



УДК 621.372

А.Н. Афанасьев, Н.Н. Войт

РАЗРАБОТКА И ИССЛЕДОВАНИЕ МОДЕЛЕЙ АВТОМАТИЗИРОВАННОЙ ОБУЧАЮЩЕЙ СИСТЕМЫ САПР

Афанасьев Александр Николаевич, кандидат технических наук, окончил факультет информационных систем и технологий Ульяновского государственного технического университета. Доцент кафедры вычислительной техники УлГТУ. Имеет монографии, статьи, учебные пособия в области создания САПР. [Тел.: (8422) 77-81-13].

Войт Николай Николаевич, кандидат технических наук, окончил факультет информационных систем и технологий Ульяновского государственного технического университета. Сотрудник кафедры вычислительной техники УлГТУ. Имеет публикации в области создания автоматизированных средств обучения САПР. [Тел.: (8422) 77-81-13].

Аннотация

Целью данной статьи является исследование моделей, лежащих в основе организации и реализации автоматизированной обучающей системы (АОС) САПР. Рассмотрены парадигмы архитектур АОС САПР. Исследованы модели сценариев обучения. Предложена модель предметной области.

Ключевые слова: автоматизированная обучающая система, модели сценариев обучения, автоматизация проектирования.

Abstract

The aim of the article is to analyze models which are the base for organization and implementation of a computer-aided training CAD system. The article deals with architecture paradigms of the computer-aided training CAD system. It analyses scenario models and gives a model of scope.

Key words: computer-aided training system, training scenario models, design automation.

1 ПАРАДИГМЫ АРХИТЕКТУР АОС САПР

Парадигмы архитектур АОС САПР имеют такие же формы представления архитектуры, как и все автоматизированные системы. Опишем парадигмы архитектур [1].

1. Объектно-ориентированная парадигма, основанная на инкапсуляции, полиморфизме, наследовании и структуризации данных.

2. Компонентно-ориентированная парадигма, базирующаяся на сборочном принципе, в которой компонент представляет собой элемент сборки.

3. Сервисно-ориентированная парадигма, в основе которой лежит принцип массового обслуживания запросов.

1.1 Объектно-ориентированная парадигма

В основе парадигмы лежит объектно-ориентированный анализ построения архитектуры. Основные характеристики парадигмы:

1. Реализация основных принципов инкапсуляции, полиморфизма и наследования.
2. Классовая структуризация данных.
3. Взаимодействие объектов, являющихся центром интересов.
4. Реализация интерфейсов в виде классов.
5. Одинаковое использование удаленных и локальных объектов.

1.2 Компонентно-ориентированная парадигма

Сложность автоматизированных систем привела к созданию независимых, проверенных и неоднократно используемых компонентов. Основные характеристики парадигмы:

1. Компоненты, представляющие собой главный интерес.
2. Взаимодействие компонентов, реализуемых с помощью интерфейсов.
3. Взаимозаменяемость компонентов.
4. Интерфейсы, обслуживающие разные компоненты.
5. Открытость свойств и методов интерфейсной части для использования.

1.3 Сервисно-ориентированная парадигма

Система обслуживает массовые Web-ориентированные запросы с помощью сервисной парадигмы архитектуры. Основные характеристики парадигмы:

1. Доступность сервисов с любого разрешенного источника.
2. Поступление данных частями к адресату от сервиса.
3. Сервисы, представляющие собой главный интерес.
4. Отсутствие состояний сервиса, при которых обмен данными между распределенными

пользователями и провайдерами может привести к существенным накладным расходам.

2 Модели построения АОС САПР

Интегрированные САПР имеют в своем составе множество систем: подсистемы технологической подготовки производства, контроля качества, моделирования, экспертные системы, адаптивные системы, системы автоматизированного обучения и т.п.

В основу проектирования и реализации таких систем положены:

- предметно-ориентированные и проблемно-ориентированные АОС, которые имеют определенную область применения, настраиваемую структуру и обладают высокой эффективностью;

- адаптивные системы подготовки специалистов в области проектирования САПР, имеющие «дружественный интерфейс пользователя»;

- экспертные системы;

- самообучающиеся системы;

- универсальные средства программного, технического обеспечения и специализированные средства, в том числе инструментальные и сервисные.

Модели объектов проектирования разрабатываются с целью прогностической оценки и выбора эффективных проектных стратегий, что позволяет значительно сократить сроки разработки и повысить эффективность АОС САПР. Достижение цели базируется на решении следующих задач:

- разработка архитектуры АОС САПР;

- разработка моделей АОС САПР в соответствии со сформированными требованиями.

Модель обучаемого может быть представлена в виде вектора понятий, каждому понятию ставится значение «знает / не знает». Состояние уровня обучаемого инженера определяется набором значений элементов вектора. Такая модель не учитывает связи понятий друг с другом. В связи с этим была рассмотрена сетевая многоуровневая модель обучаемого в виде графа, в котором вершины отображают понятия (умения) и имеют атрибуты: номер понятия (умения), «знает/не знает» («умеет/не умеет») t_i и вес понятия (умения). Уровень обучаемого вычисляется как произведение веса понятия (умения) на t_i . Дуги указывают на связи понятий (умений): «часть-целое» (показывает, что понятие (умение) дочерней вершины является составной частью родительской вершины), «ассоциация» (знание (умение) родительской вершины требует знания (умения) дочерней вершины), «слабая связь» (знание (умение) родительской вершины не требует знания (умения) дочерней вершины).

Модель обучения определяет порядок прохождения учебных элементов в зависимости

от состояния характеристик модели обучаемого, требуемых эталонных значений характеристик обучаемого и некоторых воздействий. Под воздействиями понимается представление обучаемому конкретного учебного материала теоретического или практического характера в различном виде. В зависимости от разницы характеристик обучаемого и эталонных значений происходит эталонное воздействие.

Модель объяснения необходима для расширения текстовых возможностей представления материала элементами мультимедиа-технологий.

3 Анализ моделей сценария обучения

В данном разделе детально рассматриваются способы представления порядка процесса обучения.

3.1 Одноресурсная сеть Петри как модель сценария [2]

Модель этого процесса в виде сети Петри содержит два множества узлов: множество позиций $P = \{p_1, \dots, p_7, p_M, p_D, p_T, p_J\}$ и множество переходов $T = \{t_1, \dots, t_8\}$. Узлы соединены дугами двух видов: от позиций к переходам и от переходов к позициям.

Маркировка позиций моделирует выполнение условий, а переходы при своем срабатывании — наступление событий. На первом этапе моделирования не дифференцируются виды ресурсов, используемых в системе, они ограничиваются только одним видом целочисленного типа, которому соответствует единственное цветовое множество $Color\ INT = integer$ и соответствующая переменная $var\ s:INT$.

Смысл введенного ресурса следующий: если в позиции p_i имеется хотя бы одна фишка (т.е. маркировка $m_i = k's, k > 0$), то срабатывание выходного перехода возможно. Описанная сеть называется обыкновенной сетью Петри, с изучения таких сетей началась разработка этой теории.

Условия, моделируемые позициями:

p_1 — изучение модуля возможно;

p_2 — основной материал модуля выбран;

p_3 — выбор теста возможен;

p_4 — тест выбран;

p_5 — оценивание ответа произведено;

p_6 — дополнительный материал модуля выбран;

p_7 — переход к следующему модулю возможен;

p_M — база основных учебных модулей;

p_D — база дополнительных материалов;

p_T — база тестовых материалов;

p_J — журнал учета пройденных модулей.

События, моделируемые переходами:

t_1 — изучение основного материала модуля начинается;

t_2 — изучение основного материала модуля завершается;

t_3 — тестирование начинается;

t_4 — тестирование завершается;

t_5 — изучение дополнительного материала начинается;

t_6 — изучение дополнительного материала завершается;

t_7 — повторное изучение модуля начинается;

t_8 — изучение модуля завершается.

Начальная маркировка позиций выглядит следующим образом:

$$m_1 = I's, mM = N's, ms = K's, mT = L's.$$

Здесь M — количество модулей в курсе, N — количество дополнительных разделов, L — количество тестов. Предполагается, что $M < N < L$.

Все остальные позиции в начальный момент не содержат ресурсов, т.е. имеют нулевую маркировку. В соответствии с правилами функционирования сети Петри на первом шаге может сработать переход t_1 (что соответствует событию: изучение основного материала модуля начинается). При этом будет изъято по одной фишке из позиций p_M и p_1 и одна фишка будет помещена в позицию p_2 . Выполняется условие, дающее возможность сработать переходу t_2 . Описанный процесс продолжается, аналогичным образом сработают переходы t_2, t_3 и t_4 . После выполнения условия p_3 возможно разветвление процесса по трем направлениям, т.е. может произойти одно из трех описанных выше событий, которые моделируются переходами t_3, t_7 и t_8 . При срабатывании перехода t_8 и попадании фишки $I's$ в позицию p_7 моделируемый процесс завершается. Перемещение фишки $I's$ в позицию p_7 соответствует фиксации этого факта в журнале учета АОС.

Приведенная выше одноресурсная модель не отражает ряда существенных особенностей процесса прохождения курса:

- образовательные ресурсы не идентифицированы, т.е. не учитываются номера и порядок прохождения учебных модулей, привязка дополнительного материала и тестов к номерам модулей;

- не учитываются время, затрачиваемое на обучение, доучивание и тестирование; а также вероятностный характер этого процесса.

3.2 KFS-граф как модель сценария обучения [3]

Вводится конечное множество

$$E = \{e_p, \dots, e_p, \dots, e_n\},$$

где e_i — обучающий блок (e_i -блок), соответствующий порции учебного материала, и пара

отношений на E , которые являются отображениями:

1. $e = a(e_p, e_2, \dots, e_n)$ - отношение непосредственной связности по информации (выводимости) блока e из блоков $\{e_p, \dots, e_p, \dots, e_n\}$.

2. $e = b(e_p, e_2, \dots, e_n)$ — отношение детализации знания, которое состоит из знаний $E = \{e_p, \dots, e_p, \dots, e_n\}$.

Структурной моделью учебного материала называется тройка $\langle E, a, b \rangle$, где E — множество учебных блоков, a — отношение информационной связности, b — отношение детализации. Свойства отношений информационной связности таковы, что e -блоки образуют сцепления, которые имеют начальные блоки $\{e_0\}$ и конечные (целевые) блоки $\{e_k\}$.

Образно говоря, знания имеют источники — промежуточные (выводимые) и конечные (целевые) знания, связанные сетью передачи потоков знаний от источников к целевым обучающим блокам, поэтому модель знаний $\langle E, a, b \rangle$ называется потоковой структурой знаний (Knowledge Flow Structure — KFS).

Обучающие e -блоки связываются в сеть KN (Knowledge Net) следующим образом. Каждой вершине KN сопоставляется единственный e -блок. Каждой дуге KN соотносится маркер, который является кодом формулы (описания) соответствующего знания, заключенного в учебный блок. Далее маркеры обозначаются большими буквами латинского алфавита. Исходя из того, что e -блок определяет отображение, введем понятие формулы вывода:

$$e(A_p, A_2, \dots, A_p, \dots, A_n) \rightarrow B,$$

где $(A_p, A_2, \dots, A_p, \dots, A_n)$ — входные, поставляемые в блок e знания;

B — выходные (целевые) знания, полученные в результате процедуры обучения (вывода);

« \rightarrow » обозначает некоммутативную операцию «следует».

Формула читается так: знание B является следствием процесса изучения знаний $A_1VA_2V, \dots, VA_iV, \dots, VA_n$.

Граф KN обладает следующими постулируемыми свойствами: асимметричность, ацикличность. Граф KN конечен. Имеет множество входных вершин (типа e_0) и единственную выходную вершину (типа e_k). Вывод целевых знаний реализуется системой формул вывода для каждой из его вершин, исключая входные. Граф показывает, из каких составляющих и как складывается целевое знание. Он закладывает основу методики построения учебного материала, диалектического единства группировки и выделения, обобщения и дифференциации знаний.

На графе KN логически выделяются завершённые подмножества — кластеры. Обучающим

кластером называется направленный граф KN , вершины которого размечены е-блоками, дуги B — маркерами знаний, каждой вершине сопоставляется формула вывода, и каждая вершина (е-блок) кластера выводима из начальных знаний либо является начальным знанием (типа e_0).

Таким образом, кластер обладает свойством полной выводимости. Свойство кластерности или полной выводимости является необходимым свойством активного электронного учебника. При отсутствии полной выводимости нельзя построить процесс контроля знаний и обеспечить управление процессом обучения. Процесс вывода определяется деревом вывода, которое строится по логическим формулам.

3.3 Модель сценария на основе семантической сети [4]

Рассмотрим иерархию знаний S учебного процесса. Анализ различных стратегий организации учебного процесса позволяет выделить следующие основные уровни (слои) такой модели:

$$S = (\{AR\}, \{AU\}, SO),$$

где AR — агрегат-раздел, т.е. часть семантической сети, соответствующая разделу дисциплины;

$\{AR\}$ — множество агрегатов-разделов;

AU — агрегат-урок (или агрегат-занятие), часть семантической сети, соответствующая отдельному логически законченному уроку (или занятию);

$\{AU\}$ — множество агрегатов-уроков;

SO — модель (семантическая сеть) основного учебного материала, она образует третий (основной или базовый) слой семантической сети.

SAR — слой иерархии семантической сети, отражающий представление учебного материала на уровне разделов:

$$SAR = (\{AR\}, CLR, H(SAR)),$$

где CLR — отношение следования разделов;

$H(SAR)$ — множество обобщенных характеристик модели материала на уровне разделов (главным образом, оценки требуемого времени для изучения раздела, а также оценки относительной сложности материала раздела для восприятия обучаемым).

SAU — слой иерархии семантической сети, отражающий представление учебного материала на уровне уроков (занятий):

$$SAU = (\{AU\}, CLU, H(SAU)),$$

где CLU — отношение следования уроков, отношение квазипорядка на множестве $\{AU\}$;

$H(SAU)$ — множество обобщенных характеристик модели учебного материала на уровне уроков.

IM — модель иллюстрированного учебного

материала.

VM — модель вспомогательного учебного материала.

SZ — модель учебного материала, используемого для закрепления знаний, полученных на уроке (занятии).

SP — модель учебного материала, используемого для повторения знаний, ранее полученных на занятиях.

$(SK, RK < KR)$ — блок, представляющий базу знаний для контроля учебного процесса,

где SK — модель материала для самоконтроля;

RK — модель материала для рубежных контролей;

KR — модель материала для контрольных работ.

Каждый слой модели S соответствует (обеспечивает решение) конкретным задачам информатизации процесса обучения. Так слой SAU позволяет построить общий план преподавания дисциплины и рассчитать его характеристики (показатели).

Основным компонентом SO являются объекты. Их множество обозначим через

$$O = \{O_1, O_2, \dots, O_p, \dots, O_n\}.$$

Объекты имеют имена, описания, атрибуты или свойства, а также отношения с другими объектами:

$$(WI, JI, CI, \Pi),$$

где WI — имя (наименование) объекта OI ;

JI — описание объекта OI ;

CI — свойства (атрибуты) объекта, отличающие его от других объектов предметной области и характеризующие его индивидуальность и значимость в фундаментальном и прикладном смыслах:

$$CI = (HI, BI, KI(B)),$$

где HI — множество индивидуальных атрибутов OI ;

BI — нечеткая (размытая) оценка объема учебного материала, изложенного в тексте (описании) JI , в часах;

$KI(B)$ — нечеткая (размытая) оценка сложности JI для обучающихся различных категорий из B .

Выражение $BI = f(JI, KI(B))$ задается, как правило, экспертным путем.

Π — множество отношений объекта OI с другими объектами предметной области.

$\Pi = O \rightarrow O$ является отображением множества объектов в самого себя. Так как множество отображений неоднородно, введем кортеж типов отношений:

$$T = (t1, t2, \dots, tr),$$

где r — число типов отношений в модели SO .

Каждое t_i имеет индивидуальные свойства $V(t_i)$. Кортеж свойств отношений:

$$V = (V(t_1), V(t_2), \dots, V(t_r)).$$

Тогда множество отношений, заданное на множестве объектов в SO:

$$\Gamma = \{\Gamma(t_1), \Gamma(t_2), \dots, \Gamma(t_j), \dots, \Gamma(t_r)\},$$

где $\Gamma(t_r)$ – множество отношений типа t_j на O :

$$\Gamma(t_j) = O \rightarrow O.$$

В модели SO используются следующие отношения множеств:

t_1, t_2 – родовидовые (быть родом, быть видом);

t_3, t_4 – часть-целое;

t_5, t_6 – причинно-следственные;

t_7, t_8 – процесс-подпроцесс;

t_9 – отношение квазипорядка;

t_{10} – отношение строгого порядка;

t_{11} – отношение эквивалентности;

t_{12} – отношение ассоциации.

Наиболее активно используются алгебраические свойства отношений: рефлексивность, нерефлексивность, антирефлексивность, симметричность, несимметричность, транзитивность, нетранзитивность и антитранзитивность. В зависимости от комбинации этих свойств классифицируются типы семантических связей, т.е. отношения понятий в понятийной системе предметной области.

3.4 Орграфовая модель сценария обучения [5]

В настоящее время при объектно-ориентированном подходе к разработке компьютерных систем обучения широко используется термин «объект изучения», но однозначного определения этого термина пока не существует. Некоторые авторы под объектом изучения (ОИ) понимают кванты учебной информации (УИ), другие рассматривают ОИ на двух уровнях: макроуровне (уровень тем) и микроуровне (уровень квантов УИ, хотя учебный курс также можно считать объектом изучения).

Построение оптимальной последовательности объектов изучения осуществляется на основе модели учебного материала, в качестве которой используется ориентированный граф с нагруженными ребрами $G(V, S)$. Множество V вершин графа соответствует объектам изучения (курсам, темам, разделам тем или квантам УИ), а множество S ребер – связям между ними. При этом возможны следующие степени связи:

s_1 – для изучения объекта необходимо иметь общее понятие о другом ОИ;

s_2 – при изучении объекта используются частые ссылки на другой ОИ;

s_3 – для изучения наиболее сложных (или редко используемых) понятий объекта необходимы знания из другого ОИ;

s_4 – для изучения объекта и практического

применения знаний необходимо четкое знание другого ОИ.

Каждой вершине i графа ставится в соответствие вектор $V_s = \{R_{i1}, R_{i2}, \dots, R_{im}\}$, каждый элемент которого включает пять параметров, т.е.

$$R_{ij} = (p_{ij}, t_{ij}, z_{ij}, u_{ij}, q_{ij}),$$

где i – номер ОИ;

j – номер программы (специальности) обучаемого инженера;

p_{ij} – программа (специальность) обучаемого инженера;

t_{ij} – время изучения ОИ;

z_{ij} – уровень знаний, который должен быть достигнут при изучении ОИ;

u_{ij} – уровень умений;

q_{ij} – уровень навыков, который должен быть сформирован в результате изучения ОИ.

Последние три параметра отражают цель обучения, тогда подграф G_k графа G , включающий вершины с одинаковыми значениями p_{ij} , является моделью учебного материала (УМ) для инженеров, обучающихся по специальности p_k . Такой подход позволяет осуществить адаптацию к разным обучаемым инженерам в зависимости от их специальности и/или программы обучения.

После построения моделей УМ можно определить оптимальную последовательность изучения ОИ, используя, например, математический метод, основанный на минимизации линейной функции забываемости. В результате получим сценарии обучения (последовательность ОИ) для всех обучаемых инженеров. При этом также определяются ОИ, которые могут быть изучены в произвольной последовательности, обеспечивая тем самым некоторую адаптируемость компьютерных систем обучения (КСО), т.к. в процессе обучения инженеру предоставляется возможность выбора ОИ из предлагаемого списка.

Аналогичным образом можно определить последовательность контрольных вопросов и задач, которые должны быть выполнены для достижения требуемого уровня знаний и умений.

Описанную модель можно использовать на всех уровнях УМ, т.е. определить сценарии и управлять последовательностью изучения курсов, тем, разделов и квантов-понятий УИ. Но при неизменной модели УМ, что в целом характерно для учебных дисциплин, полученные на ее основе сценарии являются постоянными. Поэтому их можно поместить в базу знаний компьютерной системы обучения и использовать лишь для управления последовательностью изучения курсов, тем, разделов и квантов-понятий. В случае включения нового курса, темы, раздела видоизменяется модель УМ и определяется последовательность обучения – новый сценарий.

3.5 Информационно-логическая модель учебного материала [6]

Для описания порядка предъявления структурных единиц учебного материала используются отношения, определяемые на множествах соподчиненных структурных единиц. Под информационно-логической моделью понимается спецификация состава этих отношений, правил их установления и образуемых ими типовых структур, которые затем применяются в качестве шаблонов при подготовке учебного материала и формировании информационных компонентов компьютерного учебника. Информационно-логическая модель, описанная в работе [6], представляется в виде ориентированного мультиграфа, вершины которого соответствуют структурным единицам учебного материала, а ребра — отношениям рассматриваемых классов. Ориентация ребер отражает направления переходов между вершинами, т.е. порядок навигации по учебному материалу в рамках множества соподчиненных структурных единиц.

Отношения подразделяются на семь типов, обозначаемых прописными латинскими буквами n, p, l, f, k, a, u . Соответствующие пометки имеют ребра мультиграфа. Из каждой вершины может исходить не более одного ребра типов n, p, l, u и произвольное число ребер типов k, a .

Ребро типа n связывает предыдущую вершину с последующей, а ребро типа p — последующую вершину с предыдущей. Переходы по вершинам, задаваемые ориентацией n -ребер, соответствуют навигации по учебному материалу в прямом направлении. И наоборот, переходы по p -ребрам соответствуют навигации в обратном направлении.

Ребра типа l указывают на последнюю, а ребра типа f — на первую вершину просмотровой последовательности.

Ребра типа k специфицируют переходы к опорным вершинам просмотровой последовательности или множество соподчиненных вершин. Опорные вершины отражают тематическую декомпозицию просмотровой последовательности на звенья без выделения дополнительных уровней иерархии. Поскольку звенья остаются связанными n и p -ребрами, просмотровая последовательность сохраняется в целостном качестве. Навигацию по учебному материалу упрощают k -ребра. Они позволяют быстро перейти к началу фрагмента просмотровой последовательности, содержащего интересующую пользователя информацию, без длительного листания кадров или страниц. Очевидно, что реализация таких отношений актуальна для просмотровых последовательностей значительной длины либо при большом объеме множества соподчиненных вершин.

Для организации ветвлений служат ребра типа a . В точке ветвления n -ребро задает базовый путь навигации (траекторию по умолчанию), а ребра типа a — альтернативные (дополнительные) пути. В случае, когда принципиально важен выбор одной из альтернатив, n -ребро, исходящее из вершины, в которой имеется ветвление, не определяется, в результате чего исходная просмотровая последовательность разбивается на две. Попав в такую вершину, нельзя продолжить навигацию в прямом направлении (допускается только возврат назад по p -ребру), не выбрав один из предложенных путей.

При наличии в множестве соподчиненных вершин элементов, декомпозируемых на нижележащем уровне иерархии, как правило, одна из первых вершин базовой просмотровой последовательности, реализуемая как опорная, содержит перечень декомпозируемых элементов и средства доступа к ним. Эта вершина связана с декомпозируемыми вершинами ребрами типа a .

Ребра типа u определяют пути возврата из просмотровых последовательностей. Как правило, для всех вершин последовательности они указывают на одну и ту же вершину, содержащую материал агрегирующего или обобщенного характера. Если переход к просмотровой последовательности возможен из единственной вершины, то u -ребра ссылаются на эту вершину.

Для связи с уровнем иерархии, на котором расположена подчиняющая вершина, служит фиктивная вершина, выступающая в качестве представителя последней для соподчиненных вершин. Из фиктивной вершины может исходить по одному ребру типов n, p, l и f . Входить в такую вершину может произвольное число ребер типов n, p, l и u .

3.6 Древоподобная графовая и орграфовая модели учебного материала [7]

Совокупность учебных материалов в работе [7] представляют в виде структурной схемы — древоподобного графа, который называют графом содержания учебного материала и строят по иерархическому принципу. Узлами (вершинами) графа являются учебные элементы (УЭ), ребрами — иерархические связи между ними. При построении графа соблюдают правила построения иерархических древоподобных структур:

- граф имеет только один корень, один УЭ — название темы;
- отсутствуют отдельные (висячие) вершины, не связанные с вышестоящими УЭ, кроме корня;
- связь осуществляется только сверху вниз;
- нижестоящий УЭ может быть связан только с одним вышестоящим УЭ;
- группировка УЭ на одном уровне осуществ-

вляется по какому-либо общему признаку (общему основанию);

- вышестоящие УЭ не должны быть связаны менее чем с двумя нижестоящими УЭ.

Параллельно с построением графа составляют таблицу УЭ, в которую вносят наименования УЭ.

После структурирования и отбора содержания учебного материала формулируют требования по уровню представления, уровню усвоения, степени автоматизации, уровню осознанности. При этом формируется таблица учебных материалов, в которой по каждому показателю знаний, умений и навыков заполняют две колонки. В первой колонке указывают «стартовый» показатель, который предположительно был получен в результате предшествующего обучения по другим темам курсов. Во второй колонке указывается «финишный» показатель, который должен быть достигнут в результате обучения по разрабатываемой теме. Учебный элемент вносят в таблицу и, следовательно, планируют его изучение лишь когда необходимо повысить хотя бы один из показателей. Таким образом устанавливают четкую преемственность и взаимосвязь различных учебных дисциплин или отдельных тем в одной учебной дисциплине.

Будем называть совокупность графа содержания и таблицы учебных элементов моделью содержания учебного материала темы. Такая модель позволяет:

- четко определить содержание учебного материала и цели обучения;

- представить содержание в наглядном и обзорном виде;

- привлечь экспертов для обсуждения полноты содержания и целевых показателей уже на начальной стадии проектирования;

- обеспечить четкую преемственность учебных дисциплин;

- перейти к машинным формам представления модели содержания;

- определить состав учебного комплекса;

- сформировать системное (целостное) представление содержания учебного материала как у разработчиков, так и у пользователей комплекса;

- сформировать требования к типу, количеству и последовательности упражнений для осмысления и закрепления теоретического материала.

Модель содержания учебного материала не дает ответов на вопросы, в какой последовательности должны изучаться УЭ и каковы логические связи между ними. Эти вопросы рассматриваются при формировании модели освоения учебного материала.

В состав модели освоения входят матрицы отношений очередности и логических связей УЭ, последовательность изучения УЭ, граф логических связей УЭ. Построение модели производят в четыре этапа:

1. Формирование матрицы отношений очередности УЭ.

2. Обработка матрицы отношений очередности и построение последовательности учебного материала в виде списка УЭ.

3. Формирование матрицы логических связей УЭ.

4. Построение графа логических связей УЭ.

Первый и третий этапы являются неформальными и выполняются на основе учебного материала. Матрицы отношений очередности и логических связей УЭ являются квадратными. Размер матриц равен количеству УЭ. Сначала строят ячейки матриц и нумеруют их строки и столбцы в соответствии с возрастанием УЭ. Далее построчно заполняют ячейки матриц нулями и единицами.

При заполнении ячеек матрицы отношений очередности анализируют простое бинарное отношение очередности между двумя УЭ. Единицу ставят в ячейку, если УЭ, указанный в номере строки, должен изучаться после УЭ, указанного в номере столбца. Противоположное отношение очередности обозначают нулем или оставляют соответствующую ячейку матрицы пустой. Все ячейки главной диагонали матрицы отношений очередности заполняют единицами. Ячейки матрицы, симметричные относительно главной диагонали, должны иметь противоположные отношения (0 или 1). Поэтому неформальный анализ парных отношений очередности можно проводить лишь для нижнего или правого верхнего треугольников матрицы, заполняя ее оставшуюся часть формально на основе свойства антисимметрии.

При заполнении матрицы логических связей УЭ ставят единицу в ячейку, если учебный материал УЭ, указанного в номере строки, логически связан с учебным материалом УЭ, указанного в номере столбца. Сопоставление матрицы логических связей удобно вести на основе матрицы отношений очередности путем исключения единиц из тех ячеек, для которых отсутствуют логические, опорные связи между элементами.

Последовательность изучения УЭ в пошаговой процедуре обучения определяют в процессе формальной обработки матрицы отношений очередности, суммируя коэффициенты каждой строки матрицы. Полученные суммы записывают в колонке справа от матрицы. Величины сумм указывают на порядковые номера соответствующих УЭ в списке последовательности изучения учебного материала.

Логические связи УЭ отображают для наглядности в виде ориентированного графа. Строят граф по матрице логических связей УЭ, которая является для него транспонированной матрицей смежности. Целесообразно располагать этот граф под списком последовательности УЭ, сохраняя указанный в списке порядок освоения учебного материала.

4 РАЗРАБОТКА МОДЕЛИ ПРЕДМЕТНОЙ ОБЛАСТИ

Под моделью будем понимать множество, состоящее из структурных компонентов, которые связаны друг с другом и взаимодействуют между собой.

Предметная область САПР в соответствии с системным подходом имеет иерархическую структуру объектов проектирования и упорядоченную последовательность этапов проектирования.

Модель предметной области имеет вид:

$$Model = \{Phase, Problem, Procedure, Pattern, MetaData \mid ortree, <, view\},$$

где $Phase = \{этан, i \in [1, E]\}$ – множество названий этапов проектирования;

$Problem = \{задача, i \in [1, Z]\}$ – множество проектных задач;

$Procedure = \{процедура, i \in [1, P]\}$ – множество проектных процедур;

$Pattern = \{Operation, Command, Sposob\}$ – множество проектных шаблонов;

$Operation = \{операция, i \in [1, O]\}$ – множество проектных операций;

$Command = \{команда, i \in [1, O]\}$ – множество проектных команд;

$Sposob = \{способ, i \in [1, S]\}$ – множество проектных способов выполнения команды;

$Atom = \{понятие, действие, i \in [1, A]\}$ – множество «атомов» знаний, состоящее из элементарных понятий и простейших действий;

$Atom \in Phase, Atom \in Procedure, Atom \in Operation, Atom \in Command, Atom \in Sposob;$

$MetaData = \{< key_i >, hash-function, i \in [1, H]\}$ – метаданные модели, где $< key_i >$ – кортеж ассоциативных ключей;

$hash-function$ – хэш-функция поиска элемента;

$ortree$ – иерархическое отношение;

$<$ – отношение порядка;

$view$ – ассоциативная функция.

Структура паттернов $PatternOperation$ проектных операций, $PatternCommand$ проектных команд, $PatternSposob$ проектных способов одинакова.

Например, структура $PatternOperation$ имеет вид:

$$PatternOperation = \{class, naznach, motiv, primenim, struct, actor, relation, result, release, example, used, rpattern\},$$

где $class$ – название;

$naznach$ – назначение, лаконично характеризует функции и дает обоснование, какие конкретные задачи проектирования можно решить с помощью паттерна;

$motiv$ – мотивация, помогает понять абстрактные описания паттерна;

$primenim$ – применимость, описывает ситуацию, в которой можно применить паттерн, и распознавание таких ситуаций;

$struct$ – структура, представляется графическими диаграммами классов программы с использованием нотации, основанной на методике Object Modeling Technique, и диаграммами взаимодействия;

$actor$ – участники, представляются классами и объектами, задействованными в данном паттерне проектирования, и их функциями;

$relation$ – отношения, описывают взаимодействие участников для выполнения своих функций;

$result$ – результаты, состоят из результатов применения, компромиссов, на которые приходится идти;

$release$ – реализация, описывает сложность при реализации паттерна;

$example$ – пример, представляется фрагментом кода, схемы, сборки и т.п.;

$used$ – известные применения в реальных системах;

$rpattern$ – родственные паттерны, имеют описание связи, важных различий, композиции других паттернов с данными.

Адаптивность модели под разные промышленные области САПР выполняется с помощью схем сводимости, которые представлены операционными базами расширения и ограничения.

СПИСОК ЛИТЕРАТУРЫ

1. Соснин П.И. Архитектурное моделирование автоматизированных систем. – Ульяновск: УлГТУ, 2007. – 146 с.

2. Доррер Г.А., Рудакова Г.М. Моделирование процесса интерактивного обучения на базе формализмов раскрашенных сетей Петри // Вестник КрасГУ. – 2004. – № 3. – С. 41–48.

3. Курганская Г.С. Модель представления знаний и система дифференцированного обучения через Интернет на ее основе // Известия Челябинского Научного Центра. – 2000. – Вып. 2. – С. 26–34.

4. Башмаков И.А., Рабинович П.Д. Анализ моделей семантических сетей как математического аппарата представления знаний об учебном материале // Справочник. Инженерный журнал. – 2002. – № 7. – С. 55–60.

5. Зайцева Л.В., Новицкий Л.П., Грибкова В.А. Разработка и применение автоматизированных обучающих систем на базе ЭВМ. – Рига: Зинатне, 1989. – 174 с.

6. Башмаков А.И., Башмаков И.А. Разработка компьютерных учебников и обучающих систем. – М.: Информационно-издательский дом, 2004. – 235 с.

7. Соловов А.В. Проектирование компьютерных систем учебного назначения. – Самара: СГАУ, 1995. – 138 с.