На правах рукописи

Редькина Александра Васильевна

ИНСТРУМЕНТАЛЬНАЯ ПОДДЕРЖКА МЕТОДОВ ОБУЧЕНИЯ АЛГОРИТМИЗАЦИИ

05.13.11 – Математическое и программное обеспечение вычислительных машин, комплексов и компьютерных сетей

АВТОРЕФЕРАТ
диссертации на соискание ученой степени
кандидата технических наук

Красноярск 2008

Работа выполнена в Институте космических и информационных технологий Сибирский федеральный университет»

Научный руководитель: кандидат технических наук, доцент Кирякова Галина Сергеевна

Официальные оппоненты:

доктор технических наук,

профессор Легалов Александр Иванович

кандидат технических наук,

профессор Сорокин Владимир Афанасьевич

Ведущая организация: ФГОУ ВПО Сибирский государственный технологический университет (г. Красноярск)

Защита состоится 1 июля 2008 года в 14 часов на заседании диссертационного совета ДМ 212.099.05 в Сибирском федеральном университете по адресу: 660074, г. Красноярск, ул. Киренского, 26, ауд. Г418.

С диссертацией можно ознакомиться в научной библиотеке по техническим наукам Сибирского федерального университета.
Тел (3912) - 497869
Факс (3912) - 912198

Автореферат разослан 29 мая 2008 г.

Ученый секретарь диссертационного совета,
кандидат технических наук, профессор Вейсов Е.А.

ОБЩАЯ ХАРАКТЕРИСТИКА РАБОТЫ

Актуальность проблемы. Использование последних достижений в области информационных технологий, разработка новых средств обучения и моделей организации процесса обучения становятся главными аспектами совершенствования современного образования.

Исследования в области компьютерной поддержки учебного процесса имеют более чем 40-летнюю историю. В настоящее время интенсивно развиваются технологии обучения с использованием возможностей глобальной сети Интернет. Работы по интеграции новых информационных технологий в образовательный процесс, способствующих формированию ключевых компетенций и развитию адаптивных методов обучения, представляются весьма перспективными как для традиционного, так и дистанционного образования. Необходимо отметить работы А.В. Соловова, И.П. Норенкова, А.М. Зимина, С.И. Маслова и др.

В настоящее время разрабатываются мощные информационно-образовательные среды (ИОС) (Lotus LMS, WebCT/BlackBoard, MOODLE), обеспечивающие поддержку методического комплекса дисциплин и возможность создания электронных учебников и тестов, но не способствующие закреплению навыков и умений.

Для формирования навыков и умений используются обучающие программы и тренажеры, однако они решают частные задачи, имеют разрозненный характер и не поддерживают полный цикл обучения. Поэтому информационные технологии не оказывают существенного влияния на традиционный образовательный процесс.

Систематическое использование компьютерных технологий для поддержки формирования навыков решения задач может ускорить совершенствование и динамичное развитие новых адаптивных форм обучения. В первую очередь это относится к сфере инженерного и естественнонаучного образования, где чаще всего об уровне знаний обучаемых судят по умению решать задачи. Задача предполагает необходимость сознательного поиска соответствующего средства для достижения цели. Обучение решению задач – наиболее трудно поддающееся развитию направление образовательных информационных технологий. Главная причина – сложность формализации процессов обучения. Большинство форм проверки приобретенных навыков до сих пор не автоматизированы даже частично и отнимают у преподавателя много времени. Разработка таких средств не только освободит преподавателя от рутинной работы, но и даст возможность студентам самостоятельно решать задачи и получать оценку достигнутых результатов.

С одной стороны, есть мощные ИОС, обеспечивающие представление декларативной части знания и управление организационно-методическими мероприятиями, с другой стороны, слабая инструментальная поддержка процесса обучения, включающая получение навыков решения задач.

Актуальной остается задача разработки программного обеспечения, позволяющего формировать навыки решения задач и поддерживать адаптивные методы обучения.

Программирование – одна из дисциплин, которую нельзя освоить без решения задач. Основой курса программирования является раздел «Алгоритмизация». Знания этого раздела определяют всю дальнейшую квалификацию программиста. Поэтому разработку программного обеспечения, поддерживающего процесс решения задач, проведем на примере раздела «Алгоритмизация».

Целью исследований является повышение эффективности и качества подготовки специалистов за счет разработки инструментальных средств для обучения основам алгоритмизации.

Для достижения указанной цели в работе решаются следующие задачи:

1 Анализ существующих ИОС для обоснования целесообразности интеграции разрабатываемой инструментальной поддержки процесса обучения основам алгоритмизации в информационно-образовательную среду.

2 Исследование процесса взаимодействия обучаемого и инструментальной среды, рассматриваемого как управляемый и контролируемый процесс решения задач, учитывающий специфические особенности дисциплины.

3 Исследование возможностей анализа, отладки и верификации алгоритмов на основе их интерактивного преобразования во внутреннее представление, позволяющее автоматизировать процесс обучения основам алгоритмизации.

4 Формализация и алгоритмизация процесса обучения решению задач в соответствии с психологически обоснованными этапами познавательной деятельности.

5 Разработка функциональной модели и архитектуры инструментальной поддержки обучения основам алгоритмизации.

6 Проведение экспериментальных исследований для подтверждения реализуемости и доказательства эффективности предложенной модели процесса обучения путем моделирования системы на языках программирования.

Методы исследования. В работе использованы понятия и методы теории графов, теории алгоритмов, теории систем, формальных грамматик, теории языков программирования, методы структурного анализа и проектирования (SADT – Structured Analysis & Design Technique). Для формализации процесса обучения синтезу алгоритмов применялись расширенные формы Бэкуса-Наура.

Научная новизна.

1 Разработана обобщенная модель процесса обучения навыкам решения задач, обеспечивающая порождение различных конфигурационных аспектов построения программного комплекса.

2 Предложен обобщенный многоэтапный алгоритм поддержки формирования навыков решения задач в инструментальной среде, отличительной чертой которого является свойство динамической адаптации к уровню знаний обучаемого.

3 Предложена модель анализа и верификации алгоритмов на основе их интерактивного преобразования во внутреннее представление, позволяющее автоматизировать процесс обучения основам алгоритмизации.

4 Разработана совокупность алгоритмов реализации интерпретатора и редактора блок-схем, отличающегося возможностью преобразования и сохранения математической модели и графического образа в XML-файле.

5 Разработана архитектура программного обеспечения, поддерживающего процесс обучения основам алгоритмизации, использующего язык предметной области и адаптацию траектории обучения к уровню знаний обучаемого.

Практическая ценность работы.

1 Разработаны алгоритмические и программные средства, которые могут быть применены для поддержки обучения навыкам решения задач по различным дисциплинам, позволяющим формализовать процесс представления и контроля решений.

2 Проведенные автором исследования позволили создать инструментальную среду «Algorithm», обеспечивающую поддержку формирования навыков решения задач по основам алгоритмизации и способствующую повышению эффективности процесса обучения.

3 Реализованы визуализаторы-демонстраторы «ListView» и «GraphView», позволяющие в интерактивном режиме показать алгоритмы обработки динамических списков и поиска путей в графах, что повышает уровень понимаемости теоретического материала.

4 Визуализаторы «ListView» и «GraphView» внедрены в учебный процесс в Сибирском федеральном университете, интегрированы в ИОС Moodle и используются при изучении курса «Программирование на языке высокого уровня» и доступны на сайте http://distant.fivt.krgtu.ru.

Достоверность научных положений, выводов и практических рекомендаций, полученных в диссертации, подтверждается экспериментальными исследованиями и сравнительным анализом существующих подходов к решению поставленной задачи, точной формулировкой критериев, компьютерным моделированием, а также результатами использования алгоритмов, предложенных в диссертации, на практике.

Апробация диссертации. Основные положения диссертационной работы представлялись на XII международной конференции «Современное образование: содержание, технологии, качество» (С-Петербург, 2006), Всероссийской научно-практической конференции «Повышение качества непрерывного профессионального образования» (Красноярск, 2005), на научно-

практической конференции «Совершенствование качества подготовки специалистов» (Красноярск, 2002), на Всероссийской научно-практической конференции «Достижения науки и техники – развитию сибирских регионов» (Красноярск, 2003), на Всероссийской научной конференции «Наука и образование» (Белово, 2003), на научно-практической конференции «Управление качеством в новых информационных технологиях, системах и образовании» (Москва, 2003), на семинаре в институте вычислительного моделирования СО РАН (Красноярск, 2007).

Сведения о внедрении. Визуализаторы «GraphView» и «ListView» используются в учебном процессе, что подтверждено актами внедрения.

Работа выполнена при поддержке научно-методического проекта Сибирского Федерального университета №10 «Решение некоторых задач прикладной математики и информатики для повышения потенциала образовательного процесса», инновационной образовательной программы «Инновационная образовательная программа Института фундаментальной подготовки» для базового цикла математических и естественнонаучных дисциплин по дисциплине «Информатика и программирование».

Публикации. По теме диссертации опубликовано 12 работ, в том числе, четыре статьи и семь работ в материалах всероссийских и международных научно-практических конференций, из которых 2 статьи в изданиях по списку ВАК, получено одно свидетельство о регистрации программ для ЭВМ.

Личный вклад автора. Автору принадлежат идеи работы, определение цели и постановка задачи, модели и алгоритмы разработанной инструментальной поддержки процесса обучения. Основные результаты, изложенные в работе, получены либо непосредственно автором, либо с его участием.

Структура диссертации. Диссертация содержит 145 страницы и состоит из введения, четырех глав, заключения и четырех приложений. В списке использованных источников указано 104 наименования. Работа иллюстрирована 34 рисунками и 5 таблицами.

СОДЕРЖАНИЕ РАБОТЫ

Во введении представлены цели и задачи исследования, раскрывается актуальность, новизна полученных результатов, практическая значимость. Приводится перечень вопросов, выносимых на защиту.

В первой главе проведен обзор существующих ИОС и обучающих программ, поддерживающих процесс обучения языкам программирования, проанализированы подходы к построению систем, поддерживающих процесс обучения решению задач по основам алгоритмизации, исследуются возможности анализа, отладки и верификации алгоритмов на основе их автоматического преобразования в ориентированный граф.

С целью исследования возможностей, представляемых для обучения алгоритмизации, проведен анализ существующих программ, обучающих языкам программирования. В настоящее время среди программ, содержащих

элементы обучения алгоритмизации, можно выделить программы на олимпиадных сайтах по программированию, где разбираются и тестируются задачи в on-line режиме. Представляют интерес программы «Конструктор блок-схем (Flowchart builder)» Нижегородского государственного университета и «AlgoMaster» Национального технического университета Украины (Киевский политехнический институт). Эти программы предназначены для исследования алгоритмов и содержат подсистемы интерактивного ввода, интерпретации и анимации. Они являются хорошим дидактическим материалом, но не содержат элементов управления процессом обучения, не поддерживают элементов контроля знаний. Таким образом, средств, выполняющих полный цикл обучения алгоритмизации и поддерживающих адаптивные формы обучения нет.

Анализ информационно-образовательных сред проведен по следующим критериям: функциональная полнота, переносимость, наличие стандартизированных инструментальных средств адаптации и сопровождения, открытость кода, стоимость, соответствие международным стандартам обмена электронными курсами.

На основе анализа существующих ИОС, учитывая их функциональные возможности, представляется целесообразной идея интеграции разрабатываемой инструментальной поддержки процесса обучения решению задач в информационно-образовательную среду для взаимного развития и расширения функциональных возможностей.

Проведен анализ подходов к построению систем, поддерживающих процесс обучения решению задач по основам алгоритмизации. Для представления алгоритмов выбран графический способ их описания. Он дает возможность начинать решать задачи до изучения языков программирования. Благодаря наглядности, объединяющей графический образ со смысловым содержанием, обеспечивается высокая «читаемость» алгоритма и явное отображение управления в нем. Инструментальная поддержка с одной стороны предоставляет возможность самостоятельного построения блок-схем, с другой стороны – контролирует правильность построения алгоритма.

Для построения блок-схем предлагается использовать графический редактор. Ответом обучаемого является графическое изображение, а для анализа ответа необходима математическая модель. Известно, что для решения одной и той же задачи можно составить несколько корректных алгоритмов, поэтому проверка на идентичность изображений не позволит вынести окончательный вердикт. Возникает необходимость распознать блок-схему, и только потом ее исследовать. Более простым является решение предоставить пользователю ограниченный набор графических инструментов в виде геометрических фигур, используемых в блок-схемах алгоритмов, и распознавать действия пользователя при построении блок-схемы, на этом и строить математическую модель алгоритма. Главная отличительная черта графического редактора – отсутствие выделенного этапа преобразования. Преобразование схемы во внутреннее представление, пригодное для дальнейшего моделирования, осуществляется в процессе рисования. К моменту завершения рисования

блок-схемы уже готова математическая модель. Эта модель, представленная ориентированным графом, используется для исполнения алгоритма, отладки и его верификации. Для исполнения алгоритма разрабатывается интерпретатор. Самым трудоемким является процесс проверки правильности алгоритма. Автоматизация этого процесса является ключевым моментом всего процесса обучения.

Проверку правильности алгоритмов предлагается вести путем сравнения результатов работы испытуемой блок-схемы алгоритма, построенной обучаемым, с результатами работы эталонной блок-схемы алгоритма, составленной экспертом. Эксперт для проверки разрабатывает наборы входных и выходных данных. Выходом из данной ситуации является тестирование результатов работы исследуемого алгоритма относительно эталонного. Условием, обеспечивающим процесс тестирования, является однозначное отображение множества входных данных $\{X\}$ в соответствующий набор выходных данных $\{Y\}$:

$$G : \{X\} \rightarrow \{Y\} \qquad (1)$$

Закон отображения определяется испытуемым алгоритмом G. На вход тестируемого алгоритма поступает вектор входных данных $x = \{x_1, x_2, \dots x_n\}$. На выходе по закону (1) получаем результаты работы тестируемого алгоритма $y = \{y_1, y_2, \dots y_m\}$, подающиеся на вход системы тестирования, где они и обрабатываются. В системе тестирования задан алгоритм H, разработанный экспертом, однозначного отображения множества $\{X\}$ в $\{Z\}$:

$$H : \{X\} \rightarrow \{Y\} \qquad (2)$$

Условием верности работы испытуемого алгоритма является условие:

$$G : \{X\} \rightarrow \{Y\} = H : \{X\} \rightarrow \{Y\}, \text{ т.е. } \{Y\} \rightarrow \{Z\} \qquad (3)$$

Испытуемый алгоритм можно рассматривать как верный, если результаты его работы совпадают с результатами работы алгоритма эксперта.

Используя ориентированный граф, в инструментальной поддержке предлагается провести семантический анализ алгоритма путем проверки на достижимость вершин и исследование оптимальности испытуемого алгоритма по объему памяти и времени относительно алгоритма эксперта. Следовательно, для анализа и проверки алгоритма предлагается трехступенчатая схема, включающая семантический анализ, тестирование и определение оптимальности.

Таким образом, можно частично автоматизировать анализ и проверку правильности алгоритма. Окончательный анализ алгоритма остается за преподавателем.

Во второй главе построена обобщенная модель процесса обучения, проведена формализация процесса формирования навыков решения задач, а

также разработан обобщенный алгоритм управления процессом обучения при изучении основ алгоритмизации.

Для эффективного взаимодействия с таким сложным объектом, как обучаемый, невозможно заранее создать точную и полную траекторию обучения, необходимы знания об обучаемом, изучаемой им дисциплине и возможностях управления учебным процессом.

Поэтому процесс обучения основам алгоритмизации рассматривается как управляемый и контролируемый процесс решения задач, обобщенную модель которого можно представить как тройку:

$$M = \langle УП, МО, СО \rangle,$$

где УП – модель структуры управления процессом обучения, МО – модель обучаемого и СО – модель специфических особенностей изучаемой дисциплины.

При построении структуры управления необходимо учитывать все составляющие деятельностной части учебного процесса, включая организационные и учебные мероприятия. Структура управления процессом обучения определяет правила и алгоритмы взаимодействия пользователя (студента, преподавателя, эксперта) с обучающей системой. Роль преподавателя играет сама обучающая система.

Обучение включает в себя два взаимосвязанных процесса: организацию деятельности обучаемого и контроль этой деятельности. Эти процессы непрерывно взаимодействуют: результат контроля влияет на содержание управляющих воздействий, т. е. на дальнейшую организацию деятельности. В свою очередь, организация определенной деятельности требует и адекватной формы контроля, и конкретного способа регистрации этой деятельности.

Мероприятия обучения связаны с предоставлением декларативной и процедурной части знаний. Декларативная часть знания – это изложение теоретического материала, который представляют в виде текстов и графических изображений. Контроль усвоения этих знаний обычно реализован обычно в виде тестов.

Процедурная часть знаний – это выработка навыков и умений. В настоящее время еще не созданы средства описания и изображения мыслительной деятельности при решении задач. Нет общего способа решения широкого класса задач, преподаватель обычно учит решать некоторую конкретную задачу или класс задач. Обучаемый становится активным участником процесса, ведущим поиск решения задач. Так как любые действия обучаемого контролировать и анализировать сложно, то необходимо ограничение языка взаимодействия пользователя и системы, формализация процесса обучения и формирование стратегии интерактивного обучения. Управление процессом решения задач строится на управлении процессами выдачи формализованных заданий, организации действий при подготовке решения задач, контроле полученных решений и генерации адекватных подсказок-объяснений.

Специфические особенности дисциплин определяют форму предъявления и контроля знаний. К специфическим особенностям дисциплины основы алгоритмизации следует отнести:

- возможность решить правильно одну и ту же задачу множеством способов;
- формирование специфического алгоритмического мышления;
- овладение навыками дедуктивных рассуждений;
- использование графического представления решения задачи;
- освоение смежных дисциплин, необходимых для решения задачи.

Для системы обучения алгоритмизации инструментами поддержки являются графический редактор блок-схем, интерпретатор блок-схем и система тестирования алгоритмов. Диагностика ошибок ведется как во время подготовки решения задачи, так и после для проверки правильности и оптимальности окончательного решения.

Модель обучаемого отражает состояние знаний обучаемого. Она является динамичной и идентифицирует уровень усвоения текущего материала, а также начальный и результирующий уровни. К модели обучаемого относится совокупность характеристик, измеряемых в процессе решения задач, и методы обработки этих характеристик.

В адаптивной обучающей системе модель обучаемого рассматривается как набор шаблонов (рисунок 1) и совокупность правил, которые на основании значений характеристик управляют процессом взаимодействия системы с обучаемым. Для каждого обучаемого в соответствии с шаблоном задается начальная настройка системы, определяется траектория работы в системе и назначается свое подмножество решаемых задач.

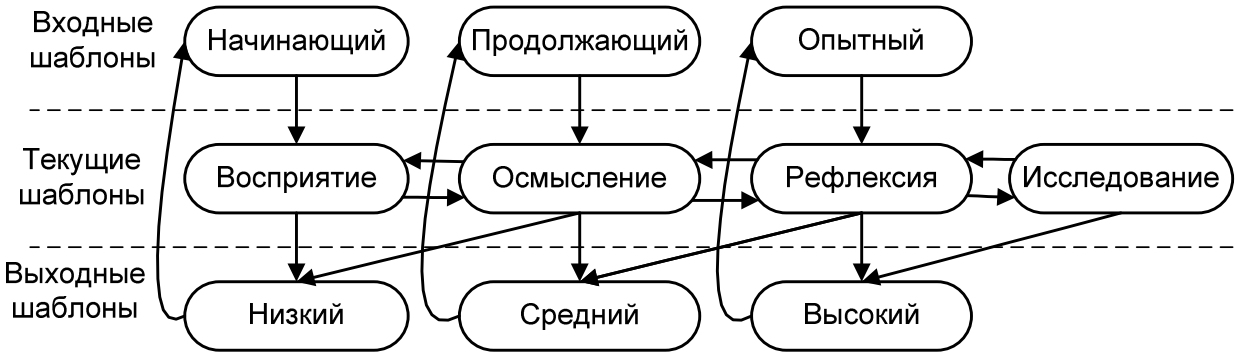

Рисунок 1 – Граф перемещения обучаемого по шаблонам

Правила обработки характеристик обучаемого основаны на диагностике состояния знаний и навыков обучаемого, т.е. сложности решаемых задач, а также сущности и степени совершаемых им ошибок. В первую очередь эти правила проводят к изменению самой модели обучаемого по результатам его работы с системой и определяют действия по приведению его к состоянию пригодности для дальнейшего процесса обучения. В начале обучения шаблоны идентифицируются на основании собственной оценки обучаемого с уче-

том результатов тестирования по теории к данной теме. В дальнейшем шаблоны корректируются в зависимости от результатов решения задач.

Во время работы с обучающей системой позиционирование шаблонов МО проводится согласно психологически обоснованной последовательности этапов познавательной деятельности: восприятие, осмысление, рефлексия и исследование.

Правила расчета характеристик для идентификации обучаемого по шаблону во время работы с системой определяются уровнем решаемых задач и допущенных ошибок. Формализация принципов позиционирования осуществляется следующим образом.

Пусть A множество задач по некоторой теме:

$$A = \{ a_1,\ a_2, \dots a_{n1}\}$$

Каждая a_i задача характеризуется шестеркой вида:

$$X = \langle t, o, s,\ R, P, d \rangle,$$

где t – весовой коэффициент самостоятельности в решении задачи;

o – максимальная оценка задачи в баллах, отражающая уровень сложности задачи по данной теме;

s – рубрика, к которой относится задача, характеризующая соответствующее уникальное подмножество стратегий решения,;

R– множество тестовых наборов данных, подготовленных для задачи a_i;

P – множество подсказок и комментариев, подготовленных для задачи a_i;

d – коэффициент оптимальности, отражающий степень оптимальности решения задачи обучаемым относительно решения эксперта. Этот коэффициент определяется следующим образом:

$$d = dm \times dt,$$

где dm – коэффициент оптимальности по памяти, вычисляемый как отношение количества переменных (байт) в алгоритме эксперта к количеству переменных (байт) в алгоритме обучаемого;

dt – коэффициент оптимальности по времени, вычисляемый как отношение времени выполнения алгоритма эксперта к времени выполнения алгоритма обучаемого на эталонных данных.

Определены правила идентификации текущих шаблонов обучаемых. Для этого разработана шкала оценивания. Анализ экспериментальных данных подтвердил существование некоторых общих тенденций в динамике обучения решению задач. Осознание общности и, следовательно, "открытие" эквивалентности происходит в среднем при столкновении с третьей задачей. Поэтому для перехода с одного шаблона на другой необходимо решить не менее трех задач.

Задача оценивается в максимальное количество баллов, если она решена полностью, то есть она выполнилась успешно на всех наборах тестовых дан-

ных. В случае неверного ответа на каком-либо тесте количество баллов снижается.

Кроме того, к снижению количества набранных баллов приводит и использование некоторых подсказок. При этом должны быть решены разнотипные задачи, т.е. принадлежащие разным рубрикам (не менее чем двум). Если обучаемый не справляется с задачами, соответствующими текущему шаблону, то он идентифицируется по тому шаблону, где предлагаются задачи, с меньшим коэффициентом самостоятельности. В результате МО представляется графом перемещения обучаемого по шаблонам (рисунок 1).

Выходные шаблоны базируются на текущих шаблонах и определяются с использованием элементов нечеткой логики на основании степени принадлежности к тому или иному текущему шаблону.

Общее количество набранных баллов рассчитывается по числу набранных баллов k_i за каждую с учетом весового коэффициента самостоятельности t_i и коэффициента оптимальности d_i в решении задачи:

$$\mathrm{Re}\,s = \sum_{i=1}^{l} t_i \times k_i \times d_i ,$$

где l – количество задач в подмножестве B $(B \subset A)$, решенных пользователем.

Предлагаемая модель обучаемого является динамической и позволяет судить об уровне знаний обучаемого по умению решать задачи.

Таким образом, используя модель обучаемого, строится адаптивная обучающая программа, учитывающая уровень знаний обучаемого. Она позволяет ошибаться, давая возможность познавать основы алгоритмизации через противоречия, а это является существенной частью процесса формирования мышления.

Формализация процесса управления обучением решению задач проведена также согласно психологически обоснованным этапам познавательной деятельности теории познания и представлена многоэтапной моделью:

1. Этап А – демонстрационный. На этом этапе изучается теория, обучаемый познакомится и детально проанализирует типовые алгоритмы с помощью визуализатора. Визуализатор – это программа, в процессе работы которой на экране компьютера динамически демонстрируется применение алгоритма к выбранному набору данных.

2. Этап В – обучение по аналогии. Здесь предлагаются задачи для решения, для которых есть аналогичное решение среди типовых алгоритмов. Обучаемому разрешается анализировать алгоритм визуализатором. На этом этапе подключается семантический анализатор, ведется отладка, тестирование и оптимизация алгоритма обучаемого относительно алгоритма эксперта.

3. Этап С – обучение по конечному результату. На этом этапе студент полностью самостоятельно решает выбранную задачу, пользуясь редактором и визуализатором, а обучающая система контролирует правильность построения и верность результатов путем тестирования.

4. Этап D – обучение в эксперименте. На последнем этапе обучаемый строит блок-схему алгоритма произвольной задачи, а обучающая система только контролирует соответствие структуры блок-схемы алгоритма правилам построения, позволяет провести ее трассировку и отладку, а за верность результата несет ответственность обучаемый.

Согласно предложенной модели и формализации процесса обучения решению задач разработан обобщенный алгоритм взаимодействия обучаемого с инструментальной поддержкой. Обучение проходит в четыре этапа. Для каждого этапа разработаны алгоритмы управления процессом обучения. В зависимости от стартовых шаблонов устанавливается точка входа в учебный процесс. Дальнейшее перемещение по шаблонам определяется уровнем решаемых задач и допущенных ошибок. Каждому этапу обучения ставится в соответствие шаблон модели обучаемого. Для выбора задач к решению используется стратифицированная выборка. Задачи разбиты по группам в соответствии с этапами обучения, а внутри групп – подгруппы, определяемые сложностью задач. Выбор подгруппы основывается на степени принадлежности обучаемого к текущему шаблону. Чем выше принадлежность, тем сложнее предлагаются задачи. Выбор задачи из подгруппы производится методом случайного поиска с исключением. Таким образом, осуществляется адаптация к уровню знаний обучаемого.

Модель процесса обучения рассматривается как сложный процесс, интегрировавший в себя управление процессом обучения, представление об обучаемом и специфических особенностях дисциплины.. Предложенная модель процесса обучения может использоваться для разработки программного обеспечения по таким дисциплинам, где исходя из специфических особенностей дисциплины, можно создать математическую модель на основе регистрации действий обучаемого при решении задач.

В третьей главе предлагается архитектура автоматизированной системы обучения синтезу алгоритмов. Предложена функциональная модель обучения синтезу алгоритмов, разработанная с использованием методов структурного анализа и проектирования (SADT – Structured Analysis & Design Technique).Разработаны структурные схемы подсистем и базы знаний.

На основе обобщенной модели процесса обучения и алгоритма управления обучением разработана функциональная модель инструментальной поддержки, представляющая совокупность диаграмм. Управление обучением основам алгоритмизации производится с использованием инструментальной поддержки согласно обобщенному алгоритму, с учетом правил построения блок-схем, на основе представленного экспертом теоретического и практического материала и начального уровня знаний обучаемого (рисунок 2).

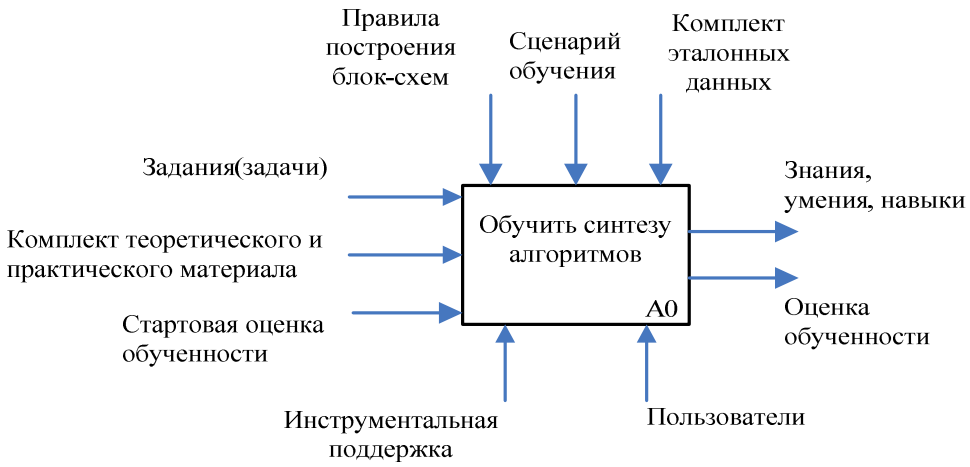

Рисунок 2 – Диаграмма верхнего уровня модели инструментальной среды

Дальнейшая детализация функциональной модели представлена на рисунке 3.

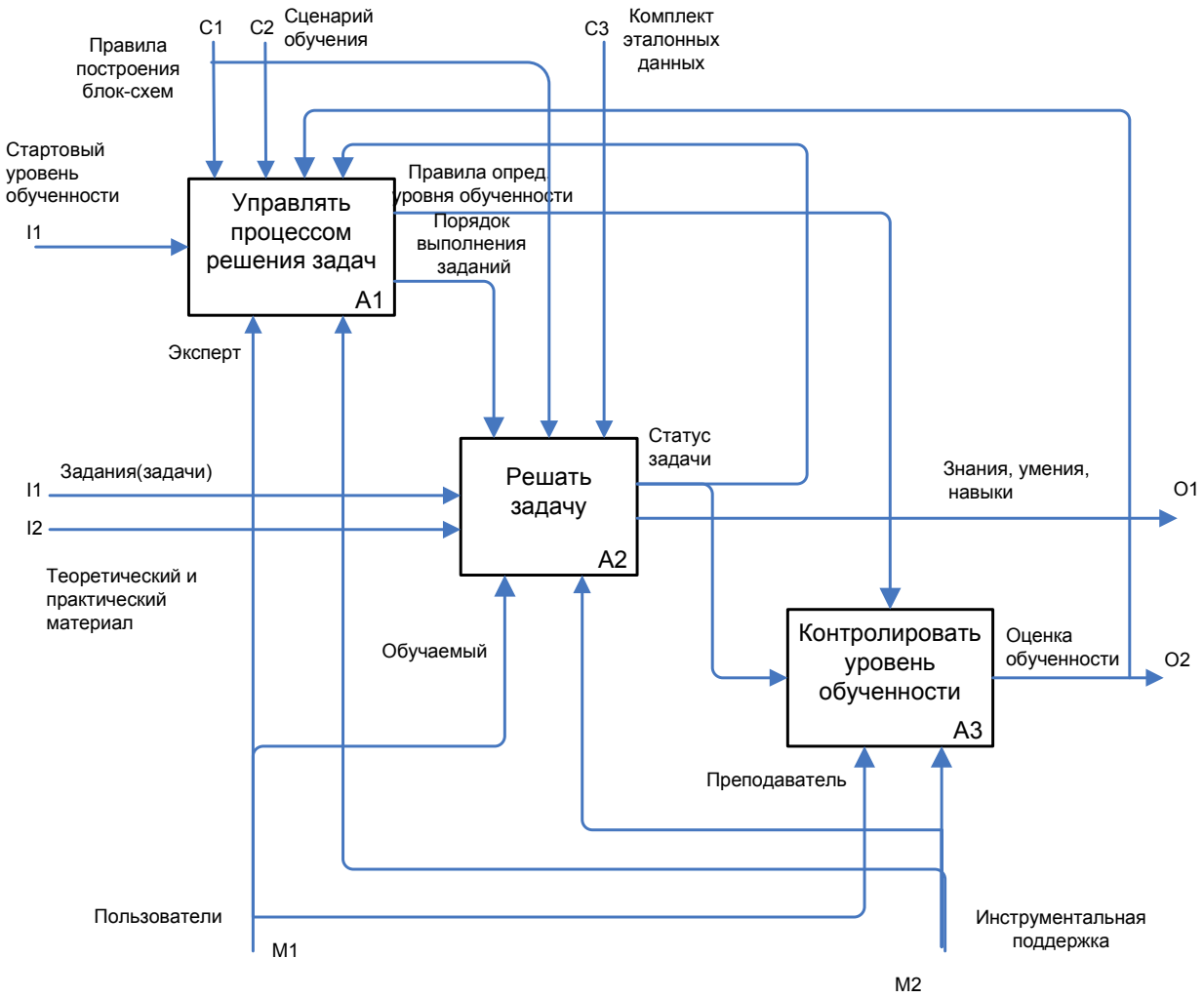

Рисунок 3 – Диаграмма первого уровня модели инструментальной среды

Диаграмма первого уровня указывает на три главные функции системы обучения: управление решением задач, решение задач и контроль уровня

обученности. Управление решением задач определяется не только обобщенным алгоритмом и начальным уровнем знаний обучаемого, но и статусом решаемых задач. Статус задачи учитывает степень решенности задачи, коэффициенты сложности и самостоятельности, а также ошибки, допущенные при решении. Он позволяет позиционировать обучаемого по текущему шаблону, сформировать адекватное управляющее воздействие при выборе следующей задачи. Для тестирования алгоритмов эксперт подбирает данные таким образом, чтобы выявить типичные ошибки, ошибки времени выполнения, проверить все граничные случаи. Это позволяет судить о возможных причинах ошибок, допускаемых обучаемым.

На основе функциональной модели, модели процесса обучения и многоэтапного алгоритма обучения основам алгоритмизации разработана архитектура инструментальной системы, обеспечивающей поддержку процесса обучения алгоритмизации (рисунок 4).

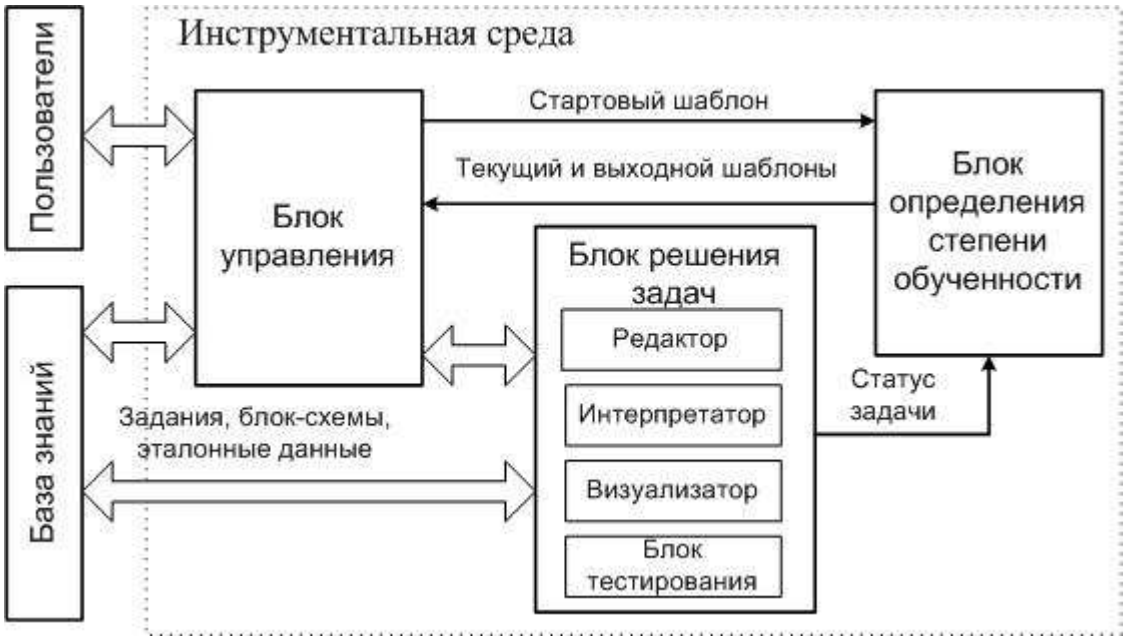

Рисунок 4 – Архитектура инструментальной системы

Блок управления осуществляет координацию взаимодействия всех блоков, контролирует реализацию сценария. Блок решения задач обеспечивает организацию процесса решения задач. Его основными составляющими являются: редактор, интерпретатор, визуализатор и блок тестирования

Блок определения степени обученности позиционирует шаблон обучаемого по правилам, заданным моделью обучаемого, в соответствии со статусом задачи.

Согласно полученному заданию пользователь, применяя графическую библиотеку, строит блок-схему алгоритма. Одновременно преобразователь редактора строит внутреннее представление в виде ориентированного графа. Преобразователь является двухступенчатым и преобразует блок схему во внутреннее представление, а затем – в XML-файл, также делает обратное

преобразование. Сохранение в XML-файле математической модели и графического образа позволяет вернуться к исполнению и редактированию блок-схемы алгоритма.

Для выполнения алгоритмов разработан интерпретатор, который исполняет алгоритм по внутреннему представлению. Входной язык интерпретатора состоит из двух языков: языка блок-схем и языка наполнения блок-схем. Язык блок-схем представляется набором геометрических фигур, представляемых редактором, поэтому функции по распознаванию геометрических фигур возложены на редактор. Отладка алгоритмов производится в интерактивном режиме, что дает возможность фиксировать и исправлять алгоритмические ошибки в схемах. Отладка может производиться в пошаговом и автоматическом режиме. Ход выполнения алгоритма отображается графическим выделением выполняемых участков блок-схемы алгоритма.

Для разработки инструментальной системы выбрана фреймовая модель представления знаний, так как она эффективна для структурного описания сложных понятий и решения задач, в которых в соответствии с ситуацией применяются различные способы вывода. Предлагаемая модель представления знаний состоит из двух фреймов. Один фрейм связан с представлением задач, а второй описывает характеристики студентов. Такая организация позволяет информацию, помещаемую в базу знаний, автоматически согласовать с уже имеющейся информацией, проверять семантику и непротиворечивость новой информации. Часть информации в базу знаний готовит эксперт, а другая часть автоматически извлекается из первой. Например, эксперт готовит для задачи формулировку, блок-схему алгоритма, тестовые наборы входных данных, а такие параметры как количество переменных (байт) и время выполнения алгоритма на эталонных данных определяется с помощью присоединенных процедур. Изменения в блок-схеме, инициированные экспертом, автоматически приводят к необходимости переопределения зависимых параметров.

Инструментальная система имеет два режима работы: режим приобретения знаний и режим решения задач. В режиме приобретения знаний эксперт дополняет базу знаний новыми задачами, пользуясь теми же самыми инструментами, но с другими полномочиями, и по другому алгоритму.

Четвертая глава посвящена экспериментальным исследованиям модели автоматизированной обучающей системы. Исследования велись в двух направлениях:

1 Разработка локальной версии инструментальной среды «Algorithm», содержащей графический редактор, интерпретатор, визуализатор и блок проверки решений, для экспериментального подтверждения эффективности применения инструментальной поддержки в учебном процессе.

2 Разработка инструментальной среды «GraphView», поддерживающая редактор графов и визуализацию поиска путей в них. Для исследования интеграции инструментальной среды в информационно-образовательную среду локализована ИОС MOODLE, в которой разработан курс «Программирование». В курс «Программирование» интегрирована среда «GraphView».

Апробация инструментальной среды «Algorithm» проводилась на примере усвоения курса «Программирование на языке высокого уровня» в политехническом институте Сибирского Федерального университета. Целью эксперимента было определить качественные изменения уровня усвоения учебного материала, возникающие при применении инструментальной среды «Algorithm». В эксперименте принимало участие 8 групп общей численностью 210 человек. Во всех группах основы алгоритмизации изучались с использованием блок-схем. Студенты первых четырех групп (104 человека) проходили обучение по обычной форме подготовки (без применения среды «Algorithm») – контрольный поток. Студенты других четырех групп (106 человек) обучались с использованием среды «Algorithm» – экспериментальный поток.

По окончании курса обучения алгоритмизации студентам было выдано итоговое задание, по результатам которого можно судить о полученных навыках. Итоговое задание выполнялось письменно и содержало по две задачи на алгоритмизацию. Если студент не справлялся с итоговым заданием, то ему предоставлялись две дополнительные попытки. Результаты экспериментов (таблица 1) подтверждают эффективность использования инструментальной среды. Экспериментальная группа, использовавшая в обучение среды «Algorithm», справилась с итоговым заданием, сократив на 19% количество попыток. Уровень подготовки обучаемых повысился, а время, затрачиваемое преподавателем на рутинную работу по контролю правильности решения задач, на подборку тестов сократилось.

Таблица 1 Анализ результатов выполнения итогового задания

Показатели	Контрольный поток				Экспериментальный поток			
	Группа 1	Группа 2	Группа 3	Группа 4	Группа 1	Группа 2	Группа 3	Группа 4
Число студентов	26	25	26	27	27	27	25	26
1 попытка	10	8	10	12	15	15	13	14
2 попытка	9	10	8	8	10	9	10	9
3 попытка	7	7	8	7	2	3	2	3
Среднее количество попыток по группам	1,88	1,96	1,92	1,89	1,51	1,55	1,56	1,58
Среднее количество попыток по потоку	1,91				1,55			

В приложении А приведены акты о внедрении в учебный процесс визуализаторов «GraphView» и «ListView».

В приложении В представлены алгоритмы этапов процесса обучения алгоритмизации.

В приложении С содержатся фрагменты программы«GraphView».

В приложении D приведены примеры формулировок задач для инструментальной поддержки процесса обучения решению задач.

ОСНОВНЫЕ РЕЗУЛЬТАТЫ РАБОТЫ

1 Разработана обобщенная модель процесса обучения, обеспечивающая порождение различных вариантов построения программных систем.

2 Предложен обобщенный многоэтапный алгоритм управления процессом решения задач, отличительной чертой которого является свойство динамической адаптации к уровню знаний обучаемого.

3 Разработана модель обучаемого как набор шаблонов и совокупность правил, которые на основании значений характеристик, определяемых уровнем решаемых задач и допущенных ошибок, управляют процессом общения системы с обучаемым.

4 Предложена модель анализа и верификации алгоритмов на основе их преобразования в форму, позволяющую автоматизировать процесс обучения основам алгоритмизации.

5 Разработана совокупность алгоритмов реализации интерпретатора и редактора блок-схем, отличающегося возможностью преобразования и сохранения математической модели и графического образа в XML-файле.

6 Разработана архитектура системы и программного обеспечение, позволяющее поддерживать процесс обучения навыкам построения блок-схем, использование языка предметной области и адаптация траектории обучения для каждого студента на основе модели обучаемого.

7 Разработано программное обеспечение, позволяющее поддерживать формирование практических навыков, повышающих уровень понимаемости теоретического материала.

8 В результате экспериментальных исследований показана эффективность применения инструментальной поддержки для формирования навыков решения задач.

Результаты работы отражены в следующих публикациях:

1 Редькина, А.В. Обучение синтезу алгоритмов / А.В. Редькина // Вестник СибГАУ. – 2008. –№1(18) – С. 30-34.

2 Редькина, А.В. Автоматизированная система обучения синтезу алгоритмов / А.В. Редькина // Системы управления и информационные технологии. – 2007. –№2.2(28). - С. 280-284

3 Редькина, А.В. Информационные технологии для решения дидактических задач в курсе программирования / А.В. Редькина // Информационные технологии моделирования и управления. Научно-технический журнал. Воронеж: Изд-во «Научная книга». – 2007. – Вып. 5(39) – С. 523-528.

4 Редькина, А.В. Визуализация поиска путей в графе «GraphView» / А.В. Редькина, Малявко С.Н. // Свидетельство об официальной регистрации программы для ЭВМ №2007613730 / Федеральная служба по интеллектуальной собственности, патентам и товарным знакам. – 31.08.2007

5 Редькина, А.В. Открытые технологии в непрерывном образовании / А.В. Редькина, Покидышева Л. И., Редькин А. В // Современное образование: содержание, технологии, качество. Материалы XII международная конференция. С-Петербург: Изд-во СПбГЭТУ «ЛЭТИ». – 2006, Т1 – С. 65-67.

6 Редькина, А.В. Обучающие системы алгоритмизации / А.В. Редькина // Совершенствование качества подготовки специалистов. Материалы научно-практической конференции. Красноярск: Изд-во КГТУ. –2002. – С. 225.

7 Редькина, А.В. Система проведения дистанционных конкурсов по программированию / А.В. Редькина, Л. И. Покидышева, А. В. Редькин // Наука и образование. Материалы Всероссийской научной конференции. Белово. – 2003. – С. 185-186.

8 Редькина, А.В. Подсистема тестирования лабораторных работ по программированию/ А.В. Редькина, А. В. Редькин // Управление качеством в новых информационных технологиях, системах и образовании. Материалы Всероссийской научной конференции. Москва. – 2003. С–125-126.

9 Редькина, А.В. Тесты в системе дистанционного образования по курсу «Алгоритмические языки и программирование» / А.В. Редькина, Редькин А. В // Достижения науки и техники – развитию сибирских регионов. Материалы Всероссийской научно-практической конференции. Красноярск: Изд-во КГТУ. – 2003. – С. 159-160.

10 Редькина, А.В. Система дистанционного образования по курсу «Алгоритмические языки и программирование» / А.В. Редькина, А. В. Редькин // Вестник КГТУ, Вып. 33, Математические методы и моделирование. Красноярск: Изд-во КГТУ. – 2004. – С. 249-256.

11 Редькина, А.В. Программная среда для обучения Moodle / А.В. Редькина, А. В. Редькин // Повышение качества непрерывного профессионального образования. Материалы Всероссийской научно-практической конференции. Красноярск: Изд-во КГТУ. – 2005. – С.236-239.

12 Редькина, А.В. Обучающие системы / А.В. Редькина // Проблемы информатизации региона (ПИР-2000). Материалы Шестой Всероссийской научно-практической конференции: Красноярск: КГТУ. – 2000. – С. 174.

Редькина Александра Васильевна
Инструментальная поддержка методов обучения алгоритмизации
Автореф. дисс. на соискание ученой степени кандидата технических наук
Подписано в печать 23.05.2008. Заказ №
Формат 60x90/16. Усл. печ. л. 1,3. Тираж 100 экз.
ИПК Сибирского федерального университета
660074, Красноярск, ул. Киренского, 28