

ИНТЕЛЛЕКТУАЛЬНАЯ ОБУЧАЮЩАЯ СИСТЕМА КОНЦЕПТУАЛЬНОМУ ПРОЕКТИРОВАНИЮ АВТОМАТИЗИРОВАННЫХ СИСТЕМ

© 2010 А.Н.Афанасьев, Н.Н.Войт

Ульяновский государственный технический университет

Поступила в редакцию 14.05.2010

В статье рассмотрены архитектура, модели и реализация интеллектуальной обучающей системы концептуального проектирования сложных автоматизированных систем, активно использующих программное обеспечение.

Ключевые слова: интеллектуальная обучающая система, концептуальное проектирование, автоматизированные системы.

Вклад в теорию интеллектуальных автоматизированных обучающих систем внесли российские ученые Рыбина Г.В., Стефанюк В.Л., Тарасов В.Б., Петрушин В.А., Брусиловский П.Л., Башмаков А.И., Башмаков И.А. и др., а также зарубежные Kabassi K., Tang T.Y., Dorca F.A., Giannotti E., Rada R. и др. Выделяется несколько основных парадигм организации и реализации систем данного класса:

1. Основанная на концепции специализированных экспертных систем.
2. Основанная на гипертексте и гипермедиа.
3. Основанная на интеграции экспертных систем и гипертекста/гипермедиа.
4. Использующая концепцию интеллектуальных обучающих инструментов, представляющих собой системы со смешанной инициативой и оверлейным типом модели обучаемого.
5. На основе интеграции экспертных систем с системами обучения.

Однако эти подходы слабо учитывают оценку знаний обучаемых, носящих нечеткий характер, не обеспечивают эффективную адаптацию обучаемого к учебно-практическому наполнению в ходе процесса обучения в условиях различной степени первичной подготовки обучаемых, ориентированны на узкую предметную область, что снижает степень их универсальности. Таким образом, для устранения указанных недостатков необходим новый подход к организации и реализации интеллектуальных адаптивных автоматизированных обучающих систем, включающий комплекс моделей и методов, повышающий эффективность и качество обучения и позволяющий сократить сроки обучения.

Афанасьев Александр Николаевич, кандидат технических наук, профессор кафедры «Вычислительная техника». E-mail: a.afanasev@ulstu.ru.

Войт Николай Николаевич, кандидат технических наук, доцент кафедры «Вычислительная техника». E-mail: n.voit@ulstu.ru

На основе модельного подхода, к построению архитектуры интеллектуальной обучающей системы, разработаны следующие основные компоненты: предметная область проектирования, обучаемый проектировщик, сценарий процесса обучения. С целью обеспечения доступности и автономности компонентов, выбрана Сервисно-компонентная организация системы обучения. Интернет-ориентированные сервисные службы позволяют поддерживать доступ к системе в любое время и с любого подключенного к сети Интернет клиентского компьютера. Интерфейсное взаимодействие компонентов обеспечивает возможность замены их на другие компоненты без перекомпилирования всей системы. Взаимодействие компонентов архитектуры показано на рис. 1.

Математическое описание моделей в предметной области автоматизированного проектирования представлена в виде дерева онтологий,



Рис. 1. Сервисно-компонентная архитектура интеллектуальной обучающей системы

которая динамически использует иерархические, порядковые и ассоциативные связи онтологий объектов и процессов проектирования. Каждой онтологии соответствует учебный элемент.

Иерархические связи используются для описания объекта и процесса проектирования с разной степенью детализации. Порядковые связи упорядочивают описание на одном иерархическом уровне и определяют цепочки онтологий. Ассоциативные связи соединяют иерархические и порядковые онтологии разных уровней.

Таким образом, модель предметной области позволяет адекватно представить учебный материал и является базой знаний промышленного проектирования.

Модель предметной области имеет вид $CADModel = \{Архитектура, Функции, Процессы, Данные, Паттерн, MetaData | орtree, <, view\}$, где $Архитектура = \{архитектура_p, i=1...E\}$ – множество архитектур проектирования;

$Функции = \{функция_i, i=1...Z\}$ – множество проектных функций;

$Процессы = \{процесс_p, i=1...P\}$ – множество проектных процессов;

$Паттерн = \{Операция, Команда, Способ\}$ – множество проектных шаблонов,

$Операция = \{операция_p, i=1...O\}$ – множество проектных операций,

$Команда = \{команда_p, i=1...C\}$ – множество проектных команд,

$Способ = \{способ_p, i=1...S\}$ – множество проектных способов выполнения команды,

$Атом = \{понятие_p, действие_p, i=1...A\}$ – множество «атомов» знаний, состоящее из элементарных понятий и простейших действий,

$Атом \in Эман, Атом \in Процедура, Атом Операция, Атом Команда, Атом Способ;$

$MetaData = \{key_p, hash-function, i=1...H\}$ – мета-данные модели,

где $<key_i>$ – кортеж ассоциативных ключей,

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

$орtree$ – иерархическое отношение;

$<$ – отношение порядка;

$view$ – ассоциативная функция.

Структура паттернов $PatternOperaion$ проектных операций, $PatternComand$ проектных команд, $PatternSposob$ проектных способов одинакова [1]. Например, структура $PatternOperation$ имеет вид:

$PatternOperation = \{название, назначение, мотивация, применимость, структура, участники, отношения, результаты, реализация, пример, применения, родственные паттерны\};$

название – название;

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

мотивация – мотивация, помогает понять абстрактные описания паттерна;

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

структура – структура, представляется графическими диаграммами классов программы с использованием нотации, основанной на методике *Object Modeling Technique* и диаграммами взаимодействий;

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

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

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

реализация – реализация, описывает сложность при реализации паттерна;

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

применения – известные применения в реальных системах;

родственные паттерны – родственные паттерны, имеют описания связи, важных различий, композиции других паттернов с данным.

Графовое представление модели $CADModel$ показано на рис. 2, числа 1, 2, 3, ..., 12 на рисунке обозначают параметры структуры паттерна (1 – название, 2 – назначение и т.д.).

Разработана модель обучаемого инженера, отражающая динамический уровень его подготовленности к решению проектных задач. Уровень описан нечеткими лингвистическими критерияльными параметрами (знания, умения, навык и компетентность). Описание модели обучаемого инженера имеет вид

$UserModel = \{OценкаЗнание_p, OценкаУмение_p, OценкаНавик_p, OценкаКомпетентност_p, haracteristika | calcZ, calcU, calcN, calcK, i=1...N\},$

где $OценкаЗнание_p, OценкаУмение_p, OценкаНавик_p$ и $OценкаКомпетентност_p$ – массивы оценок знаний, умений, навыков и компетентности соответственно, N – число контрольных точек K_i сценария. Областью значений функций расчета указанных оценок являются пары $(D,)$:

$calcZ, calcU, calcN, calcK (D,)$, где D – значение функции евклидово расстояние, – значение функции принадлежности к классу проектной характеристики [1],

$haracteristika = \{оценка_1, оценка_2, оценка_3, ..., оценка_s\}$ – множество лингвистических характеристик. $calcZ: markTeor_i, оценка_p, calcU: mark_i, оценка_p, calcN: t_i, оценка_p, calcK: calcZ, calcU, calcN, оценка_p, где markTeor_p, mark_p, t_i$ определены в модели протокола ниже.

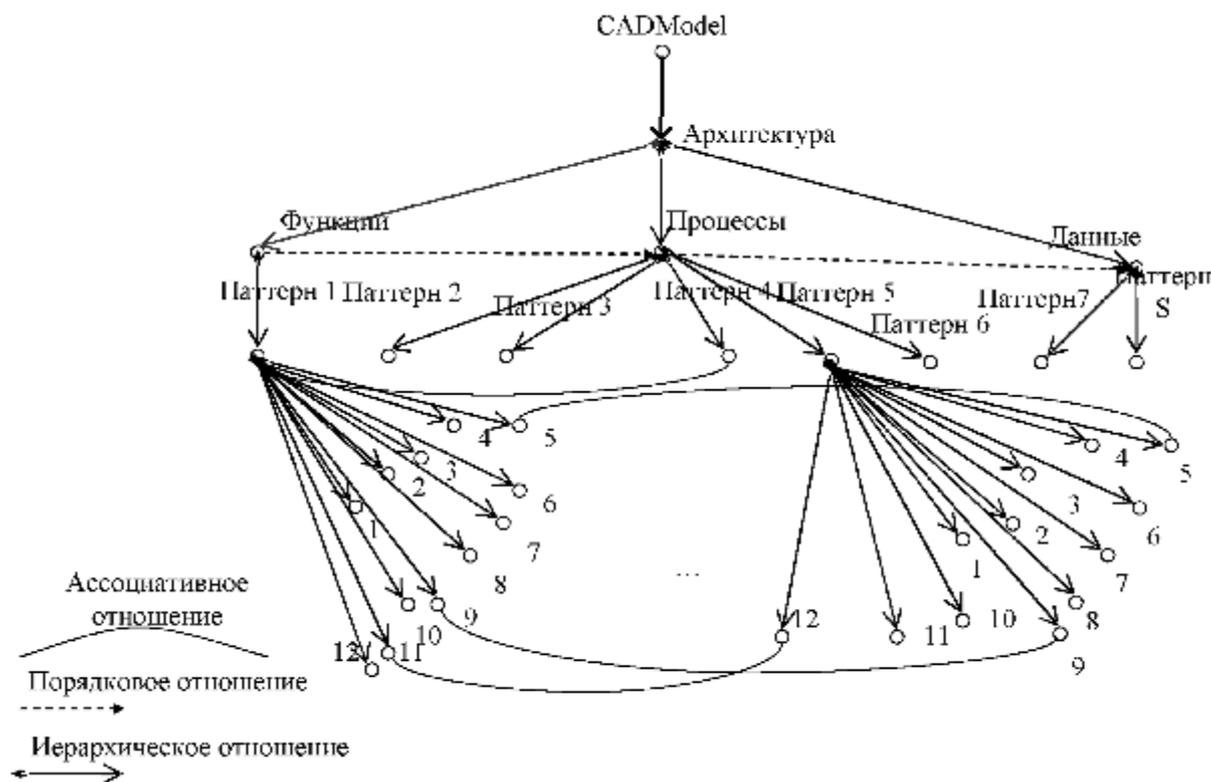


Рис. 2. Графовое представление модели CADModel

Реализация функций $calcZ$, $calcU$, $calcN$, $calcK$ выполнена с помощью нечетких карт Кохонена [1].

Разработана модель сценария обучения. В основу модели сценария положена система, состоящая из ориентированного графа, отображений вершин и альтернативного выбора траектории обучения. Модель сценария имеет вид:

$Scenario = \{G(vertex, edge), Reflection, Alternativ\}$,

где $G(vertex, edge)$ – ориентированный граф сценария, $vertex = \{v_i, i=1...V\}$ – множество атрибутивных вершин,

$edge = \{e_i, i=1...E\}$ – множество дуг;

$Reflection = \{Rf_1, Rf_2, Rf_3, Rf_4\}$ – множество гетерогенных отображений вершин в объекты проектирования (Rf_1 – архитектура, функции, процессы, данные, паттерны (см. модель CADModel), Rf_2 – тестовые вопросы, Rf_3 – практические проектные задачи, Rf_4 – контрольные точки K_i , содержащие требуемые (целевые) значения лингвистических критериальных параметров (проектных характеристик) обучаемого инженера);

$Alternativ = \{v_i, \text{если } v_i \text{ инцидентна } v_j \text{ и } v_i, K_i, v_i < v_j\}$ – выбор обучаемым инженером траектории обучения из v_i неконтрольной вершины.

Ориентированный граф сценария обладает свойствами антирефлексивности, симметричности и транзитивности.

С помощью сценария учитывается механизм ролей обучаемого.

Модель протокола *protocol* представлена вектором параметрических критериев, сформированных из квалиметрии процесса обучения, и имеет вид кортежа $protocol = \langle name, ttype, type_{ij}, test_{ij}, t_{ij}, markTeor_{ij}, mark_{ij}, povtoreniy_{ij}, countTest_{ij}, cognDif_{ij}, i=1...N, j=1...M \rangle$,

где $name$ – идентификатор обучаемого инженера,

$ttype = \{teor, pract\}$ – тип проектной задачи (*teor* – теория, *pract* – практика),

$type_{ij}$ – массив типов проектных задач,

$test_{ij}$ – массив ключей проектных задач,

t_{ij} – массив затраченного времени на выполнение проектных действий,

$markTeor_{ij}$ – массив баллов за ответы на вопросы,

$mark_{ij}$ – массив баллов за выполненные практические проектные задачи,

$cognDif_{ij}$ – массив когнитивной сложности проектных задач,

$povtoreniy_{ij}$ – массив числа повторов решения проектных задач,

$countTest_{ij}$ – массив числа решенных проектных задач,

N – число контрольных точек K_i ,

M – размерность массива.

Накопленная и обработанная информация в процессе обучения хранится в протоколе.

Реализация интеллектуальной обучающей системы представлено линейкой программ: графический конструктор разработки гипертекстовых электронных учебно-методических комп-

лексов, графический конструктор модели предметной области САПР, транслятор SCORM-спецификации проектных решений САПР, система контроля знаний ТЕСКО [2, 3, 4].

Компонентные модели разработаны по технологии *JavaBeans* с целью обеспечения многократного использования компонентов и взаимодействие с похожими компонентными структурами. Например, *Windows*-программа при наличии соответствующего моста или объекта-обертки может использовать компонент *JavaBeans* как компонент СОМ или *ActiveX*.

Таким образом, разработанная система обладает рядом преимуществ по сравнению с известными системами обучения проектной деятельности и характеризуется следующими показателями:

1. использована компонентная архитектура системы, что повышает ее гибкость, универсальность и масштабируемость;
2. предложена формализация предметной области САПР с разными типовыми связями;
3. применение нечетких карт Кохонена в качестве средств адаптации позволяет эффектив-

но учитывать динамические индивидуальные характеристики проектировщика, что повышает качество обучения;

4. реализована динамическая структура сценария обучения проектировщика с поддержкой параметризации его шаблонных решений.

СПИСОК ЛИТЕРАТУРЫ

1. *Афанасьев А.Н., Войт Н.Н.* Разработка компонентной автоматизированной обучающей системы САПР на основе гибридной нейронной сети // Автоматизация и современные технологии. 2009. № 3. С. 14 – 18.
2. Графический конструктор разработки гипертекстовых электронных учебно-методических комплексов / *Войт Н.Н., Афанасьев А.Н.* Свидетельство об официальной регистрации программы для ЭВМ № 2006612714, Роспатент, Москва. 2006.
3. Транслятор SCORM-спецификации проектных решений САПР КОМПАС / *Войт Н.Н., Афанасьев А.Н.* Свидетельство об официальной регистрации программы для ЭВМ № 2008611429, Роспатент, Москва. 2008.
4. Система контроля знаний ТЕСКО / *Афанасьев А.Н., Афанасьева Т.В., Войт Н.Н., Романов А.А.* Свидетельство об официальной регистрации программы для ЭВМ. № 2008615364, Роспатент, Москва. 2008.

INTELLIGENT TRAINING SYSTEM TO STUDY DESIGN OF COMPUTER-AIDED SYSTEMS

© 2010 A.N. Afanasev, N.N. Voit

Ulyanovsk State Technical University

In this article we described architecture, mathematic models, realization of intelligent training system to study concept design of difficult computer-aided systems which actively use software.

Key words: intelligent training system, concept design, computer-aided systems.

Alexander Afanasyev, Candidate of Technics, Professor at the Computers Department. E-mail: a.afanasev@ulstu.ru

Nicolay Voit, Candidate of Technics, Associate Professor at the Computers Department. E-mail: n.voit@ulstu.ru