

УДК 004.822:004.891.2

ИНТЕЛЛЕКТУАЛЬНЫЕ HELP-СИСТЕМЫ КАК СРЕДСТВО ОБУЧЕНИЯ ТЕХНОЛОГИЯМ ПРОГРАММИРОВАНИЯ

Пивоварчик О.В.

Учреждение образования «Барановичский государственный университет»

Объектом рассмотрения являются help-системы по технологиям разработки программ. В статье рассматриваются возможности использования help-систем в качестве компьютерных средств обучения. Описывается интеллектуальная help-система по технологии разработки программ, ориентированных на обработку семантических сетей, и возможности ее использования в обучении ИТ специалистов.

Ключевые слова: компьютерные средства обучения, help-система, технологии программирования

Введение

Настоящее время характеризуется внедрением информационных технологий во все сферы жизнедеятельности человека. В связи с этим повышается спрос на ИТ специалистов высокой квалификации. Поэтому актуальной является проблема подготовки квалифицированных специалистов в данной области. Кроме этого, в среде постоянного усовершенствования технологий разработки программного обеспечения (ПО), существующего многообразия языков программирования, высоких темпов развития современных инструментальных средств актуальной является проблема повышения уровня квалификации ИТ специалистов. Использование компьютерных средств обучения (КСО) [2] в подготовке ИТ специалистов позволяет повысить качество подготовки специалистов, снизить затраты на обучение [2], повысить оперативность. В данной работе в качестве КСО предлагается использовать интеллектуальную help-систему по технологии разработки программ, интегрированную с инструментальным средством и являющуюся консультантом в данной области. Под help-системой в работе понимается программное обеспечение, оказывающее помощь пользователю в эксплуатации технической системы.

Аналитический обзор существующих help-систем

Идеи использования компьютера в учебном процессе нашли свое выражение в 60-е годы прошлого столетия в форме концепции программированного обучения. В 70-е годы появились первые интеллектуальные обучающие системы. В настоящее время создано огромное количество программ, в той или иной мере относящихся к классу КСО. Доля КСО для изучения языков программирования и инструментальных средств разработки программ значительно меньше, чем по другим предметным областям. Низкая распространенность КСО данного класса связана, во-первых, с их созданием для собственных нужд образовательными учреждениями, во-вторых, с их содержательной локальностью [2]. Однако, существует множество help-систем по технологиям разработки программного обеспечения (ПО), обеспечивающих поддержку учебного процесса. Основной целью создания help-систем является предоставление пользователю информации по интересующим его вопросам.

По способу представления информации существующие help-системы можно разделить на следующие классы: электронные технические документации, справочные гипертекстовые системы, обучающие системы.

Электронная документация представляет собой техническое описание методов и средств разработки программ в определенном формате. В настоящее время для создания таких документов применяются около десятка различных форматов, включая PDF (Portable Document Format), RTF (Rich Text Format), DOC (Document Word), а также семейство языков гипертекстовой разметки. Причем, формат PDF предназначен для кросс-платформенного представления электронных документов и является международным стандартом ISO 32000.

В настоящее время документации распространяются на внешних носителях, а также с помощью специализированных web-сайтов.

Справочные гипертекстовые системы обладают большей функциональностью в сравнении с электронными документациями. Они обеспечивают быстрый и удобный доступ к конкретной теме, термину и т.д., а также поиск информации по определенному реквизиту, обладают гибкими возможностями редактирования текста, простым и удобным выводом текста документа в файл и на печатающее устройство. Существует множество инструментальных средств создания таких систем: WinHelp, Adobe Acrobat, HelpScribble, WebHelp, FlashHelp, HTMLHelp, NetHelp, WebHelp, JavaHelp, AnetHelp Tool, Help And Manual, Mif2GO, RoboHelp и др. Кроме этого, широко распространены браузерные варианты справочных гипертекстовых систем: www.msdn.com, www.rsdn.ru, www.codeteacher.com, www.help-site.com, www.programmingtutorials.com, www.scientific-library.net и др. Например, MSDN (www.msdn.com) представляет собой информационный сервис для разработчиков программного обеспечения, содержащий описание способов и методов программирования приложений для операционных систем семейства Windows.

Обучающие системы содержат текстовое описание раздела программирования, по которому проводится обучение. Кроме этого, они содержат подсистему, определяющую направление обучения, а также могут содержать подсистему, оценивающую знания пользователя, подсистему планирования и управления порядком изучаемого материала [3], а также возможность интеграции со средой разработки программ. Среди них можно выделить: GILD, BlueJ, Greenfoot.

Для того, чтобы help-систему можно было отнести к КСО, она должна позволять решать определенные педагогические задачи, имеющие предметное содержание. В [2] выделены следующие основные педагогические задачи, которые должны решать КСО:

- начальное ознакомление с предметной областью, освоение базовых понятий и концепций;
- базовая подготовка на разных уровнях глубины и детальности;
- выработка умений и навыков решения типовых практических задач;
- выработка умений анализа и принятия решений в нестандартных проблемных ситуациях;
- развитие способностей к определенным видам деятельности;
- восстановление знаний, умений и навыков (для редко встречающихся ситуаций, задач и технологических операций);
- контроль и оценивание уровней знаний и умений.

Кроме этого, современные КСО должны помогать различным контингентам обучаемых заниматься самостоятельной познавательной деятельностью, обеспечивая индивидуализацию обучения посредством организации адаптивного диалога с пользователем. Современные системы должны содержать огромное количество знаний по различным предметным областям, уметь их обрабатывать и обучать пользователя этим знаниям, используя опыт и методику лучших преподавателей и применяя различные стратегии и механизмы обучения. [3]

Рассмотренные выше help-системы позволяют решать только некоторые из указанных задач. Анализ существующих систем показал, что основными их недостатками являются: отсутствие должного уровня интерактивности, ограниченная функциональность, отсутствие возможности управления обучением и разработкой программ, отсутствие возможности адаптации к пользователю, отсутствие унификации представления знаний. Эти недостатки не предоставляют возможности заниматься самостоятельной познавательной деятельностью, а также объединить процессы обучения и практической работы. Функциональные возможности разрабатываемой интеллектуальной help-системы охватывают вышеперечисленные задачи.

Методика проектирования интеллектуальной help-системы

Нами разрабатывается методика проектирования интеллектуальных help-систем по технологиям разработки программного обеспечения.

Разрабатываемая help-система относится к классу sc-систем [4], т.е. она представляет собой семантическую интеллектуальную систему, в основе которой лежит семантическая сеть с базовой теоретико-множественной интерпретацией и представление знаний в sc-коде. Help-система состоит из трех подсистем: справочная подсистема, подсистема мониторинга и анализа деятельности разработчика программ, подсистема управления обучением (проектированием программ). В соответствии с семантической технологией проектирования интеллектуальных систем [4] разработка каждой из подсистем состоит из следующих этапов: разработка базы знаний; разработка машины обработки знаний; разработка пользовательского интерфейса.

Для создания интеллектуальной help-системы на первом этапе необходимо определить круг задач, которые будет решать каждая ее подсистема, четко определить назначение, а также масштаб представления предметной области. Один из способов определить масштаб представления предметной области – сформировать список вопросов, на которые будет отвечать система (подсистема) [1]. Далее среди многих полученных альтернатив необходимо определить ту, которая может лучше решить поставленную задачу и будет более наглядной, а также легко расширяемой [1].

Справочная подсистема позволяет получить любую информацию о технологии проектирования программ. К функциям справочной подсистемы относятся:

- поиск информации;
- отображение информации об обрабатываемых структурах данных и языке представления знаний;
- отображение информации об описываемом языке программирования с ориентацией на уровень подготовки разработчика программ;
- анализ программных текстов пользователя и внесение предложений по улучшению их эффективности;
- генерация ответов на запросы пользователя;
- генерация программ по запросу пользователя.

База знаний справочной подсистемы содержит формальные модели языка программирования, языка представления знаний, инструментального средства, библиотеки компонентов (подпрограмм), методики разработки ПО, методики обучения разработке ПО, методики использования самой интеллектуальной help-системы, утверждения и правила, описывающие способы проектирования программных текстов и задающие семантику языка.

Машина обработки знаний подсистемы включает базовый набор операций для интеллектуальных help-систем, а также дополняется программами, обеспечивающими реализацию задач, характерных для разрабатываемого класса help-систем. Базовый набор операций включает:

- трансляция информационных конструкций, являющихся запросами пользователя, и представление их в виде семантически эквивалентных sc-конструкций;
- анализ существующих информационных конструкций базы знаний и генерация новых, являющихся ответом на запрос пользователя;
- генерация программ по постановке задачи пользователя или по постановке задачи, сгенерированной самой системой.

Т.к. предполагается интеграция разрабатываемой help-системы с инструментальной средой проектирования программ, то в машину обработки знаний включаются задачи, которые взаимодействуют с операциями, реализованными в инструментальной среде:

- верификация программ, разрабатываемых пользователем;
- анализ эффективности программ;
- оптимизация программ.

Разработка пользовательского интерфейса заключается в доработке базового интерфейса интеллектуальных help-систем дополнительными функциональными возможностями, в которых нуждаются разработчики программ.

Подсистема мониторинга и анализа деятельности разработчика программ формирует модель каждого конкретного пользователя и позволяет help-системе адаптироваться к его индивидуальным особенностям. Главной целью подсистемы мониторинга и анализа деятельности является сбор информации о разработчике. Основные классы задач, решаемых подсистемой:

- определение общей информации о разработчике программ;
- определение уровня знаний разработчика программ;
- определение и анализ ошибок;
- анализ действий.

Основой базы знаний является формальная модель разработчика программ, которая представляется в виде шаблона. Шаблон наполняется в процессе работы с системой и фиксируется для каждого конкретного пользователя. Модель включает следующие ключевые понятия: возраст, пол, место жительства, образование, уровень квалификации, степень владения языком программирования, тип характера, ошибки, мотивация, формы проведения тестирования. Некоторые ключевые понятия конкретизируются. Например, существуют формы проведения тестирования: тестово-опросная, тестирование в ходе слежения. Ошибки могут быть синтаксическими и семантическими, когнитивными и моторными. Также в базу знаний включаются утверждения, на основании которых система генерирует соответствующие заключения для формирования модели пользователя.

Основными операциями машины обработки знаний подсистемы являются:

- анализ производимых пользователем действий;
- анализ ошибок, допущенных пользователем, и автоматическое исправление простейших опечаток;
- обработка данных и построение или обновление модели пользователя;
- применение модели пользователя для достижения эффекта адаптации пользовательского интерфейса.

Пользовательский интерфейс дополняется основными адаптационными функциями. В соответствии с моделью пользователя осуществляется подбор диалоговой структуры взаимодействия между разработчиком программ и help-системой, настройка оформительских параметров help-системы, определяется список персональных команд, форма представления информации.

Подсистема управления обучением позволяет управлять действиями пользователя. Она несет организационную, обучающую и контролирующие функции, организует адаптивный диалог с пользователем. Основной задачей подсистемы является обеспечение пользователю помощи, соответствующей ситуации, и выдача рекомендаций с учетом истории его взаимодействия с help-системой. При начале работы help-системы используется общая модель пользователя, а затем подсистема управления настраивает эту модель на основании процесса из взаимодействия. База знаний содержит формальную модель управления обучением и процессом разработки программ.

Основными операциями, производимыми подсистемой, являются:

- определение стратегии обучения для минимизации времени обучения,
- переход к новой стратегии;
- генерация тестовых заданий;
- проведение тестирования и интерпретации результатов,
- решение заданий и объяснение способов решений.

Пользовательский интерфейс отслеживает ситуацию, в которой находится разработчик программ, и, используя стратегии помощи и обучения, вмешивается в соответствующий момент или тогда, когда складывается впечатление, что пользователь испытывает трудности в работе.

Последним этапом проектирования help-системы является разработка комплекса документации.

В качестве инструментария разработки help-систем используются: семейство совместимых sc-языков [5], использующих семантический способ кодирования; среда проектирования баз знаний; язык программирования SCP [6]; среда проектирования программ на языке SCP.

Интеллектуальная help-система по технологии разработки программ, ориентированных на обработку семантических сетей

Описанная методика апробируется на примере интеллектуальной help-системы по технологии разработки программ, ориентированных на обработку семантических сетей. Это обусловлено расширением областей применения интеллектуальных программных систем, что привело к широкому использованию языков программирования, ориентированных на обработку знаний. Проектирование программ на языках данного класса требует высоких профессиональных навыков у разработчиков. Поэтому актуальной является проблема обучения языкам данного класса. Кроме этого, инструментальное средство разработки программ и help-система разрабатываются в рамках одного проекта [4] и они построены с использованием одной технологии, поэтому упрощается решение проблемы интеграции.

В качестве языка программирования используется язык SCP (Semantic Code Programming) [6], ориентированный на обработку однородных семантических сетей, имеющих базовую теоретико-множественную интерпретацию. Язык программирования SCP направлен на обработку сложноструктурированных баз знаний большого объема, сложные стратегии и механизмы решения трудноформализуемых задач. Он обладает высоким уровнем интегрируемости различных моделей представления знаний, высоким уровнем параллелизма [6]. Для представления знаний используется язык SC (Semantic Code) [5]. Язык SC является открытым формальным графовым языком представления знаний, универсальным языком семантических сетей. В алфавит языка SC входят sc-узлы, sc-дуги, sc-элементы неопределенного типа. В состав sc-текстов входят элементы: sc-константы, sc-переменные, sc-метапеременные. Графовый язык SC имеет простой синтаксис, простую базовую денотационную семантику, открытый характер денотационной семантики, обусловленный открытым набором ключевых sc-узлов, и открытую операционную семантику, определяемую открытым набором операций абстрактных sc-машин. С помощью языка SC появилась возможность описания семантической структуры предметной области в наглядной форме.

В настоящее время реализована справочная подсистема по технологии разработки программ на языке программирования SCP. Основными классами вопросов, на которые отвечает данная подсистема, являются: вывод определений или пояснений ключевых понятий, вывод пояснений по выполнению действий, отображение классификаций понятий, определение принадлежности понятия к классу, выполнение аналитических действий (анализ понятий на схожесть и отличие; анализ программ на предмет синтаксической и семантической корректности, эффективности, оптимальности), поиск связи между понятиями, вывод синтаксиса построения языковых конструкций, выдача рекомендаций по выполнению определенных действий, генерация объектов (программ, фрагментов программ, алгоритмов). Типология вопросов позволяет пользователю не только получить описание понятий предметной области, но и самостоятельно изучить глубинную семантику, найти связи между понятиями, получить рекомендации по написанию программ.

База знаний справочной подсистемы включает:

- общие понятия языка программирования SCP и среды разработки с последующей их конкретизацией и описанием их теоретико-множественных свойств: разбиение на более частные понятия; включение в более общие понятия; синонимия понятия, его этимология; пересечение и объединение с другими понятиями; основные утверждения о понятии; конкретные примеры понятия;

- отношения между выделенными понятиями;
- утверждения, на основании которых система делает логический вывод: о значениях scr-переменных и scr-констант; утверждения о scr-параметрах scr-программы; о порядке исполнения scr-операторов scr-программы; об обработке структур данных; о действиях scr-параметров; об обрабатываемых структурах данных; утверждения о согласовании атрибутов и др.
- ключевые узлы, которые имеют определенную семантику и предназначены для описания help-системы.

В help-системе на языке SCP реализован набор операций машины обработки знаний перечисленный выше, который позволяет отвечать на все классы вопросов и решать задачи из предметной области.

Справочная подсистема help-системы используется студентами четвертого курса специальности «Информационные системы и технологии» при изучении дисциплины «Искусственный интеллект».

Выводы

Представленная интеллектуальная help-система позволяет решать вышеперечисленные педагогические задачи, следовательно, ее можно отнести к КСО и использовать при изучении дисциплин, связанными с технологиями программирования. Кроме этого, она представляет собой консультанта-эксперта по технологии разработки программного обеспечения, что позволяют повысить уровень интеллектуальной информационной поддержки разработчиков программ. Это обеспечивает возможность подготовленному программисту самостоятельно повышать квалификацию либо изучать новые языки программирования, инструментальные средства и технологии разработки в целом.

Наличие help-систем в составе инструментальных средств снижает начальные квалификационные требования программистов и позволяет повысить эффективность проектируемых программ.

Предложенные принципы построения можно использовать для проектирования help-систем по технологиям программирования на традиционных языках.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Ontology Development 101: A Guide to Creating Your First Ontology [электронный ресурс] / Natalia F. Noy, Deborah L. McGuinness // Stanford Knowledge Systems Laboratory Technical Report KSL-01-05 and Stanford Medical Informatics Technical Report SMI-2001-0880 – 2001. – Режим доступа:
2. Башмаков, А. И. Разработка компьютерных учебников и обучающих систем / А. И. Башмаков, И. А. Башмаков – М.: Информационно-издательский дом «Филинь», 2003.
3. Интеллектуальные обучающие системы и виртуальные учебные организации: Монография / [В.В. Голенков, В.Б. Тарасов, О.Е. Елисеева и др.]; под ред. В.В. Голенкова, В.Б. Тарасова – Мн.: БГУИР, 2001.
4. Открытая семантическая технология компонентного проектирования интеллектуальных систем [электронный ресурс] / режим доступа – <http://ostis.net/mediawiki/index.php>
5. Представление и обработка знаний в графодинамических ассоциативных машинах. Монография / [В.В. Голенков, О.Е. Елисеева, В.П. Ивашенко и др.]; под ред. В.В. Голенкова. – Мн.: БГУИР, 2001. – 412 с.
6. Программирование в ассоциативных машинах / [В.В.Голенков, Г.С.Осипов, Н.А.Гулякина и др.] – Мн.: БГУИР, 2001. – 276 с.

Рецензент: Осипова Н.В.