

УДК 004.896

АНАЛИЗ СУЩЕСТВУЮЩИХ ИНСТРУМЕНТАЛЬНЫХ ОБОЛОЧЕК, ОСНОВАННЫХ НА ПРОДУКЦИОННЫХ МОДЕЛЯХ

Морозова О.В., Григорьев А.В.

Донецкий национальный технический университет

E-mail: olmalyavka@gmail.com

Аннотация

Морозова О.В., Григорьев А.В. Анализ существующих инструментальных оболочек, основанных на продукционных моделях. В работе выполнен анализ типичных инструментальных оболочек для создания экспертных систем. Определены достоинства и недостатки таких оболочек с точки зрения применяемых методов построения системы продукционных правил и механизма вывода. Сделан вывод о достоинствах и недостатках применения данных систем для создания интеллектуальных САПР с адаптацией на требуемую предметную область.

Введение

В настоящее время активно повышается уровень интеллектуальности разнообразных САПР за счет построения их как гибридных. Это достигается за счет добавления в них экспертных систем, автоматизирующих процессы проектирования. При этом в САПР надстраивается база знаний, которая реализует методику проектирования в данной предметной области [1]. Современное развитие систем искусственного интеллекта привело к созданию большого количества универсальных программных средств для построения экспертных систем различного назначения, так называемых «оболочек». В основе большинства таких продуктов лежит механизм накопления знаний о предметной области и способов их обработки. Большинство существующих инструментальных оболочек являются оболочками общего назначения. К ним можно отнести такие системы как Clips, Prolog, Nexpert Object, Exys Corvid, Jess и т.д. Названные системы обладают не достаточно развитыми средствами предметной адаптации. С другой стороны, имеются оболочки общего назначения типа G2, обладающие достаточно большой способностью адаптироваться на предметную область. Данные программные комплексы используют разнообразные специфические способы представления и обработки знаний, основанные на стандартных подходах. Используются фреймы, семантические сети, системы продукций, нечеткие множества, нейронные сети, генетические алгоритмы и др. Однако, в рамках данных инструментальных средствах нет возможности создания полноценных интеллектуальных САПР, так как большинство из них узконаправленные и не способны в достаточной степени адаптироваться на техническую предметную область. Актуальной является задача построения инструментальной оболочки по принципам построения оболочек общего назначения, но ориентированной на создание интеллектуальной САПР для избранной технической предметной области. Оболочка такого типа должна обладать средствами адаптации на предметную область в рамках концепции «умного эксперта», т.е. профессионального проектировщика, обладающего большим опытом проектирования в данной предметной области и способного задать методику проектирования устройств данного типа как продукционную базу знаний в рамках выбранной предметной области без привлечения инженера по знаниям [2].

Целью предлагаемой работы является анализ методов построения различных оболочек данного класса для выявления их достоинств и недостатков с последующим использованием результатов анализа при решении задачи построения оболочки общего назначения, ориентированной на технические предметные области САПР.

Инструментальные оболочки

Оболочка G2

Одной из наиболее универсальных платформ для разработки и внедрения экспертных приложений является платформа G2 компании Gensym. Данная платформа может решать несколько типов задач, такие как: моделирование; реализацию расписаний и планирований; диагностика; мониторинг в реальном времени; системы проектирования.

Основу экспертной компоненты G2 составляет система продукционных правил [3].

Правила в системе G2 разного типа и назначения: обобщенные, касающиеся целого класса объектов; специфические, относящиеся к конкретному экземпляру объекта.

Структура правил имеет вид: левая часть (антецедент) и правая часть (консеквент) или стандартная форма записи:

Если (условие) то (значение).

Важной особенностью механизма логического вывода G2 является и большой набор способов выполнения и активации правил. В G2 применяются прямой и обратный логический вывод, срабатывание правил по счетчику, применение фокусировки на определенном правиле, изменение переменной, находящейся в посылке, изменение параметров, автоматически при запуске приложения и др.

Так же в этой системе используется процедурный подход. Для процедурного подхода в системе G2 был разработан собственный язык программирования, близкий с Паскалем [3]. В данный язык добавлены элементы для работы процедуры в реальном времени: отслеживание наступления события; разрешение прерывать другим задачам данную процедуру; директивы, которые определяют порядок выполнения операций (последовательных или параллельных); итераторы, которые могут организовывать цикл над множеством экземпляров класса. Данный язык позволяет одновременно обрабатывать множество процедур для множества различных объектов. Несмотря на сложность и богатство синтаксических конструкций G2 для описания знаний, их применение упрощается за счет естественно-языкового подхода.

G2 работает с разными предметными областями, такими как телекоммуникации, химическая, нефтяная и газовая промышленность, аэрокосмическая и др. В G2 встроен механизм адаптации на предметную область. Для ориентации на участие специалиста из той или иной предметной области в процессе разработке приложений G2, поддерживаются правила как на естественном языке, так на графическом. Они организованы различными способами, такими как таблицы, древовидные диаграммы, и иерархические рабочие пространства. Правила и их выполнения представлены в виде графических структур, которые отображают взаимодействие, интеграцию и последовательность для моделей процессов.

Оболочка Clips

Наиболее распространённым языком представления знаний является Clips [4]. Более 80% экспертных систем используют именно его [5]. Программная среда Clips для создания экспертных систем. Язык Clips основан на системе правил. Эти правила являются порождающими, то есть их значения добавляются по предложению-образцу. Так как Clips оперирует понятием продукционной модели представления знаний, то она состоит из 3 основных элементов: базы фактов; базы правил; блока логического вывода.

Основными компонентами языка описания правил являются база фактов и база правил. На них возлагаются следующие функции:

- в базе фактов хранятся исходные состояния предметной области;
- в базе правил хранится информация об операциях, которые необходимо выполнить для решения поставленной задачи.

Правила в Clips имеют вид “Если (условие), то (действие)”. Под “условием” понимается некоторое предложение-образец, по которому осуществляется поиск в базе знаний, а под “действием” – действия, выполняемые при успешном исходе поиска [5].

Блок логического вывода Clips сравнивает факты и правила и решает, какие из правил можно отправить на обработку. В Clips могут задаваться шаблоны, состоящие из списка фактов. Каждое определение шаблона состоит из произвольного имени.

В языке Clips также можно определять функции, они имеют схожий вид и синтаксис с функциями языка LIPS. В записи функции должен быть записан префикс «?». Функция возвращает результат последнего выражения в списке.

Clips реализует прямой логический вывод, что обеспечивает высокую степень прогнозирования. При этом, если посылки двух правил одинаковы, то необходимо задавать еще один параметр с определенным числовым значением. Данное значение описывает приоритет выполнения правила.

Оболочка Java Expert System Shell (Jess)

Jess – это оболочка для разработки экспертных систем. Jess изначально являлся производной языка CLIPS, но вскоре вырос в полную, отдельную, динамическую среду. Используя Jess, можно построить Java приложение с возможностью обработки данных на основе знаний, представленных в виде правил [6].

В Jess разрабатываются продукционные модели, основанные на правилах. Система может работать в режиме реального времени и пакетном режиме, который состоит в том, что один и несколько файлов могут работать и выполняться одновременно. Jess также включает в себя возможность создания, управления и вызова методов Java объектов.

Представление знаний и накопление баз знаний в системе Jess происходит с помощью правил и фактов, и необходимо рассмотреть их синтаксис. Синтаксис факта:

```
(deftemplate <name> [<comment>] (slot| multislot <slotname> (<slotoptions>)*)*)
```

Синтаксис и вид записи правила в Jess:

```
(defrule <name> [<comment>]<fact>* => <function>*
```

Jess реализует прямой логический вывод. В основе механизма вывода Jess лежит Rete-алгоритм, который повышает эффективность и быстродействие на задачах множественного сравнения. Rete-алгоритм представляет собой очень быстрое средство сравнения с шаблоном. Он запоминает результат последнего тестирования знаний и заново проверяет только вновь появившиеся факты [6]. Данный алгоритм реализован с помощью построения системы узлов, каждый из которых представляет одно или несколько проверок фактов для каждого правила. Факт, который добавляется или удаляется из базы знаний, обрабатывается этой системой узлов. В основе этой системы лежат конкретные действия. В rete-алгоритме воплощены два эмпирических наблюдения, на основании которых была предложена структура данных, лежащая в его основе.

Оболочка Prolog

Еще одной средой разработки экспертной системы является Prolog. Его особенность и отличие от других языков состоит в том, что он является декларативным языком или языком исчисления предикатов. Предикат – это логическая формула от одного или нескольких

аргументов. Можно сказать, что предикат – это функция, отображающая множество {ложно, истинно}. Поэтому Prolog состоит из набора фактов и правил, которые обеспечивают нахождение решений на основе этих фактов [7].

Синтаксис факта в Prolog:

Имя(Объект).

Более сложное представление и понятие в Prolog – это правила. Они позволяют вывести один факт из других фактов. Правило является истинным заключением, если было найдено другое истинное заключение или факт является истинным.

Синтаксис правил в Прологе имеет две части – заголовок и тело:

Заголовок: — <Подцель>, <Подцель>, ... , <Подцель>.

Тело правила состоит из одной или более подцелей. Подцели разделяются запятыми, определяя конъюнкцию, а за последней подцелью правила следует точка.

Механизм логического вывода Prolog основан на сравнении фактов и является обратным логическим выводом. С помощью подбора ответов на запросы он извлекает хранящуюся (известную) информацию. Он пытается проверить истинность предположения, запрашивая для этого информацию, о которой уже известно, что она истинна. На этом основан принцип резолюции, который требует представления формул исчисления предикатов в виде набора дизъюнктов, связанных операцией конъюнкции. Метод резолюции представляет собой прохождение дерева решений в глубину. Он так же может выбирать из альтернатив и находить возможные решения. Механизм логического вывода Prolog может возвращаться назад и просматривать более одного "пути" при решении всех составляющих задачу частей.

Оболочка Exsys Corvid

Система-оболочка Exsys Corvid представляет собой инструментальное средство, которое может быть использована для разработки экспертных систем для любой предметной области. Она относится к системам дедуктивного продукционного типа, которая поддерживает различные режимы конструирования экспертных систем и способна обрабатывать неопределенности.

Эта система ориентирована на различные классы пользователей в зависимости от их подготовки в области искусственного интеллекта и программирования, имеет развитый интерфейс с современными СУБД и электронными таблицами [8], средства сбора статистики и т. д.

Знания в системе представлены в виде продукционных правил. Основной частью системы является база знаний, которая наполняется по мере работы системы. В базе знаний хранятся правила [9]. Правила могут быть двух типов с одним заключением или с альтернативным заключением, синтаксис правил:

1 тип IF (условие) THEN (заключение);

2 тип IF (условие) THEN (заключение 1) ELSE (заключение 2)

Также в правилах задается коэффициент уверенности, который определяет вероятность выполнения данного правила.

Одной из особенностей системы Exsys Corvid является возможность вывода информации из других правил, что позволяет разбивать сложные задачи на подзадачи.

В системе Exsys Corvid реализованы два механизма логического вывода – прямой и обратной. В Exsys Corvid имеется возможность объяснения полученных результатов. При необходимости, можно выяснить весь ход вычислений, вплоть до исходных данных. Система Exsys Corvid в процессе поиска решения способна выполнять ряд вычислительных задач, но некоторые задачи выходят за рамки его возможностей. В Exsys Corvid имеется возможность

вызова внешних программ для выполнения вычислений. Внешние программы могут передавать данные Exsys Corvid через БД [9].

Оболочка Nexpert Object

Система Nexpert Object фирмы Neuron Data это многоплатформенное средство с набором библиотек с элементами управления баз знаний и оригинальным механизмом двунаправленного вывода с обработкой исключаяющих ситуаций путем различия "сильных" и "слабых" связей [10]. Nexpert представляет собой гибкую инструментальную среду для разработки приложений и экспертных систем. В основе это системы лежат продукции и объектно-ориентированные схемы.

Данная инструментальная оболочка имеет «открытую» архитектуру, которая позволяет ей взаимодействовать с внешними модулями и внешними программами. Внешние модули и программы могут работать с этой системой с помощью правил и объектов. Особенности Nexpert Object являются прямые и обратные цепочки рассуждений. Основные возможности Nexpert Object: способность использовать симметричный формат правил; автоматически генерировать цели; сопоставлять с образцом; интерпретировать; динамически создавать объекты, классы, свойства, методы, множественное и пользовательские наследование; немонотонное рассуждение.

Nexpert включает в себя графический интерфейс, который позволяет разработчикам и экспертам в этой предметной области, изменить правила, а также отображать обзор правил, с использованием динамического и графического механизма просмотра. Базы знаний могут быть разработаны на разных платформах, а после перенесены на конечное место обработки. Nexpert имеет интерфейс прикладного программирования (API), который предоставляет разработчику доступ к функциям Nexpert библиотеки. Nexpert интегрирован с стандартными реляционными базами данных и электронными таблицами на двух видов [10].

Выводы

При анализе существующих инструментальных средств были выявлены следующие достоинства и недостатки.

Достоинства этих инструментальных средств состоят в, что все базы знаний строятся на основе правил, которые записываются в традиционном и удобном для пользователя виде «если .. то ..», имеется возможность работы с нечеткими множествами (Clips), работы в реальном времени, работы с шаблонами, а так же - удобные графические интерфейсы для ввода продукций.

Недостатками являются:

- прямой и обратный логический вывод, который затрачивает много времени на обработку;
- сложный для понимания пользователя механизмы вывода, который предполагает возвраты в определенные узлы, сохранение позиций нахождения правила в базе знаний, несогласованность между организацией поиска в глубину и деревом поиска;
- количество этапов логического вывода может определяться экспоненциальной зависимостью от количества базовых фактов.

Т.о., создаваемая оболочка должна быть лишена названных недостатков и обладать перечисленными достоинствами, обеспечивающими возможность предметной адаптации.

Перспективой дальнейшего развития является:

- создание нового метода построения продукционных баз знаний, удобного и понятного для эксперта в предметной области, что способствует удалению инженера по знаниям из процесса разработки базы знаний;
- разработка нового механизма логического вывода, более усовершенствованного и лишенного названных недостатков.

Список литературы

1. Григорьев А.В. Построение двухсторонних трансляторов в задаче создания интеллектуальных надстроек над проблемно-ориентированными САПР / Григорьев А.В., Морозова О.В. // Сборник трудов XI международной научной конференции им. Т.А. Таран. - Киев: Просвита, 2011. - С. 68-75.
2. Григорьев А.В. Анализ существующих способов создания интерфейса «языки формальных спецификаций — проблемно-ориентированные языки» / Григорьев А.В., Морозова О.В. // Сборник научных трудов донецкого национального технического университета. – Донецк, 2011. – № 14. – С. 270 – 275.
3. Тенденции развития систем искусственного интеллекта [Электронный ресурс] – Режим доступа <http://www.mari-el.ru/mmlab/home/AI/12/index.html>
4. Riley Gary CLIPS online documentation [Электронный ресурс] / Gary Riley – Режим доступа: <http://clipsrules.sourceforge.net/OnlineDocs.html>
5. Частиков, А. П. Разработка экспертных систем. Среда CLIPS / Частиков А.П., Гаврилова Т.А., Белов Д.Л. – СПб.: БХВ-Петербург, 2003. – 608 с.
6. Martin Strauss Jess The Java Expert System Shell / Martin Strauss // Seminar “AI Tools” – 2007. – 33с.
7. Братко И. Алгоритмы искусственного интеллекта на языке Prolog, 3-е издание. : Пер. с англ. — М. : Издательский дом "Вильямс", 2004. — 640 с. : ил. — Парал. Тит. англ.
8. Knowledge Automation Expert System [Электронный ресурс] – Режим доступа: <http://exsyscorvid.com/>
9. Рокотян И.С. Разработка баз знаний на основе экспертной системы EXSYS / Рокотян И.С., Хачатурова Е.А. – М.: Издательство МЭИ, 1998. – 28 с.
10. Nexpert Object Development System[Электронный ресурс] – Режим доступа: http://cseweb.ucsd.edu/users/little/OldSites/CSE_Uptime/v2.8/nexpert.html