

РОССИЙСКАЯ АКАДЕМИЯ НАУК
СИБИРСКОЕ ОТДЕЛЕНИЕ
ИНСТИТУТ СИСТЕМ ИНФОРМАТИКИ
имени А.П.Ершова

На правах рукописи

УДК 519.68, 681.3.06

Касьянова Елена Викторовна

АДАПТИВНЫЕ МЕТОДЫ И СРЕДСТВА ПОДДЕРЖКИ
ДИСТАНЦИОННОГО ОБУЧЕНИЯ
ПРОГРАММИРОВАНИЮ

Специальность 05.13.11 – математическое и программное обеспечение
вычислительных машин, комплексов и компьютерных сетей

Автореферат диссертации
на соискание ученой степени кандидата физико-математических наук

Новосибирск 2006

Работа выполнена в Институте систем информатики имени А.П. Ершова Сибирского отделения Российской академии наук (ИСИ СО РАН).

Научный руководитель: *В.А. Евстигнеев*
доктор физико-математических наук,
профессор

Официальные оппоненты: *В.Э. Малышкин*
доктор технических наук, профессор

Л.В. Городняя
кандидат физико-математических наук,
доцент

Ведущая организация: Сибирский государственный университет
телекоммуникаций и информатики
(СибГУТИ)

Защита состоится « 26 » декабря 2006 г. в 16 час. 00 мин. на заседании диссертационного совета К003.032.01 в Институте систем информатики имени А.П. Ершова Сибирского отделения РАН по адресу:

630090, Новосибирск, пр. Акад. Лаврентьева, 6.

С диссертацией можно ознакомиться в читальном зале библиотеки ИСИ СО РАН (г. Новосибирск, пр. Акад. Лаврентьева, 6).

Автореферат разослан « ____ » ноября 2006 г.

Ученый секретарь
Диссертационного совета К003.032.01

к.ф.-м.н.



Ф.А. Мурзин

ОБЩАЯ ХАРАКТЕРИСТИКА РАБОТЫ

Актуальность темы. В не столь далеком прошлом хороший почерк был гарантией спокойной и обеспеченной жизни до старости. Последние десятилетия характерны ускорением обновляемости технологий и знаний в различных сферах деятельности человека. Поэтому школьного и даже вузовского образования надолго уже не хватает. Сегодня особенно актуальна концепция непрерывного образования на протяжении всей жизни или, как говорят, пожизненного обучения (long-life education). Поиск соответствующей организационной структуры и учреждений образования (особенно образования взрослых), которые обеспечили бы переход от принципа "образование на всю жизнь" к принципу "образование через всю жизнь" — важнейшая проблема XXI века. Открытое образование — это образование, доступное всем. Его развитие неизбежно приведет к существенному пересмотру традиционных методик и технологий учебного процесса, а также к формированию единого открытого образовательного пространства на основе дистанционного обучения.

Системы дистанционного обучения в настоящее время активно исследуются и развиваются и уже успели пройти путь в пять поколений, начиная от систем обучения по переписке, больше известных в СССР как системы заочного обучения, и кончая системами гибкого обучения и интеллектуального гибкого обучения, определяющими настоящее и будущее дистанционного образования и базирующимися на Web-технологиях. Выгоды сетевого обучения ясны: аудиторная и платформенная независимости. Сетевое обучающее программное обеспечение, один раз установленное и обслуживаемое в одном месте, может использоваться в любое время и по всему миру тысячами учащихся, имеющих компьютеры, подключенные к Интернету. Тысячи программ сетевого обучения и других образовательных приложений стали доступны в сети за последние годы.

Проблема состоит в том, что большинство из них являются не более чем статичными гипертекстовыми страницами и не поддерживают проблемный подход к обучению.

Вместе с тем учебный процесс представляет собой совместную деятельность обучающего и обучаемого, который нельзя осуществить без решения задач, хотя в отличие от других видов деятельности (например, производственной или познавательной), где результаты решения задач являются ее прямыми продуктами, в учебной деятельности решение задач — это не цель, но средство достижения целей, а именно, учебных целей, направленных на изменение обучаемого, а не предметов внешнего мира. Особенно важен проблемный подход при начальном обучении программированию, в процессе которого обучаемый должен овладеть навыками точного формулирования алгоритмов на языке высокого уровня. Что невозможно сделать, прочитав несколько руководств или прослушав курс лекций по программированию. Необходима практика конструирования алгоритмов, и здесь невозможно обойтись без подходящего набора примеров и задач, а также без оценки разработанных алгоритмов на правильность и качество.

Появившиеся в последнее время адаптивные гипермедиа-системы существенно повышают возможности обучающих систем. Целью адаптивных систем является персонализация гипермедиа-системы, ее настройка на особенности индивидуальных пользователей. Поддержка адаптивных методов в гипермедиа-системах оказывается весьма полезной в тех случаях, когда имеется одна система, обслуживающая множество пользователей с различными целями, уровнем знаний и опытом, и когда лежащее в ее основе гиперпространство является относительно большим. Поэтому области применения адаптивной гипермедиа выходят далеко за границы обучающих систем и включают, например, такие, казалось бы далекие от обучения, области применения гипермедиа-систем, как открытые адаптивные виртуальные музеи.

Обучающие гипермедиа-системы, в которых пользователь или ученик имеет конкретную цель обучения (включая и такую цель, как общее образование), являются типичным приложением адаптивных гипермедиа-систем. В этих системах основное внимание уделяется знаниям обучающихся, которые могут сильно различаться. Состояние знаний изменяется во время работы с системой. Таким образом, корректное моделирование изменяющегося уровня знаний, надлежащее обновление модели и способность делать правильные заключения на базе обновленной оценки знаний являются важнейшей составляющей обучающей гипермедиа-системы.

Цель работы - разработка адаптивных методов и средств поддержки дистанционного обучения программированию в рамках проблемного подхода.

Методы исследования. В диссертационной работе использовались понятия и методы теории графов, теории вероятностей, теории сложности вычислений, адаптивной гипермедиа и искусственного интеллекта.

В разработанном курсе по программированию использовались методы теории формальных грамматик, автоматов и языков, математической логики и комбинаторики, а также методы разработки надежных и эффективных алгоритмов, их спецификации, верификации и анализа сложности.

Для описания синтаксиса языка программирования Zonnon и языка задания проектов ЯЗП использовались расширенные формы Бэкуса-Наура (РБНФ).

Научная новизна. В диссертационной работе получены новые научные результаты:

1. Разработаны модели знаний курса и обучаемого и методы мониторинга знаний обучаемого, поддерживающие адаптивность системы в рамках проблемного подхода к обучению и позволяющие оценивать знания обучаемого в условиях неполной информации.

2. Разработаны модель и методы тестирования знаний обучаемого, охватывающие известные подходы и позволяющие генерировать сценарий тестирования случайным образом. В основе модели разработанная классификация тестов по форме и глубине проверяемых знаний, а также их объединение в пространства тестов по содержанию проверяемого знания.

3. Разработан проект расширяемой среды, поддерживающей дистанционное обучение программированию в рамках проблемного подхода и соединяющей возможности адаптивных гипермедиа-систем и интеллектуальных обучающих систем. Среда ориентирована на поддержку дистанционного обучения, в процессе которого обучаемые, решая поставленные им индивидуальные задачи, действуют вполне самостоятельно, но постоянно обеспечены возможностью получения квалифицированной помощи, начиная с этапа понимания условий задач и кончая этапом оценки правильности их решения.

4. Разработан вводный курс программирования на базе языка Zonnon. Гипертекстовые учебные пособия «Введение в программирование» и «Практикум по программированию», поддерживающие курс, размещены на сайте русскоязычной библиотеки учебных курсов международной программы MSDN Academic Alliance .

Практическая ценность. Предложенные в диссертации подходы и методы могут быть использованы при создании новых средств поддержки дистанционного обучения.

Разработанный курс «Введение в программирование на базе языка Zonnon» может использоваться в тех учебных заведениях, где преподавание базируется на языке Паскаль и есть желание плавно перейти к новому вводному курсу программирования, охватывающему концепции современных языков программирования, таких как C#, Java и Ada.

Разработанные учебные пособия «Введение в программирование» и «Практикум по программированию», поддерживающие курс, размещены на

сайте русскоязычной библиотеки учебных курсов международной программы MSDN Academic Alliance и уже почти два года доступны для использования всем, кто имеет выход в Интернет.

Апробация работы. Основные положения диссертации обсуждались на Объединенном семинаре ИСИ СО РАН и НГУ «Конструирование и оптимизация программ» (Новосибирск, 2002-2006 гг.), а также на 2-м Всесибирском конгрессе женщин-математиков (г. Красноярск, 2002 г.), Международном конгрессе «Математика в XXI веке. Роль ММФ НГУ в науке, образовании и бизнесе» (Новосибирск, 2003 г.), 7-й Международной конференции ACM SIGCSE по инновациям и технологиям в компьютерном образовании (г. Орхус, 2003 г.), Пятой международной конференции памяти А.П. Ершова "Перспективы систем информатики" (г. Новосибирск, 2003 г.), 10-й Международной конференции по человеко-машинному взаимодействию (г. Ираклион, 2003 г.), Всероссийской научной конференции молодых ученых и студентов «Современное состояние и приоритеты развития фундаментальных наук в регионах» (г. Анапа, 2004 г.), Международной конференции "Вычислительные технологии и математическое моделирование в науке, технике и образовании" (г. Алма-Ата, 2004 г.), 17-м Всемирном конгрессе IMACS по научным вычислениям, моделированию и прикладной математике (г. Париж, 2005 г.), Всероссийской конференции "Исследования в области образования, молодежной политики и социальной политики в сфере образования" (г. Москва, 2005 г.), Конференции-конкурсе «Технологии Microsoft в теории и практике программирования» (г. Новосибирск, 2006 г.), Международной научно-практической конференции «Современные ценности и эффективность моделей образовательных систем» (г. Новосибирск, 2006 г.), Шестой международной конференции памяти А.П. Ершова "Перспективы систем информатики" (г. Новосибирск, 2006 г.), Международной конференции «Вычислительные и информационные технологии в науке и образовании» (г. Павлодар, 2006 г.).

Публикации. Основные результаты диссертационной работы опубликованы в 21 работе, из которых 12 статей, 2 учебных пособия, 1 препринт и 6 тезисов докладов.

Исследования выполнялись в соответствии с планами научно-исследовательских работ ИСИ СО РАН по проекту 3.15 «Методы и средства трансляции и конструирования программ» программы 3.1 СО РАН «Информационное и математическое моделирование в различных областях знаний, задачи поддержки принятия решений, экспертные системы, системное и теоретическое программирование» и поддерживались грантами Минобразования (грант УР.04.01.023 «Графы в программировании: обработка, визуализация и применение» научной программы «Университеты России» и грант Е02-1.0-42 «Граф-модели в программировании и визуальная обработка» по фундаментальным исследованиям в области естественных и точных наук) и Микрософт Рисечь (грант «Разработка вводного курса программирования»).

Структура диссертации. Диссертация состоит из введения, четырех глав, заключения, списка литературы и приложения. Работа содержит 122 страницы текста (за исключением приложения и библиографии) и 15 рисунков. Список литературы содержит 154 наименования.

СОДЕРЖАНИЕ РАБОТЫ

Во введении анализируется проблема разработки адаптивных методов и средств поддержки дистанционного обучения программированию, обосновывается актуальность работы и ее цель, приводится список основных результатов полученных в ходе исследований.

Первая глава посвящена вопросам дистанционного и проблемного обучения. Глава начинается с рассмотрения проблем российского образовательного рынка и целей государственной программы по созданию систем открытого образования на основе технологий дистанционного

обучения (разд 1.2). Описываются основные этапы в развитии дистанционного обучения, связанные с использованием разных моделей (модель обучения по переписке, мультимедийная модель обучения, модель телеобучения, модель гибкого обучения и модель интеллектуального гибкого обучения) и технологий (разд. 1.3). Исследуются особенности учебной деятельности и проблемного подхода к обучению (разд.1.3). Завершают изложение выводы по главе.

Вторая глава посвящена адаптивным методам и средствам дистанционного обучения.

В последнее время адаптивные гипермедиа-системы становятся все более и более популярными в дистанционном обучении, предоставляя управляемые пользователем средства доступа к информации. Активно ведутся исследования, направленные на создание продвинутых сетевых образовательных приложений, которые смогли бы обладать адаптивностью и интеллектуальностью.

Данная глава посвящена вопросам обеспечения персонализации при дистанционном обучении, особое внимание в ней уделяется анализу методов и средств адаптивной гипермедиа, используемых современными адаптивными обучающими Web-системами.

В разд. 2.1 определяются адаптивные гипермедиа-системы и дается ответ на вопрос, к чему они адаптируются, а также рассматриваются характеристики пользователя обучающей системы, важные для ее адаптации. Вопросу, что адаптируется в системах Web-обучения, посвящен разд. 2.2. В разд. 2.3 рассматриваются такие технологии адаптации, как выстраивание программы обучения, интеллектуальный анализ решений обучаемого, поддержка в решении задач, привнесенные в обучающие Web-системы из несетевых интеллектуальных обучающих систем. Разд. 2.4 и 2.5 содержат описание методов и технологий адаптации содержания и навигации, характерных для адаптивных гипермедиа-систем. Моделям предметной области, пользователя и адаптации, используемым в адаптивных гипермедиа-

системах, посвящен разд. 2.6. Разд. 2.7 содержит примеры существующих экспериментальных адаптивных обучающих Web-систем. Завершают изложение выводы по главе 2.

Третья глава посвящена проекту WARE, работа над которым ведется в Институте систем информатики СО РАН. Цель проекта — разработка адаптивной среды дистанционного обучения WARE, поддерживающей активное индивидуальное обучение программированию в рамках проблемного подхода и соединяющей возможности адаптивных гипермедиа-систем и интеллектуальных обучающих систем.

Глава начинается с общего описания системы WARE и ее возможностей с точки зрения обучаемых (студентов). Возможностям, которые система предоставляет другим пользователям (администраторам, лекторам и инструкторам), посвящен разд. 3.2. В разд. 3.3 рассматриваются модель знания курса, а в разд. 3.4 и 3.5 — вопросы моделирования знаний студента и использования при этом моделирования механизма сетей Байеса. Моделям тестирования и видам тестов посвящен разд. 3.6. В разд. 3.7 и 3.8 рассматриваются проекты (упражнения и задания) подсистемы CLASS, играющей роль виртуальной семинарской аудитории в рамках системы WARE. Разд. 3.9 содержит описание подсистемы PRACTICE, предназначенной для использования при прохождении студентами компьютерного практикума по курсу, поддерживаемому системой WARE. Виды аннотирования проектов и другие действия, которые осуществляет система в помощь студентам, рассматриваются в разд. 3.10. Разд. 3.11 посвящен режимам работы студентов, а вопросы обновления модели знаний студента и развития курса описаны в разд. 3.12 и 3.13. В разд. 3.14 описываются работы по экспериментальной реализации системы WARE. Завершают изложение выводы по главе 3.

Пользователи системы WARE. Система WARE ориентирована на поддержку дистанционного обучения и предполагает четыре типа пользователей: студенты, инструкторы, лекторы и администраторы. Все

пользователи осуществляют доступ к системе через стандартный Web-браузер, который представляет HTML-документы, предоставляемые HTML-сервером на стороне сервера. Администратор может выполнять ряд административных функций по организации учебного процесса, связанных с управлением курсами, преподавателями (лекторами и инструкторами) и студентами. Интерфейс лектора поддерживает следующие основные функции: редактирование учебной информации курса, общение со студентами и инструкторами, просмотр статистики, управление мониторингом, а интерфейс инструктора — функции проверки и оценки заданий студентов, общения со студентами и преподавателями, просмотра статистики, управления режимом работы и мониторингом.

После авторизации пользователя в качестве студента открывается подходящее меню команд. Система WARE поддерживает три уровня процесса обучения:

(1) когда студент изучает теоретический материал в некоторой специфической области с использованием гипертекстовых учебников и задачников,

(2) когда система тестирует концептуальные знания студента, соответствующие изученному теоретическому материалу,

(3) когда студент под управлением системы выполняет учебные проекты, решая задания и упражнения.

Третий уровень рассматривается как основной в использовании системы; для того, чтобы изучить курс, поддерживаемый системой, студент должен справиться с набором *проектов* (заданий и упражнений), который инструктор подбирает студенту строго индивидуально. Другой тип задач, поддерживаемый системой WARE, — это так называемые *тесты*. В отличие от проектов, решение о выполнении (или невыполнении) которых принимается инструктором, тесты — это вопросы, правильность ответов студентов на которые система оценивает полностью автоматически.

Ориентация на цели обучения является одним из важных свойств нашей WARE среды. Поскольку мы не хотим фиксировать путь обучения студента или студенческой группы от начала до конца, студенты свободны в определении своих собственных целей обучения и своих собственных последовательностей обучения. На каждом шаге они могут обращаться за помощью к системе, запрашивая подходящий материал, последовательности обучения и советы по примерам и проектам. Если студенту необходим совет по нахождению своего собственного пути обучения, он может спросить систему о следующей подходящей цели обучения. Открытые дискуссии, поддерживаемые системой WARE, обеспечивают полную виртуальную атмосферу телекласса, включая возможности кооперативного изучения курса вместе с другими студентами и средства кооперативного преподавания для инструкторов и лекторов.

Система WARE предназначена для обслуживания многих студентов с различными целями, знанием и опытом. В нашей системе основной упор делается на знание студентов, уровень которого может весьма сильно варьироваться у разных студентов. Более того, состояние знаний студента изменяется в процессе работы с системой. Поэтому большое внимание уделяется возможностям адаптивности в нашем проекте.

Модель знаний курса. Центральным объектом модели обучения в WARE является некоторый учебный курс (или просто курс); он представляет реальный курс, читаемый в некотором университете, реальном или виртуальном. Каждый курс имеет своего лектора, который создает и поддерживает глоссарий, проекты, тесты, учебники и задачки по курсу (в качестве учебного материала), а также курирует проблемное обучение групп студентов, объединенных в студенческий курс (или поток), осуществляемое под руководством группы инструкторов (ассистентов лектора). Вся информация по завершении обучения потока сохраняется в системе и может использоваться лектором для совершенствования курса по следующим направлениям: включение новых учебников в курс, создание или улучшение

примеров решения проектов, пополнение заданий новыми тестами и эталонными решениями.

В основе курса лежит его *модель знаний*, которая представляет собой конечное непустое множество *элементов знаний* S с двумя бинарными отношениями U и W на S , удовлетворяющими следующим свойствам для любых $p, q \in S$:

(1) $(p, q) \in U$ тогда и только тогда, когда p является составной единицей знания, которая является объемлющей для q ;

(2) $(p, q) \in W$ тогда и только тогда, когда единица p должна быть изучена до изучения q .

Предполагается, что в модели знаний (S, U, W) пара (S, U) образует лес, а (S, W) является ациклическим графом.

На основе модели знаний курса строится *гlossарий* курса, в котором каждая единица знаний представлена одним (или несколькими) ключевым словом (или фразой) и дополнена множеством ссылок на те элементарные информационные ресурсы курса (элементарные информационные ресурсы учебников и задачников, а также примеры, тесты и проекты), содержание которых относится к данной единице знаний.

Каждый элементарный информационный ресурс курса представлен отдельной Web-страницей и индексируется некоторым набором единиц знаний, описывающих содержание этого ресурса. Происхождение информационного ресурса несущественно для индексирования, только содержание определяет его информационный индекс. Пусть $S \neq \emptyset$ — множество всех единиц знаний, $P(S)$ — множество всех подмножеств множества S , а H — множество всех информационных ресурсов, тогда *карта содержания* — это функция

$$I: H \rightarrow P(S) \setminus \emptyset,$$

которая каждому $h \in H$ сопоставляет *информационный индекс* этого ресурса $I(h)$ — непустое множество всех тех единиц знаний, которые связаны с данным информационным ресурсом: если h является тестом или проектом, то

$I(h)$ состоит из тех единиц знания, которые используются в h , а если h является информационным элементом гиперкниги, то $I(h)$ состоит из тех единиц знания, объяснение которых в той или иной степени содержится в данном информационном ресурсе h .

Моделирование знаний студента. Знания студента x моделируются *вектором знаний*

$$K(x) = (p_1, \dots, p_n),$$

где $n = |S|$ — число единиц знаний в модели знаний курса, а для любого i , $1 \leq i \leq n$, $p_i = p(s_i / E_i)$ — условная вероятность, описывающая предположение системы о том, что студент x обладает знанием единицы знания $s_i \in S = \{s_1, s_2, \dots, s_n\}$ на базе тех свидетельств E_i , которые система собрала о знании студентом x единицы знания s_i в процессе мониторинга его работы над курсом.

Каждый элемент p_i вектора знаний $K(x)$ выражает степень знания единицы знания s_i , которым обладает студент x в данный момент. Мы используем четыре степени такого знания: знания эксперта — обозначаются E (отличные знания), знания продвинутого пользователя — обозначаются F (несколько затруднительных моментов в понимании s_i , но в целом понятие вполне усвоено), знания начинающего — обозначаются A (много затруднительных моментов, понятие s_i плохо усвоено), знания новичка — обозначаются N (пользователь еще не готов для работы с данным понятием, оно им не усвоено). Таким образом, единицы знания являются с одной стороны понятиями, описывающими предметную область курса, а с другой — случайными переменными со значениями E , F , A и N , кодирующими степень знаний данного студента.

Свидетельства, получаемые системой в процессе мониторинга действий студента, изменяются со временем. В типичном случае знание студента возрастает во время работы с курсом, хотя недостаток знания также воспринимается системой как свидетельство. Поскольку каждый отслеживаемый результат производимого наблюдения за студентом сразу же

заносится в свидетельства, вектор знаний в любой момент дает моментальный снимок текущего уровня знаний студента. Обновление свидетельств о студенте происходит только при выполнении им тестов и проектов. При этом никакое тестирование не позволяет студенту получить от системы оценки “отличные знания”, и он может достичь уровня знаний эксперта только за счет успешного выполнения проектов.

Сеть Байеса – это ациклический оргграф, вершины которого представляют случайные переменные, а дуги отражают зависимости между ними; каждая вершина сети помечена таблицей условной вероятности, которая определяет воздействие на вершину ее непосредственных предшественников. Сети Байеса являются мощным инструментом вывода в графах с зависимыми случайными вершинами.

Мы используем сеть Байеса для вычисления распределения вероятностей для каждой единицы знаний и, следовательно, для вычисления векторов знаний пользователей. Чтобы построить сеть Байеса, которая вычисляет распределение вероятностей для каждой единицы знаний курса для конкретного студента, требуется выполнить два действия. Во-первых, сгенерировать некоторый ациклический оргграф, имеющий единицы знаний в качестве вершин и обучающие зависимости между ними в качестве дуг, и, во-вторых, определить таблицы вероятности для всех вершин. Для наших целей мы строим сеть Байеса со случайными переменными, которые дают распределение вероятностей для вычисления знаний пользователя, и проводим дугу между X и Y в том и только том случае, когда $(Y, X) \in W$ и не существует такого Z , что $\{(Y, Z) (Z, X)\} \subseteq W$.

Известно, что точный вывод в сетях Байеса является *NP*-трудной проблемой. Вместе с тем существуют линейные по времени алгоритмы для тех сетей Байеса, базовые графы которых не содержат циклов, а также существует несколько методов обработки таких не полностью

ориентированных циклов: кластеризация, кондиционирование и стохастическая симуляция.

Модель тестирования. По виду различается три типа тестов: выборные, мультिवыборные и наборные. *Выборный* тест предполагает выбор одного варианта ответа из предлагаемого списка, *мультिवыборный* — нескольких вариантов, а *наборный* тест требует, чтобы студент ввел правильный ответ в специальное текстовое поле. Мы используем компьютерное тестирование со случайным позиционированием ответов для выборных и мультिवыборных тестов, чтобы ответы по номерам позиций могли привести к ошибкам студентов, заучивающих позиционные номера правильных ответов.

По глубине проверяемых тестами умений, относящихся к конкретной области изучаемого курса, и временным ограничениям, ассоциированных с ними, тесты разделяются на три группы: вербальные, качественные и аналитические. *Вербальный* тест определяет некоторую конкретную концепцию определения и имеет временное ограничение в 60 секунд. В случае *качественных* тестов, в которых должны быть объяснены более сложные понятия, обычно ответ ожидается между 120 и 240 секундами. Для *аналитического* теста, при котором должно быть объяснено некоторое трудное понятие, ответ обычно ожидается внутри временного диапазона в 360 — 480 секунд.

С каждой единицей знания курса связываются два пространства тестов, первое из которых используется для тестирования студентов, претендующих на знание данной единицы знаний на уровне “начинающего”, а второе — на уровне “продвинутого” студента.

Пространство тестов — это ациклический оргграф $G = (X, V)$, вершинами которого являются тесты, а дуги отражают последовательность их прохождения: $(p, q) \in V$ тогда и только тогда, когда после выполнения студентом теста p ему может быть предложен тест q . В пространстве тестов выделяются множества входных (не имеющих предшественников) и выходных (не имеющих преемников) тестов.

С каждым пространством тестов G лектор связывает две константы C_1 и C_2 , где C_1 — минимальный уровень знаний для начинающего студента, а C_2 — минимальный уровень знаний для продолжающего студента. По умолчанию предполагается, что пространство тестов для начинающих студентов состоит из вербальных и качественных тестов и имеет $C_2 > 1$, а пространство для продвинутых студентов состоит из аналитических тестов, и в нем $C_2 = 1$.

В процессе тестирования студенту предлагается серия тестов T , генерируемая случайным образом и образующая путь в G от некоторого входного теста до некоторого выходного.

Пусть студент x выполнил серию тестов T , и γ_x есть отношение числа правильных ответов, данных студентом x на серии тестов T , к максимальному числу возможных правильных ответов для серии тестов T . Тогда студент x продемонстрировал на серии тестов T знания новичка, если $\gamma_x < C_1$, начинающего студента, если $C_1 \leq \gamma_x < C_2$, и продвинутого студента, если $C_2 \leq \gamma_x$.

Подсистемы и проекты. Система WAPE состоит из двух основных подсистем: CLASS и PRACTICE.

Подсистема CLASS является виртуальной аудиторией, для проведения семинарских занятий по курсам. Любой курс, поддерживаемый системой CLASS, включает набор проектов, являющихся упражнениями или заданиями. Каждый проект P — это множество однотипных задач (вариантов проекта), одна из которых является примером и содержит формулировку задачи, ее решение и обоснование решения. По уровню сложности проекты разделяются на проекты стандартной сложности, которые проверяют понимание студентом изложенного в учебнике теоретического материала, и проекты повышенной сложности, которые добавляют новую или требующую размышлений информацию к материалу, изложенному в учебнике. Каждый проект стандартной сложности предполагает, что у студента, его решающего, уровень знания для всех тех единиц, которые образуют индекс данного

проекта, был бы не ниже уровня продвинутого студента, а любой проект повышенной сложности предполагает, что студент является экспертом во всех единицах знаний, входящих в индекс данного проекта.

Каждое *упражнение* P — это множество вопросов, предполагаемые ответы на которые не рассматриваются системой как исполняемые программы, а любое *задание* P — это множество задач на программирование.

С каждым заданием α связывается множество эталонных решений $S(\alpha)$, множество тестов задания $T(\alpha)$, множество правил вывода знаний студента $R(\alpha)$, вектор предварительных знаний $P(\alpha)$ и вектор итоговых знаний $F(\alpha)$. $S(\alpha)$ — это тексты программ решения задания α , не доступные студентам, к которым добавлены комментарии, поясняющие эти решения и содержащие их мотивированную оценку. $T(\alpha)$ распадается на множество $T_1(\alpha)$, используемое системой для проверки правильности понимания студентом условия задания, и $T_2(\alpha)$, используемое для автоматической проверки правильности составленной студентом программы.

Каждому тесту $t \in T_2(\alpha)$ сопоставлено диагностическое сообщение, выдаваемое студенту тогда, когда его программа не соответствует тесту t , и общедоступный комментарий, содержащий предполагаемые ограничения на рабочие характеристики программы для данного теста.

Каждое *правило вывода* знаний студента из множества $R(\alpha)$ имеет вид: $B \rightarrow N$, где B — логическое выражение, содержащее литералы вида x или $\neg x$, $x \in T_2(\alpha)$, а N — вектор знаний.

Правило $B \rightarrow N$ работает в два этапа. На первом этапе в B вместо каждого вхождения теста $x \in T_2(\alpha)$ подставляется либо значение *true*, если на входных данных теста x программа выдает предполагаемые результаты, либо значение *false* в противном случае. На втором этапе вычисляется выражения B , и, если оно принимает истинное значение, то происходит перевычисление вектора знаний $K(x)$ студента x , решающего данное задание, по правилу $K(x) := \min(N, K(x))$.

Вектора $P(\alpha)$ и $F(\alpha)$, приписанные заданию α , имеют следующий смысл. Предполагается, что студент x может приступить к выполнению проекта α только в том случае, если $K(x) \geq P(\alpha)$. Успешное выполнение проекта α приводит к перевычислению $K(x)$ по правилу $K(x) := \max(N, K(x))$.

Система PRACTICE является виртуальной лабораторией, предназначенной для прохождения студентами компьютерного практикума по курсу. Основная цель, которая ставится перед студентом при выполнении индивидуальных заданий (интегральных проектов), составляющих компьютерный практикум, — это практическое освоение всех этапов разработки надежной и наглядной интерактивной (диалоговой) программы для компьютерного решения некоторой нетривиальной задачи, требующей разработки алгоритма, обработки сложных структур данных и создания дружественного интерфейса.

С каждым интегральным заданием α связывается множество эталонных решений $S(\alpha)$ и два вектора знаний: $P(\alpha)$ — вектор предварительных знаний и $F(\alpha)$ — вектор итоговых знаний. Они имеют тот же смысл, что и у заданий системы CLASS, с тем отличием, что *эталонные решения* $S(\alpha)$ — это прокомментированные решения задания α , которые имеют вид гипертекстовых отчетов.

Работая в виртуальной лаборатории, студент изучает методический материал, содержащийся в задачнике курса, тестирует свои знания теоретического материала, прочитанного им, или решает интегральные проекты, которые для удобства разбиты на три группы: обычной сложности, повышенной и пониженной сложности.

Режимы работы студента. Для вычисления релевантности доступного по ссылке проекта текущему уровню знаний студента используется простая метафора «семафора». Ссылки на проекты, которые должен решить студент, помечены одним из трех цветовых кружков: готов-для-решения (зеленый кружок рядом с текстом ссылки), не-готов-для-решения (красный кружок)

или уже-решен (серый кружок), что позволяет студенту выбрать подходящие проекты.

Студенту часто требуется информация по определенным темам, но не хватает предварительных знаний для понимания этой информации. Для поддержки студента система сравнивает его текущий уровень знаний с необходимым для понимания рассматриваемой темы. Если у студента не хватает некоторых предварительных знаний, система может сгенерировать последовательность информационных элементов (след), который направляет его обучение в рамках данной темы.

Студент может также запросить у системы следующий разумный шаг обучения. Этот запрос на прямое руководство выполняется путем определения подходящей цели обучения, зависящей от текущего состояния знаний студента и состояния его проектов.

Инструктор может управлять тем, как знания студента влияют на возможность его работы с системой. В *жестко контролируемом* режиме студент x может выполнять только те проекты, для которых у него достаточно знаний. В *слабо контролируемом* режиме студент x может выполнять не только проекты α , для которых у него хватает знаний (т.е. $K(x) \geq P(\alpha)$), но и проекты α , где для любого i либо $K(x)(i) \geq P(\alpha)(i)$, либо $K(x)(i) = F$ и $P(\alpha)(i) = E$. В *свободном режиме* студент может выполнять любой проект. При этом вне зависимости от режима работы студент x получает всякий раз предупреждение о некомпетентности в единице знаний i , когда пытается начать выполнять проект α , для которого $K(x)(i) < P(\alpha)(i)$.

Многие адаптивные системы фиксируют факт чтения пользователем определенной информации и на этой основе обновляют оценку его знаний. Некоторые из них учитывают время чтения или последовательность прочитанных страниц для углубления этой оценки. Хотя это оправданный подход, его недостатком является трудность измерения знания, приобретенного пользователем во время чтения Web-страницы. Вместо этого мы используем для обновления модели знаний только тесты и проекты. Это

мотивировано проблемным подходом к обучению, поддерживаемым системой WARE.

Предполагается, что после завершения чтения того или иного теоретического материала система организует обратную связь. Студент последовательно указывает те темы, которые были целью изучения данного материала, и для каждой из них выбирает одну из двух категорий: «тема была несложной — я овладел материалом без труда», «тема была непростой — у меня были некоторые проблемы в понимании». Эти категории соответствуют знаниям продвинутого студента и начинающего. После этого система генерирует индивидуальный набор тестов, успешное прохождение которых позволяет студенту изменить желаемым образом оценку своих знаний.

Обновление модели знаний автоматически происходит во время работы студента над некоторым проектом каждый раз, когда решение студента не проходит некоторый тест или когда инструктор завершает оценку решения проекта студентом. Студент имеет право в любой момент понизить свою оценку для любой единицы знаний, а также попытаться повысить её. В последнем случае в зависимости от его претензий он получает индивидуальный набор тестов, сгенерированных системой, успешное прохождение которых позволяет студенту изменить желаемым образом оценку своих знаний.

Четвертая глава посвящена вводному курсу программирования на базе языка Zonnon.

При обучении программированию наиболее важным представляется начальный этап, на котором обучаемый должен овладеть навыками точного формулирования алгоритмов на языках высокого уровня. Это невозможно сделать, прочитав несколько руководств или прослушав курс лекций по программированию. Необходима практика конструирования алгоритмов, и здесь невозможно обойтись без языка программирования, пригодного для целей начального обучения, и подходящего набора примеров и задач.

В главе кратко представлен вводный курс программирования на базе языка Zonnon, описаны две книги: «Введение в программирование» и «Практикум по программированию», поддерживающие этот курс.

Zonnon — это новый универсальный язык программирования в семействе языков Паскаль, Модула-2 и Оберон, работа над которым ведется в Цюриховском институте информатики. Он сохраняет стремление к простоте, ясному синтаксису и независимости концепций, а также уделяет внимание параллельности и легкости композиции и выражения. Унификация абстракций является стержнем проектирования языка Zonnon, и она отражается в его концептуальной модели, основанной на модулях, объектах, определениях и реализациях. Язык Zonnon содержит такие новые черты, как активность в объектах, основанный на межобъектном взаимодействии диалог, перегрузка операций и обработка исключительных ситуаций. Язык Zonnon специально разрабатывается как платформно-независимый язык. Первая реализация языка Zonnon выполнена для платформы .NET. Кроме того, предполагается интеграция компилятора в систему программирования Visual Studio .NET (в сотрудничестве с компанией Microsoft).

Глава начинается с изложения основных особенностей языка Zonnon (разд. 4.1). Разд. 4.2 содержит описание цели и основных принципов курса. Разд. 4.3 является кратким описанием книги «Введение в программирование», являющейся учебником по курсу. Цели и содержание компьютерного практикума составляют разд. 4.4. Разд. 4.5 содержит описание книги «Практикум по программированию», предназначенной для поддержки практикума. Вопросы использования курса в рамках системы WARE рассматриваются в разд. 4.6. Главу завершают выводы по данной главе.

Приложение содержит полное описание языка Zonnon в его состоянии на октябрь 2004 г.

ЗАКЛЮЧЕНИЕ

Получены следующие научные и практические результаты работы:

1. На основе проведенного анализа моделей и методов открытого образования и дистанционного обучения, а также адаптивных методов и систем гипермедиа обоснована актуальность разработки адаптивных методов и средств поддержки дистанционного обучения программированию, ориентированных на проблемный подход в обучении и объединяющих возможности адаптивных гипермедиа-систем и интеллектуальных обучающих систем.
2. Разработаны модели знаний курса и обучаемого и методы мониторинга знаний обучаемого, поддерживающие адаптивность в рамках проблемного подхода к обучению и позволяющие оценивать знания обучаемого в условиях неполной информации.
3. Разработаны модель и методы тестирования знаний обучаемого, охватывающие известные подходы и позволяющие генерировать сценарий тестирования случайным образом. В основе модели разработанная классификация тестов по форме и глубине проверяемых знаний, а также их объединение в пространства тестов по содержанию проверяемого знания.
4. Разработан проект расширяемой среды, поддерживающей дистанционное обучение программированию в рамках проблемного подхода и соединяющей возможности адаптивных гипермедиа-систем и интеллектуальных обучающих систем. Среда ориентирована на поддержку дистанционного обучения, в процессе которого обучаемые, решая поставленные им индивидуальные задачи, действуют вполне самостоятельно, но постоянно обеспечены возможностью получения квалифицированной помощи, начиная с этапа понимания условий задач и кончая этапом оценки правильности их решения.
5. Разработан вводный курс программирования на базе языка Zonnon. Гипертекстовые учебные пособия «Введение в программирование» и

«Практикум по программированию», поддерживающие курс, размещены на сайте русскоязычной библиотеки учебных курсов международной программы MSDN Academic Alliance .

ПУБЛИКАЦИИ ПО ТЕМЕ ДИССЕРТАЦИИ

1. **Касьянова Е.В.** Освоение технологии программирования ребятами старших классов в процессе выполнения проекта во время летней практики // Тезисы докладов 2-го Всесибирского конгресса женщин-математиков. — Красноярск, КГУ, 2002. — С.92 – 93.
2. **Касьянова Е.В.** Освоение технологии программирования ребятами старших классов в процессе выполнения проекта во время летней практики // Труды 2-го Всесибирского конгресса женщин-математиков. - Красноярск, КГУ, 2002. — С.58 – 61.
3. **Касьянова Е.В., Касьянова С.Н.** Программирование для школьников: сборник задач повышенной сложности с решениями. — Новосибирск, 2002. — 50 С. — (Препр. / РАН. Сиб. Отд-е. ИСИ; № 95).
4. **Kasyanov V.N., Kasyanova E.V.** Web-based systems for supporting computer-science teaching and learning //ACM SIGCSE Bulletin. - New York, ACM Press, 2002. — Vol.34, N 3. — P.238. — (Proc. of the 7th ACM SIGCSE Conference on Innovation and Technology in Computer Science Education)
5. **Kasyanov V.N., Kasyanova E.V.** An environment for Web-based education of programming // HCI International 2003. — Heraklion, Crete University Press, 2003. — P.179 – 180 — (Proc. of the 10th International Conference on Human-Computer Interaction).
6. **Касьянова С.Н., Касьянова Е.В.** Опыт преподавания программирования в старших классах // Пятая международная конференция памяти А.П. Ершова "Перспективы систем информатики". Секция "Информатика образования". Доклады и тезисы. — Новосибирск, ИСИ СО РАН, 2003. — С. 31.
7. **Касьянов В.Н., Касьянова Е.В.** Дистанционное обучение: методы и средства адаптивной гипермедиа // Программные средства и

- математические основы информатики. — Новосибирск, ИСИ СО РАН, 2004. — С. 80–141.
8. **Касьянов В.Н., Касьянова Е.В.** Дистанционное обучение: методы и средства адаптивной гипермедиа // Вычислительные технологии. — 2004. — Т.9, Часть 2. — С. 333 - 341. — (Специальный выпуск по материалам Международной конференции «Вычислительные и информационные технологии в науке, технике и образовании»).
 9. **Касьянова Е.В.** Язык программирования Zonnon для платформы .NET // Программные средства и математические основы информатики. — Новосибирск, ИСИ СО РАН, 2004. — С. 189 – 205.
 10. **Касьянов В.Н., Касьянова Е.В.** Введение в программирование, 2004, 250 С. — <http://www.microsoft.com/Rus/Msdnaa/Curricula/Default.aspx>
 11. **Касьянов В.Н., Касьянова Е.В.** Практикум по программированию, 2004, 200 С. — <http://www.microsoft.com/Rus/Msdnaa/Curricula/Default.aspx>
 12. **Касьянов В.Н., Касьянова Е.В.** Адаптивные системы и методы дистанционного обучения // Информационные технологии в высшем образовании. — 2004. — Т.1, N 4. — С. 40 – 60.
 13. **Касьянова Е.В.** Вводный курс программирования на базе языка Zonnon // Методы и инструменты конструирования и оптимизации программ. — Новосибирск, ИСИ СО РАН, 2005. — С. 95 – 116.
 14. **Kasyanova E.V.** WAPE: an adaptive environment for Web-based education of programming // Proc. of the 17th IMACS World Congress— Paris, 2005. — P. 681 – 686.
 15. **Касьянова Е.В.** Методические и программные средства поддержки дистанционного обучения программированию // Технология Microsoft в теории и практике программирования. Тезисы конференции-конкурса. - Новосибирск, НГУ, 2006. — С. 89 – 91.
 16. **Касьянова Е.В.** Адаптивные технологии дистанционного обучения программированию // Современные проблемы науки и образования. — 2006. — N 2. — С. 92 – 93.

17. **Касьянова Е.В., Касьянова С.Н.** Опыт преподавания информатики в старших классах с математическим уклоном // Современные проблемы науки и образования. — 2006. — № 2. — С. 91 – 92.
18. **Касьянова Е.В.** Адаптивные методы и средства поддержки дистанционного обучения программированию // Шестая международная конференция памяти А.П. Ершова "Перспективы систем информатики". Секция "Информатика образования". Доклады и тезисы. — Новосибирск, ИСИ СО РАН, 2006. — С. 28 – 31.
19. **Касьянова Е.В.** Адаптивные методы и средства дистанционного обучения // Современные ценности и эффективность моделей образовательных систем. Материалы Международной научно-практической конференции. — Новосибирск: Изд-во НИПКИПРО, 2006. — Часть 2. — С. 97 – 100.
20. **Касьянова Е.В.** WARE - адаптивная система поддержки дистанционного обучения программированию // Труды Международной конференции. «Вычислительные и информационные технологии в науке и образовании». — Павлодар, 2006. — Том 1. — С. 606 – 615.
21. **Касьянова Е.В.** Адаптивная система поддержки дистанционного обучения программированию // Проблемы интеллектуализации и качества систем программирования. — Новосибирск, ИСИ СО РАН, 2006. — 18 С.

Подписано в печать

Объем уч.-изд. л.

Формат бумаги 60 × 90 1/16

Тираж 100 экз.

Отпечатано в ЗАО РИЦ «Прайс-курьер»

630090, г. Новосибирск, пр. Ак. Лаврентьева, 6, тел. 334-22-02

Заказ № _____