

## Лекція 5. ЕКСПЕРТНІ СИСТЕМИ

- призначення та принципи побудови;
- класи задач, які вирішуються за допомогою експертних систем;
- узагальнена архітектура;
- етапи розробки;
- придбання знань;
- пошук та пояснення рішень;

### 1. Призначення експертних систем та класи задач, які вирішуються за допомогою експертних систем

На початку вісімдесятих років у дослідженнях зі штучного інтелекту сформувався самостійний напрямок, що одержав назву "експертні системи" (ЕС). Ціль досліджень ЕС полягає в розробці програм, які при вирішенні задач, важких для експерта-людини, отримують результати, що не уступають з якості й ефективності рішенням, які б отримав експерт. Дослідники в області ЕС для назви своєї дисципліни часто використовують також термін "інженерія знань", уведений Е.Фейгенбаумом як "привнесення принципів й інструментарію досліджень із області штучного інтелекту у вирішення важких прикладних проблем, що вимагають знань експертів".

Програмні засоби (ПЗ), що базуються на технології експертних систем, або інженерії знань (надалі будемо використати їх як синоніми), одержали значне поширення у світі. **Важливість** експертних систем полягає в наступному:

- технологія експертних систем істотно розширює коло практичних задач, що розв'язуються з допомогою комп'ютерів, вирішення яких приносить значний економічний ефект;
- технологія ЕС є найважливішим засобом у вирішенні глобальних проблем традиційного програмування: тривалість й, отже, висока вартість розробки складних додатків;
- висока вартість супроводу складних систем, що часто в кілька разів перевершує вартість їхньої розробки; низький рівень повторного використання програм і т.п.;
- об'єднання технології ЕС із технологією традиційного програмування додає нові якості до програмних продуктів за рахунок: забезпечення динамічної модифікації додатків користувачем, а не програмістом; більшої "прозорості" додатка (наприклад, знання зберігаються на обмеженій природній мові (ПМ), що не вимагає коментарів до знань, спрощує навчання й супровід); кращої графіки; інтерфейсу й взаємодії.

На думку провідних спеціалістів, у недалекій перспективі ЕС знайдуть наступне **застосування**:

- ЕС будуть відігравати провідну роль у всіх фазах проектування, розробки, виробництва, розподілу, продажу, підтримки й надання послуг;
- технологія ЕС, що одержала комерційне поширення, забезпечить революційний прорив в інтеграції додатків з готових інтелектуально-взаємодіючих модулів.

ЕС призначені для так званих неформалізованих задач, тобто ЕС не відкидають і не заміняють традиційного підходу до розробки програм, орієнтованого на рішення формалізованих задач.

**Неформалізовані задачі** звичайно мають наступні особливості:

- помилковістю, неоднозначністю, неповнотою й суперечливістю вихідних даних;
- помилковістю, неоднозначністю, неповнотою й суперечливістю знань про проблемну область і розв'язувану задачу;
- великою розмірністю простору рішення, тобто перебір при пошуку рішення досить великий;
- динамікою, що передбачає зміну даних і знань.

Варто підкреслити, що неформалізовані задачі представляють великий і дуже важливий клас задач.

**Експертні системи й системи штучного інтелекту відрізняються від систем обробки даних тим, що в них, в основному, використовується символний (а не числовий) спосіб подання, символний висновок й евристичний пошук рішення (а не виконання відомого алгоритму).**

Експертні системи застосовуються для вирішення тільки важких практичних (не іграшкових) задач. По якості й ефективності вирішення експертні системи не уступають рішенням експерта-людини. Рішення експертних систем володіють "прозорістю", тобто можуть бути пояснені користувачеві на якісному рівні. Ця властивість експертних систем забезпечується їхньою здатністю міркувати про свої знання й робити висновки. Експертні системи здатні поповнювати свої знання в ході взаємодії з експертом. Необхідно відзначити, що в цей час технологія експертних систем використовується для вирішення різних типів задач (інтерпретація, прогнозування, діагностика, планування, конструювання, контроль, налагодження, інструктаж, керування) у найрізноманітніших

проблемних областях, таких, як фінанси, нафтова й газова промисловість, енергетика, транспорт, фармацевтичне виробництво, космос, металургія, гірнича справа, хімія, целюлозно-паперова промисловість, телекомунікації й зв'язок й ін.

Варто звернути увагу на те, що деякі фахівці (як правило, фахівці в програмуванні, а не в ШІ) продовжують стверджувати, що ЕС і СШІ не виправдали очікувань, що покладали на них, і вмерли. Причини таких оман полягають у тому, що ці автори розглядали ЕС як альтернативу традиційному програмуванню, тобто вони виходили з того, що ЕС на самоті (в ізоляції від інших програмних засобів) повністю вирішують задачі, що стоять перед замовником. Треба відзначити, що на зорі появи ЕС специфіка використовуваних у них мов, технології розробки додатків і використовуваного устаткування (наприклад, Lisp-машини) давала підстави припускати, що інтеграція ЕС із традиційними, програмними системами є складною й, можливо, нездійсненною задачею при обмеженнях, що накладають реальними додатками. Однак у цей час комерційні інструментальні засоби (ІЗ) для створення ЕС розробляються в повній відповідності із сучасними технологічними тенденціями традиційного програмування, що знімає проблеми, що виникають при створенні інтегрованих додатків.

Причини, що привели СШІ до комерційного успіху.

**Інтегрованість.** Розроблено інструментальні засоби штучного інтелекту (ІЗ ШІ), що легко інтегруються з іншими інформаційними технологіями й засобами (з CASE, СУБД, контролерами, концентраторами даних і т.п.).

**Відкритість і переносимість.** ІЗ ШІ розробляються з дотриманням стандартів, що забезпечують відкритість і переносимість.

**Використання мов традиційного програмування й робочих станцій.** Перехід від ІЗ ШІ, реалізованих на мовах ШІ (Lisp, Prolog і т.п.), до ІЗ ШІ, реалізованих на мовах традиційного програмування (C, C++ і т.п.), спростив забезпечення інтегрованості, знизив вимоги додатків ШІ до швидкодії ЕОМ й об'єму оперативної пам'яті. Використання робочих станцій (замість ПК) різко збільшило коло додатків, які можуть бути виконані на ЕОМ з використанням ІЗ ШІ.

**Архітектура клієнт-сервер.** Розроблені ІЗ ШІ, що підтримують розподілені обчислення по архітектурі клієнт-сервер, що дозволило: знизити вартість устаткування, використовуваного в додатках, децентралізувати додатка, підвищити надійність і

загальну продуктивність (тому що скорочується кількість інформації, що пересилає між ЕОМ, і кожен модуль додатка виконується на адекватному йому встаткуванні).

**Проблемно/предметно-предметно-орієнтовані ІЗ ІІІ.** Перехід від розробок ІЗ ІІІ загального призначення (хоча вони не втратили своє значення як засіб для створення орієнтованих ІЗ) до проблемно/предметно-предметно-орієнтованих ІЗ ІІІ забезпечує: скорочення термінів розроблення додатків; збільшення ефективності використання ІЗ; спрощення й прискорення роботи експерта; повторне використання інформаційного й програмного забезпечення (об'єкти, класи, правила, процедури).

## 2. Структура експертних систем

Типова статична **ЕС складається** з наступних основних компонентів:

- вирішувача (інтерпретатора);
- робочої пам'яті (РП), названою також базою даних (БД);
- бази знань (БЗ);
- компонентів придбання знань;
- пояснювального компонента;
- діалогового компонента.

*База даних (робоча пам'ять)* призначена для зберігання вихідних і проміжних даних розв'язуваної в поточний момент задачі. Цей термін збігається за назвою, але не за змістом з терміном.

*База знань (БЗ)* в ЕС призначена для зберігання довгострокових даних, що описують розглянуту область (а не поточних даних), і правил, що описують доцільні перетворення даних цієї області.

*Вирішувач*, використовуючи вихідні дані з робочої пам'яті й знання із БЗ, формує таку послідовність правил, які, будучи застосованими до вихідних даних, приводять до вирішення задачі.

*Компонент придбання знань* автоматизує процес наповнення ЕС знаннями, здійснюваний користувачем-експертом.

*Пояснювальний компонент* пояснює, як система одержала рішення задачі (або чому вона не одержала рішення) і які знання вона при цьому використала, що полегшує експертові тестування системи й підвищує довіру користувача до отриманого результату.

*Діалоговий компонент* орієнтований на організацію дружнього спілкування з користувачем як у ході рішення задач, так й у процесі придбання знань і пояснення результатів роботи.

**У розробці ЕС беруть участь** представники наступних спеціальностей:

- *експерт у проблемній області*, задачі якої буде вирішувати ЕС;
- *інженер зі знань* – фахівець із розробки ЕС (використовувані їм технології, методи називають технологією (методами) інженерії знань);
- *програміст* з розробки інструментальних засобів (ІЗ), призначених для прискорення розробки ЕС.

Необхідно відзначити, що відсутність серед учасників розробки інженерів зі знань (тобто їхня заміна програмістами) або приводить до невдачі процес створення ЕС, або значно подовжує його.

**Експерт** визначає знання (дані й правила), що характеризують проблемну область, забезпечує повноту й правильність введених у ЕС знань.

**Інженер зі знань** допомагає експертові виявити й структурувати знання, необхідні для роботи ЕС; здійснює вибір того ІЗ, що найбільше підходить для даної проблемної області, і визначає спосіб подання знань у цьому ІЗ; виділяє й програмує (традиційними засобами) стандартні функції (типові для даної проблемної області), які будуть використовуватися в правилах, що вводяться експертом.

**Програміст** розробляє ІЗ, що містять всі основні компоненти ЕС, і здійснює його сполучення з тим середовищем, у якій воно буде використано.

**Експертна система працює у двох режимах:**

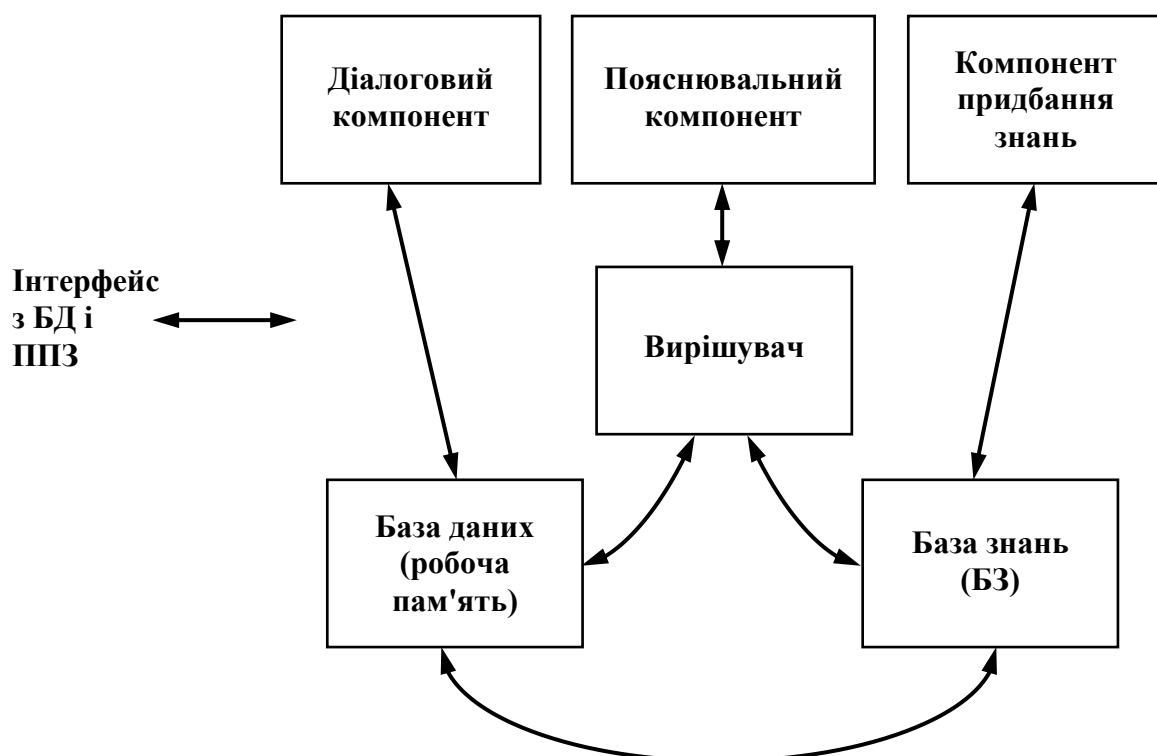
- режимі придбання знань
- режимі вирішення задачі (названому також режимом консультації або режимом використання ЕС).

У режимі придбання знань спілкування з ЕС здійснює (за посередництвом інженера зі знань) експерт. У цьому режимі експерт, використовуючи компонент придбання знань, наповнює систему знаннями, які дозволяють ЕС у режимі функціонування самостійно (без експерта) вирішувати задачі із проблемної області. Експерт описує проблемну область у вигляді сукупності даних і правил. Дані визначають об'єкти, їхні характеристики й значення, що існують в області експертизи. Правила визначають способи маніпулювання з даними, характерні для розглянутої області.

Відзначимо, що режиму придбання знань у традиційному підході до розробки програм відповідають етапи алгоритмізації, програмування й налагодження, виконувані програмістом. Таким чином, на відміну від традиційного підходу у випадку ЕС розробку програм здійснює не програміст, а експерт, що не володіє програмуванням.

У режимі консультації спілкування з ЕС здійснює кінцевий користувач, якого цікавить результат й (або) спосіб його одержання. Необхідно відзначити, що залежно від призначення ЕС користувач може не бути фахівцем у даній проблемній області (у цьому випадку він звертається до ЕС за результатом, не вміючи одержати його сам), або бути фахівцем (у цьому випадку користувач може сам одержати результат, але він звертається до ЕС із метою або прискорити процес одержання результату, або покласти на ЕС рутинну роботу). У режимі консультації дані про задачу користувача після обробки їхнім діалоговим компонентом надходять у робочу пам'ять. Вирішувач на основі вхідних даних з робочої пам'яті, загальних даних про проблемну область і правил із БЗ формує вирішення задачі. ЕС при вирішенні задачі не тільки виконує запропоновану послідовність операції, але й попередньо формує її. Якщо реакція системи не зрозуміла користувачеві, то він може зажадати пояснення:

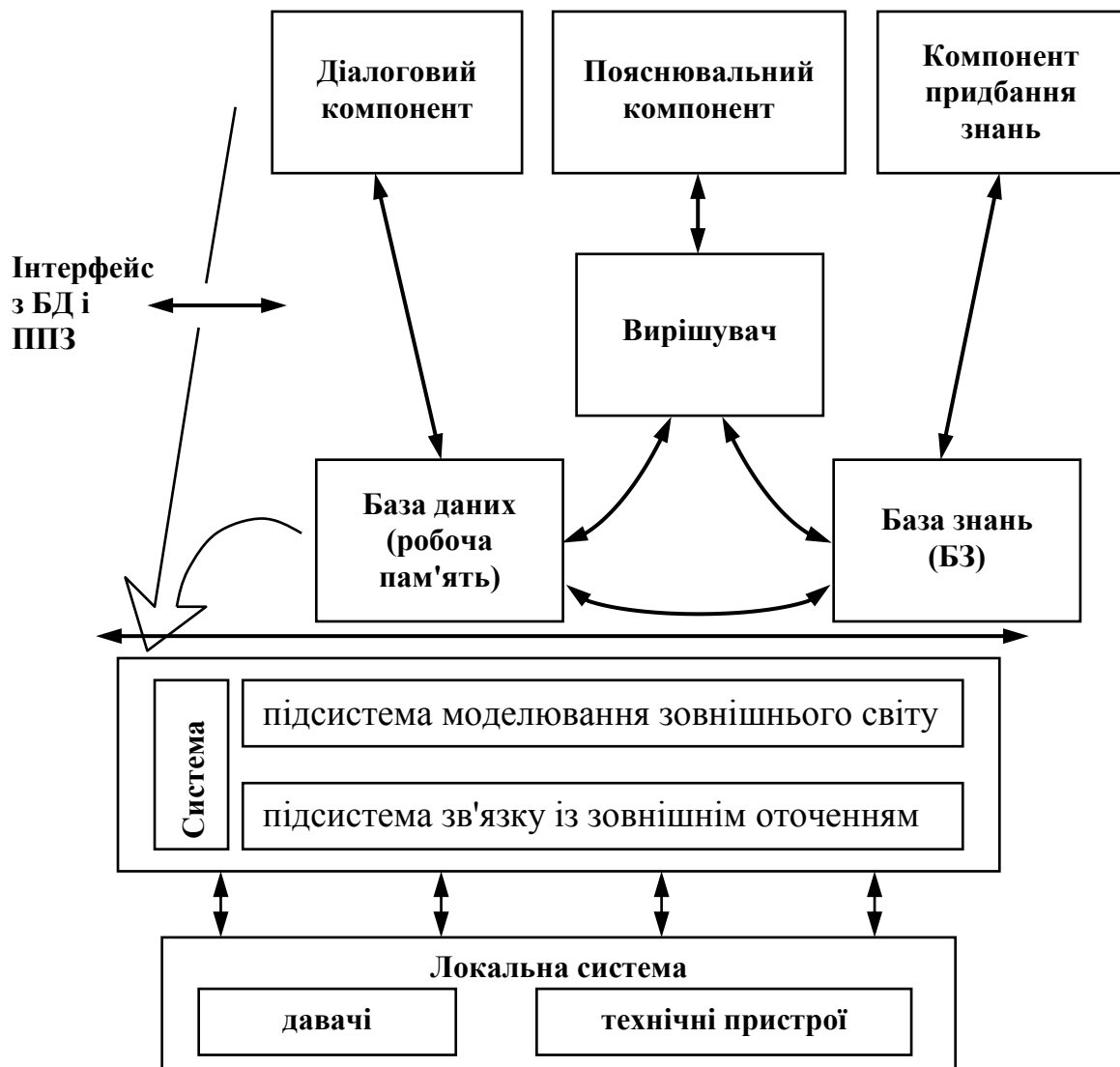
"Чому система задає те або інше питання?", "Як відповідь, що видає система, отримана?".



**Рис. 1. Структура статичної ЕС**

Структуру, наведену на рис. 1, називають структурою статичної ЕС. ЕС даного типу використовуються в тих додатках, де можна не враховувати зміни навколишнього світу, що відбуваються за час вирішення задачі. Перші ЕС, що одержали практичне використання, були статичними.

На рис. 2 показано, що в архітектуру динамічної ЕС у порівнянні зі статичної ЕС вводяться ще два компоненти: підсистема моделювання зовнішнього світу й підсистема зв'язку із зовнішнім оточенням. Остання здійснює зв'язок із зовнішнім світом через систему давачів і контролерів.



**Рис. 2. Структура динамічної ЕС**

Підкреслимо, що структура ЕС, представлена на рис. 1 й 2, відбиває тільки компоненти (функції), і багато чого залишається "за кадром". Узагальнена структура сучасного ІЗ для створення динамічних ЕС містить, крім основних компонентів, ті можливості, які дозволяють створювати інтегровані додатки відповідно до сучасної технології програмування.

### 3. Етапи розробки експертних систем

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

Використати ЕС треба тільки тоді, коли розробка ЕС можлива, виправдана й методи інженерії знань відповідають розв'язуваній задачі. Щоб розробка ЕС була можливою для даного додатка, необхідно **одночасне виконання** принаймні **наступних вимог**:

- існують експерти в даній області, які вирішують задачу значно краще, ніж починаючі фахівці;
- експерти сходяться в оцінці пропонованого рішення, інакше не можна буде оцінити якість розробленої ЕС;
- експерти здатні вербалізувати (виразити природною мовою) і пояснити використовувані ними методи, у протилежному випадку важко розраховувати на те, що знання експертів будуть "витягнуті" і вкладені в ЕС;
- вирішення задачі вимагає тільки міркувань, а не дій;
- задача не повинна бути занадто важкою (тобто її вирішення повинне займати в експерта кілька годин або днів, а не тижнів);
- задача хоча й не повинна бути виражена у формальному виді, але все-таки повинна ставитися до досить "зрозумілого" і структурованій області, тобто повинні бути виділені основні поняття, відносини й відомі (хоча б експертові) способи одержання вирішення задачі;
- вирішення задачі не повинне в значній мірі використати "здоровий глузд" (тобто широкий спектр загальних відомостей про світ і про спосіб його функціонування, які знає й уміє використати будь-яка нормальна людина), тому що подібні знання поки не вдається (у достатній кількості) вкласти в системи штучного інтелекту.

*Застосування ЕС також може бути виправдано одним з наступних факторів:*

- вирішення задачі принесе значний ефект, наприклад економічний;
- використання людини-експерта неможливо або через недостатню кількість експертів, або через необхідність виконувати експертизу одночасно в різних місцях;



- використання ЕС доцільно в тих випадках, коли при передачі інформації експертові відбувається неприпустима втрата часу або інформації;
- використання ЕС доцільно при необхідності вирішувати задачу в оточенні, ворожому для людини.

Додаток *відповідає методам ЕС*, якщо розв'язувана задача має сукупність наступних характеристик:

1. Задача може бути природно вирішена за допомогою маніпуляції із символами (тобто за допомогою символічних міркувань), а не маніпуляцій із числами, як прийнято в математичних методах й у традиційному програмуванні;
2. Задача повинна мати евристичну, а не алгоритмічну природу, тобто її вирішення повинне вимагати застосування евристичних правил. Задачі, які можуть бути гарантовано вирішені (з дотриманням заданих обмежень) за допомогою деяких формальних процедур, не підходять для застосування ЕС;
3. Задача повинна бути досить складна, щоб виправдати витрати на розробку ЕС. Однак вона не повинна бути надмірно складної (рішення займає в експерта години, а не тижня), щоб ЕС могла неї вирішувати;
4. Задача повинна бути досить вузькою, щоб вирішуватися методами ЕС, і практично значимою.

При розробці ЕС, як правило, використовується концепція "швидкого прототипу". Суть цієї концепції полягає в тому, що розроблювачі 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 додатка, а з іншого боку – час і трудомісткість їхньої розробки повинні бути досить незначні, щоб можна було максимально запаралелити процес нагромадження й налагодження знань (здійснюваний експертом) із процесом вибору (розробки) програмних засобів (здійснюваних інженером зі знань і програмістом). Для задоволення зазначеним вимогам, як правило, при створенні прототипу використовуються різноманітні засоби, що прискорюють процес проектування.

Прототип повинен продемонструвати придатність методів інженерії знань для даного додатка. У випадку успіху експерт за допомогою інженера зі знань розширює знання прототипу про проблемну область. При невдачі може знадобитися розробка нового про-

тотипу або розроблювачі можуть прийти до висновку про непридатність методів ЕС для даного додатка. У міру збільшення знань прототип може досягти такого стану, коли він успішно вирішує всі задачі даного додатка. Перетворення прототипу ЕС у кінцевий продукт звичайно приводить до перепрограмування ЕС на мовах низького рівня, що забезпечує як збільшення швидкодії ЕС, так і зменшення необхідної пам'яті.

У ході робіт зі створення ЕС зложилася певна технологія їхньої розробки, що включає шість наступних етапів (рис. 3): ідентифікацію, концептуалізацію, формалізацію, виконання, тестування, дослідну експлуатацію.

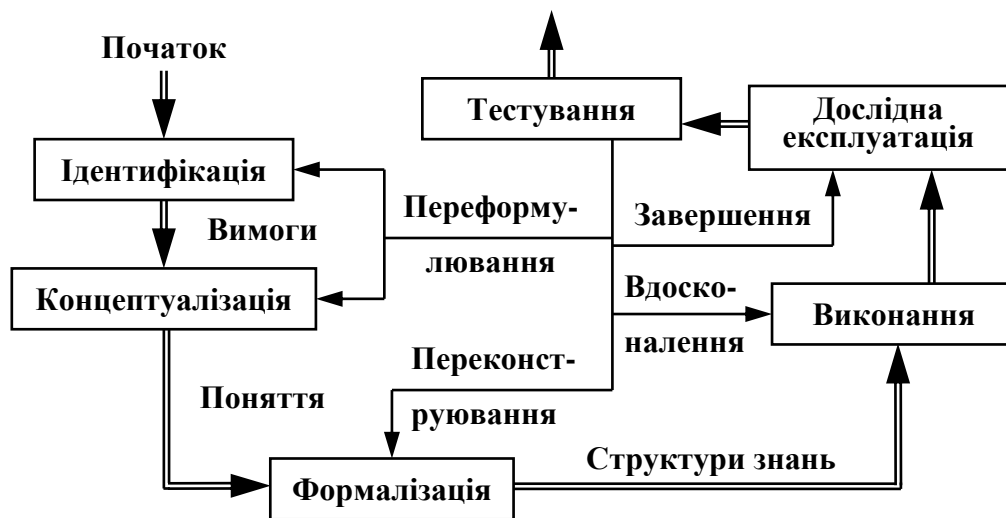


Рис. 3 Технологія розроблення ЕС

На етапі **ідентифікації** визначаються задачі, які підлягають рішенню, виявляються мети розробки, визначаються експерти й типи користувачів.

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

На етапі **формалізації** вибираються ІЗ і визначаються способи подання всіх видів знань, формалізуються основні поняття, визначаються способи інтерпретації знань, моделюється робота системи, оцінюється адекватність цілям системи зафіксованих понять, методів рішень, засобів подання й 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 здобування знань від експерта, організацію знань, що забезпечує ефективну роботу системи, і подання знань у вигляді, зрозумілому ЕС. Процес придбання знань здійснюється інженером зі знань на основі аналізу діяльності експерта за рішенням реальних задач.

## 4. Придбання знань

Прагнення виділити організацію знань у самостійну задачу викликано, зокрема, тим, що ця задача виникає для будь-якої мови подання й способи вирішення цієї задачі є однаковими (або подібними) поза залежністю від формалізму, який використовується.

Коло питань, які розв'язуються при поданні знань, включає:

- визначення складу знань, що представляють;
- організацію знань;
- подання знань, тобто визначення моделі подання.

Склад знань ЕС визначається наступними факторами:

- проблемним середовищем;
- архітектурою експертної системи;
- потребами й цілями користувачів;
- мовою спілкування.

Відповідно до загальної схеми статичної експертної системи (див. рис. 1) для її функціонування потрібні наступні знання:

- знання про процес вирішення задачі (тобто керуючі знання), що використовуються інтерпретатором (вирішувачем);
- знання про мову спілкування й способи організації діалогу, використовувані лінгвістичним процесором (діалоговим компонентом);
- знання про способи подання й модифікації знань, використовувані компонентом придбання знань;
- підтримуючі структурні й керуючі знання, використовувані пояснювальним компонентом.

Для динамічної ЕС, крім того, необхідні наступні знання:

1. знання про методи взаємодії із зовнішнім оточенням;
2. знання про моделі зовнішнього миру.

Залежність складу знань від вимог користувача проявляється в наступному:

- які задачі (із загального набору задач) і з якими даними хоче вирішувати користувач;
- які кращі способи й методи рішення;

- при яких обмеженнях на кількість результатів і способи їхнього одержання повинна бути вирішена задача;
- які вимоги до мови спілкування й організації діалогу;
- яка ступінь спільності (конкретності) знань про проблемну область, доступна користувачеві;
- які мети користувачів.

Склад знань про мову спілкування залежить як від мови спілкування, так і від необхідного рівня розуміння.

З урахуванням архітектури експертної системи знання доцільно ділити на такі, що інтерпретуються й не інтерпретуються. До першого типу ставляться ті знання, які здатні інтерпретувати вирішувач (інтерпретатор). Всі інші знання ставляться до другого типу. Вирішувач не знає їхні структури й змісти. Якщо ці знання використовуються яким-небудь компонентом системи, то він не "усвідомлює" цих знань. Знання, які не інтерпретуються, діляться на допоміжні знання, що зберігають інформацію про лексику й граматику мови спілкування, інформацію про структуру діалогу, і підтримуючі знання. Допоміжні знання обробляються мовно-природним компонентом, але хід цієї обробки вирішувач не усвідомлює, тому що цей етап обробки вхідних повідомлень є допоміжним для проведення експертизи. Підтримуючі знання використовуються при створенні системи й при виконанні пояснень. Підтримуючі знання виконують роль описів (обґрунтувань) як інтерпретованих знань, так і дій системи. Підтримуючі знання підрозділяються на технологічні й семантичні. Технологічні підтримуючі знання містять відомості про час створення описуваних ними знань, про автора знань і т.п. Семантичні підтримуючі знання містять значеннєвий опис цих знань. Вони містять інформацію про причини уведення знань, про призначення знань, описують спосіб використання знань й одержуваний ефект. Підтримуючі знання мають описовий характер.

Інтерпретовані знання можна розділити на предметні знання, знання що управляють й знання про подання. Знання про подання містять інформацію про те, яким чином (у яких структурах) у системі представлені інтерпретовані знання.

Предметні знання містять дані про предметну область і способи перетворення цих даних при рішенні поставлених задач. Відзначимо, що стосовно предметних знань знання про подання й знання про керування є метазнаннями.

Якісні й кількісні показники експертної системи можуть бути значно покращені за рахунок використання метазнань, тобто знань про знання. Метазнання не представляють деяку єдину сутність, вони можуть застосовуватися для досягнення різних цілей. Перелічимо можливі призначення метазнань:

1. метазнання у вигляді стратегічних метаправил використовуються для вибору релевантних правил;
2. метазнання використовуються для обґрунтування доцільності застосування правил з області експертизи;
3. метаправила використовуються для виявлення синтаксичних і семантичних помилок у предметних правилах;
4. метаправила дозволяють системі адаптуватися до оточення шляхом перебудови предметних правил і функцій;
5. метаправила дозволяють явно вказати можливості й обмеження системи, тобто визначити, що система знає, а що не знає.

Питання організації знань необхідно розглядати в будь-якому поданні, і їхнє рішення в значній мірі не залежить від обраного способу (моделі) подання. Виділимо наступні аспекти проблеми організації знань :

- організація знань по рівнях подання й по рівнях детальності;
- організація знань у робочій пам'яті;
- організація знань у базі знань.

## **5. Пошук та пояснення рішень**

Для того щоб експертна система могла управляти процесом пошуку рішення, була здатна здобувати нові знання й пояснювати свої дії, вона повинна вміти не тільки використати свої знання, але й мати здатність розуміти й досліджувати їх, тобто ЕС повинна мати знання про те, як представлені її знання про проблемне середовище. Якщо знання про проблемне середовище назвати знаннями нульового рівня подання, то перший рівень подання містить метазнання, тобто знання про те, як представлені у внутрішньому світі системи знання нульового рівня. Перший рівень містить знання про те, які засоби використовуються для подання знань нульового рівня. Знання першого рівня відіграють істотну роль при керуванні процесом рішення, при придбанні й поясненні дій

системи. У зв'язку з тим, що знання першого рівня не містять посилань на знання нульового рівня, знання першого рівня незалежні від проблемного середовища.

Число рівнів подання може бути більше двох. Другий рівень подання містить відомості про знання першого рівня, тобто знання про подання базових понять першого рівня. Поділ знань по рівнях подання забезпечує розширення області застосовності системи.

Виділення рівнів детальності дозволяє розглядати знання з різним ступенем подробиці. Кількість рівнів детальності багато в чому визначається специфікою розв'язуваних задач, обсягом знань і способом їхнього подання. Як правило, виділяється не менш трьох рівнів детальності, що відбивають відповідно загальну, логічну й фізичну організацію знань. Введення декількох рівнів детальності забезпечує додатковий ступінь гнучкості системи, тому що дозволяє робити зміни на одному рівні, не зачіпаючи інших. Зміни на одному рівні детальності можуть приводити до додаткових змін на цьому ж рівні, що виявляється необхідним для забезпечення погодженості структур даних і програм. Однак наявність різних рівнів перешкоджає поширенню змін з одного рівня на інші.

Особливості предметної області з погляду методів рішення можна характеризувати наступними параметрами:

- розмір, що визначає об'єм простору, у якому має бути шукати рішення;
- змінюваність області, характеризує ступінь змінюваності області в часі й просторі (тут будемо виділяти статичні й динамічні області);
- повнота моделі, що описує область, характеризує адекватність моделі, використовуваної для опису даної області. Звичайно якщо модель не повна, то для опису області використовують кілька моделей, що доповнюють один одного за рахунок відбиття різних властивостей предметної області;
- визначеність даних про розв'язувану задачу, характеризує ступінь точності (помилковості) і повноти (неповноти) даних. Точність (помилковість) є показником того, що предметна область із погляду розв'язуваних задач описана точними або неточними даними; під повнотою (неповнотою) даних розуміється достатність (недостатність) вхідних даних для однозначного рішення задачі.

Складність задачі варіюється від простих задач малої розмірності з незмінними певними даними й відсутністю обмежень на результат і спосіб його одержання до складних задач великої розмірності зі змінюваними, помилковими й неповними даними й довільними обмеженнями на результат і спосіб його одержання. Із загальних міркувань зрозуміло, що яким-небудь одним методом не можна вирішити всі задачі. Звичайно одні методи перевершують інші тільки за деякими з перерахованих параметрів.

Розглянуті нижче методи можуть працювати в статичному й динамічному проблемному середовищах. Для того щоб вони працювали в умовах динаміки, необхідно враховувати час життя значень змінних, джерело даних для змінних, а також забезпечувати можливість зберігання історії значень змінних, моделювання зовнішнього оточення й оперування тимчасовими категоріями в правилах.

Перераховані методи при необхідності повинні поєднуватися для того, щоб дозволити вирішувати задачі, складність яких зростає одночасно за декількома параметрами.