

Работа системы макетировалась с использованием языков высокого уровня (Pascal, Delphi). Система Windows-ориентирована. Среда пользователя - интегрированная, многооконная, русифицированная. Получаемый в результате разработки текст алгоритма позволяет автоматически генерировать код (функционально-адекватное внутреннее представление алгоритма), исполняемый, например, внутренней виртуальной машиной, которая и реализует интерфейс взаимодействия пользователя с алгоритмом в процессе его разработки и отладки.

Литература

1. Муравьев, Г.Л. Входной язык автоматизированной системы обучения (АСО) алгоритмизации // Новые информационные технологии в образовании: материалы 3 международной конференции НИТЕ' 98, Минск, 1998 г.: в 3 т. / БГЭУ. – Минск, 1998. - Т.2. - С. 83–88.

2. Муравьев, Г.Л. Автоматизация обучения алгоритмизации и программированию / Г.Л. Муравьев, С.В. Мухов // Вести ИСЗ. - 2000. – № 3. - С. 24-29.

УДК 004.6

ОБ ОРГАНИЗАЦИИ СИСТЕМЫ ТЕСТИРОВАНИЯ

Ванюков С.В.

*УО «Брестский государственный университет им. А.С.Пушкина», г. Брест
Научный руководитель – Силаев Н.В., доцент*

Начиная разработку системы тестирования TENMA, была поставлена задача создать, по возможности, универсальный инструмент, который можно будет легко расширить и приспособить для любой системы тестирования. В качестве инструмента разработки был выбран язык С# на платформе Visual Studio .NET 2.0. Предоставляемые ею средства для сетевого программирования, построения классов и шифрования передаваемых данных оказались подходящими для решения поставленной перед нами задачи.

Основное внимание при проектировании уделяется открытости системы и простоте её обновления и инсталляции. Разрабатываемая система позволит: строить древовидно организованные структуры связей, моделирующие диалог естественного общения при опросе; обеспечить поддержку импорта тестов формата TQF, который применяется в настоящее время в системе теоретического тестирования на математическом факультете БрГУ.

Пакет состоит из двух частей — клиентской и серверной. Клиентская часть инсталлируется на все компьютеры, за которыми производят тестирование. Серверная часть устанавливается только на один – компьютер администрирования.

Общие требования, выдвигаемые нами к системе, таковы:

Удобство администрирования.

Совместимость. Система должна поддерживать уже существующие форматы тестов или поставляться с конвертером.

Полнота. Всё необходимое для работы системы должно быть включено в её дистрибутив, чтобы до предела упростить процесс инсталляции.

Универсальность. Система должна поддерживать различные типы вопросов и ответов. Например, вопрос может содержать иллюстрации и текстовое форматирование, ответ должен быть только одним из предложенных, набором из нескольких предложенных вариантов или вводиться с клавиатуры.

Лёгкость тестирования.

Защита данных. Обе составляющие системы (клиентская и серверная) должны быть устойчивы к «атакам» различного рода.

Мобильность.

Отказоустойчивость. Система должна чётко и грамотно реагировать на сбои в работе сети. В случае ошибки и администратор, и пользователи должны чётко знать возникшую проблему.

УДК 004.891.2

ПРОГРАММА XPERTRULE KNOWLEDGE BUILDER КАК ИНСТРУМЕНТ ПОСТРОЕНИЯ ДЕРЕВЬЕВ РЕШЕНИЙ

Геращенко Е.В.

УО «Белорусский государственный экономический университет», г. Минск

Стремительное развитие информационных технологий, в частности прогресс в методах сбора, хранения и обработки данных, позволил многим организациям собирать огромные массивы данных, которые необходимо анализировать. Объёмы этих данных настолько велики, что возможностей экспертов уже не хватает, что породило спрос на методы автоматического исследования (анализа) данных.

Деревья решений – один из таких методов. Первые идеи создания деревьев решений восходят к работам Ховленда (Hoveland) и Ханта (Hunt) конца 50-х годов XX века. Однако основополагающей работой, давшей импульс для развития этого направления, явилась книга Ханта (Hunt, E.V.), Мэрина (Marin J.) и Стоуна (Stone, P.J) "Experiments in Induction", увидевшая свет в 1966г.

Деревья решений – это способ представления правил в иерархической, последовательной структуре, где каждому объекту соответствует единственный узел, дающий решение.

Область применения «деревья решений» в настоящее время широка, но все задачи, решаемые этим аппаратом, могут быть объединены в следующие три класса:

- Описание данных: Деревья решений позволяют хранить информацию о данных в компактной форме, вместо них мы можем хранить дерево решений, которое содержит точное описание объектов.
- Классификация: Деревья решений отлично справляются с задачами классификации, т.е. отнесения объектов к одному из заранее известных классов. Целевая переменная должна иметь дискретные значения.
- Регрессия: Если целевая переменная имеет непрерывные значения, деревья решений позволяют установить зависимость целевой переменной от независимых(входных) переменных. Например, к этому классу относятся задачи численного прогнозирования(предсказания значений целевой переменной).

На сегодняшний день существует значительное число алгоритмов, реализующих деревья решений CART, C4.5, NewId, IRule, CHAID, CN2 и т.д.