

2. Карданова Е. Ю., Карпинский В. Б. Специальные методы анализа результатов тестирования, основанные на свойстве объективности моделей Раша // Информационные технологии. 2008. № 4 (140). С. 72—80.

3. Карданова Е. Ю. Моделирование и параметризация тестов: основы теории и приложения. М.: Федеральный центр тестирования, 2008. 304 с.

4. Карданова Е. Ю., Карпинский В. Б. Технология обработки информации в многокритериальном мониторинге на основе политомической модели Раша // Системы управления и информационные технологии. 2007. № 3. 1 (29). С. 149—154.

5. Smith R. M. Person and Item Analysis. Chicago: Mess Press, 1992.

6. Wright B. D., Masters G. N. Rating Scale Analysis. Rasch Measurement. Chicago: Mesa Press, 1979. 206 p.

7. Карпинский В. Б. Исследование эффективности общих статистик согласия для обнаружения искажений при массовом

тестировании // Вопросы тестирования в образовании. 2006. № 1 (17). С. 7—14.

8. Карданова Е. Ю., Карпинский В. Б. О возможностях обнаружения искажений при массовых тестированиях // Моделирование и параметризация педагогических тестов. Матер. Международной конференции, Минск, Беларусь, 2007. С. 30—36.

9. Карданова Е. Ю., Карпинский В. Б. Обнаружение искажений при тестировании с использованием математической модели Г. Раша // Обзорные прикладной и промышленной математики. 2007. Т. 14. Вып. 4. С. 716—717.

10. Smith R. M. The Distributional Properties of Rasch Standardized Residuals // Educational and Psychological Measurement. 1988. V. 48 P. 657—667.

11. Вероятностные разделы математики / Под ред. Ю. Д. Максимова. СПб.: Иван Федоров, 2001. 588 с.

12. Кобзарь А. И. Прикладная математическая статистика М.: Физматлит, 2006. 814 с.

УДК 004.588 + 004.384 + 004.415.2

В. Е. Зюбин, канд. техн. наук, ст. научн. сотр.,  
Институт автоматизации и электрометрии СО РАН,  
г. Новосибирск,  
Новосибирский государственный университет,  
e-mail: zyubin@iae.nsk.su

## Использование виртуальных объектов для обучения программированию информационно-управляющих систем

*Представлен метод создания виртуальных лабораторных стендов для обучения программированию управляющих алгоритмов в области промышленной автоматизации. В качестве языка программирования использован язык Рефлекс (также известный как "Си с процессами"). Имитаторы контролируемых технологических объектов создаются в среде LabVIEW. Управляющие алгоритмы, описанные на языке Рефлекс, преобразуются к формату, позволяющему интегрировать алгоритм в среду LabVIEW через механизм Formula Node. Предлагаемый подход направлен на повышение эффективности процесса обучения, поскольку позволяет студентам исключить из рассмотрения рутинные вопросы физического моделирования и сконцентрироваться на концептуальном уровне программирования управляющих алгоритмов.*

**Ключевые слова:** обучение программированию, программные имитаторы, управляющие алгоритмы, язык Рефлекс.

### Введение

Качество образования, получаемого студентом, определяется не только уровнем теоретической подготовки, но и умением использовать полученные знания на практике. Практические навыки студент приобретает через специально предусматриваемые в учебном плане лабораторные работы и семинарские занятия. Достижимые

при этом эффекты — снижение "порога вхождения" в изучаемую область, сокращение времени на освоение материала, повышение уровня понимания теоретических положений. Для повышения уровня практической подготовки научных и технических специалистов в последние годы все более широкое применение находят информационные технологии, в частности, при организации лабораторных практикумов [1].

При изучении языков программирования, ориентированных на создание управляющих алгоритмов, требуется, чтобы лекционные курсы по информационно-управляющим системам (ИУС) были подкреплены лабораторно-практическими занятиями по решению типовых задач из области промышленной автоматизации. Однако лабораторные стенды, предполагающие физическое моделирование технологического процесса даже в упрощенном виде, означают серьезные финансовые затраты на их создание и поддержку в работоспособном состоянии. При таком подходе необходимо не только разработать и реализовать модель технологического процесса, но также оснастить ее датчиками, исполнительными органами и обеспечить сопряжение с компьютером через модули ввода-вывода. Стенды громоздки, занимают много места, а ошибки, допускаемые студентами при работе с ними, зачастую приводят к выходу стенда из строя.

В этих условиях наиболее перспективный способ организации практических занятий для студентов по курсам промышленной и физико-технической автоматизации должен быть основан на использовании программных имитаторов — виртуальных объектов управления (ВОУ). По сравнению с физическими моделями ВОУ имеют ряд очевидных преимуществ, дающих существенное сокращение материальных и временных затрат на создание, тиражирование и сопровождение лабораторных стендов.

В статье излагается методика создания лабораторных стендов на базе ВОУ, ориентированных на изучение основ программирования ИУС. Структурно статья представлена тремя разделами. В первом разделе с учетом особенностей изучения основ программирования ИУС формулируются требования к лабораторным стендам. Во втором разделе проводится анализ существующих средств моделирования и обсуждаются использованные программные и технологические средства создания ла-

бораторного стенда. Третий раздел посвящен вопросам реализации тестового лабораторного стенда.

#### Концепция виртуального лабораторного практикума по программированию информационно-управляющих систем

В качестве базовых требований к лабораторному стенду были выдвинуты следующие требования, обеспечивающие разумный баланс между наглядностью создаваемых имитаторов робототехнических комплексов, развитостью средств программирования алгоритмов управления и трудоемкостью реализации.

Требования, возникающие на этапе создания нового стенда:

- возможность создания анимированных 2D-объектов;
- допустимость использования при моделировании объекта автоматизации нескольких слоев с изображениями;
- наличие поведения у ВОУ, т. е. динамическая реакция ВОУ на внешние воздействия;
- возможность подключения стороннего алгоритмического блока, создаваемого студентом;
- стандартный интерфейс подключения стороннего алгоритмического блока, исключающий коррекцию ВОУ, а равно связей между ВОУ и сторонним алгоритмическим блоком в случае изменений алгоритма, создаваемого студентом.

Требования к этапу практической работы студента со стендом:

- наличие ручного и автоматического режима управления ВОУ;
- использование языка программирования, ориентированного на управляющие алгоритмы, для создания алгоритма управления ВОУ;
- наличие диагностической и отладочной информации.

#### Базовые программные и технологические средства

Чрезвычайно привлекательная идея использовать концепцию ВОУ для создания лабораторного практикума осложняется отсутствием программных средств, ориентированных на имитационное моделирование объектов автоматизации. Схожая задача создания операторских тренажеров решается на практике с серьезными трудозатратами: либо с использованием SCADA-пакетов, например InTouch (Wonderware), Shadow Plant (Honeywell), либо, чаще всего, вообще без использования специализированных пакетов, на языках Си/Си++ [2–5], явно не предназначенных для решения таких задач. При таком подходе цена тренажера может достигать 1 млн долл. [4]. Большинство широко известных языков имитации, таких как ARENA, Extend, WITNESS, QUEST, Enterprise Dynamics, Any-Logic и др., не имеют простых и мощных механизмов включения в модель правил и алгоритмов принятия решений [5], что не позволяет создавать на них "поведенческие" модели.

В связи с этим вполне определенный интерес представляют пакеты прикладных программ технических вычислений и системы автоматизации научных исследований типа MatLab и LabVIEW, широко используемые, в частности, как средства имитационного моделирования [6].

В результате анализа возможных претендентов на роль базовой среды программирования выбор был сделан в пользу пакета LabVIEW [7]. Хотя LabVIEW позиционируется как средство разработки программно-аппаратных комплексов для тестирования, измерения, ввода данных,

анализа и управления внешним оборудованием, с точки зрения решаемой задачи пакет имеет целый ряд привлекательных свойств. Интерфейс пользователя позволяет не только отображать результаты в виде графиков и целого спектра графических элементов, но также работать с изображениями в tiff и bmp форматах. Специальный механизм (блок-диаграмм) дает возможность программировать функции управления объектами графического интерфейса. Вполне очевидный интерес представляет и возможность оформить создаваемые имитационные модели в виде автономных модулей (.EXE) и в виде совместно используемых динамических библиотек (.DLL), предоставляемая в LabVIEW Pro, поскольку эта возможность позволяет тиражировать создаваемые стенды и исполнять их автономно от среды разработки. Выбор обусловлен также популярностью LabVIEW и широким использованием в обучающем процессе, в частности, для создания лабораторных стендов. И хотя такое использование затрагивает в основном аналоговые случаи (электротехника, оптика, термодинамика и т. п. [8]), опыт показывает, что средствами LabVIEW можно создавать и несложные поведенческие алгоритмы, в частности, дискретные и событийно-управляемые (см., например, [9]).

В качестве языка программирования алгоритмов управления был выбран язык Рефлекс [10]. Язык ориентирован на программирование управляющих алгоритмов в промышленной автоматизации и робототехнике: для систем, предполагающих активное взаимодействие с внешней средой, технологическим оборудованием, физическими процессами через датчики и органы управления. Язык по синтаксису очень похож на язык Си (язык известен также под именем "Си с процессами"), что обеспечивает простоту его изучения большинством практикующих программистов. Язык имеет англоязычный и русскоязычный синтаксис, а также допускает идентификаторы на русском языке, и это делает его крайне привлекательным для русскоязычных пользователей. В отличие от языка Си, где программы строятся как иерархия функций, базовое понятие языка Рефлекс — процесс. Программа на языке Рефлекс — это множество параллельно исполняемых процессов, которые могут запускать друг друга, останавливать и контролировать текущее состояние.

Для интеграции алгоритмов, создаваемых на языке Рефлекс, в среду LabVIEW был использован механизм Formula Node. Этот механизм не позволяет исключить использование во время лабораторного практикума пакета разработки LabVIEW, однако вполне пригоден для практической апробации базовой идеи. При предлагаемом подходе в блок-диаграмме выделяется базовая Formula Node для вставки алгоритма управления, создаваемого студентом во время работы со стендом (рис. 1).

Для преобразования текста на языке Рефлекс к Си-подобному синтаксису Formula Node использовался существующий транслятор языка Рефлекс в Си, который был дополнен автоматическим преобразованием прямо задаваемых констант и констант, задаваемых через перечислитель (enum), в переменные. Также было предложено монолитное представление алгоритма, исключающее использование функций, — механизма, отсутствующего в

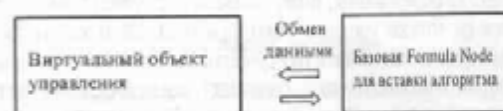


Рис. 1. Схема взаимодействия ВОУ и базовой Formula Node



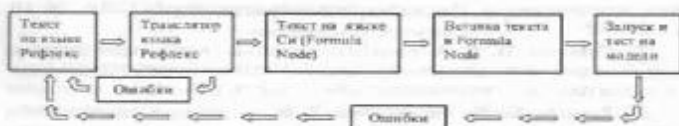


Рис. 2. Схема получения управляющего алгоритма и его отладка на модели

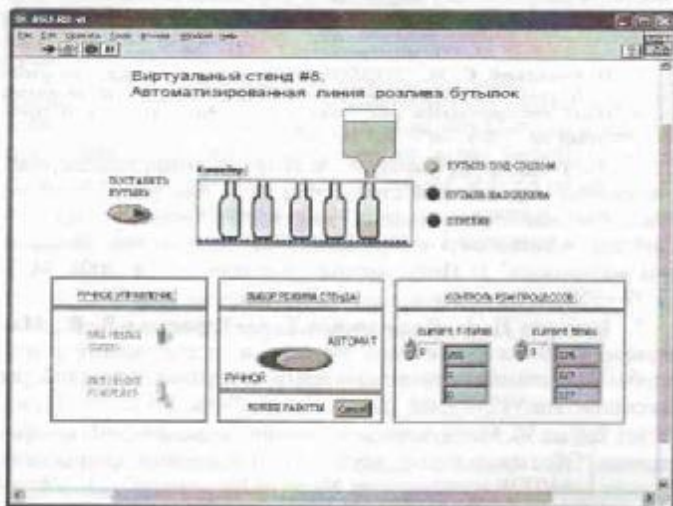


Рис. 3. Реализация виртуального стенда "Автоматизированный розлив бутылок": внешний интерфейс стенда

Formula Node. Тактирование гиперавтомата (программной модели, лежащей в основе языка Рефлекса [11]) проведено штатными средствами LabVIEW: основной цикл гиперавтомата "обернут" тактируемым циклом.

Подключение базовой Formula Node к проектируемому BOU реализовано через массивы ввода-вывода.

Для автоматической выборки входных-выходных переменных из массива использован стандартный синтаксис языка Рефлекса. Внутренние переменные гиперавтомата, обеспечивающие идентификацию текущих функций-состояний процессов  $fcur$  и текущих времен  $tcur$ , сохраняются в регистровых массивах, что гарантирует сохранение значений от цикла к циклу и независимость структурных связей от проектируемого алгоритма. Такой подход обеспечивает неизменность структурных связей между программными модулями для заданного виртуального стенда, поскольку позволяет полностью исключить влияние создаваемого алгоритма. Замечательно, что при предлагаемом подходе структура связей сохраняется и для различных лабораторных стендов, за исключением места присоединения модели виртуального объекта к массивам ввода-вывода, которые, очевидно, должны быть откорректированы в

соответствии со сценарием задачи и моделью объекта автоматизации.

Работа студента с виртуальным стендом в рамках предлагаемой концепции происходит следующим образом (рис. 2). Сначала создается текстовое описание алгоритма на языке Рефлекса, затем запускается трансляция в Си-формат. При обнаружении ошибок во время трансляции происходит возврат к редактированию. После успешной трансляции алгоритм управления, представленный уже в Си-формате, вставляется в Formula Node. Полученная программа запускается на исполнение, и алгоритм управления тестируется на корректность путем создания различных ситуаций, предусмотренных сценарием задачи. В случае обнаружения некорректного поведения алгоритма происходит возврат к начальному этапу редактирования текста алгоритма управления.

### Тестовый пример виртуального лабораторного стенда

Практическая применимость предлагаемой методики для создания виртуальных лабораторных стендов исследовалась на задаче управления автоматизированной линией розлива бутылок.

Автоматизированная линия розлива бутылок состоит из конвейера, по которому слева направо двигаются пустые бутылки, и бака с разливаемой жидкостью. Конвейер можно включать и отключать. В баке имеется сопло, которое можно открывать и закрывать. Также в системе присутствуют два дискретных датчика: для определения нахождения бутылки под соплом и контроля уровня жидкости в бутылке.

Имитационная модель линии розлива создавалась штатными средствами LabVIEW для работы с изображениями путем наложения графических примитивов конвейера и бутылки (рис. 3). Наполнение бутылки жидкостью имитировалось через постепенное закрасивание горизонтальных слоев пикселей от дна бутылки. Срабатывание датчиков наличия бутылки под соплом и уровня жидкости в бутылке определялось путем анализа цвета пикселей в за-

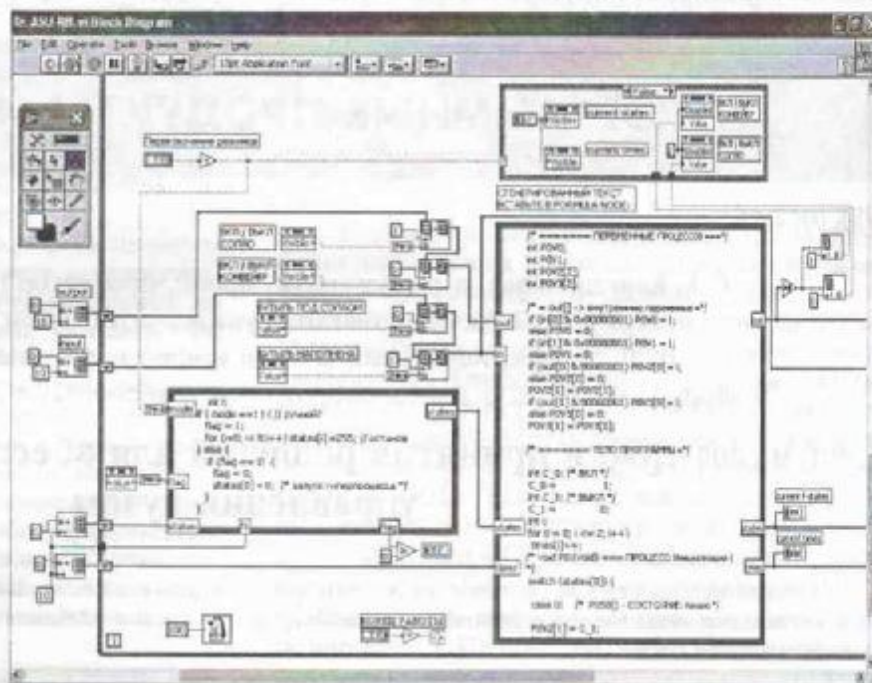


Рис. 4. Реализация виртуального стенда "Автоматизированный розлив бутылок": блок-диаграмма виртуального стенда (структурные связи Formula Node)

данных координатах сцены. В случае открытия сопла при отсутствии бутылки или при переполнении бутылки предусмотрена имитация протива жидкости на конвейер.

Стенд предусматривает два режима управления конвейером и соплом: ручной и автоматический. В ручном режиме имеется возможность непосредственного управления линией (конвейером и соплом). В автоматическом режиме возможность ручного управления блокируется, а команды управления соплом и конвейером формируются алгоритмом, созданным студентом по описанной выше схеме. В обоих режимах имеется возможность установки бутылки на конвейер. В специальном окне отображаются числовые идентификаторы текущих функций-состояний процессов.

Фрагмент блок-диаграммы, показывающий связи базовой Formula Node для встраивания алгоритма, приведен на рис. 4.

### Заключение

В работе предложена методика создания виртуальных лабораторных стендов по изучению основ программирования робототехнических комплексов. Методика основана на использовании пакета LabVIEW и языка описания управляющих алгоритмов Рефлекс.

Использование виртуальных лабораторных стендов позволяет студенту сконцентрироваться исключительно на вопросе создания алгоритма управления и упрощает изучение основ программирования ИУС. Разработанная технология интеграции гиперавтоматных блоков в среду LabVIEW может быть также использована при создании моделей объекта автоматизации для отладки реальных алгоритмов управления, а равно и для функционального расширения среды LabVIEW управляющими алгоритмами, заданными на языке Рефлекс.

### Список литературы

1. Зякин А. М., Букетин Б. В., Почев А. П., Шужов А. В., Щепетников О. А. Учебная Интернет-лаборатория "Испыта-

ние материалов" // Информационные технологии. 2006. № 10. С. 58–65.

2. Суворов М. В. Автоматизированная обучающая система диспетчеров промышленной электрической сети // Промышленные АСУ и контроллеры. 2003. № 7. С. 44–45.

3. Ершова О. В., Чистикова Т. Б. Структура тренажерно-обучающего комплекса для процесса производства желтого фосфора // Промышленные АСУ и контроллеры. 2003. № 7. С. 46–49.

4. Гершберг А. Ф., Подьяпольский С. В., Соркин Л. Р. Компьютерный тренажер для обучения операторов установки капитального ремонта в ООО "ПО "Криштифторсинтез" // Промышленные АСУ и контроллеры. 2003. № 7. С. 52–53.

5. Яценковский С. И. SDBUILDER: интеллектуальная гибридная система имитационного моделирования и управления сложными дискретными системами // Автоматизация в промышленности. 2006. № 7. С. 36–42.

6. Десятков А. Д., Сирота А. А. Имитационное моделирование систем с адаптивной структурой на основе технологий автоматизированного создания моделей в среде MatLab + Simulink + StateFlow Учебная Интернет-лаборатория "Испытание материалов" // Информационные технологии. 2008. № 3. С. 59–66.

7. Бутырин П. А., Васильковская Т. А., Каратаева В. В., Материкин С. В. Автоматизация физических исследований и эксперимента: компьютерные измерения и виртуальные приборы на основе LabVIEW 7. М.: ДМК Пресс, 2005. 264 с.

8. Труды VI Международной научно-практической конференции "Образовательные, научные и инженерные приложения в среде LabVIEW и технологии National Instruments", 23–24 ноября 2007. М.: Российский ун-т дружбы народов, 2007. [<http://digital.ni.com/worldwide/russia.nsl/web/01/46A43B32E2F8DD50C32573E400278DD7>].

9. Вавилов К. В., Шалыто А. А. LabVIEW и SWITCH-технология // Промышленные АСУ и контроллеры. 2006. № 6. С. 43–45.

10. Зюбин В. Е. "Си с процессами" язык программирования логических контроллеров // Мехатроника. 2006. № 12. С. 31–35.

11. Зюбин В. Е. Программирование информационно-управляющих систем на основе конечных автоматов: Учеб.-метод. пос. Новосибирск: Новосибир. гос. ун-т, 2006. 96 с. [<http://reflex-language.narod.ru/index.html>].

## ПРОГРАММНЫЕ ПРОДУКТЫ И СИСТЕМЫ

УДК 004.416.6

С. К. Каргапольцев, д-р техн. наук, проф., проректор по научной работе,  
Иркутский государственный университет путей сообщения,

Н. В. Лашук, начальник Центра информационных технологий,  
Забайкальский институт железнодорожного транспорта, e-mail: vc@zab.megalink.ru

### Система поддержки принятия решений для обеспечения автоматизации управления вузом

*Рассматривается применение системы поддержки принятия решений в автоматизированной системе управления вузом, что позволяет организовать корректное функционирование и взаимодействие всех подсистем автоматизированной системы управления вузом.*

*Ключевые слова: система поддержки принятия решений, автоматизированная система управления вузом, хранилище данных, оперативная аналитическая обработка данных, витрина данных.*