

А.С. Дорофеев, С.С. Сосинская

Модели обучающего курса в разработке систем дистанционного обучения

В статье рассматриваются актуальные вопросы разработки информационных технологий в образовании и построения математических моделей образовательного процесса. Авторами рассмотрено описание процесса обучения и обучающего курса с использованием теории сетей Петри и цепей Маркова, предложен сквозной подход к проектированию и анализу обучающей системы с применением современных CASE-средств. Представлены результаты апробации разработанной системы на основе СУБД Caché и предложенных моделей.

Современное образование — это не только результат обучения, воспитания и развития личности, но и результат ее самообучения и саморазвития [18]. Применение в учебном процессе системы обучения позволяет систематически и оперативно оказывать обучаемому необходимую дидактическую помощь, выполнять непрерывный контроль за его действиями в процессе обучения, развивать познавательную активность и инициативу, а также снизить долю непроизводительного труда преподавателя [16]. Основными целями разработки и внедрения новых методов обучения является увеличение эффективности труда преподавателей и повышение качества подготовки специалистов.

Развитие информационных телекоммуникационных сетей дает новый импульс системам дистанционного обучения (СДО), обеспечивает доступ к информации, хранящейся в различных уголках нашей планеты.

Разработка мультимедийных обучающих курсов — довольно трудоемкая задача. Большинство имеющихся курсов представляют собой модели энциклопедии, которые методически немодифицируемы, поэтому их использование затруднено для аналогичных дисциплин с разным объемом изучения.

Проведенный анализ литературы показал, что в настоящее время уделяется недостаточно внимания вопросам общего

подхода к эффективному построению/сопровождению обучающего курса и организации процесса дистанционного (интерактивного) обучения с его использованием на основе общесистемных принципов, с применением современных объектно-ориентированных CASE-средств и объектных систем управления базами данных.

В связи с этим необходимо разрабатывать математические модели и развивать информационные технологии для выработки общего подхода к построению обучающего курса, который позволит, настраивая его на конкретные задачи, получать наиболее удобные и полезные для обучаемого формы представления знаний без значительных затрат на разработку каждого отдельного курса. Объектно-ориентированный подход, как наиболее современный, находит свое отражение и в разработке приложений, и в организации баз данных, необходимых для хранения информации наиболее эффективно и наименее избыточно.

На сегодняшний день российский рынок представлен несколькими пакетами для создания обучающих курсов. Среди отечественных систем наиболее известны, такие как «Орокс», «Прометей», «Гиперметод», «ДОЦЕНТ», «ГЕКАДЕМ» и др. Некоторые из них описаны в отчете «Системы ДО в России и за рубежом», опубликованном в Internet [20]. Сравнивая существующие сис-

темы, можно сделать вывод, что каждое средство разработки имеет свои достоинства и недостатки. Как правило, чем сложнее инструмент в использовании, тем он более гибок. Весь вопрос в том, какая степень гибкости окажется достаточной для достижения требуемого уровня сложности курса. В любом случае выбор среды разработки необходимо производить с учетом круга задач, сложности проекта и возможности приобретения готовых сред.

Впечатляющий прогресс в развитии аппаратных и инструментальных программных средств информационных технологий в образовании (ИТО) предоставляет технические возможности для реализации различных дидактических идей. Однако как показывает анализ литературы, отечественных и зарубежных компьютерных систем учебного назначения, ряд из них по своим дидактическим характеристикам нельзя назвать даже удовлетворительными [5, 11, 14], так как уровень качества продукта учебного назначения закладывается на этапе его проектирования при подготовке учебного материала для наполнения баз данных электронных учебников, при создании сценариев учебной работы с компьютерными системами моделирующего типа, при разработке задач и упражнений.

Создание средств ИТО для поддержки профессионального образования осложняется еще и необходимостью хорошо знать содержание предметной области и учитывать присущую ей специфику обучения. Именно отставание в разработке методологических проблем, «нетехнологичность» имеющихся методик являются одними из основных причин разрыва между потенциальными и реальными возможностями ИТО [23].

Моделирование обучающего курса и процесса обучения

В настоящее время нельзя назвать область человеческой деятельности, где в той или иной степени не использовались бы методы моделирования. Особенно это относится к сфере управления различными сис-

темами, где основными являются процессы принятия решений на основе получаемой информации. В последние десятилетия XX века моделирование стало проникать во все сферы деятельности, в том числе и в образовательную. Выстраивать любую систему без предварительно сформированной модели практически невозможно, поэтому важнейшим элементом образования становится моделирование.

Различные модели дистанционного обучения отличаются не только используемыми ими технологиями, но и степенью управления и ответственности преподавателей и учащихся. В некоторых моделях преподаватели и учебное заведение сохраняют свои функции полного управления процессом обучения, как в случае традиционной системы аудиторного обучения. В других — управление обучением переходит к обучаемым.

Согласно системному подходу при описании модели необходимо опираться на методологические принципы раскрытия объекта как системы [4]. Управление обучением является системным, так как предполагает наличие, по меньшей мере, двух подсистем: управляемой (объект управления) и управляющей (субъект управления). Будучи сами по себе системами, они и образуют самоуправляющую систему, в которой объект и субъект управления выступают уже в качестве подсистем. Сущность управления состоит в том, что управляющая система (субъект управления) систематически, целенаправленно воздействует на управляемую подсистему (объект управления), с тем, чтобы привести его к заданному результату, поставленной цели. При этом объект управления не является пассивным, он активно участвует в функционировании и развитии системы, в свою очередь воздействует на субъект и вместе с субъектом «соучаствует» в достижении результата. Это взаимодействие субъекта и объекта осуществляется посредством прямых и обратных информационных связей [3].

При системном подходе обучение рассматривается как двуединая деятельность

преподавателя и студента, имеющая системный характер, предметом анализа которой является взаимодействие между обучающим и обучаемым. Процесс обучения — это комплекс взаимосвязанных компонентов, объединенных общей целью функционирования и единством управления. Следовательно, вскрыть действительные условия, факторы эффективности обучения можно только на основе системного анализа взаимодействия преподавателя и студента.

Принципы обучения являются необходимыми средствами при проведении, организации, управлении, в построении и функционировании системы учебного процесса и ее компонентов. В совокупности принципы обучения определяют главные направления и содержание педагогических действий [19]. На основе этих принципов в реальном учебном процессе делаются выводы и суждения о дидактической правильности или ошибочности его состояния, построения и функционирования.

К основным принципам обучения можно отнести [2, 24] принципы научности, систематичности, связи теории с практикой, сознательности и самостоятельности обучения, единства конкретного и абстрактного, доступности, прочности знаний, наглядности, когнитивности коммуникации, присутствия компьютерному обучению [1].

Компьютерное обучение строится в соответствии с теми же целями и содержанием, что и обычное обучение. Целостные компьютеризированные курсы являются по своей сути кибернетическими системами. Их можно отнести к сложным человеко-машинным системам. Поэтому при их проектировании и использовании целесообразно придерживаться основных общесистемных принципов, конкретизированных применительно к процессу дистанционного обучения. К числу таких принципов, обеспечивающих реализацию уже названных, можно отнести целеустремленность, организованность, динамичность, целостность, эволюционность, открытость, приспособляемость, полифункциональность и технологичность [21]. Все

перечисленные принципы взаимосвязаны и представляют собой единую систему, в которой каждый из них необходим. Вместе они достаточны для того, чтобы их реализация способствовала конструированию педагогически полезной компьютерной поддержки по любой области знаний. Они связаны между собой логикой конструирования компьютерной поддержки, отвечающей целям, задачам и специфике протекания процесса обучения в дистанционном режиме. Системообразующим здесь является принцип целостности, которому подчиняются все остальные.

Результаты проведенных исследований и сравнительный анализ современных сетевых курсов позволяют сформулировать основные требования к составу программных компонентов сетевого курса. Обобщенный сетевой курс в его минимальной конфигурации должен включать ряд программных подсистем, каждая из которых, в общем случае, может содержать множество программных средств [20]: средства преподавателя, студентов, коммуникаций, а также подсистемы управления курсом и администрирования курса.

Модель описания структуры учебного курса и учебных материалов в значительной степени зависит от используемого алгоритма обучения.

Обычно алгоритм обучения с помощью такой системы состоит из следующих этапов [6]:

- 1) подготовка курса преподавателем — формирование содержания курса, разработка обучающих страниц и контрольных вопросов;
- 2) обучение — изучение обучающих страниц и ответы на контрольные вопросы обучаемым, сохранение результатов обучающей системой;
- 3) оценивание обучаемого преподавателем на основе результатов обучения.

Данный алгоритм обладает серьезными недостатками. Во-первых, преподавателю

трудно сразу четко сформулировать содержание нового курса. Во-вторых, он видит только информацию о том, как студент сдал тесты, но не видит сам процесс обучения, время и траекторию прохождения курса, т. е. не может дать рекомендаций по методике использования учебного средства. Кроме того, преподавателю трудно модифицировать разработанный курс, так как нет информации о восприятии курса обучаемыми. Алгоритм не учитывает возможность предъявления каждому обучаемому в зависимости от уровня его подготовки различных подмножеств обучающих страниц и тестов.

Возможно следующее решение указанных проблем.

Для четкого формулирования содержания нового курса необходимо ввести в обучающую систему понятие «цели обучения», в котором четко указано, что должен знать и уметь обучаемый после изучения курса. Иногда цели обучения называют также моделью требуемых знаний [22].

Для наблюдения за процессом обучения вводится трек обучения, в котором хранится информация о том, как обучается каждый обучаемый (например, время и порядок прохождения разделов и тестов). На основе информации, хранящейся в этом треке, ему можно давать рекомендации по эффективной методике использования обучающего средства.

Также следует хранить усвоенные термины курса в модели тезауруса обучаемого, чтобы дать возможность оценить преподавателю и обучаемому текущие знания обучаемого.

Для эффективной модификации курса необходимо видеть, как он воспринимается обучаемыми. Профиль курса строится на основе треков обучения и содержит некоторые усредненные характеристики данного процесса обучения. Отсюда вытекает необходимость разработки следующих моделей:

- обучающего курса, которая включает в себя модели описания учебных целей и содержания учебных материалов;

- трека обучения;
- текущих знаний обучаемого;
- тезауруса обучаемого;
- профиля курса.

1. Модель обучающего курса включает:

а) модель требуемых знаний или учебные цели, которая содержит информацию о том, чему необходимо научить обучаемых;

б) план курса, разрабатываемый на основе модели требуемых знаний и представляющий собой оглавление курса (может быть представлен в виде дерева/сети, вершинами которого являются обучающие страницы, блоки контрольных вопросов и внешние ссылки, а ребрами — связи между ними);

в) модель описания учебных материалов. На основе плана курса разрабатываются учебные материалы. Модель описания учебных материалов должна включать в себя такие понятия, как «обучающие страницы», «учебные элементы», «контрольные вопросы» и т. д.

2. Модель трека обучения. Трек обучения — это план курса, на который наложены результаты обучения одного студента, траектория обучаемого по сети Петри. Трек может иметь вид подграфа графа, представляющего весь курс, и должен включать в себя такую информацию, как количество посещений обучающих страниц, количество переходов по ссылкам, результаты ответов на контрольные вопросы, время, затраченное на изучение того или иного раздела и т. п. Наличие трека обучения позволяет преподавателю видеть не только результаты, но и процесс обучения и давать студентам рекомендации по методике обучения. На основе трека рассчитываются вероятности переходов обучаемого из одного состояния в другое.

3. Модель текущих знаний представляет собой модель учебных целей, на которую наложены результаты, хранящиеся в треке обучения. Модель предназначена для оценки знаний обучаемого на определенный мо-

мент времени. Ее возможно показывать не только преподавателю, но и самому обучаемому, чтобы он мог видеть, что он уже изучил и что еще осталось изучить. Модель текущих знаний может быть представлена как совокупность изученных разделов курса или совокупность достигнутых целей обучения.

4. В предлагаемой модели тезауруса обучаемого хранятся усвоенные им термины по изучаемому курсу. Данная информация может использоваться для дальнейшего обучения различным курсам в целях не избыточности представляемого материала (определения некоторых терминов будут изображаться в виде ссылок с названием термина, при необходимости щелкнув на которые можно будет увидеть его определение. Если термин обучаемому не знаком, даются более подробные сведения внутри информационного блока обучающей страницы).

5. Модель профиля курса содержит усредненные характеристики процесса обучения и показывает преподавателю, как воспринимается курс. Данная информация может использоваться для доработки курса. Основу модели составляют материалы работы [6].

Архитектура курса оказывает серьезное влияние на модель хранения данных и организацию удобного конструктора курсов [15]. Курс может иметь один или несколько сценариев (заранее описанную последовательность действий над объектами курса, которая может изменяться в зависимости от состояния данных в определенных точках сценария). Объектами курса могут быть как визуальные элементы, так и не визуальные, предназначенные для совершения операций над данными, анализа данных и управления траекторией выполнения сценария. Аппарат не визуальных элементов должен извлекать изменяющуюся последовательность изложения материала, например, в зависимости от степени усвоения курса обучаемым. Суть процедур корректировки заключается в оптимальном подборе воздействия на каждом

шаге обучения, отслеживании уровня знаний обучаемого и тенденции его состояний.

В качестве параметров адаптивного блока могут выступать, например, объем курса, объем заданий, время их выполнения, сложность заданий, темп выполнения, которые помогут настроиться на необходимый уровень подготовки. Этот блок может включать технические параметры сети, такие как ее загрузка, скорость передачи данных. Например, если загрузка сети значительна, то возможно отключение отображения некоторых сложных графических объектов, которые визуализируются по мере необходимости.

Несмотря на то что в настоящее время разработано большое количество различных подходов к моделированию, существует необходимость адаптации известных тематических моделей и методов к процессу обучения. При моделировании системы в качестве базовой информации используются данные о логической взаимосвязи событий в системе. Для моделирования процесса обучения целесообразно использовать теорию сетей Петри [7]. Сети Петри являются полезным инструментом для формулирования и решения проблем в области параллельных систем и процессов. Как аппарат моделирования, сети Петри могут отображать конфликтные ситуации, описывать параллельные процессы и логические взаимосвязи событий в системе [17].

Моделирующие возможности сетей Петри и их эффективность в приложениях объясняются прежде всего тем, что сеть Петри — это интеграция графа и дискретной динамической системы. Таким образом, она может служить и статической, и динамической моделью представляемого с ее помощью объекта [10, 12].

Функционирование сетей Петри состоит в изменении разметок, которое происходит в результате срабатывания переходов, моделирующих события в системе. Переход запускается удалением всех разрешающих фишек из его входных позиций и образованием новых фишек, помещаемых в его выходные позиции.

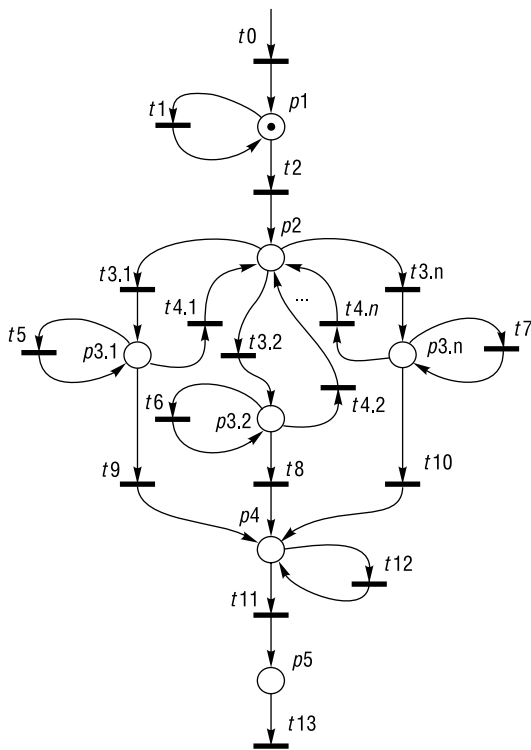


Рис. 1. Моделирование процесса обучения (фрагмент сети Петри)

На рис. 1 приведен фрагмент сети Петри для моделирования процесса обучения раздела курса, состоящего из двух тем. Обозначим условия и события.

Условия: $p1$ — проверка базовых знаний; $p2$ — изучение темы T_1 ; $p3.1$ — прохождение набора тестов 1 по теме T_1 ; $p3.2$ — прохождение набора тестов 2 по теме T_1 ; $p3.n$ — прохождение набора тестов n по теме T_1 ; $p4$ — изучение темы T_2 ; $p5$ — конец обучения.

События: $t0$ — начало обучения; $t1$ — повторная проверка базовых знаний; $t2$ — проверка базовых знаний завершена; $t3.1$ — выдача набора тестов 1 по теме T_1 ; $t3.2$ — выдача набора тестов 2 по теме T_1 ; $t3.n$ — выдача набора тестов n по теме T_1 ; $t4.1$ — тесты 1 не пройдены; $t4.2$ — тесты 2 не пройдены; $t4.n$ — тесты n не пройдены; $t5$ — повторное прохождение набора тестов 1 по теме T_1 ; $t6$ — повторное прохождение набора тестов 2 по теме T_1 ; $t7$ — повторное прохождение набора тестов n по теме T_1 ;

$t8$ — тесты 2 пройдены, изучение T_1 завершено; $t9$ — тесты 1 пройдены, изучение T_1 завершено; $t10$ — тесты n пройдены, изучение T_1 завершено; $t11$ — изучение темы T_2 завершено; $t12$ — повторное изучение темы T_2 ; $t13$ — окончание обучения.

Анализ построенных сетей на предмет достижимости был проведен с помощью программно-инструментального средства автоматизации разработки классических сетей Петри *Visual Petri*, позволяющего строить и выполнять построенные сети в автоматическом и пошаговом режимах.

При моделировании учебного курса целесообразно использовать сети Петри, в которых переходы срабатывают случайным образом. Приоритет одновременного срабатывания нескольких переходов можно задать вероятностями срабатывания каждого из них. Первоначально вероятности переходов могут быть оценены исходя из опыта работы с курсом и известного среднего уровня обучающихся (по уже известным результатам изучения курсов и анализу предварительного тестирования), а в дальнейшем — уточняя вероятность на основе собранной статистики прохождения курса обучаемыми. Если принять в качестве состояния сети разметку M_i и допустить, что вероятности не зависят от работы сети на предыдущих шагах, получится цепь Маркова, описывающая вероятностное поведение системы. Таким образом, для получения вероятностных оценок процесса изучения курса может использоваться аппарат теории цепей Маркова.

Для моделирования и анализа систем с объектной структурой в книге [13] изложен новый формализм — вложенные сети Петри, которые представляют собой расширение стандартного формализма этих сетей, где фишки в позициях сети сами могут быть сложными объектами с сетевой структурой. Название «вложенные сети» выбрано потому, что элементы сетей в них сами являются сетями. Таким образом, с помощью вложенных сетей Петри можно представить обобщенную структуру курса, состоящего из отдельных разделов, а сами разделы будут яв-

ляться элементарными сетями Петри. Оценив вероятности прохождения каждого из вложенных элементов, можно найти вероятность завершения изучения всего курса описанным в следующем пункте методом.

Процесс обучения можно рассматривать как динамическую систему, находящуюся в каждый из моментов времени в одном из n состояний. Это изменение определяется матрицей переходных вероятностей P . Если исключить из матрицы P строки и столбцы, которые соответствуют поглощающим состояниям, и обозначить полученную матрицу через Q , можно вычислить фундаментальную матрицу цепи Маркова:

$$N = (I - Q)^{-1}, \quad (1)$$

где I — единичная матрица.

Каждый элемент матрицы N есть среднее число пребываний процесса в состоянии S_j . В данной модели достаточно рассматривать только первую строку матрицы N , так как изучение определенного курса (раздела) начинается из состояния S_1 . При известных среднем времени, необходимом для прохождения j -го шага процесса обучения, трудоемкости Θ_j и первой строке матрицы N можно вычислить среднюю трудоемкость прохождения всего курса (раздела) Θ_Σ :

$$\Theta_\Sigma = \sum_{j=1}^l n_{1j} \cdot \Theta_j, \quad (2)$$

где l — размерность матрицы Q .

Для оценки дисперсии трудоемкости курса вычисляется матрица дисперсий числа пребываний процесса во множестве невозвратных состояний:

$$D = N(2N_{dg} - I) - N_{sq}, \quad (3)$$

где N_{dg} — матрица, полученная выделением из матрицы N диагональных элементов; N_{sq} — матрица, в которой каждый элемент матрицы N возведен в квадрат [9].

Для вычисления среднеквадратичного отклонения (СКО) числа пребываний про-

цесса во множестве невозвратных состояний от среднего применяется известная формула:

$$\sigma_{ij} = \sqrt{d_{ij}}, \quad (4)$$

где d_{ij} — элемент матрицы D .

Зная начальное распределение вероятностей, можно вычислить вероятности нахождения системы в соответствующих состояниях на любом шаге (в качестве шага можно использовать, например, час, занятие, день). В данной статье за шаг взят один академический час. Чтобы вычислить основные характеристики марковской цепи, было разработано соответствующее приложение для статистической обработки данных и их анализа.

Входной информацией для расчетов являются состояния, трудоемкости и таблица вероятностей переходов. Первоначальная трудоемкость и вероятности переходов оцениваются экспертно преподавателем-разработчиком курса. По мере накопления статистики вероятности переходов рассчитываются в соответствии со статистикой прохождения обучения. На каждом шаге (занятие, час и т. д.) для каждого обучаемого в СДО накапливается число переходов из одного состояния в другое и количество нахождений системы в каждом из состояний. Условная вероятность перехода из одного состояния в другое равна отношению:

$$P_{ij(t)} = P(S_{j(t)}/S_{i(t-1)}) = \frac{m_{ij}}{a_i}, \quad (5)$$

где m_{ij} — количество нахождений обучаемого в состоянии S_j при условии его нахождения в состоянии S_i на предыдущем шаге; a_i — общее число нахождений обучаемого в состоянии S_i .

Переходы нескольких обучаемых в определенные состояния являются независимыми. Зная вероятности переходов каждого k -го обучаемого, можно найти средние статистические значения вероятностей переходов:

$$\bar{P}_{ij} = \frac{\sum_{k=1}^{KOL} P_{ij}^k}{\sum_{j=1}^n \sum_{k=1}^{KOL} P_{ij}^k}, \quad (6)$$

где n — размерность матрицы переходов;
 KOL — общее количество обучаемых.

Формула (6) включает нормирование элементов вычисленной матрицы. Полученные средние статистические вероятности переходов сохраняются в профиле курса.

Сравнительный анализ современных сетевых курсов позволил сформулировать основные требования к составу компонентов сетевого курса. Разработанная система включает минимальный набор необходимых компонентов: регистрацию различных пользователей, инструмент для создания обучающих курсов и поддержку процесса обучения (представление теоретического материала, контрольные и лабораторные работы, тестирование и самоконтроль, накопление статистики). Кроме того, используется индивидуализация процесса обучения, которая состоит в различных уровнях представления материала: сильному обучаемому предоставляется материал для более глубокого изучения, слабому — минимальный набор понятий, знаний и навыков. Накопленные в процессе обучения статистические данные для конкретного курса позволяют получить информацию о процессе обучения, сложности и эффективности разработанного курса для возможной модификации его структуры с целью настройки на конкретный уровень знаний обучаемых.

В настоящей статье предлагается сквозной подход к проектированию и разработке обучающей системы с использованием различных моделей и CASE-средств структурного и объектно-ориентированного анализа на основных этапах ее жизненного цикла.

Сквозной подход к проектированию и анализу системы

Программный комплекс можно представить как

$$PC = (R, F, D, C),$$

где PC — программный комплекс;

R — формализация процесса постановки целей и задач, решаемых с помощью разрабатываемой системы, взаимодействие пользователей и системы;

F — функциональные компоненты, осуществляющие обработку данных;

D — база данных;

C — управляющий программный комплекс [8].

Схема интегрированного комплекса представлена на рис. 2.

Каждая составляющая данной тетрады представляет собой один из последовательных этапов проектирования системы. На начальном этапе описывается взаимодействие системы и пользователей, затем разрабатывается функциональная модель, строится схема базы данных и подбирается управляющий программный комплекс.

В этом комплексе функциональными компонентами F являются исполняемые модули прикладных подсистем в совокупности со служебными модулями, необходимыми для обеспечения совместной работы объединяемых подсистем.

Для проведения функционального анализа прикладных областей и разработки функциональных моделей используется международный стандарт IDEF0, который поддерживается CASE-средством *BPwin*.

Нотация стандарта IDEF1X для построения концептуальных моделей данных использована для представления схемы данных системы в *ERwin*.

Для формализации целей и задач, решаемых с помощью разрабатываемой системы, применяется унифицированный язык моделирования — UML (Unified Modeling Language), который используется в объектно-ориентированном CASE-средстве *Rational Rose*.

Одним из лучших формализмов моделирования управляющих комплексов являются сети Петри (*Petri Nets*), с помощью которых можно смоделировать многие системы, в особенности с независимыми компонентами, как, например, программное обеспече-

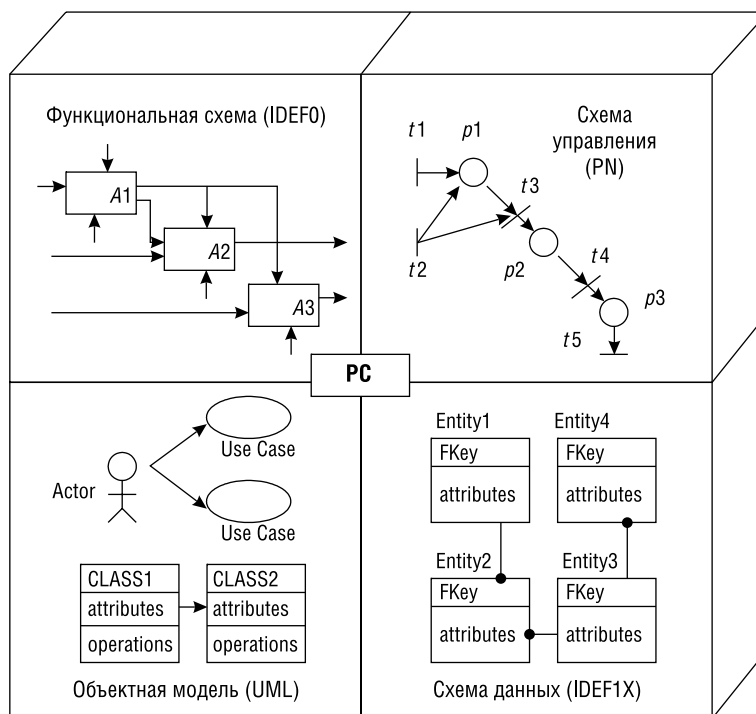


Рис. 2. Программный комплекс

ние. В связи с этим в качестве средства для моделирования управляющего программного комплекса (С) целесообразно использовать сети Петри, которые удачно представляют структуру управления программ [10]. Они предназначены для моделирования упорядочения событий и потока информации, а не для выполнения вычислений. Стандартный способ отображения структуры управления программ — это блок-схемы, которые представляют поток управления в программе. Основными компонентами блок-схем являются узлы двух типов: прямоугольники, представляющие вычислительные блоки, и ромбы, изображающие управляющие блоки, которые связываются между собой дугами. При переводе блок-схемы в сеть Петри узлы блок-схемы заменяются переходами, а дуги — позициями. Каждая дуга соответствует точно одной позиции сети Петри.

Успех реализации поставленной задачи во многом зависит от правильного выбора

модели данных и применяемой системы управления базой данных (СУБД) — ядра любой информационной системы [26].

В качестве такой системы была выбрана реляционная СУБД *Caché* (произносится «ка`шэ», означает «качество или свойство, которое приносит известность или престиж») компании *InterSystems*, основанная на новой технологии, обеспечивающей высокую производительность и масштабируемость прикладных систем, и обладающая возможностью отображать реальный мир, не искажая внутренних закономерностей предметной области.

Данная объектная СУБД идеально подходит для интерпретации сложных данных, в отличие от реляционной СУБД, добавление нового типа данных в которой достигается ценой потери производительности или за счет резкого увеличения сроков и стоимости разработки приложений. Объектная база данных не требует модификации ядра при добавлении нового типа данных.

Отличительной особенностью СУБД *Caché* является независимость хранения данных от способа их представления, что реализуется с помощью единой архитектуры данных *Caché*, в рамках которой существует единое описание объектов и таблиц, отображающихся непосредственно в многомерные структуры ядра базы данных. К одним и тем же данным, хранящимся под управлением многомерного сервера данных *Caché*, существует три способа доступа: прямой, объектный и реляционный. Имеющиеся серверы *Caché Objects* и *Caché SQL* предоставляют в распоряжение разработчика все наиболее популярные интерфейсы, посредством которых и унаследованные реляционные, и новые объектно-ориентированные прикладные системы получают равноправный доступ к данным, а данные хранятся в оптимизированной для обработки транзакций форме, что позволяет строить высокопроизводительные приложения.

Объектный доступ позволяет хранить и использовать в базе данных интеллектуальные объекты. С точки зрения объектной архитектуры, *Caché* относится к классу активных объектных баз данных, т. е. объекты не просто хранятся в базе, но и могут активировать свои методы непосредственно на сервере, что позволяет строить сложные интеллектуальные приложения.

Используемая технология *CSP (Caché Server Pages)* — серверные страницы (*Caché*) предлагает изящные средства создания быстродействующих, хорошо масштабируемых web-приложений за короткое время. Она также упрощает дальнейшее сопровождение и развитие таких приложений.

Caché Server Pages — объектная web-технология *Caché (CSP-технология)*, позволяющая использовать высокопродуктивные средства разработки *Caché* для быстрого создания сложных Intranet- и Internet-приложений, взаимодействующих с базой данных. При этом *Caché* служит не только СУБД, но и сервером приложений. В ней хранятся классы, соответствующие *CSP*-страницам, при

обращении к которым выполняются методы, генерирующие HTML или XML. С помощью последнего возможно представление структуры курса в формате, поддерживаемом стандартом *SCORM (Shareable Content Object Reference Model)* — промышленный стандарт для обмена учебными материалами), цели создания которого — обеспечение многократного использования учебных модулей, интероперабельности учебных курсов (их использования в средах разных обучающих систем), легкого сопровождения и адаптации курсов [25].

Caché предоставляет модуль интеграции со средой проектирования информационных систем *Rational Rose*. При этом UML-диаграмму классов можно экспортировать напрямую в базу данных *Caché*.

Апробация разработки

Разработанная программная оболочка дистанционного обучения разбита на ряд подсистем для работы определенных пользователей (администратор, сотрудники отдела кадров, учебного отдела, деканата, преподаватель, обучаемый), вызываемых через соответствующие страницы, на которых доступно меню пользователя.

Для апробации этой системы был спроектирован курс по дисциплине «Информатика и программирование». Основные разделы курса объединены в 15 модулей:

M1 — Основные понятия информатики и программирования. История развития языков программирования;

M2 — Алфавит языка Паскаль и типы данных. Стандартные процедуры и функции;

M3 — Структура программы. Отладка программ. Методы отладки;

M4 — Линейный и разветвляющийся вычислительные процессы;

M5 — Циклический вычислительный процесс;

M6 — Файлы;

M7 — Массивы;

M8 — Литеры и строки;

M9 — Множества;

M10 — Записи;

M11 — Нисходящее проектирование. Общая структура подпрограмм, параметры подпрограмм. Локальные и глобальные определения переменных. Отладка модульных подпрограмм;

M12 — Процедуры и функции;

M13 — Понятие рекурсии. Рекурсивные алгоритмы;

M14 — Динамическая память. Списки;

M15 — Модули: структура, определение переменных, компиляция, отладка.

Анализ построенной сети для данного курса показал, что сеть является безопасной, а все позиции достижимыми. Соответствующая цепь Маркова состоит из 75 различных состояний. Изначально экспертно были заданы трудоемкости достижения каждого из узлов цепи.

Состояние S_{75} является поглощающим и соответствует завершению процесса изучения курса. Вероятности переходов были заданы априорно на основе пятилетнего опыта преподавания курса в вузе.

По подсчетам средняя трудоемкость процесса изучения курса «Информатика и программирование» составит около 110 ч, СКО трудоемкости процесса от среднего

равно около 69 ч, завершение процесса изучения курса с вероятностью около 0,99 возможно на 104 шаге (рис. 3). При $K = 80$ вероятность этого составляет около 0,6; при $K = 90$ около 0,895.

По учебной программе курс «Информатика и программирование» содержит 64 ч лекций, 64 ч практических (лабораторных) занятий, 52 ч самостоятельной работы студентов. Следовательно, объем спроектированного курса не превышает число часов по программе.

Диаграмма вероятностей завершения изучения курса на определенном шаге представлена на рис. 4.

Разработанная система и спроектированный курс по дисциплине «Информатика и программирование» были апробированы при обучении студентов первого курса специальности «Информационные системы в экономике» очной формы обучения Института экономики Иркутского государственного технического университета. В качестве раздела курса для обучения выбран раздел «Массивы» (модуль M7, состояния S_{24} – S_{32}):

S_{24} — Изучение модуля M7;

S_{25} — Углубленное изучение модуля M7;

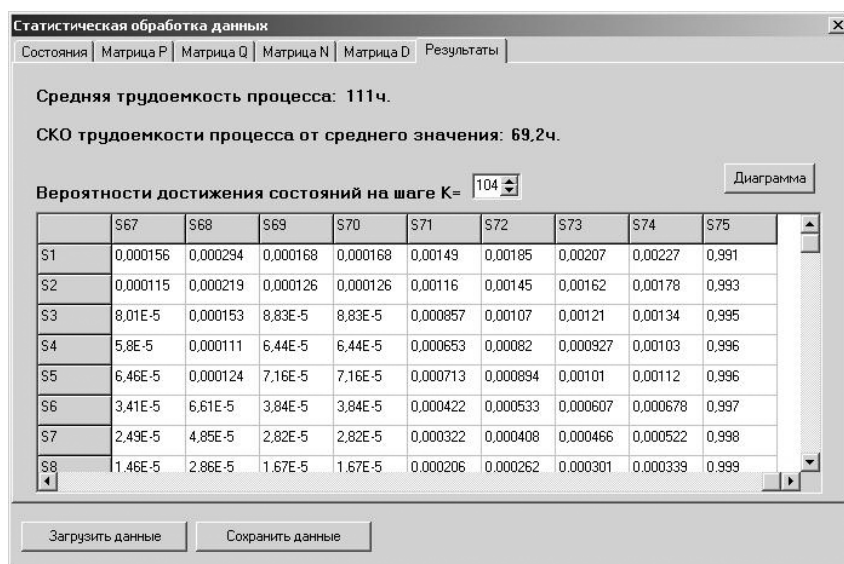


Рис. 3. Окно результатов

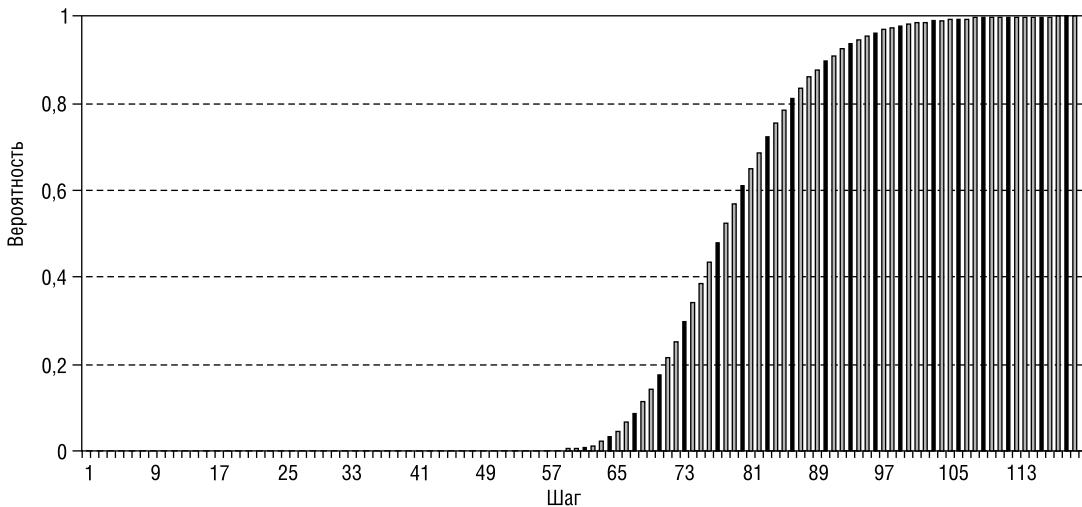


Рис. 4. Диаграмма вероятностей завершения изучения курса

S_{26} — Изучение модуля $M7$ с большим числом пояснений;

S_{27} — Тестирование по модулю $M7$ (теория);

S_{28} — Выполнение лабораторной работы;

S_{29} — Прохождение набора тестов 1;

S_{30} — Прохождение набора тестов 2;

S_{31} — Прохождение набора тестов 3;

S_{32} — Выполнение контрольной работы.

При этом состоянии S_{32} считалось поглощающим. Задержка в состоянии фиксируется, если обучаемый остается в нем более чем на время, указанное в трудоемкости для данного состояния.

Для каждого обучаемого в процессе обучения автоматически строились таблицы, содержащие статистические данные, накопленные в процессе освоения курса «Информатика и программирование» в треках обучения. Вероятности рассчитывались по формуле (5). Для большей точности проверка состояния обучаемого производилась с интервалом в одну минуту.

Средняя трудоемкость изучения данного раздела курса, включая тестирование по нему, составила 8,7 ч. Статистическая матрица вероятностей переходов вычислялась по формуле (6) и сохранялась в профиле для обучаемых по курсу «Информатика и программирование». Расчетная вероят-

ность завершения процесса изучения данного раздела достигает около 95% на 10-м шаге. Чем больше количество обучаемых, тем точнее и репрезентативнее полученные вероятности переходов. Сравнение расчетов с наблюдениями за реальным учебным процессом показывает качественно верное описание учебного курса и процесса обучения с помощью предложенных моделей.

Предлагаемый в статье сквозной подход к проектированию позволил снизить сложность создания системы дистанционного обучения, повысил удельный вес и качество ее анализа и проектирования. Применение объектной СУБД в качестве основы системы позволило эффективно хранить и использовать созданные объекты курса. Подход к моделированию обучающего курса и процесса обучения, а также их реализация расширяют возможности при разработке методологической основы применения информационных технологий в процессе обучения.

Использование СДО в процессе обучения сократит время на проверку знаний студентов и позволит сократить время изучения теоретического материала. Для студентов дневной формы обучения желательно совмещать обычные занятия с занятиями в обучающей системе с целью развития познавательной активности и инициативы

обучаемого, снижения доли непроизводительного труда преподавателя.

Список литературы

1. Аксенов М.В. Технология разработки экспертно-обучающих систем, ориентированных на обучение точным дисциплинам / Диссертация на соискание ученой степени кандидата технических наук. СПб.: Санкт-Петербургский государственный университет, 2004.
2. Архангельский С.И. Учебный процесс в высшей школе, его закономерные основы и методы. М.: Высшая школа, 1980.
3. Афанасьев В.Г. Системность и общество. М.: Политиздат, 1980.
4. Баляева С.А., Углова А.Н. Проектирование модели обучения на основе системно-деятельного подхода // *Системный анализ в проектировании и управлении: Труды VII Международной научно-практической конференции*. СПб.: Издательство СПбГПУ, 2003.
5. Борг А. Компьютеры в обучении: чему учит история // *Информатика и образование*. 1990. № 5.
6. Гапанюк Ю.Е. Модель описания обучающего пространства автоматизированного учебного курса; <http://iu5.bmstu.ru>.
7. Горбаченко И.М. Методы моделирования процесса обучения и разработка интерактивных обучающих курсов / Диссертация на соискание ученой степени кандидата технических наук. Красноярск: Сибирский государственный технологический университет, 2001.
8. Евгеньев Г.Б. Интеграция прикладных систем на основе баз знаний // *Программные продукты и системы. Приложение к Международному журналу «Проблемы теории и практики управления»*. 2005. № 3.
9. Кемени Дж., Снелл Дж. Конечные цепи Маркова. М.: Наука. Главная редакция физико-математической литературы, 1970.
10. Котов В.Е. Сети Петри. М.: Наука. Главная редакция физико-математической литературы, 1984.
11. Кривошеев А.О. Проблемы развития компьютерных обучающих программ // *Высшее образование в России*. 1994. № 3.
12. Лескин А.А., Мальцев П.А., Спиридонов А.М. Сети Петри в моделировании и управлении. Л.: Наука, 1989.
13. Ломазова И.А. Вложенные сети Петри: моделирование распределенных систем с объектной структурой. М.: Научный мир, 2004.
14. Машбиц Е.Н. Психолого-педагогические проблемы компьютеризации обучения. М.: Педагогика, 1988.
15. Наровлянский А.В., Рекуц В.С. Программное обеспечение конструирования и использование дистанционных учебно-исследовательских курсов: архитектура системы и структура хранения данных // *Информационные технологии в образовании: Сб. трудов XII Международной конференции. Часть V*. М.: Просвещение, 2003.
16. Пасхин Е.Н., Митин А.И. Автоматизированная система обучения Экстерн. М.: Издательство Московского университета, 1985.
17. Питерсон Дж. Теория сетей Петри и моделирование систем / Пер. с англ. М.: Мир, 1984.
18. Попов В.В. Дистанционное образование в свете креативной педагогики // *Дистанционное образование*. 1997. № 2.
19. Розенберг Н.М. Проблема измерений в дидактике. М.: Высшая школа, 1979.
20. Системы ДО в России и за рубежом; <http://www.eduhmao.ru>.
21. Скибицкий Э.Г. Дидактическое обеспечение процесса дистанционного обучения // *Дистанционное образование*. 2000. № 1.
22. Скибицкий Э.Г. Принципы построения компьютерной поддержки для дистанционного обучения // *Дистанционное образование*. 2004. № 3.
23. Соловов А.В. Проектирование компьютерных систем учебного назначения: Учебное пособие. Самара: СГАУ, 1995.
24. Соловьев В.А. Дидактический анализ проблематики электронного обучения // *IEEE International Conference on Advanced Learning Technologies*. Казань: КГТУ, 2002.
25. Ульянов Д.А. Марковская модель адаптивного тестирования и ее программная реализация в условиях дистанционного обучения / Диссертация на соискание ученой степени кандидата технических наук. Иркутск: Байкальский государственный университет экономики и права, 2006.
26. Федоров В.А. Постреляционная эра: реляционная или объектная? www.intersystems.ru.