дозволяють ЕС в режимі рішення самостійно (без експерта) вирішувати задачі з проблемної області. Експерт описує проблемну область у вигляді сукупності даних і правил. Дані визначають об'єкти, їх характеристики і значення, що існують в області експертизи. Правила визначають способи маніпулювання з даними, характерні для даної області.

Відзначимо, що режиму придбання знань в традиційному підході до розробки програм відповідають етапи алгоритмізації, програмування і відладки, що виконуються програмістом. Таким чином, розробку програм здійснює не програміст, а експерт (за допомогою ЕС), що не володіє програмуванням.

У режимі консультації спілкування з ЕС здійснює кінцевий користувач, якого цікавить результат і (або) спосіб його отримання. Необхідно відзначити, що залежно від призначення ЕС користувач може не бути фахівцем в даній проблемній області (в цьому випадку він звертається до ЕС тільки за результатом), або бути фахівцем (в цьому випадку користувач може

сам отримати результат, але він звертається до ЕС з метою або прискорити процес отримання результату, або покласти на ЕС рутинну роботу).

У режимі консультації дані про завдання користувача після обробки їх діалоговим компонентом поступають в робочу пам'ять. Вирішувач на основі вхідних даних з робочої пам'яті, загальних даних про проблемну область і правил з БЗ формує рішення задачі. ЕС при рішенні задачі не тільки виконує вказану послідовність операції, але і заздалегідь формує її.

Якщо реакція системи не зрозуміла користувачу, то він може вимагати її пояснення.

3.4 Етапи розробки експертних систем.

Розглянемо докладно основні етапи розробки експертних систем.

Розробка ЕС має істотні відмінності від розробки звичайного програмного продукту. Досвід створення ЕС показав, що використання при їх розробці методології, прийнятої в традиційному програмуванні, або надмірно затягує процес створення ЕС, або взагалі приводить до негативного результату.

Використовувати ЕС слід тільки тоді, коли розробка ЕС можлива, виправдана і методи інженерії знань відповідають вирішуваній задачі. Щоб розробка ЕС була можливою для даного застосування, необхідне одночасне виконання принаймні наступних вимог:

- існують експерти в даній області, які вирішують задачу значно краще, ніж фахівці, що починають;
- експерти сходяться в оцінці запропонованого рішення, інакше не можна буде оцінити якість розробленої ЕС;
- експерти здатні вербалізувати (виразити на природній мові) і пояснити використовувані ними методи;
- рішення задачі вимагає тільки міркувань, а не дій;
- завдання не повинне бути дуже важким, тобто її рішення повинне займати у експерта декілька годин або днів, а не тижнів;
- завдання повинне відноситися до достатньо „зрозумілої” і структурованої області, тобто повинні бути виділені основні поняття, відносини і відомі способи отримання рішення задачі;
- рішення задачі не повинне в значній мірі використовувати „здоровий глузд” (тобто широкий спектр загальних відомостей про світ і про спосіб його функціонування, які знає і уміє використовувати будь-яка людина), оскільки подібні знання поки не вдається у достатній кількості вкласти в інтелектуальні системи.

Використання ЕС в даному конкретному випадку може бути можливе, але не виправдано. Застосування ЕС може бути виправдане одним з наступних чинників:

- коли рішення задачі принесе значний ефект (економічний, тощо);

- коли використання людини-експерта неможливе або через недостатню кількість експертів, або із-за необхідності виконувати експертизу одночасно в різних місцях;

- коли при передачі інформації експерту відбувається неприпустима втрата часу або інформації;

- коли потрібно вирішувати задачу в оточенні, ворожому для людини.

Застосування відповідає методам ЕС, якщо вирішувана задача володіє сукупністю наступних характеристик:

- задача може бути природним чином вирішена за допомогою маніпуляції з символами (тобто за допомогою символічних міркувань), а не маніпуляцій з числами, як прийнято в математичних методах і в традиційному програмуванні;

- завдання повинне мати евристичну, а не алгоритмічну природу, тобто її рішення повинне вимагати застосування евристичних правил. Завдання, які можуть бути гарантовано вирішені (з дотриманням заданих обмежень) за допомогою деяких формальних процедур, не підходять для застосування ЕС;

- завдання повинне бути достатньо складним, щоб виправдати витрати на розробку ЕС. Проте вона не повинна бути надмірно складною, щоб ЕС змогла її вирішувати;

- завдання повинне бути достатньо вузьким, щоб вирішуватися методами ЕС, а також бути і практично значущою.

При розробці ЕС, як правило, використовується концепція „швидкого прототипу”. Суть цієї концепції полягає в тому, що розробники не намагаються відразу побудувати кінцевий продукт. На початковому етапі вони створюють прототип (прототипи) ЕС. Прототипи повинні задовольняти двом суперечливим вимогам: з одного боку, вони повинні вирішувати типові задачі конкретного додатку, а з іншої - час і трудомісткість їх розробки повинні бути незначні, щоб можна було максимально запаралелювати процес накопичення і відладки знань (здійснюваний експертом) з процесом вибирання (розробки) програмних засобів (які здійснюються інженером по знаннях і програмістом). Для задоволення вказаним вимогам, як правило, при створенні прототипу використовуються різноманітні засоби, прискорюючі сам процес проектування.

Прототип ЕС повинен продемонструвати придатність методів інженерії знань для даного додатку, який розробляється. У разі успіху експерт за допомогою інженера по знаннях розширює знання прототипу про проблемну область. При невдачі потрібна розробка нового прототипу або розробники можуть прийти до висновків про непридатність методів ЕС для даного додатку. При збільшенні знань прототип може досягти такого стану, коли він успішно вирішує всі задачі даного додатку. Перетворення прототипу ЕС в кінцевий продукт зазвичай приводить до перепрограмування ЕС на мові низького рівня, що забезпечують як збільшення швидкодії ЕС, так і зменшення необхідної пам'яті. Трудомісткість і час створення ЕС в значній мірі залежать від типу використовуваного інструментарію.

В ході робіт із створення ЕС склалася певна технологія їх розробки, що включає шість наступних етапів, а саме: ідентифікацію, концептуалізацію, формалізацію, виконання, тестування та досліду експлуатацію.

На етапі ідентифікації визначаються завдання, які підлягають рішенню, виявляються цілі розробки, визначаються експерти і типи користувачів.

На етапі концептуалізації проводиться змістовний аналіз проблемної області, виявляються використовувані поняття і їх взаємозв'язки, визначаються методи рішення задач.

На етапі формалізації вибираються потрібні інтелектуальні системи і визначаються способи представлення всіх видів знань, формалізуються основні поняття, визначаються способи інтерпретації знань, моделюється робота системи, оцінюється адекватність цілям системи, методів рішень, засобів уявлення і маніпулювання знаннями.

На етапі виконання здійснюється наповнення експертом бази знань. У зв'язку з тим, що основою ЕС є знання, даний етап є найбільш важливим і найбільш трудомістким етапом розробки ЕС. Процес придбання знань розділяють на отримання знань від експерта, організацію знань, що забезпечує ефективну роботу системи, і представлення знань у вигляді, зрозумілому ЕС. Процес придбання знань здійснюється інженером по знаннях на основі аналізу діяльності експерта за рішенням реальних завдань.

3.5 Взаємодія людей-розробників при створенні ЕС.

Зазвичай у розробці ЕС беруть участь представники наступних спеціальностей:

- експерт в проблемній області, задачі якої вирішуватиме ЕС;
- інженер по знаннях - фахівець з розробки ЕС. Використовувані їм технологію, методи називають технологією або методами інженерії знань;
- програміст по розробці інструментальних засобів (ІЗ), призначених для прискорення розробки ЕС.

Експерт визначає знання (дані і правила), що характеризують проблемну область, забезпечує повноту і правильність введених в ЕС знань.

Інженер по знаннях допомагає експерту виявити і структурувати знання, необхідні для роботи ЕС; здійснює вибір ІЗ, які найбільш підходять для даної проблемної області, і визначає спосіб представлення знань в ІЗ; виділяє і програмує (традиційними засобами) стандартні функції (типові для даної проблемної області), які використовуватимуться в правилах, що вводяться експертом.

Програміст розробляє ІЗ, що містять всі основні компоненти ЕС, і здійснює їх сполучення з тим середовищем, в якому вони будуть використані.

Процес взаємодії інженера по знаннях з експертом-фахівцем включає 3 основні етапи.

Етап 1. Підготовчий. Для успіху рід час спілкування обидва учасники повинні ретельно підготуватися до діалогу. Бажано, щоб експерт був не

тільки компетентним фахівцем, але і зацікавленою (морально або матеріально) особою в досягненні кінцевій мети - побудові системи.

Він повинен вміти добре пояснювати свої знання, найкраще коли експерт має досвід викладацької роботи. Аналітику необхідно: глибоко познайомитися із спеціальною літературою по предметній області, щоб не ставити незручних питань; вміти слухати і грамотно ставити питання; розбиратися в моделях когнітивної психології, а також в моделях представлення знань, щоб із знань експерта виділяти чіткі структури.

Приналежність науковця до того або іншого типу визначається за допомогою непрямих методик (тестів особи, інтелекту, когнітивних стилів, проектних методик).

Етап 2. Встановлення „загального коду”. Для створення лінгвістичного альянсу взаємодії учасники взаємодії повинні намагатися скоротити відстань між об'єктом (тобто досліджуваною наочною областю) і аналітиком. Необхідно визначити головні поняття, тобто виробити словарну основу бази знань; рівень деталізації та взаємозв'язки між поняттями.

Етап 3. Гносеологічний. На цьому етапі відбувається з'ясування закономірностей, властивих наочній області, умов достовірності і істинності тверджень, структуризація за рахунок введення відносин і т.ін. Цей етап є таким, що визначає у взаємодії аналітика і експерта. В процесі аналізу або діалогу вербалізуються і формалізуються знання експерта і часто для нього самого породжується нове знання. В процесі виявлення знань спочатку бажано отримати від експерта поверхневі знання, такі наприклад, як репрезентація ознак, поступово переходячи до глибинних структур і абстрактніших понять (такі, наприклад, як прототипи). При формуванні поля знань враховуються особливості емпіричного знання: модальність, суперечність, неповнота і т.ін. Аналітик винен за приватним завжди бачити загальне, тобто будувати систему по наступному алгоритму „факт - узагальнений факт - емпіричний закон - теоретичний закон”. Мистецтво аналітика полягає в прагненні до створення ясної і зрозумілої моделі проблемної області.

Розглянемо докладно підготовчий етап роботи із знаннями.

Етап 1. Чітке визначення завдань проектованої системи (звуження поля знань): визначення, що є на вході і виході системи; визначення режиму робіт, консультації, навчання.

Етап 2. Вибір експертів: визначення кількості експертів; вибір рівня компетентності; визначення способів зацікавити експертів в роботі; тестування експертів.

Етап 3. Знайомство аналітика із спеціальною літературою в проблемній області.

Етап 4. Знайомство аналітика і експертів (експерта).

Етап 5. Знайомство експерта з відповідною літературою з обраної теми.

Етап 6. Аналітик створює поле знань першого наближення по апріорним знанням з літератури, це так званий прототип поля знань.

Розглянемо докладно основний етап роботи із знаннями.

Етап 1. Накопичення поля знань: залежно від предметної області порводиться вибір способу інтерв'ювання; проводиться протоколювання думок вголос або записом на магнітофон міркувань експерта (без участі аналітика, поки що).

Етап 2. „Домашня робота”. Спроба аналітика виділити деякі причинно-наслідкові зв'язки в міркуваннях експерта; побудова словника наочної області (можливо, на картках) і підготовка питань до експерта.

Етап 3. Накопичення знань для поля зору системи. Обговорення з експертом прототипу поля знань і „домашньої роботи” а також відповіді на питання аналітика.

Етап 4. Формалізація концептуальної моделі.

Етап 5. Побудова поля знань другого наближення.

Перелічимо основні етапи реалізації системи придбання знань.

Етап 1. Інтерв'ю для визначення актуальної області, в якій відбувається процес вирішення проблеми, і розчленовування її на автономні області.

Етап 2. Автоматизоване інтерв'ю для виявлення і формування декларативної моделі предметної області.

Етап 3. Протокольний аналіз до виявлених на попередньому етапі понять і відносин предметної області для поповнення моделі знаннями.

Етапи 2 і 3 можна використовувати поперемінно до тих пір, поки модель не досягне потрібної повноти (знань та даних).

Етап 4. Протокольний аналіз для поповнення декларативних знань моделі.

Етап 5. Перевірка повноти моделі. Протокольний аналіз виявляє порожнечі в моделі. Є випадки коли поняття, використані в „думках вголос”, недостатньо описані. В цьому випадку інтерв'ю і протокольний аналіз повторюються.

Вже розроблена система функціонує в наступному циклічному режимі:

Крок 1. Діалоговий компонент запрошує дані або результатів аналізів, спостережень (цей етап може бути реалізований у вигляді системи питань до користувача) і поміщає їх в робочу пам'ять.

Крок 2. Вирішувач інтерпретує результати за допомогою правил, витягнутих з бази знань.

Крок 3. У разі браку інформації для остаточного вирішення процес продовжується до тих пір, поки не поступить достатня кількість інформації.

У будь-який момент часу в системі існують три типи знань: статичні знання про наочну область (після того, як ці знання виявлені, вони вже не змінюються); динамічні знання про наочну область (вони оновлюються у міру виявлення нової інформації); робочі знання — це знання, які використовуються для вирішення конкретного завдання або проведення консультації.

Всі перераховані вище знання зберігаються в базі знань. Для її побудови потрібно провести опит фахівців, що є експертами в конкретній наочній області, а потім систематизувати, організувати і забезпечити ці

знання показчиками, щоб згодом їх можна було легко витягнути з бази знань.

Відповідно до загальної схеми статичної експертної системи для її функціонування потрібні наступні знання:

- знання про процес рішення завдання, використовуваний інтерпретатором (вирішувачем);
- знання про мову спілкування і способи організації діалогу, використовуваний лінгвістичним процесором (діалоговим компонентом);
- знання про способи уявлення і модифікації знань, використовуваний компонентом придбання знань;
- структурні знання та знання, які керують, використовуваний пояснювальним компонентом.

3.6 Труднощі при розробці експертних систем.

Розробка ЕС пов'язана з певними труднощами, які необхідно добре знати, так само як і способи їх подолання. Розглянемо докладніше ці проблеми.

1. Проблема витягу знань експертів. Жоден фахівець ніколи просто так не розкриє секрети своєї професійної майстерності, свої таємні знання в професійній області. Він повинен бути зацікавлений матеріально або морально, причому добре зацікавлений. Ніхто не прагне рубати сук, на якому сидить. Часто такий фахівець побоюється, що, розкривши всі свої секрети, він буде не потрібний компанії. Замість нього буде працювати експертна система. Уникнути цю проблему допоможе вибір висококваліфікованого експерта, зацікавленого в співробітництві.

2. Проблема формалізації знань експертів. Експерти-Фахівці в певній області, як правило, не в змозі формалізувати свої знання. Часто вони ухвалюють правильні розв'язки на інтуїтивному рівні й не можуть аргументовано пояснити, чому прийняте те або інший розв'язок. Іноді експерти не можуть прийти до взаєморозуміння (фраза «зустрілися два геологи, у них було три думки» - не жарт, а реальне життя). У таких ситуаціях допоможе вибір експерта, що вміє ясно формулювати свої думки й легко пояснювати іншим свої ідеї.

3. Проблема нестачі часу в експерта. Обраний для розробки експерт не може знайти досить часу для виконання проекту. Він занадто зайнятий. Він усім потрібний. У нього є проблеми. Щоб уникнути цієї ситуації, необхідно одержати від експерта, перш ніж почнеться проект, згода витрачати на проект час у певному фіксованому обсязі.

4. Правила, формалізовані експертом, не дають необхідної точності. Проблему можна уникнути, якщо вирішувати разом з експертом реальні завдання. Не треба придумувати «іграшкових» ситуацій або завдань. В умовах завдань потрібно використовувати реальні дані, такі як лабораторні дані, звіти, щоденники й іншу інформацію, узятую із практичних завдань. Постарайтеся говорити з експертом на одній мові, використовуючи єдину

термінологію. Експерт, як правило, легше розуміє правила, записані мовою, близькому до природнього, а не мовою типу LISP або PROLOG.

5. Недолік ресурсів. У якості ресурсів виступають персонал (інженери знань, розроблювачі інструментальних засобів, експерти) і засобу побудови ЕС (засобу розробки й засобу підтримки). Недолік доброзичливих і грамотних адміністраторів породжує скептицизм і нетерпіння в керівників. Підвищена увага в пресі й перебільшення викликали нереалістичні очікування, які приводять до розчарування відносно експертних систем. ЕС можуть давати не найкращі розв'язки на границі їх застосовності, при роботі із суперечливими знаннями й у міркуваннях на основі здорового глузду. Можуть знадобитися значні зусилля, щоб добитися невеликого збільшення якості роботи ЕС. Експертні системи вимагають багато часу на розробку. Так, створення системи PUFF для інтерпретації функціональних тестів легенів зажадало 5 людино-лет, на розробку системи PROSPECTOR для розвідки рудних родовищ пішло 30 людино-літ, система XCON для розрахунків конфігурації комп'ютерних систем на основі VAX 11/780 зажадала 8 людино-лет. ЕС останніх років розробляються більш швидкими темпами за рахунок розвитку технологій ЕС, але проблеми залишилися. Подвоєння персоналу не скорочує час розробки наполовину, тому що процес створення ЕС - це процес із безліччю зворотних зв'язків. Усе це необхідно враховувати при плануванні створення ЕС.

6. Неадекватність інструментальних засобів розв'язуваному завданню. Часто певні типи знань (наприклад, тимчасові або просторові) не можуть бути легко представлені на одному ЯПЗ, так само як і різні схеми вистави (наприклад, фрейми й продукції) не можуть бути досить ефективно реалізовані на одному ЯПЗ. Деякі завдання можуть бути непридатними для розв'язку за технологією ЕС (наприклад, окремі завдання аналізу сцен). Необхідний ретельний аналіз розв'язуваних завдань, щоб визначити придатність пропонованих інструментальних засобів і зробити правильний вибір.

3.7 Методологія побудови експертних систем. Приклади

Розглянемо методику формалізації експертних знань на прикладі створення експертних діагностичних систем (ЕДС).

Метою створення ЕДС є визначення стану об'єкта діагностування (ОД) і наявних у ньому несправностей.

Станами ОД можуть бути: справно, несправне, працездатне. Несправностями, наприклад, радіоелектронних ОД є обрив зв'язку, замикання провідників, неправильне функціонування елементів і т.д.

Число несправностей може бути досить велике (кілька тисяч). В ОД може бути одночасно кілька несправностей. У цьому випадку говорять, що несправності кратні.

Уведемо наступні визначення. Різні несправності ОД проявляються в зовнішньому середовищі інформаційними параметрами.

Сукупність значень інформаційних параметрів визначає «інформаційний образ» (ІО) несправності ОД. ІО може бути повним, тобто містити всю необхідну інформацію для постановки діагнозу, або, відповідно, неповним. У випадку неповного ІО постановка діагнозу носить імовірнісний характер.

Основою для побудови ефективних ЕДС є знання експерта для постановки діагнозу, записані у вигляді інформаційних образів, і система вистави знань, що вбудовується в інформаційні системи забезпечення функціонування й контролю ОД, інтегровані з відповідної технічною апаратурою.

Для опису своїх знань експерт за допомогою інженера по знаннях повинен виконати наступне.

1. Виділити безліч усіх несправностей ОД, які повинна розрізняти ЕДС.
2. Виділити безліч інформативних (істотних) параметрів, значення яких дозволяють розрізнити кожну несправність ОД і поставити діагноз із деякою ймовірністю.

3. Для обраних параметрів слід виділити інформативні значення або інформативні діапазони значень, які можуть бути як кількісними, так і якісними. Наприклад, точні кількісні значення можуть бути записані: затримка 25 нсек, затримка 30 нсек і т.д. Кількісний діапазон значень може бути записаний: затримка 25-40 нсек, 40-50 нсек, 50 нсек і вище. Якісний діапазон значень може бути записаний: індикаторна лампа світиться яскраво, світиться слабо, не світиться.

Для більш зручного подальшого використання якісний діапазон значень може бути закодований, наприклад, у такий спосіб:

- світиться яскраво P1 = +++ (або P1 = 3),
- світиться слабо P1 = ++ (або P1 = 2),
- не світиться P1 = + (або P1 = 1).

Процедура одержання інформації з кожного з параметрів визначається індивідуально в кожній конкретній системі діагностування. Ця процедура може полягати в автоматичному вимірі параметрів в ЕДС, у ручному вимірі параметра за допомогою приладів, якісним визначенні параметра, наприклад, світиться слабо, і т.д.

4. Процедура створення повних або неповних ІО кожної несправності в алфавіті значень інформаційних параметрів може бути визначена в такий спосіб. Складаються діагностичні правила, що визначають імовірний діагноз на основі різних комбінацій діапазонів значень обраних параметрів ОД. Правила можуть бути записані в різній формі. Нижче наведена форма запису правил у вигляді таблиці 1.

Таблиця 1 – Форма запису правил

Номер	P1	P2	P3	Діагноз	Імовірність діагнозу	Примітки
1		+++		Несправний блок А1	0.95	
2	12-15	+		Несправний блок А2	0.80	

Для запису правил з урахуванням змін за часом слід увести ще один параметр P0 - час (ще один стовпець у таблиці). У цьому випадку діагноз може ставитися на основі декількох рядків таблиці, а в графові Примітки можуть бути зазначені використані тести. Діагностична таблиця в цьому випадках представлено в таблиці 2.

Таблиця 2 - Динамічні діагностичні правила

Номер	P0	P1	P2	P3	Діагноз	Імовірність діагнозу	Примітки
1	12:00	+	+	+			тест T1
2	12:15	++	++	+	Несправний блок A3	0.90	

Для записів послідовності проведення тестових процедур і завдання обмежень (якщо вони є) на їхнє проведення може бути запропонований аналогічний механізм. Механізм записів послідовності проведення тестових процедур у вигляді правил реалізується, наприклад, у такий спосіб:

ЯКЩО: P2 = 1

ТЕ: тест = T1, T3, T7

де T1, T3, T7 - тестові процедури, що подається на ОД при активізації (спрацьовуванні) відповідної продукції.

У сучасних ЕДС застосовуються різні стратегії пошуку розв'язку й постановки діагнозу, які дозволяють визначити необхідні послідовності тестових процедур. Однак пріоритет в ЕС віддається насамперед знанням і досвіду, а лише потім логічному висновку.

Дана методика буде застосована в наступній лекції при створенні експертної системи керування технологічним процесом.

Розглянемо експертну систему діагностування (ЕСД) цифрових і цифроаналогових обладнань, у якій використовувалися системи продукцій і фрейми, а також пряма й зворотний ланцюжок міркувань одночасно. У якості об'єкта діагностування (ОД) в ЕСД можуть використовуватися цифрові обладнання (ЦУ), БІС, цифро-аналогові обладнання. На рисунку нижче показане, що така ЕСД працює разом з автоматизованою системою контролю й діагностування (АКД), яка подає в динаміку впливу на ОД (десятки, сотні й тисячі впливів у секунду), аналізує вихідні реакції й дає висновок: придатний або не придатний. У випадках, якщо реакція, що перевіряється ОД не відповідає еталонним значенням, то підключається заснована на знаннях підсистема діагностування. ЕСД запитує значення сигналів у певних контрольних крапках і веде оператора за схемою ОД, рекомендуючи йому зробити виміри в певних контрольних крапках або підтвердити проміжний діагноз, і в результаті приводить його до місця несправності. Вихідними даними для роботи ЕСД є результати машинного моделювання ОД на етапі проектування. Ці результати моделювання передаються в ЕСД на магнітних носіях у вигляді тисяч продукційних правил. Рух по контрольних крапках здійснюється на основі моделі, записаної у вигляді мережі фреймів для ОД.

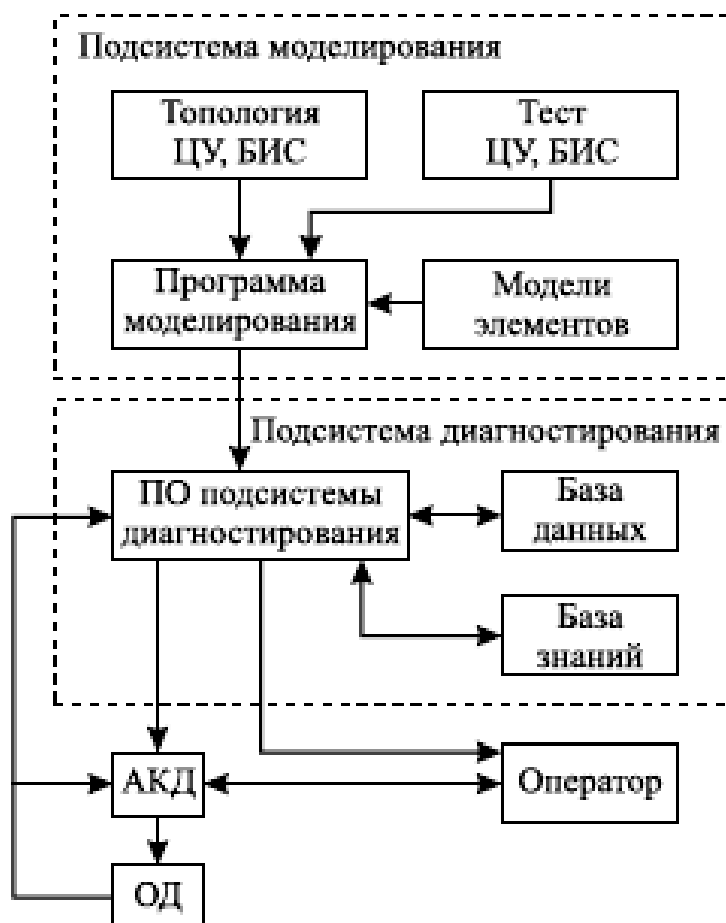


Рисунок - Загальна структура експертної системи діагностування

Така ЕСД не була б інтелектуальною системою, якби вона не накопичувала досвід. Вона запам'ятовує знайдену несправність для даного типу ОД. Наступного разу при діагностиці несправності ОД цього типу вона пропонує перевірити відразу ж цю несправність, якщо реакція ОД говорить про те, що така несправність можлива. Так надходять досвідчені майстри радіоелектронної апаратури (РЭА), що знають «слабкі» місця в конкретних типах РЭА, що й перевіряють їх у першу чергу. ЕСД накопичує імовірнісні знання про конкретні несправності з метою їх використання при логічному висновку. При русі по дереву пошуку розв'язків на черговому кроці використовується критерій - максимум відносини ймовірності (коефіцієнта впевненості) постановки діагнозу до трудомісткості розпізнавання несправності. Коефіцієнти впевненості автоматично коректуються під час роботи ЕСД при кожнім підтвердженні або не підтвердженні діагнозу для конкретних ситуацій діагностування. Трудомісткості елементарних перевірок спочатку задаються експертом, а потім автоматично коректуються в процесі роботи ЕСД.

ЕСД не була реалізована у вигляді ИПС із економічних міркувань. Невелика серійність апаратури, що перевіряється, недостатня уніфікація й дешева робоча сила (останній фактор і в наш час відіграє в Росії немаловажну роль) перешкодили реалізувати автоматичне діагностування.

Серед сучасних комерційних систем хочеться виділити експертну систему - оболонку G2 американської фірми Gensym (USA) як неперевершену експертну комерційну систему для роботи з динамічними об'єктами. Робота в реальному часі з малими часами відповіді часто необхідна при аналізі ситуацій у корпоративних інформаційних мережах, на атомних реакторах, у космічних польотах і безлічі інших завдань. У цих завданнях необхідно ухвалювати розв'язки протягом мільсекунд із моменту виникнення критичної ситуації. ЕС G2, призначена для розв'язку таких завдань, відрізняється від більшості динамічних ЕС такими характерними властивостями, як:

- робота в реальному часі з розпаралелювальним процесом міркувань;
- структурований природно-языковий інтерфейс із керуванням по меню й автоматичною перевіркою синтаксису;
- зворотний і прямий висновок, використання метазнаний, сканування й фокусування;
- інтеграція підсистеми моделювання з динамічними моделями для різних класів об'єктів;
- структурування БЗ, спадкування властивостей, розуміння зв'язків між об'єктами;
- бібліотеки знань є Ascii-Файлами й легко переносяться на будь-які платформи й типи ЕОМ;
- розвитий редактор для супроводу БЗ без програмування, засобу трасування й налагодження БЗ;
- керування доступом за допомогою механізмів авторизації користувача й забезпечення бажаного погляду на додаток;
- гнучкий інтерфейс оператора, що включає графіки, діаграми, кнопки, піктограми й т.п.;
- інтеграція з іншими додатками (по TCP/IP) і базами даних, можливість вилученої й багатокористувацької роботи.