

Компьютерное моделирование в научных исследованиях и образовании

Многие пользователи, искренне желая применить компьютерное моделирование в своей практической деятельности, сталкиваются с серьезными трудностями при освоении и использовании современных программных средств. Для работы с ними все еще требуются знания, не относящиеся непосредственно к моделированию, а проведение вычислительного эксперимента остается кропотливой и многотрудной работой. В то же время типовых задач моделирования не так уж и много, и для них можно создать удобный и понятный интерфейс в рамках одного, «универсального» пакета.

Выделение типовых задач и создание для них удобного входного языка и инструментов анализа расширит круг пользователей и одновременно позволит сформулировать требования к структуре «универсального» пакета и функциональному назначению его блоков. Как показывает анализ, прототипы таких блоков существуют во всех пакетах, и они во многом дублируют друг друга, оставаясь при этом полностью несовместимыми и невазимоменяемыми. Опыт многих областей показывает, что надо разрабатывать и использовать стандартизованные, совместимые модули, и на их основе создавать уникальные продукты.

Преимущество стандартизованных блоков и согласованных интерфейсов очевидно при переходе к «массовому» производству компьютерных моделей.

Предлагаемая классификация типовых задач и попытка проанализировать, какие из существующих программных средств наиболее приспособлены для решения каждого конкретного типа, дают возможность читателю сравнить между собой различные современные технологии моделирования.

Создание прототипа универсального пакета из стандартных модулей, ориентированного на пользователя, не являющегося специалистом в области программирования и численного моделирования, приведет к тому, что компьютерное моделирование действительно станет инструментом научного работника, инженера и преподавателя.

первой группы условно назовем промышленными

Однако такие проекты невозможны без предварительных исследований, выполняемых отдельными учеными или проектировщиками. Стартовой точкой в них является гипотеза, а основной задачей — ее проверка. Исходным материалом служат плохо формализованные модели, то есть модели, чьи свойства еще не вполне осознаны. В самом начале исследований обычно ничего другого предложить невозможно, кроме как двигаться вперед на ощупь, практически без плана, формируя его по мере накопления материала. Главное — пробовать и видеть отклик. Это означает, что необходимо уметь организовывать и поддерживать непрерывную обратную связь между исследователем и исследуемой моделью. Аналогичная задача возникает и при обучении, когда необходима обратная связь между обучающей программой и учеником, или когда учитель прямо на занятии с помощью модели объясняет суть явления.

Промышленные пакеты слишком сложны и громоздки для проведения исследований на ранних стадиях и тем более обучения, для этого нужны специальные программные средства. Именно они, с нашей точки зрения, и образуют вторую группу пакетов. Назовем пакеты второй группы универсальными, подчеркивая этим, что они уступают по количеству уникальных возможностей промышленным, зато более просты для освоения и доступны отдельному исследователю при решении относительно несложных задач из практически любой прикладной области. Под несложными мы понимаем не простые задачи, а задачи, посильные одному разработчику, не являющемуся специалистом в области программирования и методов вычислений. В универсальных пакетах нужны разнообразные численные библиотеки, способные справиться с широким спектром проблем, а не методы, ориентированные на узкий класс задач. Для них нужны графические библиотеки, обеспечивающие показ изучаемого явления с разных сторон, а не одним, принятым в конкретной области, способом и, конечно же, поддержка интерактивного вмешательства в ход компьютерного эксперимента.

С момента появления пакета Simulink универсальные, не ориентированные на конкретные прикладные области пакеты для моделирования и исследования динамических систем в широком понимании этого термина, включая и дискретные, и непрерывные, и гибридные модели, стали повседневной реальностью. Относительная простота и интуитивная ясность входных языков универсальных пакетов в сочетании с разумными требовани-

Введение

Компьютерное моделирование широко используется как средство познания действительности, проектирования и обучения [1, 2, 4]. Программные средства для моделирования можно разделить на две группы.

К первой отнесем пакеты, предназначенные для решения сложных промышленных и научно-исследовательских задач большими производственными или научными коллективами. В таких проектах ведущую роль играет организация работ: хорошо налаженное взаимодействие между отдельными группами, быстрый доступ к многочисленным экспериментальным данным и библиотекам программ, тщательное документирование и тестирование, многовариантные расчеты. При этом обычно используются хорошо изученные модели, которые лишь модифицируются и приспособиваются для решения конкретных задач. В некотором смысле это относится и к большим научным проектам, когда успех во многом предопределен предварительными исследованиями, но для получения окончательных результатов требуется хорошо скоординированная совместная работа. Пакеты

ями к мощности компьютеров позволяют использовать эти пакеты в учебном процессе.

Изучаемые с помощью универсальных пакетов модели можно условно разделить на модели для естественнонаучных областей и модели технических объектов. Мы намеренно проводим эту грань, подчеркивая, что в первом случае мы обычно имеем дело с моделью, сведенной к одной, итоговой системе уравнений, или, другими словами, с однокомпонентной моделью, а во втором — со структурированной, многокомпонентной моделью, итоговая система для которой должна строиться автоматически по описанию отдельных компонент.

И среди однокомпонентных, и среди многокомпонентных, наибольший интерес представляют модели, чье поведение меняется во времени в зависимости от наступающих событий. Их часто называют гибридными системами. В отечественной литературе также используются синонимы — непрерывно-дискретные, системы с переменной структурой, реактивные, событийно-управляемые. Еще недавно единственным способом изучения гибридных систем было исследование их отдельных фаз или режимов и «склеивание» общего поведения вручную, подобно тому, как мы склеиваем панораму из отдельных фотографий. Теперь появилась возможность моделировать глобальное поведение таких объектов [4].

Под гибридными системами мы понимаем динамические системы с различным поведением в разных областях фазового пространства. Их фазовая траектория в зависимости от происходящих событий оказывается то в одной области, то в другой. Таким образом, к гибридным можно отнести классические динамические системы, чье фазовое пространство разбивается на ячейки с различным поведением, системы с разрывными правыми частями и системы, у которых меняется размерность в различных областях фазового пространства. Достижение фазовой траекторией границы областей будем называть событием, приводящим к смене поведения. Каждой области можно поставить в соответствие вершину некоторого графа, а его направленные дуги трактовать как возможные пути смены текущего локального поведения. Границы областей обычно задают с помощью предикатов, которые приписываются соответствующим дугам графа. Таким образом, гибридная система может быть представлена в виде графа, вершинам которого поставлены в соответствие классические динамические системы, и одна из вершин помечена как начальная, а дугам — условия смены поведения и мгновенные действия, выполняемые при смене поведения. Такая формализация называется гибридным автоматом. На наш взгляд, это наиболее наглядная и удобная форма описания поведения гибридных систем, совпадающая при описании дискретных систем с картой состояний, принятой в «унифицированном языке моделирования» UML [2]. Глобальное

поведение гибридной системы определяется всеми возможными путями, которые можно построить из начальной вершины. По мере движения модельного времени, фазовая траектория пересекает границы областей, при этом меняется вид решаемых уравнений. Можно также представить себе ситуацию, когда какая-либо из областей будет покидаться системой немедленно, как только система туда попадет. В этом случае гибридная система будет демонстрировать как длительные, «непрерывные» поведения, так и мгновенные, «дискретные».

Необходимость обеспечения обратной связи между исследователем и моделью опять же приводит нас к событийно-управляемым системам и дополнительно заставляет проводить и визуализировать вычислительный эксперимент в реальном времени. Назовем такой способ познания действительности *активным компьютерным экспериментом*, в отличие от традиционного пассивного вычислительного эксперимента, план которого может быть составлен заранее.

Отличительной чертой современных пакетов является объектно-ориентированный подход, позволяющий обеспечить еще одно очень важное и характерное для научных исследований и обучения требование — возможность легко пополнять и модифицировать разрабатываемую библиотеку, представляющую обычно последовательность все более сложных моделей, свойства которых приходится постоянно сравнивать.

Обратим внимание читателей на то, что практически все существующие современные и широко используемые пакеты не приспособлены в полной мере для проведения активного вычислительного эксперимента. Одновременно мы хотим обсудить требования к идеальному, гипотетическому универсальному пакету и рассказать о реальном международном проекте, в котором делается попытка разработать блочный универсальный пакет для моделирования динамических систем.

Типовые модели и компоненты универсальных пакетов

Модели, используемые на ранних стадиях научных исследований и проектирования, и практически все модели, используемые в образовании, названные во введении несложными, можно в свою очередь разделить на группы. Каждая из таких групп обычно является основной для определенной категории пользователей, у которых уже сложились свои требования к возможностям пакетов и свое видение способа общения с ними. Не следует нарушать требования пользователя и отвергать сложившиеся технологии моделирования без веских на то оснований. Наоборот, здесь надо идти пользователю навстречу и фиксировать в программных разработках отобранные практикой технологии.

Наша классификация моделей возникла потому, что для моделей различных типов существуют, на наш взгляд, различия в тех-

нологии моделирования. Мы выделяем изолированные и открытые системы, а также учитываем, является ли модель однокомпонентной или многокомпонентной.

Изначально вычислительные пакеты были ориентированы на эксперименты с изолированными моделями, изучать которые можно с помощью пассивного вычислительного эксперимента, то есть, по сути, речь идет об иной форме столь памятного еще многим пакетного режима. И именно эту черту унаследовали знаменитые MATLAB, Maple, Mathematica и другие математические пакеты. Существующие в них диалоги активно используются в основном при создании модели, но не при ее изучении. Сегодня, с появлением потребности в активном вычислительном эксперименте, ситуация принципиально изменилась. Диалог стал активно использоваться и при проведении эксперимента. Таким образом, основным типом изучаемых моделей становятся открытые системы.

Аналогичная ситуация наблюдается и со вторым «измерением» нашей классификации. В тех же прекрасно работающих с однокомпонентными моделями математических пакетах практически невозможно создать многокомпонентную модель. Пакет Simulink, позволяющий быстро создавать модели из типовых блоков, был обречен на успех, особенно среди инженеров. Однако реализованный в нем подход устаревает, так как восходит к эре аналоговых машин. В последнее время все больше внимания начинает уделяться многокомпонентным моделям различной физической природы и их автоматическому синтезу с использованием других подходов (например, проект Modelica [7]). И хотя в отдельных областях есть несомненные достижения, задачи здесь еще только ставятся.

Попробуем перечислить базовые модели, технологии их построения и исследования в универсальном пакете.

Изолированная однокомпонентная динамическая система

Такая модель задается системой обыкновенных дифференциальных и алгебраических уравнений с заданными коэффициентами и начальными условиями. Уравнения могут быть записаны явно, либо, как это часто бывает в теории управления, используется аппарат передаточных функций. Примером простейшей изолированной непрерывной системы может служить математический маятник, поведение которого изучают, например, при наличии или отсутствии трения.

Строгая трактовка изолированной системы не позволяет проводить с ней никаких активных экспериментов. Действительно, даже если параметры модели принадлежат некоторой области, и необходимо исследовать ее поведение при различных значениях параметров, то неявно предполагается, что проводится серия заранее спланированных экспериментов. В каждом конкретном эксперименте сначала фиксируются новые значе-

ния параметров, а затем находится решение аналитически или численно. Иными словами, записывается и реализуется цикл на языке пакета.

Изолированные непрерывные системы наиболее просто реализуются в математических пакетах. Пользователь пишет уравнения, цикл, позволяющий менять коэффициенты, внутри которого предусматривает вызов той или иной процедуры визуализации результатов, компилирует или интерпретирует модель и последовательно просматривает результаты в графических окнах. И этого оказывается достаточно для огромного числа пользователей математических пакетов. Если мы хотим, чтобы универсальный пакет был востребован и этой категорией пользователей, необходимо упростить до минимума интерфейс и свести его к привычному для пользователей математических пакетов.

В современных пакетах для моделирования, а они, несомненно, могут применяться для изучения изолированных систем, особенно когда нет возможности получить решение аналитически, перед пользователем иногда открывается столько ненужных ему окон, что он просто теряется и возвращается к испытанным программным средам. Упрощенный интерфейс универсального пакета для пользователей математических пакетов может сводиться к окну редактора уравнений, окну декларации используемых переменных и окну плана эксперимента.

На визуализации результатов необходимо остановиться особо. В математических пакетах традиционно используется математическая двумерная и трехмерная графика и «живые таблицы», позволяющие контролировать ход вычислений. Сопоставить модели ее динамический графический образ сложно, так как для этого требуются навыки программирования. В универсальных пакетах динамический графический образ становится основным инструментом визуализации, и для его формирования существуют специальные редакторы. Если динамический графический образ является элементом описания, то требуется также редактор (двумерной и трехмерной) анимации уже на стадии проектирования модели (такой подход принят, например, в пакетах Anylogic и Ejs). Последовательность заполнения этих окон может быть произвольной: математикам и проектировщикам привычнее начинать с уравнений, преподавателям — с динамического образа.

Изолированная однокомпонентная гибридная система

В изолированных однокомпонентных гибридных системах смена поведения зависит исключительно от внутренних событий. Например, в уравнении математического маятника коэффициенты могут меняться периодически или при достижении маятником заданного положения. Модели этого типа достаточно сложно реализуются в математических пакетах, в то же время дискретные системы

(частный случай гибридных систем) воспроизводятся достаточно просто во всех пакетах, опирающихся на карту состояния Харела (например, пакет Xjcharts).

Интерфейс пользователя для изолированных однокомпонентных гибридных систем требует еще одного окна — например, окна редактора карт состояния (Simulink, Stateflow) или их расширения — гибридных карт состояний (Model Vision, Anylogic). Существует много различных способов описать смену фаз или режимов — это и сети Петри, и графы связей, но нам кажется наиболее перспективным использование гибридных автоматов.

Для описания дискретных действий в гибридных моделях требуется расширенный набор типов переменных, а также некоторый набор алгоритмических операторов (как минимум, оператор присваивания, условный оператор и оператор цикла).

Открытая однокомпонентная гибридная система

Это наиболее востребованная практикой модель, которая позволяет проводить активный компьютерный эксперимент и может быть использована в качестве библиотечной компоненты. Такая модель имеет доступные извне переменные, посредством которых она может взаимодействовать с окружающим миром: с другими компонентами и с экспериментатором. Используя модель маятника в виде изолированной системы, можно построить открытую систему, объявив, например, длину маятника не коэффициентом, а входом, и связав его с движком в окне двумерной анимации. Переход от изолированной модели к открытой может быть нетривиальным: например, маятник, у которого длина стержня непрерывно меняется, описывается совсем другими уравнениями. Окна переменных («живые таблицы») и анимации (по крайней мере, двумерной) становятся интерактивными. Пользователь может с помощью ползунков, крутовых регуляторов, кнопок и тумблеров активно воздействовать на модель во время эксперимента. Очевидно, что активный эксперимент требует синхронной визуализации.

Авторы языка Modelica называют компоненту ориентированной, если все ее внешние переменные можно классифицировать либо как входы, значения которых не могут изменяться внутри компоненты, либо как выходы, значения которых, напротив, могут изменяться только внутри компоненты. Ориентированная компонента также называется блоком «вход–состояние–выход».

Блоки до сих пор трактовались как единственно возможные элементарные кирпичики, из которых строятся сложные модели. Блок может работать независимо и параллельно с другими блоками. Из блоков, соединенных связями, можно строить иерархические структурные схемы. Такая трактовка имеет место и в «классическом» пакете Simulink, и в новейшем языке Modelica. Различия касают-

ся только характера связей (ориентированные или неориентированные) и способов автоматического построения эквивалентной однокомпонентной непрерывной динамической системы.

При использовании гибридного автомата для описания динамических систем общего вида даже в элементарном блоке появляется новая «тонкая» структура. В самом деле, гибридная карта состояний, по сути, определяет гибридную систему как композицию (возможно иерархическую) более простых динамических систем (на самом низком уровне вложенности классических динамических систем), приписанных узлам графа. В каждый конкретный момент модельного времени функционирует локальная динамическая система, приписанная текущему состоянию, демонстрируя локальное поведение. Эта локальная динамическая система может быть приписана сразу нескольким состояниям.

Очевидно, что между блоком и локальным поведением нет никакой принципиальной разницы: и то, и другое — это динамические системы (классические или гибридные), и можно говорить о другом типе многокомпонентности — поведенческом, в котором элементарной компонентой становится динамическая система общего вида. Такие компоненты можно складывать (например, объединять уравнения в единую систему) и вычитать (вычеркивать уравнения), формируя из них нужное локальное поведение. Для гибридных автоматов динамическая система может играть роль элементарного строительного блока. В частном случае таким элементарным блоком может являться система уравнений, так же как и для математических пакетов. Таким образом, гибридную карту состояний можно рассматривать как последовательное соединение динамических систем, в то время как структурная схема является параллельным соединением.

Структурная многокомпонентная система с ориентированными блоками

Эта модель строится из ориентированных блоков. Модели этого класса наиболее приспособлены для изучения и проектирования технических объектов, собираемых из типовых блоков. Такой способ проектирования принят во многих пакетах — Simulink, Dymola, Model Vision Studium, Anylogic. Отличия технологий проектирования можно свести к двум. Первое связано с возможностью пользователя самостоятельно конструировать и добавлять новые элементарные (неструктурированные) блоки, используя входной язык пакета (Model Vision Studium, Anylogic, Dymola), либо обращаться к «низкоуровневым» процедурным языкам (Simulink). Второе связано с возможностью использовать механизмы наследования и полиморфизма при проектировании на уровне входного языка (Modelica, Model Vision). К новым элементам интерфейса следует отнести окно структуры. Автоматическое построение совокупной системы

уравнений в этом случае не вызывает никаких проблем (заметим, что для гибридных моделей такое построение должно выполняться динамически при переключениях). При визуализации многокомпонентных систем возникает интересная и еще не до конца решенная проблема об автоматическом построении анимационного образа всей многокомпонентной системы, составленной из анимационных образов отдельных компонент (AnyLogic). Примером простейшей такой системы может служить маятник в падающем лифте.

Структурная многокомпонентная система с неориентированными блоками

Успех проектирования электрических схем из компонент вселяет надежду на создание пакетов, в которых на основании компонентных и топологических уравнений можно будет строить сколь угодно сложные модели, объединяющие объекты различной физической природы. Основой этих моделей служит блок с особыми внешними переменными — контактами (Modelica), или неориентированный блок. Значения контактов могут меняться как внутри блока, так и извне. Одним из препятствий на этом пути построения многокомпонентных моделей служит то, что возникают трудности при автоматическом построении совокупной системы уравнений (особенно динамически). Но даже если такую систему удастся построить, практически невозможно заранее ответить на вопросы: имеет ли она решение, единственное ли оно, какими численными свойствами оно обладает. Во всех существующих пакетах ответы на эти вопросы «перекладываются на плечи» численных методов, а они для этого меньше всего приспособлены. В то же время для ряда областей — проектирование электрических цепей или электромеханических приборов — использование блоков с контактами является общепринятым приемом.

Многокомпонентная гибридная система переменной структуры

Модели этого наиболее сложного типа не реализованы в полном объеме ни в одном из известных пакетов. Речь идет о событийно-управляемых многокомпонентных иерархических моделях переменной структуры, построенных из блоков с переменными «вход», «выход», «состояние» и «контакт».

Объектно-ориентированное моделирование

Объектно-ориентированный подход в последнее время стал так прочно ассоциироваться с программированием (даже аббревиатуру ООП обычно расшифровывают как Объектно-Ориентированное Программирование), что многие забывают о его прямой связи с моделированием — ведь первоначально он был использован в языке моделирования SIMULA-67. Однако в дальнейшем объектно-ориентированный подход развивался почти

исключительно программистами.

Своеобразным итогом тридцатилетнего развития объектно-ориентированного программирования можно считать появление UML, предназначенного для создания объектно-ориентированных спецификаций программных систем на ранних этапах разработки. И только в последние годы объектно-ориентированный подход стал востребованным в своей «родной области» — моделировании. *Объектно-Ориентированное Моделирование* (ООМ) сегодня весьма модный термин, но на практике этот подход поддерживают не так уж много пакетов.

Объектом принято называть некоторую сущность, которая инкапсулирует в себе данные и методы как единое целое и взаимодействует с внешним окружением через определенный интерфейс. С понятием объекта тесно связано отношение двойственности — «класс-экземпляр». Каждый объект всегда является экземпляром какого-то класса.

Естественным кандидатом на роль объекта в ООМ является, конечно же, компонента. Компонента — совокупность переменных и поведения, она взаимодействует с внешним миром только через внешние переменные. Компонента всегда — явно или неявно — является экземпляром некоторого класса. Например, когда вы в пакете Simulink, который формально не поддерживает ООМ, размещаете на функциональной схеме новый блок, вы неявно порождаете новый экземпляр выбранного вами предопределенного класса из библиотеки. В современном ООМ компонента несет в себе информацию о поведении, структуре и динамическом образе моделируемого объекта. Это и есть три основных элемента определения класса.

Модель может содержать много экземпляров одного и того же класса. Без понятия класса практически невозможно моделировать многокомпонентные системы, имеющие регулярную или переменную структуру.

ВООП различают объекты пассивные и активные. *Пассивные* объекты (большая часть объектов в программах) только «откликаются» на вызовы методов и сообщений извне, но сами ничего не делают, т. е. не могут изменять значения своих данных по собственной инициативе. *Активные* объекты (например, экземпляры класса Thread в языке Java) имеют свою собственную «нить управления» и функционируют независимо от других объектов и параллельно с ними. Объекты ООМ, конечно же, активные, причем их активность связана не с циклической последовательностью дискретных действий, а с непрерывным воспроизведением поведения и непрерывной реакцией на внешние воздействия.

Более сложными механизмами ООМ являются наследование и полиморфизм. Наследование позволяет строить новые классы, модифицируя старые путем введения новых элементов и переопределения уже существующих. Это очень похоже на объектное программирование, но элементами компонен-

ты модели теперь являются переменные, системы уравнений и карты состояний. Например, чтобы создать модель прыгающего мячика в воздухе, можно взять модель прыгающего мячика в вакууме и переопределить поведение в состоянии «Полет», изменив уравнения с учетом сопротивления воздуха.

Полиморфизм позволяет использовать вместо компоненты одного класса компоненту другого похожего класса. Обычный полиморфизм по наследованию предполагает, что вместо компоненты базового класса всегда может использоваться компонента любого производного класса (например, для локационной станции и аэробуса, и истребитель выступают как летательный аппарат). Иногда оказывается полезным полиморфизм по интерфейсу, когда одна компонента может быть заменена другой, имеющей точно такой же набор внешних переменных (например, сопротивление, конденсатор и индуктивность — это все двухполюсники, с точки зрения собирания электрической схемы они одинаковы).

Существующие программные средства и их соответствие требованиям активного компьютерного эксперимента

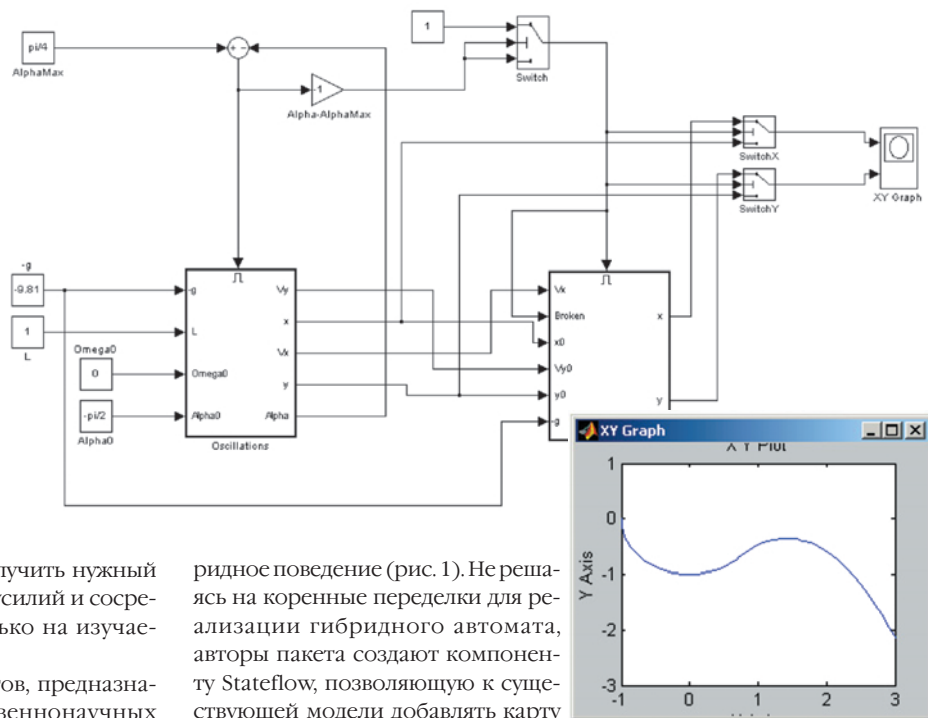
Традиционные математические пакеты, такие как Mathcad, Maple, Mathematica, наилучшим образом приспособлены для проведения хорошо спланированного пассивного вычислительного эксперимента для естественнонаучных дисциплин, когда производится серия длительных расчетов с заранее выбранной формой обработки и визуализации результатов для большого числа значений параметров моделей или различных модификаций моделей. В таких многовариантных расчетах накладные расходы, связанные с написанием специальной программы на языке пакета, управляющей экспериментом, с лихвой окупаются той легкостью, с которой возможно повторить все вычисления заново при внесении изменений в исходную модель. В учебном же процессе преимущества математических пакетов наиболее полно проявляются при решении простых, хорошо продуманных преподавателем задач, иллюстрирующих одно конкретное свойство изучаемого объекта. Программирование таких задач сводится к написанию относительно небольших по объему программ, состоящих в основном из макрооператоров, что позволяет получить нужный результат почти без видимых усилий и сосредоточить свое внимание только на изучаемом явлении.

Особое место среди пакетов, предназначенных для изучения естественнонаучных

дисциплин, занимают специализированные учебные пакеты, такие как Modellus, xyZET. Они ориентированы на школьников, школьных преподавателей и студентов университетов, где компьютерное моделирование изучается как один из возможных способов познания действительности. Эти пакеты просты для освоения, содержат достаточно выразительную графику и прекрасно приспособлены для изучения простых неструктурированных моделей, заданных системами дифференциальных уравнений невысокого порядка. Более подробную информацию о них можно найти в [5, 6, 8].

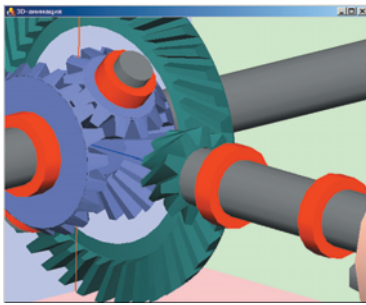
Среди существующих пакетов для моделирования многокомпонентных объектов наиболее распространенными и известными являются коммерческие пакеты, требующие больших усилий для освоения и предназначенные, в основном, для проектирования больших промышленных систем. Стоимость этих пакетов, особенно ориентированных на конкретную прикладную область, достаточно высока, что обычно заставляет пользователя, будь то отдельный ученый или университет, выбирать какой-либо один пакет на долгие годы.

К числу универсальных пакетов, которые используются и для обучения, можно отнести компоненты пакета MATLAB, а именно Simulink и Stateflow, и построенные по их образу и подобию пакеты VisSim, MBTU. Пакет Simulink пользуется заслуженной популярностью, а его язык блок-схем во многих публикациях стал средством описания изучаемых объектов. Но именно он и подвергается наибольшей критике со стороны авторов языка Modelica, выступающих за «физический» способ построения моделей из неориентированных блоков. В пакете Simulink практически невозможно наглядно реализовать гиб-



■ Рис. 1.

ридное поведение (рис. 1). Не решаясь на коренные переделки для реализации гибридного автомата, авторы пакета создают компоненту Stateflow, позволяющую к существующей модели добавлять карту



▲ Рис. 2.

состояния Харела, управляющую сменой поведения модели. Такой способ несколько ранее был реализован в пакете Model Vision 2.1, но от него решено было отказаться, так как модели становились чрезвычайно сложными для восприятия. Это приводило к многочисленным ошибкам при проектировании, и возникали сложности численного моделирования в окрестности точки смены поведения из-за возможных разрывов правой части дифференциальных уравнений.

Среди пакетов для моделирования специально выделим пакет Dymola, поддерживающий продолжающий развиваться универсальный язык моделирования Modelica. Несмотря на то, что Modelica поддерживает как ориентированные, так и неориентированные блоки, основным достоинством авторы этого пакета считают возможность соединять неориентированные блоки с помощью контактов. В этом плане, несомненно, достигнуто очень много. Однако и здесь смена поведения остается камнем преткновения. Если проследить за развитием языка в ежегодно публикуемых документах, то можно увидеть, как сокращаются возможности смены поведения блока. И это не случайно, так как возникают трудности математического характера при построении совокупной системы уравнений при соединении неориентированных блоков, меняющих свое поведение.

Пакеты Model Vision Studium [3], AnyLogic наиболее приспособлены для проведения активных компьютерных экспериментов. Они используют современные объектно-ориентированные входные языки. Пакеты Model Vision Studium, AnyLogic используют гибридные автоматы как элементы входного языка, однако не могут работать с неориентированными блоками. Пакет Model Vision Studium снабжен редактором трехмерной анимации (рис. 2), компактен и прост в освоении. Пакет AnyLogic позволяет строить апплеты, описывать поведение непосредственно на языке Java. Следовательно, он более приспособлен для созда-

ния моделей со сложным дискретным поведением.

Здесь будет уместно перечислить достоинства гибридного автомата при использовании его в пакетах моделирования.

Гибридный автомат:

- компактно и наглядно описывает все возможные варианты смены поведения подобно тому, как конечный автомат столь же наглядно описывает все допустимые цепочки принимаемого им языка;
- требует указания в явном виде особых событий, приводящих к смене поведения, и облегчает работу численных методов в окрестности точки смены поведения;
- позволяет ввести «алгебру» локальных поведений, упрощая тем самым применение объектно-ориентированного подхода;
- позволяет легко описать сложные эксперименты с моделью, облегчая проектирование испытательных стендов;
- позволяет в наглядной форме следить за появлением событий, приводящих к смене поведения, и самими переключениями при отладке (рис. 3).

Не останавливаясь на описании других пакетов, можно утверждать, что все они настолько различны (и в данном случае это недостаток), что нельзя выделить какой-либо один. Это чрезвычайно важно как для отдельного пользователя, так и для учебного процесса в целом.

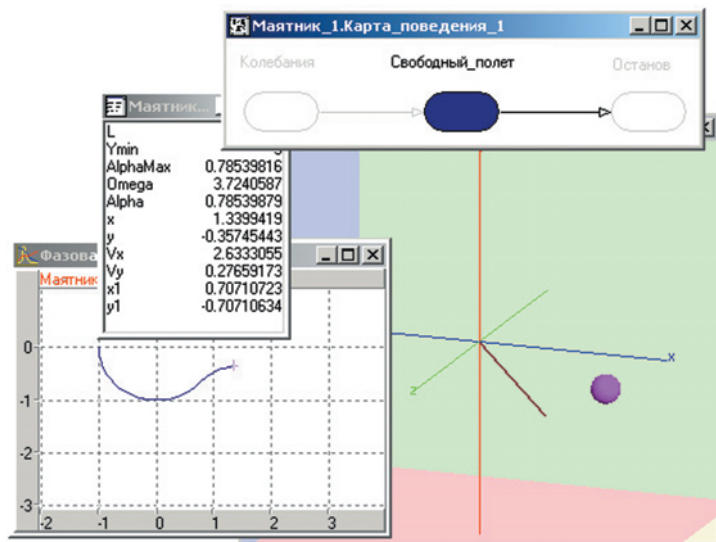
Все перечисленные выше пакеты обладают еще одним, чрезвычайно важным с нашей точки зрения недостатком. В силу того, что они являются коммерческими, они вбирают в себя в момент создания последние научные достижения, профильтрованные через индивидуальные предпочтения разработчиков, и надолго замораживают их. Действительно, создание конкурентоспособного нового продукта требует огромного стартового финансирования и усилий больших творческих коллективов специалистов различного профиля, и его коренная перестройка экономически невыгодна.

Модульный подход к разработке универсальных пакетов. Проект CoLoS

Количество универсальных существующих и разрабатываемых пакетов для моделирования все увеличивается. Это дает право утверждать, что мы имеем дело с востребованной областью вычислительного эксперимента, и позволяет сделать некоторые обобщения. Нам представляется, что в этой новой области мы имеем сейчас ситуацию аналогичную той, которая предшествовала созданию первых специализированных коллекций, таких как EISPACK и LINPACK.

Накопленного опыта достаточно для того, чтобы сформулировать единые требования к пакетам для проведения активного вычислительного эксперимента. По крайней мере, все яснее вырисовывается их структура и требования к отдельным компонентам.

■ Рис. 3.



Перечислим необходимые компоненты пакета. Это — *Редакторы уравнений, гибридного автомата, структуры, панелей управления, двумерной и трехмерной анимации*. Добавим сюда *Движитель гибридного времени*, а также *численную библиотеку* и получим список компонент универсального пакета. Разработать их все одновременно на высоком уровне трудно даже большому коллективу специалистов, что и находит отражение в сегодняшних пакетах. Это легко увидеть, сравнив, например, численные библиотеки математических и универсальных пакетов.

Несмотря на существование прекрасно организованных сайтов, где можно найти тщательно документированные и протестированные программы, отбор методов в пакетах производится порой случайным образом. При разработке нового пакета для создания требуемой библиотеки достаточно проделывать высококвалифицированную, но большей частью организационную работу. Взяв единую библиотеку за основу, ее можно дополнять при создании конкретного пакета. Наличие такой общей библиотеки только усилит доверие пользователя к новому программному средству.

Аналогичная ситуация наблюдается и с *Редакторами уравнений* или *структуры* — это относительно независимые компоненты, и они могут работать во многих пакетах одновременно. Более того, если зафиксировать интерфейс таких блоков и способы обмена данными, то можно просто заменять их целиком. Если бы удалось согласовать интерфейс этих блоков, а в качестве первоначальных вариантов взять уже готовые компоненты, к работе над созданием новых универсальных пакетов могли бы подключиться многие специалисты. Как минимальный результат, через несколько лет можно было бы создать прототип работающей системы.

Как бы ни фантастически звучали эти предложения, работа в этом направлении началась. В пакетах Model Vision и Anylogic используется одна и та же численная библиотека, написанная на языках Fortran и Java. Пакет OSP и пакет Ejs объединяет единая численная библиотека и *Редактор анимации* [9]. Авторы пакетов Model Vision, Ejs и проекта OSP объединяет то, что они члены междуна-

родной ассоциации CoLoS [5], и предложение объединить свои усилия родилось как следствие многолетнего обсуждения проблем, связанных с применением компьютеров в обучении.

Выводы

Обсуждаемый «идеальный» универсальный пакет моделирования должен:

- поддерживать все перечисленные типовые модели (таких пакетов в настоящее время нет);
- не просто поддерживать типовые модели, а поддерживать на естественном для данной типовой модели входном языке;
- обеспечивать проведение активного вычислительного эксперимента;
- создаваться на принципах ООМ;
- на наш взгляд, в современных условиях такой пакет может быть разработан только как открытый программный продукт из типовых модулей, прототипы которых уже существуют.

Литература

1. Колесов Ю. Б., Сениченков Ю. Б. Визуальное моделирование сложных динамических систем. — СПб: Мир и семья и Интерлайн, 2000. — 240 с.
2. Бенькович Е. С., Колесов Ю. Б., Сениченков Ю. Б. Практическое моделирование сложных динамических систем. — СПб: БХВ, 2001. — 441 с.
3. Колесов Ю. Б., Сениченков Ю. Б. Model Vision Studium — инструмент для объектно-ориентированного визуального моделирования сложных динамических систем // Труды Международной научно-технической конференции: Гибридные системы (7–9 июня, 2001). — СПб. — С. 5–46.
4. Колесов Ю. Б., Сениченков Ю. Б. Компьютерное моделирование динамических систем // Научно-технические ведомости СПбГТУ. — 2002. — № 3. — С. 93–102.
5. Сениченков Ю. Б. CoLoS — новые методы в обучении. Опыт международного сотрудничества // Компьютерные инструменты в образовании. — 1999. — № 1. — С. 11–15.
6. Иванов В. Б. Опыт использования пакета MODELUS при обучении физике и математике // Компьютерные инструменты в образовании. — 2000. — № 3–4. — С. 53–59.
7. Introduction to physical modeling with Modelica / Edited by M. Tiller. The Kluwer international series in engineering and computer science. — Vol. 615.
8. Haertel H., Ludke M. A computer supported course in mechanics. (Based on the simulation program xyZET and hypertext) // Computer Aided Learning and Instructions in Science and Engineering. Proceedings of CALISCE'96. Lecture Notes in Computer Science. — 1996. — P. 381–389.

**Рассылка новостей
«Научное ПО: MATLAB, Mathcad, Statistica»
от компании SoftLine каждую неделю!**

Подписка на сайтах www.exponenta.ru
www.matlab.ru
www.softline.ru
www.subscribe.ru
www.content.ru



Новая политика распространения пакета Model Vision Studium

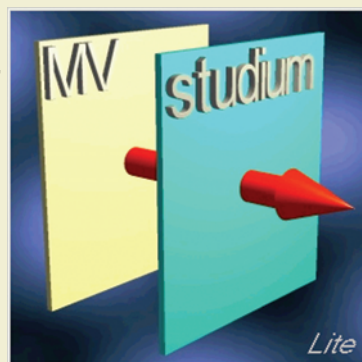
В настоящее время фирмой «Экспериментальные объектные технологии» (ХТ) разработана мощная профессиональная система моделирования AnyLogic, которая для непрерывных и гибридных моделей использует решения, апробированные в пакете Model Vision Studium (MVS). Так как пакет AnyLogic для многих пользователей является слишком сложным и дорогим, разработчики пакета визуального моделирования MVS решили продолжать совершенствовать свой пакет, позиционируя его как компактный, несложный и недорогой инструмент для научных исследований и обучения, и одновременно рассматривая его как «полигон» для отработки новых решений, которые затем будут использованы в AnyLogic и других проектах.

Разработчики пакета визуального моделирования MVS объявили также об изменении политики его распространения. Вместо ограниченной бесплатной версии MVS Lite предлагается полноценная, свободная для некоммерческого использования версия MVS Free. В ноябре 2002 г. на сайте Exponenta.ru была размещена русскоязычная версия MVS Free 3.1 с новым редактором математических выражений, подсистемой оптимизации и другими существенными усовершенствованиями. Эту версию можно копировать и передавать другим пользователям.

Одновременно с экспериментальной версией будет продолжено распространение недорогой версии MVS Standard для отдельных преподавателей и университетов.

Основным направлением развития пакета MVS остается объектно-ориентированное моделирование непрерывно-дискретных систем с использованием формализма гибридного автомата. В ближайшее время предполагается внедрить в пакет «свободную» форму записи уравнений (с использованием производных произвольного порядка, а также уравнений, не разрешенных относительно производных), неориентированные блоки и связи (подобные компонентам языка Modelica), а также динамические структуры и графический язык управления экспериментом на основе карт состояния.

Разработчики также предполагают разместить на сайте Exponenta.ru материалы о «внутреннем устройстве» пакета MVS и готовы к дискуссии по проблемам современного моделирования. Тем, кто хотел бы разработать для пакета какие-либо свои расширения, они готовы предоставить исходные тексты программ.



IV Международная конференция «Компьютерное моделирование 2003» и школа-семинар «Виртуальные лаборатории для естественнонаучных дисциплин»

Санкт-Петербургский государственный политехнический университет, журнал «Компьютерные инструменты в образовании» и международная ассоциация CoLoS проводят 24-28 июня 2003 года на кафедре Распределенных Вычислений и Компьютерных Сетей факультета технической кибернетики IV Международную конференцию пользователей «Компьютерное моделирование 2003» и школу-семинар «Виртуальные лаборатории для естественнонаучных дисциплин».

Предварительная программа конференции

24-25 июня. Компьютерное моделирование в университетах и школах. Школа-семинар для преподавателей естественнонаучных дисциплин.

26-28 июня. Современные проблемы компьютерного моделирования. Школа-семинар для разработчиков и пользователей современных пакетов для моделирования сложных динамических систем.

Основные направления работы конференции

- Секция 1. Компьютерные модели.
- Секция 2. Математическое моделирование.
- Секция 3. Программные комплексы для моделирования сложных динамических систем.
- Секция 4. Компьютерное моделирование и обучение.

Основные направления работы школы-семинара

- Секция 1. MVS и AnyLogic. Вводный курс.
- Секция 2. Отечественные пакеты для моделирования сложных динамических систем.
- Секция 3. Программные комплексы и виртуальные лаборатории ассоциации CoLoS.