

## ИНСТРУМЕНТАЛЬНЫЕ СРЕДСТВА ПОСТРОЕНИЯ ЭКСПЕРТНЫХ СИСТЕМ

**Мокрый Г.В., Фонов А.М., Юшин А.В.**

Донецкий национальный технический университет, кафедра АСУ.

**Abstract.**

*Mokriy G., Fonotov A., Yushin A. The tool means for development expert systems. The tool means for development expert systems effectively allow to create new expert systems and to support existing. Their analysis will allow to reveal the strong and weak sides of resources available at present for construction systems of diagnostics objects with a plenty parameters and to confirm or deny necessity of creation new.*

Компьютеризация общества как одно из основных направлений научно-технического прогресса вызвала существенные изменения в технологии разработки и использования программных средств. Наиболее значимым результатом развития искусственного интеллекта (ИИ) явился переход к новой информационной технологии создания экспертных систем (ЭС). Суть произошедших изменений заключается в появлении нового класса средств, которые позволяют существенно упростить процесс создания ЭС при этом сохранить и даже улучшить возможности таких "интеллектуальных" систем.

Экспертная система - это комплекс технических и интеллектуальных программных средств, который моделирует рассуждения человека-эксперта в некоторой предметной области и использует для этого базу знаний, содержащую факты и правила об этой области, и некоторую процедуру логического вывода.

Основываясь на вышесказанном, на рис.1 представлена структура ЭС [1]:

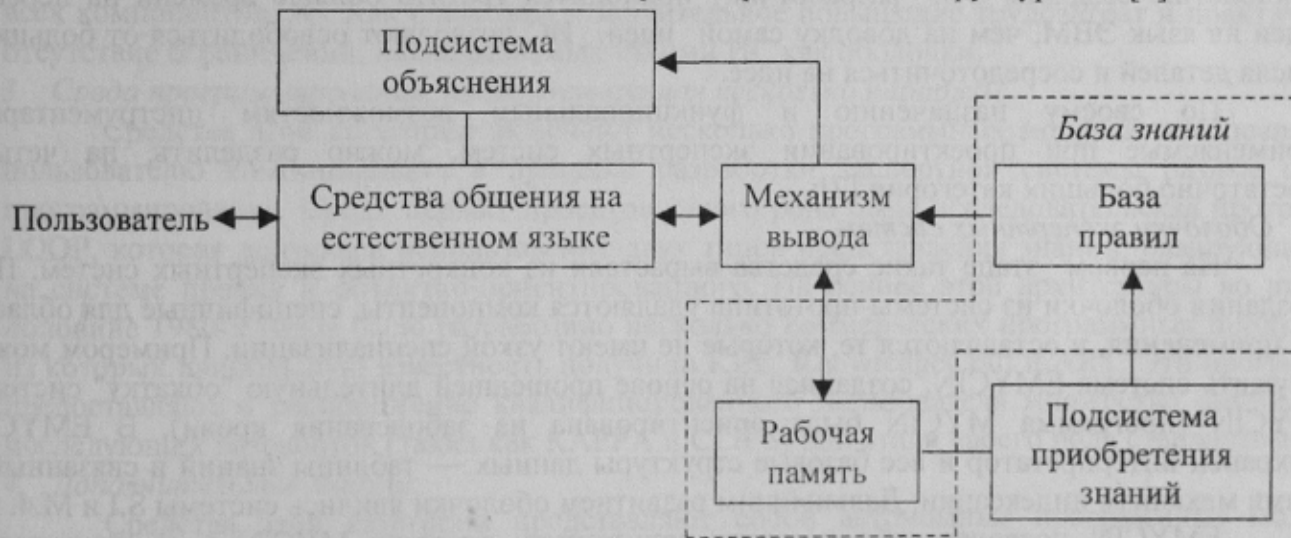


Рисунок 1 – Структура ЭС

- *Память для хранения правил (база правил).* Обнаружив, что анализируемая система неисправна, специалист пытается определить ту ее подсистему, в которой могут быть неполадки. После выявления такой подсистемы он приступает к поиску неисправных компонентов. Этот процесс выглядит так, как будто специалист постоянно обращается к неявно заданному множеству правил, получая промежуточные заключения при выполнении каких-либо условий. Промежуточные заключения в свою очередь становятся условиями для

вывода следующих заключений. Таким образом, наша система, основанная на знаниях, в качестве одного из компонентов должна включать память для хранения правил

- *Рабочая память (база данных)*. В рабочей памяти ЭС хранятся множество фактов, описывающих текущую ситуацию, и все пары атрибут-значение, которые были установлены к определенному моменту. База знаний включает в себя совокупность правил и содержимое рабочей памяти.

- *Механизм вывода (интерпретатор правил)*. Механизм вывода (интерпретатор правил) выполняет две функции:

1. Просмотр существующих фактов и добавление новых;

2. Определение порядка просмотра и применения правил. Он управляет процессом консультации.

- *Подсистема приобретения знаний*. Подсистема приобретения знаний предназначена для добавления в базу знаний новых правил и модификации имеющихся.

- *Средства общения на естественном языке*. Для того чтобы с системой мог взаимодействовать неподготовленный пользователь, в нее требуется включить средства общения на естественном языке.

- *Подсистемы объяснений*. Компонент экспертной системы, который отвечает на вопросы пользователя о том, как именно получено решение, называется подсистемой объяснения. Во время проведения консультаций эта подсистема должна быть способна в любой момент привести обоснование принятого решения.

Процесс создания новых экспертных систем в последнее время эволюционирует, претерпев большие изменения. Уменьшились не только сроки разработки готовых продуктов, но и трудоемкость их создания. Это связано с появлением на рынке искусственного интеллекта таких новых изделий, как инструментальные средства (ИС) создания ЭС. Целью разработки таких средств является облегчение пути от идеи к ее формализованному описанию в доступной для ЭВМ форме. Специфика средств, языков программирования и пользовательских интерфейсов оказывают существенное влияние на технологию создания ЭС: разработчику приходится тратить больше времени на перевод идеи на язык ЭВМ, чем на доводку самой идеи. ИС позволяют освободиться от большего числа деталей и сосредоточиться на идее.

По своему назначению и функциональным возможностям инструментари, применяемые при проектировании экспертных систем, можно разделить, на четыре достаточно больших категории [2]:

#### 1. *Оболочки экспертных систем.*

На первом этапе такие средства выросли из конкретных экспертных систем. При создании оболочки из системы-прототипа удаляются компоненты, специфичные для области ее применения, и оставляются те, которые не имеют узкой специализации. Примером может служить система ЕМУСIN, созданная на основе прошедшей длительную "обкатку" системы МУСIN (программа МУСIN была ориентирована на заболевания крови). В ЕМУСIN сохранен интерпретатор и все базовые структуры данных — таблицы знаний и связанный с ними механизм индексации. Дальнейшим развитием оболочки явились системы S.I и M.4.

ЕМУСIN позволяет использовать архитектуру системы МУСIN в приложении к другим областям немедицины. На базе ЕМУСIN были разработаны экспертные системы как для медицины (например, система PUFF для диагностики легочных заболеваний), так и для других областей знаний, например программа структурного анализа SACON.

Как правило, оболочки содержат набор "инструментов" для реализации различных схем вывода и объяснения. К ним относятся [3]: интерпретаторы и/или компиляторы для обработки знаний, представленных в виде логических выражений, продукционных правил, фреймовых моделей; механизмы логического вывода, обрабатывающие вышеперечисленные модели знаний, и средства контроля за корректностью решения;



средства управления базой данных; средства для извлечения знаний и модификации баз знаний; средства для построения интерфейса с пользователем.

Только немногие современные оболочки ЭС поддерживают этот набор в полной мере. К числу таких систем можно отнести ART, LOOPS, KEE.

Оболочки этих систем являются программами, ориентированными на достаточно узкий класс задач, хотя и более широкий, чем та программа, на основе которой была создана та или иная оболочка. Они приспособлены для работы пользователя-непрофессионала в области программирования.

## 2. Языки программирования высокого уровня.

ИС этой категории избавляют разработчика от необходимости углубляться в детали реализации системы — способы эффективного распределения памяти, низкоуровневые процедуры доступа и манипулирования данными и т.д. Одним из наиболее известных представителей таких языков является OPS5.

Языки высокого уровня являются в руках опытного программиста прекрасным средством быстрого создания прототипа экспертной системы, позволяют обеспечить гибкость процесса разработки при одновременном снижении материальных затрат и сокращении сроков выполнения проекта.

Языки описания порождающих правил (например, OPS5), объектно-ориентированные языки и процедурные дедуктивные системы (например, PROLOG) предоставляют проектировщику экспертных систем большую свободу действий, чем оболочки. Особенно это касается программирования процедур управления и обработки неопределенности, которые не могут быть затем изменены в процессе построения на ее основе конкретной экспертной системы. Та гибкость, которую предоставляют программисту языки высокого уровня, особенно важна при создании экспериментальных систем, в которых заранее выбрать оптимальный режим управления вряд ли возможно.

Построение ЭС на языках высокого уровня требует от разработчика высокой квалификации в области программирования и ИИ. В его задачу входит программирование всех компонентов ЭС. Как следствие — значительное повышение трудозатрат и практически отсутствие ограничений, накладываемых такими ИС как оболочки.

## 3. Среда программирования, поддерживающая несколько парадигм.

Средства этой категории включают несколько программных модулей, что позволяет пользователю комбинировать в процессе разработки экспертной системы разные стили программирования. Среди первых проектов такого рода была исследовательская программа LOOP, которая допускала использование двух типов представления знаний: базирующегося на системе правил и объектно-ориентированного. На основе этой архитектуры во второй половине 1980-х годов было разработано несколько коммерческих программных продуктов, из которых наибольшую известность получили KEE, KnowledgeCraft и ART. Эти программы предоставляют в распоряжение квалифицированного пользователя множество опций. Для последующих разработок (таких как KAPPA и CLIPS) они стали своего рода стандартом.

## 4. Дополнительные модули.

Средства этой категории представляют собой автономные программные модули, предназначенные для выполнения специфических задач в рамках выбранной архитектуры системы решения проблем. Хорошим примером здесь может служить модуль работы с семантической сетью, использованный в системе VT. Этот модуль позволяет отслеживать связи между значениями ранее установленных и новых параметров проектирования в процессе работы над проектом.

Промежуточной категорией между оболочками ЭС и языками программирования высокого уровня являются среды программирования, поддерживающие несколько парадигм.

Многofункциональные программные среды позволяют опытному программисту

экспериментировать при решении новых классов проблем, выбирая подходящие сочетания различных методов, представленных в имеющемся модульном наборе. Поскольку не существует единственного универсального языка представления знаний для произвольной экспертной системы, у разработчиков возникает желание объединить несколько различных схем представления и логического вывода. Поэтому понятно стремление объединить разные методики таким образом, чтобы достоинства одних компенсировали слабости других.

Одной из первых многофункциональных сред искусственного интеллекта является LOOPS, в которой в рамках единой архитектуры обмена сообщениями были объединены четыре парадигмы программирования:

- Процедурно-ориентированное программирование. Она была представлена языком LISP, в котором активным компонентом являются процедуры, а пассивным — данные.
- Программирование, ориентированное на правила. Здесь парадигма аналогична предыдущей, но роль процедур играют правила "условие-действие".
- Объектно-ориентированное программирование. Структурированные объекты обладают свойствами и процедур, и данных, причем побочные эффекты обычно локализируются в пределах объекта.
- Программирование, ориентированное на данные. Доступ к данным и обновление данных запускает определенные процедуры, причем не имеет значения, почему изменен компонент данных, — то ли это результат побочного эффекта, то ли результат действия других процедур.

В рамках основной объектно-ориентированной парадигмы модули среды, поддерживающие разные стили программирования, можно комбинировать. Обычно условия в порождающих правилах и логические фразы связываются со значениями слотов структурированных объектов, а правила модифицируют значения этих слотов. Именно такой стиль объединения парадигм в настоящее время реализован в языке CLIPS.

CLIPS (C Language Integrated Production System) является одним из распространенных ИС разработки экспертных систем. Разработчиком CLIPS является Национальное Аэрокосмическое Агентство США. Первая версия системы вышла в 1984 г.

CLIPS использует производственную модель представления знаний и поэтому содержит 3 основных элемента [4]: список фактов, базу знаний, блок вывода.

В CLIPS используется оригинальный LIPS-подобный язык программирования, ориентированный на разработку ЭС. Кроме того, CLIPS поддерживает еще две парадигмы программирования: объектно-ориентированную и процедурную.

Основными элементами написания программ на CLIPS являются [4]: простые типы данных; функции для манипулирования данными; конструкции для пополнения базы знаний.

Для представления информации в CLIPS предусмотрено восемь простых типов данных, обеспечивающих хранение численных и символьных данных.

Под функцией в CLIPS понимается фрагмент исполняемого кода, с которым связано уникальное имя и который возвращает полезное значение или имеет полезный побочный эффект. Существует несколько типов функций. Пользовательские и системные функции — это фрагменты кода, написанные на внешних языках (например, на C) и связаны со средой CLIPS. Системными называются те функции, которые были определены изначально внутри среды CLIPS, а пользовательскими — вне CLIPS.

В CLIPS существует несколько описывающих конструкций. Определение конструкции в точности направлено на изменение состояния среды путем внесения изменений в базу знаний CLIPS.

Факты являются одной из основных форм представления информации в системе CLIPS. Каждый факт представляет фрагмент информации, который был помещен в текущий список фатов. Факт представляет собой основную единицу данных, используемую правилами. Факты могут добавляться к списку фактов, удаляться из него, изменяться и



дублюватися самим користувачем або програмою.

Одним из основных методов представления знаний в системе CLIPS являются правила. Правила используются для представления эвристик, определяющих ряд действий, которые необходимо выполнить в определенной ситуации. Каждое правило состоит из двух частей: антицедента (условия) и консеквента (заключения), который записывается справа. В CLIPS принято считать, что условие выполняется, если соответствующий ему факт присутствует в списке фактов. Одним из типов условных элементов может быть образец, состоящий из набора ограничений, которые используются для описания того, какие факты удовлетворяют условию, определяемому образцом. Процесс сопоставления фактов и образцов выполняется блоком вывода CLIPS, который автоматически сопоставляет образцы, исходя из текущего состояния фактов, и определяет, какие из правил являются применимыми. Если все условия правила выполняются, то оно активизируется и помещается в список активных правил.

Правая часть правила представляет собой совокупность действий, которые должны быть выполнены, если правило применимо. Действия, описанные в применимых правилах, выполняются тогда, когда блок вывода CLIPS получает команду начать выполнение применимых правил. Если существует множество применимых правил, то для того, чтобы выбрать правило, действия которого должны быть выполнены, блок вывода использует стратегию разрешения конфликтов. Действия, описанные в выбранном правиле, выполняются (при этом список применимых правил может измениться), а затем блок вывода выбирает другое правило и т.д. Этот процесс продолжается до тех пор, пока не останется ни одного применимого правила, т.е. пока список активированных правил не окажется пуст.

Использование объектно-ориентированных средств в CLIPS позволяет значительно упростить программирование правил, поскольку для обновления данных можно применять механизм передачи и обработки сообщений методами классов. Описание класса состоит из [2]:

- системных слотов, необходимых объектно-ориентированной надстроке CLIPS
- слоты, представляющие свойства и члены данных класса.

Таким образом, в рамках единой CLIPS программы "уживаются" правила и объекты. Правила управляют ходом вычислений, но некоторые операции объекты выполняют самостоятельно, получив сообщение от правил. Объекты не являются резидентами рабочей памяти, но члены левой части правил (условий) могут быть сопоставлены с содержимым их слотов. Объекты не могут самостоятельно активизировать правила, но их обработчики сообщения могут возвращать определенную информацию о результатах, которая используется для управления логикой выполнения действий в правой части правил.

Среди других ИС построения ЭС необходимо выделить интегрированную среду G2. В 1988 г. частная фирма Gensym вышла на рынок с ИС под названием G2, версия 1.0. В настоящее время функционирует версия 4.2 и готовится к выпуску версия 5.0. Фирме Gensym сегодня принадлежат 50% мирового рынка экспертных систем, используемых в системах управления [5].

Классы задач, для которых предназначена система G2 и подобные ей системы: мониторинг в реальном масштабе времени; системы управления верхнего уровня; системы обнаружения неисправностей; диагностика; составление расписаний; планирование; оптимизация; системы - советчики оператора; системы проектирования.

Немалую долю успеха фирме Gensym обеспечивают основные принципы, которых она придерживается в своих новых разработках: проблемно/предметная ориентация; следование стандартам; независимость от вычислительной платформы; совместимость снизу-вверх с предыдущими версиями; универсальные возможности, не зависящие от решаемой задачи; обеспечение технологической основы для прикладных систем; комфортная среда разработки; поиск новых путей развития технологии; распределенная

архитектура клиент-сервер; высокая производительность.

Основным достоинством оболочки экспертных систем G2 является возможность применять ее как интегрирующий компонент, позволяющий за счет открытости интерфейсов и поддержки широкого спектра вычислительных платформ легко объединить уже существующие, разрозненные средства автоматизации в единую комплексную систему управления.

Несмотря на то, что первая версия системы G2 появилась не так давно - в 1988 г., ее даже в богатой Америке никто не назовет дешевой. G2 можно назвать бестселлером на рынке программных продуктов - на начало 1996 г. в мире было установлено более 5000 ее копий. Фирма Gensym обслуживает более 30 отраслей - от аэрокосмических исследований до производства пищевых продуктов. 25 самых крупных индустриальных мировых корпораций используют G2. На базе G2 написано более 500 действующих приложений.

Чем же объясняется успех инструментального комплекса G2? Прежде всего G2 - динамическая система в полном смысле этого слова. G2 - это объектно-ориентированная интегрированная среда для разработки и сопровождения приложений реального времени, использующих базы знаний. Система G2 реализована на всех основных вычислительных платформах, включая рабочие станции Sun, HP9000, RS/6000 и ПЭВМ, работающие под управлением Windows NT. Возможна работа с системой в режиме клиент—сервер в сети Ethernet [6].

Одним из перспективных и динамично развивающихся направлений ИИ является технология искусственных нейронных сетей. Такие сети позволяют решать задачи классификации образов, кластеризации/категоризации, аппроксимации функций, диагностики/прогнозирования, оптимизации, управления. Причем, благодаря своим особенностям, нейронная сеть решает неформализуемые или трудно формализуемые задачи. В разрезе построения ЭС нейронные сети являются очень мощным инструментом.

Под нейронными сетями подразумеваются вычислительные структуры, которые моделируют простые биологические процессы, обычно ассоциируемые с процессами человеческого головного мозга [7]. Элементарным преобразователем в данных сетях является искусственный нейрон или просто нейрон, названный так по аналогии с биологическим прототипом - биологической нервной клеткой, которая обрабатывает информацию. Таким образом, искусственная нейронная сеть представляет собой совокупность нейроподобных элементов, определенным образом соединенных друг с другом и с внешней средой с помощью связей, определяемых весовыми коэффициентами.

В настоящее время известно много удачных примеров применения нейросетевого подхода для построения интеллектуальных информационных систем и, в частности, экспертных систем.

Комбинированное использование экспертной системы и аппарата искусственных нейронных сетей обеспечивает необходимую гибкость и самообучение на основе знаний, в то же время, полученные от экспертов знания позволяют существенно упростить структуру нейронных сетей, уменьшить число нейронов и связей в сети.

Например, медицинские нейросетевые экспертные системы проявили себя как серьезный соперник традиционных экспертных систем, составляя конкуренцию квалифицированным экспертам. Исследования в Боткинской больнице (С.-Петербург), проведенные с использованием разработанной нейросетевой экспертной системы, показали ее превосходные возможности по диагностике некоторых классов болезней, которые плохо диагностируются врачами: внезапные и беспричинные боли в спине, неопределенные боли в брюшной полости и т.д. [7]. Результаты проверки свидетельствуют о высокой достоверности результатов, достигаемых такими системами (94%).

Для практического использования аппарата нейронных сетей существует достаточное количество соответствующих программ, обладающих такими основными функциями:



формирование нейронной сети, обучение нейронной сети, тестирование обученной нейронной сети. К наиболее мощным и распространенным относят: NeuroSolutions, NeuralWorks Professional, Process Advisor, NeuroShell, BrainMaker. Результаты тестирования и сравнения приведены в [7].

Обзор существующих ИС разработки ЭС показывает, что на этом рынке присутствуют программные продукты, которые отличаются сферой использования, кругом решаемых задач и функциональными возможностями. Они позволяют разработчикам сосредоточиться на идее и освободиться (в большей или меньшей степени) от деталей. К основным достоинствам таких средств относят:

- появляется возможность создания ЭС непрофессионалами – программистами;
- сокращается время разработки;
- большее удобство изменения и поддержки ЭС.

Перечисленные достоинства делают использование существующих ИС вполне оправданным в тех случаях, когда в минимальные сроки необходимо разработать ЭС в узкой предметной области. Разработку сложных ЭС ограничивают функциональные возможности этих продуктов.

К основным недостаткам большинства имеющихся инструментариев ЭС (за редким исключением) отнесем отсутствие связи с внешними современными СУБД – возможными хранилищами исходных данных и результатов. Частично это можно объяснить быстрым развитием технологий СУБД и достаточной давностью разработок ИС (основная часть была создана в 80-х – начале 90-х). Другим недостатком промышленных инструментариев ЭС как конечных программных продуктов является их высокая стоимость.

Возможность подключать внешние современные хранилища данных позволит включать ЭС в автоматизированные системы управления, например, в АСУ ТП, как подсистему анализа состояний сложных и многопараметричных объектов.

Такой значительный пробел в существующих ИС подводит к необходимости разработки нового инструментария ЭС, который включит в себя разумную часть уже имеющихся продуктов и позволит эффективнее использовать современные информационные технологии.

### *Литература*

1. Таунсенд К., Фохт Д. Проектирование и программная реализация экспертных система на персональных ЭВМ. – М.: Финансы и статистика, 1990. – 320 с.
2. Джексон Питер. Введение в экспертные системы. – Издательский дом "Вильямс", 2001. – 624 с.
3. Криевич В.С., Кузьмич Л.А., Шиф А.М и др. Экспертные системы для персональных компьютеров: методы, средства, реализации. Справочное пособие. – Мн.: Высшая школа, 1990. – 197 с.
4. Корнеев В.В., Гареев А.Ф., Васютин С.В., Райх В.В. Базы данных. Интеллектуальная обработка информации. – М.: "Нолидж", 2000. – 352 с.
5. Попов Э.В., Фоминых И.Б., Кисель Е.Б., Шапот М.Д. Статические и динамические экспертные системы. – М.: Финансы и статистика, 1996. – 320 с.
6. Гаврилова Т.А., Хорошевский В.Ф. Базы знаний интеллектуальных систем. – СПб.: Питер, 2001. – 384 с.
7. Круглов В.В., Борисов В.В. Искусственные нейронные сети. Теория и практика. - М.: Горячая линия – Телеком, 2001. – 382 с.