УДК 004.89

С.В. Шаров, С.О. Хрустальов

Мелітопольський державний педагогічний університет імені Богдана Хмельницького, Мелітополь

ІНСТРУМЕНТАЛЬНІ ЗАСОБИ ТА ЕТАПИ РОЗРОБКИ ЕКСПЕРТНИХ СИСТЕМ

У статті розглядаються основні етапи створення експертних систем, аналізуються інструментальні засоби, які можуть бути використані для розробки інтелектуальних систем. Виявлено, що у якості інструментальних засобів можуть бути використані мови програмування високого рівня (процедурні мови, інструментальні середовища швидкої розробки додатків), оболонки експертних систем, які не потребують значних знань у програмуванні, додаткові модулі у складі спеціалізованих програмних систем, спеціалізовані мови логічного та функціонального програмування. Авторами зазначається, що вибір інструментального засобу повинен залежати від задач, які висуваються до майбутньої експертної системи.

Ключові слова: *інтелектуальна система, експертна система, етапи розробки, інструментальні засоби.*

Вступ

Постановка проблеми. Сьогодні експертні системи знайшли своє використання у багатьох сферах, зокрема в медицині, науці, освіті, економіці тощо. Як один із типів інтелектуальних систем, вони призначені, у першу чергу, для генерування правильних та ефективних рішень у конкретній предметній галузі. Як наслідок, попит на програмні продукти такого типу залишається на високому рівні. Однак для того, щоб створити якісну експертну систему, слід ретельно підійти до планування всіх етапів розробки, а також до вибору адекватного інструментального засобу, який дозволить розробити саме той програмний продукт, який задовольнить замовника

Аналіз останніх досліджень. Питаннями розробки інтелектуальних, зокрема експертних, систем за допомогою різних інструментальних засобів займалося багато науковців та практиків: А.П. Частіков (Clips), Д.І. Муромцев (Drools Guvnor), Б. Тейт, Е. Хювенен (Lisp), М. Брамер, С.П. Хабаров (Prolog), О.П. Солдатова (Visual Prolog) та інші вчені. Розробкою інтелектуальних систем в різних галузях займалися Р.И. Баженов, Чванова М.С., Осадчий В.В., Леженко А.И. Желнин М.Е. Висвітленням питань щодо використання інтелектуальних інформаційних систем та засобів їх розробки займалися СП. Хабаров, В.С. Тоіскін, Ю.Ф. Тельнов, В.В. Литвин, А.М. Козлов та інші дослідники.

Метою статті є аналіз етапів розробки експертної системи та інструментальних засобів, які можуть бути використані для розробки інтелектуальних систем.

Основна частина

Кожний програмний засіб, особливо якщо він пов’язаний з обробкою знань та виконанням специфічних функцій людини, повинен бути ретельно спланований, а сам процес розробки поділений на декілька етапів. Як вважає А. Частіков [20, с. 32], процес розробки промислової експертної системи, призначеної для вирішення завдань в певній галузі, складається з декількох незалежних етапів, а саме: вибір проблеми, розробка прототипу, доопрацювання, оцінка, співпадіння, підтримка. Інші дослідники [7; 12] виділяють такі етапи як ідентифікація, концептуалізація та інші. Останній етап передбачає дослідну експлуатацію експертної системи. Опишемо етапі більш докладно.

Визначення завдань, мети і цілей розробки здійснюється на етапі ідентифікації, на якому складається неформальний (вербальний) опис загальних характеристик завдання, ключових понять, вхідних даних, можливі варіанти відповіді, перелік знань, що відносяться до конкретної задачі, підзадачі, які повинні бути вирішені всередині завдання [12; с. 104]. Тут же визначаються експерти та приблизні ролі користувачів.

Етап концептуалізації передбачає обробку значної кількості відповідного матеріалу з метою визначення основних понять проблемної області, їх взаємозв’язків, основних методів, якими буде оперувати майбутня експертна система.

Вибір способів подання та інтерпретації знань в ЕС, інструментальних засобів для розробки здійснюється на етапі формалізації. Тут відбувається представлення вибраних понять та термінів у вигляді формальної мови, яка вже була винайдена та розроблена, або створена для конкретної експертної системи. У результаті інженер знань отримує формальний опис розглянутої задачі. Результатом етапу формалізації є опис того, як розглянута задача може бути представлена в обраному або розробленому формалізмі [17, с. 64]. Тут же відбувається створення моделі системи та оцінка її адекватності.

Далі відбувається створення модулів експертної системи (етап виконання), перевірка її працездатності, виправлення помилок, аналіз відповідності можливостей ЕС вимогам кінцевих користувачів (етап дослідної експлуатації) [7, с. 100].

Слід зазначити, що розробка ЕС пов'язана з певними труднощами та формуванням способів їх подолання, а саме:

1. Проблема вилучення знань експертів. Жоден фахівець не розкриє секрети своєї професійної майстерності, якщо не буде зацікавлений в цьому. Це може бути матеріальна зацікавленість, або підвищення авторитету через публікацію своїх знань, або партнерство у проекті, який розробляється тощо.

2. Проблема формалізації знань експертів. Часто експерти, які є фахівцями у певній галузі, не в змозі формалізувати свої знання на рівні, достатньому для формалізації у базі знань. Багато фахівців приймають рішення інтуїтивно, без обдумування всього ланцюга доказів, який призвів до кінцевого правильного рішення. Тому рекомендується вибирати такого експерта, який в змозі пояснити хід своїх думок навіть непрофесіоналам.

3. Проблема нестачі часу у експерта. Часто обраний експерт не має вільного часу для того, щоб дати свої знання. Для уникнення такої ситуації необхідно заздалегідь домовитися із фахівцем з приводу декількох зустрічей для витягування знань.

4. Недостатня точність правил, які формує експерт під час роботи з інженером знань. Для уникнення такої ситуації слід використовувати реальні задачі, на які експерт може дати адекватні рішення. Крім того, при спілкуванні з експертом доречно використовувати єдину термінологію, яку потім використовувати при наповненні бази знань.

5. Недолік ресурсів. Ця проблема пов'язана із значними затратами часових ресурсів на розробку ефективної ЕС. Крім того, людський фактор теж має великий вплив час розробки. Потрібно правильно підбирати групу розробників, які будуть знати свої завдання та способи їх виконання.

6. Неадекватність інструментальних засобів розв'язуваній задачі. Часто вибір інструментального засобу або моделі подання знань не відповідає тим завданням, які повинна вирішувати ЕС. В даному випадку, рекомендується провести ґрунтовний аналіз мети та переліку завдань, які потрібно вирішити, а тільки потім вибрати інструментарій [13].

Ми згодні з В. Захаровою та В. Хорошевським, які вважають програмну реалізацію інтелектуальних програмних проектів доволі простим з точки зору виконання етапом. На цьому етапі застосовується цілий арсенал мов та середовищ програмування, що орієнтовані на ефективну та зручну реалізацію різних класів задач, трансляторів та компіляторів, що забезпечують отримання якісних комп'ютерних програм. Крім того, зараз стали використовуватися Case-засоби, які дозволяють певним чином автоматизувати синтез програмних продуктів [6, с. 7].

Більшість програмних засобів, які можуть використовуватися для розробки інтелектуальних, зокрема експертних систем, можна поділити на декілька категорій:

1. Мови програмування високого рівня позбавляють розробників від необхідності заглиблюватися в тонкощі програмної реалізації. В даному випадку не потрібно слідкувати за розробкою низькорівневих процедур доступу до даних та маніпулювання ними, ефективним розподілом пам'яті тощо. Умовно їх можна розділити на традиційні процедурні мови типу Pascal, C ++ та традиційні середовища програмування, які підтримують візуальну розробку додатків та об'єктно-орієнтоване програмування, такі як C# та Delphi.

2. Оболонки експертних систем створюються, як правило, на базі конкретної ЕС, яка вже давно та успішно використовується. Оболонка представляє собою ту ж саму експертну систему, з якої видалені специфічні для конкретної наочної області компоненти, а залишаються ті компоненти, які мають більш-менш загальну спеціалізацію.

Прикладом є експертна система EMYCIN, основою для якої стала відома медична система MYCIN. До складу EMYCIN входить інтерпретатор, а також більшість базових структур даних. Крім того, функціонал оболонки був доповнений спеціальною мовою, яка поліпшує роботу із програмами та бібліотеками типових висновків, що були раніше виконані ЕС [5, с. 272]. Добре себе показали такі інструментальні засоби як KAPPA, ЕКО та інші, за допомогою яких можна створювати різні прикладні системи зі штучного інтелекту з можливістю оновлення баз знань [2, с. 15].

Оболонка Clips є OVS-подібною виробничою системою, розробленою на мові C. Clips підтримує декілька технологій програмування: процедурну, об'єктно-орієнтовану, засновану на правилах [4, с. 553]. Вона має власний механізм логічного висновку, інтегровану об'єктно-орієнтовану мову COOL, які взаємодіють безпосередньо між собою. Clips добре інтегрується в інші прикладні розробки, працює на декількох платформах. На основі Clips була створена оболонка FuzzyClips, заснована на правилах. В її основу покладено дві базисних концепції: невизначеність та нечіткість. Завдяки зазначеним принципам, крім батьківських можливостей Clips, FuzzyClips має механізми для роботи з нечіткими, точними знаннями, складними міркуваннями. Всі дані представляються у вигляді фактів та правил експертної системи [8, с. 61].

За допомогою простих оболонок експертних систем, таких як «Мала експертна система 2.0» або

Exsys Corvid Eval, можна створювати корисні програмні продукти. Наприклад, Р.И. Баженов повідомляє про створення експертної системи, яка дозволяє початківцю провести первинну діагностику несправностей комп'ютерної техніки. Оболонка «Мала експертна система 2.0» використовує Байєсовську систему логічного висновку (Баєсовська мережа довіри). Вона призначена для проведення консультації з користувачем в будь-якій прикладній області за умови, якщо буде завантажена відповідна база знань. Оболонка дозволяє визначати ймовірність можливих результатів через оцінку правдоподібності деяких передумов, що отримуються від користувача [11].

3. Додаткові модулі, які можна представити у вигляді корисних програм, що виконуються разом з додатком, для виконання специфічних завдань у межах конкретної інтелектуальної системи. Такі програми, як правило, реалізують певні спеціальні функції без необхідності здійснення програмування або індивідуального настроювання інтелектуальної системи.

Прикладом додаткового модулю можна вважати програмний пакет Simkit, який входить до комплекту середовища KEE. З його допомогою експертна система володіє методами моделювання. Також непоганим прикладом можна вважати модуль роботи з семантичною мережею, який інтегрується до системи VT. Іншим прикладом додаткових модулів є механізм обробки безлічі різних контекстів логічних міркувань, який є у комплекті експертних систем KEE і ART [5, с. 280].

4. Програмування на спеціалізованих мовах, які застосовуються для створення баз знань та обробки об'єктів та відносин між ними: Lisp, Prolog, Clips [9, с. 134] та ін. Розглянемо їх більш докладно.

Prolog. Відомо, що будь-яка мова чи середовище програмування (крім Visual Studio, Eclips та ін.) зазвичай орієнтована на певне коло завдань, при вирішенні яких вона вважається найбільш дієвою. Для даної мови логічного програмування типовими є завдання, що пов'язані з розробкою різноманітних програмних комплексів зі штучним інтелектом. До таких програмних засобів відносяться комп'ютерні перекладачі, експертні системи, інтелектуальні ігри тощо [18, с. 4].

Мова програмування Prolog вважається самою відомою мовою логічного програмування, яка використовується для практичної реалізації багатьох завдань: у дослідженнях штучного інтелекту; створенні онтологій, експертних систем та систем обробки природних мов тощо. На відміну від процедурної мови, де використовується послідовність інструкції (процедура, функція), Prolog виконує пошук рішення серед опису декількох об'єктів та правил, які входять до його бази знань, за допомогою механізму уніфікації та рекурсії. Пролог вирішує завдання за допомогою логічного висновку з відомих (які були раніше введені до системи) фактів. Програма (програмний код) на цій мові являє собою набір фактів і правил, обробка яких дозволяє отримати логічні висновки з наявних фактів. Тому дана мова програмування вважається декларативною [15, с. 17].

Prolog володіє доволі потужними засобами пошуку, за допомогою яких здійснюється відбір інформації з БЗ. При цьому методи пошуку, властиві Прологу, суттєво відрізняються від звичайних, до яких звикли більшість програмістів [18, с. 4].

Використовуючи Turbo Prolog, можна отримати декілька переваг, а саме:

– використовувати сукупність фраз, яка застосовується з метою представлення правил, процедур або даних;

– зіставляти дані за допомогою універсального механізму зіставлення даних;

– застосовувати правила спадного пошуку та обчислення зліва направо у стратегії управління знаннями [5, с. 278].

Ця мова одночасно використовує дві техніки програмування: обробку списків та рекурсію. Завдяки унікальній структури програми обробка даних у списках займає декілька рядків замість великого блоку програмного коду в інших мовах програмування. Ця мова існує на декількох операційних платформах, зокрема Unix, Windows, Macintosh. Існують які платні, так і безкоштовні версії цього інструментального засобу логічного програмування [21].

Turbo Prolog продовжив існувати у вигляді об'єктно-орієнтованого розширення Visual Prolog, що є розробкою Prolog Development Center. Його використання зменшує час на виконання тривіальних операцій, дозволяє автоматизувати побудову складних програмних конструкцій. Відмінності Visual Prolog від Turbo Prolog полягають у наявності графічного середовища, візуальних засобів та Експертів кодів для розробки програми [23].

Е. Акчурин визначає такі функціональні відмінності зазначених середовищ логічного програмування: програма на Visual Prolog складається з декількох файлів; для написання програми на Visual Prolog можна використовувати класичні об'єктно-орієнтовані функції; за допомогою концепції «область ідентифікації», програма, складена на Visual Prolog, може використовувати функціональність інших модулів [1].

На думку К. Фернандо, сьогодні Пролог має приблизно однакову ефективність у порівнянні з іншими мовами символьної обробки (наприклад, Lisp) у багатьох сферах застосування, зокрема для обробки природної мови [22, с. 6].

Lisp. Мова програмування Lisp, поряд із Prolog, використовується у дослідженнях зі штучного інтелекту та для обробки символьної інформації. Багато програмних продуктів, які використовують символьну обробку або працюють з природньою мовою, розроблені саме на Lisp. Також з цієї мови були взяті багато методів, які широко використовуються у штучному інтелекті [19, с. 7].

У «чистому» вигляді мова Ліспе містить перелік принципових особливостей програмування у функціональному стилі. Програмування на Lisp передбачає вибір відповідної структури та невеликий набір функцій, які можна виконувати над обраною структурою. Інформаційна обробка на мові програмування Ліспе відрізняється від більшості підходів до програмування трьома важливими принципами: подобою до машинних мов, природою даних, самоопису обробки символьних виразів [3, с. 9]. Для того, щоб почати розробляти додатки на цій мові програмування, необхідно вибрати конкретну реалізацію Lisp. Особливість Lisp полягає в тому, що не існує єдиної реалізації (як, наприклад, у випадку з Perl і Python), ні канонічної реалізації, яка здійснюється однією компанією (як, наприклад, у випадку з VB, C# або Java). Будь-який програміст може створити власну реалізацію Lisp на основі загальноприйнятого стандарту, який контролюється асоціацією ANSI.

Як вважає Б. Тейт, Lisp не стала досить поширеною мовою, однак її опанування дозволить виявити безліч технічних прийомів, які змінять техніку кодування програміста на краще [16].

Слід зазначити, що програмні засоби, розроблені за допомогою мов програмування Lisp і Prolog, дозволили вирішити багато завдань, покладених на них. Однак останнім часом попит ці мови програмування для вирішення інтелектуальних завдань дещо знизився, що можна пояснити суттєвими змінами у вимогах до розробки програмних комплексів. Це, в першу чергу, стосується користувацького інтерфейсу та підтримки кросплатформності. Крім того, досить часто мови логічного програмування є частиною великих додатків, тому більш доречно використовувати єдине середовище для розробки програмного продукту. У цьому аспекті можна назвати такі сучасні інструментальні засоби як C ++, Java, Smalltalk та ін. Однак, програмування Lisp і Prolog знаходять своє використання у створенні прототипів, а також обґрунтуванні доцільності використання інструментів та механізмів, які входять до складу інструментальних засобів [10, с. 608].

Одним із відомих представників спеціалізованих мов високого рівня є мова OPS5, яка є простою у вивченні та надає розробнику доволі потужний функціонал у порівнянні з типовими спеціалізованими оболонками. Слід зауважити, що окремі перспективні інструментальні засоби так і залишилися на рівні прототипу [5, с. 272].

В. Осадчий, К. Осадча та інші вчені, які займалися аналізом засобів для розробки інтелектуальних систем, дійшли висновку, що сьогодні часто використовуються такі мови програмування: Java у вигляді складової частини інтелектуальних платформ, таких як Aglobe, Coguaar, DARPA; C# та технологія .Net; мова програмування Python для розробки бази знань та інших додатків [14, с. 39].

Висновки

Отже, на сьогодні існує багато інструментальних засобів для розробки експертних систем, які передбачають різний рівень програмування та функціоналу. Найбільш швидкий спосіб створення та наповнення експертних систем полягає у використанні оболонок, які дозволяють з короткий час створити та наповнити достатньо-функціональний програмний засіб. Вибір конкретного інструментального середовища залежить, у першу чергу, від призначення експертної системи, розміру бази знань, типом представлення знань, рівнем підготовки розробника тощо.

Список літератури

1. Акчурин Э.А. Программирование на языке Visual Prolog в ИСР Visual Prolog: учеб. пособ. для студентов направления «Информатика и вычислительная техника»: [Електронний ресурс] / Э.А. Акчурин. – Самара, 2012. – Режим доступу: http://ivt.psuti.ru/files/IVT_mag/IntSys/LK_VisualProlog_Akchurin_2012.pdf.

2. Гаврилова Т.А. Базы знаний интеллектуальных систем / Т.А. Гаврилова, В.Ф. Хорошевский. – СПб: Питер, 2000. – 384 с.

3. Городная Л. Введение в программирование на языке Лисп: Учеб. пособ. для начинающих / Л. Городная. – Новосибирск, 2005. – 46 с.

4. Джарратано Д. Экспертные системы: принципы разработки и программирование / Д. Джарратано, Г. Райли. – М.: ООО «И.Д. Вильямс», 2007. – 1152 с.

5. Джексон П. Введение в экспертные системы / Питер Джексон. – СПб.: Вильямс, 2001. – 393 с.

6. Искусственный интеллект: в 3-х кн. Кн. 3. Программные и аппаратные средства: справочн. / под ред. В.Н. Захарова, В.Ф. Хорошевского. – М.: Радио и связь, 1990. – 368 с.

7. Козлов А.Н. Интеллектуальные информационные системы: учебн. / А.Н. Козлов. – Пермь: Изд-во ФГБОУ ВПО Пермская ГСХА, 2013.– 278 с.

8. Леженко А.И. Использование экспертных систем для интеллектуального анализа данных / А.И. Леженко, И.А. Кузнецов, С.К. Кузнецов // Информационные технологии и вычислительные системы. – 2012. – № 1. – С. 60-64.

9. Литвин В.В. Модель представления знаний посредством объектов для построения интеллектуальных систем поддержки принятия решений / В.В. Литвин, Д.Г. Досин, Р.Р. Даревич // Известия Известия Южного федерального университета. Технические науки.– 2004. – № 9. – С. 128-134.

10. Люгер Дж. Искусственный интеллект: стратегии и методы решения сложных проблем / Дж. Ф. Люгер. – М.: Издательский дом «Вильяме», 2003. – 864 с.

11. Мазилов А.О. Разработка экспертной системы диагностирования неисправности персонального компьютера: [Електронний ресурс] / А.О. Мазилов, Р.И. Баженов // Nauka-rastudent.ru. – 2015. – No. 06 (18). – Режим доступу: http://nauka-rastudent.ru/18/2753.

12. Макаренко С.И. Интеллектуальные информационные системы: учебное пособие / С.И. Макаренко. – Ставрополь: СФ МГГУ им. М. А. Шолохова, 2009. – 206 с.

13. Методология построения экспертных систем: [Електронний ресурс]. – Режим доступу: http://www.intuit.ru/studies/courses/46/46/lecture/1378?page=2.

14. Осадчий В.В. Аналіз програмних засобів для створення інтелектуальних систем в освітніх цілях / В.В. Осадчий, К.П. Осадча // Науково-педагогічний журнал «Молодь і ринок». – №8 (127). – Дрогобич: ДДПУ ім. І. Франка, 2015. – С. 37-42.

15. Солдатова О.П. Логическое программирование на языке Visual Prolog: учеб. пособ. / О.П. Солдатова, И.В. Лёзина. – Самара: СНЦ РАН, 2010. – 81 с.

16. Тейт Б. Пересекая границы: красота Lisp. Эльдорадо языков программирования: [Електронний ресурс] / Б. Тейт. – Режим доступу: http://www.ibm.com/developerworks/ru/library/j-cb02067/index.html.

17. Тоискин В.С. Интеллектуальные информационные системы: учеб. пособ. / В.С. Тоискин. – Ставрополь: Изд-во СГПИ, 2009. – 181 с.

18. Хабаров С.П. Интеллектуальные информационные системы. PROLOG – язык разработки интеллектуальных и экспертных систем: учеб. пособ. / С.П. Хабаров. – СПб.: СПбГЛТУ, 2013.– 138 с.

19. Хювёнен Э. Мир Лиспа. Введение в язык Лисп и функциональное программирование / Э. Хювёнен, И. Сеппянен. – М.: Мир, 1990. – 458 с.

20. Частиков А.П. Разработка экспертных систем. Среда Clips / А.П. Частиков, Т.А. Гаврилова, Д.Л. Белов. – СПб: «БХВ-Петербург», 2003. – 393 с.

21. Bramer M. Logic Programming with Prolog / M. Bramer. – Springer, 2005. – 223 p.

22. Fernando C.N. Prolog and Natural-Language Analysis / Fernando C.N. Pereira and Stuart M. Shieber. – Massachusetts: Microtome Publishing, 2005. – 194 p.

23. Visual Prolog: [Електронний ресурс]. – Режим доступу: https://ru.wikipedia.org/wiki/Visual_Prolog.

Надійшла до редколегії 28.10.2016

Рецензент: д-р техн. наук, проф. В.С. Єремєєв, Мелітопольський державний педагогічний університет імені Богдана Хмельницького, Мелітополь.

ИНСТРУМЕНТАЛЬНЫЕ СРЕДСТВА И ЭТАПЫ РАЗРАБОТКИ ЭКСПЕРТНЫХ СИСТЕМ

С.В. Шаров, С.О. Хрусталев

В статье рассматриваются основные этапы создания экспертных систем, анализируются инструментальные средства, которые могут быть использованы для разработки интеллектуальных систем. Выявлено, что в качестве инструментальных средств могут быть использованы языки программирования высокого уровня (процедурные языки, инструментальные среды быстрой разработки приложений), оболочки экспертных систем, которые не требуют значительных знаний в программировании, дополнительные модули в составе специализированных программных систем, специализированные языки логического и функционального программирования. Авторами отмечается, что выбор инструментального средства должен зависеть от задач, которые предъявляются к будущей экспертной системе.

Ключевые слова: *интеллектуальная система, экспертная система, этапы разработки, инструментальные средства.*

TOOLS AND STAGES OF DEVELOPMENT EXPERT SYSTEMS

S.V. Sharov, S.O. Hrustal’ev

The article considers the main stages of creation of expert systems, analyzed tools that can be used to develop intelligent systems. It was revealed that in the specialized language of logic as a tool can be used high level programming languages (procedural languages, instrumental environment of rapid application development), shell of expert systems that do not require significant knowledge in programming, additional modules as part of a specialized, bathrooms software systems and functional programming. The authors noted that the choice of tool should depend on the tasks that apply to the future expert system.

Keywords: *intelligent system, expert system, stages of development, tools.*