

динамические экспертные системы. М., 1996. 5. Нильсон Н. Принципы искусственного интеллекта. М., 1985. 6. Представление и использование знания / Под ред. Х.Уэно, М.Исидзука. М., 1989. 7. Экспертные системы. Принципы работы и примеры. / Под редакцией Р.Форсайта. М., 1987.

УДК 007.159.955

Я.А.Кардаш

*Львівський національний університет імені Івана Франка,
кафедра обчислювальної математики*

МЕТОДИКА ВИДОБУВАННЯ ЗНАНЬ ДЛЯ ОДНІЄЇ БАЗОВАНОЇ НА ЗНАННЯХ СИСТЕМИ

© Я.А.Кардаш, 2000

This publication describes methodology for knowledge elicitation from the experts by which was developed specific knowledge based system. This methodology covers initial inquiry of knowledge — system identification, familiarization and a part of detailed investigation — conceptual design. Separatly acquisition of knowledge for forementioned system from textological sources is described.

1. Вступ

Набуття знань є процесом видобування і перенесення, нагромадження, перетворення, структурування та організації досвіду експертів із розв'язання проблем чи набуття знань з документованих джерел для конструювання чи розширення бази знань комп'ютерної програми. До потенційних джерел знань належать експерти, підручники, мультимедійні документи, бази даних, спеціальні дослідні звіти та джерела інформації з Internet.

Базова модель інженерії знань змальовує робочу групу, в якій інженер знань є з'єднувальною ланкою між експертом та базою знань. Інженер знань видобуває (перебирає) знання від експерта, очищає їх з ним і представляє в базі знань. Видобування знань від експерта можна робити вручну або за допомогою комп'ютерів. Існує три основні стратегії набуття знань:

1. *Ручні.* Без застосування комп'ютерів через контактування інженера знань з безпосереднім джерелом знань. Такі методи набуття знань називаються видобування знань.
2. *Напівавтоматичні.* Від експерта та з текстологічних джерел із застосуванням комп'ютера при наявності відповідного опрограмування.
3. *Автоматичні.* Із застосуванням комп'ютера при наявності програм навчання та існуванні відповідної репрезентативної вибірки прикладів прийняття рішень у предметній області.

Більшість технік ручного видобування знань походять з психології та системного аналізу. Ці методи добування класифікуються різноманітними способами та з'являються під різними назвами [1, 2, 3].

У цій статті розглядається методика видобування знань з експерта та видобування знань з книжок, монографій, методик та інших носіїв текстологічної інформації, яка застосовувалась під час побудови базованої на знаннях системи підтримки рішень в епілеп-тології [4].

2. Механізм ручного набуття знань

Процес набуття знань для базованих на знаннях та експертних систем можна відобразити схемою на рис. 1. У цій публікації розглядається питання отримання інформації від експерта та з текстологічних джерел для побудови системи. Послідовність набуття знань від експерта описується шестью стадіями (пп. 3-8), окремим пунктом (п.9) опи-сується процес набуття знань з текстологічних джерел.

На першій стадії (див. п. 3) пропонується процес визначення того, що буде робити система. На цій стадії необхідно відповісти на такі запитання:

- Чого досягне впровадження системи?
- Які конкретні завдання система виконуватиме?
- Хто використовуватиме систему?
- Як система впишеться в поточне робоче середовище?
- Хто буде постачати знання та досвід?

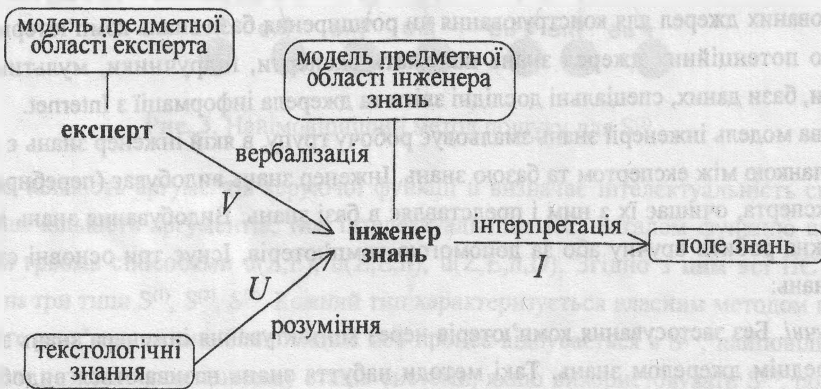


Рис. 1. Схема ручного набуття знань

На другій стадії (п. 4), після того, як ідентифікація системи закінчена, інженер знань повинен в загальних рисах освоїти те, як експерт виконує завдання. За методикою інженеру знань пропонується в'яснити, які кроки здійснює експерт, виконуючи завдання, типи

інформації, які експерт використовує під час виконання завдання, та результати чи виходи після виконання завдання. Отримують широкий огляд завдання, відбувається ознайомлення з усіма аспектами та повним набором ситуацій, в яких виконується завдання.

Інформація, одержана на стадії отримання загального розуміння завдання, дає підстави для подальшої дискусії інженера знань разом із засновниками, експертами та користувачами проекту про способи, за допомогою яких повна реалізація системи може бути поділена для циклічної розробки (п.5). Перед початком розробки для першого циклу повинен бути вибраний початковий обсяг та визначені прирости для майбутніх циклів. На початку побудови системи та початку кожного циклу необхідно в'яснити, як експерт виконує завдання в обсязі поточного циклу. Цього пропонується досягти за допомогою трьох чергових стадій.

Четверта стадія (п.6) описує діяльність інженера знань для розрізнення окремих кроків, які здійснює експерт при виконанні завдання, розпізнання порядку, в якому ці кроки здійснюються, та умов, за яких кожен крок виконується в різноманітних ситуаціях.

Зрозуміти процес міркувань експерта від вхідних даних до вихідних висновків, на яких базуються результати, можна за допомогою прийомів, запропонованих у п'ятій стадії (п.7).

На шостій стадії (п. 8) пропонується методика, яка дає змогу ідентифікувати ті характеристики ситуації, що є відповідні до виконання завдання.

В 9-му пункті пропонується техніка набуття знань з текстологічних джерел.

3. Визначення системи

Зрозуміння контексту. Перед дискусією про визначення системи учасникам проекту, а саме ініціаторам або засновникам проекту, інженерам знань, експертам та користувачам пропонується провести оглядову зустріч, метою якої є зрозуміти потенційні переваги базованих на знаннях та експертних систем та зрозуміти поточну ситуацію, яку система адресує. На зустрічі подається короткий огляд інженерії знань. Ще одним пунктом зустрічі є вступний огляд, який описує ситуацію чи проблему, що її адресує система. У табл. 1 визначено, яку інформацію повинен отримати інженер знань протягом вступного інтерв'ю та наступних дискусій. Після оглядової зустрічі група повинна виробити плани щодо збирання необхідної інформації для визначення системи.

Визначення цілей. Після того, як наданий контекст для системи, можна повернутись до цілей системи — чого досягне впровадження системи, тобто перша ціль, для якої служитиме система, друга ціль, для якої служитиме система тощо. Табл. 2 ілюструє зразок цілей системи.

Для встановлення пріоритету цілей можливе застосування тримірної шкали: необхідна ціль, важлива ціль, можлива ціль.

Таблиця 1

Вступна інформація для розробки системи

Загальна фоновіа інформація	Що учасник проекту повинен знати щоб зрозуміти роль експертних систем? (Це може включати означення найважливіших термінів)
Організаційні проблеми	Чи компетентність є рідкісною? Чи досягнення компетентності забирає багато часу? Чи плінність кадрів є високою? Чи керівництво хоче покращити деякі аспекти поточного виконання?
Поточні дії	Яка нормальна послідовність дій? Яка роль експерта?
Важливість компетентності	Чому це завдання вимагає експерта? Скільки часу забирає набуття майстерності в завданні? Наскільки вишколені є майбутні користувачі системи? Наскільки важливим є рівень майстерності експерта для завдання?

Таблиця 2

Цілі системи

Фактор	Ціль
Задоволення потреб хворих	■ Покращити якість лікарських послуг ■ Скоротити час надання послуг
Використання штату	■ Звільнити час експерта для важливішої діяльності ■ Зменшити час для підготовки нових членів штату ■ Дозволити менш досвідченому персоналу брати на себе більшу відповідальність ■ Дозволити тому, хто приймає рішення, дослідити та оцінити більше альтернатив за обмежений проміжок часу перед вибором варіанту на якому наполягає ■ Виявити рідкісну компетентність та знання та поширити її серед всіх практикуючих лікарів ■ Зменшити зусилля, необхідні для підтримки інформаційної системи на необхідному рівні у зв'язку із змінністю інформації
Однаковість послуг	■ Покращити узгодженість виконання ■ Збільшити прихильність до політики
Здобутки	■ Зменшити собівартість лікування ■ Зменшити тривалість часу надання лікарських послуг ■ Збільшити кількість лікарських послуг, які можуть бути надані хворим

Визначення завдання. Після визначення цілей системи фокус дискусій зміщується до завдання — що система повинна робити для того, щоб досягнути ці цілі. Опис завдання повинен визначити в загальних термінах всі важливі функції системи та цілі, які кожна функція адресує. Дискусія про завдання повинна розглянути, до якої міри кожна ціль може бути досягнута.

Ідентифікація користувачів. Для пробного означення системи необхідно встановити особистість її користувачів. Чітке розуміння того, хто використовуватиме систему, є інтегральною частиною визначення того, що система буде робити. Чітке

розуміння потреб користувачів може допомогти забезпеченню того, що система буде використовуватись.

Визначення середовища, в якому оперуватиме система. Повне визначення системи повинно містити ясний опис того, як система впишеться в робоче середовище користувачів. Щоб оцінити середовище системи, пропонується розглянути всі вхідні дані, які система потребуватиме, та всю продукцію, яку вона виробляє. Дискусія про операційне середовище повинна допомогти ідентифікувати початкові вимоги до користувацького інтерфейсу системи.

Ідентифікація джерел обізнаності. Після того, як завдання визначене, з представниками експертної групи треба обговорити такі питання:

Письмові матеріали. Які підручники, довідники, каталоги та інші письмові документи містять знання щодо застосування? Наскільки інженер знань потребує допомоги в інтерпретації цих документів? Хто буде здатний надати цю допомогу?

Людські експерти. Які різні ділянки обізнаності стосуються завдання? Хто є експертом і для якої ділянки?

Включення експертів в групу розробників. Скільки часу може кожен експерт присвятити проекту? Якщо проект вимагає більше ніж одного експерта, як вони будуть працювати з інженером знань?

Остаточні повноваження. Хто буде оцінювати, чи система поводитьсь правильно? Якщо експерти не погоджуються, хто влаштує диспути, які мають розв'язати неузгодженість?

4. Отримання загального розуміння завдання

Для одержання початкового розуміння завдання протягом серії інтерв'ю з експертом першим кроком пропонується оглянути, як експерт виконує завдання та обговорити конкретні випадки. Огляд завдання розглядається як ідеальний тема першого інтерв'ю із набуття знань. Огляд завдання обов'язково протоколюється. Перед інтерв'ю встановлюється, яка фоновна інформація потрібна для розуміння завдання — означення специфічних термінів, об'єктів, процедур, що стосуються завдання; структура об'єктів, центральних у завданні; деталі важливих процедур. В початкове інтерв'ю пропонується включити: фонову інформацію, яку не можна було одержати з наявних джерел; процес виконання експертом завдання; аспекти завдання, які є особливо важливі; аспекти, що є особливо важкі. Огляд завдання бажано підсумувати дискусією, щоб впевнитись, чи отримана вся необхідна інформація та чи зрозумілі всі важливі питання. Протягом серії інтерв'ю не важливо занурюватись у всі „як” та „чому” кожної акції, здійснюваної експертом під час виконання завдання. Потрібно зібрати достатньо інформації для розробки семи майбутніх сесій з набуття знань та проектування системи.

Наступним кроком пропонується обговорення випадку. Обговорення випадку — це розгляд виконання завдання експертом в деякій конкретній ситуації. В кожному випадку для процесу розв'язання проблеми потрібно визначити набір вхідних даних, що описують ситуацію, в якій виконується завдання. Випадки пропонується поділяти на три категорії: поточні (за виконанням яких просто можна спостерігати), історичні (для яких завдання виконувались в минулому) та гіпотетичні (експерт описує, як завдання слід би було виконувати в певній гіпотетичній ситуації). Першим пунктом, який пропонується зробити під час обговорення випадків, — це вибрати певну їх множину до обговорення. Достатньо взяти невелике число: п'ять-десять випадків. Випадки найкраще вибирати з тієї категорії, яка найчастіше зустрічається. Потрібно попросити експерта надати приклади випадків. Вони повинні ілюструвати широкий діапазон ситуацій з даної множини, щоб отримати широку картину обсягу тих ситуацій, з якими система буде працювати. Під час розробки системи [4] спочатку обговорювали простіші випадки, пізніше поступово складніші. Розглядали один випадок, який вважався експертом складним, та один випадок, який експерт вважав неординарним. На початку стадії інженер знань кілька разів спостерігав за тим, як експерт виконує завдання. Використовували гіпотетичні та історичні випадки, щоб впевнитись, чи було обговорено широкий набір ситуацій. Наступним пунктом була підготовка до обговорення випадку. Необхідно було дати експертові час, щоб він подумав про завдання та зібрав матеріали, які стосуються випадків. Кожен історичний та гіпотетичний випадок письмово описувався. Опис містив таку інформацію: дані (дані про ситуацію, які експерт використовував, виконуючи завдання); акції (кроки, які експерт здійснював при виконанні завдання); висновки (основні рішення, які експерт пам'ятає, що приймав, а також результати виконаного завдання) та часова послідовність (порядок, згідно з яким експерт збирав дані, виконував акції та робив висновки).

Третім кроком при обговоренні випадків був розгляд експертом випадку. Експерт крок за кроком описував процес виконання завдання для конкретного випадку. Вияснялася інформація про те, як експерт починав і що траплялося на кожному кроці. Встановлювалося, яку інформацію експерт використовував та які висновки робив. Обговорюючи випадок, не слід відволікати експерта запитаннями, які можуть збити хід його думок чи відволікти увагу від процесу виконання завдання. Такі запитання треба занотувати і задавати пізніше.

Для підвищення продуктивності подальшої роботи була розроблена стандартна форма опису випадків. Для цього необхідно ідентифікувати конкретні дані, акції-дії та висновки в тій послідовності, в якій вони застосовувались (див. вище).

5. Реалізація системи приростами

На закінчення стадії початкового набуття пропонується реалізувати останню ціль цієї стадії — зрозуміння того, як повний обсяг функціональності системи може бути розділений для циклічної реалізації. Ця стадія має бути необхідною в плануванні проекту створення базованої на знаннях чи експертної системи. Спочатку ідентифікувались

можливі класифікації функціональності. Для цього необхідно було одержати неупереджене розуміння організації знань експерта. Другим кроком необхідно було оцінити відносний розмір, складність та важливість категорій в межах кожної функціональної класифікації. Тут необхідно було зрозуміти, як оцінити неясний розмір, обсяги складності та важливості. І, нарешті, останнім кроком визначалося, котру з цих класифікацій або яку їх комбінацію використовувати, щоб розділити функціональність системи. Тут створювався план з неточної інформації, яку було отримано на попередніх двох кроках.

Ідентифікація можливих класифікацій. Першим кроком в розділенні обсягу системи була ідентифікація різноманітних способів, якими можна було розбити загальну функціональність системи. Розглядалися такі класифікації:

- функції або підзавдання, які система має виконувати;
- вводи, на які система буде здатна реагувати;
- результати, які система буде здатна продукувати.

Протягом перевірки класифікацій інженер повинен старатись в'яснити, як системна місткість може бути розбита згідно з класифікаціями, які здаються найбільш природними для експерта.

Завдання системи розбивали на підзавдання, які були реалізовані на різних циклах розробки. Спочатку ідентифікувались основні функції, які система мала виконувати. Якщо чітко не було ідентифіковано дискретні функції, необхідно виділити основну продукцію системи і визначити повне системне завдання як процес генерування всіх продуктованих системою результатів. Після ідентифікації функцій визначалося, які з них були необхідні для початкового розгортання системи, а які є необов'язкові на першому циклі розробки.

Також необхідно було постаратись обмежити ввід, на який система мала відповідати, класифікувавши випадки за ввідними категоріями. Для цього потрібно було зрозуміти, як експерт класифікував різноманітні ситуації, в яких кожне підзавдання могло бути виконано. Слід зауважити, що експерти часто лише підсвідомо порівнюють нову ситуацію з їхнім минулим досвідом та шукають знайомі зразки — ввідні категорії. Для їх визначення експерта просили згрупувати описи випадків — посортувати індексні карти з описами, щоб встановити, як експерт ділить випадки в конкретний спосіб на категорії і які характеристики випадку визначають ці категорії. Потрібно було зрозуміти, чому експерт бачить деякі випадки подібними один на одного, але відмінними від інших випадків. Таке розбиття на категорії можна робити в двох модельних ситуаціях — в першій експерт сам може класифікувати випадки. Якщо ж це експерту самотійно зробити не вдалося, то необхідно самому пропонувати можливі категорії для характеристики вхідних даних випадків.

Якщо так само вдалося розбити можливі результати, які системи продукує у групививідні категорії, які мають значення для експерта, то можна обмежити систему виконан-

ням лише цієї окремої категорії, в якій досягаються ці результати. А результати з інших категорій додавати на наступних циклах.

Порівняння категорій. Після ідентифікації класифікацій за трьома розмірностями (функція, ввід, результат) досліджували розмір, складність та важливість різноманітних категорій в кожній класифікації, щоб зібрати та організувати інформацію, на підставі якої приймалося рішення про планування реалізації системи.

У проектах майже неможливо одержати точні числа, потрібно отримати принаймні приблизну оцінку розміру ідентифікованих категорій за кожною класифікацією охоплення системи. Для цього складали список різноманітних категорій, одержаних на попередньому кроці і виконували їх оцінку або в абсолютних числах, або у числах в деякій одиниці стосовно до розміру різноманітних категорій. Інший спосіб — це визначити відсотки, які дають інформацію про відносний розмір різних категорій, ще один — це надати якесь значення певній категорії і відштовхуватись від нього як від бази, вибираючи числа для інших категорій. Якби не вдалося підібрати оцінку цими способами, то залишається останній варіант — скористатись описовою шкалою: дуже мало, мало, середня кількість, багато, дуже багато.

Наступний пункт у порівнянні категорій — це визначення відносної складності категорій застосування, яка є індикацією складності процесу міркування та стратегій, які експерт застосовує під час виконання завдання. Складність та розмір подібні, бо це приблизні, неточні величини і способи їх оцінювання однакові, тільки описова шкала трансформується у шкалу з такими значеннями: дуже легка, легка, середньо важка, важка, дуже важка. Оцінюючи складність, слід звернути увагу на таке: надзвичайно висока складність оцінки може вказувати на те, що процес розв'язання проблеми для цієї категорії не є добре зрозумілий. Якщо ж експерт оцінює категорію як винятково важку чи важку, то потрібно перевірити, чи експерт має метод для виконання завдання в ситуації, яка представляє категорію.

Порівнюючи категорії, враховували відносну важливість категорії чи підзавдання. Оцінку важливості робили, зважаючи на такі фактори:

Частоту. Чим частіше трапляється проблема, тим вона важливіша.

Потребу допомоги. Чим більше допомоги потребує користувач, застосовуючи цю категорію, тим вона важливіша.

Наслідки. Чим більше користі від включення функціональності, тим вона важливіша.

Застосовуючи ці фактори, враховуючи важливість кожної категорії функціональності, визначалась загальна оцінка важливості категорій функціональності. При її визначенні ідентифіковувались проміжки у важливості, тобто чи найважливіша категорія є тільки трошки важливіша за наступну категорію, чи набагато важливіша за неї тощо.

Планування розробки. За допомогою двох попередніх кроків було зібрано інформацію, на основі якої будувався план проекту. Після цього потрібно було застосувати таку процедуру:

1. Ідентифікувати прирости розробки або сегменти функціональності, кожен з яких

буде додано до експертної системи протягом окремого циклу розробки.

2. Визначити послідовність, за якою прирости будуть реалізовуватись.

3. Оцінити час розробки для додавання кожного приросту до системи.

Якщо відбулась класифікація функціональності більш ніж за одним параметром (функції-підзавдання, ввід, результати), то вирішується, які з цих класифікацій мають бути використані як базис для розбиття обсягу системи. Наприклад, якщо одна з цілей проекту — показати керівництву здійсненність технології розробки системи, то потрібно вибирати класифікацію, яка дозволяє ранню демонстрацію. Якщо мета — позиціонувати систему перед тим, як вона досягне повної функціональності, то необхідно вибрати класифікацію розробки системи, яка демонструє переваги користувачам, навіть якщо вона функціонально обмежена. Практичне розбиття функціональності системи показано на рис. 2.

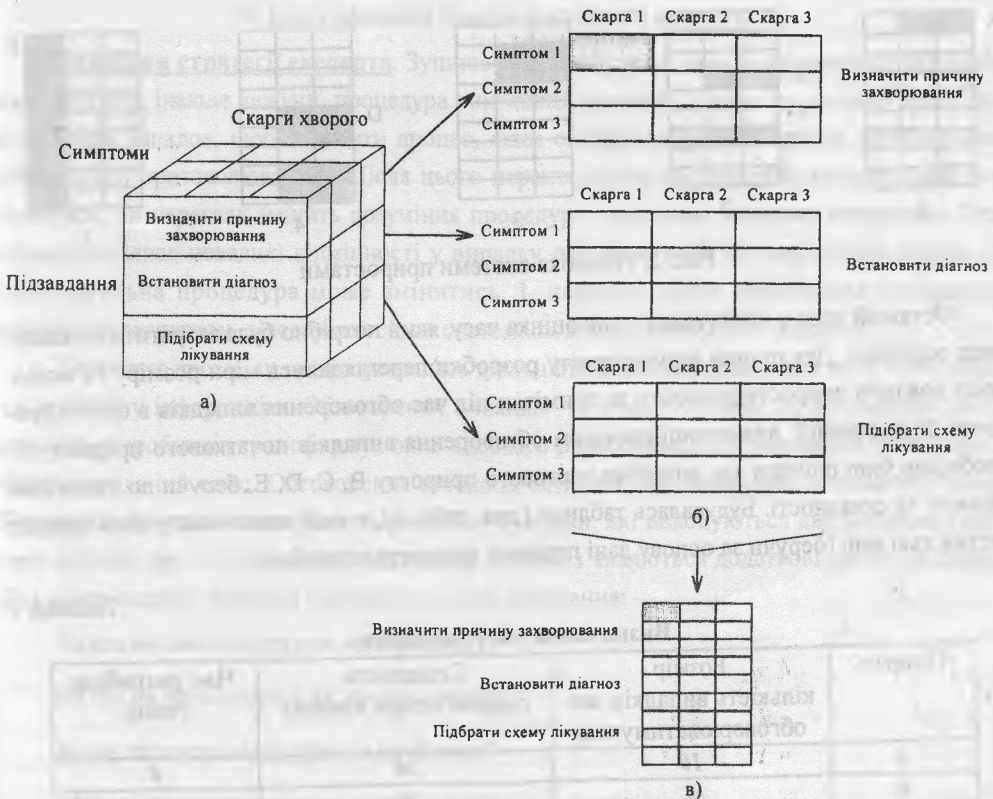


Рис. 2. Розбиття функціональності системи

Деякі комірки на рис. 2 в) можуть представляти істотно відмінну частину роботи. Групи комірок, які представляли малу кількість роботи, об'єднувались в окремий приріст виконання (враховуючи оцінки за розміром та складністю відповідних категорій).

Після того, як було визначено прирости розробки, вибиралась послідовність, за якою вони повинні бути реалізованими, спираючись на оцінки важливості, складності та розмі-

ру та враховуючи пріоритет цілей. У багатьох проектах команда розробників визначає послідовність всіх пристроїв; вона створює план цілого проекту, який охоплює повну розробку системи. В інших проектах команда планує лише перший цикл розробки. Після закінчення кожного циклу команда вибирає пристрій, який потрібно додати в наступному циклі.

Учасники проекту (інженер знань, представник засновників та експерт) розбивають функціональність системи, наприклад, вздовж трьох вимірів, враховуючи якісь три категорії функціональності: це підзавдання, скарги хворого та симптоми хвороби. Для простоти проілюструємо обмежене число підзавдань та симптомів.

Наприклад, послідовність реалізації для нашого випадку може бути реалізована за п'ять циклів (рис. 3), враховуючи розмір, складність категорій функціональності та пріоритет цілей.

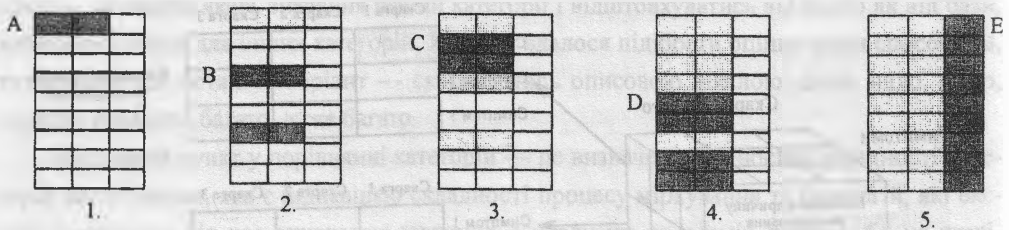


Рис. 3. Розробка системи приростами

Останній крок у плануванні — це оцінка часу, який потрібно буде затратити на кожен цикл розробки. Для оцінки часу приросту розробки переглядалися міри розміру та складності кожного приросту розробки та примітки під час обговорення випадків в обсязі приросту. З інформації, визначеної протягом обговорення випадків початкового приросту А, необхідно було оцінити час розробки чергового приросту В, С, D, Е, беручи до уваги дані розміру та складності. Будувалась таблиця (див. табл. 3), в якій кожен наступний пристрій містив такі дані (беручи за основу дані першого приросту розробки):

Таблиця 3

Визначення часу розробки

Приріст	Розмір кількість випадків, що обговорюватимуться	Складність години інтерв'ювання	Час розробки тижні
А	10	30	4
В	8	20	3
С	15	40	5
D	5	10	1
Е	6	12	1

Не всі прирости можуть бути оцінені експертом, бо можуть бути не пов'язані з видобуттям знань від експерта (наприклад, це завдання — друк результатуючої інформації чи внесення змін у наявну базу даних). Для встановлення загального значення необхідно врахувати час тестування (яке може здійснюватись після кожного етапу) та виділити

деякий час на неочікувані труднощі та затримки. Весь план обговорюється з експертом, щоб ідентифікувати розбіжності — прирости розробки, яким надано занадто багато або занадто мало часу відносно інших проростів.

З наступної стадії починається етап детального дослідження процесу набуття знань, трьома цілями якого є:

1. Вияснити точні кроки, які експерт здійснює під час виконання завдання.
2. Точно зрозуміти, як експерт міркує від вхідних даних до досягнення остаточних результатів.
3. Ідентифікувати конкретні факти та гіпотези, які експерт використовує в процесі міркування.

6. Ідентифікація кроків діяльності експерта

Вивчення стратегії експерта. Зупинюся на тому, яким чином визначалась стратегія експерта або, інакше кажучи, процедура виконання завдання. Спочатку інженер знань обговорював випадок, щоб вияснити процес, яким експерт виконує завдання. Обговорення обов'язково протоколювалося. Після цього переглядалося обговорення випадку, щоб подивитись, чи перегляд змінить розуміння процедури виконання завдання експертом. Тоді обговорювалися невеликі відмінності у випадку для того, щоб ідентифікувати умови, за яких загальна процедура може змінитись. І, нарешті, після визначення процедури виконання експертом, з'ясовувалося, як система повинна виконувати завдання.

Деталізую трохи, як організувалась діяльність із набуття знань для того, щоб зібрати необхідну інформацію. Обговорення випадків — саме той засіб, за допомогою якого команда розробників трансформує опис високого рівня процедури виконання експертом в більш детальний опис. Обговорення випадків служило для очищення детального опису. Протягом обговорення зверталось увагу на: 1) кроки, які виконуються автоматично і експерт забуває про них згадати; 2) обставини, за яких з'являються додаткові кроки і підходи. Для цього експерт повинен відповісти на такі запитання:

На що ви наголошуєте як на важливе?

На що ви наголошуєте як на незвичайне?

Як ви плануєте підходити до проблеми?

Що ви плануєте зробити першим (другим, ... , наступним)?

Визначення стратегії експерта. Після обговорення випадку команда розробників перевіряла, чи можна „очистити” визначення процедури експерта із виконання завдання. Команда розглядала процес, який експерт проілюстрував протягом обговорення випадку та порівнювала його з їхнім означенням процедури. Під час обговорення випадку можна:

з'ясувати деталі про крок в поточному визначенні;

ідентифікувати крок, пропущений в поточному визначенні;

ідентифікувати нові кроки, що є відповідними лише до ситуацій, подібних до обговореного випадку;

ідентифікувати кроки в поточному визначенні, що не є відповідними до ситуацій, подібних до обговореного випадку.

Вся команда обговорювала, як перші декілька випадків могли змінити означення стратегії експерта. Після розробки ясного, детального визначення процедури інженер знань сам розглядав нові випадки. З експертом консультувались лише у разі розбіжності між новими випадками та поточним визначенням. Якщо протягом попередніх діалогів обговорень випадків інформація збиралась, то під час огляду метою було підтвердити інтерпретацію, ідентифікувати кроки, що потребують перевизначення та виявити пропозиції щодо більш детальних підкроків. З аналізом інженера знань повинен був погодитись експерт, тобто впевнитись, що експерт та інженер знань розуміють одне одного.

Тестування загальності означення. Після того, як команда розробників оновила означення процедури експерта із виконання завдання, нею обговорювалося, чи незначні відмінності у випадку не призведуть до змін в процедурі. Для цього розглядались гіпотетичні альтернативи і ставились запитання на зразок “якщо — то”. Шукали за відмінностями в процедурі експерта при різних випадках з тим, щоб вияснити, чи поточне означення чітко пояснює, чому процедура змінюється від випадку до випадку.

Експерт описував, як він застосовував принципи, що спрямовують процедури. Після пояснення експертом цих принципів інженер знань дивився, як ці принципи могли бути застосовані в різних випадках. Водночас інженер знань та експерт створювали словник проекту, який описував терміни нових концепцій, що обговорювалися.

Визначення процедури роботи системи. На цьому кроці стратегія роботи експерта транслировалась у процес, яким система може виконати те саме завдання. Інженеру знань потрібно було ідентифікувати вхідні кроки, кроки з міркування та вихідні кроки системи. Визначався порядок, за яким система повинна виконувати ці кроки, та умови, за яких ці кроки здійснювати. Визначаючи процедуру виконання системи, враховували: 1) дії, які комп'ютерна програма не може здійснити (потрібно вирішити, що необхідно зробити замість цих акцій); 2) знаходили кроки, які не виконувалися експертом, а які система повинна була виконати (наприклад, оновлення бази даних); 3) знаходили кроки, які повинна виконувати система, а їх зараз ніхто не виконує. Потрібно уявити точне визначення того, як комп'ютерна програма виконує завдання. Також ідентифіковувались межові умови, які, можливо, не були згадані експертом, та обговорювалось, що системи повинна робити в таких ситуаціях.

Слід чітко розуміти, що експерт сам виконував завдання, а натомість система буде допомагати персоналу виконувати завдання. Потрібно відділити кроки, які виконуватиме система, від тих кроків, які здійснюватиме користувач. Якщо експерт робив якісь кроки самостійно, то система може тільки попросити користувача виконати цей крок і залежить від звіту користувача про результат.

7. Процес міркування експерта

За допомогою процесу міркування експерт використовує доступну інформацію про випадок, щоб визначити додаткові факти та гіпотези про випадок (це, наприклад, міркування експерта на підставі даних, які експерт обсервує чи знаходить в довідниках). На базі таких даних експерт робить висновки про можливі проблеми та їх причини. Експерт використовує ці висновки і ще додаткові дані спостережень, щоб зробити ще якісь висновки — і так доти, доки він не досягне рішень, які формують результати завдання — наприклад, причини хвороби та схему лікування. Ціль цієї стадії — зрозуміти на основі початкових тестів проблему та скласти список можливих причин в широкій множині ситуацій. Щоб зрозуміти процес міркування експерта, спочатку ідентифікували висновки експерта та інформацію, яка формувала основу цих висновків. Пізніше перевірялось, чи експерт в деяких зі своїх рішень не спирався на додаткові приховані припущення. Нарешті визначається, чи системі прийдеться працювати з нечіткою інформацією (цей аспект публікація не охоплює; це становитиме тему подальших досліджень).

Ідентифікація висновків експерта. Вивід визначає умову та висновок, який може бути зроблений, коли умова істинна. Умова виводу тестує факти та гіпотези про базові характеристики — характеристики випадку, які експерт розглядає, приймаючи рішення. Висновок визначає факти чи гіпотези, які експерт виводить про характеристики висновку.

Першим кроком в ідентифікації висновків експерта було розпізнання висновків. Для їх ідентифікації ідентифікувались знання, які дозволяли експерту прийняти рішення. Це могли бути означення, формули чи евристики (типово евристики виражають відношення, які експерт вважає істинними більшість часу). Другим кроком було обговорення випадків, яке надавало ідеальну можливість вияснення, як експерт міркує. Протягом інтерв'ю із набуття знань виявилось, що краще сфокусуватися на процесі міркування експерта не під час інтерв'ю-обговорення експертом виконання завдання, а в інтерв'ю-огляді оригінального обговорення випадку. У початковому обговоренні випадку треба було зосередити свою увагу на стратегії експерта, як це описувалось у п.6 цієї статті, і відмічати будь-які вияви того, що експерт робить висновки, в подальшому обговоренню. На початку огляду обговорення пояснювалось, що потрібно переглянути випадок, щоб інженер знань міг зрозуміти, як експерт приймає рішення. Вияснялось:

- як експерт міркував під час прийняття рішення;
- чому такий ланцюжок міркування був важливим;
- як експерт робив кожен висновок;
- коли експерт мав би робити такий самий висновок;
- коли експерт мав би робити інший висновок.

Команда розробників старалася ідентифікувати всі фактори, які робили дійсним кожне рішення. Пропонувались легкі відмінності у випадку, щоб побачити, чи експерт дійде інших висновків за інших обставин.

Перевірка на приховані припущення. В обговоренні випадку інженер знань покладається на експерта в ідентифікації основних характеристик для кожного виводу. І якщо експерт не згадує частину інформації при обговоренні окремого виводу, інженер знань може не усвідомити, що ця інформація була частиною логічного обґрунтування рішення експерта. Щоб цього уникнути, розглядались фактори, які експерт часто забував згадати, бо вважав такими, що маються на увазі, а також очевидні фактори.

Отже, як здійснювалась перевірка на припущення, які малися на увазі? Коли експерт та інженер знань вже обговорили окрему характеристику випадку, експерт може використовувати цю характеристику як основу для деяких майбутніх рішень, нічого не згадуючи про неї під час прийняття рішення. Інженер знань міг ідентифікувати, які рішення залежать від цієї характеристики, запитуючи експерта, що той мав би робити інакше, якщо б змінилось значення цієї характеристики. Також після того, як був ідентифікований вивід, перевірялося, чи він залишиться дійсним, коли обсяг системи розшириться.

Часто експерти забувають згадувати факти, що здаються їм очевидними, — наприклад, дані, які вони можуть спостерігати, чути чи відчувати. Тому для того, щоб виявити очевидні припущення, експерта просили пояснити виконання завдання іншій особі, професіоналу в тій самій галузі, ніби телефоном, як для окремого випадку. Цей другий спеціаліст готував повний опис випадку для дискусії. При такому обговоренні експерт не повинен бачити ні опис випадку, ні будь-яких фізичних об'єктів, які стосуються завдання. Експерт описував, як виконувати завдання. Коли дії експерта залежали від даних про випадок, експерт випитував ці дані в особи, що підготувала опис випадку. Особа, яка підготувала опис випадку, не могла сама ні приймати рішення стосовно випадку (лише постачати дані на запитання експерта), ні добровільно надавати якусь інформацію. Експерт задавав запитання на кшталт так/ні або запитання з кількома варіантами відповідей, наскільки це було можливо.

Другою ціллю етапу детального дослідження є достатньо глибоке зрозуміння процесу міркування експерта, з тим, щоб спроектувати систему, яка може виконувати подібні кроки міркування. Інженер знань повинен ідентифікувати виводи, які застосовує експерт, та визначити фактори, на яких ці виводи базуються.

8. Характеристики ситуації, що стосуються завдання

На попередніх стадіях інженер знань довідався про стратегію, яка спрямовувала дії експерта, та визначив процедуру, якої має дотримуватися система. Також вже є зрозумілі кроки міркування, які система мусить здійснювати, та ідентифіковано означення, формули й евристики, які дають змогу системі міркувати. Остаточна ціль етапу детального дослідження — це довідатись про загальні характеристики випадку, відповідні до виконання завдання. Загальні характеристики випадку — це вхідні дані, які система буде отримувати від користувачів чи з інших джерел, вихідні результати, які система продукує, та проміжні висновки, до яких система доходить, коли міркує від вхідних даних до вихідних результатів, — ідентифікуються протягом обговорення процедури виконання та процесу міркування.

Під час виконання завдання в цій ситуації система повинна акумулювати інформацію про ситуацію. Вона повинна містити факти та гіпотези, які система виводить, чи про які вона довідується; факти та гіпотези ідентифікують конкретні приклади різноманітних загальних характеристик.

Інженеру знань необхідно було одержати детальне означення кожної загальної характеристики, що ідентифікувалось. Означення надавало деталі про конкретні приклади характеристики та метод для в'яснення, котрий з цих конкретних прикладів стосується цього випадку. Означення вхідного даного вимагало деталей, які дозволяли б системі отримати відповідну інформацію з відповідного джерела. Означення вихідного результату ідентифікувало одержувача результату та вказувало, як інформація мала бути передана.

Ідентифікація конкретних характеристик. Загальні характеристики визначають розмірність предметної області, діапазон значень цих характеристик визначає її розмір.

Для дослідження загальних характеристик випадку користувались таким методом:

1. Складали список всіх характеристик, які експерт згадував в попередніх обговореннях (для цього розглядалися речення, які експерт вживав, описуючи випадок).
2. Далі складали список конкретних значень цих характеристик, які були згадані при обговоренні випадків (для цього розглядалось, що експерт говорив про загальні характеристики в кожному випадку).

Для ідентифікації конкретних значень загальних характеристик потрібно було відповісти на такі запитання:

1. Який тип значення утворює конкретний приклад? (Річ у тім, що багато характеристик можуть мати описові значення, а не числові).
2. Скільки значень цієї характеристики може мати частковий випадок?
3. Якими є всі конкретні приклади цієї характеристики?

Якщо характеристика була числовою, то старались ідентифікувати діапазон її значень. Якщо значення характеристики були описовими, то старались отримати список можливих значень. Якщо ж неможливо було перерахувати всі можливі значення, то старались одержати хоча б опис характеристик.

Наведені вище питання необхідно було в'яснити протягом обговорення випадку про діапазон можливих значень характеристик. В цьому обговоренні потрібно було зімітувати огляд виконання завдання іншим спеціалістом "по телефону" і постаратись визначити всі різні варіанти, які другий спеціаліст міг надати. Також слід було в'яснити, чи є якісь інші запитання, які потрібно задавати при інших обставинах. Деякі питання треба було задавати різними способами, щоб експерт не пропустив якоїсь характеристики чи варіанта значення. Протягом обговорення увага повинна бути зосереджена на конкретних прикладах загальних характеристик, хоча в діалозі також відбувається набуття деталей про процедуру завдання та процес міркування. Інженеру знань можна злегка відхилитись, щоб удосконалити процедуру виконання системи. Але такі відхилення не повинні переби-

вати хід думок експерта. Вони відбувались після того, як експерт викладав всю інформацію, що стосувалась окремої характеристики чи частини процедури. В кінці відхилення потрібно переорієнтовувати експерта, повертаючись до протоколу обговорення випадку.

Визначення, як система повинна одержати інформацію. Для проектування системи інженеру знань потрібно точно знати, як система буде отримувати кожну частину інформації про випадок. Деяку інформацію отримують із зовнішніх джерел, деяка інформація буде введена самою системою. Інженеру знань потрібно було оглянути характеристики з експертом, щоб вияснити:

- Як система визначить конкретну характеристику окремого випадку?
- Чи система повинна застосовувати різні методи для визначення цієї характеристики в інших випадках?
- Чи є альтернативні методи визначення цієї характеристики випадку, якщо стандартний метод підведе?
- Що повинна робити система, якщо вона буде нездатна визначити цю характеристику випадку?

Ідентифікація вхідних даних. Щоб отримати повне визначення характеристик, які формують вхідні дані системи, інженеру знань необхідно було детально вияснити джерело знань. Для цього спочатку ідентифіковувались зовнішні джерела даних. Експерт не буде спроможний надати всю потрібну інформацію, частково її потрібно зібрати від інших спеціалістів, залучених до проекту (це може бути інформація про базу даних, яку використовують в установі чи про застосування специфічних пристроїв). Експерт буде в стані запропонувати, як сформулювати фрази-запитання для користувачів, або чи відповідні дані повинні бути повідомлені в окремій формі чи через серію запитань, або чи включати діаграми з підказками. На цій стадії розробки інженер знань повинен достатньо довідатись, щоб розробити інтерфейс користувача, можливий для тестування системи. Здебільшого інтерфейс та системна процедура виконання завдання еволюціонує у процесі розробки при тестуванні та очищенні системи. Тому на цій стадії не потрібно витрачати надто багато часу на проектування форм та діаграм і на остаточне формулювання питань.

Визначення вихідних результатів. Так само, як для побудови системи необхідно було знати деталі про джерела вхідних даних, аналогічно потрібні деталі про призначення для вихідних результатів (це можуть бути користувачі, бази даних, якісь файли специфічного формату, інші пристрої) — див. табл. 4.

Вивід для користувачів містить не тільки системні висновки, але й також допомогу, яка пояснює, як користуватись системою. В деяких системах вивід може включати пояснення системного міркування. Інженер знань разом з експертом визначали побудову фраз та формат виводу, але не затрачали час на остаточне формулювання змісту того, що буде виводитись, перед реалізацією системи.

Розуміння призначення вихідних результатів

Користувач	Який рівень деталізації хоче користувач? Який формат виводу зробить його легшим до розуміння та використання?
База даних	В якій таблиці бази даних і в якому полі повинна бути записана інформація? Яка відповідна форма для запитів про оновлення бази даних? Як може експертна система доступитись до бази даних?
Файл	В який файл повинна бути записана інформація? Який формат файлу? Де у файлі повинна бути записана інформація?
Принтер (чи інший пристрій)	Який формат виводу зробить його легшим до розуміння та використання? Як система посилатиме інформацію до пристрою?

9. Набуття знання з текстологічних джерел

Завдання “ручного” видобування знань з текстів можна сформулювати як завдання розуміння та виділення змісту тексту. При такому підході текст можна розділити на дві структури: 1) зміст, який намагався закласти автор та 2) зміст, який формується в інженера знань в процесі розуміння тексту (рис. 1).

Як відбувається сам процес розуміння? Пропоную таку процедуру:

1. З'ясувати значення незрозумілих слів в тексті (спеціальної термінології).
2. Уточнити значення термінів та інтерпретацію окремих фрагментів тексту під впливом багажу попередніх знань.
3. Сформулювати деяку змістовну структуру тексту за рахунок встановлення внутрішніх зв'язків та формування абстрактних понять, що узагальнюють конкретні фрагменти знань.
4. Сформулювати загальну гіпотезу стосовно знань, що містяться в тексті.

Центральними моментами процесу розуміння є кроки 3 та 4, тобто формування структури змісту та виділення змістових віх і кінцеве зведення змістових віх в єдину цілісну семантичну структуру. В процесі розуміння застосовуються як дедуктивні (від загального до частинного), так і індуктивні методи (від частинного до загального).

Текстологічні методи (до останнього часу) рідко застосовуються як самостійний метод видобування знань, а переважно використовуються як деяка підготовка до серії інтерв'ю із набуття знань. Але й для прочитання текстів потрібна підготовка. Для цього необхідно разом з експертом оглянути деякий базовий список літератури, який поступово введе інженера знань в предметну область. Це можуть бути підручники для початківців, глави та фрагменти з монографій, популярні видання.

Наприкінці запропоную практичну послідовність “ручного” аналізу текстів, яка застосовувалась для витягнення знань при побудові системи:

1. Складання базового списку літератури та ознайомлення з ним.
2. Вибір тексту для витягнення знань.
3. Знайомство з текстом (швидкий перегляд). При виявленні незрозумілих термінів — консультації зі спеціалістами.
4. Формулювання попередньої гіпотези про зміст тексту.
5. Уважне читання тексту з виписуванням ключових слів та виразів, тобто виділення змістових віх.
6. Визначення зв'язків між ключовими словами, розробка тексту у вигляді графа або “конспекту”, формування змісту тексту.

10. Висновки

У статті подана методика видобування знань з експертів та текстологічних джерел, яка застосовувалась при розробці конкретної базованої на знаннях системи. Апробація системи показала, що подальшим напрямком наукових робіт повинно бути застосування знань в умовах невизначеності та технологія набуття таких знань і автоматичне формування знань в системі на основі репрезентативних вибірок.

1. Turban, J., E. Aronson. *Decision Support Systems and Intelligent Systems*. Prentice Hall, Upper Saddle River, N. Jersey, 1998. 2. G. Shreiberg et al. *Knowledge Engineering and Management*. MIT Press, Harvard, Mass., 1999. 3. A. Scott, et al. *A Practical Guide to Knowledge Acquisition*. Addison-Wesley, Readings, Mass., 1991. 4. Я.Кардаш. Базована на знаннях система підтримки рішень в епілептології. //Матеріали міжнародної конференції “Автоматика-2000”, Інститут інформаційної інфраструктури, Львів, 2000.