

УДК 681.3

В.В. ЛИТВИНОВ, В.Ю. ЖИГУЛЬСКАЯ

МОДЕЛИРОВАНИЕ В ОБУЧАЮЩИХ СИСТЕМАХ

Abstract: In the article we propose a method of creating intelligent training systems, the main aim of which is learning concepts of the subject area, including the testing of trainees. It is proposed to use for knowledge representation of concept and notation of Universal Modeling Language (UML). It is proposed the method of monitoring a trainee knowledge, which allows you to make conclusions about future training strategy.

Key words: intelligent Teaching Systems, modeling, UML, testing, knowledge control, question formalization.

Анотація: У статті пропонується методика створення інтелектуальних навчальних систем, основною метою яких є навчання поняттям предметної області, включаючи тестування учня. Для представлення знань пропонується використовувати концепції та нотації Універсальної мови моделювання (UML). Запропоновано методику контролю знань учня, яка дозволяє зробити висновки про подальшу стратегію навчання.

Ключові слова: інтелектуальні навчальні системи, моделювання, UML, тестування, контроль знань, формалізація питань.

Аннотация: В статье предлагается методика создания интеллектуальных обучающих систем, основной целью которых является обучение понятиям предметной области, включая тестирование обучаемого. Для представления знаний предлагается использовать концепции и нотации Универсального языка моделирования (UML). Предложена методика контроля знаний обучаемого, которая позволяет сделать выводы о дальнейшей стратегии обучения.

Ключевые слова: интеллектуальные обучающие системы, моделирование, UML, тестирование, контроль знаний, формализация вопросов.

1. Постановка проблемы

Для выхода на современные рубежи в культурном, социальном, экономическом, научном и техническом развитии украинскому государству необходимы разносторонне развитые, профессионально подготовленные специалисты. Поэтому все большую актуальность приобретает проблема разработки и внедрения в практику наиболее эффективных информационных технологий обучения, отвечающих общей концепции образования. Суть этой концепции состоит в подготовке личности, которая обладает, кроме достаточного для требований современного уровня производства, науки, культуры и государства объемом базовых знаний, умений и навыков, способностью к активной творческой профессиональной и общественной деятельности. Важная роль в этом отводится новым программно-технологическим средствам индивидуализации процесса обучения.

Современная стратегия образования ориентирована на международные стандарты, описанные Болонской декларацией. Это еще в большей степени влияет на развитие компьютерных технологий индивидуализации обучения.

Каждое из направлений разработки информационных технологий обучения базируется на методологических и теоретических принципах и имеет присущие им преимущества и недостатки при решении тех или других проблем в организации процесса приобретения знаний.

В методологическом плане разработка и использование компьютерных технологий обучения с самого начала развивались по двум направлениям.

Первое направление в своей основе опирается на идеи программированного обучения [1].

В его рамках разрабатываются и эксплуатируются автоматизированные обучающие системы (АОС) по различным учебным дисциплинам. Ядром АОС являются так называемые авторские системы, позволяющие преподавателю-разработчику вводить свой учебный материал в базу данных и программировать с помощью специальных авторских языков или других средств алгоритмы его изучения.

Второе направление внедрения компьютерных технологий в обучение тесно связано с процессами информатизации различных отраслей человеческой деятельности. В рамках данного направления созданы отдельные учебные компьютерные программы, пакеты программ, элементы автоматизированных систем, предназначенные для автоматизации трудоемких расчетов, оптимизации, исследования свойств объектов и процессов на математических моделях и т.п.

С начала 80-х гг. XX ст. интенсивно развивается новое направление в компьютеризации обучения – интеллектуальные обучающие системы (ИОС), основанные на работах в области искусственного интеллекта [2]. Существенной частью ИОС являются модели обучаемого, процесса обучения, предметной области, на основе которых для каждого обучаемого может строиться рациональная стратегия обучения. Базы знаний ИОС могут содержать, наряду с формализованными знаниями, экспертные знания в предметных областях и в сфере обучения. Работы в области создания ИОС, безусловно, перспективны, но находятся пока на стадии лабораторных исследований и, несмотря на некоторые примеры успешного применения [3], на уровень массовой технологии еще не вышли.

В современных обучающих системах технология автоматизированного обучения использует, как правило, классические приёмы: предоставление обучаемому порции текстово-графического материала учебных курсов и последующее тестирование обучаемого после ознакомления с учебным материалом.

Эти порции учебного материала могут определяться структуризацией учебного курса (оглавления, алфавитные терминологические указатели и т.д.) или сценарием. Последний может быть разработан заранее или формироваться на основе степени усвоенности материала.

2. Интеллектуальные системы и представление в них знаний

Среди всего разнообразия обучающего программного обеспечения гибкость обучения обеспечивает только один класс систем – ИОС. В них, как правило, используются модели знаний, которые можно отнести к следующим разновидностям:

- знания о предметной области учебного курса, рассматриваемые как эталон;
- знания о предметной области учебного курса, сформированные в представлениях обучаемого;
- знания о процессе обучения.

В ИОС знания представляются в формализованном виде.

Имеется достаточно широкий спектр способов представления знаний [4]:

- логические методы;
- семантические сети;
- фреймы;

– производственные системы.

Логические методы. Основная идея подхода при построении логических моделей представления знаний – вся информация, необходимая для решения прикладных задач, рассматривается как совокупность фактов и утверждений, которые представляются как формулы в некоторой логике. Знания отображаются совокупностью таких формул, а получение новых знаний сводится к реализации процедур логического вывода. В основе логических моделей представления знаний лежит понятие формальной теории, задаваемое кортежем: $S = \langle B, F, A, R \rangle$, где B – счетное множество базовых символов (алфавит), F – множество, называемое формулами, A – выделенное подмножество априори истинных формул (аксиом), R – конечное множество отношений между формулами, называемое правилами вывода.

Семантические сети. Модель представления знаний с помощью семантических сетей состоит из вершин, называемых узлами, соответствующих объектам, концепциям или событиям, и связывающих их дуг, описывающих отношения между рассматриваемыми объектами [5]. Дуги могут быть определены разными методами. Обычно для представления иерархии используются дуги типа «IS-A» (отношение "является") и «HAS-PART» (отношение "имеет часть"). Они также устанавливают иерархию наследования в сети, т.е. элементы более низкого уровня в сети могут наследовать свойства элементов более высокого уровня, что экономит память, поскольку информацию о наследуемых свойствах не нужно повторять в каждом узле сети.

Фреймы. Модель представления знаний с помощью фреймов предложена Марвином Минским, который описывает их следующим образом [6]: "Фрейм – это структура данных, представляющая стереотипную ситуацию вроде нахождения внутри некоторого рода жилой комнаты или сбора на вечеринку по поводу рождения ребенка. К каждому фрейму присоединяются несколько видов информации. Часть этой информации о том, как использовать фрейм. Часть о том, чего можно ожидать далее. Часть о том, что следует делать, если эти ожидания не подтвердятся". Фреймовая модель по своей организации во многом похожа на семантическую сеть. Она является сетью узлов и отношений, организованных иерархически: верхние узлы представляют общие понятия, а подчиненные им узлы – частные случаи этих понятий. В системе, основанной на фреймах, понятие в каждом узле определяется набором атрибутов-слотов и значениями этих атрибутов. Каждый слот может быть связан со специальными процедурами, которые выполняются, когда информация в слотах (значения атрибутов) меняется. С каждым слотом можно связать любое число процедур.

Производственные системы. В этой модели знания представляются в виде совокупности правил типа «Если-то». Системы обработки знаний, использующие такое представление, получили название производственных систем. В состав систем производственного типа входят база правил (знаний), рабочая память и интерпретатор правил (решатель), реализующий определенный механизм логического вывода. Факты представлены в виде пары: атрибут – значение, а в случаях, когда имеются несколько объектов с идентичными атрибутами, – в виде тройки: объект – атрибут – значение. Операционные знания представляются правилами вида условие – действие или условие – цель. Правило интерпретируется следующим образом: для достижения цели правила

необходимо достичь целей условия. При решении задачи управляющая программа выбирает цель и пытается достичь ее.

Формализованное представление знаний является крайне важным элементом ИОС, поскольку именно формализация обеспечивает успех основных преобразований, осуществляемых в системе.

В этом плане привлекательным является использование для представления знаний нотаций Универсального языка моделирования (UML). Этот язык широко используется для представления предметных областей при проектировании прикладных программных систем. В нем достаточно удобно отражаются как структурные свойства моделей, представляемые в виде классовых диаграмм, так и динамические свойства моделей, описываемые диаграммами состояний, объектными диаграммами, диаграммами последовательностей и др.

Основная идея нашего подхода – представлять модель предметной области в виде набора диаграмм классов на языке UML.

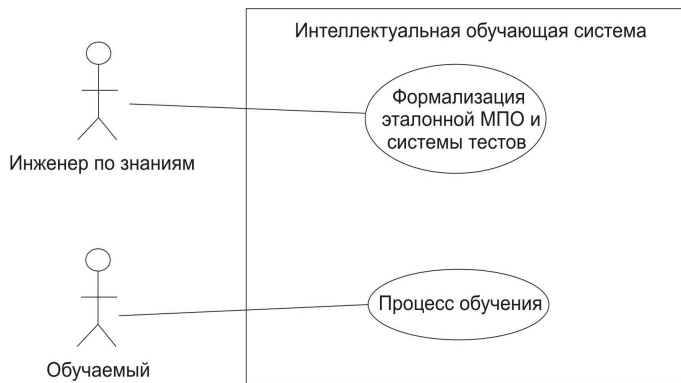


Рис. 1. Use case структура интеллектуальной обучающей системы

Классы выделяются в процессе анализа предметной области, их имена включаются в состав словаря предметной области учебного курса.

ИОС должна поддерживать работу двух категорий пользователей: инженера по знаниям, в функции которого входит составление эталонной модели предметной области (МПО), и обучаемого (рис. 1).

2.1. Use case формализации эталонной МПО и построение множества тестов

Инженер по знаниям создает эталонную МПО ($S_{мпo}^э$) на основе анализа текстово-графической информации учебного курса (Т). Этот процесс рассматривается как человеко-машинная процедура формирования $S_{мпo}^э$ и осуществляется экспертом.

В результате моделирования предметной области осуществляется:

- идентификация классов понятий или концептуальных классов компьютерной технологии обучения;
- детализация объектов предметной области;
- концептуальное представление понятий предметной области.

На языке UML МПО представляется в виде набора диаграмм классов. МПО может отображать следующее:

- объекты предметной области или концептуальные классы;
- атрибуты и операции концептуальных классов.
- отношения между концептуальными классами.

При создании МПО, как правило, используются стандартные шаги [7], представленные на рис. 2. Последовательность этих шагов зачастую задается не жестко.



Рис. 2. Этапы создания модели предметной области

При создании МПО рационально использовать приёмы и стратегии идентификации объектов и отношений, принятые в объектно-ориентированном проектировании [7].

Стратегии идентификации объектов:

1. Стратегия на базе подчеркивания существительных.

Используется для составления первичного списка объектов. Инженер по знаниям подчеркивает каждое существительное или фразу с существительным в текстовом

описании проблемы и рассматривает его как потенциальный объект.

2. Стратегия на базе идентификации причинных объектов.

Инженер по знаниям идентифицирует источники действий, событий и сообщений, включая координаторов действий.

3. Стратегия на базе идентификации сервисов (пассивных вкладчиков).

Инженер по знаниям идентифицирует цели действий, сообщений, а также сущностей, которые пассивно обеспечивают сервисы, когда они запрошены.

4. Стратегия на базе идентификации сущностей реального мира.

Инженер по знаниям идентифицирует объекты реального мира, которые являются сущностями, существуют в реальном мире, но не обязательно являются электрическими приборами.

5. Стратегия на базе идентификации физических приборов.

Инженер по знаниям идентифицирует физические приборы, которые включают сенсоры, актуаторы, предоставляемые системой, электронные приборы, которые осуществляют мониторинг или контроль.

6. Стратегия на базе идентификации ключевых концептов.

Инженер по знаниям идентифицирует ключевые концепты, которые могут моделироваться как объекты.

7. Стратегия на базе идентификации транзакций.

Инженер по знаниям идентифицирует транзакции, которые являются определенными экземплярами ассоциаций между объектами, они сохраняются в течение большого периода времени.

8. Стратегия на базе идентификации долговременной информации (запоминаемой).

Инженер по знаниям идентифицирует информацию, которая должна сохраняться в течение длительного времени, она может быть или объектом, или атрибутом. Это сохранение может быть расширено на период, когда устройство отключено от источника питания.

9. Стратегия на базе идентификации визуальных элементов.

Инженер по знаниям идентифицирует элементы пользовательского интерфейса, которые отображают данные, являются объектами в предметной области пользовательского интерфейса.

10. Стратегия на базе идентификации управленческих элементов.

Инженер по знаниям идентифицирует объекты, которые предоставляют интерфейс для пользователя или некоторого внешнего устройства для управления поведением.

11. Стратегия на базе использования идентифицированных объектов, проигрывание сценария.

Инженер по знаниям, используя идентифицированные объекты, проигрывает сценарий. Опущенные объекты становятся очевидными, когда требуемые действия не могут быть достигнуты с помощью существующих объектов.

Стратегии идентификации отношений:

1. Стратегия на базе идентификации сообщений.

Инженер по знаниям идентифицирует каждое сообщение, которое подразумевает ассоциацию между участвующими объектами.

2. Стратегия на базе идентификации источников сообщений.

Инженер по знаниям идентифицирует источники сообщений, которые посылают информацию другим объектам для обработки или запоминания.

3. Стратегия на базе идентификации хранилищ сообщений.

Инженер по знаниям идентифицирует хранилища сообщений, которые запоминают информацию для целей архивации или представляют центральный репозиторий информации для других объектов. Хранилища имеют ассоциации с источниками сообщений так же, как и с пользователями информации.

4. Стратегия на базе идентификации обработчиков сообщений.

Инженер по знаниям идентифицирует объекты, которые централизуют диспетчирование сообщений и обработку. Они формируют связи как с источниками сообщений, так и с его хранилищами сообщений.

5. Стратегия на базе идентификации структуры типа часть-целое.

Инженер по знаниям идентифицирует отношения типа часть-целое, которые становятся отношениями агрегации среди объектов. Целые часто шлют сообщения их компонентам.

6. Стратегия на базе идентификации абстрактных структур типа больше-меньше.

Инженер по знаниям идентифицирует объекты, которые относятся друг к другу на основании разницы в уровнях абстракции. В этом случае больший объект становится композитным объектом, а меньшие объекты становятся его компонентами. Отношения среди них становятся отношениями композиции.

7. Стратегия на базе использования идентифицированных объектов, проигрывание сценария.

Инженер по знаниям проигрывает сценарии, используя идентифицированные объекты. Сценарии четко показывают, как посылаются сообщения между объектами.

При построении МПО можно использовать как отдельные стратегии, так и наборы стратегий. В любом случае $S_{mno}^э$ строится на базе текстово-графического представления учебного курса.

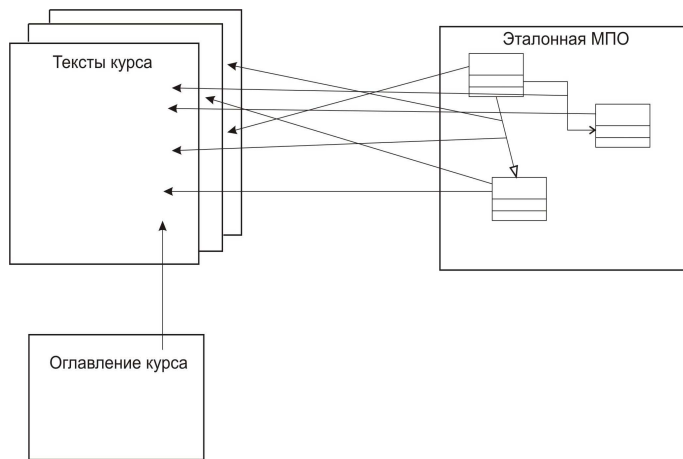


Рис. 3. Связь эталонной МПО с текстовыми описаниями курса

В процессе создания $S_{mno}^э$ должны формироваться обратные ссылки (R) от основных понятий предметной области (ПО) на тексты, содержащие описания этих понятий и их свойств (рис. 3). Ссылки могут быть и множественными. Поскольку структура учебного курса определяется оглавлением, то ссылки рационально формировать в таком виде: <раздел, параграф, страница, начало абзаца, конец текстового фрагмента>.

Наличие в $S_{mno}^э$ ссылок на тексты учебного курса позволяет выстроить гибкий процесс обучения, при котором проблемы усвоения материала отображаются в виде несоответствия $S_{mno}^э$ и МПО обучаемого ($S_{mno}^о$), и в связи с этим рассогласованием (Δ^s) формируется необходимый набор ссылок на тексты, которые обучаемый должен повторно изучить.

До сих пор рассматривались ссылки только на исходный текст учебного курса, однако на базе этих ссылок возможно сформировать более компактное текстовое описание ПО – T_1 и в $S_{mno}^э$ расставить новые ссылки на него.

Целью тестирования обучаемого является определение плохо усвоенных понятий и связей между ними.

Тест рассматривается как конструкция $P = \langle Q_i, A_i, R_i \rangle$, где Q_i – вопрос по разделу учебного курса, A_i – множество возможных вариантов ответа, один из которых верный, R_i – ссылка на раздел текстово-графического описания курса.

Вопрос понимается как совокупность постановочной части и возможных вариантов ответа – вопросно-ответных соответствий.

Набор тестовых заданий является инструментом для построения $S_{mno}^т$.

Для формализации вопроса будем пользоваться концепциями и терминологией эротетической логики согласно Н. Белнапу и Т. Стиллу [8]. Рассмотрим вопросы, состоящие из двух частей: субъекта и предпосылки. Субъект – это множество всех возможных альтернатив, из

которых может быть выбран ответ. Предпосылка определяет, какие альтернативы должны содержаться в ответе. Вопросы, субъекты которых задаются явно конечным списком альтернатив, называются ли-вопросами. Вопросы, субъекты которых предоставляют множество (возможно, бесконечное) альтернатив с помощью задания некоторого предиката, называются какой-вопросами.

Формализация вопросов.

Ли-вопрос задает конечное множество альтернатив через свои субъекты, причем это множество эксплицитно содержится в вопросе. К ли-вопросам относятся обычные да-нет-вопросы, например:

«Является ли стекло жидкостью при температуре 700 F?»

Утверждения, предоставляемые в качестве альтернатив:

«Стекло является жидкостью при температуре 700 F?» и «Стекло не является жидкостью при температуре 700 F?».

Другой пример ли-вопроса:

«Шестиугольник – это линия, многоугольник, отрезок, ломаная, кривая?»

Субъект этого вопроса состоит из пяти альтернатив, первые две из которых: «Шестиугольник – это линия»; «Шестиугольник – это многоугольник». Область значения ли-вопроса совпадает с субъектом этого вопроса.

Система обозначений:

Запись интеррогатива

$? p\delta$, где p – предпосылка, δ – субъект вопроса.

Абстрактный субъект ли-вопроса есть множество высказываний типа $\{A_1, \dots, A_n\}$, которые могут быть представлены формулами логики предикатов первого порядка как список $\{A_1, \dots, A_n\}$.

Тогда интеррогатив имеет вид

$$? p(A_1, \dots, A_n).$$

Например, для вопроса «Является ли A атрибутом класса X ?», интеррогатив имеет вид

$? p(L, \neg L)$, где L – сокращение для « A является атрибутом класса X ».

Какой-вопрос задает свои альтернативы косвенным путем с помощью вычисления некоторого предиката и проверки одного или нескольких категорных условий. Категорное условие – это условие принадлежности переменных, от которых зависит предикат, их областям определения. Например, предикат $Q(x, y)$: «класс x обладает атрибутом y » сопровождается парой категорных условий: « x есть класс», т.е. $x \in V, V$ – множество классов и « y есть атрибут», т.е. $y \in P, P$ – множество атрибутов. Вопрос может быть поставлен так: «Какие классы какими атрибутами обладают?».

Интеррогатив имеет вид

$$?(< k, k + m >)(x_1, \dots, x_n \in V // A(x_1, \dots, x_n)),$$

где V – множество классов, $x_1, \dots, x_n \in V$ – категорное условие, k – кардинальное число, определяющее нижнюю границу числа выборов, $k + m$ – кардинальное число, определяющее верхнюю границу числа выборов, $A(x_1, \dots, x_n)$ – формула логики предикатов вида

$$\forall x_1 \dots \forall x_n (P_1(x_1, \dots, x_n) \& \dots \& P_k(x_1, \dots, x_n) \& \neg P_{k+1}) \& \dots \& \neg P_{k+m}(x_1, \dots, x_n)),$$

где предикаты P_1, \dots, P_k задают список альтернатив, содержащихся в ответе, $\neg P_{k+1}, \dots, \neg P_{k+m}$ – дистракторы (свойства, которыми заданные объекты не обладают).

В предположении, что ПО описывается диаграммами классов UML, можно на основании атрибутов, операций класса и отношений между классами выделить следующие типы вопросов тестирования обучаемого по ПО:

Вопросы, составленные на базе отношений класс – атрибуты:

Вопрос:

«Является ли A атрибутом класса X ?»

Субъект имеет вид:

« A является атрибутом класса X » и « A не является атрибутом класса X ».

Вопрос:

«Какими атрибутами обладает класс X ?»

Субъект имеет вид списка $A_1, A_2, \dots, A_n, Q_1, Q_2, \dots, Q_m$, где $A_i \in A$, A – множество атрибутов, $Q_i \in Q$, Q – множество атрибутов – дистракторов.

Вопрос:

«Какие классы обладают атрибутами A_1, A_2, \dots, A_n ?»

Субъект имеет вид списка $X_1, X_2, \dots, X_n, R_1, R_2, \dots, R_m$, где $X_i \in X$, X – множество классов, $R_i \in R$, R – множество классов – дистракторов.

Вопрос:

«Указать значения атрибутов классов X_1, X_2, \dots, X_n ».

Субъект имеет вид списка $B_1, B_2, \dots, B_n, C_1, C_2, \dots, C_m$, где $B_i \in B$, B – множество значений атрибутов, $C_i \in C$, C – множество значений атрибутов – дистракторов.

Вопрос:

«Указать общие атрибуты, которыми обладают классы: X_1, X_2, \dots, X_n ».

Субъект имеет вид списка $A_1, A_2, \dots, A_n, Q_1, Q_2, \dots, Q_m$, где $A_i \in A$, A – множество атрибутов, $Q_i \in Q$, Q – множество атрибутов – дистракторов.

Вопросы, составленные на базе отношений класс – операции.

Вопрос:

«Является ли F операцией класса X ?»

Субъект имеет вид:

« F является операцией класса X » и « F не является операцией класса X ».

Вопрос:

«Какими операциями обладает класс X ?»

Субъект имеет вид списка $F_1, F_2, \dots, F_n, G_1, G_2, \dots, G_m$, где $F_i \in F, F$ – множество операций, $G_i \in G, G$ – множество операций – дистракторов.

Вопрос:

«Какие классы обладают операциями F_1, F_2, \dots, F_n ?»

Субъект имеет вид списка $X_1, X_2, \dots, X_n, R_1, R_2, \dots, R_m$, где $X_i \in X, X$ – множество классов, $R_i \in R, R$ – множество классов – дистракторов.

Вопросы, составленные на базе отношения ассоциации:

? $X_1 R_{ac}^{yo} X_2$.

Вопрос:

«Использует ли класс X методы (операции) класса Y ?»

Субъект имеет вид:

«Класс X использует методы класса Y » и «Класс X не использует методы класса Y ».

Вопросы, составленные на базе отношения агрегации:

Вопрос:

«Является ли класс X частью класса Y ?»

Субъект имеет вид:

« X является частью класса Y » и « X не является частью класса Y ».

Вопрос:

«Укажите классы, из которых состоит класс Y ».

Субъект имеет вид $X_1, X_2, \dots, X_n, R_1, R_2, \dots, R_m$, где $X_i \in X, X$ – множество классов, $R_i \in R, R$ – множество классов – дистракторов.

Вопросы, составленные на базе отношения композиции:

«Укажите классы, которыми владеет класс Y »

Субъект имеет вид $X_1, X_2, \dots, X_n, R_1, R_2, \dots, R_m$, где $X_i \in X, X$ – множество классов, $R_i \in R, R$ – множество классов – дистракторов.

Вопросы, составленные на базе отношения обобщения (наследования):

Вопрос:

«Является ли класс X экземпляром класса Y ?»

Субъект имеет вид:

« X является экземпляром класса Y » и « X не является экземпляром класса Y ».

«Объединить в группы по некоторому признаку», например, «Какие геометрические фигуры имеют длину, а какие площадь?» «ромб, окружность, отрезок, прямоугольник, ломаная, дельтоид, параллелограмм».

Вопросы, составленные на базе отношения зависимости:

«Зависит ли реализация класса X от спецификации операций класса Y ?»

Субъект имеет вид:

«Реализация класса X зависит от спецификации операций класса Y » и «Реализация класса X не зависит от спецификации операций класса Y ».

Естественно, здесь представлена структура вопроса, но не его «художественное» оформление. Последнее является предметом работы для инженера по знаниям, который осуществляет их окончательное редактирование.

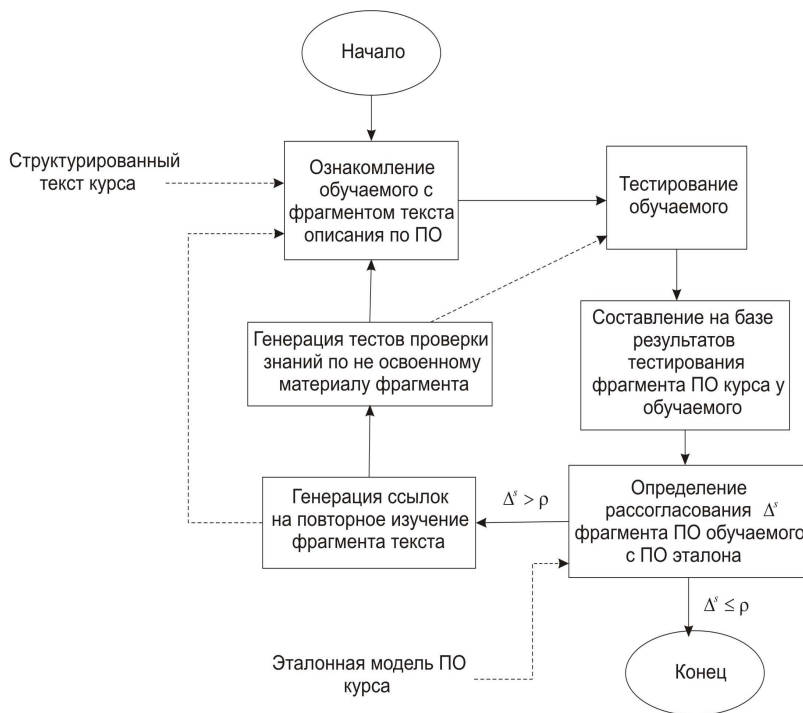


Рис. 4. Схема обучения фрагменту ПО

2.2. Use case процесса обучения

Предполагается, что учебный курс делится на ряд фрагментов, для каждого из которых процесс обучения осуществляется по следующей схеме (рис. 4).

Учащемуся предлагается ознакомиться с некоторым фрагментом текста учебного курса. С этим фрагментом, очевидно, связывается фрагмент S_{iii}^y и набор вопросов по нему, составляющих множество тестов фрагмента. При правильных ответах на

вопросы тестирования можно полностью восстановить фрагмент S_{iii}^y . После ознакомления с фрагментом текста обучаемому предоставляется набор вопросов тестирования, на которые он дает один из предложенных вариантов ответа. На основе этих ответов строится или дополняется МПО фрагмента в представлениях обучаемого. Далее на базе сравнения фрагмента S_{iii}^y и фрагмента S_{iii}^f определяется Δ^s этих предметных областей, и на базе Δ^s строится набор ссылок на повторное изучение текста учебного курса. Кроме построения ссылок, Δ^s дает возможность выделить тот набор вопросов, который необходимо задать обучаемому после повторного изучения исходного текста.

Указанные выше моменты для фрагмента учебного курса повторяются до тех пор, пока по фрагменту не будет достигнуто условие $\Delta^s \leq \rho$, где ρ – пороговое значение числа неосмысленных элементов предметной области.

Далее для учебного курса выбирается новый фрагмент.

3. Имитационное моделирование в интеллектуальных обучающих системах

Структуризация ПО должна учитывать не только структурные, но и динамические свойства МПО. Кроме набора ссылок на T или T_1 , должны быть ссылки на поведенческие аспекты.

Динамические модели поведения принято описывать:

- конечными автоматами;
- сетями Петри;
- блочными или агрегатными моделями;
- моделями систем массового обслуживания (СМО).

В таких моделях присутствует и структурная, и динамическая составляющая.

Конечные автоматы. Конечный автомат представляет собой математическую абстракцию, позволяющую описывать пути изменения состояния объекта в зависимости от его текущего состояния и входных данных при условии, что общее возможное количество состояний конечно. Если объект описывается автоматной моделью, то структурная составляющая – это объект, динамическая составляющая – поведение этого объекта.

Сети Петри. Сети Петри – сложная, хорошо развитая теория, основанная на классической теории графов и являющаяся расширением теории конечных автоматов. В ней строго определены такие понятия, как состояния, условия, переходы и т.п. Она включает также графическую нотацию (систему графических обозначений, с помощью которых можно изображать соответствующие графы).

Блочные модели. В блочном моделировании присутствует понятие блока, его поведение, преобразование вход – выход. Взаимодействие между блоками (выходы одного являются входами другого).

Агрегатные модели. Агрегатные модели родственные блочному моделированию в плане взаимодействия отдельных переменных агрегатов (блоков), однако в них явно представлены состояние агрегата, функции переходов, входов и выходов.

Модели СМО. СМО состоит из какого-то количества обслуживающих объектов, которые называются каналами обслуживания. СМО предназначена для обслуживания какого-то потока заявок (требований), поступающих в какие-то случайные моменты времени. Обслуживающие объекты СМО в совокупности образуют статические объекты, заявки – динамические объекты [9, 10].

В современных языках моделирования структурная составляющая – это компоненты, подкомпоненты и связи, поведение каждой компоненты – это динамическая составляющая.

В нашем подходе с использованием UML к описанию МПО поведение некоторых объектов описывается диаграммами состояний, а поведение группы объектов – в виде диаграмм последовательности или взаимодействия. При этом можно абстрагироваться от тех дополнительных свойств диаграмм состояний, которые направлены на сокращение записи (исключение псевдосостояний, исключение объектов, не обладающих поведением).

По этим диаграммам можно так же, как и по классовым диаграммам, задавать вопросы обучаемому.

Круг вопросов по диаграммам состояний включает:

1. Требование перечислить набор состояний. Например,

«Укажите набор состояний, которыми обладает объект O ».

2. Требование понимания особенности функций переходов. Например,

«Возможен ли переход T объекта O из состояния S_1 в состояние S_2 под действием сигнала E ?»

3. Требование указания выходных сигналов или в процессе перехода или по завершении.

Например,

«Укажите выходные сигналы, которые получаются в процессе перехода из состояния S_1 в состояние S_2 ».

Круг вопросов по диаграммам последовательности или диаграммам взаимодействия включает:

1. Требование предоставления перечня объектов различных классов, между которыми возможно взаимодействие. Например,

«Укажите объекты различных классов, которые взаимодействуют между собой (обмениваются сообщениями)».

2. Требование перечня взаимодействий между парами объектов. Например,

«Укажите сообщения, посылаемые между объектами O_1 и O_2 различных классов».

3. Требование указания порядка взаимодействий между всеми объектами. Например,

«Укажите порядок взаимодействий между объектами O_1, O_2, \dots, O_n ».

Под взаимодействием понимается вызов методов объекта (возможно, с указанием фактических параметров).

Важно отметить, что при использовании UML структурное представление статических элементов предметной области хорошо сочетается со структурным представлением динамики системы. Так, объект определенного класса может содержать другие объекты, объект определенного класса может обладать поведением, в том числе и таким, которое можно представить в виде преобразователей входной и выходной информации, ассоциативная связь между объектами может рассматриваться как «контактное» соединение агрегатов. Наконец, объекты нижнего уровня могут быть представлены как машины состояний или элементарные агрегаты.

Это традиционный подход к конструированию моделей в имитационном моделировании.

Поскольку последнее представляет собой методологию проведения экспериментов с моделями, интересно перенести эту особенность на проблемы обучения.

4. Выводы

Такая ориентация ИОС на использование МПО придает необходимую гибкость обучения, позволяет сформировать у обучаемого связанную структурированную модель изучаемого

фрагмента ПО. Последовательное выявление атрибутов и операций объектов формирует понятия, на основе сходства объекты объединяются в классы. Из набора разрозненных факторов и сведений формируется связная структура знания обучаемого, приближая процесс обучения к творческому. Так, постепенно расширяя свою базу объектов, выявляя их свойства, определяя и исследуя их поведение и взаимодействие между ними, учащийся изучает некоторую проблемную область.

СПИСОК ЛИТЕРАТУРЫ

1. Джордж Ф. Основы кибернетики: Пер. с англ. / Под ред. А.Л. Горелика. – М.: Радио и связь, 1984. – 272 с.
2. Человеческий фактор: В 6 т. / Пер. с англ. / Д. Холдинг, Н. Голдстейн, Р. Эбертс и др. – Т. 3: Моделирование деятельности, профессиональное обучение и отбор операторов. – Ч. 2: Профессиональное обучение и отбор операторов). – М.: Мир, 1991. – 302 с.
3. Растрат Л.А., Эренштеш М.Х. Адаптивное обучение с моделью обучаемого. – Рига: Зинатне, 1988. – 160 с.
4. Уэно Х., Исудзука М. Представление и использование знаний. – М.: Мир, 1989. – 220 с.
5. Скрэгг Г. Семантические сети как модели памяти // Новое в зарубежной лингвистике. – М.: Радуга, 1983. – Вып. 12. – С. 228 – 271.
6. Минский М. Фреймы для представления знаний. – М.: Мир, 1979. – 152 с.
7. Bruce Powel Douglass. Real – Time UML Second Edition. Developing Efficient Objects for Embedded Systems. – Addison-Wesley. – 1999. – 238 p.
8. Белнап Н., Стил Т. Логика вопросов и ответов. – М.: Прогресс, 1981. – 288 с.
9. Бусленко Н.П., Калашников В.В., Коваленко И.Н. Лекции по теории сложных систем. – М.: Советское радио, 1973. – 440 с.
10. Литвинов В.В., Марьянович Т.П. Методы построения имитационных систем. – К.: Наукова думка, 1991. – 120 с.

Стаття надійшла до редакції 12.05.2009