

Инструментальные средства для построения встраиваемых продукционных экспертных систем.

В наши дни значительный объем всего разрабатываемого программного обеспечения составляют информационные системы для автоматизации той или иной деятельности. Многие из таких систем разрабатываются для некоторой узкоспециализированной предметной области, что создает идеальную почву для использования в них интеллектуальной компоненты в виде экспертной системы. В качестве примера можно привести медицинскую учетно-диагностическую систему, в которой функции учета симптомов и результатов анализов пациентов удобно совместить с диагностическими функциями.

Для разработки экспертных систем применяются различные представления знаний и стратегии вывода, однако можно утверждать, что безусловным лидером является продукционная модель. При построении реальных систем, как правило, используется некоторая оболочка, содержащая процессор вывода и средства диалогового интерфейса с пользователем, и основной задачей является построение базы знаний. Для построения т.н. встраиваемых экспертных систем, функционирующих как составная часть некоторых программных продуктов, в большинстве случаев необходимы специализированные средства.

Автором разработан и реализован набор инструментальных средств для построения встраиваемых экспертных систем. Поддерживаются два основных способа встраивания: высокоуровневая кодогенерация и использование специализированной внешней оболочки. В обоих случаях используется одинаковое представление знаний и синтаксис правил, что позволяет применять оба способа для одной и той же базы знаний. Используется простая продукционная модель представления знаний с парами атрибут-значение и обратная стратегия вывода. Такое простое представление с одной стороны упрощает реализацию и является достаточно популярным, но с другой – ограничивает выразительность, хотя и оставляет простор для применения в широком диапазоне задач.

Высокоуровневая кодогенерация состоит в построении по исходным правилам экспертной системы программного кода на языке высокого уровня. Разработанный инструментарий поддерживает языки Си/Си++, Java и Object Pascal/Delphi, что позволяет применять такой подход для широкого спектра средств разработки. Сгенерированный код представляет собой систему взаимно-рекурсивных функций (по одной на каждый атрибут базы знаний), при этом вызов целевой функции приводит к рекурсивному вызову, аналогичному процессу обратного вывода в экспертной системе. Такой подход особенно удобен для небольших баз знаний, при которых объем сгенерированного кода не слишком велик и не ведет к падению эффективности. Кроме того, после сборки проекта интеллектуальная часть становится неотделимой от исполняемого модуля системы, что позволяет скрывать собственно текст базы знаний, однако делает его немодифицируемым.

Другой подход состоит в применении разработанной автором специализированной оболочки для распределенной консультации diet, вызываемой из основного модуля системы. В этом случае интерфейс с интеллектуальной частью оформляется в виде вызовов DLL-библиотеки или ActiveX-компонента, что позволяет использовать такой подход в большинстве средств разработки для Win32. Оболочка diet поддерживает распределенное накопление и использование знаний, что, в частном случае, допускает построение конфигураций из нескольких информационно-интеллектуальных систем, использующих общий централизованный сервер экспертного вывода.

Описанная технология была успешно применена автором в реализации интеллектуальной информационной системы по учету и диагностике больших заболеваниями предстательной железы, внедренной в клиническую практику урологической клиники ГКБ им. С.П.Боткина.

Рассмотренный подход предоставляет средства для интеграции как небольших экспертных систем, так и более широкомасштабных клиент-серверных решений в информационные системы, и может быть полезен для построения широкого класса приложений.