

- сервер 1С:Предприятие;
- сервер базы данных MS SQL Server.

На сервере осуществляются запросы к базе данных, запись данных, проведение документов, различные расчеты, формирование отчетов и «выходных» документов, подготовка форм к отображению. Клиентское программное обеспечение выполняет функции по получению и открытию форм, организации ввода, просмотру и редактированию данных.

В состав выходных документов включены:

- учебный план группы по специальности;
- детализированные и сводные данные по распределению учебной нагрузки;
- сведения, подаваемые диспетчеру факультета;
- сведения для индивидуальных планов преподавателей;
- различные виды отчетности.

Конечное приложение (конфигурация) расчета нагрузки имеет возможность расширения функциональных возможностей.

## **ФОРМАЛИЗАЦИЯ ЗНАНИЙ СРЕДСТВАМИ АЛГЕБРЫ КОНЕЧНЫХ ПРЕДИКАТОВ И РЕЛЯЦИОННЫХ СЕТЕЙ**

**Мамедов А.А., Шабанов-Кушнаренко С.Ю. (Украина, Харьков, ХНУРЭ)**

Возникновение множества взаимосвязанных и основанных на знаниях процессов обработки информации приводит к возрастанию уровня сложности задач, требующих применения методов искусственного интеллекта. Решение таких задач требует применения новых знаний. Существующие подходы к формализации неявных знаний основаны, в частности, на проведении интервью с экспертами и группами экспертов, работе с фокус-группами, что не всегда позволяет получить однозначные результаты. Однако такие зависимости могут быть получены на основе анализа последовательности событий, отражающих последовательность действий процесса с привязкой ко времени.

Использование знаний в интеллектуальных системах основывается на формализованном описании составляющих их элементов, а также организации таких элементов в единую структуру в соответствии с заданными правилами. Такая структура должна обеспечивать эффективное использование знаний.

Знания основываются на данных, которые рассматриваются как множество разрозненных фактов [1]. Сами по себе данные обладают ограниченной применимостью, поэтому их необходимо структурировать, выявить и понять связи между ними, чтобы получить информацию. Интерпретация, оценка этой информации позволяет получить знания. Иными словами, знания отражают связи уже между некоторыми шаблонами представления информации. Затем знания интегрируются, из совокупности знаний выделяются принципы построения знаний и, на их основе устраняются противоречия между знаниями. Как следствие, полученные интегрированные знания содержат явные причинно-следственные связи между представленными на основе шаблонов фактами и потому легко понимаемы и интерпретируемы.

Основные модели представления знаний включают в себя: правила продукции; семантические сети; логические модели; фреймовые модели; скрипты; концептуальные графы; объектно-ориентированные модели; средства вычислительного интеллекта.

Логические модели основаны на представлении знаний в виде набора правильно построенных формул в выбранной формальной системе. Так, знания в логике предикатов представляются логическими формулами, с использованием предварительно выбранных констант, функциональных и предикатных символов. Константы определяют объекты предметной области. Функциональные и предикатные символы определяют функциональные зависимости и отношения между объектами.

Логические модели обладают следующими преимуществами:

- представление знаний в виде единой формальной системы, обеспечивающей математически строгий вывод;
- возможность формальной проверки несогласованности, а также неполноты знаний;
- процедурные и декларативные знания могут быть описаны единым образом.

Для формализации знаний нами был избран математический аппарат алгебры конечных предикатов (АКП) и реляционных сетей. Такой выбор обусловлен тем, что математика информационных процессов имеет преимущественно не числовую, а логическую природу, а также рядом положительных свойств аппарата АКП. Это дискретный аппарат для описания многозначных функций (отношений). В качестве алфавита в АКП используются многозначные символы. В АКП развиты средства для формульной записи любых конечных отношений. Доказано, что АКП полна, то есть на ее языке могут быть описаны любые конечные отношения.

Литература:

1) Aalst, W.M.P. van der Process mining in web services : the websphere case / Aalst, W.M.P. van der, Verbeek, // Bulletin of the IEEE Computer Society Technical Committee on Data Engineering. – 2008. - №31 (3), С. 45-48.

## **ВИРТУАЛЬНОЕ ИЗУЧЕНИЕ МИКРОКОНТРОЛЛЕРОВ AVR**

**Матвеев И.П. (Республика Беларусь, Минск, БГАТУ)**

Различные системы автоматизированного управления производственными и технологическими процессами, как правило, включают электронные схемы с использованием микроконтроллеров различных типов. Однако изучение и отладка работы реальных контроллеров оказывается затратной задачей, так как недостаточно только написать программу в определенной среде, необходимо записать в него разработанную программу, подключить к выходу контроллера исполнительные устройства и только тогда наглядно увидеть результат своей работы.

Изучение микропроцессорной техники осуществляется на базе однокристалльных микроконтроллеров семейства AVR, которые приобрели большую популярность в настоящее время, привлекая разработчиков удобными режимами программирования, доступностью программно-аппаратных средств поддержки и широкой линейкой выпускаемых типов. Изучение таких микроконтроллеров можно осуществить с помощью программы AVR Studio 6 и программы Proteus.

AVR Studio 6 предоставляет возможность осуществлять разработку и отладку программ для микроконтроллеров AVR фирмы ATMEL, поддерживает большое количество средств программирования и отладки.

Программы пишутся на языке ассемблер (Assembler), поддерживается также язык программирования Си. При работе с ассемблером нет необходимости в непосредственном соединении с микроконтроллером. Задача состоит в том, чтобы микроконтроллер принял информацию, обработал по заданному алгоритму и выдал результат в понятной форме. В простейшем случае, чтобы увидеть результат работы микроконтроллера, к его выходным портам подключают светодиоды, которые должны загораться в соответствии с алгоритмом. Но можно моделировать и более сложные устройства.

После создания проекта (программы) в среде AVR Studio 6 необходимо перейти к программе Proteus (by Labcenter Electronics), которая представляет собой симулятор принципиальных электронных схем. Сначала собирается схема, включающая контроллер выбранного типа, вспомогательные элементы, исполнительные устройства. Proteus содержит большую библиотеку электронных компонентов. Когда схема собрана с помощью программы Proteus можно проверить работу спроектированной электрической схемы (рис.1).