МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное бюджетное образовательное учреждение
высшего образования

«УЛЬЯНОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

ИНФОРМАТИКА, МОДЕЛИРОВАНИЕ, АВТОМАТИЗАЦИЯ ПРОЕКТИРОВАНИЯ

(ИМАП-2016)

VIII Всероссийская школа-семинар аспирантов,
студентов и молодых ученых
(Россия, г. Ульяновск, 25-26 октября 2016 г.)

Сборник научных трудов

Ульяновск
УлГТУ
2016

УДК 004
ББК 32.97
И74

Редакционная коллегия:

Афанасьев А. Н. – доктор технических наук, профессор кафедры ВТ УлГТУ, первый проректор – проректор по дистанционному и дополнительному образованию, научный редактор сборника;

Негода В. Н. – доктор технических наук, профессор кафедры ВТ УлГТУ;

Соснин П. И. – доктор технических наук, профессор, заведующий кафедрой «Вычислительная техника» УлГТУ;

Ярушкина Н. Г. – доктор технических наук, профессор, первый проректор – проректор по научной работе УлГТУ.

УДК 004

Информатика, моделирование, автоматизация проектирования (ИМАП-2016). VIII Всероссийская школа-семинар аспирантов, студентов и молодых ученых (Россия, г. Ульяновск, 25-26 октября 2016 г.) : сборник научных трудов / под ред. А. Н. Афанасьева. – Ульяновск : УлГТУ, 2016. – 263 с.

В сборнике отражены материалы лекций ведущих ученых и докладов аспирантов, студентов и молодых ученых, представленные на VIII Всероссийской школе-семинаре «Информатика, моделирование, автоматизация проектирования» (ИМАП-2016, г. Ульяновск, 25-26 октября 2016 г.). Тематика докладов охватывает следующие направления предметной области: проектирование автоматизированных систем и систем управления, анализ данных, прикладная информатика и вычислительный интеллект, математическое, программное обеспечение и технологии САПР, моделирование программных и аппаратных средств ВТ, встроенные системы, коммуникационные системы, автоматизация обучения, технологии программирования.

Статьи представлены в авторской редакции.

ISBN 978-5-9795-1599-1

© Коллектив авторов, 2016
© Оформление. УлГТУ, 2016

СОДЕРЖАНИЕ

ЧАСТЬ I. МАТЕРИАЛЫ ЛЕКЦИЙ, ОРГАНИЗУЕМЫХ В РАМКАХ ПРОГРАММЫ ШКОЛЫ-СЕМИНАРА ..... ….7

П.И. Соснин М.В. Галочкин АВТОМАТИЗАЦИЯ МЫСЛЕННОГО ВООБРАЖЕНИЯ В ПРОЦЕССАХ РЕШЕНИЯ ПРОЕКТНЫХ ЗАДАЧ ….8

В.Н. Негода ФОРМАЛИЗАЦИЯ ПОТОКОВ СООБЩЕНИЙ О ДОСТИЖЕНИЯХ УЧАСТНИКОВ ЧЕМПИОНАТА ШКОЛЬНИКОВ ПО ПРОГРАММИРОВАНИЮ……..14

В.Г. Тронин ИНФОРМАЦИОННАЯ ПОДДЕРЖКА НАУЧНОЙ ДЕЯТЕЛЬНОСТИ ВНУТРИ ВУЗА……29

ЧАСТЬ II. МАТЕРИАЛЫ ДОКЛАДОВ УЧАСТНИКОВ ШКОЛЫ-СЕМИНАРА ..... ……35

А.Е. Абрамов АНАЛИЗ ВЛИЯНИЯ ИЗМЕНЕНИЙ В РАЗРАБОТКЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ…….36

А.Е. Абрамов ОПИСАНИЕ ПРОЦЕССА ПРОЕКТИРОВАНИЯ ЗАДАЧ ДЛЯ СИСТЕМЫ J-MARK……...43

Аль- Хусеейни Ханса Азиз Анализ и управление рисками программного обеспечения ……..48

А.Н. Афанасьев, Н.Н. Войт , М.Е. Уханова, И.С. Ионова АНАЛИЗ УСПЕШНОСТИ ПОДГОТОВКИ ПРОИЗВОДСТВА ПРОЕКТНОЙ ОРГАНИЗАЦИИ ……..55

Т.В. Афанасьева О МЕТОДОЛОГИИ НЕЧЕТКОГО ПОДХОДА К АНАЛИЗУ ОРГАНИЗАЦИОННО-ТЕХНИЧЕСКИХ СИСТЕМ…….63

Т.В. Афанасьева, Д.В. Заварзин, А.П. Курлов ОБНАРУЖЕНИЕ АНОМАЛЬНЫХ СОСТОЯНИЙ В ПОВЕДЕНИИ ПРОЦЕССОВ В ОРГАНИЗАЦИОННО-ТЕХНИЧЕСКИХ СИСТЕМАХ ……..69

С.И. Бочков РАЗРАБОТКА И ИССЛЕДОВАНИЕ ВИРТУАЛЬНЫХ РАБОЧИХ МЕСТ В СРЕДЕ OPENSIM …….76

С.И. Бригаднов ИССЛЕДОВАНИЕ МОДЕЛЕЙ ОБУЧАЕМОГО В ПЕРСОНИФИЦИРОВАННЫХ АВТОМАТИЗИРОВАННЫХ СИСТЕМАХ ОБУЧЕНИЯ ……..81

А.А. Васильев ОПИСАНИЕ МОДЕЛИ НАКОПЛЕНИЯ И ФИКСАЦИИ ОПЫТА ПРОЕКТИРОВЩИКОВ ПОСРЕДСТВОМ СОЗДАНИЯ И ПОВТОРНОГО ИСПОЛЬЗОВАНИЯ МЕТОДИК ПРОЕКТИРОВАНИЯ, ОСНОВАННОЙ НА МЕТОДЕ «EXPERIENTIAL LEARNING» Д.А. КОЛБА ……..87

Е.Ю. Воеводин АДАПТИВНАЯ МОДЕЛЬ СИСТЕМЫ ПРИНЯТИЯ РЕШЕНИЙ НА ОСНОВЕ ПЛАСТИЧНЫХ СИНАПСОВ С СИГНАЛАМИ НЕОЖИДАННОСТИ В ЭЛЕКТРОННОМ ОБУЧЕНИИ …….95

Н.Н. Войт, И.В. Беляева АВТОМАТИЗАЦИЯ ПРОЦЕССА ОБУЧЕНИЯ ЯЗЫКАМ ВЫСОКОГО УРОВНЯ …….99

П.М. Волков, И.С. Блохин, З.В. Степчева ОСОБЕННОСТИ ПРИМЕНЕНИЯ АЛГОРИТМОВ НА ГРАФАХ В ПРИКЛАДНЫХ ЗАДАЧАХ ПОВЕДЕНЧЕСКОЙ АНАЛИТИКИ ……104

П.М. Волков, И.С. Блохин, З.В. Степчева СОЦИАЛЬНЫЙ СЕРВИС «FASTCAST» …….110

Д.А. Горшков, В.В. Снежкин, В.А Фолунин SEAM CARVING: АЛГОРИТМ СЖАТИЯ ИЗОБРАЖЕНИЙ С УЧЁТОМ КОНТЕКСТА …114

А.В. Гузаева ROOT CAUSE ANALYSIS - АНАЛИЗ ОСНОВНЫХ ПРИЧИН. ……119

Р.И. Замалетдинов , Д.А. Евсевичев , О.В. Максимова РАЗРАБОТКА ПРОГРАММЫ СОПРЯЖЕНИЯ ВЕБ-КАМЕРЫ И ПЕРСОНАЛЬНОГО КОМПЬЮТЕРА ДЛЯ ИССЛЕДОВАНИЯ ПАРАМЕТРОВ ВИБРАЦИИ…….124

И.И. Исхаков, Д.В. Сазонов, В.А. Фолунин АЛГОРИТМ ПОИСКА A* НА ПРИМЕРЕ ИГРЫ «ПЯТНАШКИ» ……..129

Л.Р. Латыпова СТРАТЕГИЯ ПРИНЯТИЯ РЕШЕНИЯ «ЗА И ПРОТИВ»……135

Д.Г. Лямкин ПЕРЕДВИЖЕНИЕ РОБОТА НА РОЛИКОНЕСУЩИХ КОЛЕСАХ БЕНГТА ИЛОНА ……...139

О.В. Максимова, Д.А. Евсевичев ПРОГРАММА КОНСТРУКТОРСКОГО ПРОЕКТИРОВАНИЯ ТОНКОПЛЕНОЧНЫХ ЭЛЕКТРОЛЮМИНЕСЦЕНТНЫХ СТРУКТУР …….143

В.В. Мерясева ОБЗОР ДИАГРАММНЫХ ГРАФИЧЕСКИХ ЯЗЫКОВ САПР ……...149

Р.С. Молотов ПРОЕКТИРОВАНИЕ МОДУЛЯ СЕТЕВОЙ КОММУНИКАЦИИ ИНФОРМАЦИОННОЙ СИСТЕМЫ ……...152

В.С. Мошкин ПРЕДСТАВЛЕНИЕ ПРЕДМЕТНОЙ ОНТОЛОГИИ С ИСПОЛЬЗОВАНИЕМ ИНСТРУМЕНТОВ ПЛАТФОРМЫ 1С:ПРЕДПРИЯТИЕ ……...160

Обаид Али хамза Обаид Аль-Фурат Аль-Аусат ИСПОЛЬЗОВАНИЕ ПРОТОТИПИРОВАНИЯ В ТЕСТИРОВАНИИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ……167

П.Ю. Павлов , П.И. Соснин РОБОТИЗИРОВАННЫЕ СВАРОЧНЫЕ КОМПЛЕКСЫ В СБОРКЕ ТРУБОПРОВОДОВ ЛЕТАТЕЛЬНОГО АППАРАТА ……...174

А.А. Пушкарева АНАЛИЗ РАЗЛИЧНЫХ ПОДХОДОВ К РЕШЕНИЮ И ПЕРСПЕКТИВ РЕАЛИЗАЦИИ ЗАДАЧ ФОРМИРОВАНИЯ И ИСПОЛЬЗОВАНИЯ ПРОЕКТНЫХ ОНТОЛОГИЙ ……..186

Т.В. Старшова Исследование разработки средств управления человеко-компьютерными прерываниями в вопросно-ответной моделирующей среды WIQA.NET ……..191

Т.В. Старшова ОБЗОР РАЗРАБОТКИ БИЗНЕС-ЭКСПЕРИМЕНТА …..197

А.С. Степанов ОПИСАНИЕ ПРОЦЕССА РАЗРАБОТКИ В ВИРУТАЛЬНОЙ СРЕДЕ OPENSIM ……201

А.Н. Стратонов ОБЗОР ГОТОВЫХ РЕАЛИЗАЦИЙ КОМПЬЮТЕРНЫХ ТРЕНАЖЕРНЫХ КОМПЛЕКСОВ ……..205

А.Н. Стратонов ОБЗОР ДИАГРАММ ИШИКАВЫ. ШЕСТЬ ПРОСТЫХ ШАГОВ ИХ ПОСТРОЕНИЯ. ……...210

И.А. Тимина МОДЕЛИРОВАНИЕ ПРОГНОЗА ПРОЕКТНЫХ МЕТРИК ПУТЕМ ИХ ИНТЕЛЛЕКТУАЛЬНОГО АНАЛИЗА ……..216

В.А. Хомяков ОПРЕДЕЛЕНИЕ БУДУЩИХ ПОСЛЕДСТВИЙ СОБЫТИЯ ……...222

Д.Э. Цыганков , А.Ф. Похилько ДЕКОМПОЗИЦИЯ ЭЛЕКТРОННОЙ МОДЕЛИ ИЗДЕЛИЯ ДО УРОВНЯ СТРУКТУРНО-ФУНКЦИОНАЛЬНЫХ ЭЛЕМЕНТОВ ……...225

Г.И. Шайдуллова АВТОМАТИЗАЦИЯ ОЦЕНКИ КАЧЕСТВА РЕАЛИЗАЦИИ ТАБЛИЦ КАСКАДНЫХ СТИЛЕЙ (CSS-MARK) …….229

Г.И. Шайдуллова ИСПОЛЬЗОВАНИЕ ДИАГРАММ СИСТЕМ …….234

А.Н. Афанасьев, В.А. Гордеев, О.Г. Шаров ОБЗОР ТЕХНОЛОГИЙ И РАБОЧИХ ПРОЦЕССОВ SHAREPOINT……238

А.Н.Афанасьев, С.Ю.Кириллов, Е.Ю.Воеводин РАЗРАБОТКА ГЕНЕРАТОРА НЕГАТИВНЫХ BPMN ДИАГРАММ НА ОСНОВЕ RV-ГРАММАТИКИ …….247

А.Н. Афанасьев, С.Ю. Кириллов, Ю.А. Савичева РАЗРАБОТКА ДИНАМИЧЕСКОГО ГЕНЕРАТОРА НЕГАТИВНЫХ BPMN ДИАГРАММ НА ОСНОВЕ RV-ГРАММАТИКИ ……..255

ЧАСТЬ I

МАТЕРИАЛЫ ЛЕКЦИЙ, ОРГАНИЗУЕМЫХ В РАМКАХ ПРОГРАММЫ ШКОЛЫ-СЕМИНАРА

УДК 621.372

АВТОМАТИЗАЦИЯ МЫСЛЕННОГО ВООБРАЖЕНИЯ В ПРОЦЕССАХ РЕШЕНИЯ ПРОЕКТНЫХ ЗАДАЧ

П.И. Соснин[1] М.В. Галочкин[2]

Рассматриваются вопросы автоматизации мысленного экспериментирования. Предлагается типизированный набор образно-семантических моделей, экземпляры которых выполняют функцию интерфейса между процессом решения задачи и мысленным воображением.

Введение

Расширяющаяся компьютеризация всех сфер человеческой активности, включая различные виды персональной и коллективной деятельности, уже вывела на необходимость осуществления профессиональной деятельности в условиях социо-киберфизической реальности (СКФ-реальности), в которой различные неодушевленные предметы и физическая реальность, за счет использования разнообразных вычислительных ресурсов и сенсорных возможностей, наделяются подобием интеллектуальных функций и интеллектуального поведения.

Создавая человека, Природа, адаптировав его сущность под свое естество, не рассчитывала на появление таких кибер-посредников во взаимодействии человека с его окружением, что уже сейчас является источником серьезных проблем, среди которых выделяется «проблема чрезвычайно низкой успешности (степень успешности около 40%, несмотря на попытки её повысить за последние 20 лет) разработок таких составляющих СКФ-реальности как системы, интенсивно использующие программное обеспечение.

Отмеченная причина проблем (существенное различие между естественным взаимодействием человека с физическим миром и его взаимодействием с компьютеризованными приложениями) и сами проблемы приводят к новым постановкам вопросов и идей о формах человеко-компьютерного взаимодействия с СКФ-реальностью. Среди

[1] *432027, Ульяновск, ул. Северный Венец, 32, УлГТУ, e-mail:sosnin@ulstu.ru*

[2] *432027, Ульяновск, ул. Северный Венец, 32, УлГТУ, e-mail:m.v.galochkin@ulstu.ru*

таких вопросов и идей особо важны те, которые имеют отношение к процессам разработки составляющих СКФ-реальности, потому что в них нельзя не учитывать необходимость эффективного переплетающегося комплексирования естественных и компьютеризованных составляющих взаимодействия разработчиков (в первую очередь проектировщиков) со средой разработки. Более того, эти специалисты должны учитывать будущее взаимодействие пользователей с разрабатываемыми компьютеризованными продуктами.

Одной из таких идей, центральной для предлагаемого проекта, является следующая:

Повышению естественности (и посредством этого повышению эффективности) взаимодействия проектировщика со средой его деятельности в создании составляющих СКФ-реальности, способствует применение им совокупности методов и средств интеллектуальной обработки решений проектных задач, в которой используются аналогии с интеллектуальной обработкой условных рефлексов [3].

Отметим, что в интеллектуальную обработку условных рефлексов природой вовлекаются интеллектуальные механизмы правого полушария, интегрирующие активность органов чувств и/или имитирующие такую активность, и механизмы левого полушарий, включающие в интеллектуальную обработку использование естественного языка, причём в условиях, когда интеллектуальные механизмы активизируются согласованно [4]. Уменьшая указанные выше различия, полезно адекватно моделировать такие составляющие интеллектуальной обработки как: мысленное экспериментирование на основе данных, поступающих от органов чувств и активности феномена "интеллектуальное воображение (mental imagery, имитация восприятия без потока данных от органов чувств)"; диалоговая структуризация мысленных представлений и их семантический анализ; итеративное согласование названных интеллектуальных активностей, порождающее понимание результатов обработки [5].

Предварительные предпосылки

Идея (с учётом уточнений) определена так, чтобы её материализация помогла внести вклад в решение проблемы успешности разработок таких принципиальных составляющих СКФ-реальности как SIS в её следующей формулировке:

Существующие на настоящий момент времени существенные различия между формами естественной активности интеллектуальных механизмов правого и левого полушарий человека в условиях применения им наличного опыта в задачных ситуациях и формами использования таких

механизмов, которые в подобных ситуациях «навязаны» проектировщикам средой разработки SIS, являются источником проявлений человеческого фактора, оказывающих весомое негативное воздействие на процесс проектирования и его результат.

Привязка идеи проекта и выделенной проблемы к проектировщикам SIS обусловлена тем, что успешность в области их профессиональной деятельности оценивается и анализируется более 20 лет, что является богатым источником «подсказок», как и уже проведённые попытки изменить положение дел в этой предметной области.

Акцент на указанные выше интеллектуальные активности правого и левого полушарий используется из-за того, что в предлагаемом проекте центральное место занимает моделирование феномена «интеллектуальное воображение», без включения которого в проектную деятельность она просто невозможна. Проектировщикам особенно на концептуальном этапе проектирования, постоянно приходится «интеллектуально воображать», представляя сложившееся в процессе проектирования (или необходимое для процесса или его результата) без опоры на информационные потоки, порождаемые органами чувств при взаимодействии с физической реальностью, то есть без опоры на естественное предназначение органов чувств [3].

Отметим, что природа интеллектуального воображения не замыкается на зрении (слепые, и даже слепоглухонемые люди), но для любого органа чувств эффект от воображения повышается, если к этому процессу подключается использование его полезных моделей. Для зрения роль таких моделей выполняют визуальные образы, использование которых для проектирования носит принципиальный характер.

Основная задача

Исходя из выбранных установок, сформулирована следующая обобщённая задача [1]:

1. Разработать комплекс средств, обеспечивающих пошаговое формирование постановки проектной задачи, состояние которой представляется взаимосвязанной совокупностью текстовых, графических и концептуально-алгоритмических моделей.

2. В основу пошагового формирования положить механизмы разработок, управляемых моделями, в процессе которых положить итеративное согласование текстов с графикой и алгоритмической составляющей, нацеленное на обеспечение необходимого понимания и повторного использования.

3. Разработка должна осуществляется в инструментально моделирующей среде WIQA, так чтобы построенный комплекс средств был включен в нее как расширение.

Проведён вопросно-ответный анализ задачи, одним из результатов которого является выбор следующего набора типов образно-семантических моделей [2]:

1. Изобразительный тип, обеспечивающий визуальную поддержку мысленного воображения и представления его эффектов в прецедентно-ориентированном решении проектных задач, использующем автоматизированное мысленное экспериментирование.

2. Декларативный тип, обслуживающий акты понимания в формировании постановки задачи и логическую проверку её приращений в пошаговом вопросно-ответном анализе.

3. Концептуально–алгоритмический тип, обслуживающий визуальную поддержку в построении концептуальных программных решений в работе с проектными задачами и их моделями прецедентов.

Образно-семантическое моделирование

В основу теоретизации положено отображение процессов концептуального решения проектных задач на семантическую память инструментальной среды WIQA, обобщённое представление которой приведено на рисунке 1.

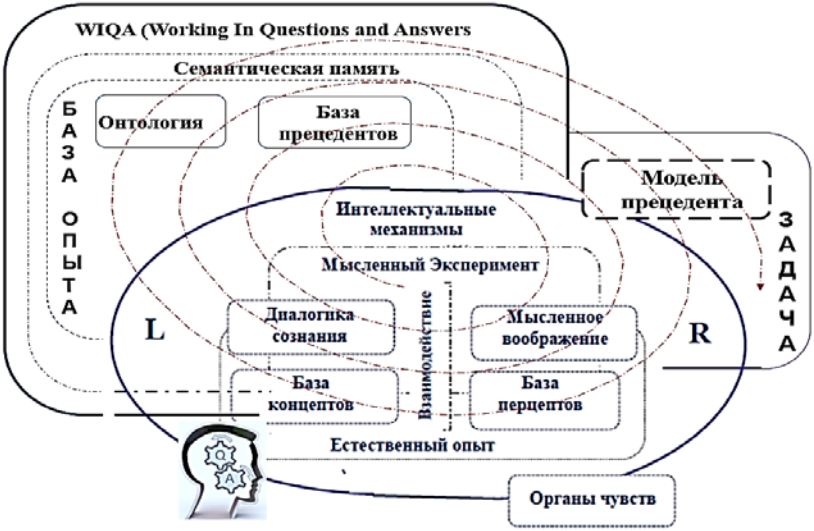


Рисунок 1. Среда концептуального решения проектных задач

На схеме показано, что в процессе решения задачи проектировщик, решая задачу и создавая для неё модель прецедента, проводит автоматизированные мыленные эксперименты, используя базу (моделей) опыта, включающую «Онтологию» проектирования и «Базу прецедентов». В автоматизацию любого из мысленных экспериментов конструктивно вовлечены диалогика сознания и мысленное воображение, взаимодействующие с естественным опытом, включающем прообразы базы концептов (онтология, левое полушарие L) и базы перцептов (правое полушарие R) [3].

За «спиралью» на схеме стоит реализация метода понятийно-образной поддержки процесса пошаговой детализации в прецедентно-ориентированном решении задач, обобщённое представление которого приведено на рисунке 2, где центральное место занимает нормативная структура прецедента P, включающая текст постановки задачи PT, вопросно-ответную модель PQA его анализа, логическую модель PL, графическое представление PG, исходный исполняемый псевдокод PI и программное представление PE прецедента.

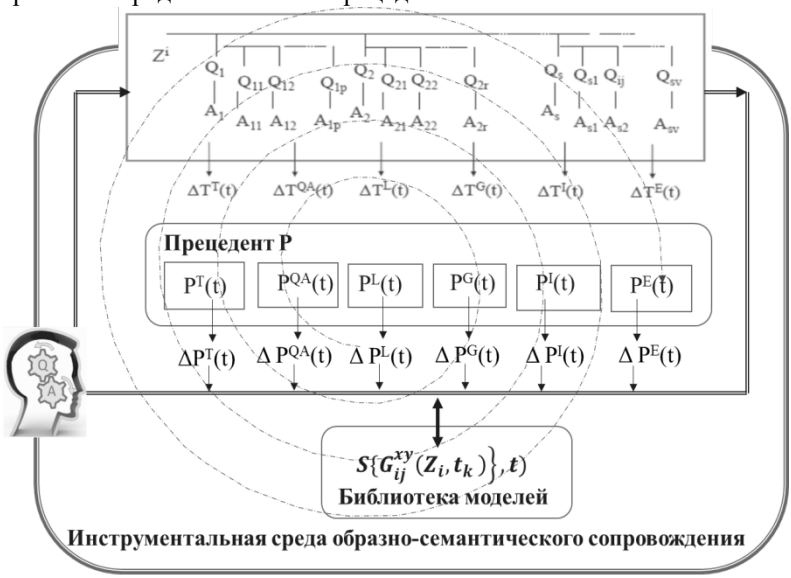


Рисунок 2. Метод пошаговой детализации постановки задачи

Специфику метода определяют следующие его составляющие:

1. Итеративная реализация метода управляется текущими результатами {ΔTX(t), X=(T, QA, L, G, I, E)} пошаговой детализации

постановки задачи TP(Z, t), регистрация которых проводится в дереве PQA(t) вопросно-ответного анализа.

2. Каждое приращение ΔTX(t) способно привести к приращению ΔPX(t) соответствующей составляющей прецедента P, среди которых для образно-семантической поддержки принципиальное место занимают образно-семантические модели, которые аккумулируются в библиотеку моделей.

3. Каждая графическая модель, в зависимости от её типа (изобразительный, декларативный, концептуально-алгоритмический) создаётся проектировщиком в соответствующей операционной среде специализированного графического редактора.

4. Каждую графическую модель, если в этом имеется необходимость, проектировщик может подключить к соответствующему блоку постановки задачи.

5. Для повышения эффективности визуального моделирования, проектировщику предоставлена возможность автоматизированного преобразования типов (рисунок 3) для создаваемых или созданных графических моделей.

Список литературы

1. Галочкин М.В. Соснин П.И. Подход к представлению текстовой информации в виде графо-семантических диаграмм // В мире научных открытий. Красноярск: Научно-инновационный центр, 2015. № 8.2(68). - с.734-746
2. Соснин П.И., Галочкин М.В. Средства псевдокодовой программируемой графики в проектировании автоматизированных систем //«автоматизация процессов управления» № 1 (39) 2015 - с. 82-88
3. Sosnin P. Question-answer programming and modeling in expert systems. In Proceedings of the IASTED International Conference on Artificial 2009 IASTED International Conference on on Artificial Intelligence and Applications, AIA 2009, Innsbruck, 2009, pp. 208-213.
4. Sosnin P. Question-answer approach to human-computer interaction in collaborative designing. Cognitively Informed Intelligent Interfaces: Systems Design and Development 2012, pp. 157-176.
5. Sosnin P. Scientifically experimental way-of-working in conceptual designing of software intensive systems. In Proceedings of the 12th IEEE International Conference on Intelligent Software Methodologies, Tools and Techniques (SoMeT 2013), 2013, pp. 43-50.

УДК 004.588

ФОРМАЛИЗАЦИЯ ПОТОКОВ СООБЩЕНИЙ О ДОСТИЖЕНИЯХ УЧАСТНИКОВ ЧЕМПИОНАТА ШКОЛЬНИКОВ ПО ПРОГРАММИРОВАНИЮ

В.Н. Негода[3]

Рассматривается вопросы формализации потоков сообщений о достижениях участников чемпионата ИТ-сферы Ульяновской области по программированию среди школьников. Строятся формальные представления потоков событий и сообщений, содержащих большое количество данных, существенно влияющих на мотивацию участников.

Введение

Чемпионат ИТ-сферы Ульяновской области по программированию среди школьников проводится с марта 2015 года [1]. В 2016 году это система из 13 турниров [2], разбитых на 4 сезона: зима, весна, лето, осень. Каждый сезон проводится 3 турнира: тренировочный, отборочный и финальный. 12 сезонных турниров предполагают индивидуальное участие с подсчетом суммарных показателей по учебным заведениям. Каждый турнир представляет собой один контест, проводимый под управлением среды Contester 2.4 [3]. После индивидуальных турниров в рамках сезона «Осень-2016» запланирован заключительный командный суперфинал, в который приглашаются команды по три человека от лучших школ по итогам всех турниров года.

С целью увеличения массовости и результативности чемпионата в 2016 году в его организацию вовлекается широкий спектр мотивационных механизмов, основанных на технологиях геймификации обучения [3]. Основу геймификации обычно составляет система рейтингов, дающая основание порождать различные события успеха, с которыми связываются различные «пряники», представляющие собой либо сообщения о достижениях, либо символические награды – бэйджи.

В чемпионате 2016-го года используется система из многих десятков рейтингов и индексов прогресса [4], значительная часть текущих значений которых отображается в формате таблиц на сайте чемпионата ulivt.ru.

[3] *432027, Ульяновск, ул. Северный Венец, 32, УлГТУ, e-mail:nvn@ulstu.ru*

Такое представление рейтингов хорошо мотивирует школы и участников, претендующих на верхние строки таблиц. С целью мотивации остальных организована публикация истории достижений. Эта история формируется таким образом, чтобы даже участники или школы, имеющие относительно низкие рейтинговые показатели, могли получать сообщения о достижениях.

Система формирования рейтингов и истории достижений развивается эволюционно. Эволюционный характер развития и большое разнообразие достижений порождает риски нарушения концептуальной целостности проектных решений. Для обеспечения концептуальной целостности целесообразно выполнить формализацию потоков сообщений о достижениях. Формализация охватывает базовые множества рейтинговых показателей, потоки событий и потоки сообщений, формируемых на основе событий.

1. Базовые множества рейтинговых показателей

В качестве основы геймификации чемпионата используются количество решенных задач и следующее множество типов рейтинговых показателей:

$MARPL = \{M, LM, A, LA, R, LR, P, LP\}$,

где M – массовость, оцениваемая суммой числа учеников школ, решивших хотя бы одну задачу;

LM – место в таблице рейтинга массовости;,

A – активность, оцениваемая суммой $Abase + Abonus$, где $Abase$ – число задач, решенных в любых турнирах, $Abonus$ – бонусы за стартовую активность в тренировочных и отборочных турнирах (по одному бонусному баллу за каждую задачу в первые двое суток турнира) и активность вне турниров, оцениваемую оргкомитетом чемпионата;

LA – место в таблице рейтинга активности;

R – успешность (результативность), подсчитываемая по формуле $R = C+3*F+5*S$, где C – число задач, решенных в отборочных турнирах, F – число задач, решенных в сезонных финалах, S – число задач, решенных в итоговом командном суперфинале; рейтинг успешности определяет выход в финалы сезонов и заключительный суперфинал;

LR – место в таблице рейтинга успешности;

P – мастерство (профессионализм), оцениваемое по формуле $P = \Sigma\ 5/N_k$, где N_k – число участников турнира, решивших k-ю задачу; суммирование проводится по всем задачам турнира;

LP – место в таблице рейтинга мастерства.

Рассмотренная выше система показателей MARPL является структурообразующей. Сами базовые множества рейтинговых показателей

порождаются на основе локализации показателей по контингенту участников (школы-участники, учащиеся-участники) и по этапам чемпионата. Эти локализации порождают шесть множеств базовых показателей:

Vs = {*Ms*, *LMs*, *As*, *LAs*, *Rs*, *LRs*, *Ps*, *LPs*} – множество множеств рейтингов и мест в годовых таблицах рейтингов для школ, ученики которых участвуют в чемпионате; каждое множество содержит значения соответствующего рейтингового показателя для все школ, ученики которых решили хотя бы одну задачу в течение года.

Vu = {*Au*, *LAu*, *Ru*, *LRu*, *Pu*, *LPu*} – множество множеств рейтингов и мест в годовых таблицах рейтингов для учащихся школ, которые рассматриваются как основные пользователи системы управления чемпионатом. Каждое множество содержит значения соответствующего показателя для всех участников чемпионата, решивших хотя бы одну задачу в течение года.

Vsc = {*Msc*, *LMsc*, *Asc*, *LAsc*, *Rsc*, *LRsc*, *Psc*, *LPsc*} – множество множеств рейтингов и мест в таблицах рейтингов отдельных турниров (контестов) для школ. Для 13 турниров 2016 года имеется 13 наборов множеств указанного типа.

Vut = {*Aut*, *LAut*, *Rut*, *LRut*, *Put*, *LPut*} – множество множеств рейтингов и мест в таблицах рейтингов отдельных турниров для учащихся.

Vss = { *Mss*, *LMss*, *Ass*, *LAss*, *Rss*, *LRss*, *Pss*, *LPss*} – множество множеств рейтингов и мест в сезонных таблицах рейтингов для школ, ученики которых участвуют в чемпионате. Для четырех сезонов имеет место 4 набора множеств данного типа.

Vus = {*Aus*, *LAus*, *Rus*, *LRus*, *Pus*, *LPus*} – множество множеств рейтингов и мест в сезонных таблицах рейтингов для учащихся школ.

2. Индексы прогресса рейтингов

Индексы прогресса рейтингов позволяет выделять тех участников и те школы, которые на каком-то отрезке времени сделали «рывок». В коротких образовательных мероприятиях это могут быть минуты, например: первые десять минут, вторые, первые полчаса, вторые полчаса.

В круглогодичном чемпионате целесообразно выделять три отрезка времени, существенно отличающиеся размером – например: день, неделя, месяц; день, 10 дней, 100 дней. Ограничение количества отрезков связано со сложностью отображения большего количества на экране.

Для уменьшения все нарастающей сложности в обозначениях рейтингов индексы прогресса будем выделять префиксами имен: *d* – день, w – средний интервал (неделя или 10 дней), *q* – максимальный интервал

(квартал либо 100 дней). В итоге для общегодовых результатов школ получаем следующие множества индексов:

IPS = {dMs, wMs, qMs, dAs, wAs, qAs, dRs, wRs, qRs, dPs, wPs, qPs, dLMs, wLMs, qLMs, dLAs, wLAs, qLAs, dLRs, wLRs, qLRs, dLPs, wLPs, wLPs} – множество индексов прогресс школ;

IPU = {dAu, wAu, qAu, dRu, wRu, qRu, dPu, wPu, qPu, dLAu, wLAu, qLAu, dLRu, wLRu, qLRu, dLPu, wLPu, wLPu} – множество индексов прогресс участников.

Введенное выше правило именование позволяет уменьшить сложность определения новых переменных формальной модели до такой степени, что само определение можно вообще опустить, если оно базируется на соглашениях. Осознание соглашений делает очевидным восприятие, например, dAs как дневного прогресса рейтинга активности школы, а qPu – как прогресса рейтинга мастерства участника на самом длинном интервале времени. Нарушение традиций использования однобуквенных имен в формулах (сначала латинские, затем греческие, затем готические) оправдано двумя обстоятельствами:

- в формализацию вовлекается очень большое число различающихся элементов данных, а значит и имен;
- порождаемые формализмы являются инструментом реализации автоматизированной системы, что требует уменьшения семантического разрыва между содержанием формализмов и спецификациями баз данных, массивов, функций, классов.

3. Потоки событий чемпионата

Потоки сообщений о достижениях в чемпионате продуцируются на основе потоков событий. Опыт создания средств автоматизации проведения чемпионата, накопленный в шести турнирных сезонах 2015-го и 2016-го годов, обуславливает целесообразность выделения четырех видов потоков событий: календарного, исходного, основного производного и дополнительного производного.

Календарный поток событий представляет собой множество:

$EC = \{<ectype_k, ectime_k> \mid k \in N\}$,

где $N = \{1,2, \ldots\}$;

$ectype_k$ – спецификатор календарного события;

$ectime_k$ – время наступления события, включающее в себя дату и астрономическое время.

Во множество типов событий включаются моменты начала и завершения турниров (контестов), сезонов и года. Для поддержки оценки прогресса рейтингов в календарный поток событий включается моменты времени перехода к следующим турнирным дням.

Исходный поток событий отражает результаты решения задач участниками чемпионата. Он порождается проверяющими машинами, которые формируют те или иные вердикты по итогам проверки решений турнирных задач. Вердикты проверяющих машин образуют множество $VM = \{vm_k \mid k \in N\}$. Для того, чтобы информация о вердиктах смогла обслуживать генерацию мотивирующих сообщений, элемент vm_k должен содержать идентификатор участника IDu, идентификатор турнира (контеста) IDc, идентификатор задачи IDp, код вердикта VC и момент времени поступления задачи в адрес проверяющей машины в формате дата-время DT, что порождает вектор следующего вида:

$vm_k = < IDu_k, IDc_k, IDp_k, VC_k, DT_k>$.

Упорядочение по времени DT_k множества событий VM образует последовательность вердиктов:

$SV = (sv_1, sv_2, \ldots), sv_k \in VM, \forall ij\ (i<j \rightarrow DT_i < DT_j)$.

Именно эта последовательность является формальной спецификацией исходного потока событий.

Основной производный поток событий отражает измененные значения рейтинговых показателей школ и участников. Этот поток формируется анализатором событий исходного потока на основе системы рейтинговых показателей, относящихся к участникам и школам, что предопределяет два множества:

$EU = \{eu_1, eu_2, \ldots\}$ – совокупность событий, относящихся к участникам;

$ES = \{es_1, es_2, \ldots\}$ – совокупность событий, относящихся к школам.

Элементы этих множеств являются векторами, структура которых предопределяется изложенной выше системой рейтинговых показателей и имеет вид:

$eu_i = <IDu_i, IDc_i, NPu_i, Au_i, Ru_i, Pu_i>$, где IDu_i – идентификатор участника, IDc_i – идентификатор контеста, NPu_i, Au_i, Ru_i, Pu_i – соответственно число решенных задач, значение рейтингов активности, успешности и мастерства в контесте IDc_i.

$es_i = <IDs_i, IDc_i, NPs_i, Ms_i, As_i, Rs_i, Ps_i>$, где IDs_i – идентификатор школы, NPs_i, Ms_i, As_i, Rs_i, Ps_i – соответственно число решенных задач, значение рейтингов массовости, активности, успешности и мастерства школы в контесте IDc_i

В незавершенном контесте последние компоненты векторов eu_i и es_i представляют собой нарастающие значения рейтинговых показателей. По завершению контеста IDc_i в этих векторах фиксируются итоговые результаты.

Рассмотренные множества элементов событий трансформируются в потоки путем упорядочения по дате-времени контеста, что порождает две базовые последовательности:

$SU = (su_1, su_2, \ldots)$ – поток событий изменения рейтингов участников, где $su_i \in EU$;

$SS = (ss_1, ss_2, \ldots)$ – поток событий изменения рейтингов школ, где $ss_i \in ES$.

Пересчет значений рейтингов запускается по событию успешного решения задачи, обнаруженному в основном потоке *SV*. Это же событие запускает пересчет сезонных и годовых рейтингов, формируемых как суммы рейтингов всех турниров сезона и суммы рейтингов всех турниров года.

Мотивирующий потенциал основного производного потока событий ограничен несколькими обстоятельствами:

- мотивация действует в основном только на тех участников и школ, которые располагаются в верхних строках таблиц рейтингов; осознание сложности работы, которую нужно выполнять для перемещения с нижних строк на верхние, зачастую демотивирует;
- решение значительного числа задач может не приводить к улучшению положения в таблицах рейтингов, если соперники решили больше;
- динамика изменения положения в таблицах рейтингов имеет свойство ослабевать – лидеры уходят в отрыв, аутсайдеры после решения относительно простых задач сталкиваются с тем, что каждая новая задача требует все больших усилий, а шансы заметно улучшить турнирное положение стремятся к нулю;
- в публикации таблиц рейтингов зачастую сложно распознать изменение к лучшему.

Дополнительный производный поток событий предназначен для компенсации влияния перечисленных обстоятельств. Это делается на основе следующего:

- введения понятия «порог вхождения в рейтинг», благодаря которому попадание даже на последнюю строку таблицы рейтинга становится достижением;
- такой интерпретации событий основного потока, в которой важным достижением участника или школы является улучшение собственных показателей, позволившее преодолеть какой-то рубеж;
- интерпретации достижений школы как личных достижений того ученика, чье решение задачи позволило школе преодолеть какой-то рубеж;

- учета фактов решения турнирной задачи первым, что для простых задач может быть следствием не столько мастерства участника, сколько активности в стартовой фазе многодневного турнира;
- фиксация турнирных, сезонных и годовых рекордов как всего чемпионата в целом, так и внутришкольных рекордов.

Дополнительный поток событий в системе управления чемпионатом называется историей достижений и представляет собой последовательность:

$H = (h_1, h_2, \ldots)$,

где $h_i = < IDs_i, IDu_i, type_i, us_i, time_i, par_i >$;

IDs_i, IDu_i – идентификаторы школы и участника, к которым относится достижение;

$type_i$ – тип события;

us_i – код принадлежности достижения школе или участнику (0 – только школе, 1 – только участнику, 2 – участнику и школе);

$time_i$ – время события; par_i – вектор параметров события, который может включать в себя значения: *year* – год; *season* – сезон, *tour* – тип турнира, *rate* – тип рейтинга, *problems* – номер или количество задач, *newval*, *oldval* – новое и старое значения, *period* – код временного отрезка, на котором оцениваются прогрессы рейтингов, *oldyear*, *oldseason* – год и сезон, к которому относится превзойденное достижение.

С каждым типом события связаны два предиката, истинность которых определяет процесс генерации. Первый предикат является инициирующим, поскольку определяет моменты времени, когда запускается процесс генерации события из потока *H*. В генераторе событий этого потока используются следующие инициирующие предикаты:

pAcc – обнаружение в исходном потоке *SV* события успешного решения задачи (вердикт «accepted» проверяющей машины);

pEndT – наступило календарное событие «завершения турнира»;

pEndS – наступило календарное событие «завершение сезона»;

pEndY – наступило календарное событие «завершение года»;

pEndD – наступило календарное событие «завершение турнирного дня».

Второй предикат идентифицирует факт обстоятельств, вызывающих генерацию конкретного элемента потока *H*. В обобщенной спецификации этих элементов, приводимой ниже, оба предиката представляются в качестве посылки импликации, следствием которой является запуск генерации спецификации элемента. Поскольку с каждым решением задачи связаны идентификаторы участника и школы, где он учится, а также с каждым типом достижения связаны конкретное время решения и

конкретное правило генерации, компоненты *IDu*, *IDs*, *type* и *time* в приводимой ниже спецификации не описываются. Генерация этих компонентов не зависит от типа достижений

По состоянию на октябрь 2016 года во множество событий дополнительного потока входят:

- решение первой задачи в чемпионате:

pAcc&(*pYcount*==0) → (*us* ∈ {1, 2} & *par* = <*year*>),

где *pYcount* – значение счетчика числа решенных задач школой или участником от начала года до момента *time*;

- решение первой задачи в сезоне:

pAcc&(*pScount*==0) → (*us* ∈ {1, 2} & *par* = <*year, season*>),

где *pScount* – значение счетчика числа решенных задач школой или участником от начала сезона до момента *time*;

- решение первой задачи в турнире:

pAcc&(*pTcount*==0) → (*us* ∈ {1, 2} & *par* = <*year, season, tour*>),

где *pTcount* – значение счетчика числа решенных задач школой или участником от начала турнира до момента *time*;

- преодоление порога вхождения в рейтинг:

pAcc&(*valRate* > *minRate*) → (*us* ∈ {0, 1, 2}, *par* = <*rate*>),

где *valRate* – значение рейтинга *rate*, *minRate* – порог вхождения;

- раньше других решена задача в турнире:

pAcc&(*countRproblem* = 0) → (*us* = 2, *par* = <*problems*>),

где *countRproblem* – число решений задачи *problems* в данном турнире;

- перемещение с одного места рейтинговой таблицы на другое в годовом рейтинге:

pAcc&(*newval* < *oldval*) → (*us* ∈ {1, 2}, *par* = <*year, rate, newval, oldval*>),

где в качестве *newval* и *oldval* используются позиции годовой рейтинговой таблицы до данного решения и после него;

- перемещение с одного места рейтинговой таблицы на другое в сезонном рейтинге:

pAcc&(*newval* < *oldval*) → (*us* ∈ {1, 2}, *par* = <*year, season, rate, newval, oldval* >),

где *newval* и *oldval* задают позиции в сезонных таблицах рейтингов;

- перемещение с одного места рейтинговой таблицы на другое в турнирном рейтинге:

pAcc&(*newval* < *oldval*) → (*us* ∈ {1, 2}, *par* = <*year, season, tour, rate, newval, oldval* >),

где *newval* и *oldval* задают позиции в турнирных таблицах рейтингов;

- по итогам чемпионата удалось занять более высокое место в указанном рейтинге по сравнению с предыдущими результатами:

pEndY&(*newval* < *oldval*) → (*us* ∈ {0,1}, *par* = <*year, rate, newval, oldval*>),
где *newval* и *oldval* задают соответственно позицию в текущей годовой турнирной таблице рейтинга *rate* и лучшее значение этого рейтинга в предыдущие годы;

- по итогам сезона удалось занять более высокое место в указанном рейтинге по сравнению с предыдущими результатами:

pEndS&(*newval* < *oldval*) → (*us* ∈ {0,1}, *par* = <*year, season, rate, newval, oldval* >),
где *newval* и *oldval* задают соответственно позицию в текущей сезонной турнирной таблице рейтинга *rate* и лучшее значение этого рейтинга в предыдущие сезоны;

- по итогам турнира удалось занять более высокое место в указанном рейтинге по сравнению с предыдущими результатами:

pEndT&(*newval* < *oldval*) → (*us* ∈ {0,1}, *par* = <*year, season, tour, rate, newval, oldval* >),
где *newval* и *oldval* задают соответственно позицию в текущей таблице рейтинга *rate* и лучшее значение этого рейтинга в предыдущие турниры;

- удалось войти в круг решивших *problems* задач в чемпионате:

pAcc&(*problems-1* ∉ *bYproblems*)&(*problems* ∈ *bYproblems*) →
(*us* ∈ {1,2}, *par* = <*year, rate, problems*>),
где *bYproblems* – множество рубежных значений для числа решенных задач в одном году;

- удалось войти в круг решивших *problems* задач в сезоне:

pAcc&(*problems-1* ∉ *bSproblems*)&(*problems* ∈ *bSproblems*) →
(*us* ∈ {1,2}, *par* = <*year, rate, problems*>),
где *bSproblems* – множество рубежных значений для числа решенных задач в одном сезоне;

- удалось войти в круг решивших *problems* задач в турнире:

pAcc&(*problems-1* ∉ *bTproblems*)& (*problems* ∈ *bTproblems*) →
(*us* ∈ {1,2}, *par* = <*year, rate, problems*>),
где *bTproblems* – множество рубежных значений для числа решенных задач в одном турнире;

- самый большой прогресс за указанный период по указанному рейтингу в чемпионате:

pEndD&(*newval* > *maxProgress*) → (*us* ∈ {0,1}, *par* = <*year, rate, period, newval*),
где *newval* – оценка прогресса рейтинга *rate* за период времени *period*, *maxProgress* – максимальный прогресс данного рейтинга за указанный период, зарегистрированный в данном году;

- решены все задачи в турнире:

pAcc&(*problems* = *allProblems*) → (*us* ∈ {1,2}, *par* = <*year, season, tour*),
где *problems* – число решенных школой или участником задач, *allProblems* – общее число задач в контесте;

- превзойден абсолютный рекорд чемпионата по числу задач, решенных за один год:

pAcc&(*newval* > *oldval*) → (*us* ∈ {1,2}, *par* = <*newval, oldval, oldyear*>),
где *newval* и *oldval* – соответственно достигнутое значение и старый рекорд, *oldyear* – год, в который был установлен старый рекорд;

- превзойден абсолютный рекорд чемпионата по числу задач, решенных за один сезон:

pAcc&(*newval* > *oldval*) → (*us* ∈ {1,2}, *par* = <*newval, oldval, oldyear, oldseason*>),
где *oldyear* и *oldseason* – год и сезон, в которые был установлен старый рекорд;

- превзойден абсолютный рекорд чемпионата по числу задач, решенных за один турнир указанного типа:

pAcc&(*newval* > *oldval*) → (*us* ∈ {1,2}, *par* = < *tour, newval, oldval, oldyear, oldseason*>),
где *oldyear* и *oldseason* – год и сезон, в которые был установлен старый рекорд;

- установлен новый абсолютный рекорд чемпионата по числу задач, решенных за один год:

pEndY&(*newval* > *oldval*) → (*us* ∈ {0,1}, *par* = <*newval, oldval, oldyear*>),
где *newval* и *oldval* – соответственно достигнутое значение и старый рекорд, *oldyear* – год, в который был установлен старый рекорд;

- установлен новый абсолютный рекорд чемпионата по числу задач, решенных за один сезон:

pEndS&(*newval* > *oldval*) → (*us* ∈ {0,1}, *par* = <*newval, oldval, oldyear, oldseason*>),
где *oldyear* и *oldseason* – год и сезон, в которые был установлен старый рекорд;

- установлен новый абсолютный рекорд чемпионата по числу задач, решенных за один турнир указанного типа:

pEndT&(*newval* > *oldval*) → (*us* ∈ {0,1}, *par* = < *tour, newval, oldval, oldyear, oldseason*>),
где *oldyear* и *oldseason* – год и сезон, в которые был установлен старый рекорд;

- перемещение с одного места внутришкольной рейтинговой таблицы на другое в годовом рейтинге:

pAcc&(*newval* < *oldval*) → (*us* = 1, *par* = <*year, rate, newval, oldval*>),

где *newval* и *oldval* задают позиции во внутришкольных турнирных таблицах рейтингов учащихся;

- перемещение с одного места внутришкольной рейтинговой таблицы на другое в сезонном рейтинге:

pAcc&(*newval* < *oldval*) → (*us* = 1, *par* = <*year, season, rate, newval, oldval*>);

- перемещение с одного места внутришкольной рейтинговой таблицы на другое в турнирном рейтинге:

pAcc&(*newval* < *oldval*) → (*us* = 1, *par* = <*year, season, tour, rate, newval, oldval* >);

- по итогам года удалось улучшить место во внутришкольной рейтинговой таблице по рейтингу *rate*:

pEndY& (*newval* < *oldval*) → (*us* = 1, *par* = <*year, rate, newval, oldval*>),

где *newval* и *oldval* задают соответственно позицию в текущей годовой внутришкольной турнирной таблице рейтинга *rate* и лучшее значение этого рейтинга для данного учащегося в предыдущие годы;

- по итогам сезона удалось улучшить место во внутришкольной рейтинговой таблице по рейтингу *rate*:

pEndS& (*newval* < *oldval*) → (*us* = 1, *par* = <*year, season, rate, newval, oldval*>),

где *newval* и *oldval* задают соответственно позицию в текущей сезонной внутришкольной турнирной таблице рейтинга *rate* и лучшее значение этого рейтинга для данного учащегося в предыдущие сезоны;

- по итогам турнира удалось улучшить место во внутришкольной рейтинговой таблице по рейтингу *rate*:

pEndT& (*newval* < *oldval*) → (*us* = 1, *par* = <*year, season, tour, rate, newval, oldval* >),

где *newval* и *oldval* задают соответственно позицию в текущей внутришкольной таблице рейтинга *rate* и лучшее значение этого рейтинга в предыдущие турниры для данного учащегося;

- прохождение в финал сезона:

pEndT&(problems>=*minProblems*) → (*us* ∈ {0,1}, *par* = <*year, season, newval*>),

где предикат *pEndT* идентифицирует завершение отборочного турнира, *minProblems* – минимальное число решенных задач, дающих право прохождения в финал, при *us*=0 *newval* – количество учеников школы, прошедших в финал;

- участие в суперфинале:

pEndT&(*Rs* >=*minRs*) →(*us* = 0, *par* = <*year, season, newval*>),

где предикат *pEndT* идентифицирует завершение последнего финального турнира года, *Rs* – значение рейтинга успешности школы перед

суперфиналом, *minRs* – минимально допустимое значение рейтинга успешности, дающее право прохождения в суперфинал;

- получение диплома степени *newval* в финале сезона:

$pEndT\&(problems>=minProblemsD) \rightarrow (us \in \{0,1\},\ par = <year,\ season,\ newval>)$,

где предикат *pEndT* идентифицирует завершение финального турнира, *minProblemsD* – минимальное число решенных задач, дающих право получения диплома степени *newval*;

- лучший результат участия во всех финалах года:

$pEndT\&(newval = maxSumProblems) \rightarrow (us \in \{0,1\},\ par = < newval>)$,

где предикат *pEndT* идентифицирует завершение последнего финального турнира, *newval* – сумма числа задач, решенных в различных финалах, *maxSumProblems* – максимальная сумма для школ при *us*=0 и участников при *us*=1.

4. Потоки сообщений

Потоки сообщений делятся на внутренние и внешние. Внутренние потоки генерирует анализатор вердиктов для подсистемы визуализации сообщений. Потребность во внутренних потоках обусловлена необходимостью обеспечения производительности генераторов потоков сообщений в условиях, когда на один вердикт проверяющей машины приходятся сотни запросов на публикацию сообщений. Внешние потоки порождаются самой подсистемой визуализации на основе потоков событий и внутренних потоках сообщений.

Потоки сообщений о значениях годовых рейтингов школ материализуются в формате таблицы четырех рейтингов, формируемых на основе потока событий *SS*.

Каждая из этих таблиц является продуктом публикации потока сообщений, имеющего формат последовательности:

$ST = (r_1, r_2, \dots, r_K, s, srt, srs)$,

где $r_1, r_2, \dots, r_K$ - векторы результатов K школ, преодолевших порог вхождения в рейтинг;

s – вектор суммарных результатов тех школ, которые не преодолели порог вхождения в рейтинг;

srt – вектор суммарных результатов всех школ по турнирам и годовым итогам;

srs – вектор суммарных результатов всех сезонов.

Векторы школ формируются из событий потока SS на основе отношения порядка календарных событий начала и конца турниров:

$r_i = <L_i, sname_i, rt_1, rt_2, \dots, rt_M, sy_i, sp_i>$,

где L_i – место i-й школы в рейтинге (“*” для школ, участвующих вне конкурса);

$sname_i$ – имя школы;

rt_j – рейтинг i-й j-го турнира i-й школы;

sy_i – сумма рейтингов школы по всем турнирам года;

sp_i – число задач, решенных учащимися i-й школы за весь год;

Таблица индексов прогрессов школ строится на основе внутреннего потока сообщений о текущем состоянии индексов прогресса:

$SP = (p_1, p_2, \ldots)$,

где p_i – сообщение о состоянии индексов прогресса i-й школы, представляющее собой вектор:

<IDs, Ms, As, Rs, Ps, dMs, wMs, qMs, dAs, wAs, qAs, dRs, wRs, qRs, dPs, wPs, qPs>.

Формализм потока самой таблицы индексов прогресса представляет собой последовательность сообщений *QSP*, элементы которой упорядочены по выражению 10*dAs+3*wAs,+qAs+As.

Для отражения рейтингов и индексов их прогресса участников чемпионата используется внутренний поток сообщений, содержащий векторы:

<IDu, Au, Ru, Pu, LAu, LRu, LPu, dAu, wAu, qAu, dRu, wRu, qRu, dPu, wPu, qPu>.

Поскольку число участников чемпионата может составлять несколько сотен, публикация общей таблицы затруднено. В этой связи рейтинги и индексы их прогресса участников локализуется на внутришкольной странице в формате последовательностей вида:

$URP = (urp_1, urp_2, \ldots)$,

где urpi вектор показателей i-го ученика следующего формата:

<num, fio, Au, LAu, dAu, wAu, qAu, Ru, LRu, dRu, wRu, qRu, Pu, LPu, dPu, wPu, qPu >,

num – номер по порядку, причем стартовый алфавитный порядок может быть изменен на убывающий порядок по любому рейтингу или индексам прогресса;

fio – фамилия и инициалы учащегося.

Формальной моделью потока событий истории достижений школ является последовательность:

$SH = \langle sh_1, sh_2, \ldots, sh_L \rangle$,

где $sh_i = \langle num_i, dt_i, sname_i, text_i \rangle$;

num_i – номер сообщения, равный $L - i + 1$, т.е. используется убывающая нуперация (ретро-порядок);

dt_i – дата-время в соответствии с элементом из потока событий *H*, обработка которого порождает *i*-й элемент потока сообщений;

$sname_i$ – имя школы, к которой относится достижение;

$text_i$ – текст достижения, генерируемый на основе значения полей события из потока *H*.

История достижений отдельной школы является последовательностью векторов, получаемой из *SH* путем выделения сообщения только заданной школы, удаления компонента *sname* и организации локальной нумерации записей с использованием ретро-порядка.

История достижения участников локализуется на школьных страницах и ее формализм представляет собой последовательность, отличающуюся от последовательности *SH* тем, что вместо компонента *sname* в векторах используется компонент *uname,* представляющий фамилию и инициалы учащегося.

Заключение

Формальные представления потоков сообщений являются важной частью проектных решений этапа концептуального проектирования таких автоматизированных систем, которые претерпевают активное эволюционное развитие и в которых содержание сообщений охватывает большое число разнообразны данных, нацеленных на мотивацию пользователей.

В результате частичной реализации проектных решений, представленных выше, в еще не завершенном чемпионате ИТ-сферы Ульяновской области по программированию среди школьников в 2016 году удалось более чем в полтора раза увеличить число решенных задач относительно чемпионата 2015 года. При этом, относительно высокая результативность чемпионата 2015-го года была достигнута за счет активного использования административного ресурса – в школы рассылались письма за подписью первого заместителя председателя правительства Ульяновской области и письма из регионального министерства образования. С начала чемпионата 2016 года оргкомитет чемпионата прекратил эксплуатацию административного ресурса.

Список литературы

1. Негода В.Н. Использование технологий электронного обучения в организации чемпионата области по программированию среди школьников. // Электронное обучение в непрерывном образовании. Сб. трудов III междунар. науч.-практич. конференции. – Ульяновск : УлГТУ, 2016. – с. 335 – 343.
2. Чемпионат ИТ-сферы Ульяновской области по программированию среди школьников. // URL: ulivt.ru

3. Геймификация обучения. Сайт Techyjmatrix: e-Learning Solutions // URL: http://tmx-learning.ru/?p=2987
4. Негода В.Н. Геймификация чемпионата школьников по программированию: формирование динамичной системы рейтингов // Информатика и вычислительная техника. Сб.трудов VIII Российской конференции. – Ульяновск : УлГТУ, 2016. – с. 16 – 23.

УДК 519.688

ИНФОРМАЦИОННАЯ ПОДДЕРЖКА НАУЧНОЙ ДЕЯТЕЛЬНОСТИ ВНУТРИ ВУЗА

В.Г. Тронин[4]

Рассмотрены требования по автоматизации информационной поддержки научной деятельности в университете. Анализ возможностей и недостатков Карты российской науки.

Введение

Повышение эффективности научной деятельности в университете может быть достигнуто за счет автоматизации информационной поддержки процессов связанных с научной деятельностью, упорядочение процессов (реинжиниринг) и повышение доступности необходимой информации всем участникам процесса. Именно база данных позволит эффективно применить наукометрические инструменты внутри организации для стимулирования научных достижений. Еще одна причина для автоматизации – большое количество отчетов запрашиваемых от вуза МинОбрНауки, региональными властями, фондами, общественными организациями, и вузу необходимо обеспечить непротиворечивость подаваемых данных, снизить затраты времени на формирование отчетности.

1. Пользователи данных

Рассмотрим задействованных в цепочке передачи данных участников:

- Сотрудники управления научных исследований: научно-исследовательский отдел, отдел интеллектуальной собственности, экономисты, отдел научной технической информации, отдел аспирантуры и докторантуры. Другие структуры университета на последующих этапах.
- Ученые: руководители научных школ, доктора и кандидаты наук, аспиранты и т.д.
- Ответственные по факультетам, кафедрам за науку, НИРС.
- Руководство университета.

[4] *432027, Ульяновск, ул. Северный Венец, 32, УлГТУ, e-mail:v.tronin@ulstu.ru*

2. Общие требования

Для обеспечения качества и достоверности данных в системе нужно сделать так, чтобы каждый ученый вел самостоятельно профиль и был заинтересован в его корректном заполнении. Это требование может быть реализовано, если рейтинги и дополнительные выплаты ученый получает на основании данных из системы. Самостоятельная забота пользователя о своем резюме успешно реализована в глобальных социальных сетях с миллионами пользователей – Research Gate, LinkedIn, Google Schcolar [4].

В целом по университету можем сформулировать такие требования по информационной поддержке научной деятельности:

- Наличие портфолио всех ученых, фотографии ученых, оборудования, награждений. На основе этих данных персональные странички ученых с доступом через Интернет.
- Контактные данные всех ученых (телефон и электронная почта).
- Регистрация в системе заявок на гранты с ведением статистики выигравших заявок, планирование сумм по научным направлениям и университету в целом.
- Информирование о значимых для ученых мероприятиях, конференциях проводимых внутри вуза и по сторонним приглашениям [6].
- Данные для участия в конкурсах с точки зрения университета в целом (например, у кого из ученых более 5 публикаций в журналах Scopus и Web of Science за последние 5 лет, у кого есть публикации по определенной теме).
- Данные для расчета рейтинга научной активности [9]. Учет специфичных данных: распределение долей нескольких авторов в заявке на грант, распределение долей участия в публикации. Международные системы имеют большой интервал времени до размещения публикаций в наукометрических системах.
- Информация о регистрации ученых в РИНЦ [7], Карте Российской науки, OrcID, ResearcherID, ResearchGate и др.
- Хранение отчетности по грантам РФФИ, Госзаданию и т.д.
- Хранение количественных показателей с расшифровками по отчетам для МинОбрНауки.
- Работа с аспирантами и их научными руководителями.
- Автоматизированное составление различных рейтингов, используя данные внутри системы.
- Доступ к профилям организации в различных системах (РИНЦ, Госзадание, РФФИ, РГНФ, РНФ, Google Shcolar и т.д.).

- Ведение сайта в Интернет с автоматическим обновлением информации по деятельности связанной с наукой.
- Некоторые элементы документооборота между участниками [1, 3].
- Возможность прогнозирования научной активности [8].

3. Требования к системе с точки зрения ученого

Активному ученому приходится готовить много материалов (для публикаций, заявок, отчетов), отслеживать сроки различных мероприятий и планировать свой рабочий график. Для этого ученому важно:

- получать информацию о конкурсах, конференциях, планировать свой график в этой части;
- иметь в доступе информацию для подготовки заявок, аннотации предыдущих заявок;
- иметь информацию для отчетов по грантам;
- иметь доступ с любого места, как в университете, так и удаленно;
- затрачивать минимальное время на внесение информации;
- чтобы университет повторно не требовал данные, которые ученый предоставил ранее;
- получать максимально интегрированный доступ к ресурсам по науке УлГТУ;
- получать доступ к научной периодике, в первую очередь англоязычной.

4. Возможности в Карте Российской науки

Карта российской науки (https://mapofscience.ru), разработана по заказу МинОбрНауки по аналогии с международными системами наукометрического анализа InCites (надстройка Web of Science), SciVal Spotlight (надстройка Scopus). В Карте российской науки задумана максимальная интеграция данных по ученому, позволяющая оценить уровень его достижений и коллаборацию в масштабах России. Показатели из Карты российской науки запрашивались при подготовке заявок по конкурсу Госзадание 2016 года. В современной науке роль научной коллаборации возрастает, и научные социальные сети представляют бесплатный инструмент доступа к публикациям в глобальном масштабе, который следует использовать [2].

В Карте Российской науки реализовано портфолио ученого:

- Перечень журнальных публикаций и книг, выгруженный из РИНЦ.
- Перечень публикаций в изданиях, индексируемых Web of Science.
- Перечень публикаций в изданиях, индексируемых Scopus.
- Перечень докладов конференций Web of Science, Scopus.

- Зарегистрированные в ЦИТИС гранты, НИОКР для руководителя и участников с суммами.
- Перечень патентов из ФИПС и Web of Science.
- Научная деятельность и достижения (образование, ученая степень, ученое звание, награды, членство в профессиональных сообществах, редколлегии журналов).
- Преподавательская деятельность.
- Идентификаторы ученого (ResearcherID, SPIN-код).

Преимущества перед РИНЦ:

- Интеграция с ФИПС по патентам.
- Выгрузка из Web of Science и фактически получение бесплатного доступа к части профиля ученого в Web of Science.
- Попытка очистить РИНЦ от малозначимых публикаций и ссылок за счет импорта только журнальных публикаций и книг.
- Составление карты научного сотрудничества.

Несмотря на то, что Карта российской науки реализована и запущена в эксплуатацию в 2014 году, качество реализации и достоверность данных вызвали волну критики в научных кругах. Рассмотрим некоторые недостатки Карты Российской науки, присутствующие в настоящее время.

При импорте из РИНЦ и других баз обеспечивается охват данных глубиной с 2007 года. Для молодого исследователя это нормально, но для заслуженных ученых результатом становится искажение уровня достижений (например, заниженный индекс Хирша).

Наукометрические индексы в Карте российской науки не соответствуют тем системам, по которым они посчитаны. Например, если указан у ученого индекс Хирша равный 2 в Scopus, то при просмотре средствами Scopus он может составлять 3 или 4.

Удивительно, но выгрузка из РИНЦ может проходить некорректно даже при идентификации ученого по SPIN-коду. В частности, это может случаться, если у исследователя есть публикации с аффиляцией от разных организаций. Таким образом, в Карте российской науки может быть большое количество клонов одного автора, у каждого из клонов часть публикаций реального автора.

Слабо организовано сопровождение по исправлению ошибочных данных в части авторства публикаций, нет регламента по времени внесения исправлений, неэргономичный интерфейс в части внесения изменений.

Странно, но в системе предусмотрен идентификатор автора для Web of Science, а идентификаторы к Scopus и другим системам полезным для интеграции данных отсутствуют.

Нет доступа к самим публикациям (для скачки с самого ресурса или для доступа через интернет) и для исправления некорректного описания публикаций.

Не реализован профиль по управлению учеными по организации в целом. По сравнению с РИНЦ, сопровождение Карты российской науки организовано неудовлетворительно для выполнения поставленных задач, таких как выбор научных коллективов для получения государственного финансирования.

5. Ресурсы университета

Сайт УлГТУ активно используется и занимает в настоящее время 112-ю строчку в России по Webometric [5]. Среди ресурсов, которые активно используются учеными, отметим регулярно обновляемые и актуальные: новостная лента сайта УлГТУ, сайт отдела интеллектуальной собственности (ois.ulstu.ru) и сайт издательства (venec.ulstu.ru).

Заключение

Поддержка научной деятельности в университете должна максимально интегрироваться с Интернет-ресурсами. Реализация поставленных задач может быть разбита на этапы, среди которых наиболее критичным является повышение научной активности молодежи – аспирантов.

Список литературы

1. Радионова Ю.А., Тронин В.Г. Проектирование структуры данных архива электронных документов НПО с использованием средств MS SQL SERVER 2000 // Автоматизация процессов управления. 2008. № 2. С. 119-123.
2. Сафиуллин А., Тронин В. Сотрудничество и научно-исследовательские сети в экономике знаний // Проблемы теории и практики управления. 2015. № 2. С. 104-111.
3. Селяев А.Г., Тронин В.Г. Системы автоматизации документооборота // Автоматизация процессов управления. 2004. № 1. С. 34-36.
4. Тронин В.Г. Профессиональные социальные сети LinkedIn и академия Google // Вестник Ульяновского государственного технического университета. 2015. № 3 (71). С. 34-36.
5. Тронин В.Г. Рейтинг сайта УлГТУ по Webometric и возможности улучшения результата // Вестник Ульяновского государственного технического университета. 2013. № 4 (64). С. 54-58.
6. Тронин В.Г. Современные возможности представления научных мероприятий в электронных ресурсах // Электронное обучение в непрерывном образовании. 2016. № 1 (3). С. 400-407.
7. Тронин В.Г. Факторы, влияющие на рейтинг УлГТУ в РИНЦ / В сборнике: Вузовская наука в современных условиях Сборник материалов 48-й научно-технической конференции в 3-ех частях. 2014. С. 185-188.

8. Тронин В.Г., Галныкина К.С., Стенина А.С. Оценка возможностей прогнозирования данных в РИНЦ с применением сервиса анализа временных рядов // Вестник Ульяновского государственного технического университета. 2014. № 3 (67). С. 40-44.
9. Ярушкина Н.Г., Тронин В.Г. Рейтинги оценки деятельности инженерного вуза // Высшее образование в России. 2014. № 5. С. 72-79.

ЧАСТЬ II

МАТЕРИАЛЫ ДОКЛАДОВ УЧАСТНИКОВ ШКОЛЫ-СЕМИНАРА

УДК 621.372

АНАЛИЗ ВЛИЯНИЯ ИЗМЕНЕНИЙ В РАЗРАБОТКЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ.

А.Е.Абрамов

Так как компоненты программного обеспечения занимают все большую область разработки ПО, появляется все большая зависимость компонентов между друг другом, следовательно, каждое изменение несет за собой все больше последствий. Если раньше существовали методики отслеживания последствий изменений, то на текущий момент эти инструменты не успевают адаптироваться к новым технологиям. Системы растут по размеру и сложности, в данной статье я проведу исследование возможности отслеживания результатов изменений в текущих программных системах.

Введение

В последнее время программное обеспечение на практике все чаще должно охватывать полный спектр операционных систем, соответственно каждое изменение несет за собой все больше последствий, поэтому необходимо уметь проводить анализ последствия изменений.

Например, когда заменяется компонент, то внутренние данные, зависимости также заменяются. Таким образом, проектирование программных систем фокусируется на интерфейсах, а не конкретной реализации. На сегодняшний день веб-сервисы являются отличной платформой для комплексного исследования последствий изменений.

Существующие модели программного обеспечения меняют модель анализа последствия изменений. Программные системы все чаще используют промежуточное ПО и различные паттерны проектирования. Это частично позволяет избежать сложности поддержки исходного кода за счет использования общепринятых конструкций. К тому же, ПО растет в размерах и количестве зависимостей до таких масштабов, что один человек уже не в состоянии охватить весь спектр применяемых технологий и используемых исходных кодов. По этой причине, ошибки в работе ПО часто остаются незаметными до тех пор, пока не произойдет системный сбой. Последствия зависят от многих факторов и появляются в системе во время разработки.

Анализ влияния изменения ПО определяет возможные последствия предлагаемых изменений. К настоящему моменту, он все еще не включен в процесс разработки во множестве компаний, однако, некоторые производители стараются уже внедрять некоторые средства для анализа и это дает положительный результат.

Анализ влияния изменений – это трудоемкий, часто ручной процесс, выполняется только когда это абсолютно необходимо из-за связанных с этим расходов, эффективно контролируя качество, количество и последовательность изменений, которые могут быть внесены в ПО.

Изменение ПО

Ключом к эффективному изменению ПО является полное понимание принципов работы ПО. В отличие от многих других продуктов, ПО должно быть максимально пластичным. Всегда должна быть предусмотрена возможность расширения и добавления дополнительного функционала в максимально короткие сроки с меньшими последствиями. Хотя ПО и не устаревает со временем, любое его изменение потенциально влечет за собой ухудшение продукта, если не контролировать эти изменения. В последнее время все больше растет потребности в поддержке существующих крупных систем, нежели создании новых. Поэтому возникла потребности в анализе влияния изменений.

В основном, изменение требований к существующему ПО ограничено некоторыми рамками, требования к системе постоянно меняются. Если система полностью удовлетворяет требованиям, то считается, что требования устарели. Основными видами изменения ПО являются: анализ вносимых изменений в контексте системы, внедрение их в существующую систему и повторное тестирование.

Каждый из этих видов деятельности по-своему влияет на систему:

- для анализа вносимых изменений, нужно определить те части системы, которые будут затронуты, изучить возможные последствия от изменений;
- во время внедрения изменений нужно не упускать из виду возможные последствия и следить за их появлением;
- после внедрения изменений, необходимо провести полный анализ системы и выявить те места, которые должны будут быть покрыты дополнительными тестами в соответствии с новыми требованиями.

Анализ последствия изменения ПО

Анализ последствия изменения ПО -- понимается как «определение потенциальных последствий изменений исследуемой системы в результате предлагаемого изменения программного обеспечения.

Процесс анализа воздействий является итеративным и раскрывается в процессе. Чем больше изменений будет внедрено, тем больше произойдет последствий. Основным принципом, лежащим в основе анализа влияния изменения, является то, что даже небольшое изменение в системе может повлечь за собой серьезные последствия или так называемые, «волновые последствия». Многие последствия могут быть неочевидными, и их будет не так легко обнаружить. Те же «волновые последствия» могут оказывать как прямое, так и косвенные воздействие.

- Прямое воздействие – возникает, когда затронутый объект связан напрямую с текущим модулем системы. Этот тип воздействия также называется «влиянием первого уровня» и может быть найден из графа матрицы зависимостей.
- Косвенное воздействие – возникает, когда затронутый объект связан с набором других объектов. Его также можно назвать «влиянием N-ого уровня», где N – количество промежуточных связей между объектами. Этот объект может быть получен из достижимости графа зависимостей.

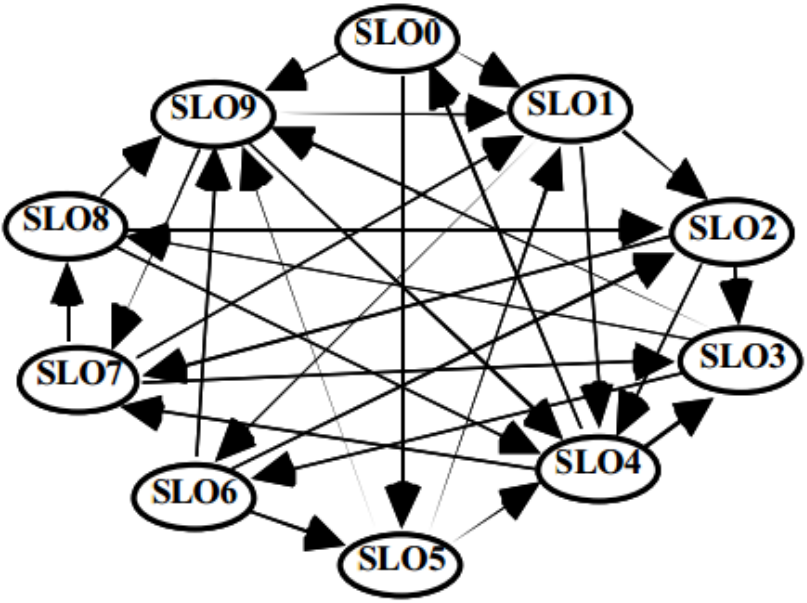


Рисунок 1. Граф зависимостей.

Анализ последствий изменения ПО фокусируется на структуре и семантике. Во-первых, анализируется структурный уровень, на рис. 1 показан граф зависимостей 10 программных компонентов, каждый из

которых имеет зависимости. Каждая из стрелок показывает зависимость одного компонента от другого.

Структурный анализ.

Стоит отметить, что SLO1 имеет прямую зависимость от SLO9 и косвенную от SLO7. Необходимо построить граф зависимостей.

Таблица 1. Матрица отношений.

SLO=>	0	1	2	3	4	5	6	7	8	9
SLO0		X				X				X
SLO1			X		X		X			
SLO2				X	X			X		
SLO3							X		X	X
SLO4	X			X				X		
SLO5		X			X					X
SLO6			X			X				X
SLO7		X		X					X	
SLO8			X		X					X
SLO9		X			X			X		

Данная матрица (табл. 1) может быть легко преобразована в граф достижимости (табл. 2). По этому графу можно будет легко определить, какой из компонентов системы будет затронут изменением. Однако, такой простой способ обнаружения будет часто выдавать ложноположительные результаты.

Таблица 2. Граф достижимости.

SLO=>	0	1	2	3	4	5	6	7	8	9
SLO0		X	X	X	X	X	X	X	X	X
SLO1	X		X	X	X	X	X	X	X	X
SLO2	X	X		X	X	X	X	X	X	X
SLO3	X	X	X		X	X	X	X	X	X
SLO4	X	X	X	X		X	X	X	X	X
SLO5	X	X	X	X	X		X	X	X	X
SLO6	X	X	X	X	X	X		X	X	X
SLO7	X	X	X	X	X	X	X		X	X
SLO8	X	X	X	X	X	X	X	X		X
SLO9	X	X	X	X	X	X	X	X	X	

Одним из способов ограничения количества ложноположительных результатов является добавление дополнительной характеристики. Косвенное воздействие предполагает, что существует расстояние между компонентами. В табл. 3 показано отношением между компонентами с учетом их расстояний.

Таблица 3. Матрица отношения с расстоянием.

SLO=>	0	1	2	3	4	5	6	7	8	9
SLO0		1	2	3	2	1	2	2	3	1
SLO1	2		1	2	1	2	1	2	3	2
SLO2	2	2		1	1	3	2	1	2	2
SLO3	3	2	2		2	2	1	2	1	1
SLO4	1	2	3	1		2	2	1	2	2
SLO5	2	1	2	2	1		2	2	3	1
SLO6	3	2	1	2	2	1		2	3	1
SLO7	3	1	2	1	2	3	2		1	2
SLO8	3	2	1	2	1	3	3	2		1
SLO9	2	1	2	2	1	3	2	1	2	

Это нововведение позволяет нам отбрасывать малозначительные вероятности при поиске.

Семантика в анализе влияния изменений

Только при анализе зависимостей с большим количеством компонентов необходимо использовать дополнительную семантическую информацию для повышения точности обнаружения потенциальных изменений и сокращения числа ложных срабатываний. С тех пор, как объектно-ориентированное программирование переносит объекты из реального мира в разработку программного обеспечения, то имеет смысл использовать это свойство для определения влияния изменений в объектно-ориентированных системах.

В настоящее время, существует три проблемы, которые делают анализ влияния изменений сложным:

1. Объем информации – существует большое количество источников, которые необходимо проанализировать.
2. Измененная семантика -- методы для описания важных изменений ограничены областью действия компонента.
3. Методы анализа – методы анализа программных зависимостей и отладки продуктов не до конца изучены для быстрорастущих сфер ПО (таких как веб-сервисы, BigData и т.д.).

Таким образом, было выяснено, что из себя представляет анализ влияния изменений, принципы изменения ПО, проведен структурный анализ последствий возможных изменений на абстрактном примере, выявлены основные проблемы анализа влияния изменений.

Заключение

Подведя итоги, можно перечислить следующие преимущества внедрения анализа изменений в повседневную работу:

- Эффективность тестирования и его качество повышается: функции, которые должны быть протестированы не потеряются, становится меньше неожиданных сюрпризов на этапе эксплуатации.
- Время тестирования также уменьшается: тестировщик не будет наугад выбирать слабые места системы, а будет их тестировать целенаправленно.
- Увеличиваются обязанности разработчиков: теперь им перед отправкой изменений приходится определять места в системе, которые могут быть затронуты.
- Также растет общее понимание всей структуры проекта у разработчиков и тестировщиков.

К тому же, используя данную методику, можно выявлять проблемы еще на стадии проектирования вносимых изменений, предугадывать слабые места и устранять их, теряя при этом минимальное количество ресурсов и получая максимальное качество готового продукта.

Список литературы

1. Alimadadi S., Mesbah A., Pattabiraman K. Hybrid DOM-sensitive change impact analysis for JavaScript //LIPIcs-Leibniz International Proceedings in Informatics. – Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2015. – Т. 37.
2. Bohner S. A. Software change impact analysis. – 1996.
3. Porter A. L. et al. Guidebook for technology assessment and impact analysis. – 1980.

УДК 621.372

ОПИСАНИЕ ПРОЦЕССА ПРОЕКТИРОВАНИЯ ЗАДАЧ ДЛЯ СИСТЕМЫ J-MARK.

А.Е. Абрамов

В статье излагается описание процесса проектирования задач для системы J-MARK. Рассмотрен анализ коэффициента покрытия для задачи, рассмотрен рейтинг значимости каждого из элементов интерфейса при составлении оценки, составлена схема жизненного цикла состояния заданий в системе.

Введение

В последнее время компьютеризация все сильнее проникает во все сферы нашей жизни. Особенно в тех областях, где можно четко выделить некоторый алгоритм действий. Одной из таких областей является оценка алгоритмических результатов. Там присутствует четкая детерминированная последовательность действий (алгоритм) и конечный результат, который можно в большинстве случае соотнести с эталонным.

Существуют различные виды олимпиад: индивидуальные, командные и т. п., но в большинстве случаев во всех них можно выделить некоторые стадии:

1. участник решает задачи в строго отведенное время;
2. результат выполнения передается жюри;
3. жюри проверяет правильность решения на наборе тестов;
4. жюри выставляет оценку и формируется рейтинговая таблица.

Все эти действия может производить и обученный человек, но такая система обладает рядом преимуществ, а именно:

- скорость проверки задач;
- автоматизация проверки;
- область покрытия знаниями;
- отсутствие субъективного фактора;
- более точные результаты за счет того, что на проверку одинаковых задач уходит одинаковое время;
- возможность проведения турниров онлайн;

- возможность использования ее как системы проверки домашних заданий.

Актуальность данной идеи обусловлена тем, что в последнее время растет количество проводимых турниров по программированию в целом среди школьников и студентов. Турниры позволяют школам и университетам показывать работодателям навыки подготавливаемых специалистов, участникам – применить навыки на практике в нестандартных ситуациях, возможность получить уважение среди других участников и преподавателей.

Коэффициент покрытия

Для примера, разберем одно из заданий ИТ-многоборья в номинации «разработка диалоговых сценариев при помощи Swing». Необходимо создать приложение при помощи диалог-дизайнера Swing на языке Java (см. рис. 1).

Для не строгой оценки разрешается произвольный выбор местоположения.

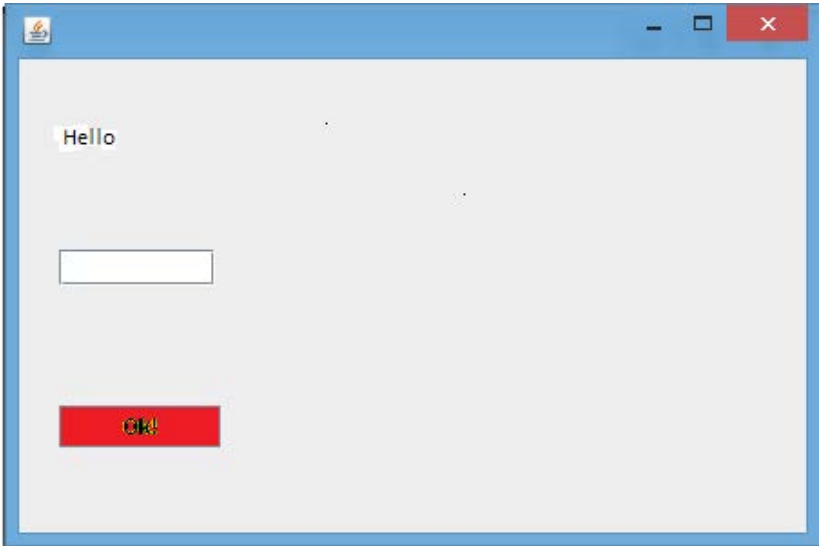


Рисунок 1. Пример заданного диалогового сценария.

Рассмотрим упрощенный вариант концептуальной таблицы предметной области. В данном случае концептами являются классы технологии Swing.

Таблица 4. Концептуальная таблица предметной области.

индекс	Связь / Концепт	setText	setFont	setBackground	setForeground	setVisible	setEnable	setValue	setToolTipText
1	jButton	+	+	+	+	+	+	-	+
2	JScrollPane	-	-	-	-	+	+	-	+
3	JRadioButton	+	+	+	+	+	+	-	+
4	JCheckBox	+	+	+	+	+	+	-	+
5	JLabel	+	+	-	+	+	+	-	+
6	JTextField	+	+	+	+	+	+	-	+
…	…								
34	JRadioMenuButton	+	+	+	+	+	+	-	+

Рассмотрим, насколько это задание наполнено знанием. Введем коэффициент покрытия задания ТЗ.

1. Так как расположение элементов в видимой части приложения не влияет на результат, то следуя задаче №1, наличие кнопки типа JButton с именем «btn» и цветом «red» нам дает:
   ТЗ = ТЗ + 2

2. Экзепляр JLabel с именем «Hello» и setText(“Hello”) дает:
   ТЗ = ТЗ + 2

3. Экземпляр JTextFIeld с именем «log» и setVisible(true) дает:
   ТЗ = ТЗ + 2

Следовательно, ТЗ = 6 после выполнения всех проверок.

Т – коэффициент знаний, посчитанный по таблице. Т = 34.

$$\frac{6}{34} * 100 = 17\ \%$$

Таким образом, данное задание покрывает лишь 17% знаний предметной области. Зачем это нужно? Это необходимо при создании новых заданий и улучшения предыдущих, ведь для объективной оценки знаний тестируемого, необходимо оценивать полный спектр знаний технологии.

Рейтинг важности элементов

Механизм оценивания базируется на сложности реализации элемента, количестве его свойств, которые возможно проверить. От количества свойств также растет условная сложность проверяемого задания, что напрямую влияет на расчет итоговой оценки в турнирной таблице.

Таблица 5. Рейтинг важности элементов.

Индекс	Элемент	Степень
1	jButton	5
2	JScrollPane	7
3	JRadioButton	5
4	JCheckBox	3
5	JLabel	5
6	JTextField	5
7	JPasswordField	6
8	JFormattedTextField	7
9	JTextArea	7
10	JTextPane	7

Жизненный цикл заданий в системе

Основополагающим принципом составления и последующей корректировки задания является принцип, схожий с принципом работы планировщика задач в системе. По итогам турнира задачи, которые решило малое количество участников, отправляются на доработку человеку, отвечающему за составление заданий. Он должен выяснить причину, по которой задание осталось нерешенным. Не должно быть ситуаций, когда в системе присутствуют задачи, которые никто не может решить из-за некорректной формулировки, сложность в реализации или по другим причинам. Таким образом, можно выделить следующие состояния задания в системе:

- **в ожидании:** задание полностью готово к решению. Ожидает выбора участником.
- **решается:** задание решается участником.
- **на доработке:** задание дорабатывается администратором.

- **заблокирована:** задание не может быть выбрана участником для решения.

Используя данный механизм, можно позволить корректировать задания в ходе турнира, а не только после его завершения. Схематично отобразить данный процесс можно следующим образом:

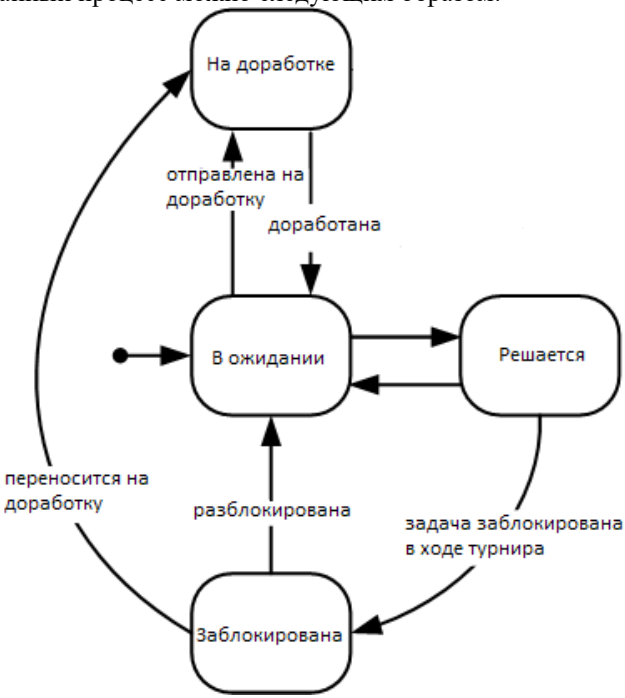


Рисунок 2. Жизненный цикл задания в системе.

Таким образом, реализация отдельного модуля для работы с заданиями позволит достигнуть большей гибкости системы, реализовать возможность поддержки системы в реальном времени во время проведения турниров.

Список литературы

1. Герберт Шилдт. SWING: руководство для начинающих = SWING: A BEGINNER'S GUIDE. — М.: «Вильямс», 2007. — С. 704.
2. М. Руссинович, Д. Соломон. Внутреннее устройство Microsoft Windows, 6-е издание (Часть 1). — СПб.: «Питер», 2013. — С. 800.

УДК 621.372

Анализ и управление рисками программного обеспечения

Аль- Хусеейни Ханса Азиз

Программное обеспечение для управления рисками является выявление, оценка и управление проектными рисками. Выявленные риски анализируются с целью определения их потенциального воздействия и вероятности возникновения. Планы управления рисками разработаны с целью документирования подхода проекта к управлению рисками, риски и решения, о том, что должно быть сделано с каждым риском.

Введение

Риск программного обеспечения включает в себя вероятность появления неопределенных событий и их потенциал для потери внутри организации. Управление рисками стало важным компонентом разработки программного обеспечения, как организации продолжают внедрять больше приложений на многостраничного технологии, многоярусной среды. Как правило, программное обеспечение риск рассматривается как комбинация надежности, эффективности производительности, безопасности и транзакционной риска, распространяющейся по всей системе.

Риск является ожидание потери, потенциальная проблема, которая может или не может произойти в будущем. Это, как правило, возникает из-за недостатка информации, контроля или времени. А возможности страдания от потери в процессе разработки программного обеспечения называется риск программного обеспечения. Потеря может быть что угодно, увеличение стоимости производства, качества разработки программного обеспечения, не будучи в состоянии завершить проект вовремя. Риск программного обеспечения существует, потому что будущее остается неопределенным, и есть много известных и неизвестных вещей, которые не могут быть включены в план проекта. Риск программное обеспечение может быть двух типов [1] внутренние риски, которые находятся под контролем руководителя проекта и [2] внешних рисков, которые находятся вне контроля менеджера проекта.

1 Управление рисками Программное обеспечение

Риск является возможность потери. Это зависит как от вероятности неблагоприятных событий, происходящих и ее последствий; воздействие проявляется в сочетании финансовых потерь, время задержки и потери производительности. Риск является предшественником к проблеме; вероятность того, что в любой заданной точке жизненного цикла программного обеспечения, предсказанные цели не могут быть достигнуты в рамках имеющихся ресурсов. Риск не может быть исключена из проекта программного обеспечения, но им можно управлять. Управление рисками имеет решающее значение для успеха любых усилий программного обеспечения и является стратегическим аспектом всех программных проектов.

Проект программного обеспечения могут столкнуться с различными видами рисков:

Технические риски включают в себя проблемы, связанные с языками, размер проекта, функциональные возможности проекта, платформы, методы, стандарты или процессы. Эти риски могут возникнуть в результате чрезмерных ограничений, отсутствие опыта, плохо Определенные параметры, или зависимостей от организаций за пределами непосредственного контроля проектной команды.

Риски управления включают отсутствие планирования, отсутствие опыта управления и обучения, коммуникации проблемы, организационные вопросы, отсутствие полномочий, а также проблемы управления.

Риски персонала включают в себя кадровые лаги, опыт и учебные проблемы, этические и моральные проблемы, персонал конфликты и проблемы производительности.

Другие риски ресурсов включают недоступность или несвоевременной поставки оборудования и расходных материалов, неадекватных инструментов и временем реакции.

Управление рисками Программное обеспечение является инженерная практика программного обеспечения с процессами, методы и инструменты для управления рисками в проекте.

Определяет четыре причины для осуществления управления рисками программного обеспечения:

1. Как избежать катастроф проекта программного обеспечения, в том числе убегают бюджеты и графики, дефект охваченном программное обеспечение продукты и операционные сбои.
2. Как избежать переделок, вызванных ошибочными, отсутствует или неоднозначных требований, проектирования или кода, которые, как

правило, потребляет 40-50% от общей стоимости разработки программного обеспечения.

3. Как избежать излишеством с методами обнаружения и предотвращения в районах минимальной или вообще без риска.
4. Стимулирование программное решение беспроигрышной, где клиент получает продукт они нуждаются, и продавец делает прибыль они ожидают.

2 Управление рисками программного обеспечения с использованием программы WIQA Client

2.1 WIQA Client

Необходимо выделить причинно-следственное представление рисков и представить его в вопросно-ответной среде моделирования с учетом возможности накопления опыта и создания базы знаний рисков проектов. Также необходимо использовать существующие решения и инструменты системы WIQA.

В результате статьи проектом были выполнены поставленные задачи и спроектирована система управления рисками в вопросно-ответной среде моделирования проектной деятельности WIQA.

Существует несколько различных методик управления рисками при проектировании. Выбор определенной методики зависит от ситуации и типа проекта, но вне зависимости от самого проекта риски.

2.2 Профиль риска

Результатом процесса анализа риска, может быть использован для получения профиля риска, который дает оценку значимости для каждого риска и предоставляет инструмент для определения приоритетов усилий по лечению риска. Это входит в каждый выявленный риск, с тем чтобы дать представление об относительной важности.

Этот процесс позволяет риск быть отображен в бизнес-зоне воздействия, описываются процедуры первичного управления на месте, и указывает на области, в которых может быть увеличен уровень инвестиций управления рисками, снижение или reapportioned. Отчетность помогает гарантировать, что «право собственности» риска признается и соответствующий ресурс управления, выделенный.

Профиль риска – совокупность сведений об области риска, индикаторах риска, а также указания о применении необходимых мер по предотвращению или минимизации рисков . Профиль риска представляет собой перечень вопросов, которые касаются традиционных областей неопределенности по проекту .

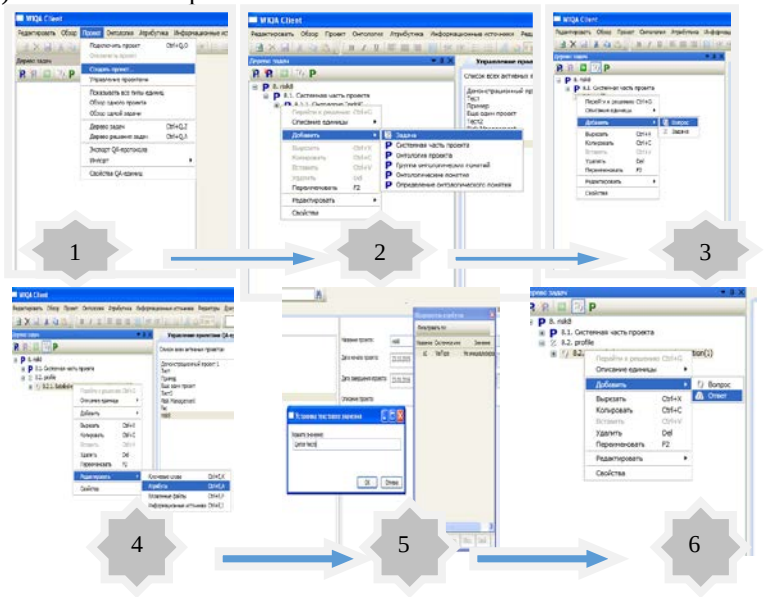
Профиль риска представляет собой перечень вопросов, которые касаются традиционных областей неопределенности по проекту.

Таблица 1. Профиль риска вопросы

Технические требования : требования стабильны?	Качественный: соображения качества воздуха, встроенный в дизайн?
Дизайн :делает дизайн зависит от нереальных или оптимистическим предположениям?	управление :Знаете ли люди, кто имеет власть для чего?
Тестирование :Хорошо испытательное оборудование доступны, когда это необходимо?	производственная среда :Работают люди совместно через функциональные границы?
Развитие :Поддерживается ли процесс развития с помощью совместимого набора процедур, методов и инструментов?	Кадровые "Является ли персонал неопытен или недоукомплектованы?
График :Является ли график зависит от завершения других проектов?	Клиент :Понимает ли клиент, что потребуется для завершения проекта?
Бюджет :Насколько надежны смета расходов?	Подрядчики :Существуют ли какие-либо неясности в контрактах определений?

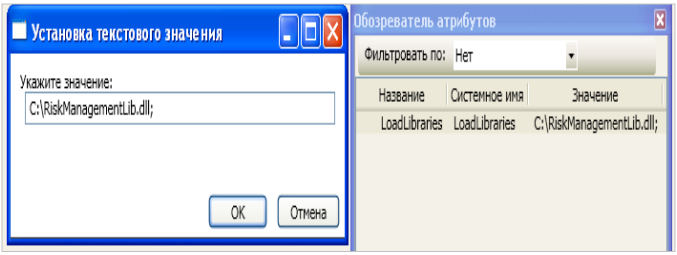
Создание проекта в программе WIQA выберите :-

1) Создание нового проекта . проект ⟶ создать проект .
2) Создать задачу .
3) Добавить вопрос к задаче .
4) Введите имя атрибута и введите имя атрибута системы .
5) Определяя величину .
6) Ответ на вопрос .

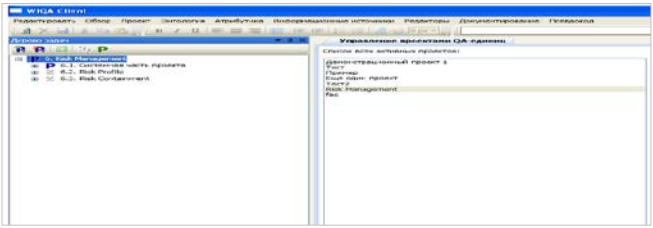


Язык программирования на C # (Профиль риска и риска Сдерживание) .

- Вызов (class dll) к программе .



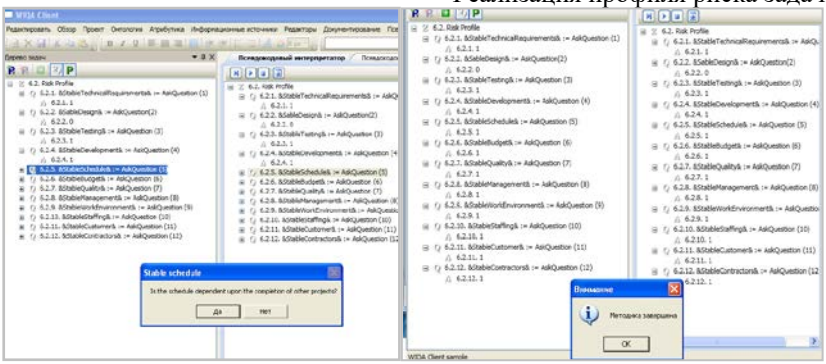
- Откройте Управление рисками проекта и определить Профиль риска задачей.



Результаты профиля риска Вопросы следующим образом В программе WIQA: -
выполнить задание выбрать:-

- псевдокод ⟶ запустить интерпретатор
- В ходе реализации будет отвечать да или нет вопросов по типу риска.

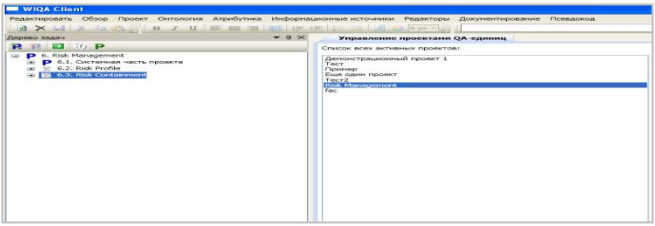
Реализация профиля риска задачи



2.3 Сдерживание риска

После того, как оцениваются все выявленные риски, планы должны быть сделаны, чтобы контролировать самый разрушительный и наиболее вероятные риски. Стратегии, используемые для сдерживания риска:
Избегайте риска :-
- Обсудите с клиентом, чтобы уменьшить объем работы.
- Предоставление стимулов для инженеров, чтобы избежать риска оборота рабочей силы.
Передать риска :-
-Получение Рискованные компоненты, разработанные третьей стороной.
- Страхование Покупка обложка.
Сокращение рисков :-
- Планирование способов контроля повреждения из-за риска .
- Если есть риск того, что может оставить некоторые ключевые кадры, новый набор может быть запланировано.

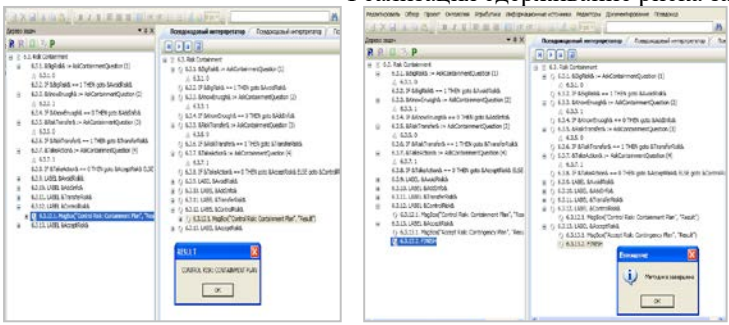
- выбрать второй Сдерживание риска задачей .



выполнить задание выбрать :-

- Псевдокод ⟶ Запустить Интерпретатор
- В ходе реализации будет отвечать да или нет вопросов по типу риска.

Реализация сдерживание риска задачи



Заключение

Есть много рисков, связанных с созданием высококачественное программное обеспечение в установленные сроки и в рамках бюджета. Тем не менее, для того, чтобы быть полезным, чтобы взять эти риски, они должны быть компенсированы воспринятого вознаграждения. Чем больше риск, тем больше награда должна быть, чтобы сделать его стоит рискнуть. При разработке программного обеспечения, возможность вознаграждения высока, но так есть потенциал для катастрофы.

Выделить причинно-следственное представление рисков и представить его в вопросно-ответной среде моделирования Концептуальное решение задач с учетом возможности накопления опыта и создания базы знаний рисков проектов. Также необходимо использовать существующие решения и инструменты системы WIQA.

Помогают менеджеру проекта разрабатывать меры реагирования на риск, основываясь на более достоверной информации. Достоверная и более подробная информация позволяет принимать более эффективные решения, которые в свою очередь снижают возможность возникновения риска. Произведение вероятности и воздействия определяет важность риска - его ценность - показатель, который может использоваться в процессе принятия решения.

Список литературы

1. Соснин П.И. Псевдо-кодовое программное управление потоками работ в проектировании автоматизированных систем. 2012.
2. Allison Robin. MSF Risk Management Discipline v.1.1, 2002.
3. Зайцева Л. В. Методы и модели адаптации к учащимся в системах компьютерного обучения // Educational Technology & Society.— no. 4.
4. Котов В. Е. Сети Петри.— М.: Наука, 2013.
5. Курганская Г. С. Модели, методы и технология дифференцированного обучения на базе Интернет.— М. Вильямс, 2011.
6. http://iso27000.ru/chitalnyi-zai/upravlenie-riskami-informacionnoi-bezopasnosti/kak-upravlyat-riskami-informacionnoi-bezopasnosti .

УДК 004

АНАЛИЗ УСПЕШНОСТИ ПОДГОТОВКИ ПРОИЗВОДСТВА ПРОЕКТНОЙ ОРГАНИЗАЦИИ[5]

А.Н. Афанасьев[6], Н.Н. Войт[7], М.Е. Уханова, И.С. Ионова

Авторами выполнено исследование степени успешности разработки сложных автоматизированных систем на крупных проектных организациях с учетом результатов исследований лидеров на рынке исследований и аналитических услуг, таких как Gartner, Forrester Research и Standish Group.

Проблематика

Одной из ключевых задач при подготовке производства является управление потоками проектных работ. Руководителю приходится согласовывать планы со смежниками, планировать деятельность своих подчиненных, а также быть задействованным в планах других исполнителей. Для организации обмена заданиями и контроля выполнения работ в конструкторско-технологических подразделениях создается схема информационных потоков предприятия. В эту схему (шаблон будущего процесса) вводятся участники (конструкторы, технологи, нормировщики, начальники групп и отделов, главный конструктор и т.д.). Для них назначаются задания, сроки их выполнения и направления движения информации.

После запуска проектных процессов каждый участник в установленный срок получает необходимые исходные данные (чертежи, результаты расчетов и т.д.), выполняет свою часть работы и передает задание следующему участнику проектного процесса. При этом ответственные исполнители могут контролировать ход процесса и автоматически получать уведомления о нарушении исполнителями срока выполнения здания.

Проблема успешности выполнения проектных задач предприятия исследуется более 25 лет и вызвана высоким выходом разработок за рамки

[5] *Исследование выполнено при финансовой поддержке РФФИ в рамках научного проекта № 15-07-08268 а.*

[6] *432027, Ульяновск, ул. Северный Венец, 32, УлГТУ, e-mail: a.afanasev@ulstu.ru*

[7] *432027, Ульяновск, ул. Северный Венец, 32, УлГТУ, e-mail: n.voit@ulstu.ru*

запланированных временных, финансовых и функциональных параметров. Выявлены причины и выработаны рекомендации по повышению успешности проектирования сложных автоматизированных систем (САС), однако, по данным компаний Gartner, Forrester Research и Standish Group, в настоящее время лишь 40% разработок заканчиваются успешно [1, 2].

В достижении успешности разработки САС большую роль играют проектные потоки работ, представленные в диаграмматическом базисе с использованием визуальных языков UML, IDEF, ER, DFD, eEPC, SDL, BPMN, SharePoint и др. Использование таких моделей повышает как эффективность процесса проектирования, так и качества проектных решений за счет унификации средств взаимодействия заказчиков и проектировщиков, строгого документирования проектно-архитектурных, функциональных решений и формального контроля корректности диаграммных нотаций [3-10].

В части создания оперативных потоков работ в настоящее время все больше доминирует парадигма, связанная с динамическим и распределенным характером их сущности. Динамика определяется необходимостью немедленного реагирования на возникающие производственные запросы. Распределенность определяется как разработка моделей не только различными удаленными группами проектировщиком, но и часто с использованием различных диаграмматических базисов (например, UML и IDEF).

Большое значение эффективного использования диаграмматических моделей потоков работ имеют задачи реинжиниринга САС, также обладающие свойствами динамичности и распределенности.

В настоящее время существует проблема успешности разработки сложных автоматизированных систем. Суммарное представление статистики с 2011 года по 2015 год приведено в таблице 1, из которой понятно, что доля завершенных вовремя и с планируемым бюджетом проектов мало.

Таблица 1. Статистика успешности выполнения проектных задач на предприятиях [1]

Доля в процентах					Результат
29	27	31	28	29	Успех
22	17	19	17	19	Провал
49	56	50	55	52	Изменение
2011	2012	2013	2014	2015	год

«Успех» в статистике ставится проекту, если он выполнен за запланированное время и стоимость.

«Провал» в статистике ставится проекту, если он не выполнен.

«Изменение» в статистике ставится проекту, если он выполнен, но были изменения времени и стоимости.

Отчеты Xaoca Standish Group были опубликованы ежегодно с 1994 и обеспечивают снимок состояния разработок программного обеспечения. Отчет 2015 года базируется на 50 000 проектах, ранжированных от улучшения маленьких проектов до крупных системных реализаций реинжиниринга.

Каскадная модель управления проектами [11]

Каскадная модель управления проектами в виде фаз работ называется диаграммами Ганта, по имени их создателя Генри Ганта. С распространением в 1980-е годы персональных компьютеров стало проще создавать разные затейливые диаграммы – и делать их по-настоящему комплексными. Весь ход проекта детально размечен: выделены шаги, стадии, поставлены даты (рис. 1). Диаграммы Ганта производят глубокое впечатление. Существует лишь единственная проблема: они всегда неправильны – без исключений.

Рис. 1. Модель диаграммы Ганта

Генри Гант придумал свои знаменитые диаграммы в 1910 году. Впервые ими начал пользоваться генерал Уильям Крузер, начальник артиллерийско-технической службы вооруженных сил США в Первую мировую войну. Каждый, кто изучал историю этой войны, знает, что ни подготовка кадровых ресурсов, ни система организации никогда не была ее сильными сторонами.

На самом деле все выглядит невероятно заманчиво, когда работа, которую предстоит выполнить по крупному проекту, детально изображена графически и представлена на всеобщее обозрение. Беда лишь в том, что как только это превосходно отточенный план сталкивается с реальностью, он сразу рассыпается в прах. Но вместо того чтобы выбросить в мусорную

корзину и сам план, и свой подход к нему, руководители делают вид, будто план работает, и даже нанимают для этого специалистов. По сути, они платят людям за то, что те им лгут.

Модель SCRUM для управления проектами [11, 12]

В основе методологии Scum лежит простая идея. Когда бы не был запущен проект, нам никто не мешает регулярно проверять ход работ и последовательно выяснять: справляетесь ли вы с заданием; в нужном ли направлении движетесь; создаете ли именно то, что на самом деле хочет получить заказчик. Вам также ничто не мешает постоянно поднимать следующие вопросы: есть ли способы усовершенствовать методы разработки и выполнять работу наиболее качественно и быстро; существуют ли факторы, препятствующие вашим задачам.

Описанный процесс называется «проверять и адаптировать». Иначе говоря, в любой подходящий момент и как можно чаще следует прерывать сиюминутную работу, пересматривать то, что уже сделано, и выяснять, все ли сделано, что нужно, и как это выполнить лучше. Конечным результатом применения методологии Scrum – является команды, наглядно увеличивающие свою производительность.

Результаты бывают настолько впечатляющими, что, по мнению лидеров рынка исследований и аналитических услуг, таких как Gartner, Forrester Research, Standish Group, Whitestien, испытанный подход себя изживает. Компании, которые продолжают упорно пользоваться старыми, но отнюдь не добрыми концепциями контроля и управления, опирающимися только на прогнозирование, обречены на поражение в битве с конкурентами, перешедшими на методику Scrum.

В статье, опубликованной в Harvard Business Review в 1986 году двумя японскими преподавателями экономики – Хиротака Такеучи и Икуджиро Нонакой «The New New Product Development Game» («Разработка нового продукта. Новые правила игры»), проанализирована инновационная деятельность крупнейших транснациональных компаний, таких как Honda, Fuji-Xerox, 3M, Hewlett-Packard и др. Компании утверждали, что традиционный последовательный подход к работе над проектом – так называемый каскадный тип процесса разработки, – в основе которого лежит система поэтапного планирования программ НАСА[8], безнадежно устарел. Ведущие компании мира, отказавшись от линейной модели, перешли на метод параллельных процессов разработки, который требовал

[8] НАСА (NASA) – аббревиатура от National Aeronautics and Space Administration (Национальное управление по воздухоплаванию и исследованию космического пространства).

скорости и гибкости исполнения. Многофункциональные проектные группы обладали полной автономностью и свободой принимать самостоятельно решения. Целеустремленность людей была связана с чем-то больше, чем просто с личным интересом. Они старались выйти за границы собственных возможностей. Над ними не довлел тотальный контроль руководства. Специалистам никто не диктовал, что делать и как поступать при разработке нового проекта. Напротив, управляющие высшего звена относились к своему лидерству как к служению, поскольку считали, что их основная задача – быть помощником и устранять с пути своих команд все препятствия.

Согласно старому подходу, процесс разработки продукта двигался как эстафета. Группы специалистов последовательно передавали друг другу эстафетную палочку. Разработка продукта шла последовательно, от фазы к фазе: разработка концепции, проверка осуществимости, дизайн продукта, процесс разработки, пилотный продукт и финальный продукт. Согласно этому методу функции были специализированы и сегментированы. Маркетинг исследовал потребности клиентов и их мнение о концепции продукта. R&D инженеры выбирали подходящий дизайн. Инженеры производства создавали реальный продукт. Разные специалисты несли эстафетную палочку на разных этапах гонки.

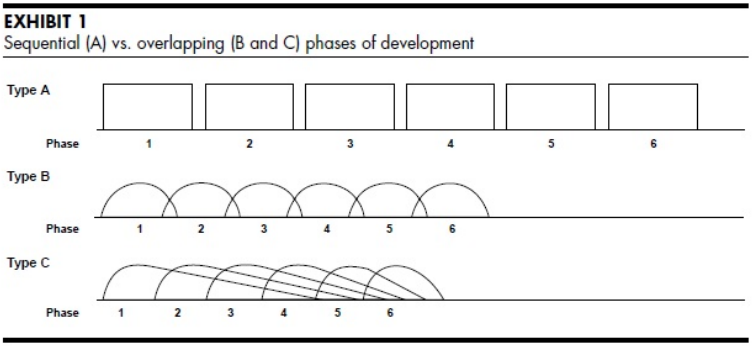


Рис. 2. Потоки проектных работ по разработке сложных автоматизированных систем

Согласно подходу регби, процесс разработки продукта возникает из постоянного взаимодействия многопрофильной команды, члены которой работают вместе от начала и до конца. Вместо того, чтобы двигаться по определенным, хорошо структурированным стадиям, процесс рождается из взаимодействия членов команды (рис. 2). Группа инженеров, например,

может начать делать дизайн продукта (фаза три) еще до того, как готовы все результаты тестов на осуществимость. Или команда может быть вынуждена пересмотреть решение на основе более поздней информации. Команда не останавливается, а участвует в итеративном экспериментировании. Так продолжается даже на поздних стадиях процесса разработки.

Рис. 2 иллюстрирует различие между традиционным, линейным, подходом и подходом регби. Тип А – последовательный подход, он воплощен в системе PPP (NASA). Тип В – подход с нахлестами, перекрытие происходит только на границах фаз. Тип С – подход, в котором перекрытие проходит сразу через несколько фаз. Подход с нахлестом типа В используется в Fuji-Xerox. Подход с нахлестом типа С используется в Honda и Canon.

Такой подход важен для компаний, которые хотят быстро и гибко разрабатывать новые продукты. Переход от линейного подхода к интегральному поощряет путь проб и ошибок и бросает вызов существующему положению вещей. Он стимулирует новые виды обучения и мышления внутри организации на разных уровнях и в различных областях. Не менее важно, что эта стратегия разработки продукта может выступать фактором перемен в крупной организации. Энергия и мотивация усилий распространяются по компании и начинают разрушать устаревшие порядки.

Компании из Японии и Соединенных Штатов, которые используют новый подход для управления процессом разработки, являются транснациональными компанияи (Fuji-Xerox, Canon, Honda, NEC, Epson, Brother, 3M, Xerox и Hewlett-Packard), которые применяли новых подход к разработке своих продуктов: FX-3500 среднего ксерокса (введен Fuji-Xerox в 1978 г.); PC-10 персонального ксерокса (Canon, 1982); Городского автомобиля с 1200 кубовым двигателем (Honda, 1981); Персонального компьютера PC 8000 (NEC, 1979); Однообъективной зеркальной камеры AE-1 (Canon, 1976); Камеры Auto Boy, известной как Sure Shot в США (Canon, 1979).

Выбор перечисленных продуктов основывался на их влиянии, на использовании нового процесса разработки, новизне функций продукта, успехе на рынке, и на доступности данных по этим продуктам.

Заключение

В теории и практике коллективного применения диаграмматических моделей потоков работ в задачах проектирования, сопровождения и реинжиниринга САС отсутствуют эффективные в смысле полноты классов обнаруживаемых ошибок методы и средства

анализа и контроля диаграмматических представлений проектных процессов САС, что приводит к серьезным ошибкам проектирования. Решение этой задачи является актуальной фундаментальной научно-технической проблемой.

Список литературы

1. Отчет Standish Group об успешности разработки программного обеспечения. The Standish Group's Chaos Report. http://www.morphis-insights.com/bi-for-software-development/
2. Charette, R.N. Why software falls / R.N. Charette // IEEE Spectrum, Vol. 42 (9), 2005, pp. 36-43. http://ieeexplore.ieee.org/document/7338527/
3. Афанасьев А.Н., Войт Н.Н. Интеллектуальная агентная система анализа моделей потоков проектных работ // Автоматизация процессов управления. – 2015. – № 4 (42). – С. 42-61. http://elibrary.ru/item.asp?id=25144484.
4. Афанасьев А.Н., Войт Н.Н. Разработка и исследование средств извлечения из САПР КОМПАС-3D и представления в веб-системах конструкторского описания, 3D-моделей промышленных деталей и сборок //Системы проектирования, технологической подготовки производства и управления этапами жизненного цикла промышленного продукта (CAD/CAM/PDM - 2015): труды международной конференции. Под ред. А.В. Толока. – 2015. – С. 208-212. http://elibrary.ru/item.asp?id=25421518.
5. Афанасьев А.Н., Игонин А.Г., Афанасьева Т.В., Войт Н.Н. Использование нейросемантических сетей для автоматизированного проектирования вычислительной техники // Автоматизация. Современные технологии. – 2008. – № 1. – С. 21-24.
6. Афанасьев А.Н., Войт Н.Н. Разработка компонентно-сервисной платформы обучения: диаграммы использования и деятельности программного компонента сценария на UML-языке // Вестник Ульяновского государственного технического университета. –2012. – № 1 (57). – С. 66-68.
7. Афанасьев А.Н., Войт Н.Н. Реализация конструктора сценария обучающих курсов // Вестник Ульяновского государственного технического университета. – 2011. – № 1 (53). – С. 54-59.
8. Alexander Afanasyev, Nikolay Voit Intelligent Agent System to Analysis Manufacturing Process Models // Proceedings of the First International Scientific Conference «Intelligent Information Technologies for Industry» (IITI'16) Volume 451 of the series Advances in Intelligent Systems and Computing. – Russia, 2016 – Pp. 395-403. http://link.springer.com/chapter/10.1007/978-3-319-338163_39.
9. Alexander Afanasyev, Nikolay Voit, Rinat Gainullin The analysis of diagrammatic models of workflows in design of the complex automated systems // Proceedings of International conference on Fuzzy Logic and Intelligent Technologies in Nuclear Science - FLINS2106. – France, 2016. – Pp. 509-514. http://flins2016.ensait.fr
10. Alexander Afanasyev and Nikolay Voit Multi-agent system to analyse manufacturing process models // Proceedings of International conference on Fuzzy Logic and Intelligent Technologies in Nuclear Science - FLINS2106. – France, 2016. – Pp. 444-449. http://flins2016.ensait.fr

11. Scrum. Революционный метод управления проектами / Джефф Сазерленд ; пер. в англ. М. Гескиной – М. : Манн, Иванов и Фербер, 2016. – 288 с.
12. Разработка нового продукта. Новые правила игры. – The New New Product Development Game. http://agilerussia.ru/methodologies/разработка-нового-продукта-новые-пра/

УДК 004.89

О МЕТОДОЛОГИИ НЕЧЕТКОГО ПОДХОДА К АНАЛИЗУ ОРГАНИЗАЦИОННО-ТЕХНИЧЕСКИХ СИСТЕМ[9]

Т.В. Афанасьева[10]

В данной статье описан подход к исследованию состояний и поведения процессов организационно-технических систем (ОТС) на основе нечеткого моделирования. Предлагается и обосновывается необходимость разработки языка нечеткого моделирования в терминах нечетких оценок и грамматики, для которой выделены определены основные элементы.

Введение

Эксплуатация и управление современным предприятием выявили проблемы, решить которые можно лишь на основе комплексного анализа качества различных по своей природе факторов, разнородных связей, внешних условий и внутренних процессов.

Моделью современных предприятий могут служить организационно-технические системы, в которых выделены элементы трех классов (технические, информационные и гуманистические) и имеется множество связей между ними.

Под организационно-технической системой (ОТС), согласно [1], будем понимать прикладную систему, в которой организационная составляющая (персонал, клиенты) и техническая составляющая (оборудование, сети и т.п.) взаимодействуют посредством информационно обеспечения (баз данных, баз знаний, веб-сервисы) для достижения поставленных целей.

В работе [2] приводится определение ОТС в виде сложных многоуровневых систем, структурными элементами которых являются

[9] *Работа выполнена при поддержке РФФИ, проект № 16-07-00535*

[10] *432027, Ульяновск, ул. Северный Венец, 32, УлГТУ, e-mail:tv.afanasjeva@gmail.com*

люди и технические средства/устройства, осуществляющие преобразование ресурсов этой системы.

Проблема неопределенности ОТС

Проблема неопределенности, обусловленная внутренней и внешней нестационарностью, в том числе, значительной ролью человека (гуманистического элемента), является неотъемлемой принадлежностью сложных организационно-технических систем и на практике характеризуется:

- невозможностью или ограниченной возможностью экспериментального исследования процессов, не позволяющей получить достаточную и достоверную статистическую информацию;
- субъективной неопределенностью из-за многоцелевого, многокритериального, экспертного оценивания информации;
- неоднородностью, неточностью информации.

Применительно к анализу процессов в ОТС эти неопределенности выражены в:

1. *гетерогенности значений характеристик процессов* – количественном и качественном значении, неточности значений ВР;

2.*неопределенности в количестве наблюдений* – недостаточности наблюдений (7-60 значений). Согласно концепции информационного подхода к измерению неопределенности неопределенность убывает с увеличением наблюдений;

3.*неопределеннсти поведения* процессов – нестационарном поведении процессов, нечеткости и нестабильности тенденций.

При этом для многих предметных областей, таких как экономика, энергетика, ИКТ, образование, наука, социология, в которых статистика, формируемая по результатам мониторинга деятельности, собирается через большие временные интервалы, именно анализ таких процессов представляет практический интерес.

Традиционный путь учета факторов и оценивания неопределенности на основе вероятностного и статистического моделирования, зачастую оказывается неадекватным решаемым задачам и может привести к неверным результатам.

Одной из фундаментальных проблем является изучение свойств ОТС путем применения методов моделирования с последующим анализом ее компонент и функционирования (поведения).

Вследствие сложности задача анализа ОТС может быть представлена на разных уровнях абстракции на основе принципов

грануляции, множественности описания и интеграции моделей для достижения лучшего понимания природы и поведения системы.

Нечеткое моделирование оценок ОТС

Для моделирования и анализа поведения сложных ОТС в условиях неопределенности получили широкое распространение нечеткие модели и методы, оперирующие качественными нечеткими оценками, основанными на опыте и знаниях эксперта (или группы экспертов).

Известны два направления логики оценки: логика абсолютной оценки(например, «хорошо», «плохо», «безразлично») и логика сравнительной оценки (например, «лучше, чем», «больше, чем», «стабильно»). Каждое из указанных направлений основывается на введении нескольких уровней качества проранжированных в порядке возрастания сложности рассматриваемых свойств. Границы выделенных уровней качества зачастую размыты, нечетки.

Известны модели и методы, позволяющие формализовать получение качественной оценки в виде комплексной числовой оценки состояний, содержащей информацию о текущем состоянии процесса, например, на основе теории эффективности и функций полезности. При этом рассматривают эффективность как показатель процесса функционирования системы. Под эффективностью процесса понимается комплексное свойство, отражающее соответствие исхода операции поставленной цели по выбранному критерию эффективности.

В то же время из практики известно, что для принятия решений удобнее пользоваться не столько количественными, сколько качественными оценками, которые используют понятия теории нечетких множеств. Качественные оценки являются результатом экспертного оценивания. При этом вывод результата может быть представлен в виде: «1.Качественная лингвистическая оценка i-того параметра» – «2.Выбор методов свертки» – «3.Свертка оценок по всем параметрам в лингвистическую оценку».

Для получения стратегических оценок развития ОТС строят прогнозные модели, в которых используют не только оценку текущей ситуации (локальную оценку), но и историю ее возникновения, то есть предшествующий ей процесс, описанный на основе временных рядов.

Известны подходы к анализу организационно-технических систем, в основе которых лежат нечеткие когнитивные карты (Maikel Leon, Kimmo Raivio, А.В. Боженюк, Л.А. Гинис), нечеткие полумарковские модели (В.В. Борисов, Ю.Г. Бояринов), нечеткие игровые модели (В.В.Круглов). Нечеткие когнитивные карты, хорошо зарекомендовали себя в задачах

исследования структуры модели систем и получения прогнозов ее развития при различных управляющих воздействиях. Однако они требуют задания отношений для выражения прич инно-следственных связей между концептами и не решают комплекса задач анализа состояния поведения организационно - технических систем, в частности классификации и прогнозирования гранул локальных тенденций. Нечеткие полумарковские модели для анализа систем и процессов являются расширением полумарковского процесса на основе введения нечеткости во входные данные, в частности для фаззификации вероятностей пребывания системы в различных состояниях, следует отметить, что не всегда имеется возможность получить значения матрицы вероятностей, что является ограничением для их применения для решения задачи прогнозирования поведения системы в случае недостаточности данных. При моделировании оптимальных решений в условиях конфликтов и оптимизации образовании коалиций актуальны нечеткие игровые модели. Однако в этих моделях недостаточно развиты средства долгосрочного прогнозирования. В то же время еще в 1973 году L. Zadeh опубликовал статью о применении нечетких моделей для анализа организационно-технических систем, в дальнейшем нечеткие модели получили применение для поддержки принятия решений в условии неопределенности, в экспертных системах, при решении задач классификации, кластеризации, прогнозирования, однако, проблема комплексного анализа организационно-технических систем, включающая анализ нечетких гранул на основе нечетких моделей не получила достаточного развития как в отечественных, так и в зарубежных исследованиях. В частности, в исследованиях слабо представлен анализ эффективности применения разных классов нечетких моделей к прогнозированию различных классов систем. Другая задача, требующая своего решения - это поиск периода повторения гранул локальных тенденций и обнаружение аномалий.

Для анализа процесса, состояния которого выражены нечеткими значениями и распределены на упорядоченной временной шкале, естественный интерес представляет использование нечетких временных рядов*(см.:Ярушкина, Н.Г. Основы теории нечетких и гибридных систем, 2004).* Интерпретируя значения короткого числового временного ряда или прогнозные результаты, полученные для длинных временных рядов, лингвистическими терминами, представляется возможным перейти от чисел к нечетким значениям и, затем, к нечетким временным рядам.

При этом *обнаружение тенденций(краткосрочных, среднесрочных)* на основе временных рядов характеристик поведения элементов ОТС

выступает как отдельная задача анализа, которая приобретает особую актуальность для слабо-структурированных процессов.

Все это определяет современные требования со стороны пользователей к системам анализа временных рядов:

- Анализ тенденций ВР, обладающих высокой степенью неопределенности.
- Быстрое и простое моделирование, не требующее специальной математической подготовленности пользователя.
- Учет в моделях предметно-ориентированных знаний.
- Лингвистическая интерпретация результатов.

Под комплексной оценкой поведения ОТС будет пониматься некоторая лингвистическая характеристика, генерируемая на основе нечеткого моделирования поведения элементов ОТС и связей между ними. Поведение элементов определяется характером тенденций временного ряда их признаков, хранимых в базе данных. При этом классы поведения временных рядов могут быть различными, что требует применения разнотипных моделей и их сочетаний.

Следует отметить, что при развитии нечеткого моделирования к отдельным объектам организационно-технических систем отсутствует единый комплексный подход к решению актуальных прикладных задач Data Mining на основе алгоритмов и моделей нечеткого подхода.

Сформулируем основную задачу нечеткого моделирования ОТС, как задачу структурно-параметрической идентификации модели ОТС, порождающую проблему определения языка представления структуры и параметров этой модели. Так как значения могут быть представимы значениями лингвистической переменной, то естественно представить структуру процессов в ОТС как цепочку семантически определенных лингвистических элементов, терминальных символов V_T грамматики некоторого языка. Отношения следования(предшествования) между этими термами, обнаруживаемые в процессах элементов ОТС, фиксируют синтаксические правила, правила подстановки P. Промежуточные структуры, составленные из последовательности термов, образуют нетерминальные символы V_N искомой грамматики, которые на основе свертки порождают аксиому грамматики S.

Параметрами рассматриваемой структурной модели могут выступать нечеткие ограничения, представимые функциями принадлежности, а также и числовые значения.

Таким образом, при рассмотрении проблемы определения языка для представления модели ОТС интерес представляет исследование и решение следующих задач:

1.Возможно ли свести задачу структурной идентификации процессов ОТС к задаче определения синтаксиса, грамматики некоторого языка?

2. К какому типу будет принадлежать такая грамматика?

3. Как определить семантические правила «вычисления» новых значений?

4. Какие информационные технологии и математические методы наиболее эффективны при реализации языка представления модели поведения ОТС?

Сформулированная выше совокупность проблем-задач позволяют обозначить необходимость применение нечеткого моделирования в решении задачи анализа состояния и поведения ОТС, как задачи структурно-параметрической идентификации, выраженной в форме нечетких оценок и зависимостей с последующей семантической интерпретацией.

Заключение

В статье приведено описание свойств и задач моделирования ОТС. Обосновано применение нечеткого моделирования, как инструмента качественного оценивания состояния и поведения ОТС на различных уровнях. При этом предлагается для анализа ОТС в рамках нечеткого моделирования разработать язык моделирования и определены принципы и абстрактные элементы такого языка.

Список литературы

1. Запорожцев А.В. Принципы проектирования организационно-технических систем // Машиностроение и автоматизация. – Труды Нижегородского государственного технического университета им. Р.Е. Алексеева. – № 3 (100), 2013. – с. 106-115.
2. Аверченков В.И. Автоматизация проектирования комплексных систем защиты информации: монография/ В.И. Аверченков, М.Ю. Рытов, О.М. Голембиовская – Брянск: БГТУ, 2012. – 139с.

УДК 004.89

ОБНАРУЖЕНИЕ АНОМАЛЬНЫХ СОСТОЯНИЙ В ПОВЕДЕНИИ ПРОЦЕССОВ В ОРГАНИЗАЦИОННО-ТЕХНИЧЕСКИХ СИСТЕМАХ[11]

Т.В. Афанасьева[12], Д.В. Заварзин[13], А.П. Курлов[14]

В данной статье описан подход к исследованию аномальных состояний процессов организационно-технических систем (ОТС) на основе анализа временных рядов (ВР). Анализ ВР здесь предлагается осуществлять при помощи метода скользящих окон и изменения дисперсии ВР.

Введение

В современном мире одно из важных мест занимают специальные сложные программно-технические комплексы, которые помогают человеку в решении широкого класса практических задач. Какой бы вид деятельности не рассматривался, использование таких систем имеет место быть:

1. Медицина и здравоохранение.
2. Безопасность и защита программного обеспечения.
3. Финансы и экономика.
4. Политическая и социальная инженерия и т.д.

Множество взаимосвязанных материальных объектов (технических средств и персонала, обеспечивающего их функционирование и применение по назначению), предназначенных для непосредственного выполнения операции в рамках функционирования такого программного-технического комплекса – называются организационно-техническими системами (ОТС) [1].

В процессе проектирования, внедрения и мониторинга работы систем требуется непрерывный анализ данных и их показателей. Под анализом

[11] *Работа выполнена при поддержке РФФИ, проект № 14-07-00247*

[12] *432027, Ульяновск, ул. Северный Венец, 32, УлГТУ, e-mail:dzavarzin91@gmail.com*

[13] *432027, Ульяновск, ул. Северный Венец, 32, УлГТУ, e-mail: tv.afanasjeva@gmail.com*

[14] *432027, Ульяновск, ул. Северный Венец, 32, УлГТУ, e-mail: kreb.lak@mail.ru*

данных здесь подразумевается процесс обнаружения сведений в крупных наборах данных, поступающих с датчиков ОТС. В настоящее время широко применяется математический анализ данных процессов для выявления закономерностей и тенденций их протекания. Некоторые закономерности невозможно обнаружить традиционными методами анализа данных (например, при визуальном анализе), поскольку процесс может быть слишком сложным и многофакторным, а данные могут быть большими и неоднородным.

Существует ряд закономерностей, которые характеризуют поведение процесса в случаях, когда правила функционирования ОТС нарушаются, (например, происходит сбой программного обеспечения и т.д.). Такие ситуации характеризуют аномальное поведение процесса, а наборы данных, полученные в ходе мониторинга протекания процесса, содержат аномальные измерения. Такие измерения явно выделяются из набора данных, к которому они принадлежат, и существенно отличаются от других элементов выборки.

Как правило, современные математические методы в рамках анализа измерений процессов опираются на временные ряды (ВР) [2].

В данной работе предлагается новый и актуальный алгоритм для поиска аномалий в процессах, выраженных временными рядами. Подход основан на анализе временного ряда комбинацией метода скользящего окна и анализе дисперсии. Кроме того в рамках данной работы была выполнена классификация аномалий на ВР.

В первой главе работы рассматриваются существующие методы идентификации аномальных состояний процессов. Во второй главе приводится математическое описание метода определения аномальных состояний, а также описание программного продукта для определения аномальных состояний. В четвертой главе описывается эксперимент.

1. Обзор существующих методов идентификации аномальных состояний

Поиск аномалий это обнаружение на временном ряду таких значений, которые бы резко отличались от остальных измерений выборки (рис.1). Поиск аномалий сейчас очень актуален, так он находит применение в различных областях деятельности, таких как медицины, геологии, информационных технологии и т.д. [3, с. 60]. Например, поддержание качества оборудования за счёт своевременного обнаружения дефекта в работе какой-то его части.

В общем случае будут рассмотрены следующие виды аномалий:

- Аномалии, превышающие верхнюю границу нормальных значений. Под верхней границей понимается максимально

возможное значение в данном временном интервале. Все точки, которые находятся за пределами данного порога, являются аномальными (Рисунок 1). Ярким примером таких аномалий может служить резкое повышение температуры в зимний период.

- Аномалии, превышающие нижнюю границу. Аналогичны предыдущей аномалии, но в данном случае определяются значения, которые вышли за предел нижнего порога значений.
- Внутренняя аномалия. В данный класс входят все аномалии, которые находятся в границах допустимых значений, но резко отличаются от всех других или соседних значений.

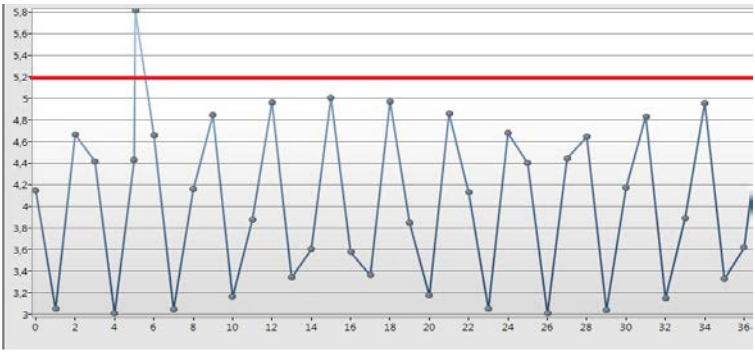


Рис. 1. Схема распределенной обработки

Своевременное выявление отклонений, что может позволить сократить убытки при коммерческой деятельности, спасти жизни людей в медицине, сократить ущерб, который может быть нанесён стратегически важным объектам в результате природных явлений, обеспечить информационные системы программой обнаружения сетевых атак. Моделирование потенциальных угроз для их дальнейшей классификации с целью быстрого определения алгоритма действий при их обнаружении.

2. Описание подхода

В данном разделе будут рассмотрены математические основы алгоритма генерации случайных аномалий. Процесс генерации делится на две основные функции: генерация внутренних аномалий и генерация аномалий выходящих за пределы интервала.

В данном методе происходит сравнение нормального ряда и аномального для выявления аномалии. Из-за наличия нормального ряда, данный метод относится к типу «с учителем», так как для сравнения системе необходимо предоставить пример для решения поставленной

задачи. С помощью метода скользящего окна задаётся размер окна и шаг. Окно осуществляет проход сразу по двум временным рядам: нормальному временному ряду, как идеальному ряду, и ряду с внедрённой аномалией [4, с.19]. В каждом окне считается среднее значение:

$$\overline{x} = \frac{\sum_{i=1}^{n} x_i}{n} \qquad (1)$$

где, x_i - значение признака в данном окне, $i = 1,...,n$; n – количество значений в данном окне.

После нахождения среднего значения, в каждом окне считается среднеквадратическое отклонение по формуле:

$$\sigma = \sqrt{D} = \sqrt{\sigma^2} = \sqrt{\frac{\sum_{i=1}^{n}(x_i - \tilde{x})^2}{n}} \qquad (2)$$

где, x_i - значение признака в данном окне, $i = 1,...,n$; n – количество значений в данном окне; $\tilde{x}$ - среднее значение в данном окне;

Условие нахождения аномалии в окне будет характеризоваться следующей формулой:

$$\sigma_{anom} - 0,05 * \sigma_{norm} > \sigma_{norm} \qquad (3)$$

где σ_{anom} - значение дисперсии аномального ВР, σ_{norm} - значение дисперсии нормального ВР.

Данное окно признаётся аномальным, и все его точки добавляются в аномальный участок. В случае если для следующего окна также выполняется условие, и оно содержит часть тех же значений, что и предыдущее окно, его точки не добавляются в аномалии, так как эта аномалия уже была учтена.

Система должна предоставлять пользователю следующие основные функции: создание нормального временного ряда; моделирование аномалий; поиск аномалий.

Эти функции составляют основу программного обеспечения. К ним также предъявляются следующие функциональные требования:

1.Для моделирования аномалий должен предоставляться набор функций, на основе которых строится исходный ВР (линейную, квадратическую, синуса, косинуса, логарифмическую, экспоненциальную функции). Данные функции характеризуются 4 коэффициентами.

2.Обеспечить возможность создания временного ряда посредством загрузки данных из табличного процессора Excel и генерации ряда на

основе следующих параметров, которые являются исходными данными для данной функции:

3.Количество точек, на основе которых строится временной ряд.

4.Шаг – временной интервал между соседними точками.

В качестве выходных данных программа должна визуализировать таблицу с координатами временного ряда и график, построенным по этим координатам.

3. Описание эксперимента

Создаётся временной ряд со следующими параметрами (рис. 2):

- Функция: квадратическая;
- Коэффициент a равен трём;
- Коэффициент b равен двум;
- Коэффициент c равен единице;
- Количество точек равно сто.
- Шаг равен единице.

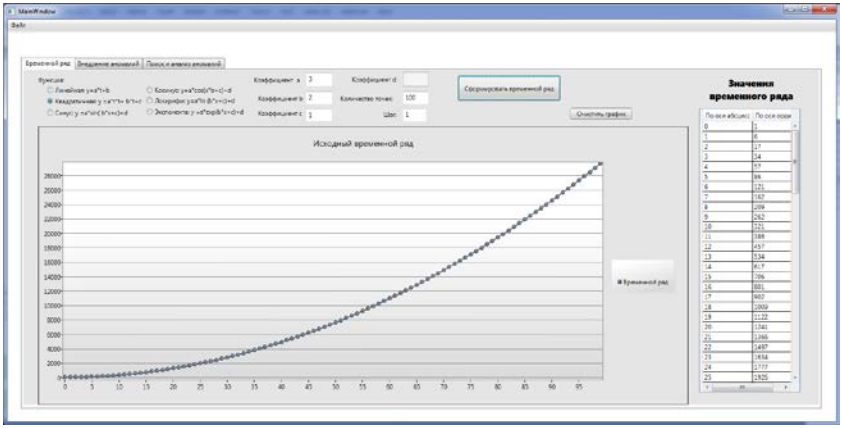


Рис. 2. График исходного временного ряда

Случайным образом внедряется одна из трех аномалий (рис. 3):

- вид аномалии: превышение нижней границы; количество равно единице; период аномальности: с трёх до пяти;
- вид аномалии: превышение верхней границы; количество равно единице; период аномальности: с шести до семи;

- Вид аномалии: внутренняя аномалия; количество равно единице; период аномальности: с 40 до 46.

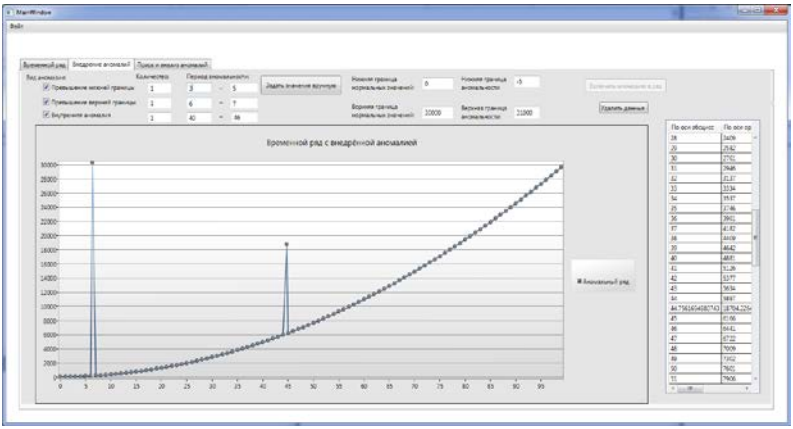


Рис. 3. Временной ряд с внедренными аномалиями

В результате сгенерировано три аномалии:

- Превышение нижней границы: координаты (4.75,-1.77);
- Превышение верхней границы: координаты (6.34,30264);
- Внутренняя аномалия: (44.75, 18704);

Далее осуществляется поиск аномалий (рис. 4):

- Длина окна равна четырём
- Шаг равен единице;

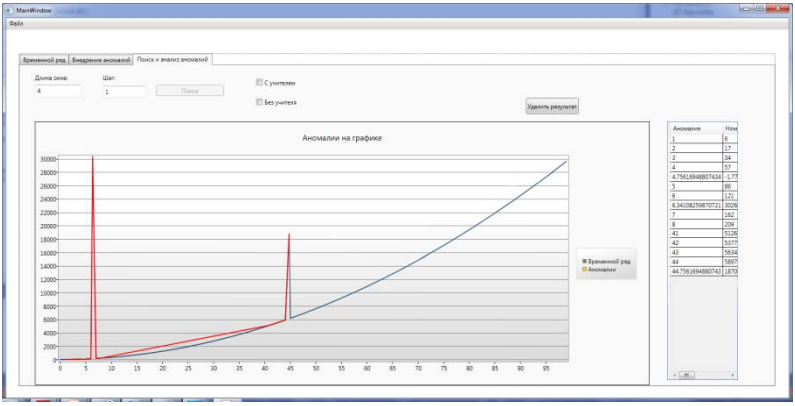


Рис. 4. График временного ряда с отмеченным окном, в котором найдены аномалии

В результате эксперимента найдено три аномальных окна: первое: 1 – 4,75; второе: 5 – 8; третье: 41 – 44,75;

Заключение

В данной статье был предложена система для исследования аномальных состояний в процессах, протекающих в ОТС. В рамках разработки ПО был создан модуль генерации аномалий на ВР, а также модуль поиска аномалий на ВР, путем анализа его методом скользящего окна. Также был выработан критерий аномальности анализируемого окна на основе его дисперсии.

В результате эксперимента было установлено, что предложенный алгоритм успешно справляется с поставленной задачей поиска одиночных аномалий (выбросов).

Дальнейшая работа будет связана с анализом групповых, а не одиночных аномалий в процессах, протекающих в условиях неопределенности.

Список литературы

1. Запорожцев А.В. Принципы проектирования организационно-технических систем // Труды Нижегородского государственного технического университета им. Р.Е. Алексеева. – № 3(100). – с.106-115.
2. Афанасьева Т.В., Ярушкина Н. Г. Нечеткий динамический процесс с нечеткими тенденциями в анализе временных рядов // Вестник Ростовского государственного университета путей сообщения. – 2011. – №3. – С. 7-16.
3. Яшков Д. Д. Проблема понижения размерности в задаче поиска аномалий в многомерных временных рядах [диссертация], МФТИ (ГУ), 2014.
4. Deepthi Cheboli. Anomaly Detection of Time Series. Facility Of The Graduate School Of The University Of Minnesota. — 2010. — 75 с. — [Электронный ресурс] — Режим доступа. — URL : http://conservancy.umn.edu/bitstream/handle/11299/92985/?sequence=1.

УДК 004.946

РАЗРАБОТКА И ИССЛЕДОВАНИЕ ВИРТУАЛЬНЫХ РАБОЧИХ МЕСТ В СРЕДЕ OPENSIM[15]

С.И. Бочков[16]

В данной статье рассматриваются виртуальные миры как инструмент компьютеризации учебного процесса. Разработка проведена с помощью виртуальной платформы OpenSimulator (сокр. OpenSim), что позволило добавлять в виртуальное пространство многокомпонентные модели объектов реального рабочего места – инструментов, комплектующих.

Введение

Промышленность часто сталкивается с необходимостью набора новых кадров. Зачастую это только что закончившие ВУЗ молодые специалисты, едва ли имеющие опыт работы с дорогостоящим оборудованием, что порождает побочный эффект в виде опасных факторов производства или вероятности вывода из строя. Можно проводить обучение с помощью опытного мастера, но ему нужно доплачивать за каждого ученика, что в нынешних условиях финансирования весьма затратно. Поэтому возникает потребность в виртуальных тренажёрах, позволяющих получить представление о технологических процессах на предприятии и пройти практику перед допуском к реальному рабочему месту.

Используемые с 1990 в армии и медицине впервые виртуальные миры привлекли внимание масс в 2003, когда Linden Lab запустила знаменитый виртуальный мир Second Life [7].

Сегодня виртуальные миры широко применяются в медицине, где поле для разработок весьма обширно – с помощью виртуальных программ тренируются молодые врачи, отрабатываются командные операции. Европейское Космическое агентство работает над созданием системы дополненной реальности для обучения космонавтов оказания помощи и

[15] *Исследование выполнено при финансовой поддержке РФФИ в рамках научного проекта № 15-07-08268 а.*

[16] *432027, Ульяновск, ул. Северный Венец, 32, УлГТУ, e-mail: bochkovsam1@rambler.ru*

диагностики заболеваний друг друга в космосе. Прототип системы прошёл успешные испытания в больнице университета Сан-Пьер в Брюсселе. [2]

1. О платформе OpenSim.

OpenSim представляет собой серверную платформу для создания многопользовательских трёхмерных виртуальных миров. Подключённые к серверу пользователи представлены в виде своих 3-х мерных виртуальных образов – аватаров.

Сферы применения OpenSim:

- виртуальные представительства организаций (офисы, образовательные учреждения и др.);
- 3D-моделирование;
- моделирование физических процессов;
- ландшафтный дизайн.

Для программирования сценариев используется язык LSL (Linden Scripting Language), разработанный компанией Linden Labs. [4] Допустимо написание сценариев на языке C# [1]: несмотря на снижение читаемости кода, данных подход позволяет использовать дополнительные библиотеки.

2. Особенности реализации виртуальных рабочих мест.

В лаборатории НИР ИДДО УлГТУ реализованы виртуальные рабочие места по специальностям монтажник РЭАиП[17], слесарь-сборщик РЭАиП и регулировщик РЭАиП. В связи с этим ведётся внедрение и исследование виртуальных рабочих мест на АО «Ульяновский механический завод»; данная статья рассматривает в качестве примера рабочее место монтажника РЭАиП. (рис. 1). На рис. 2 приведена структура виртуального тренажёра. Клиентская среда визуализирует пространство, исходя из данных, хранящихся в БД OpenSim. В свою очередь, OpenSim передаёт команды ввода-вывода на сервер, который содержит файлы истории пользовательских действий, параметры состояния рабочих столов, а также сценарии ввода-вывода в указанные файлы.

На данный момент сформированы 3D-модели реальных объектов рабочего места – инструментов, комплектующих, готов базовый сценарий.

[17] РЭАиП – радиоэлектронная аппаратура и приборы

Рис. 1. Общий вид на виртуальную лабораторию.

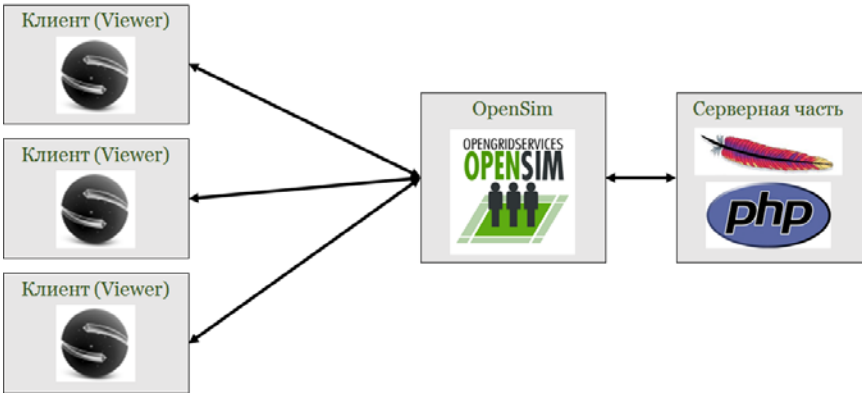


Рис. 2. Организация виртуального пространства.

Особое внимание уделяется процессам составления алгоритмов обучения. Так, для монтажника проектируются два сценария: пайка элементов и пайка проводов.

Пайка элементов включает в себя подсценарии припаивания следующих радиоэлементов: резистор С2-33Н, микросхема 1533ЛА3, конденсаторы К52-1 (электролитический), К10-17б (керамический). Для каждого элемента предусмотрены свои параметры пайки, такие как посадочное место на печатной плате, время и температура пайки.

В каждое виртуальное рабочее место встроена система анализа ошибок. Если обучающийся совершил какое-либо неверное действие,

система пытается вернуться к предыдущему состоянию, предварительно сигнализируя об ошибке с помощью текстовых сообщений (рис. 2).

Таким образом, в ходе обучения монтажник должен приобрести следующие знания и навыки: температурный режим пайки, время пайки, последовательность выполнения действий при монтаже радиоэлементов типа «микросхема», «конденсатор», зачистка и облуживание проводов, пайка проводов.

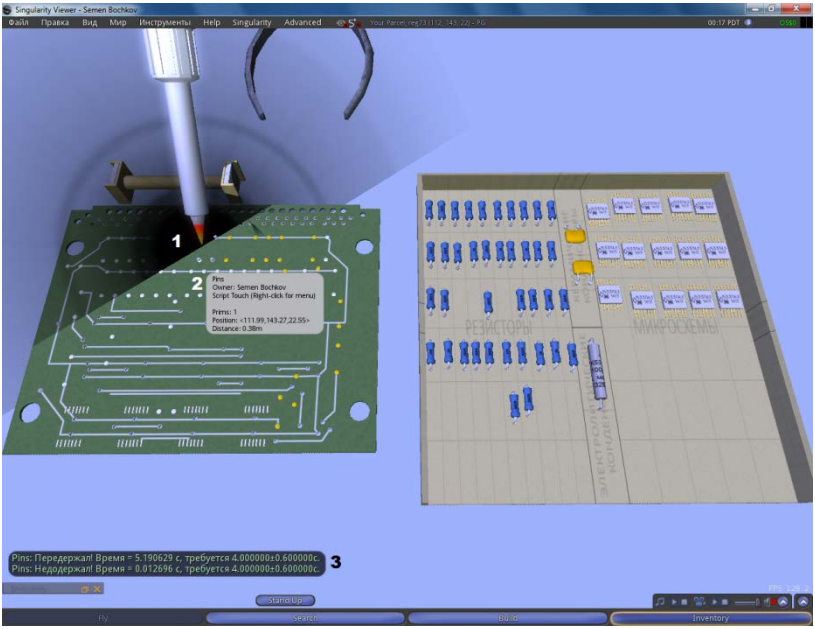


Рис. 2. Процесс пайки. Для лучшей видимости припой на паяльнике выделен красным цветом (1). Курсор мыши установлен на контактной площадке, о чём свидетельствует всплывающее сообщение (2), при этом время пайки не соответствует требуемому (3).

Заключение

Последние исследования показали, что среди взрослой аудитории виртуальные «я» есть лишь у 4% людей. Сегодня виртуальные представительства компаний на публичных платформах больше похожи на «города-призраки», однако для решения внутренних нужд продолжают создаваться «закрытые» площадки виртуальной реальности [7].

К таким закрытым площадкам относятся виртуальные рабочие места, разрабатываемые в Ульяновском государственном техническом

университете. Её отличие от виртуальных лабораторных площадок многих других университетов состоит в том, что она с высокой точностью и приближённостью к действительности моделирует рабочее место сотрудника завода.

Практическим результатом данного проекта является система формирования навыков у обучающегося при выполнении типичных рабочих операций.

Перспективными направлениями в области виртуализации являются:

- совершенствование проектируемых виртуальных рабочих мест;
- внедрение экспертной системы, оценивающей действия пользователя;
- интеграция системы виртуальных рабочих мест с LMS Moodle;
- освоение новых областей в сфере науки и промышленности для применения технологий виртуализации обучения.

Список литературы

1. C# scripting in OpenSim [Электронный ресурс] – Режим доступа: http://justincc.org/blog/2008/11/21/c-scripting-in-opensim/, свободный.
2. Diagnosing Medical Problems in Space Using Augmented Reality - The Atlantic [Электронный ресурс] – Режим доступа: http://www.theatlantic.com/health/archive/2012/02/diagnosing-medical-problems-in-space-using-augmented-reality/252880/, свободный.
3. LSL Portal - Second Life Wiki [Электронный ресурс] – Режим доступа: http://wiki.secondlife.com/wiki/LSL_Portal, свободный.
4. OpenSim [Электронный ресурс] – Режим доступа: www.opensimulator.org, свободный.
5. US Defense Agency Develops Virtual Reality Contact Lenses - Business Insider [Электронный ресурс] – Режим доступа: http://www.businessinsider.com/virtual-reality-contact-lenses-will-give-troops-a-tactical-edge-on-the-battlefield-2012-3, свободный.
6. Бахарева В.А., Захарова У.С., Сербин В.А., Фещенко А.В. Технологии виртуальной и дополненной реальности в образовательной среде вуза. // Открытое и дистанционное образование. Томск, 2015. №4 (60). С. 12-20.
7. Виртуальные миры: практическое применение / Блог компании Гарс Телеком / Хабрахабр [Электронный ресурс] – Режим доступа: https://habrahabr.ru/company/garstelecom/blog/140081/, свободный.
8. ИНОБР > Проекты > Компьютерные тренажерные системы [Электронный ресурс] – Режим доступа: http://misis.ru/inobr/kts_inobr, свободный.
9. Трухин А.В. Анализ существующих в РФ тренажерно-обучающих систем // Открытое и дистанционное образование. Томск, 2008. №1 (29). С. 32-40.

УДК 004.94

ИССЛЕДОВАНИЕ МОДЕЛЕЙ ОБУЧАЕМОГО В ПЕРСОНИФИЦИРОВАННЫХ АВТОМАТИЗИРОВАННЫХ СИСТЕМАХ ОБУЧЕНИЯ

С.И. Бригаднов[18]

В работе рассмотрена модель адаптивного контроля знаний на базе модели Л.А. Растригина. Произведен анализ и исследование моделей обучаемого, разработанных в различных учебных заведениях.

Введение

При разработке автоматизированных систем обучения особое внимание уделяется проблеме адаптации обучаемого, которая осуществляется на основе модели студента, включающей различные параметры [1], такие, как программа образования, специальность, личностные характеристики студента, используемые педагогические методы обучения и другие.

1. Модель адаптивного контроля знаний

Профессором Л.А. Растригиным [2] было предложено рассматривать процесс обучения как процесс управления сложной технической системой, в которой объектом управления является обучаемый [3]. Аналогично можно представить и процесс управления адаптивным контролем знаний (рис. 1).

Блок “*Алгоритм контроля*” выполняет следующие функции:

- анализ деятельности студента (проверка правильности его ответов и выполняемых действий);
- управление процессом контроля знаний на основе выбранного метода;
- определение результатов контроля, которое обычно сводится к выставлению оценки студенту.

[18] *432027, Ульяновск, ул. Северный Венец, 32, УлГТУ, e-mail:sergbrig@yandex.ru*

База знаний (БЗ) содержит методы и/или модели процесса контроля, а также совокупность знаний предметной области. *База данных* (БД) включает наборы вопросов и задач, предназначенных для проверки знаний студента и/или данные для формирования заданий. Контрольные задания могут также генерироваться автоматически на основе БЗ. База данных и база знаний совместно с моделью студента образуют репозиторий системы контроля.

Модель студента включает разнообразную информацию о студенте: предыстория обучения; результаты текущей работы (тип выполненных заданий, время выполнения заданий, число обращений за помощью и т.д.); личностные психологические характеристики (тип и направленность личности, репрезентативная система, способность к обучению, уровень беспокойства-тревоги, особенности памяти и др.); общий уровень подготовленности и другие [4].

Формировщик вопросов и задач используется для формирования и выдачи студенту очередного задания. Контроль знаний осуществляется следующим образом: студент выполняет предложенное задание, и результат его работы помещается в модель студента. Блок "Алгоритм контроля" на основе анализа ответа студента, целей контроля Z и используемого метода проведения контроля, учитывая внешние ресурсы R_1 (например, возможности системы контроля) и внутренние ресурсы студента R_2 (например, время контроля), а также состояние среды D_x, определяет параметры задания, которое должно быть предложено студенту. Формировщик вопросов и задач, получив от “Алгоритма контроля” данные о параметрах следующего задания, выбирает из БД и/или БЗ необходимую информацию I, формирует текст задания и выдает его студенту [5].

Рис.1. Модель адаптивного контроля знаний.

Таким образом, для управления адаптивным контролем знаний необходимо наличие:

- методов и моделей организации (проведения) контроля;
- моделей определения и оценки знаний, умений и навыков студента по результатам выполнения контрольных заданий.

2. Модели студента в автоматизированных системах обучения

Модели студента могут быть разделены на две основные группы: фиксирующие и имитационные. Фиксирующие, в свою очередь, включают скалярные, оверлейные (векторные и сетевые) и генетические графы. К имитационным можно отнести модели ограничений, ошибок и фальшправил [6]. Для отображения информации об обучаемом используются различные возможности: граф знаний, база знаний, сети Байеса и другие.

Произведен анализ моделей обучаемого, разработанных в различных учебных заведениях, далее рассмотрены некоторые из них:

1. Обучающая среда на базе гипермедиа (HBLE, Hypermedia-Based Learning Environment) основана на графовой модели, в вершинах которой находится отметка студента за определенную тему [7].
2. Экспертная оценка в реальном времени (OLAE, Online Assessment of Expertise) использует сети Байеса и определяет, что студент знает в отличие от остальных систем, которые определяют то, сколько студент знает [8].
3. Адаптивная тренировочная система (ATS, Adaptive Training System) отображает положения знаний студента и его переход в положение эксперта, предлагая адаптивное обучение в зависимости от его стиля [8].
4. Система Eon использует покрывающую модель для отображения знаний студента по каждому разделу изучаемой темы. Модель студента Eon также может быть использована как «библиотека ошибок», т.к. возможно определение того, что студент не понял в необходимой мере [9].
5. Формальные языки и среда автоматов (FLUTE, Formal Languages and aUTomata Environment) включает такие компоненты, как уровень текущих знаний, текущий прогресс, время и количество попыток при выполнении задания, поведение студента, инициатива [10].
6. Подход к моделированию студента для отзывчивого обучения (SMART, Student Modeling Approach for Responsive Tutoring) учитывает уровень знаний, умений и навыков. Изменение значения модели зависит от выполнения задания [11].

7. Система KBS гиперкнига (KBS Hyperbook System) основана на определении целей обучаемого для выбора проектов, а также отображения начальных знаний, умений и навыков [12].
8. Система IDEAL использует сети доверия Байеса, с помощью которых отображаются навыки студента в виде распределения на уровни (новичок, средний, эксперт и др.) [13].
9. Интернет-среда удаленного обучения (IDLE, Internet Distance Learning Environment) основана на парадигме гипертекста/гипермедиа. Эта среда использует направленный граф и учитывает уровень знаний студента [14].
10. Система Sydney принимает во внимание результаты психологического теста для определения последовательности обучения [15].
11. Сеть Хопфилда использует нейронные сети и учитывает следующие параметры: тип мышления обучаемого, воспринимаемая форма представления знаний, свойство уверенности при ответе, уровень усвоения знаний, оптимальная стратегия получения знаний [16].

В результате было установлено, что в настоящее время наиболее часто встречаются векторные и сетевые модели – 45% и 48% соответственно (рис. 2). В сетевых моделях информация о студенте отображается, как правило, с помощью графа знаний, вершины которого соответствуют квантам учебной информации, а дуги показывают отношения между ними.

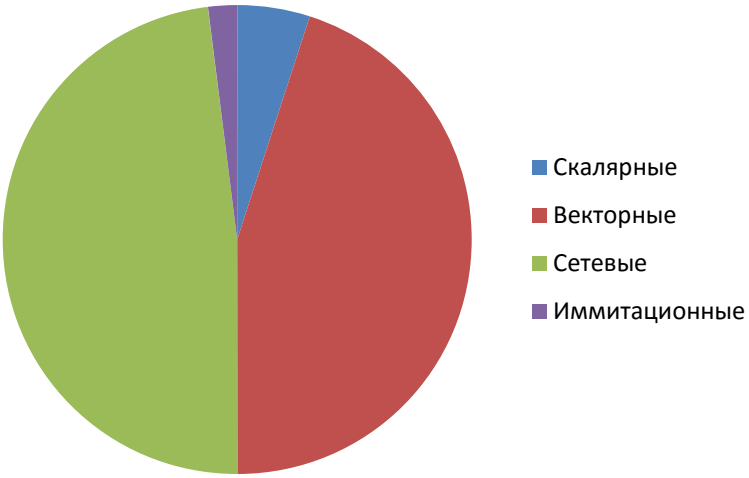


Рис.2. Распространенность моделей обучаемого.

Анализ моделей также проводился с точки зрения параметров, которые используются для отображения информации об обучаемом. Самыми распространенными параметрами [1], учитываемыми в моделях студента, являются следующие:

1. уровень знаний;
2. психологические характеристики (тип личности, ориентация и др.);
3. скорость/стиль обучения (усвоения, изучения);
4. выполнение заданий;
5. способность к обучению (очень внимательный, средне, мало);
6. уровень умений и навыков;
7. метод/стратегия обучения;
8. структура курса.

В таблице 1 приведены некоторые исследованные модели и используемые в них параметры.

Таблица 1. Модели обучаемого и учитываемые в них параметры.

Модель обучаемого	**Параметры**							
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**
Hypermedia-Based Learning Environment	+							+
Online Assessment of Expertise	+							
Adaptive Training System	+		+					
Eon	+						+	+
Formal Languages and aUTomata Environment	+			+	+			+
Student Modeling Approach for Responsive Tutoring	+		+			+		
KBS Hyperbook System	+	+			+			
IDEAL	+					+		
Internet Distance Learning Environment	+							
Sydney	+	+						
Сеть Хопфилда	+	+		+			+	

Анализ моделей показывает, что на результат учебного процесса влияют различные характеристики, при этом некоторые из них отображаются при помощи нескольких составляющих. По этой причине структура модели студента не является однотипной [1], а будет содержать следующие виды:

- вектор (опыт работы с автоматизированными системами обучения, психологические характеристики, текущая работа с курсом, общий уровень подготовки);
- взвешенный граф (предыстория обучения);
- скаляр (стратегия, метод обучения, специальность).

Заключение

Таким образом, для разработки персонифицированных автоматизированных систем обучения наиболее рациональным представляется использование смешанной структуры модели студента, содержащую необходимую информацию для организации адаптивного обучения.

Список литературы

1. Буль Е. Е. Обзор моделей студента для компьютерных систем обучения // Образовательные технологии и общество. – 2003. –Т. 4(6)
2. Растригин Л.А., Эренштейн М.Х. Адаптивное обучение с моделью обучаемого. - Рига : Зинатне, 1986. – 160 с.
3. L.Zaiceva, J.Bule, U.Kuplis. Advanced e-learning system development. // Proceedings of the International Conference on Advanced Learning technologies and Applications (ALTA'03). 11-12 September. Kaunas, Lithuania. – 14-18 pp.
4. Зайцева Л.В. Модели и методы адаптации к учащимся в системах компьютерного обучения // Educational Technology & Society. - Nr. 6(3), 2003. – с.204 – 212.
5. Зайцева Л.В., Прокофьева Н.О. Модели и методы адаптивного контроля знаний// Educational Technology & Society. - Nr. 7(4), 2004. – с.265 – 277.
6. Атанов Г.А., Пустынникова И.Н. Обучение и искусственный интеллект, или Основы современной дидактики высшей школы. – Донецк : Изд-во ДОУ, 2002. – 504 с.
7. Nykänen O., Ala-Rantala M. A Design for Hypermedia-Based Learning Environment.
8. Stauffer K, Applications of Student Modeling.
9. Murray T. Authoring Knowledge Based Tutors: Tools for Content, Instructional Strategy, Student Model, and Interface Design. Journal of the Learning Sciences, Vol 7, No 1, 1998, pp. 5-64.
10. Devedzic V., Debenham J., Popovic D. Teaching Formal Languages by an Intelligent Tutoring System. Educational Technology & Society 3(2) 2000. ISSN 1436-4522
11. Shute V.J. SMART: Student Modeling Approach for Responsive Tutoring.
12. Henze N., Nejdl W. Student Modeling in an Active Learning Environment using Bayesian Networks.
13. Yi Shang, Hongchi Shi, and Su-Shing Chen. An Intelligent Distributed Environment for Active Learning. http://www10.org/cdrom/papers/207/node4.html
14. Удальцов С.В. Среда разработки и поддержки систем дистанционного обучения IDLE: технологический аспект.
15. Хатьков Н. Д., Павличенко Ю. А. SYDNEY Интегрированная мультимедийная система компьютеризированного обучения.
16. Кольцов Ю. В. Добровольская Н. Ю. Нейросетевые модели в адаптивном компьютерном обучении. Educational Technology & Society 5(2) 2002 ISSN 1436-4522, с.213 – 216.

УДК 004.02

ОПИСАНИЕ МОДЕЛИ НАКОПЛЕНИЯ И ФИКСАЦИИ ОПЫТА ПРОЕКТИРОВЩИКОВ ПОСРЕДСТВОМ СОЗДАНИЯ И ПОВТОРНОГО ИСПОЛЬЗОВАНИЯ МЕТОДИК ПРОЕКТИРОВАНИЯ, ОСНОВАННОЙ НА МЕТОДЕ «EXPERIENTIAL LEARNING» Д.А. КОЛБА

А.А. Васильев

Статья посвящена описанию возможности применения циклической модели процесса обучения и усвоения человеком новой информации («Experiential Learning Model») в профессиональной деятельности проектных организаций. В статье описан механизм создания в проектной организации дополнительных условий, стимулирующих самообучение проектировщиков путем фиксации, анализа, обобщения и повторного использования опыта проектирования. Приведено математическое описание рассматриваемой модели.

Введение

Опыт деятельности проектных организаций демонстрирует, что в оперативных процессах проектирования возникают ситуации, когда члену группы проектировщиков или группе в целом требуется обучиться в сжатые сроки по определенной теме и, как следствие, решить возникшую новую проектную задачу.

Ключевая идея заключается в том, чтобы создать в проектной организации дополнительные условия, стимулирующие самообучение проектировщиков на основе циклической четырехступенчатой эмпирической модели процесса обучения и усвоения человеком новой информации («Experiential Learning Model»), предложенной Дэвидом А. Колбом (David A. Kolb) и его коллегами из Case Western Reserve University. Данная тема подробно описана в предыдущей статье автора в рамках сборника научных трудов ИВТ-2015 [1].

В основе циклической модели Д.А. Колба положена идея «обучение практикой», обучение состоит из повторяющихся этапов «выполнения» и «мышления». Стадии цикла обучения Д. Колба представлены на рисунке 1.

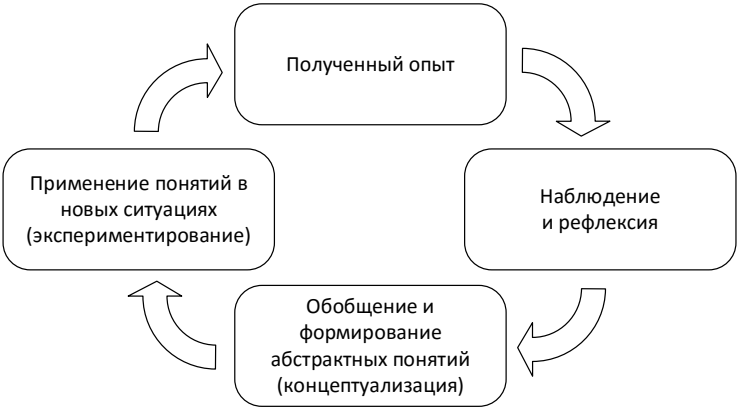


Рис. 1. Цикл обучения Д.А. Колба

Отправным моментом естественного обучения является приобретение конкретного опыта, который дает материал для рефлексивного наблюдения. Обобщив новые данные и интегрировав их в систему имеющихся знаний, человек приходит к абстрактным представлениям и понятиям, отстраненным от непосредственного опыта. Эти новые знания представляют собой гипотезы, которые проверяются в ходе активного экспериментирования в разнообразных ситуациях: воображаемых, моделируемых и реальных. Для того чтобы обучение состоялось, необходимо пройти все четыре этапа. Процесс обучения протекает циклически – до тех пор, пока не сформируется требуемый навык; как только один навык освоен, мозг готов к обучению следующему.

Способом создания подобных условий было выбрано введение в профессиональную деятельность проектировщиков процесса накопления и фиксации их опыта путем создания и повторного использования методик проектирования.

Для фиксации опыта проектировщика выбрана вопросно-ответная моделирующая среда WIQA.Net, являющаяся мощным инструментом для разработки систем автоматизированного проектирования [2], принятие решений в которой фиксируется в форме вопросно-ответного протокола, формализованной фиксации опыта в виде прецедентов [3].

Применение моделирующей среды WIQA.Net позволило инструментально-моделирующий потенциал взаимодействий с доступным опытом и его моделями довести до создания и использования «Баз Опыта» проектных организаций, разрабатывающих семейства автоматизированных систем [4].

1. Основные принципы модели

Модель накопления и фиксации опыта проектировщиков посредством создания и повторного использования методик проектирования схематично представлена на рисунке 2.

Разберем ее на примере «*Проектирование систем электропитания для корабельных комплексов средств автоматизации*» (данный пример детально описан в предыдущей статье автора [5]).

Проектировщику поставлена задача *Z1 «Спроектировать систему электропитания для корабельного комплекса средств автоматизации (КСА) «А»*.

При отсутствии конкретных методик решения данной задачи проектировщик вынужден решать задачу так, как он это умеет, как подсказывает его личный опыт, вне зависимости от того, являются ли его навыки достаточными.

Таким образом, на начальном этапе основа его действий не имеет полного состава ориентиров. Ориентиры действий должны далее самостоятельно открываться проектировщиком на основе проб и ошибок.

С целью фиксации опыта решения задачи, параллельно с ее решением проектировщик обязан проанализировать свои действия, привести полученный опыт в определенный порядок, достаточно удобный для того, чтобы сделать то или иное умозаключение.

При использовании инструментария вопросно-ответной моделирующей среды WIQA.Net опыт (прецедент) решения задачи Z1 проектировщик представил в виде вопросно-ответного протокола:

Q 0.1 Какие операции необходимо выполнить проектировщику, чтобы спроектировать систему электропитания комплекса «А»?

A 0.1 Осуществить выбор наиболее оптимальной схемы из имеющихся типовых схем системы электропитания, применяемых на предприятии.

A 0.2 Рассчитать параметры сети (произвести расчет электрической мощности).

A 0.3 Осуществить выбор трансформаторов, определить их количество.

A 0.4 Осуществить выбор аппаратуры коммутации и управления системы электропитания.

A 0.5 Обеспечить защиту цепей электропитания технических средств (ТС) от перегрузок по току и коротких замыканий в цепях электропитания.

A 0.6 Обеспечить индикацию состояния системы электропитания на местных и удаленных пультах управления и контроля.

Дав ответы на каждый вопрос, проектировщик получит решение задачи Z1 в виде QA протокола решения – обозначим его как QA_{s1}.

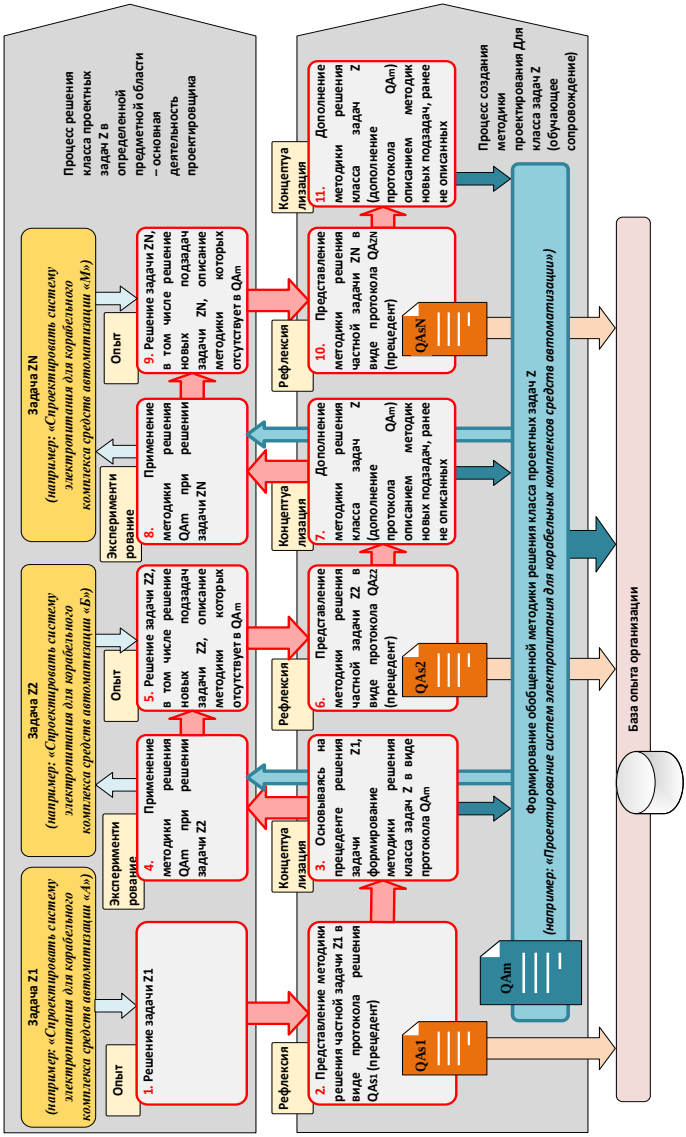


Рис. 2. Пример совмещения цикла обучения Д.А. Колба с деятельностью проектной организации

Представление решения задачи в таком виде можно соотнести с этапом «наблюдения и рефлексии» в цикле Д.А. Колба.

Исключив из протокола QA_{s1} конкретные параметры значений, оставив лишь описание методики решения задачи (описание каждой операции), проектировщик получит методику решения задачи *Z1*. Данную методику в среде WIQA.Net также можно представить в виде вопросно-ответного протокола методики решения проектной задачи – QA_m.

Пример представления вопросно-ответного протокола в среде WIQA.Net приведен на рисунке 3.

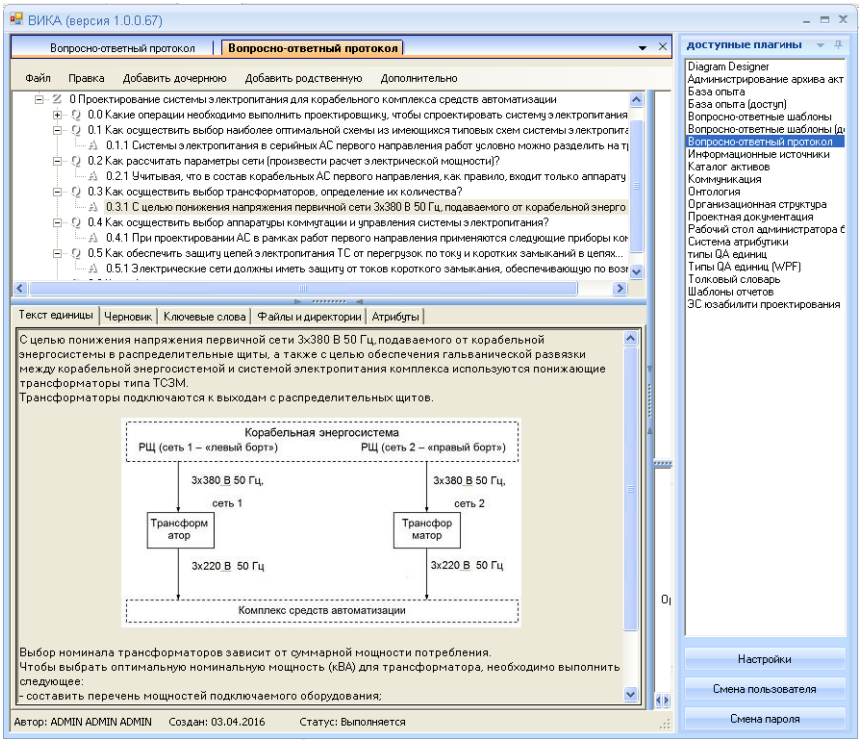


Рис. 3. Вид вопросно-ответного протокола в среде WIQA.Net

Представление решения задачи в таком виде соотносится с этапом «обобщение и формирование абстрактных понятий (концептуализация)» в цикле Д.А. Колба. Описание методик и наполнение их теоретическими основами по предметной области также заставляет проектировщика еще

больше углубиться в изучаемую предметную область и структурировать свои знания в ней.

Началом «второго витка» спирали цикла Д.А. Колба будет использование полученной методики решения комплекса задач Z для решения частной задачи *Z2 «Спроектировать систему электропитания для корабельного комплекса «Б»* (что соответствует этапу «применение понятий в новых ситуациях» в цикле Д.А. Колба).

Предположим, в условиях задачи Z2 присутствует требование, которого не было в условиях задачи Z1, *(например, требование по снижению уровня помех в цепях электропитания ТС до заданного значения, которого не было в задаче Z1).*

В результате обобщенная методика решения задач класса Z – протокол QA_m, полученный на опыте решения задач Z1 и Z2, будет содержать в себе опыт реализации всех требований, встречающихся в задачах Z1 и Z2.

Также на «втором витке» спирали цикла Д.А. Колба проектировщик, получив определенный опыт в изучаемом конкретном вопросе, анализирует все плюсы и минусы приобретенного опыта и получает выводы о том, что было сделано удачно, а что можно было бы сделать лучше или по-другому. Следуя методике Д.А. Колба, после получения практических знаний человек обращается к получению теоретических знаний о том, как действовать правильно в связке с приобретенным опытом и его анализом. Можно говорить, что на «втором витке» спирали Д.А. Колба проектировщику предоставляется совокупность частных ориентиров, необходимых для правильного и разумного выполнения действий.

Соответственно и далее с каждым витком *«спирали»* в методику решения класса задач *Z «Проектирование систем электропитания для корабельных комплексов»* добавляется новая информация. Методика становится полнее.

С каждым последующим «витком» спирали цикла Д.А. Колба, добавляя в методику решения новые операции, проектировщик получает мощный инструмент, пригодный для анализа и решения класса задач, а также быстрого обучения новых специалистов.

2. Математическое описание модели

Выразим модель создания обобщенных методик проектирования языком теории множеств.

Пусть s1 – решение задачи z1 (например, «*Спроектировать систему электропитания для корабельного комплекса средств автоматизации «А»*). Где z1 относится к классу задач Z («*Проектирование систем электропитания для корабельных комплексов средств автоматизации*»).

$z1 \in Z$

Решение s1 состоящее из последовательности выполняемых операций можно представить в виде вопросно-ответного QA протокола решения задачи z1, обозначим его множеством QA_{s1}. Множество QA_{s1} содержит решение задачи в виде единиц опыта qa.

$$\{s1\} = \{op_{s.1}, \dots , op_{s.i}\} \sim QA_{s1} = \{qa_{s.1}, \dots , qa_{s.i}\}$$

Для решения s1 мы можем выделить методику ее решения m’, которую также, в свою очередь, представить в виде QA протокола методики решения задачи z1. Обозначим данную методику множеством QA_{m1}.

$$\{m'\} = \{op_{m.1}, \dots , op_{m.i}\} \sim QA_{m1} = \{qa_{m.1}, \dots , qa_{m.i}\}$$

QA_{m1} отличается от QA_{s1} тем, что методика решения не содержит конкретных значений параметров, заданных в задаче z1 и имеет описанный порядок решения в общем виде.

Тогда для частной задачи z2 (например, задачи «*Спроектировать систему электропитания для корабельного комплекса средств автоматизации «Б»*») решение имеет вид:

$$\{s2\} = \{op_{s.k}, \dots , op_{s.j}\} \sim QA_{s1} = \{qa_{s.k}, \dots , qa_{s.j}\}$$

При этом методику решения задач z1 и z2 содержащую описание решения в общем виде, можно представить как:

$$\{m''\} \sim QA_{m1} \cup QA_{m2} = \{qa_m \mid qa_m \in QA_{m1} \vee qa_m \in QA_{m2}\}$$

При наличии N решений задач из области Z, методика будет выражаться:

$$\{m'^N\} \sim QA_{m1} \cup \dots \cup QA_{mN} = \{qa_m \mid qa_m \in QA_{m1} \vee \dots \vee qa_m \in QA_{mN}\}$$

Можно сделать вывод: чем больше частных задач решено, тем более полноценной получается сохраненная в базе опыта обобщенная методика решений QA_m класса задач Z.

Заключение

В статье рассмотрена модель применения метода Д.А. Колба в деятельности проектных организаций посредством создания и использования методик проектирования. На рабочем месте человек постоянно сопряжен с получением нового опыта, поэтому включение в процесс его основной деятельности обучения (а точнее самообучения) по методу Д.А. Колба приведет его к более глубоким абстрактным представлениям и понятиям, а новые знания повлекут за собой новые гипотезы решения поставленных задач.

Таким образом, создание подобных методик просто необходимо применять на практике, чтобы добиться высокой эффективности в вопросах повышения квалификации проектировщиков и повышения профессиональной зрелости организации в целом.

Совершенствование способов обучения и самообучения персонала является важным аспектом в вопросе исследования методов и средств, способствующих управляемому воздействию на профессиональную зрелость [6] в решении задач повышения степени успешности современных проектных организаций.

Список литературы

1. Васильев А.А. Персонифицированное обучающее сопровождение в деятельности проектных организаций, связанных с разработкой автоматизированных систем, на основе модели «experiential learning». // Информатика и вычислительная техника: сборник научных трудов / под ред. В. Н. Негоды. – Ульяновск: УлГТУ, 2015. С. 50-58.
2. Маклаев В.А., Соснин П.И. Инструментально-технологическая среда формирования и использования опыта проектной организации // Автоматизация процессов управления. – 2009. – № 2 (16). – С. 8–13.
3. Соснин П.И. Вопросно-ответное моделирование в разработке автоматизированных систем. – Ульяновск : УлГТУ, 2007. − 333 с.
4. Маклаев В.А., Соснин П.И. Создание и использование автоматизированной базы опыта проектной организации. – Ульяновск : УлГТУ, 2012. − 359 с.
5. Васильев А.А. Персонифицированное обучающее сопровождение в деятельности проектных организаций на основе модели «Experiential Learning» Д.А. Колба // Автоматизация процессов управления. – 2016. – № 2 (44). – С. 77–86.
6. Curtis, B.: People Capability Maturity Model (P-CMM) Version 2.0/ B. Curtis, B. Hefley, S. Miller // Technical Report CMU/SEI-2009-TR-003 P, 2009.

УДК 621.372

АДАПТИВНАЯ МОДЕЛЬ СИСТЕМЫ ПРИНЯТИЯ РЕШЕНИЙ НА ОСНОВЕ ПЛАСТИЧНЫХ СИНАПСОВ С СИГНАЛАМИ НЕОЖИДАННОСТИ В ЭЛЕКТРОННОМ ОБУЧЕНИИ

Е.Ю. Воеводин[19]

В этой статье рассматриваются нейронные сети с пластичными синапсами, представлен анализ исследований в области исследований пластичных синапсов, предложена модель адаптивной системы принятия решений в области электронного обучения на основе пластичных синапсов с сигналами неожиданности.

Введение

Создание адекватной модели адаптивного обучения и многоагентной системы поддержки принятия решений в области электронного обучения является стратегической задачей в области развития информационных систем[1]. Такие системы позволяют выбрать оптимальное решение в условиях неопределенности, основываясь на опыте и памяти, полученном в результате обучения и эксплуатации системы. Большинство существующих систем поддержки принятия решений оперируют с помощью сетей, где скорость изменения пластичности нейрона является константой, или скорость изменения пластичности всех нейронов сети является общей для всех нейронов сети, и изменяется в зависимости от количества результатов или вердиктов, выданных сетью. Однако исследования в области бионейронов показывают, что уровень пластичности синапсов может меняться в зависимости от количества активаций конкретного нейрона, а нейронная сеть, на основе пластичных синапсов выполняет некоторые задачи успешнее, чем классические нейронные сети.

[19] *432027, Ульяновск, ул. Северный Венец, 32, УлГТУ, e-mail:voevodineu@gmail.com*

Предпосылки для создания модели

Люди и другие животные обладают способностью к адаптации принятия решений к изменениям в среде обитания. Эксперимент под названием "многорукий бандит" показывает этот процесс в действии. Ролью испытуемого в решении этой задачи является выбор между несколькими аппаратами. На каждом этапе один из них имеет более высокую вероятность получения награды, или более высокий суммарный уровень награды, чем три других, и, на этом этапе, будет пользоваться предпочтением в выборе у испытуемого. Если другой аппарат начинает чаще выдавать награду, или суммарный уровень качества награды становится выше, испытуемый меняет свое предпочтение в сторону этого аппарата. Однако, исследование показало, что наибольшую эффективность в результате эксперимента проявили ситуации, в которых происходила частая смена предпочтительного аппарата. Другими словами, обучение происходит эффективнее в условиях неопределенности.[2]

Изменения прочности связей между нейронами называются синапсами. Получение награды усиливает синапсы между активированными нейронами, и, соответственно, отсутствие награды, ослабляет их. Этот процесс получил название синаптической пластичности. Однако, существующие модели с изменением синаптических связей с постоянной скоростью не объясняют феномена более эффективного обучения в условиях неопределенности.

Однако, известно, что существуют различные химические процессы, происходящие в биосинапсах, которые влияют на синаптическую пластичность. В каскадной модели, построенной по принципу метапластичности, синапсы способны изменять свою пластичность в зависимости от частоты использования связи между нейронами. Эксперимент «многорукий бандит» с одним фаворитом, или плавным изменением фаворита показал еще одну интересную особенность: наиболее эффективной оказалась модель на основе метапластичных синапсов с параметром силы синапса. Сила синапса определяет, насколько часто была использована синаптическая связь, и уменьшает скорость изменения пластичности синапса в случае получения награды, и, соответсвенно, увеличивает скорость изменения пластичности синапса в случае отсутствия вознаграждения, или, в случае эксперимента с качественным характером награды (например денежный эквивалент выигрыша), изменение скорости пластичности происходит в сравнении выигрыша со средним арифметическим общего уже полученного выигрыша.[3] Таким образом задача оптимизации успешно решается в

экспериментах с постоянным фаворитом, или плавным изменением фаворита.

Однако такая модель показывает свою неэффективность в ситуациях с быстро меняющимся фаворитом, или в условиях полной неопределенности фаворита. Чем больше подходов к аппаратам тем прочнее связи между нейронами, и тем более статична сила нейронов. Изменение этой силы требует немало неэффективных подходов к аппаратам в случае резкой смены фаворита, а в случае постоянной смены фаворитов на определенном участке времени дает эффективность сравнимую со случайным выбором аппарата.

Таким образом очевидно, что каскадная модель на основе метапластичных нейронов нуждается в дополнительном внешнем сигнале, полностью сбрасывающем память. Этот сигнал должен характеризоваться моделью, как полная смена правил получения выигрыша, и либо сбрасывать силы и пластичности синапсов на усредненные значения начала обучения сети, либо ослаблять силы синапсов в процентном взаимоотношении.

Применение модели в агентной системе

Легко представить себе плавное изменение фаворитов в рекомендациях адаптивной системы. Промежуточные тестовые испытания буду представлять собой аналог награды, так как оценки за компетенции являются измеряемыми, и, используя относительный и абсолютный критерий сравнения, легко понять, положительный или отрицательный эффект имели предыдущие рекомендации. Однако, нередки ситуации, когда следующая по списку изучения компетенция имеет слабую связь с предыдущей, не имеет связи вообще, или личный опыт обучаемого не обладает пререквизитами для изучения данной компетенции. После получения данных о успешности изучения данная ситуация может характеризоваться системой как стрессовая, и должна привести к переосмыслению расстановки приоритетов в принятиях решений о рекомендациях учебных материалов.

Заключение

Ожидается, что модель системы с метапластичными синапсами с внешним сигналом неопределенности будет являться более универсально-функционирующей системой по сравнению с системами постоянной скорости изменения силы синапсов. Тестирование системы будет включать внезапную смену успехов в изучении компетенций, а также вариативную продолжительность успехов изучения компетенций.

Список литературы

1. Афанасьев А.Н., Белухина Н.Н., Канев Д.С., ИНТЕЛЛЕКТУАЛЬНЫЕ СРЕДЫ ОБУЧЕНИЯ В ОТКРЫТОМ ОБРАЗОВАНИИ / Дистанционное и виртуальное обучение. 2016. № 8 (110). С. 4-14.
2. Jeanette Hellgren Kotaleski & Kim T. Blackwell, MODELLING THE MOLECULAR MECHANISMS OF SYNAPTIC PLASTICITY USING SYSTEMS BIOLOGY APPROACHES / Nature Reviews Neuroscience 11, 239-251 (April 2010)
3. Stefano Fusi & L F Abbott, LIMITS ON THE MEMORY STORAGE CAPACITY OF BOUNDED SYNAPSES / Nature Neuroscience 10, 485 - 493 (2007)

УДК 004

АВТОМАТИЗАЦИЯ ПРОЦЕССА ОБУЧЕНИЯ ЯЗЫКАМ ВЫСОКОГО УРОВНЯ[20]

Н.Н. Войт[21], И.В. Беляева[22]

Авторами представлено показательные занятия обучению Турбо Паскалю по персонифицированным траекториям учебно-методического материала в автоматизированной среде обучения, что обеспечило повышение качества процесса обучения в рамках запланированных занятий.

Вопросы повышения качества процесса обучения, автоматизация контрольно-измерительных персонифицированных характеристик обучающихся, параллельного процесса обучения, компьютеризации учебных процессов являются актуальными как для научного сообщества, так и для практиков.

Значимым направления развития автоматизированных систем обучения и управления является их интеллектуализация (рис. 1).

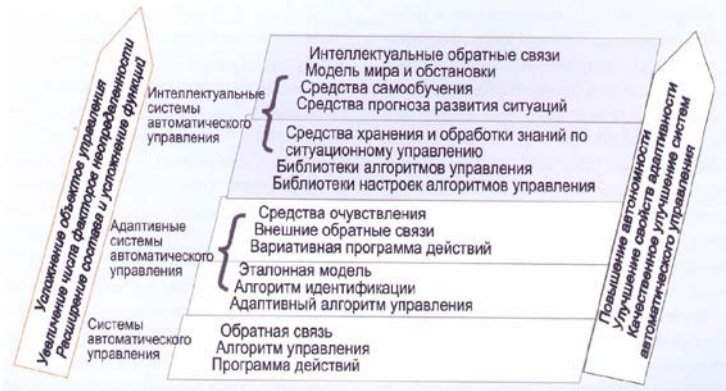


Рис. 1. Развитие принципов автоматического и интеллектуального управления

---

[20] *Исследование выполнено при финансовой поддержке РФФИ в рамках научного проекта № 15-07-08268 а.*

[21] *432027, Ульяновск, ул. Северный Венец, 32, УлГТУ, e-mail: n.voit@ulstu.ru*

[22] *432027, Ульяновск, ул. Северный Венец, 32, УлГТУ, e-mail: belyaeva_iv@mail.ru*

Интеллектуальные системы обучения классифицируются на адаптивные системы с интеллектуальной обратной связью и экспертные (рекомендательные) системы (рис. 1). Одной из ключевых задач при обучении является контроль и мониторинг успеваемости обучающихся.

Авторами разработана и предложена интеллектуальная система обучения [1-6] для прохождения теоретического учебно-методического материала, состоящего из текста, рисунков, таблиц, диаграмм как лекционного материала и вопросов, сгруппированных в тесты, как контрольно-измерительного материала.

Авторами разработан курс обучения Турбо Паскалю. Обучение проходит в браузере (рис. 2).

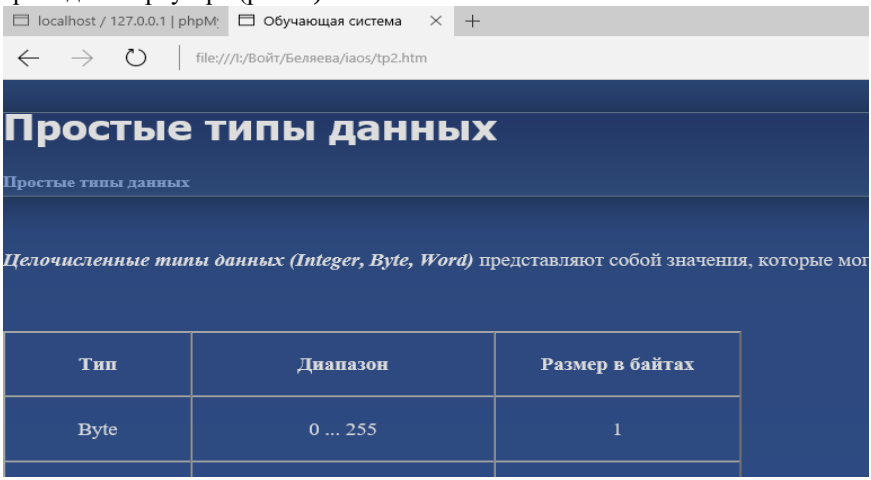


Рис. 2. Экранная форма обучения Турбо Паскалю. Тема: «Простые типы данных»

Интеллектуальная система обучения создана по клиент-серверной архитектуре и содержит компоненты редактирования теоретического материала, хранения его в виде онтологической модели (рис. 3) и представления в локальной сети учебной аудитории. Вопросы привязаны к конкретной грануле (дозе) знаний, что позволяет строить дополнительные динамические персонифицированные траектории обучения, если на вопрос дан неправильный ответ.

Разработанная система обеспечивает параллельный процесс обучения по языкам высокого уровня для студентов начальных курсов на компьютерах в локальной сети кафедры «Вычислительная техника» УлГТУ, мониторинг и контроль успеваемости обучающихся, представленной статистикой. Демонстрация показательного занятия представлена на рис. 4.

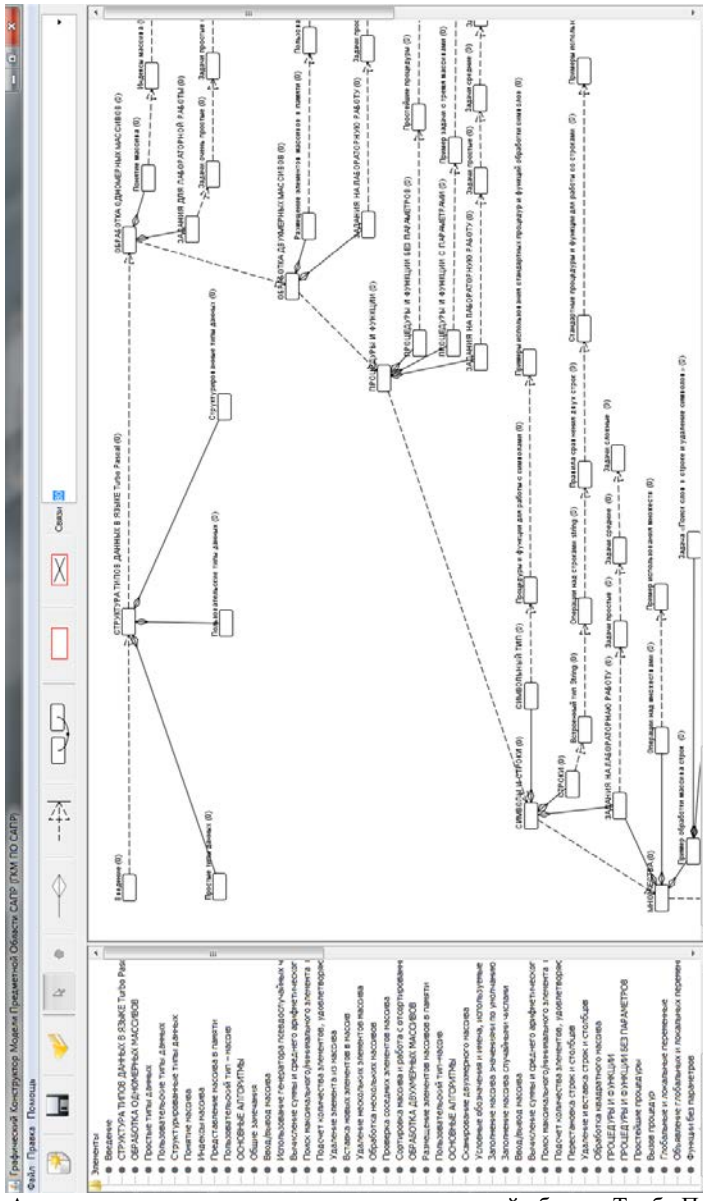


Рис. 3. Ассоциативно-иерархическая модель предметной области Турбо Паскаль

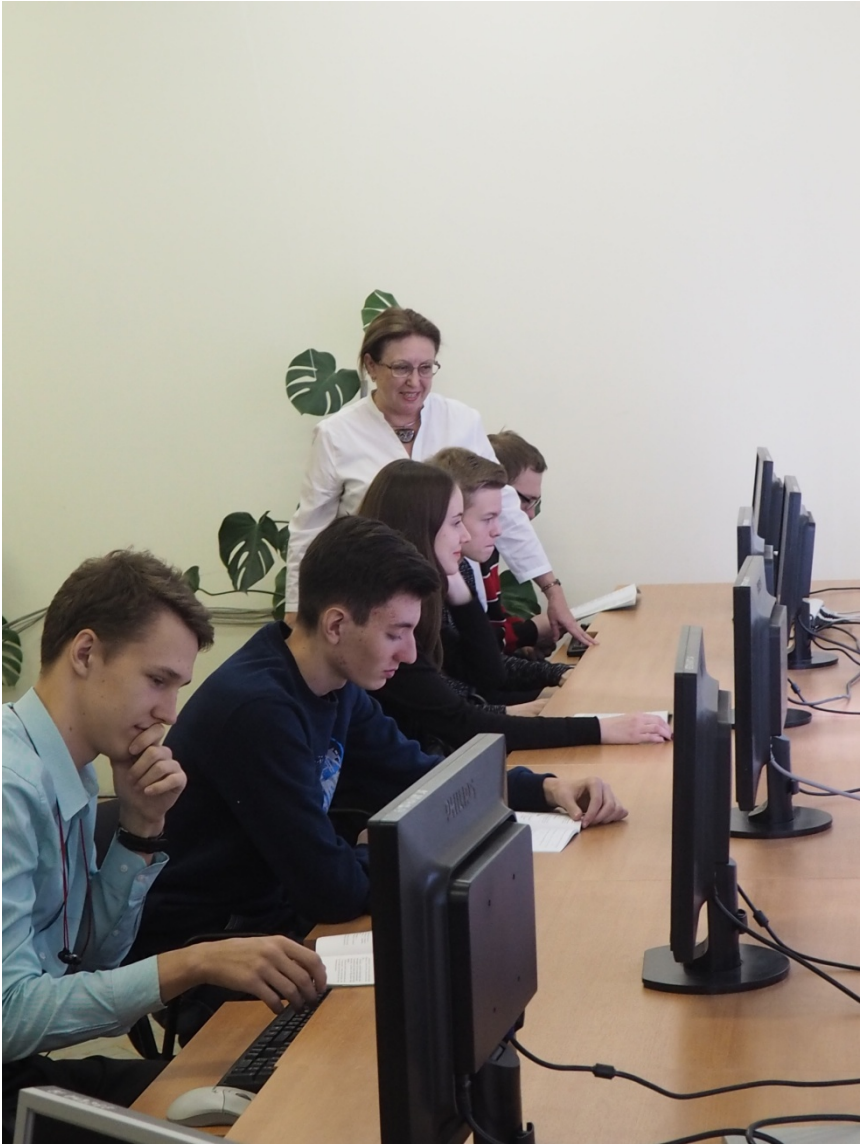


Рис. 4. Демонстрация процесса обучения по персонифицированным траекториям в автоматизированной среде обучения [6]

Список литературы

1. Н.Н. Войт, И.В. Беляева Реализация электронного курса по Турбо Паскалю // Вузовская наука в современных условиях: сборник материалов 50-й научно-технической конференции. В 3 ч. Ч. 3. - Ульяновск: УлГТУ, 2016. - С. 306-307.
2. Н. Н. Войт, И. В. Беляева Разработка 3D-модели корпуса конденсатора К73-17 в среде WINDGS3D для проектирования печатной платы в САПР KICAD // Информатика, моделирование, автоматизация проектирования: сборник научных трудов / под ред. А. Н. Афанасьева. - Ульяновск: УлГТУ, 2015. – С.–104-113.
3. Афанасьев А.Н., Игонин А.Г., Афанасьева Т.В., Войт Н.Н. Использование нейросемантических сетей для автоматизированного проектирования вычислительной техники // Автоматизация. Современные технологии. – 2008. – № 1. – С. 21-24.
4. Афанасьев А.Н., Войт Н.Н. Разработка компонентно-сервисной платформы обучения: диаграммы использования и деятельности программного компонента сценария на UML-языке // Вестник Ульяновского государственного технического университета. –2012. – № 1 (57). – С. 66-68.
5. Афанасьев А.Н., Войт Н.Н. Реализация конструктора сценария обучающих курсов // Вестник Ульяновского государственного технического университета. – 2011. – № 1 (53). – С. 54-59.
6. Интеллектуальная программа диагностики знаний, умений, владением навыков и компетентности обучаемого / А.Н. Афанасьев, Н.Н. Войт // Свидетельство о государственной регистрации программы для ЭВМ №2012612448; заявитель и правообладатель Ульян. гос. тех. ун-т. - № 2012610034; заявл. 10.01.2012; зарегистр. 06.03.2012г. – Москва, Роспатент, 2012.

УДК 004.021

ОСОБЕННОСТИ ПРИМЕНЕНИЯ АЛГОРИТМОВ НА ГРАФАХ В ПРИКЛАДНЫХ ЗАДАЧАХ ПОВЕДЕНЧЕСКОЙ АНАЛИТИКИ

П.М. Волков, И.С. Блохин, З.В. Степчева[23]

В работе исследована возможность применения вычислительных алгоритмов на графах для построения и анализа поведенческих моделей пользователей веб-приложений на примере Интернет-магазина.

Введение

Ни для кого не секрет, что самым выгодным бизнесом в глобальной сети является интернет-магазин. Обороты рынка электронной коммерции выросли во много раз и скоро достигнут отметки в $ трлн. Компаниям Интернет предоставил новый инструмент ведения бизнеса, а их клиенты, в свою очередь, получили новый информационный источник о товарах и услугах, а также возможность более удобно взаимодействовать с ними. Но как и с помощью чего привлекать на свои сайты клиентов?

Появилась необходимость анализировать интернет-рынок. Начали появляться инструменты аналитики веб-страниц и приложений, появились узкоспециализированные сервисы, как правило, с платным доступом, которые на основе поведенческой аналитики пользователей оптимизируют сайт, поднимая его выше на странице результатов поисковых систем. Делают они это, в основном, путем улучшения, так называемых, поведенческих факторов.

Поведенческие факторы − это ряд всевозможных действий пользователей, которые они могут произвести на сайте: вход, просмотр страниц, клики на ссылки т.д. [1].

Основным недостатком подобных сервисов является явно завышенная цена для предоставляемых ими услуг, а, следовательно - недоступность для малого бизнеса, а также невысокая эффективность для специализированных веб-приложений, требующих нестандартного

[23] *432027, Ульяновск, ул. Радищева, 102, МБОУ Лицей при УлГТУ, e-mail:stzv78@gmail.com*

подхода к анализу поведенческих факторов [1-2]. Таким образом, можно сделать вывод, что подобные сервисы не всегда актуальны и эффективны.

Эффективным альтернативным инструментом для изучения поведенческой аналитики является моделирование поведения пользователей в виде взвешенного графа и использование алгоритмов на графах для анализа поведенческих факторов [3].

1. Основная идея

Родившись при решении головоломок и занимательных игр, теория графов стала в настоящее время простым, доступным и мощным средством решения вопросов, относящихся к широкому кругу проблем. За последние четыре десятилетия теория графов превратилась в один из наиболее бурно развивающихся разделов математического моделирования, а широкие возможности алгоритмов на графах [3] позволяют использовать их для моделирования поведения пользователей в сети интернет и дальнейшего анализа поведенческих факторов.

2. Цель работы

Цель работы – выявить особенности применения алгоритмов на графах при реализации прикладных задач поведенческой аналитики на примере простейшего интернет-магазина.

Для достижения цели требовалось решить ряд задач:

- изучить архитектуру веб-приложений, модели и технологии программирования клиентского и серверного программного обеспечения интернет-магазинов, технологии построения и использования базы данных,
- определить место модуля построения графа в архитектуре веб-приложения,
- проанализировать и выбрать эффективный алгоритм анализа графа,
- разработать и апробировать соответствующее программное обеспечение, произвести оценку результата.

3. Разработка программного обеспечения и анализ результатов тестирования

Разработанное веб-приложение, решающее поставленную выше задачу, построено в клиент-серверной архитектуре, представленной на рис. 1.

Для практической реализации поставленной задачи был написан интернет- магазин по продаже велосипедов и их комплектующих.

Рассмотрим модуль взаимодействия с базой данных (БД). Принцип работы прост.

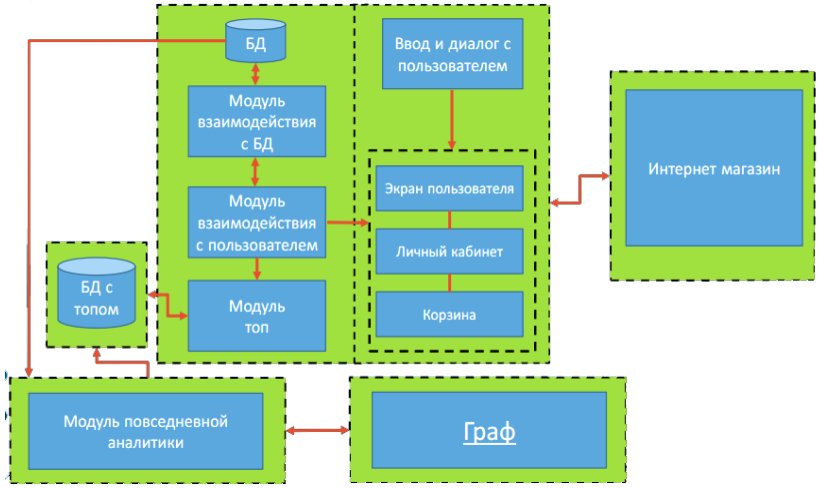


Рис.1. Общая схема проекта

На каждой странице выполняется скрипт, который добавляет в таблицу с действиями пользователей запись: IP адрес пользователя, страницу, на которой он сейчас находится, а также дату и время посещения. Далее данные из БД передаются модулю аналитики. Модуль позволяет построить граф (визуализация показана на рис. 3.) и проанализировать следующие поведенческие факторы: время, проведенное на сайте, глубину просмотра (страниц), возвращение к повторному поиску, непосредственное возвращение на сайт, минуя поиск. Пример результата работы модуля показан на рис. 2.

```
<terminated> Main (1) [Java Application] C:\Program Files\Java\jre1.8.0_71\bin\javaw.exe (10 апр. 2016 г., 17:41:09)
1 Bicycle Shop - Главная   Все велосипеды
2 Все велосипеды   Описание Corratec TwoCircle TZ
3 Описание Corratec TwoCircle TZ   Bicycle Shop - Главная
4 Bicycle Shop - Главная   Все аксессуары
5 Все аксессуары   Описание Cycledesign BB remover
6 Описание Cycledesign BB remover    Описание GT Essence Multutool 10
7 Описание GT Essence Multutool 10   О магазине
8 О магазине   О доставке
9 О доставке   Все велосипеды
10 Все велосипеды   Описание Mongoose TEOCALI Exp
11 Описание Mongoose TEOCALI Exp    Корзина
12 Корзина   Покупка из корзины
Куплен Покупка из корзины за 8
```

Рис. 2. Результат работы модуля аналитики

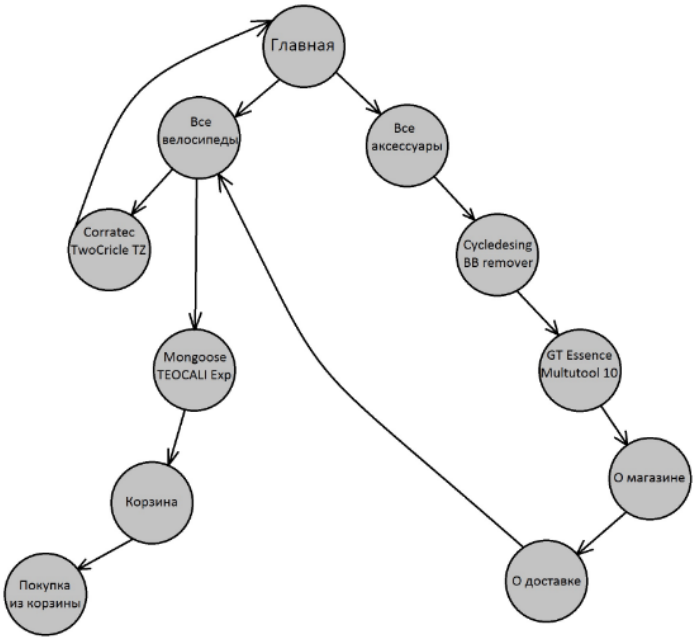


Рис. 3. Визуализация построенного графа

Из рис. 2 и рис. 3 следует, что до факта покупки товара пользователь прошел путь из семи страниц интернет-магазина. На основании этих данных мы стараемся сделать так, чтобы этот путь был наименее

коротким. Для этого в полученном графе реализуем поиск кратчайшего пути [3], листинг 1.

```
public void path() {
  int startTree = 0;
  vertexList[startTree].isInTree = true;
  nTree = 1;

  for (int i = 0; i < nVerts; i++) {
    int tempDist = adjMat[startTree][i];
    sPath[i] = new DistPar(startTree, tempDist);
  }

  while (nTree < nVerts) {
    int indexMax = getMax();
    int maxDist = sPath[indexMax].distance;

     if (maxDist == INFINITY) {
      System.out.println("Граф построен");
      break;
    } else {
      currentVert = indexMax;
      startToCurrent = sPath[indexMax].distance;
    }

    vertexList[currentVert].isInTree = true;
    nTree++;
    adjust_sPath();
  }
    displayPaths(); // Формирование отчёта
  nTree = 0;
  for (int i = 0; i < nVerts; i++)
    vertexList[i].isInTree = false;
}
```

Листинг 1. Реализация алгоритма поиска кратчайшего пути в графе

Затем на основе результатов алгоритма поиска динамически изменяем содержимое веб-страниц интернет-магазина, отображаемых пользователю: помещаем ссылки на самые посещаемые страницы выше, выделяем самые популярные товары и т.п.

Заключение

Таким образом, аналитика на графах действительно имеет право на существование: она легко встраивается, а за счет информации, которую она предоставляет, владелец интернет-магазина сможет более эффективно распределять товары по «виртуальным прилавкам» существенно поднимая

продажи. Также под данную аналитику можно разрабатывать дополнительные модули (например, автоматическое редактирования контента на сайте на основании полученных данных).

Список литературы

1. Демин, В. Поведенческая аналитика как инструмент эффективного продвижения сайта// Media Factory, 2015. – Режим доступа: http://factory.ws/article/analytic.
2. Федотов, А. Поведенческая аналитика в малом бизнесе // iForum-2016, stream: INTERNET BUSINESS, – Режим доступа: https://vk.com/video205562829_456239019?list=bc55b8ed6f7cddfc2b.
3. Седжвик, Р. Алгоритмы на Java. 4-е изд./ Р. Седжвик, К.Уэйн. – Вильямс, 2015. – 848 с.

УДК 04.522 , 316.77

СОЦИАЛЬНЫЙ СЕРВИС «FASTCAST»

П.М. Волков, И.С. Блохин, З.В. Степчева[24]

В работе оценены перспективы организации коммуникации пользователей в сети интернет посредством публикации аудио постов и аудио комментариев на базе разработанного сервиса «Fastcast» (www.fastcast.ru)

Введение

Активное развитие интернет-технологий на современном этапе привело к формированию в обществе новых способов свободного распространения и обмена информацией, оказывая влияние на формирование социального пространства граждан и их коммуникации.

Социальные сети, являясь новым форматом коммуникации, прочно вошли в нашу жизнь, и теперь мы уже не можем представить свое существование без активного их использования с целью обмена сообщениями, а также для активного отслеживания новостей, которые посвящены какой-либо тематике.

Важным аспектом использования социальных сетей является организация бизнес-коммуникации − площадки для развития собственного бизнеса, предоставляя широкие возможности для рекламы и продвижения своих товаров или услуг.

В России основными инструментами в социальных сетях являются: группы ВКонтакте, Facebook, Одноклассники; публичные страницы ВКонтакте, Facebook, Google+; мероприятия Facebook, ВКонтакте; «персонажи» (аккаунты) в социальных сетях; блоги и дневники на платформах Livejournal, Wordpress, LiveStreet; микроблоги, хэш-теги в Twitter; каналы YouTube [1].

Указанные социальные сети предоставляют, в основном, для этих целей средства для текстового (в виде сообщений) и визуального (в виде изображений) донесения информации до целевой аудитории.

Целью данной работы является исследование, как влияет аудио- способ донесения информации до пользователя социальной сети. Ведь пока ни

[24] *432027, Ульяновск, ул. Радищева, д. 102, МБОУ Лицей при УлГТУ, e-mail:stzv78@mail.ru*

одна социальная сеть не реализовала возможность относительно быстрого обмена аудио-постами, а также, не предоставляет возможности отправки аудио- комментария к имеющимся постам.

Для достижения поставленной цели необходимо решить ряд задач: 1) разработать в клиент-серверной архитектуре сервис, реализующий основные функции социальной сети для аудио-постов и комментариев; 2) экспериментально оценить перспективы общения с использованием аудиоинформации. А именно – выяснить реакцию пользователей на данный вид предоставления информации, оценить, как меняются взаимоотношения между людьми, которые пользуются нашим сервисом, а также выяснить, возможно ли с помощью данного способа передачи информации эффективно доносить рекламу до целевой аудитории.

1. Разработка программного обеспечения

Разработанный сервис для решения первой поставленной задачи построен в клиент-серверной архитектуре. В качестве реализации приложения-клиента была выбрана операционная система Android. Базовыми возможностями клиента являются:

- создание и отправка аудио-подкаста;
- просмотр своих и чужих аудио-подкастов;
- создание и отправка аудио-комментариев;
- просмотр своих и чужих аудио-комментариев;
- наличие возможности давать оценку постам пользователей (аналог «лайков»);
- поиск пользователей по уникальному имени («логину»);
- наличие возможности подписки на интересующих людей для формирования «ленты новостей».

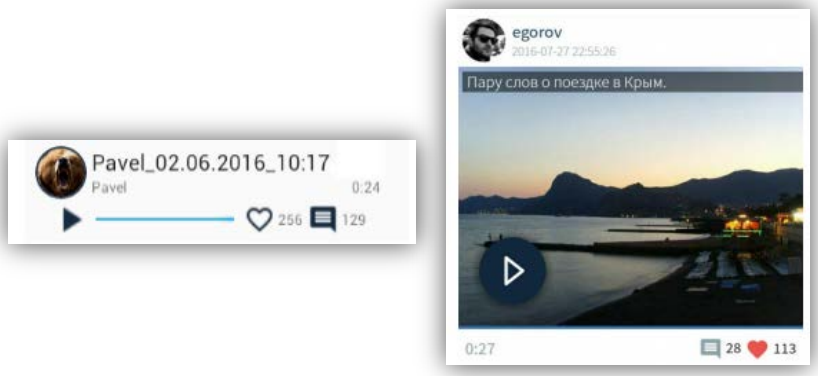
Помимо вышеперечисленных функций, приложение обеспечивает возможность смены изображений профиля пользователя, установку и изменение реального имени для повышения узнаваемости профилей интересующих пользователей. В качестве реализации сервера был арендован vps-сервер под управлением стабильного дистрибутива linux – debian. В качестве серверного языка программирования был выбран php, в качестве СУБД − MySQL.

2. Апробация программного обеспечения

Для экспериментальной апробации программного обеспечения была собрана фокус-группа из 10 человек, работавшая с сервисом на протяжении двух недель. Следует отметить, что в процессе апробации не изучалась возможность отправки личных сообщений – диалог происходил

в процессе создания подкастов и комментариев, как аудио-, так и текстовых.

Результаты апробации показали, что использование только аудио-постов не является привлекательным для пользователей (рис. 1, а).



а) б)

Рис.1. а) − первоначальный вид подкаста; b) − редизайн подкаста

Рис.3. Внешний вид приложения: а) – профиль пользователя, б) – визуализатор записи аудио-комментария/подкаста

Использование изображения и текстовой информации в подкастах (рис. 1, б) и в профиле пользователя (рис. 3) стимулирует интерес к их обсуждению в сети. К концу исследования на один подкаст приходилось около 13 % текстовых комментариев, остальные 87 % – представляли собой аудио- комментарии.

Заключение

Таким образом, в проекте экспериментально доказана привлекательность применения аудио способа коммуникации при условии дополнения контента визуальным рядом и текстом.

Список литературы

1. Архангельская, А.С. Социальные сети как площадка для бизнес-коммуникаций / А.С. Архангельская, И.Б. Архангельская // Вестник Нижегородского университета им. Н.И. Лобачевского, 2013. − №4 (2).− С. 186-189.
2. Том Демарко. Deadline. Роман об управлении проектами. Вершина, 2008. – 288 с.
3. Кононов, Н. В. Код Дурова. Реальная история "ВКонтакте" и ее создателя / Николай В. Кононов: Манн, Иванов, Фербер. – Москва, 2012. – 208 с.
4. Голощапов, А. «Google Android. Системные компоненты и сетевые коммуникации». БХВ-Петербург, 2012. – 384 с.
5. Мэтт Зандстра. PHP: Объекты, шаблоны и методики программирования. Вильямс, 2011. – 528 с.
6. Вайсфельд, М. Объектно-ориентированное мышление. Питер, 2014. – 304 с.

УДК 621.372

SEAM CARVING: АЛГОРИТМ СЖАТИЯ ИЗОБРАЖЕНИЙ С УЧЁТОМ КОНТЕКСТА

Д.А. Горшков, В.В. Снежкин[25], В.А Фолунин

В статье рассмотрен алгоритм сжатия изображений с учётом контента Seam Carving. Приведено общее описание алгоритма, указаны детали программной реализации. Показано теоретическое и эмпирическое обоснование вычислительной сложности процедуры удаления цепочки пикселей минимального веса.

Введение

В настоящее время в связи с широким распространением адаптивного дизайна возникает задача размещения в шаблоне документа различных иллюстраций, размер которых в ходе вёрстки должен быть скорректирован. При этом традиционные приёмы масштабирования изображений — сжатие и обрезка — обнаруживают известные недостатки: в первом случае нарушаются пропорции изображений, во втором — утрачивается часть контента [3].

В данной ситуации более перспективными являются так называемые алгоритмы контекстного масштабирования изображений (Content-Aware Scaling). Они основаны на анализе пикселей изображения и выделении наименее информативных областей, пригодных для удаления.

В настоящей работе обобщён опыт программной реализации одного из данных алгоритмов — Seam Carving.

1. Общее описание алгоритма

Алгоритм Seam Carving был разработан в Mitsubishi Electric Research Laboratories (MERL) в 2007 году [1].

Предположим, без потери общности, что выполняется сжатие изображения по вертикали. При этом требуется удалить некоторую цепочку пикселей, соединяющих левую и правую стороны изображения; из всех возможных цепочек удаляется та, которая содержит минимальное количество информации.

[25] *432027, Ульяновск, ул. Северный Венец, 32, УлГТУ, e-mail: snezhkinv@ya.ru*

Определим неявный сетчатый граф G на пикселях изображения следующим образом: от каждого пикселя (i; j) направлены ориентированные рёбра к пикселям (i - 1; j + 1), (i; j + 1), (i + 1; j + 1), как показано на рис. 1. Каждому ребру AB назначим целочисленный вес W_{AB} = E(B), где E(X) — функция энергии пикселя X, значение которой определяется как сумма квадратов разностей цветовых компонент пикселя X и четырёх пикселей, имеющих с ним общую сторону. Дополнительно в граф G добавляется стартовая вершина S, соединённая рёбрами нулевого веса со всеми вершинами, соответствующими пикселям левой границы изображения, и конечная вершина T, в которую идут рёбра нулевого веса из всех вершин, соответствующих пикселям правой границы изображения.

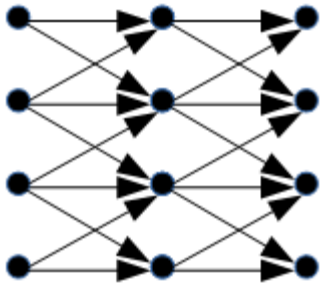


Рисунок.1. Ориентированный граф G на пикселях изображения

Путь S-T минимального веса будет содержать пиксели изображения, удаление которых минимальным образом сказывается на его содержании.

Нетрудно видеть, что описанный граф G является ациклическим орграфом, что позволяет использовать для поиска кратчайших путей линейный алгоритм, основанный на методе динамического программирования (см. главу 24.2 в [2]).

2. Программная реализация

При подготовке работы было разработано приложение на языке программирования C#, предоставляющее графический интерфейс для выбора файла изображения, задания настроек масштабирования с сохранением контента, выполнения и просмотра результата масштабирования.

В программе определён интерфейс AbstractSeamCarver, вводящий следующий набор методов:

- void Load(Bitmap img) — инициализация графа G по загруженному изображению;
- void CarveHorizontal() — поиск и удаление цепочки пикселей минимального веса, соединяющей левую и правую границы изображения;
- void UpdateEnergy() — обновление весов для пикселей, имеющих общую сторону с удалёнными.

Рассматриваемый интерфейс реализуется классами StandardSeamCarver и ShowSeamsSeamCarver. Первый из них содержит базовую реализацию рассматриваемого набора функций (уменьшения изображения без потери данных путём поиска минимальной цепочки в графе с последующим её удалением). Второй класс дополнительно поддерживает демонстрацию цепочек минимального веса пользователю перед удалением.

3. Теоретическое обоснование вычислительной сложности

Приведём асимптотическую оценку процедуры удаления одной цепочки пикселей минимального веса. Следует отметить, что в данную оценку не входит сложность процедуры инициализации графа G, выполняемой однократно для каждого изображения.

Введём обозначения: H – высота изображения, W – ширина изображения, P = HW – общее количество пикселей изображения.

- Поиск минимального пути между границами изображения предполагает обход всех пикселей с возможным обновлением предка для каждого из них за константное время — O(P).
- Удаление найденного пути из изображения предполагает сдвиг всех пикселей, находящихся ниже найденного пути, на один пиксель вверх, худший случай — удаление верхней границы изображения, — O(P).
- Обновление весов для пикселей, имеющих общую сторону с удалёнными, затрагивает не более 2W пикселей — O(W).

Общая оценка времени выполнения процедуры — O(P) + O(P) + O(W) = O(P).

4. Эмпирическое доказательство вычислительной сложности

При помощи средств языка программирования проведём измерение времени работы процедуры удаления минимального пути для $P = 200^2$, 300^2, …, 1200^2. Аппроксимируем полученные данные при помощи

линейной модели T = kP + b. При помощи метода наименьших квадратов определим значения коэффициентов k и b, минимизирующие невязку.

Из табл. 1 видно, что при k = 2 · 10-7 , b = 0,009 абсолютная погрешность расчётного значения T составляет не более 0,037, относительная погрешность — не более 0.179, что позволяет судить о фактическом линейном порядке роста сложности процедуры.

Таблица 1. Время работы процедуры

P	**$T_{факт}$, с**	**$T_{расч}$, с**	**ΔT**	**δT**
40000	0,016	0,017	0,001	0,063
90000	0,024	0,027	0,003	0,125
160000	0,037	0,041	0,004	0,108
250000	0,053	0,059	0,006	0,113
360000	0,074	0,081	0,007	0,095
490000	0,092	0,107	0,015	0,163
640000	0,123	0,137	0,014	0,114
810000	0,145	0,171	0,026	0,179
1000000	0,188	0,209	0,021	0,112
1210000	0,219	0,251	0,032	0,146
1440000	0,260	0,297	0,037	0,142

Заключение

В статье рассмотрен алгоритм сжатия изображений с учётом контента Seam Carving. Приведено общее описание алгоритма, указаны детали программной реализации. Показано теоретическое и эмпирическое обоснование вычислительной сложности процедуры удаления цепочки пикселей минимального веса.

Результаты использования разработанного приложения позволяют сделать вывод о том, что алгоритм Seam Carving демонстрирует удовлетворительные результаты при масштабировании ландшафтных изображений, содержащих множество фоновых областей и сравнительно небольшое количество контрастных объектов. С другой стороны, алгоритм плохо подходит для масштабирования портретов и изображений, насыщенных деталями.

Следует отметить, что алгоритм может быть применён для удаления с изображения нежелательных объектов путём присвоения областям с ними малых значений функции E.

Список литературы

1. Avidan S., Shamir A. Seam Carving for Content-Aware Image Resizing [Электронный ресурс] / Shai Avidan, Ariel Shamir. // IDC Herzliya, 2007. — Режим доступа: http://www.faculty.idc.ac.il/arik/SCWeb/imret/imret.pdf
2. Introduction to Algorithms / Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein. — 3rd ed. — Cambridge, MA: MIT Press, 2009. — 1292 p.
3. Лыков К. Алгоритм seam carving для изменения размера изображения [Электронный ресурс] / Кирилл Лыков. // Habrahabr, 2013. — Режим доступа: https://habrahabr.ru/post/183638/

УДК 621.372

ROOT CAUSE ANALYSIS - АНАЛИЗ ОСНОВНЫХ ПРИЧИН.

А.В. Гузаева[26]

Статья посвящена структурированному пошаговому методу выявления и анализа реальной причины проблемы и реагирования на нее. Данный процесс помогает заинтересованным сторонам понять причины проблемы в достаточной степени, чтобы достигнуть ее полного разрешения.

Введение

Отвечая на вопрос о качестве своей работы, большинство людей делают акцент не на ее положительных сторонах, а на желаемых улучшениях. Отправной точкой работы по улучшению обычно является проблема, обнаруженная и идентифицированная человеком или несколькими людьми, желающими добиться единой цели - решить проблему. Очевидно, что необходимым условием решения проблемы является определение и устранение ее основной причины. Только в этом случае мы можем быть уверены, что проблема действительно решена, и что она не нанесет нам неожиданный удар в будущем. Для того, чтобы быстро добраться до основной причины проблемы и улучшить процесс, специалисты по решению проблем используют эффективный подход, называемый анализ основных причин.

1. Проблемы и причины

Что такое проблема? Электронный словарь WordNet дает следующее определение проблемы: «Проблема - это затруднительная ситуация, которая должна быть разрешена». Понятно, что проблемы могут возникать в любой сфере человеческой жизни и принимать любую форму и вид. Можно выделить две позиции, которые будут относиться к проблеме, не важно технического, личного или организационного она характера.

- наличие проблемы означает состояние дел, находящихся в нежелательном состоянии или испытывающих затруднение;

[26] *432027, Ульяновск, ул. Северный Венец, 32, УлГТУ, e-mail:nastya19940311@yandex.ru*

- проблема представляет собой вызов, который подталкивает к ее решению, чтобы установилась более предпочтительная обстановка.

Проблемы повторяются по двум причинам:

- не устранен корень проблемы;
- корректирующие действия не устранили причины.

Из этого следует вывод: чтобы эффективно решать проблемы качества продукции или услуги. Нужно работать с *причинами.*

Проблема часто является результатом нескольких причин, или причин нескольких "уровней". Это означает, что некоторые причины влияют на другие причины, которые, в свою очередь, порождают наблюдаемую проблему. Все причины могут быть классифицированы следующим образом:

Симптомы. Они являются не настоящими причинами, а скорее признаками существования проблемы.

Причины первого уровня. Это причины, которые напрямую являются источником проблемы.

Причины более низкого уровня. Это причины, которые порождают причины первого уровня. Хотя они и не являются непосредственным источником проблемы, причины более низкого уровня являются связующим звеном в цепочке причинно-следственных связей, которая приводит к возникновению проблемы.

На самом нижнем уровне находится основная причина, которую необходимо найти. Основная причина - это "корень зла", который приводит в движение цепь причинно-следственных связей и так создает одну или сколько угодно проблем.

Лучшая аналогия будет из медицины. Мы можем бороться с симптомами заболевания, а можете вылечить причину этих симптомов т.е. болезнь. Например, вы сломали руку или ногу, и вы чувствуете сильную боль! Вы можете принимать болеутоляющее, которое на короткое время снимет боль, но то, что вам действительно нужно, так это обратиться к врачу. Точно так же и в нашей работе. Мы можем бесконечно заниматься, так сказать, «лечением симптомов», но так и не приступить к «лечению самой болезни». Это приводит к тому, что проблемы возникают вновь и вновь, ресурсы расходуются неэффективно, проблемы не исчезают, а клиенты всегда недовольны.

Из вышеизложенного следует вывод - нужно устранять основную причину (причины). Другие решения могут дать временное облегчение, но никогда не дадут длительного эффекта.

- Если вы устраните только симптомы, ситуация может ухудшиться. Проблема останется, но не будет легко различимых симптомов, которые можно контролировать.

- Устранение причин первого и более низкого уровня может временно облегчить проблему, но основная причина в итоге найдет другой путь, чтобы заявить о себе в виде другой проблемы.
- Когда вы устранили основную причину, следите за симптомами, чтобы удостовериться, что проблема не появилась вновь.

2. Анализ основной причины

Анализ основной причины (Root Cause Analysis) - это структурированное исследование, которое имеет целью выявление основной причины проблемы и определение мер, необходимых для ее устранения. Анализ основной причины - это не один инструмент или методика, а значительное число разных инструментов, зачастую используемых в комбинации. Это всегда комплексный и всесторонний анализ проблемы.

Анализ основных причин является популярным и часто используемым методом, который помогает людям ответить, в первую очередь, на вопрос о том, **почему проблема возникла.** Он стремится определить происхождение проблемы, используя определенный набор шагов, помогающих найти основную причину проблемы. Цель:

1. Определить, что произошло.
2. Определить, почему это произошло.
3. Выяснить, что нужно сделать, чтобы уменьшить вероятность того, что это произойдет снова.

Существует три основных вида причин:

1. Физические причины. Материальный дефект, который приводит к проблеме. Например: отказ тормозов автомобиля.
2. Человеческие причины. Как правило, это человеческая ошибка. Часто такие причины приводят к физическим причинам. Например, никто не заполнил бачок тормозной жидкостью, что привело к отказу тормозов.
3. Организационные причины. Поведение системы или процесса. Например, ни один человек не был ответственным за техническое обслуживание транспортного средства, и все решили, что кто - то другой заполнил бачок тормозной жидкостью.

Как правило, существует несколько главных причин.

Итак, анализ основных причин включает в себя пять идентифицирующих шагов.

Первый шаг. Необходимо определить проблему. Для этого нужно ответить на два основных вопроса: Что вы видите или что происходит? Каковы конкретные симптомы?

Второй шаг. Сбор информацию о проблеме. Вам необходимо полностью проанализировать ситуацию, прежде чем двигаться дальше. Для того, чтобы максимально увеличить эффективность Вашего RCA, нужно собрать вместе экспертов и сотрудников, кто понимает ситуацию. Люди, которые наиболее знакомы с этой проблемой могут помочь привести вас к лучшему пониманию вопросов. Выполнить данный шаг поможет известный метод описания проблемы «5W2H». Метод заключается в том, что необходимо ответить на следующий перечень вопросов:

Who Кто жалуется? Для кого конкретно это является проблемой? Идентифицируйте этих людей (конкретно имя, контактные данные);

What Опишите максимально точно в чем проблема;

When Когда проблема была обнаружена (время, время года)?

Where Где проблема обнаружена?

Why Почему это является проблемой для конкретного человека, службы, организации?

How При каких обстоятельствах была обнаружена проблема?

How many Сколько брака? Каков масштаб проблемы? Числовой показатель проблемы.

Третий шаг. Определение факторов, повлиявших на появление проблемы. Любая проблема может иметь несколько причин и невероятное количество комбинаций этих причин. Поэтому при анализе причин важно ответить на следующие вопросы:

Какая последовательность событий привела к этой проблеме?

Какие условия позволили данной проблеме появиться?

Какие сопутствующие проблемы возникли вместе с обсуждаемой центральной проблемой?

Чтобы системно и структурировано подойти к этому вопросу можно использовать следующие диаграммы:

1. Диаграмма причинно-следственных связей - диаграмма Ишикавы («Рыбья кость»);

2. Диаграмма связей (Mind Map) - способ проведения мозгового штурма, сопровождающийся визуализацией связанных ассоциаций и явлений.

Так же в данном шаге можно смело использовать метод «Пять почему?». Когда появляется проблема, просто продолжайте задавать вопрос «Почему» до тех пор пока не найдете корень проблемы. Если причина проблемы найдена, то корректирующие действия станут очевидными. Каждый раз, когда вы задаете вопрос «Почему?», в ответ должны засчитываться только факты, только то, что действительно произошло. Если хоть раз вы примете свое допущение за реальный факт,

то вы можете уйти в сторону от решения коренной проблемы. Продолжайте спрашивать «Почему?» столько раз, сколько потребуется, чтобы дойти до сути. Когда вопрос «Почему?» больше нельзя задать, то это и есть причина проблемы. Идея инструмента в том, что корень проблем не так очевиден, как это может показаться на первый взгляд.

Шаг четвертый: Определение причин (-ы). Необходимо ответить на вопросы: Почему причинный фактор существует? Какова реальная причина возникновения проблемы?

Используйте те же инструменты, используемые для выявления причинных факторов (на третьем этапе), чтобы посмотреть на корни каждого фактора. Эти инструменты предназначены, чтобы побудить вас копать глубже на каждом уровне причины и следствия.

Шаг пятый. Разработка корректирующих действий. Здесь поможет полезный инструмент «Анализ воздействий» (Impact Analysis). Он поможет изучить возможные позитивные и негативные последствия изменения на различных частях системы или организации.

Заключение

В заключении хочется отметить, что выбор подходящего метода и комплексное использование разных инструментов в ходе всего процесса анализа основной причины является непростой задачей. Перед тем как приступить к выбору инструмента, следует обратить внимание на следующее предостережение. Вашей главной задачей является нахождение основной причины имеющейся проблемы и ее устранение. Инструменты, упомянутые в данной статье, являются всего лишь подспорьем в достижении этой цели. Слишком большой акцент на инструменты может привести к тому, что сама проблема окажется на втором плане.

Список литературы

1. Root Cause Analysis. Tracing a Problem to its Origins. [Электронный ресурс]. — URL: https://www.mindtools.com/pages/article/newTMC_80.htm;
2. Ammerman, Max. 1998. The Root Cause Analysis Handbook: A Simplified Approach to Identifying, Correcting, and Reporting Workplace Errors. Quality Resources;
3. Анализ первопричины. [Электронный ресурс]. — URL: http://www.pmtoday.ru/project-management/other-topics/root-cause-analyze.html.

УДК 621.397

РАЗРАБОТКА ПРОГРАММЫ СОПРЯЖЕНИЯ ВЕБ-КАМЕРЫ И ПЕРСОНАЛЬНОГО КОМПЬЮТЕРА ДЛЯ ИССЛЕДОВАНИЯ ПАРАМЕТРОВ ВИБРАЦИИ

Р.И. Замалетдинов[27], Д.А. Евсевичев[28], О.В. Максимова [29]

В статье представлен программный продукт, позволяющий в реальном времени отслеживать и измерять вибрационные параметры стендовой лабораторной конструкции. Применение разработанной системы позволит оценивать вибропрочность конструкций радиоэлектронной аппаратуры.

Введение

Одной из важнейших проблем современного приборостроения является определение параметров вибропрочности как отдельных узлов, так и конструкции в целом. С целью решения поставленной задачи был предложен метод расчета, основанный на изменении положения наблюдаемой точки с использованием интерфейса веб-камеры.

1. Реализация проекта сопряжения веб-камеры и персонального компьютера

Для реализации проекта была выбрана среда разработки Microsoft Visual Studio. Microsoft Visual Studio – это линейка продуктов компании Microsoft, включающих интегрированную среду разработки программного обеспечения и ряд других инструментальных средств. Данные продукты позволяют разрабатывать как консольные приложения, так и приложения с графическим интерфейсом, в том числе с поддержкой технологии Windows Forms, а также веб-сайты, веб-приложения и другое [1].

При решении поставленной задачи оптимально использовать для представления информационных материалов язык C#, который является языком высокого уровня и позволяет быстро и эффективно создавать приложения.

[27] *432027, Ульяновск, ул. Северный Венец, 32, УлГТУ, e-mail: rishattz@mail.ru*

[28] *432027, Ульяновск, ул. Можайского, 8/8, УИГА, e-mail: denistk_87@mail.ru*

[29] *432027, Ульяновск, ул. Северный Венец, 32, УлГТУ, e-mail: first32007@yandex.ru*

Преимущества C# по сравнению с аналогичными программными продуктами: быстрота разработки приложения; высокая производительность разработанного приложения; низкие требования разработанного приложения к ресурсам компьютера; наращиваемость за счет встраивания новых компонентов и инструментов в среду C#; возможность разработки новых компонентов и инструментов собственными средствами C# (существующие компоненты и инструменты доступны в исходных кодах); удачная проработка иерархии объектов.

Система программирования C# рассчитана на программирование различных приложений и предоставляет большое количество компонентов.

Разработанная программа позволяет подключить веб-камеру к персональному компьютеру. Алгоритм работы программы представлен на рисунке 1.

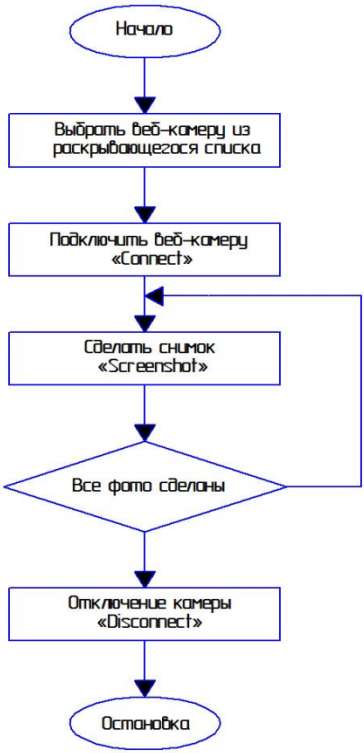


Рис. 1. Алгоритм работы программы

Окно программы содержит следующие элементы (рисунок 2):

1 – Выпадающий список доступных для выбора веб-камер.
2 – Подключение веб-камеры.
3 – Отключение веб-камеры.
4 – Окно предпросмотра.
5 – Снимок с веб-камеры.

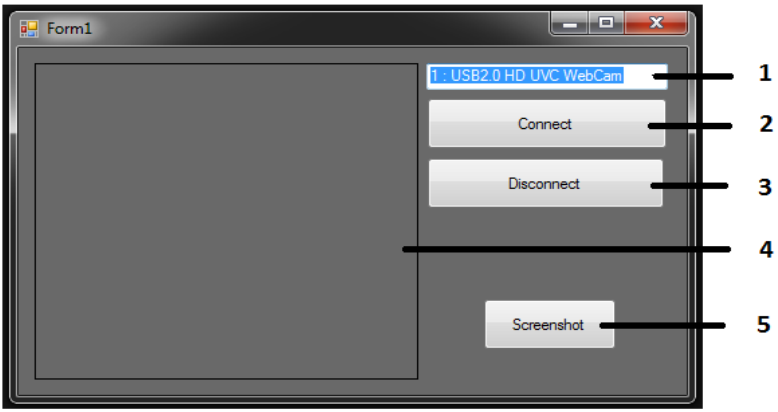


Рис. 2. Интерфейс программы сопряжения веб-камеры и персонального компьютера

Программный код представляет собой исходный код программы, сохраненный в текстовых файлах с расширением .cs:

– Form1.cs — содержит файл, в который записан код формы. Содержит функции, реализующие описанный алгоритм, а также подключает библиотеки для работы с вебкамерой;

– Form1.Designer.cs — содержит элементы окна (текстовые надписи, кнопки, списки), т.е. инициализирует все графические элементы окна программы.

Для каждого элемента окна Visual C# Express добавляет в этот файл соответствующий фрагмент кода, который описывает данный экземпляр элемента. Файл используется только самой системой и служит для представления в коде тех элементов, которые добавляются в область конструктора. Обычно код в этом файле не изменяется и не добавляется. Фактически, именно из него система получает информацию о представлении конструктора.

– Program.cs – содержит стандартный метод Main() и код для автоматического создания экземпляра класса Form. Он является точкой входа приложения и выполняет визуализацию интерфейса и алгоритм. То есть за запуск всей программы отвечает именно этот файл.

Описание работы программы сопряжения веб-камеры и персонального компьютера представлено в листинге 1.

```
namespace WebCam
{
    public partial class Form1 : Form
    {
        FilterInfoCollection videoDevices;
        public Form1()
        {
            InitializeComponent();

            videoDevices = new
FilterInfoCollection(FilterCategory.VideoInputDevice
);
            for (int i = 1, n =
videoDevices.Count; i <= n; i++)
            {
                string cameraName = i + " : " +
videoDevices[i - 1].Name;

comboBoxWebcamList.Items.Add(cameraName);
            }
        }
        private void StartCameras()
        {
            try
            {
                VideoCaptureDevice videoSource =
new
VideoCaptureDevice(videoDevices[comboBoxWebcamList.S
electedIndex].MonikerString);

videoSourcePlayerWebCam.VideoSource = videoSource;
                videoSourcePlayerWebCam.Start();
            }
            catch { MessageBox.Show("Выберите
камеру"); }
        }
        private void button1_Click(object sender,
EventArgs e)
        {
            StartCameras();
```

```
            //  button1.Enabled = false;
          }
                    private void button2_Click(object
sender, EventArgs e)
          {
              SaveFileDialog sf = new
SaveFileDialog();
              sf.Filter = "Imagenes JPG | *.jpg";
              sf.ShowDialog();
              if (sf.FileName != null)
              {
                  Bitmap img =
videoSourcePlayerWebCam.GetCurrentVideoFrame();
                  img.Save(sf.FileName,
System.Drawing.Imaging.ImageFormat.Jpeg);
                  img.Dispose();
                                                  }
          }
          private void button3_Click(object sender,
EventArgs e)
          {

videoSourcePlayerWebCam.SignalToStop();
          }
      }
}
```

Листинг 1. Работа основной процедуры программы сопряжения веб-камеры и персонального компьютера

Заключение

Программа позволяет подключить веб-камеру к персональному компьютеру, а также делать снимки. В последствии ее можно будет использовать для оптического метода измерения амплитуды виброперемещений исследуемого объекта. При этом необходимо будет учитывать расстояние от камеры до объекта и угол наклона для получения истинного значения амплитуды его колебаний.

Список литературы

1. Ник Рендольф, Дэвид Гарднер, Майкл Минутилло, Крис Андерсон. Visual Studio 2010 для профессионалов = Professional Visual Studio 2010. — М.: «Диалектика», 2011. — С. 1184. — ISBN 978-5-8459-1683-9.

УДК 621.372

АЛГОРИТМ ПОИСКА А* НА ПРИМЕРЕ ИГРЫ «ПЯТНАШКИ»

И.И. Исхаков[30], Д.В. Сазонов, В.А. Фолунин

В статье рассмотрен алгоритм А* для поиска оптимального решения игры-головоломки «Пятнашки». Разобрана работа алгоритма и техническая реализация, выполнен анализ теоретической сложности алгоритма. Рассмотрен ряд эвристик для алгоритма, приведены результаты сравнения быстродействия алгоритма при их применении.

Введение

Процесс поиска оптимального решения в игре «Пятнашки» можно представить как задачу поиска на графе. Вершинами графа будут состояния поля, полученные в результате перемещения костяшек. Поиск решения можно свести к поиску терминального состояния игрового поля. Для решения задачи поиска терминальной вершины на графе можно использовать алгоритмы полного перебора (DFS или BFS), но количество возможных решений окажется очень большим, вследствие чего время работы программы окажется для нас неприемлемым.

Алгоритм А* позволяет существенно сократить количество состояний для перебора за счёт использования эвристической функции, обобщающей различные предположения о направлении оптимального пути. В качестве такой функции предлагается брать предполагаемое количество перестановок, необходимых для получения терминального состояния [3].

Этот алгоритм был впервые описан в 1968 году Питером Хартом, Нильсом Нильсоном и Бертрамом Рафаэлем. Данный алгоритм является расширением алгоритма Дейкстры для поиска кратчайших путей во взвешенном графе, который был создан в 1959 году. Новый алгоритм достигал более высокой производительности с помощью эвристики. В оригинальной работе он упоминается как «алгоритм А», но в настоящее время более известен под названием А* [2].

[30] *432027, Ульяновск, ул. Северный Венец, 32, УлГТУ, e-mail: ilnur11922@gmail.com*

1. Описание алгоритма

Алгоритм A* предполагает наличие двух списков вершин графа: открытого и закрытого. В первом находятся вершины, еще не проверенные алгоритмом, а во втором – те вершины, которые уже встречались в ходе поиска решения. На каждом новом шаге из списка открытых вершин выбирается вершина с наименьшим весом. Вес F каждой вершины вычисляется как сумма расстояния от начальной вершины до текущей G и эвристического предположения о расстоянии от текущей вершины до терминальной H. Таким образом, $F_i = G_i + H_i$, где i – текущая вершина (состояние игрового поля).

Для «Пятнашек» можно сделать несколько эвристических предположений:

- для достижения терминальной вершины необходимо выполнить перемещений не меньше, чем сумма ортогональных расстояний каждой костяшки до своей терминальной позиции;
- для достижения терминальной вершины необходимо выполнить перемещений не меньше, чем количество костяшек, находящихся не на своих местах;
- для достижения терминальной вершины необходимо выполнить перемещений не меньше, чем сумма эвклидовых расстояний каждой костяшки до своей терминальной позиции;

Более подробно ознакомиться с перечнем эвристических функций можно в источнике [1], глава 2 (Heuristics).

На основе этих предположения будем рассчитывать значение H. Функцию G следует рассчитывать как количество сделанных перестановок, считая от начальной вершины до текущего состояния. Расчёт значения H выполняется для каждой вершины при инициализации.

В качестве примера на рис. 1 показан граф для эвристики «количество костяшек не на своем месте».

Из начальной вершины проводятся рёбра в дочерние вершины, соответствующие конфигурациям поля, которые могут быть получены перемещением костяшек на пустую клетку. Каждая вершина имеет ссылку на родительскую вершину. Выполняется перебор дочерних вершин. Каждая дочерняя вершина проверяется на предмет наличия в списке закрытых. Если вершина не встречалась ранее, она перемещается в список открытых вершин. Для неё рассчитывается эвристическое расстояние до терминальной вершины.

В качестве списка открытых вершин удобно использовать очередь с приоритетом, так как на каждом шаге нам нужно брать вершину с

наименьшим значением F. В качестве списка закрытых вершин удобно использовать множество, так как нам нужно знать, обрабатывали ли мы данное состояние поля или нет. Выбор этих структур данных значительно ускорит время работы алгоритма.

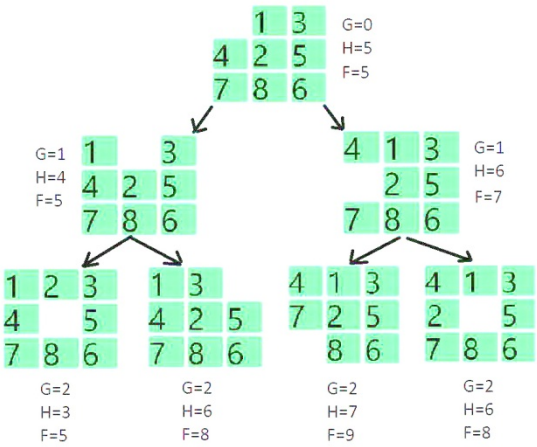


Рисунок 3. Граф состояний игрового поля

Алгоритм будет перебирать состояния до тех пор, пока из списка открытых вершин мы не извлечём терминальную. Но прежде чем начать искать решение, необходимо проверить, имеет ли решение данная начальная конфигурация костяшек на поле. Пусть фишка с числом i расположена до (если считать слева направо и сверху вниз) n_i фишек с числами, меньшими i. Также пусть e – номер ряда пустой клетки (считая с 1). Если сумма

$$N = \sum_{i=1}^{m} n_i + e$$

является нечётной, то решения головоломки не существует.

Данная формула используется не только для поля 4×4, но и для всех полей, размерность которых чётна. Если же перед нами поле с нечётной размерностью, то в формуле не нужно учитывать слагаемое e, то есть номер строки пустого элемента. Формула будет иметь вид:

$$N = \sum_{i=1}^{m} n_i$$

2. Техническая реализация

В ходе изучения алгоритма было разработано приложение на языке программирования Java с графическим интерфейсом. На входе программа получает состояние игровой доски с расположением каждой костяшки и пустого поля. Формирование начального расположения костяшек реализовано следующими способами:

- загрузка исходного состояния доски из файла формата txt, где прописаны размеры и элементы игрового поля;
- случайное заполнение начального состояния;
- пользовательское заполнение поля при помощи перемещения элементов доски курсором мыши.

Имеется возможность выбора эвристической функции и просмотра графиков сравнения эвристик по количеству выполненных перестановок.

Реализация алгоритма осуществляется методами двух классов.

Класс Board предназначен для хранения информации о конкретном состоянии игровой доски: расположения элементов на поле, позиции пустого элемента, состояния-родителя, то есть состояния, из которого мы получили текущее, и значений эвристических функций.

Класс Astar выполняет поиск оптимального решения. Конструктор этого класса принимает начальное состояние поля и находит решение.

3. Теоретическое обоснование сложности алгоритма

Приведём асимптотическую оценку конструктора класса Astar, который выполняет поиск оптимального пути.

Введем обозначения: N – размерность поля, M – количество элементов в закрытом списке, K – количество элементов в открытом списке.

Для начала нужно проверить, является ли начальное поле разрешимым – необходимо обойти все элементы двумерного массива – $O(N^2)$; если поле разрешимо, то положить начальное состояние поля в открытый список – O(1) и запустить цикл с условием «пока не достигнуто терминальное состояние поля», в котором нужно выполнить следующие действия:

1. Извлечь элемент (состояние поля) из вершины открытого списка (приоритетной очереди) – O(1)
2. Извлеченный элемент добавить в закрытый список (множество) – O(logM)
3. Проверить, является ли извлеченный элемент терминальным состоянием доски – O(1), так как признаком терминального состояния служит равенство эвристической функции H нулю.
4. Сгенерировать дочерние элементы путем передвижения костяшек на пустой элемент – O(1)

5. Проверить дочерние элементы на вхождение в закрытый список – $O(N^2 \cdot \log M)$, так как двумерный массив (N^2) необходимо сравнить logM раз в закрытом списке (множестве)
6. Если дочерний элемент не был обнаружен в закрытом списке, то добавить его в открытый список – O(logK)

Таким образом, асимптотическая сложность алгоритма – $O(M \cdot (N^2 \cdot \log M + \log K))$, так как всего итераций будет выполнено M раз, пока не достигнем терминального состояния.

Стоит сказать, что временная сложность алгоритма A* зависит от эвристики. В худшем случае число вершин, исследуемых алгоритмом, растёт экспоненциально по сравнению с длиной оптимального пути, но сложность становится полиномиальной, когда эвристика удовлетворяет следующему условию:

$$|h(x) - h^*(x)| \leq O(\log h^*(x))$$

где h* — оптимальная эвристика, то есть точная оценка расстояния из вершины x к цели. Другими словами, ошибка h(x) не должна расти быстрее, чем логарифм от оптимальной эвристики.

4. Сравнительный анализ эвристик

В данной работе было рассмотрено 3 вида эвристических функций:

- Манхэттенское расстояние (manhattan) – сумма ортогональных расстояний каждой костяшки до своей терминальной позиции
- Костяшки не на своих местах (chiponpos) – количество костяшек, которые располагаются не на своих терминальных позициях
- Евклидово расстояние (eucldist) - сумма эвклидовых расстояний каждой костяшки до своей терминальной позиции

Ниже приведена таблица сравнения использованных эвристик на практике, где M – количество итераций в цикле (количество рассмотренных состояний), T – время работы алгоритма в наносекундах, C – число ходов до терминального состояния.

Таблица 1. Сравнение эвристик

C	M manhattan	T manhattan	M chiponpos	T chiponpos	M eucldist	T eucldist
24	428	1182989	7301	100683465	1844	13950674
26	241	513473	13416	57452501	2283	10190416
28	821	2553379	57958	365898219	3855	20376626
30	369	1143512	209966	174327660	13401	71210688

34	1059	3599303	106121	658701717	3677	43886502
40	1727	6189923	54384	301890039	15837	82874337

Анализируя таблицу, можно сделать вывод, что наиболее удачная эвристика для решения игры «Пятнашки» – Манхэттенское расстояние, которая за наименьшее число итераций находит терминальное состояние поля.

Заключение

В статье рассмотрен алгоритм A* на примере игры «Пятнашки». Приведено общее описание алгоритма, указаны детали технической реализации. Показано теоретическое обоснование сложности алгоритма.

Результаты выполненной работы показывают, что на практике алгоритм A* выдает положительные показатели по времени работы, поэтому он так популярен в разработке игр.

Для данного алгоритма существует ряд улучшений, с которыми можно ознакомиться в источнике [1].

Список литературы

1. Amit P. Amit's A* Pages [Электронный ресурс] / Amit Patel. //Red Blob Games, 2016. — Режим доступа: http://theory.stanford.edu/~amitp/GameProgramming/
2. Лестер П. Алгоритм A* для новичков [Электронный ресурс] / Патрик Лестер. // TU Clausthal, 2004. — Режим доступа: http://www2.in.tu-clausthal.de/~zach/teaching/info_literatur/A_Star/A_star_tutorial/aStarTutorial_rus.htm.html
3. Попов В. Интеллектуальные системы. A* и игра «Пятнашки» [Электронный ресурс] / Владимир Попов. // Dokwork, 2012. — Режим доступа: http://www.dokwork.ru/2012/03/blog-post.html

УДК 621.372

СТРАТЕГИЯ ПРИНЯТИЯ РЕШЕНИЯ «ЗА И ПРОТИВ»

Л.Р. Латыпова[31]

В данной статье рассматривается стратегия принятия решения «за и против». Анализ стратегии основан на рассмотрении ее плюсов и минусов, а так же на разработке инструмента для применения стратегии на практике.

Введение

Многие из нас сталкивались с проблемой принятия решения. Очень часто мы боимся принять неправильное решение, и тратим много времени на рассмотрение различных вариантов.

Однако, некоторые, напротив, абсолютно уверены в своем решении и не думают о возможных последствиях.

Для того чтобы избежать попадания в подобные ситуации используется стратегия принятия решения «за-против». Данная стратегия является простой, но в, то, же время достаточно эффективной. Она позволяет оценить ситуацию с разных сторон, а так же принять быстрое и правильное решение.

В данной статье будет рассматриваться инструмент для применения стратегии принятия решения «за и против», реализация данного инструмента, а так же будет проведен анализ плюсов и минусов стратегии.

1. Инструмент

В данном параграфе рассмотрим, как использовать данную стратегию на практике.

Оценка всех «плюсов» и «минусов» ситуации может ускорить процесс принятия решения, а так же повлиять на понимание ситуации. Использование списка факторов «за и против» поможет принять объективное решения, основанное на реальных фактах.

Для начала использования стратегии необходимо:

- сверху на листке бумаги написать решение, которое нужно принять;
- поделить лист на 2 колонки: плюсы и минусы;

[31] *432027, Ульяновск, ул. Северный Венец, 32, УлГТУ, e-mail: lesyan2594@mail.ru*

- написать в колонке плюсы все положительные последствия относительно принятия данного решения, а в колонки минусы – отрицательные;
- присвоить каждому пункту «за и против» значение, в зависимости от степени его важности. Например, +5 (очень важно), – 1 (не совсем приемлемо). Оценка должна быть объективной.
- подсчитать сумму в каждой колонке;
- сложить результаты обеих колонок.

Если окончательный результат положительный, то вы идете в правильном направлении, и вам стоит принять данное решение. Если результат отрицательный, рекомендуется обратить внимание на другие варианты. Однако если возникают сомнения относительно точности и верности результата, следует перепроверить, все ли факторы были учтены при принятии решения.

1.2. Практическое применение инструмента

Представьте, что вы управляете отделом маркетинга. Несколько ваших сотрудников хотят работать удаленно несколько дней в неделю. Несмотря на то, что работа на дому имеет свои преимущества, существует так же много недостатков. Следовательно, проанализируем плюсы и минусы работы на дому.

Таблица 1. Плюсы и минусы удаленной работы

Удаленная работа	
Плюсы	**Минусы**
Работа дома сократит уровень стрессовых ситуаций и **увеличит** производительность. (+5)	Некоторым сотрудникам будет **сложно** продолжить работать с другими на расстоянии. (-4)
Удаленная работа подразумевает под собой **гибкий график**, который особо важен для работающих родителей. (+4)	Компании придется **закупить** соответствующее оборудование для работы на дому. (-4)
Появляется возможность нанять **высококвалифицированных сотрудников**, которые живут в другом городе или у которых нет возможности приезжать каждый день в офис. (+4)	Компания **не всегда сможет контролировать**, чем сотрудники занимаются дома в рабочее время. (-3)
Сотрудники, которых обычно отвлекает офисный шум, дома будут работать **продуктивнее**. (+5)	Компания может **потерять контроль над данными**, так как оборудование и документы находятся за пределами офиса. (-3)

Финальный результат +4, это значит, что имеет смысл разрешить сотрудникам работать несколько дней в неделю работать удаленно.

2. Ограничения и требования

Представленный метод намечен для его автоматизации в среде OwnWIQA. Принято решение при разработке использовать следующие ограничения и требования.

1. Ограничить шкалу оценивания

Рекомендуется использовать шкалу от 1-5. Она позволит наиболее объективно оценить факторы (оценка 5 - все отлично, оценка 4 - хорошо, но можно и лучше).

2. Присвоить значения после проведения «мозгового штурма»

Рекомендуется сначала составить список плюсов и минусов, а потом уже присваивать им определенные баллы. Оценка будет более точной, если будет видна вся картинка в целом.

3. Присвоить самые низкие и самые высокие баллы

Рекомендуется сначала присвоить пунктам «за и против» самые высокие и самые низкие баллы, т.е. сначала отметить все пункты, которым вы присваиваете 5 или 1 балл. Затем отмечайте пункты по 4, 3 и 2 балла. Присваивая баллы одного уровня (например, 5) вы можете сравнить факторы и удостовериться, правильное ли решение вы приняли, т.е. равноценны по важности оба эти фактора.

3. Метод формирования оценки

Схема формирования оценки для принятия решения.

1. Определить совокупность плюсов $\{P_i\}$

2. Определить совокупность минусов $\{N_j\}$

3. Используя разработанную типологию потенциальных «плюсов» и «минусов» приписать каждой из этих составляющих нормирующий коэффициент.

4. $O = \sum_{i=1}^{I} K_1 \; P_i - \sum_{j=1}^{J} K_2 \, N_j$,

где O – результат,

K – нормирующий коэффициент

5. Если результат отрицательный O<0, то данное решение принимать не стоит.

6. Если результат положительный и $O>N^p$ (минимальное положительное значение, при котором имеет смысл принять решение), то выбранное вами решение является правильным.

Реализация данной стратегии подразумевает под собой разработку пустой таблицы, в которую пользователь может занести данные, т.е.

присвоить факторам соответствующие баллы, которые будут подсчитаны автоматически.

Заключение

В заключение статьи приведем выявленные плюсы и минусы стратегии принятия решения «за и против».

Плюсы

- Данная стратегия позволяет посмотреть на ситуацию с разных сторон, рассмотреть различные варианты решения проблемы и сделать правильный выбор.
- Детальное изучение всех плюсов и минусов увеличит понимание ситуации и поможет избежать *«страха принятия решения».*
- Стратегия «за и против» подразумевает под собой исключительно *объективную оценку* всех положительных и отрицательных моментов. Т.е. так называемое «хорошее предчувствие» не должно повлиять на принятие решения.
- Данная стратегия эффективна для принятия быстрых решений.
- Позволяет сократить время принятия решения, путем установления *лимита на обсуждения*.
- Побуждает сотрудников к активным обсуждениям *(«мозговой штурм»).*
- Список «за и против» позволяет определить насколько важен тот или иной фактор для принятия решения, путем присвоения ему определенного значения.

Минусы

- Не подходит, если необходимо сравнить сразу несколько вариантов принятия решений, а так же если требуется глубокое изучение некоторых из вариантов. В этом случае лучше использовать *«дерево принятия решений».*
- Некоторые факторы могут быть упущены.
- Необъективная оценка некоторых факторов.

Список литературы

1. https://www.mindtools.com/pages/article/newTED_05.htm
2. http://www.ehow.com/info_8179604_pros-cons-qualitative-quantitative-research.html
3. https://www.researchoptimus.com/blog/qualitative-vs-quantitative-research/
4. http://leandecisions.com/2012/09/how-to-create-an-effective-weighted-pro-con-list.html

УДК 621.372

ПЕРЕДВИЖЕНИЕ РОБОТА НА РОЛИКОНЕСУЩИХ КОЛЕСАХ БЕНГТА ИЛОНА

Д.Г. Лямкин[32]

В статье рассматривается решение одной из главных проблем передвижения роботов, а именно, устранение поворотного скраба, то есть трения, которое создается колесами в процессе поворота робота при их боковом волочении по окружности. В качестве решения изучается принцип работы роликонесущих колес Бенгта Илона.

Введение

Одной из главных проблем передвижения роботов всегда было трение колес в процессе поворота робота. Такое явление получило название «поворотный скраб». С целью решения данной проблемы, мы рассмотрели принцип вращения роликонесущих колес, их способность передвигать робота с характерной наименьшей силой трения, сопротивляющейся движению робота, а так же запрограммировали траекторию разворота робота на данном типе колес.

1. Принцип вращения типичной ходовой части с однобортным управлением

Рассмотрим движение ходовой части робота с однобортным управлением с целью аналитики причин появления поворотного скраба, а также дальнейшего сравнение с иным типом ходовой части робота.

Причиной появления поворотного скраба является возникновение поворотного момента в связи с приложением силы к каждому колесу ходовой части робота, в итоге все они подвержены боковому волочению.

Рассмотрим схему взаимодействия движения колес и появления поворотного скраба (рисунок 1).

Параллельно расположенные колесо 1 и колесо 2, с помощью приложенных к ним линейных сил (сила 1 и сила 2), соответственно, создают поворотный момент ходовой части вокруг точки поворота. Эти

[32] *432027, Ульяновск, ул. Северный Венец, 32, УлГТУ, email: lyamkind95@mail.ru*

колеса не подвергаются волочению при выполнении поворота, следовательно они не способствуют возникновению поворотного скраба.

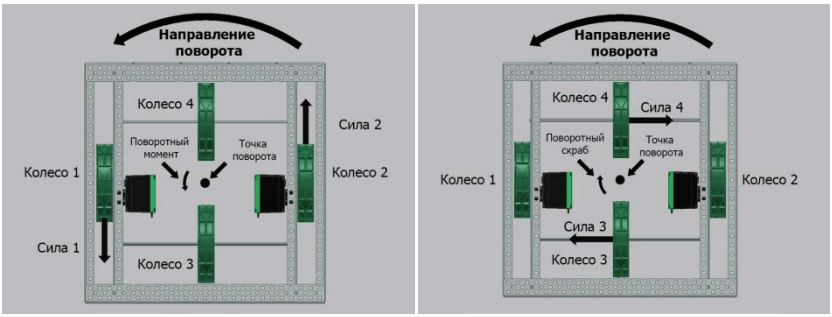


Рис.1. Схема взаимодействия движения колес и появления поворотного скраба

Колесо 3 и колесо 4, в свою очередь, не принимают участия в создании поворотного момента, однако скользят в сторону поворота, от того и начинается возникновение силы трения приложенной к колесу 3 и колесу 4 (сила 3 и сила 4 соответственно), что в конечном итоге способствует возникновению скраба.

2. Особенность роликонесущих колес Бенгта Илона

За счет наличия роликов на основании соприкосновения колес боковое трение практически пропадает, следовательно, при движении колес «вперед/назад» ролики направления «влево/вправо» не подвергаются боковому волочению.

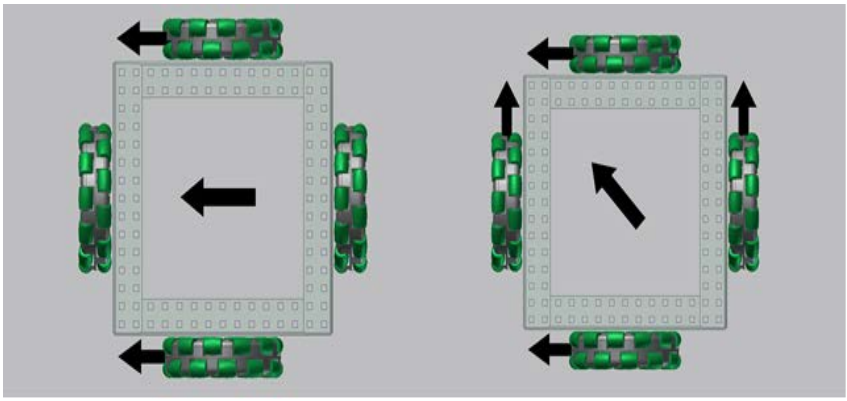


Рис.2. Схема силовой передачи для передвижения роликонесущей ходовой части

За счет подключения обоих комплектов колес к силовой передаче движение робота может быть реализовано в любом направлении.

Пример такой реализации изображен на изображении (рисунок 2), при том, при движении ходовой части роликонесущих колес в направлении диагонали, сила трения сводится к минимуму.

3. Программирование робота на роликонесущих колесах в робосимуляторе V-Rep Pro Edu

Для движения робота по траектории эллипс роботу необходимо придать разную скорость вращения колёс Илона. Для вращения вокруг перпендикулярной площади поверхности оси вращения, находящейся между колёсами 3 и 4, придаём колесу 3 -5v5=-50 см/с, колесу 4 2v5-el=44 см/с, колесу 2 -5v5=-50 см/с и колесу 1 максимальное значение -6v5=-60 см/с. Таким образом колесо 4 является единственным в своём направлении и, собственно, колесо 4 является поворачиваемым на острие эллипса.

Код данного алгоритма приведен ниже (листинг 1).

```
for i=1,4,1 do
      simSetNameSuffix(suffixNb+i-1)
      omniPads[i]=simGetObjectHandle('OmniWheel_regul
arRotation')
end
simSetNameSuffix(suffixNb)
v5=10 el=6
simSetJointTargetVelocity(omniPads[1],-6*v5)
simSetJointTargetVelocity(omniPads[2],-5*v5)
simSetJointTargetVelocity(omniPads[3],-5*v5)
simSetJointTargetVelocity(omniPads[4],2*v5-el)
simWait(e)
```

Листинг 1. Программный код движения робота по траектории эллипс

Заключение

Исследования силы трения колес в процессе поворота робота привели к рекомендуемому использованию роликонесущих колес Бенгта Илона,

практически исключающих силы трения при передвижении. Разработанный программный код позволяет удостовериться, что никаких специальных конструкций при использовании роликонесущих колес программировать не приходится.

Список литературы

1. Ловин Д. Создаем робота-андроида своими руками. – М.: ДМК Пресс, 2007.
2. Шахинпур М. Курс робототехники. – М.: Мир, 1990.

УДК 681.883.7

ПРОГРАММА КОНСТРУКТОРСКОГО ПРОЕКТИРОВАНИЯ ТОНКОПЛЕНОЧНЫХ ЭЛЕКТРОЛЮМИНЕСЦЕНТНЫХ СТРУКТУР

О.В. Максимова[33], Д.А. Евсевичев[34].

В настоящее время наиболее перспективными средствами отображения информации являются индикаторные устройства на основе электролюминесцентных излучателей. Разработка тонкопленочных электролюминесцентных (ТПЭЛ) структур является комплексной задачей и требует значительного труда разработчика. Создаваемые программные модули обеспечат автоматизацию расчетов основных функциональных характеристик и параметров, необходимых для проектирования ТПЭЛ индикаторов.

Введение

К числу наиболее перспективных дисплейных устройств относятся индикаторы на основе электролюминесцентных излучателей, которые занимают особое место среди активных индикаторных устройств благодаря своей плоской твердотельной конструкции, высокому быстродействию, широкому диапазону рабочих температур.

Наиболее типичная конструкция электролюминесцентных конденсаторов содержит пять слоев, нанесенных на диэлектрическую подложку [1]: проводящий нижний, диэлектрический нижний, люминесцентный, верхний диэлектрический, верхний проводящий.

Кроме перечисленных пленок в состав электролюминесцентного конденсатора могут входить дополнительные слои, например, слой светопоглощающего контрастного материала.

Кроме указанной конструкции исследованы другие типы источников излучения, имеющие меньшее число диэлектрических слоев (к достоинствам индикаторных устройств таких типов относятся, как правило, меньшие значения управляющего напряжения, а к недостаткам –

[33] *432027, Ульяновск, ул. Северный Венец, 32, УлГТУ, e-mail: first32007@yandex.ru*

[34] *432027, Ульяновск, ул. Можайского, 8/8, УИГА, e-mail: denistk_87@mail.ru*

более низкая яркость и светоотдача, меньшая надежность). В качестве подложки и одновременно первого диэлектрика может быть использована керамическая пластина, тогда верхний электрод должен быть прозрачным.

1. Разработка методики проектирования конструкций тонкопленочных электролюминесцентных индикаторных устройств

Процесс проектирования ТПЭЛ индикаторов заключается в принятии проектных решений, обеспечивающих выполнений разрабатываемым индикатором предъявляемых к нему требований.

Конструкторское проектирование индикатора включает следующие этапы: разработка и анализ технического задания (ТЗ) на проектируемый индикатор, структурный синтез и анализ конструкции индикатора, параметрический синтез и анализ спроектированной конструкции, разработка комплекта конструкторской документации.

Синтез ТПЭЛ индикатора включает в себя параметрический и структурный. Проектирование индикатора начинается с анализа ТЗ и структурного синтеза устройства, при котором определяется принципиальное решение. Таким решением является выбор одной или нескольких конструкций индикатора с различными количеством и свойствами материалов слоев. Задача синтеза структуры является сложно формализуемой задачей, что проявляется в описании процесса синтеза структур, включающим большое числа факторов, влияющих на разновидности, свойства и параметры синтезируемых ТПЭЛ структур. Для описания множества структурных решений наиболее удобным является использование морфологического И–ИЛИ деревьев (рисунок 1). Такой подход позволяет выбрать структуру наиболее соответствующую требованиям как ТЗ, так и предпочтениям оператора – проектировщика.

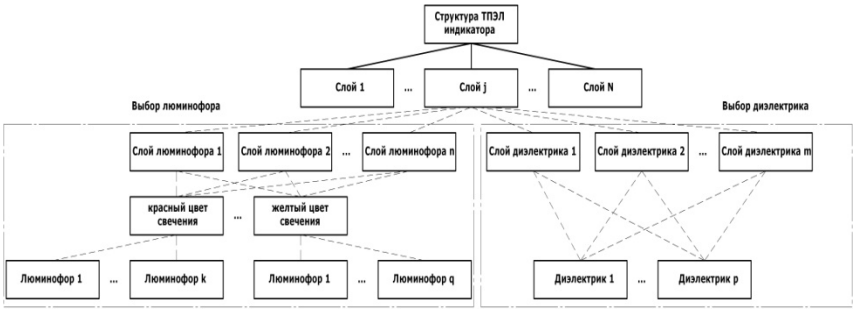


Рис. 1. Морфологическое И-ИЛИ дерево структуры ТПЭЛ индикаторного элемента

В процессе выполнения структурного синтеза ТПЭЛ индикатора формируется множество альтернатив конструкций элемента. Итоговая конструкция ТПЭЛ индикатора выбираются из множества заданных в параметрическом виде, т. е. без указания числовых значений параметров элементов. Поэтому, следующим этапом проектирования является задание или расчет значений этих параметров, т. е. выполнение параметрического синтеза.

Задача параметрического синтеза индикатора заключается в определении наилучших значений конструктивных параметров для выбранной структуры объекта с учетом всех требований ТЗ (рабочий диапазон электрических и светотехнических параметров) на проектируемый индикатор.

Для формирования и решения задачи параметрического синтеза необходимо формирование целевой функции, отражающей качество функционирования проектируемой системы или объекта и нахождение его экстремума. Для рассматриваемых ТПЭЛ ИУ такой функцией может выступать зависимость максимальной яркости индикатора от толщины люминесцентного слоя и зависимость порогового напряжения от толщин люминесцентного и диэлектрических слоев. Эти функции характеризуют такие параметры, как средняя мощность, средняя яркость, продолжительность свечения источника излучения. Формально задачу параметрического синтеза ИУ можно представить в следующем виде:

$$\begin{cases} F_1(X) = U_n(d_л, d_{д1}, ..., d_{дm}) \to min \\ F_2(X) = B_0(d_л) \to max \end{cases} \quad (1)$$

при ограничениях

$$\begin{cases} U_A < U_{max}(d_л, d_{д1}, ..., d_{дm}) \\ U_A \geq U_n(d_л, d_{д1}, ..., d_{дm}) \\ d_л, d_{д1}, ..., d_{дi} > 0 \end{cases}, \quad (2)$$

где $U_n(d_л, d_{д1}, ..., d_{дm})$ – функция зависимости порогового напряжения от толщин слоев;

$B_0(d_л)$ – функция зависимости максимальной яркости от толщины люминесцентного слоя;

$U_{max}(d_л, d_{д1}, ..., d_{дm})$ – функция зависимости максимального напряжения от толщин слоев;

U_A – амплитуда рабочего напряжения;

$d_л$ – толщина люминесцентного слоя проектируемого индикатора;

$d_{\partial i}$ – толщина i-го диэлектрического слоя проектируемого индикатора ($i = 1…m$).

Описанная многокритериальная задача решается методом идеальной точки с использованием штрафных функций. Система (1) в этом случае принимает вид:

$$\begin{cases} -\dfrac{\eta_e \cdot N}{F \cdot \tau} \cdot d_л + r \cdot \left(\dfrac{1}{U_A - E_{пл}(d_л + \dfrac{\varepsilon_л}{\varepsilon_{\partial 1}} \cdot d_{\partial 1} + ... + \dfrac{\varepsilon_л}{\varepsilon_{\partial m}} \cdot d_{\partial m})} + \dfrac{1}{E_{пл} \cdot d_л + E_{п\partial 1} \cdot d_{\partial 1} + ... + E_{п\partial m} \cdot d_{\partial m} - U_A} \right) \to min \\ E_{пл}(d_л + \dfrac{\varepsilon_л}{\varepsilon_{\partial 1}} \cdot d_{\partial 1} + ... + \dfrac{\varepsilon_л}{\varepsilon_{\partial m}} \cdot d_{\partial m}) + \\ + r \cdot \left(\dfrac{1}{E_{пл}(d_л + \dfrac{\varepsilon_л}{\varepsilon_{\partial 1}} \cdot d_{\partial 1} + ... + \dfrac{\varepsilon_л}{\varepsilon_{\partial m}} \cdot d_{\partial m})} + \dfrac{1}{E_{пл} \cdot d_л + E_{п\partial 1} \cdot d_{\partial 1} + ... + E_{п\partial m} \cdot d_{\partial m} - U_A} \right) \to min \end{cases}, \quad (3)$$

где η_e – оптический выход;

N – концентрация центров свечения;

τ – постоянная времени для излучательных переходов;

$E_{пл}$ – пороговая напряженность электрического поля в люминофоре;

$E_{п\partial i}$ – пробивная напряженность электрического поля в i-ом диэлектрике;

r – параметр штрафа (изменяется с каждой итерацией).

Решение задачи синтеза (3) позволяет найти оптимальные значения толщин люминофора и диэлектриков, удовлетворяющих ограничениям (2). Найденные конструктивные параметры позволяют сформировать окончательную структуры ТПЭЛ ИУ.

Представленная методика расчета конструкции индикатора послужила основой для формирования алгоритма и программы автоматизированного проектирования ТПЭЛ индикаторов IDECSoft.

Следующим этапом проектирования является анализ разработанной конструкции. Алгоритм решения задачи квалиметрического анализа конструкций ТПЭЛ индикаторов позволяет рассчитывать основные электрические и светотехнические характеристики при определенных на этапах синтеза конструктивных параметрах, сравнивать их с заданными

техническими условиями и в случае необходимости корректировать задачи селективного структурного синтеза, редукционного параметрического синтеза или изначальные технические условия на проектирование.

2. Система автоматизации проектирования конструкций тонкопленочных электролюминесцентных индикаторных устройств IDECSoft

Результатом проведённой исследовательской работы является завершённая программа системы автоматизированного проектирования для ТПЭЛ индикатора.

На основе разработанных алгоритмов в среде разработки Delphi 7 с использованием императивного, структурированного, объектно-ориентированного языка программирования Delphi разработана система IDECSoft, позволяющая выполнять автоматизированное проектирование ТПЭЛ индикатора по описанным методикам синтеза и анализа. Разработанная автоматизированная система представляет собой прикладное программное обеспечение, запускаемое из операционной системы Windows. Программа работает непосредственно с пользователем, устанавливая необходимые связи для проведения расчета. Результаты работы оформлены в виде программы «Система автоматизированного проектирования тонкопленочных электролюминесцентных индикаторов», зарегистрированной Федеральной службой по интеллектуальной собственности (свидетельство о регистрации программы для ЭВМ №2013618689).

С использованием программы проведена оценка точности расчетов функциональных характеристик от конструктивных параметров относительно экспериментальных данных. Проведенные исследования позволяют говорить о достаточно высокой точность выбранных математических моделей. Относительная погрешность составила не более 5%, что объясняется погрешностью производственного и измерительного оборудования.Окно программы разделено на четыре области:

- Область управления проектом, представляющая собой набор функциональных кнопок, задействованных на всех этапах проектирования ТПЭЛ-индикатора.
- Область «Исходные данные» отображает входные параметры, необходимые для расчета конструкции ТПЭЛ-структуры.
- Область расчета отображает выходные данные по спроектированному устройству.

- В таблице представлены множества альтернатив сформированных конструкций

Разработанный программный продукт позволяет спроектировать индикаторное устройство по выбранным оператором параметрам расчета. Результаты вычислений представляются на главной странице программы и в отдельном окне в виде отчета, который можно сохранить в формате pdf или сразу распечатать.

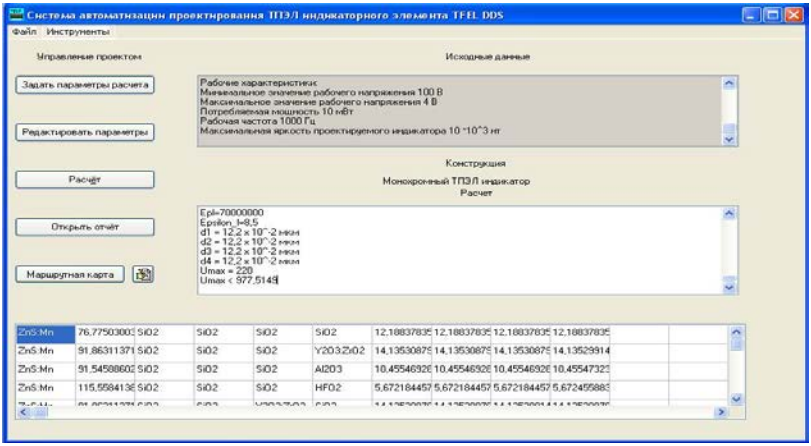


Рис.2. Интерфейс программы IDECSoft после выполнения вычислений

Программа включает несколько дополнительных опций, позволяющих осуществлять исследование конструкций ТПЭЛ-индикаторов (модуль TFEL ARSM).

Заключение

Представленные методики и разработанный на их основе программный продукт обеспечивают автоматизацию конструкторского проектирования ТПЭЛ ИУ. Полученные в результате выполнения данной работы выводы и результаты могут быть использованы в лабораториях и конструкторских бюро, занимающихся проектированием и исследованием электролюминесцентных источников излучения.

Список литературы

1. Евсевичев Д.А., Самохвалов М.К. Автоматизация исследования материалов люминофоров и диэлектриков в тонкопленочных электролюминесцентных индикаторах // Автоматизация в промышленности. – М.: ИнфоАвтоматизация, 2011 – № 9. – С. 55–57.

УДК 621.372

ОБЗОР ДИАГРАММНЫХ ГРАФИЧЕСКИХ ЯЗЫКОВ САПР

В.В. Мерясева[35]

В данной статье обозреваются основные методики диаграммных графических языков САПР, используемые при концептуальном проектировании систем.

Введение

В наши дни процесс создания сложных систем невозможно представить без этапа проектирования. В ходе проектирования создается проектная документация, включающая текстовые описания, диаграммы, модели системы. Диаграммные графические языки являются удобнейшим инструментом при создании такой документации.

Диаграммные графические языки САПР

Диаграммный графический язык – это язык, предложения которого состоят из графических примитивов, таких как вершины и дуги. Предложениями такого языка служат ориентированные графы.[1]

Существует множество видов диаграммных графических языков, ниже приведены наиболее используемые из них.

DFD (Data Flow Diagrams) – диаграммы потоков данных. DFD-методология является одним из основных инструментов проектирования и исследования информационных систем изучаемой предметной области.

Модель DFD представляет собой иерархию функциональных процессов, связанных потоками данных, каждый ее процесс, при необходимости, может подвергаться декомпозиции. Это позволяет продемонстрировать, как в каждом процессе преобразуются его входные данные, и наглядно показать отношения между процессами. [2]

Элементы внешней среды называют внешней сущностью. Они порождают потоки данных, которые переносят информацию к процессам, где происходит преобразование информации и порождение новых потоков данных. Они могут поступать на вход других процессов,

[35] *432027, Ульяновск, ул. Северный Венец, 32, УлГТУ, e-mail: valeriameryaseva@gmail.com*

храниться в накопителях данных или извлекаться из них, а также передаваться внешним сущностям.

STD (State Transition Diagram) – диаграмма переходов состояний. STD характеризует поведение проектируемой системы во времени, показывает ее реакцию на события, описывает отношения между входными и выходными управляющими потоками.

STD представляет собой графическую форму конечного автомата, где состояния системы представлены узлами, переходы из состояния в состояние – дугами, а условия переходов – атрибутами дуг. Управляющие процессы STD могут быть подвергнуты декомпозиции. [3]

UML (Unified Modeling Language) – унифицированный язык моделирования. UML является графическим языком, служащим для описания и визуализации различных систем. Чаще всего используется в области разработки программного обеспечения.

В UML используется множество видов диаграмм: классов, объектов, прецедентов, развертывания, компонентов, состояний и т.д. Это позволяет детально описать исследуемую систему со всех сторон.

IDEF – методология для концептуального проектирования систем. Методики IDEF позволяют разрабатывать функциональные, информационные, поведенческие модели системы.

К методологии IDEF относится множество стандартов, основные из них:

IDEF0 – методология функционального моделирования, позволяющая представить систему в виде набора функций.

IDEF1 – методология информационного моделирования, позволяющая представить структуру информационных потоков внутри системы.

IDEF1X – методология построения реляционных структур. Используется для моделирования реляционных баз данных, имеющих отношение к рассматриваемой системе.

IDEF3 – методология поведенческого моделирования, описывает последовательность действия каждого процесса.

Заключение

Рассмотрев виды и методики диаграммных графических языков, можно сделать вывод, что они являются мощным инструментом концептуального проектирования, позволяют рассмотреть систему с разных точек зрения.

Список литературы

1. Шаров О.Г. Исследование и разработка методов и средств реализации диаграммных графических языков САПР: дис. на соиск. уч. ст. канд. техн.

наук, УлГТУ, Ульяновск, 2007. http://venec.ulstu.ru/lib/disk/2012/Warov.pdf

2. DFD методология. Нотация, принципы моделирования. http://www.nazametku.com
3. Диаграммы переходов состояний (STD). http://studopedia.org

УДК 004:378.14

ПРОЕКТИРОВАНИЕ МОДУЛЯ СЕТЕВОЙ КОММУНИКАЦИИ ИНФОРМАЦИОННОЙ СИСТЕМЫ[36]

Р.С. Молотов[37]

В рамках данной работы рассматривается проектное решение модуля сетевой коммуникации для информационной системы для хранения, обработки и визуализации данных при проведении шахматных турниров. Приводится пример реализации данного решения для приложения WinForms, работающего с аппаратными единицами по TCP протоколу.

Введение

Значительная часть коммерческих информационных систем (ИС), разрабатываемых в данный момент, является распределенными системами. Для автоматизации бизнес-процессов здесь наиболее популярна технология «клиент-сервер». Далее будет рассмотрено проектное решение модуля сетевой коммуникации и его реализация на основе TCP-протокола для модульной ИС, обеспечивающей хранение, обработку и визуализацию данных при проведении шахматных турниров.

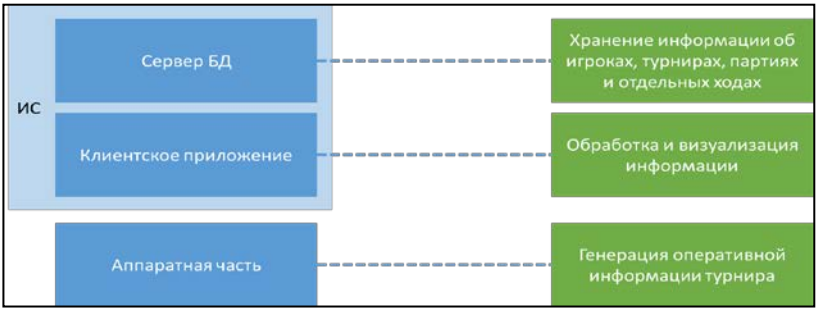


Рис. 1. Архитектура программно-аппаратного комплекса

[36] *Исследование выполнено при финансовой поддержке ФСРМФП в НТС, договор № 9521ГУ/2015 от 01.02.2016.*

[37] *432027, Ульяновск, ул. Северный Венец, 32, УлГТУ, e-mail: r.molotov@yahoo.com*

Как видно из рис. 1, если рассматривать весь программно-аппаратный комплекс в целом, то (относительно природы возникновения, подходов к обработке и назначению информационных потоков) можно выделить два глобальных уровня:

1. Программный. ИС, построенная по технологии «клиент-сервер» (доступ приложения к данным на сервере обеспечивает Entity Framework [1, с. 795]).
2. Аппаратный. Аппаратные единицы (АЕ) – электронные шахматные доски – обеспечивающие генерацию и передачу ИС оперативной информации о ходе турнира.

Связь приложения с БД обеспечивает фреймворк доступа к данным, а вот для организации подключения АЕ необходимо добавить в приложение дополнительную логику и реализовать более сложную, многослойную архитектуру [ИС].

1. Постановка требований к модулю и обзор TCP протокола

ИС должна обеспечивать хранение, обработку и визуализацию информации. Чтобы декомпозировать это требование и понимать назначение модуля сетевой коммуникации, необходимо понимать, что информационные потоки в ИС могут быть двух типов:

- данные (записи об игроках, турнирах и партиях), получаемые из БД;
- оперативная информация о ходе турнира, получаемая в режиме реального времени от АЕ.

Именно обработку, визуализацию (вызов методов перерисовки компонентов GUI) и занесение (обновление) в БД оперативной информации и должен обеспечивать проектируемый модуль. Выделим основные функциональные требования:

1. поддержка идентификации подключенных аппаратных единиц и их централизованная инициализация (отображение на экране АЕ фамилий игроков и установку времени согласно регламенту турнира);
2. прием и обработка ИС сообщений от АЕ о совершаемых игроками ходах в полной алгебраической (SAN) шахматной нотации с идентификатором (флагом) стороны, совершающей ход;
3. обеспечение приема и обработки ИС сообщений от АЕ об оставшемся у игроков времени – визуализацию хода часов каждой АЕ в клиентском приложении ИС;
4. в случае завершения партии – прием и обработка сообщения с результатом партии;
5. в случае аварийного завершения партии на отдельной АЕ – уведомление пользователя (судьи, оператора) приложения, предоставление ему возможности завершить партию вручную.

В то же время, программная реализация модуля должна удовлетворять основным принципам проекта: быть простой и понятной, обеспечивать прозрачность программных интерфейсов ИС. Любая сетевая коммуникация строится на основе определенного сетевого протокола. Для соблюдения принципа прозрачности и открытости системы имеет смысл рассматривать модель OSI и её сетевой протокол IP. На транспортном уровне возможно применение UDP или TCP.

Если говорить об использовании UDP, то организовать работу клиентского приложения с АЕ очень просто: необходимо лишь обеспечить прослушку порта, на который будут поступать сообщения определенного формата. Этот протокол транспортного уровня хорошо подходит для получения информации с часов игроков: информация идет плотным непрерывным потоком, даже если одно из сообщений («одна секунда») потеряется, то через секунду эта «информационная дыра» исчезнет. Куда более критичны потери пакетов при передаче ходов игроков. Так как ходы, для удобства их обработки передаются в SAN, то пропуск одного хода ведет к ошибкам визуализации и сложности логгирования всей партии. Именно поэтому более надежным и достаточным вариантом будет применение решения на основе TCP протокола, обеспечивающего гарантированную доставку пакетов и обработку отключений клиентов, что необходимо в случае аварийного завершения какой-либо партии.

В фреймворке .NET существует пространство имен System.Net.Sockets, включающее классы TcpListener для организации TCP сервера, принимающего сообщения и TcpClient для представления клиентских подключений. На основе этих двух классов и будет далее построена реализация программного модуля.

2. Архитектура программного модуля сетевой коммуникации

Как было сказано ранее, модуль построен по архитектуре клиент-сервер: есть список экземпляров класса TcpListener, представляющих клиентские подключения (АЕ) и объект сервера, этими подключениями управляющий. Базовая реализация клиента представлена в листинге 1а. Здесь реализован функционал подключения клиента, приема клиентом сообщений и, в случае, окончания приема сообщений – отключение клиента. На стороне сервера (листинг 1б), в общем случае, должны быть реализованы подключение клиентов, отправка и прием сообщений, отключение клиентов. Также, в обоих случаях необходимо добавить конструкцию try-catch для обработки непредвиденного отключения клиентов и ошибок сетевого взаимодействия. Такая базовая реализация не

подходит для применения в ИС, однако, на базе класса TcpClient удобно будет построить контролер, сущность, отображающее АЕ на объект.

```
TcpClient client = new TcpClient();
client.Connect(server, port);
byte[] data = new byte[256];
StringBuilder response = new StringBuilder();
NetworkStream stream = client.GetStream();
do {
      int bytes = stream.Read(data, 0, data.Length);
      response.Append(Encoding.UTF8
            .GetString(data, 0, bytes));
}
while (stream.DataAvailable);
Console.WriteLine(response.ToString());
stream.Close();
client.Close();
```

Листинг 1а. Базовая реализация TCP-клиента

```
IPAddress localAddr = IPAddress.Parse("127.0.0.1");
server = new TcpListener(localAddr, port);
server.Start();
while (true) {
      TcpClient client = server.AcceptTcpClient();
      NetworkStream stream = client.GetStream();
      string response = "message";
      byte[] data = Encoding.UTF8.GetBytes(response);
      stream.Write(data, 0, data.Length);
      client.Close();
}
```

Листинг 1б. Базовая реализация TCP-сервера

Для расширения класса TCP-клиента до класса, отображающего в приложении подключенную АЕ конечный контроллер [доски] должен быть связан с элементом GUI (в WinForms - контролом), обеспечивающим визуализацию конкретной доски в приложении и иметь доступ к записи в БД, соответствующей той шахматной партии, которая в данный момент проходит на сопоставленной АЕ, т. е. обрабатывается данным контроллером. Данная архитектура проектного решения соответствует паттерну MVC, разделяя модель данных и бизнес-логику по работе с ними, средство визуализации информации и контроллер, объединяющий все вместе и осуществляющий управление целой сущностью АЕ. Класс

контроллера ClientObject дополняется двумя полями, ссылками GameObject и BoardObject:

```
public class ClientObject {
      protected internal string Id { get; private
set; }
      protected internal NetworkStream Stream { get;
private set; }
      TcpClient client;
      ServerObject server;
      public string boardIP;
      public BoardControl.BoardControl BoardObject;
      public Game GameObject;
}
```

Листинг 2. Класс-контроллер

Кастомный контрол, реализующий визуализацию изменений ситуации на шахматной доске содержит (листинг 3):

- метод для установки часов SetBoard;
- методы для визуализации хода часов и совершения ходов игроками – DoMove, WhiteClockTick и BlackClockTick;
- методы отображения результата (DisplayResult) и элемента GUI для завершения партии вручную, в случае отключения АЕ (DisplayPanel);
- обработчик события отключения АЕ SetResClick.

```
public class BoardControl : UserControl {
      public Label BlackPlayer;
      public Label BlackTime;
      public Label WhitePlayer;
      public Label WhiteTime;
      public event EventHandler SetResClick;
      public void BlackClockTick(string time) {...}
      public void DisplayPanel() {...}
      public void DisplayResult(string result) {...}
      public void DoMove(string move, string player)
      {...}
      public void SetBoard(string wp, string bp,
DateTime wt, DateTime bt) {...}
      public void WhiteClockTick(string time) {...}
}
```

Листинг 3. Кастомный контрол, визуализирующий подключенную АЕ

Используя экземпляры описанных классов, запрограммировать запуск раунда турнира будет довольно просто (листинг 4): для каждой парии необходимо создать отдельный контроллер, сопоставить с ним соответствующую запись [заготовки] партии в БД (объектное отображение в коде – Game GameObject), и подключить к контроллеру «свободный» BoardObject, подписав контроллер на его событие BoardObjectGetResult:

```
TCPServer.ConnectedBoards = TCPServer.ConnectedBoards
      .OrderBy(b => b.boardIP).ToList();
foreach (Game game in currentGamesList) {
      ClientObject controller =
TCPServer.ConnectedBoards.First(b => b.GameObject ==
null);
      controller.GameObject = game;
      if (controller.BoardObject == null) {
            controller.BoardObject = BoardsGUI
                  .First(b => !TCPServer
                  .ConnectedBoards.Select(c =>
                  c.BoardObject).Contains(b));
            controller.BoardObject.SetResClick +=
                  new EventHandler(controller
                  .BoardObjectGetResult);
      }
      controller.BoardObject.Visible = true;
      string setString = String.Format("st{0}{1}{2}",
            controller.GameObject.MyWhitePlayer,
            controller.GameObject.MyBlackPlayer,
            currentGameTime.TimeOfDay);
      TCPServer.SendMessage(setString,
            controller.Id);
      controller.BoardObject
            .SetBoard(game.MyWhitePlayer,
                  game.MyBlackPlayer,
                  currentGameTime,
                  currentGameTime);
}
```

Листинг 4. Логика запуска раунда и сопоставление парии, АЕ и контрола

Подписка контролера на событие контрола необходима для того, чтобы, в случае завершения партии вручную, логику обработки описывать

не в классе контрола, а в отдельном методе (листинг 5) контроллера, который будет срабатывать при возникновении события. Также, в случае отключения АЕ и удалении контрола из списка подключенных клиентов необходимо отписать контроллер от события SetResClick контрола:

```
public void BoardObjectGetResult(object sender,
EventArgs e) {
      this.GameObject.Result = (string)sender;
      Database.db.SaveChanges();
      this.GameObject = null;
      server.RemoveConnection(this.Id);
      this.BoardObject.SetResClick -=
            BoardObjectGetResult;
      CurrentSession.CheckRound();
}
```

Листинг 5. Кастомный контрол, визуализирующий подключенную АЕ

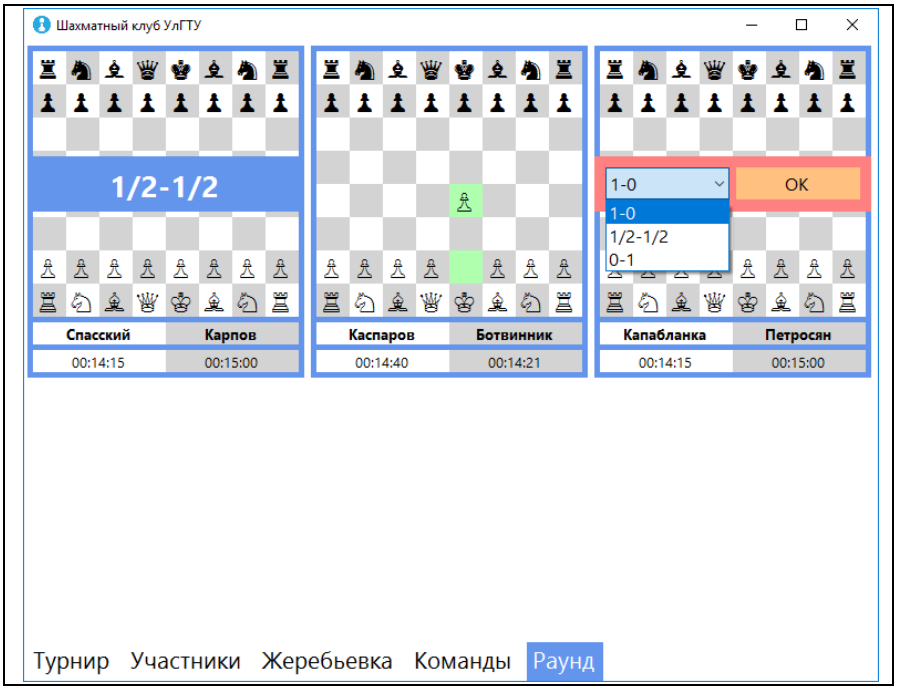


Рис. 2. Визуализация хода турнира в клиентском приложении

В результате был реализован полноценный модуль сетевой коммуникации, отвечающий всем функциональным требования (см. раздел 1) и успешно интегрированный в ИС. Пример работы (отображение результата партии, визуализация ходов игроков, завершение партии вручную) модуля в составе ИС показан на рис. 2.

Заключение

В начале данной статьи было сказано, что программно-аппаратный комплекс разделен на два глобальных слоя. Однако, если подробно рассматривать механизмы подключения клиентского приложения к БД и взаимодействие клиента с подключенными АЕ, то можно заметить, что и они работают по технологии, разделяющей два уровня – клиента и сервера. Таким образом, говоря об архитектуре как ИС, так и о программно-аппаратном комплексе как о распределенной системе, можно отметить тот факт, что они имеют более сложную, многослойную архитектуру.

Для проектирования и реализации таких систем важно разделять логику и представление, что и было сделано при проектировании программного модуля. Необходимо отметить, что каждый [современный] инструмент или технология будут «навязывать» разработчику те или иные шаблоны проектирования, как например Entity Framework предполагает неукоснительное следование паттерну Units Of Work и провоцирует на применение Active Record. Слепое следование за технологиями и паттернами проектирования может привести к ненужному усложнению проекта. Именно поэтому, при разработке модуля сетевой коммуникации применялся не классический MVC, а его адаптация для использования с TCP протоколом: совмещение бизнес-логики контроллера с классом TcpClient для обработки информации, поступающих от АЕ.

Список литературы

1. Троелсен, Э. Язык программирования C# 5.0 и платформа .NET 4.5, 6-е изд.: Пер. с англ. - М.: ООО «И.Д. Вильямс», 2013.
2. Раттц, Д. LINQ: язык интегрированных запросов в C# 2008 для профессионалов: Пер. с англ. - М.: ООО «И.Д. Вильямс», 2008.
3. Р.С. Молотов. Проектирование информационной системы для автоматизации проведения турниров в шахматном клубе УлГТУ / Электронное обучение в непрерывном образовании 2016. III Международная научно-практическая конференция: сборник научных трудов. – Ульяновск: УлГТУ, 2016. – 1320 с.
4. Р.С. Молотов. Программно-аппаратная система для хранения, обработки и визуализации данных при проведении шахматных турниров / Материалы V ульяновского молодежного инновационного форума, 2016. URL: http://ify.ulstu.ru/sites/default/files/Molotov%202_0.pdf

УДК 004.82

ПРЕДСТАВЛЕНИЕ ПРЕДМЕТНОЙ ОНТОЛОГИИ С ИСПОЛЬЗОВАНИЕМ ИНСТРУМЕНТОВ ПЛАТФОРМЫ 1С:ПРЕДПРИЯТИЕ[38]

В.С. Мошкин[39]

В данной работе рассмотрен подход реализации онтологического формата представления знаний посредством использования встроенных механизмов и типов метаданных технологической платформы 1С:Предприятие в рамках решения задачи интеллектуализации поддержки принятия решений в области управленческого учета, планирования и прогнозирования финансового состояния организации.

Введение

При решении сложных задач, в том числе и в области проектирования и автоматизации бизнес-процессов конкретных предприятий, процесс принятия решений нуждается в создании эффективных методов и инструментальных средств повышения их качества. Для этого необходимы соответствующие модели, ориентированные на описание объектных структур рассматриваемой предметной области (ПрО), позволяющие идентифицировать, анализировать и манипулировать всем многообразием объектов и отношений, имеющихся в предметной области.

Решением поставленных задач чаще всего является разработка экспертных систем, в качестве базы знаний которых выступают предметные онтологии в формате OWL (Web Ontology Language)[1]. Несмотря на все преимущества, которыми обладает данный формат представления знаний, не все технологические платформы способны использовать полностью его функционал. В связи с этим актуальным

[38] *Исследование проводится при поддержке Фонда содействия развитию малых форм предприятий в научно-технической сфере по программе «УМНИК» по теме «Разработка системы оценки и прогнозирования финансового состояния коммерческой организации на основании интеллектуального анализа временных рядов бухгалтерской отчетности» в рамках договора № 9289ГУ/2015 от 24.12.2015 г.*

[39] *432027, Ульяновск, ул. Северный Венец, 32, УлГТУ, e-mail: PostForVadim@yandex.ru*

становится вопрос о разработке моделей и механизмов экспорта онтологий в форматы метаданных прикладного решения на используемой платформе с сохранением базового функционала и преимуществ онтологического формата представления знаний.

1. Формальная модель предметной OWL-онтологии

В широком смысле, онтологии – это модели, представляющие собой форму представления знаний о предметных областях в виде семантических информационно-логических сетей взаимосвязанных объектов, где в качестве главных элементов выступают понятия предметной области с их свойствами и отношения между объектами. Онтологии выполняют интегрирующую функцию, обеспечивая общий семантический базис в процессах принятия решений и единую платформу для объединения разнообразных информационных систем [2, 4].

В настоящее время построением экспертных систем с использованием предметных онтологий и онтологических структур занимается множество отечественных и зарубежных исследователей, среди которых: T. Gruber, N. Guarino, M. King, M. Uschold, G. Rzevski, F. Bobillo и U.Straccia, Гаврилова Т.А., Хорошевский В.Ф., Соснин П.И., Загорулько Ю.А., Грибова В.В., Вагин В.Н., Боргест Н.М., Голенков В.В и т.д.

Предметная область проектирования сложных бизнес-процессов накладывает определенные требования к структуре прикладной онтологии. Онтологию предметной области запишем в виде кортежа:

$$O = (T, R^D, F),$$

где T – термины прикладной области, которую описывает онтология. Множество терминов представлено в виде:

$$T = \{C, In_C\},$$

где C – множество классов онтологии, In_C – множество объектов классов онтологии.

R^D - множество отношений (свойств) объектов онтологии:

$$R^D = \{R_{In}, R_F, R_{InF}, R_{Tran}, R_{Sym}, R_{Asym}, R_{Ref}, R_{Irref}\},$$

где R_{In} – множество обратных отношений, R_F – множество функциональных отношений, R_{InF} – множество обратных функциональных отношений, R_{Tran} – множество транзитивных отношений, R_{Sym} – множество симметричных отношений, R_{Asym} – множество асимметричных свойств, R_{Ref} – множество рефлексивных отношений, R_{Irref} – множество иррефлексивных отношений [5].

Формат OWL предполагает возможность представления свойств объектов онтологии в трех видах:

- свойства аннотаций (annotation properties);
- свойства типов данных (datatype properties);
- свойства объектов (object properties) [7].

F – множество функций интерпретации (аксиоматизации), заданных на терминах и/или отношениях онтологии.

Представление знаний в таком формате позволяет проводить анализ особенностей предметной области и делать логический вывод новых знаний на основе имеющихся.

Данный функционал полезен при реализации систем, работающих с большими объемами данных, а также затрагивающих процессы учета, управления, планирования и прогнозирования ресурсов предприятия. В настоящее время одной из ведущих в России технологических платформ, позволяющих разрабатывать программные системы (конфигурации), предназначенные для решения поставленных задач, является 1С:Предприятие 8.

2. Трансляция онтологии в форматы метаданных платформы 1С:Предприятие

Конфигурация 1С — это набор форм и алгоритмов, написанных на платформе 1С, для ведения учета хозяйственной деятельности предприятия. Под объектами конфигурации понимаются средства 1С:Предприятия, предназначенные для отражения реальных объектов и явлений предметной области, например, справочник "Контрагенты", документ "Счет" и т.д.

Кроме того, разработчик может создавать объекты метаданных, не имеющие явного физического воплощения в предметной области, но необходимые для решения задачи, например, регистры сведений, обработки и т.д (рис.1) [3].

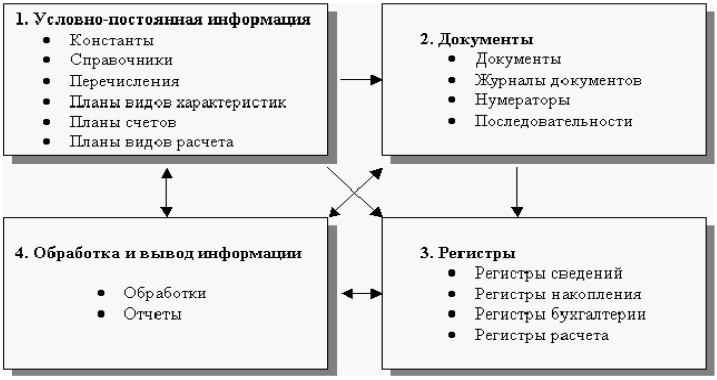


Рис.1. Набор объектов конфигурации на платформе 1С:Предприятие

Использование онтологической формы представления знаний в рамках конфигурации на платформе 1С:Предприятие позволит решить несколько проблем:

1. Связать используемые объекты конфигурации с целью последующего извлечения зависимостей между ними с учетом семантики конкретной предметной области.

2. Обеспечить возможность логического вывода новых знаний в задачах поддержки принятия решений на экспертном уровне в процессе анализа данных стандартной бухгалтерской, налоговой и других форм отчетности.

Наиболее сложной задачей в процессе трансляции онтологической формы представления знаний в форматы метаданных платформы 1С:Предприятие, является определение характера взаимодействия объектов метаданных с сохранением их логической целостности. Хранение свойств объектов осуществляется посредством использования механизма описания характеристик.

Механизм описания характеристик - это один из прикладных механизмов платформы. Он позволяет организовать хранение свойств объектов (справочников, документов и т.д.), которые еще не известны на момент разработки прикладного решения.

Для реализации механизма описания характеристик используются следующие объекты метаданных платформы 1С:Предприятие:

- Справочники. Справочники используются для хранения объектов ПрО в прикладном решении.
- ПланыВидовХарактеристик. Планы видов характеристик используется для создания и хранения перечня характеристик, которые могут использоваться в прикладном решении.

- Регистры сведений. Регистр сведений можно использовать для хранения значений конкретных характеристик, указанных для некоторого объекта прикладного решения [6].

Конкретные объекты наборов метаданных, а также особенности характера их взаимодействия в процессе трансляции OWL-онтологии в форматы метаданных платформы 1С:Предприятие представлены в таблице 1.

Таблица 1. Объекты используемых метаданных

Тип объекта метаданных	**Объект метаданных**	**Назначение объекта метаданных**	**Особенности объекта метаданных**
Справочник	ОбъектОнтологии	Хранение объектов онтологии, а также задание отношений иерархии	Справочник имеет свойство иерархичности
Справочник	ЗначенияСвойств Объектов	Хранение объектов, определяемых в процессе расширения онтологии	
План видов характеристик	ВидыХарактеристик	Определение набора отношений для каждого объекта онтологии (справочник «ОбъектОнтологии»)	Тип значения характеристик: Составной тип. Дополнительные значения характеристик: ЗначенияСвойствОбъектов
Регистр сведений	Характеристики	Хранение значений характеристик объектов экспортируемой онтологии	**Измерения:** ОбъектОнтологии, Характеристика **Ресурсы:** Значение.

Алгоритм трансляции онтологии в форматы метаданных платформы 1С:Предприятие включает в себя следующие этапы:

1. Создание объектов справочника «ОбъектОнтологии» в соответствии с объектами предметной онтологии с учетом указанной в ней иерархии.
2. Определение набора отношений для каждого объекта справочника «ОбъектОнтологии» путем заполнения плана видов характеристик «ВидыХарактеристик» (рис.2). В данном объекте конфигурации могут быть определены как свойства типов данных (datatype properties), так и свойства объектов (object properties).

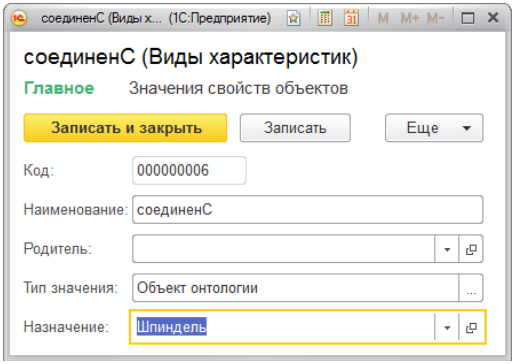


Рис.2. Определение свойства «соединенС» с помощью Плана видов характеристик «Виды характеристик»

3. Определение значений свойств объектов онтологии посредством заполнения полей регистра сведений «Характеристики». В случае, если типом значения объекта Вида характеристики на п.2. был выбран справочник «Значения свойств объектов», то необходимо создать объекты справочника «ЗначенияСвойствОбъектов», которые могут являться значениями характеристик объекта.

Набор аксиом, являющихся составной частью онтологий, в процессе трансляции на встроенный язык 1С, могут быть представлены:

- в виде ограничений на типы данных плана видов характеристик «ВидыХарактеристик»;
- в форме функций и процедур на встроенном языке 1С, программно задающих логические связи между объектами соответствующих элементов конфигурации.

Заключение

Таким образом, трансляция предметных онтологий в объекты конфигураций на платформе 1С с возможностью сохранения основных особенностей и преимуществ онтологической формы представления знаний является решаемой задачей с дальнейшей перспективой использования полученной структуры при решении задач анализа данных, а также поддержки принятия решений на экспертном уровне.

Функционал полученных объектов метаданных конфигурации будет полезен при реализации систем, работающих с большими объемами данных, а также затрагивающих процессы учета, управления, планирования и прогнозирования ресурсов предприятия.

Список литературы

1. OWL Web Ontology Language Overview https://www.w3.org/TR/owl-features
2. Гаврилова Т. А., Хорошевский В. Ф. Базы знаний интеллектуальных систем. Учебник. – СПб.: Питер, 2000. – 384 с.
3. Гончаров Д. И., Хрусталева Е. Ю. Технологии интеграции 1С:Предприятия. Серия «1С:Профессиональная разработка» – М: 1С-Паблишин, 2012.
4. Митрофанова О.А., Константинова Н.С. Онтологии как системы хранения знаний // Всероссийский конкурсный отбор обзорно-аналитических статей по приоритетному направлению «Информационно-телекоммуникационные системы». — 2008. — С. 54
5. Мошкин В.С., Ярушкина Н.Г. Методики построения нечетких онтологий сложных предметных областей // Открытые семантические технологии проектирования интеллектуальных систем (OSTIS-2015): материалы V Междунар. научн.техн. конф. (Минск, 19-21 февраля 2015 г.) / редкол.: В. В. Голенков (отв. ред.) [и др.]. – Минск: БГУИР. – 2015. – С. 401–406.
6. Радченко, М.Г. 1С:Предприятие 8.2. Практическое пособие разработчика. Примеры и типовые приемы.– М.: ООО "1С-Паблишинг", 2009.– 874 с., ил.
7. Ярушкина Н.Г., Мошкин В.С. Применение алгоритма логического вывода на основе FuzzyOWL-онтологии // Радиотехника. – 2015. – № 6. – С. 68–72.

УДК 621.372

ИСПОЛЬЗОВАНИЕ ПРОТОТИПИРОВАНИЯ В ТЕСТИРОВАНИИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Обаид Али хамза Обаид
Аль-Фурат Аль-Аусат

Совершенствование потоков работ, обеспечивающих спецификации концептуализаций, Использование моделей в тестировании программного обеспечения. В статье предлагается использование программы WIQA.

Введение

Использование моделей в тестировании программного обеспечения позволяет легко сократить время в требованиях к программе первого этапа проектирования, тестирования программного обеспечения считается важным аспектом в жизненном цикле программы, из-за его важности в производстве программ приемлемое изображение.

1 Прототипирование

Реализация базовой функциональности для анализа работы системы в целом. На этапе прототипирования малыми усилиями создается работающая система (возможно неэффективно, с ошибками, и не в полной мере) в целях уточнения требований и/или элементы дизайна, может создавать макеты и прототипы экранов, отчетов и процессов. Хотя некоторые прототипы могут оказаться очень значительными, они вообще похожи на съемочной площадке: все выглядит хорошо с фронта, но нет ничего в спину. Когда прототип создается, застройщик производит минимальное количество кода надо уточнить требования, или элементы дизайна в стадии рассмотрения. Никаких усилий, направленных на Обеспечение соответствия стандартам кодирования, обеспечивают надежное управление ошибка или интегрировать с другими таблицами базы данных или модули. В результате, как правило, стоят дороже, для установки прототипа необходимые элементы для создания модуля производства, чем на разработку модуля с нуля, используя окончательный проект документа системы. По этим причинам, опытные образцы не предназначены для коммерческого использования, и, как правило, калеки в одну или другую сторону, чтобы предотвратить их от быть ошибочно использованы в качестве производственных модулей конечными пользователями. Прототипирование, по мнению некоторых разработчиков, является самым важным этапом разработки. После этапа прототипирования

обязательно следуют этапы пересмотра архитектуры системы, разработки, реализации и тестирования конечного продукта.

2 Процесс Прототипирования

Прототипирование может использоваться как для крупных, так и небольших приложений. Как правило, большие системы по-прежнему требуют использования традиционного системного подхода развития, но деталей таких систем часто могут быть прототип. Прототипирование сочетает в себе этапы традиционного цикла разработки систем, и позволяет быстрое развитие и тестирование рабочей модели. Затем модель уточнялась до тех пор, пока это приемлемо для конечного пользователя. Качества которыми должен обладать эффективный прототип:Этап создания прототипа не должен быть затяжным.

3 Тестирование программного обеспечения

Тестирование программного обеспечения - это метод оценки функциональности программного обеспечения. Есть много различных типов тестирования программного обеспечения, но две основные категории динамических испытаний и статических испытаний. Динамическое тестирование- это оценка, которая проводится при выполнении программы; статических испытаний, с другой стороны, это проверка кода программы и сопутствующей документации. Динамические и статические методы часто используются совместно.

4 Цели тестирования

Цель тестов - обеспечить адекватное покрытие метрик, проверка соответствия требованиям и системе качества данных, таких, что достаточных данных для принятия решения о выпуске.

5 Типы тестирования

Типы тестов вводятся как средства четкого определения цели определенного уровня для программы или проекта. Тест на тип ориентирован на определенную цель испытаний, которые могут быть тестирование функции, которые будет выполнять компонента или системы; нефункциональных характеристик качества, таких как надежность или удобство использования; структура или архитектура компонента или системы; или связанным с изменениями, т.е, подтверждающую, что дефекты были исправлены (подтверждение тестирования или повторного тестирования) и ищите непреднамеренных изменений (регрессионное тестирование). В зависимости от его задач, тестирование будет организовано по-другому. Следовательно, существует четыре вида тестирования программного обеспечения:

1. Функциональное тестирование
2. Нефункциональное тестирование
3. Структурное тестирование
4. Изменения, связанные с тестирования

Существует множество видов тестирования, как (Модульное Тестирование ,Интеграционное Тестирование , Функциональное Тестирование , Тестирование Системы , Стресс-Тестирование , Тестирование Производительности , Юзабилити-Тестирование , Приемочные Испытания , Регрессионное Тестирование , Бета-Тестирование).

6 WIQA

Изначально комплекс WIQA разрабатывался для инструментально-технологического сопровождения процессов решения задач в концептуальном проектировании АС. Этот этап разработки особо критичен к дорогостоящим ошибкам и промахам, причём, основной причиной их появления является проблемы с пониманием, которое в любых своих проявлениях, строится с использованием взаимодействий с доступным опытом.

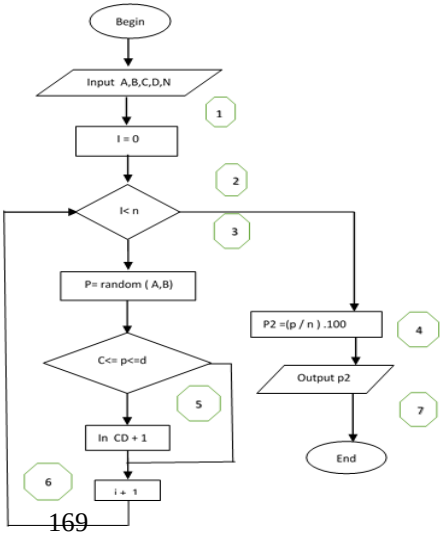
7 Концептуальное решение задач

Инструментарий WIQA предназначен для решения задач на этапе концептуального проектирования АС, когда осуществляется построение концептуальных решений задач, то есть построения решений задач на естественно-профессиональном языке в его алгоритмическом употреблении. Специфика концептуального решения задач в нструментальной

среде WIQA отражена на рис. и рис. . На первом из этих рисунков показано, что по ходу решения основной задачи коллектив проектировщиков шаг за шагом формирует дерево задач проектов, подзадачи которого оперативно распределяются между ними. Каждый из них в рамках возложенной на него ответственности, проводит вопросно-ответный анализ очередной подзадачи для перехода на очередной уровень пошаговой детализации Потоков работ.

8 Потоков работ

1. алгоритм, используемый



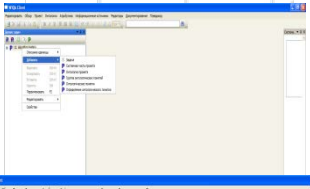
Открыть редактор OwnWIQA

2.Создать Проект

Проект → Создать Проект →Создать Здача (Новый Проект)

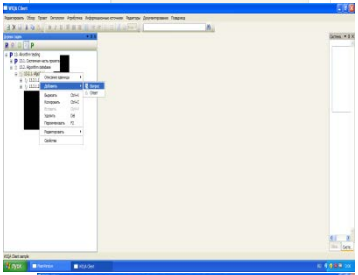
(Algorithm testing)

3. Добавит Здача



4. Добавит (Вапрос И Ответ)

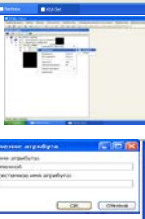
В атрибутах этой вопросно-ответной

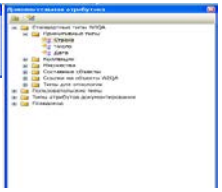


5.Редактировать Атрибуты

Открыть атрибуты этой задачи (через контекстное меню или сочетание клавиш Ctrl+E+A), найти атрибут WIQA.Models. свойство атрибута

К появившейся в вопросно-ответном дереве вопросно-ответной единице прикрепить стандартными средствами файл, который будет связан с соответствующимю





6.Дерево Задача

7 . Таблица Декларативное представление

Описание ситуации состоит из нескольких текстовых строк следующего формата: <предикат> (<субъект>; <объект>) <ассоциативная составляющая>.

Элементы <предикат>, <субъект> и <объект> представлены видом: <имя элемента>, <атрибуты>. Атрибуты для элемента пишутся после имени элемента через запятую и имеют необязательный характер. Количество свойств для элемента неограниченно.

Элемент <АС> содержит в себе только название составляющей или список названий через запятую: <название1>, <название2>. Ассоциативная составляющая относиться к предикату и пишется в конце описания. Ассоциативная составляющая имеет необязательный характер и может отсутствовать.

Предикат в строке с субъектом и объектом показывает отношение между ними. Если объект отсутствует, то предикат показывает свойство субъекта.

Все имена и названия в строке должны состоять из любых символов, кроме управляемых (, ; ( )).

Текстовое описание ситуации должно быть записано в QA-узле посредством включения в него других узлов с названиями текстовых строк.

Таблица 1. Декларативное представление

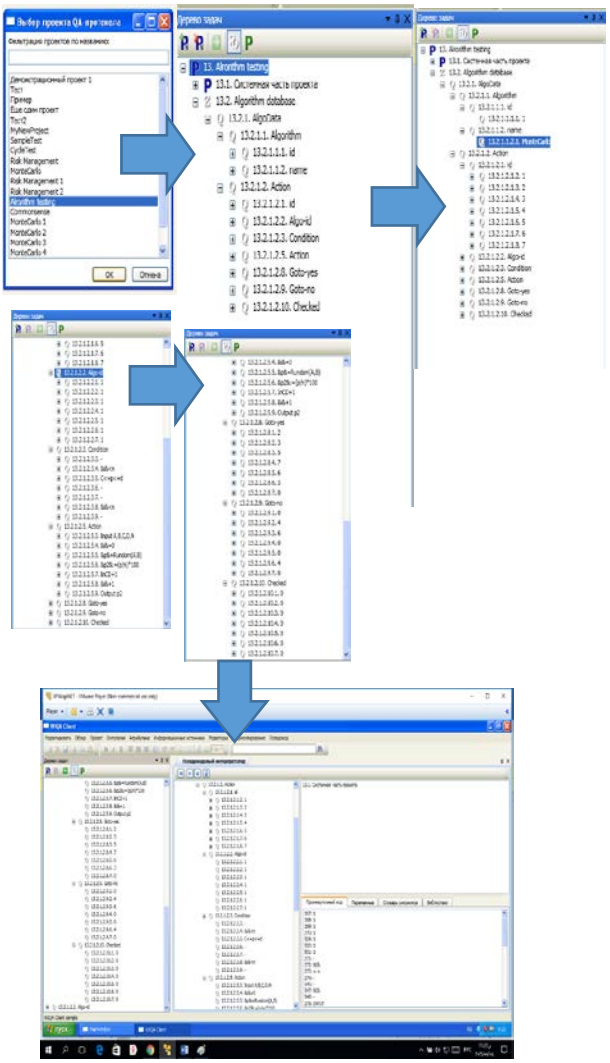
Id	Al-id	Condition	Action	Goto-yes	Goto-no	Checked
1	1	-	Input A,B,C,D,N	2	-	0
2	1	I< n	I=0	3	4	0
3	1	C<= p<=d	P= random (A,B)	5	6	0
4	1	-	P2 =(p / n) .100	7	-	0
5	1	-	InCD+1	6	-	0
6	1	I< n	I+1	3	4	0
7	1	-	Output p2	-	-	0

9. Отладка полученного кода в интерпретаторе

Для отладки и исполнения модели необходимо выбрать задачу, а затем выбрать пункт меню «псевдокод» -> «запустить интерпретатор». Данное меню представлено на рисунке 5.

Интерпретатор обладает большими возможностями по отладке и исполнению псевдокода. Важно отметить, что после отладки существует возможность вернуться к первому шагу, т.е. осуществить построение диаграммы по отлаженному коду. Типичный результат выполнения программы представлен ниже:

10 Открыть редактор и Результаты на WIQA



Заключение

В тестировании программного - это процесс устранения ошибок работы программного обеспечения. Отладка порой занимает столько же времени, сколько программист тратит на написание исходного кода. Программист устраняет одну ошибку, а взамен может появиться несколько новых, поэтому отладка является итерационным процессом.

Основные типы ошибок, которые устраняются на этапе отладки – это ошибки в логике и алгоритмах.

Основным средством отладки являлся встроенный в среду WIQA , позволяющий выполнять пошаговую трассировку, отслеживать значения

переменных в процессе выполнения программы, устанавливать точки или условия останова.

Список литературы

1. Соснин П.И. Псевдо-кодовое программное управление потоками работ в проектировании автоматизированных систем / П.И. Соснин, Ю. А. Лапшов, Маклаев В.А. // Научно-технический журнал «Автоматизация процессов управления». №3(29) 2012. Ульяновск: ФНПЦ ОАО НПО «Марс».
2. Соснин, П. И. Концептуальное моделирование компьютеризованных систем: учебное пособие/ П.И. Соснин. – Ульяновск: УлГТУ, 2008. – 198 с.
3. Соснин, П.И. Архитектурное моделирование автоматизированных систем: учебное пособие/ П.И. Соснин. –Ульяновск: УлГТУ, 2007. –146 с.
4. Sosnin, P. Question-Answer Expert System for Ship Collision Avoidance/P.I. Sosnin // In Proc of 51st International Symposium ELMAR – 2009, Zadar, Croatia, 2009. pp.185-188.
5. Sosnin, P. Question-Answer Means for Collaborative Development of Software Intensive Systems/Sosnin P.I. // Collection of scientific paper Complex Systems Concurrent Engineering, Part 3, Springer London, 2007, 151-
6. П. И. Соснин Министерство Образования И Науки Российской Федерации
7. http://www.cs.uu.nl/docs/vakken/mbr/node1.html
8. http://www.amzi.com/ARulesXL/docs/manual/index.htm
9. http://softwaretestingfundamentals.com/software-testing-methods/
10. http://www.tutorialspoint.com/software_testing/software_testing_methods.htm.

УДК 004.89, 629.7

РОБОТИЗИРОВАННЫЕ СВАРОЧНЫЕ КОМПЛЕКСЫ В СБОРКЕ ТРУБОПРОВОДОВ ЛЕТАТЕЛЬНОГО АППАРАТА

П.Ю. Павлов[40], П.И. Соснин[41]

Статья посвящена созданию средств онтологической поддержки работ на этапе технологической подготовки авиационного производства на основе использования прототипирования на уровне *CNC*-программ с повторным применением фрагментов кода, позволяющего накапливать опыт успешных решений в базе данных и повторно использовать при разработке управляющих программ для роботизированного сварочного комплекса.

Введение

В жизненном цикле *(ЖЦ)* производства трубопроводных систем летательного аппарата на этапе конструкторско-технологической подготовки производств *(КТПП)* инженерному персоналу предприятия приходится принимать большое количество технических решений и решать задачи изготовления трубопроводов из различных материалов (в основном металлы). Основным методом соединения фрагментов трубопроводов между собой является сварка.

Рассмотрим состав трубопроводов систем летательного аппарата *(ЛА)* и материалы, из которых они изготавливаются.

Трубопроводы входят в следующие системы: кондиционирования воздуха; гидравлической; топливной; пневматической; противообледенительной; пожаротушения; аварийной подачи кислорода и прочие.

Трубопроводы состоят: труба (прямая или гнутая); патрубок; тройник; фитинг; проставка; кронштейны крепления.

Основные материал, из которых изготавливаются трубопроводы: сталь; титан; алюминий; магний; медь.

[40] *432071, Ульяновск, ул. Крымова, 10а, АО «УКБП», e-mail: pavel.y.pavlov@mail.ru*

[41] *432027, Ульяновск, ул. Северный Венец, 32, УлГТУ, e-mail: sosnin@ulstu.ru*

ЖЦ изготовления трубопроводов систем ЛА

Далее рассмотрим жизненный цикл изготовления трубопроводов на примере предприятия АО «Авиастар-СП»

Производство трубопроводов начинается с получения электронной конструкторской документации *(ЭКД)* от разработчика *ЛА* и запуска инженером-конструктором отдела систем *ЛА*, входящим в состав управления главного конструктора *(УГК)*. В случае выявления замечаний к конструкторской документации *(КД)* инженер-конструктор делает запрос разработчику *ЛА* или вносит изменения в *КД* и проходит процедуру согласования.

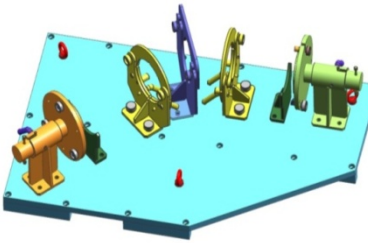
Запущенная *ЭКД* загружается в базу данных электронного определения изделия *(БД ЭОИ)*. Далее к *ЭКД* получает доступ управление главного технолога *(УГТ)* для технологической проработки и инженерной подготовки производства.

Инженерная подготовка производства начинается с анализа *КД*, технических требований, технических условий *(ТУ)* разработчика и т. д. В случае выявления замечаний к *КД* или невозможности изготовления ввиду отсутствия технологий или оснащения, инженер-конструктор дает замечания к *КД*, и *УГК* вносит изменения в *КД* и проходит процедуру согласования с разработчиком *ЛА*.

После получения *КД* инженер-технолог проводит анализ на возможность изготовления, и в случае, если оборудование и освоенная технология не позволяют изготовить трубопровод, разработчику *ЛА* выставляются предложения по изменению конструкции с целью возможности ее изготовления. Если разработчик *ЛА* отвергает предложения, то начинается поиск предприятия-подрядчика. Если возможность изготовления имеется, то инженер-технолог на основании нормативной документации и своего личного опыта выбирает технологию, производит технологическое членение, разрабатывает порядок сборки и формирует *ТП*. Основная задача технолога на этапе формирования *ТП* – выбрать вид *ТО*. Основные применяемые виды *ТО* в настоящее время – это специальное технологическое оснащение (СТО), универсальные сварочные приспособления *(УСвП)* и роботизированный сварочный комплекс *(РСК)*. *СТО* с трубопроводом представлены на рисунке 1.

В *ТП* указывается способ сварки, порядок сборки деталей, места фиксации, технологические припуски и т. п. При необходимости *СТО* или *УСвП* инженером технологом открывается ведомость подготовки производства *(ВПП)* и разрабатывается *ТЗ* на *ТО*. В данном *ТЗ* отражается состояние поставки деталей, точность изготовления *ТО*, краткий *ТП*

сборки, базы фиксации сборочной единицы, способы крепления инструмента и прочие требования. В случае использования *РСК* [7] инженер-технолог формирует *ТЗ* на разработку управляющей программы *(УП)*.



а) б)

Рис. 1. *СТО* (а) и изготавливаемая часть трубопровода (б)

Далее *ТЗ* передается в конструкторское бюро при отделе *УГТ*. Инженер-конструктор на основании *КД*, *ТЗ* на *СТО* и нормативных документов разрабатывает *КД* на *СТО*. Здесь перед инженером появляется задача поиска и выбора оптимальных конструктивных решений. В случае нахождения ошибок или невозможности применения технологии, выбранной инженером-технологом, инженер-конструктор при *УГТ* выставляет замечания и производится пересмотр *ТП* и *ТЗ* или *КД*.

В случае использования *РСК ТЗ* передается в отдел по разработке *УП*. Инженер-программист на основании *КД*, *ТЗ* на *УП* и нормативных документов разрабатывает *УП*. Разработка *УП* состоит из выбора оптимального положения деталей трубопровода в пространстве, выборе оптимальных точек захвата деталей манипуляторами и выборе оптимальных точек сварки. При обнаружении ошибок в *ТП* и *ТЗ* или *КД* инженер-программист выставляет замечания и производится пересмотр *ТП* и *ТЗ* или *КД*.

После завершения этапа инженерной подготовки *ТП*, *КД* и *УП* передаются в подразделения по производству средств ТО. Изготавливаются средства *ТО*, производится отработка технологии или части технологии, подготавливается оборудование и т. д. На данном этапе также могут выявляться недоработки в *КД*, *ТП*, *СТО*, но стоимость выявленных ошибок сильно увеличивается за счет того, что могут быть потрачены материальные ресурсы и время работы оборудования и персонала. Готовая *ТО* передается на центральный комплектовочный склад *(ЦКС)*.

Параллельно может быть запущено изготовление макета трубопровода для настройки *ТО* и последующего контроля размеров и формы готового изделия.

При использовании *РСК* технологическая подготовка может быть минимальной за счет значительного уменьшения металлоемкости применяемого *ТО* (*РСК* перестраивается по *УП* для выполнения конкретных сборочных работ) и в отсутствии необходимости изготавливать макет, т. к. дополнительная настройка при помощи макета не потребуется, а контроль размеров и формы можно проводить при помощи специальных лазерных трекеров или *3D* сканеров [16].

На основании *КД*, *ТП* и условий поставки изготавливаются детали трубопровода в цехах основного производства: в заготовительно-штамповочном производстве *(ЗШП)* изготавливаются детали из листовой заготовки и в механическом цехе – детали из труб. Изготовленные детали проходят проверку в бюро технического контроля *(БТК)* и передаются в промежуточный склад *(ПРОСК)* цеха сборки трубопроводов.

Цех сборки трубопроводов, получив детали, проверяет комплектность и по план-графику начинает сборку трубопровода. Сборка начинается с настройки *ТО* и подготовки поверхностей под сварку. После чего детали фиксируются в пространстве при помощи *ТО*, производится точечная сварка (обычно в 3–4 точках), после чего трубопровод извлекается из *ТО* и передается на участок полуавтоматической сварки, где производится окончательная сварка. Сваренный трубопровод проходит приемку в *БТК*.

Длительность *ЖЦ* трубопровода от запуска *КД* до изготовления составляет от 3 месяцев до 1 года. Это обусловлено не только факторами прямо валяющими на длительность (сложность технологии, большая трудоемкость изготовления *ТО*, деталей и т. д.), но и рядом косвенных факторов (наличие инструмента, материалов, загруженность производства, болезнь работников и т. д.).

В случае возникновения ошибки, приходится возвращаться к предыдущим этапам *ЖЦ*, производить корректировку *ТП*, повторное изготовление *СТО*, деталей, макета, что приводит к значительному увеличению цикла изготовления трубопровода и цикла сборки *ЛА* [3, 4]. Поэтому требуется свести к минимуму количество ошибок, возникающих на стадии разработки *ТП*, выбора вида *ТО* и написания *ТЗ* на разработку или использование *ТО*.

Часть *ЖЦ* трубопровода представлена на рисунке 2.

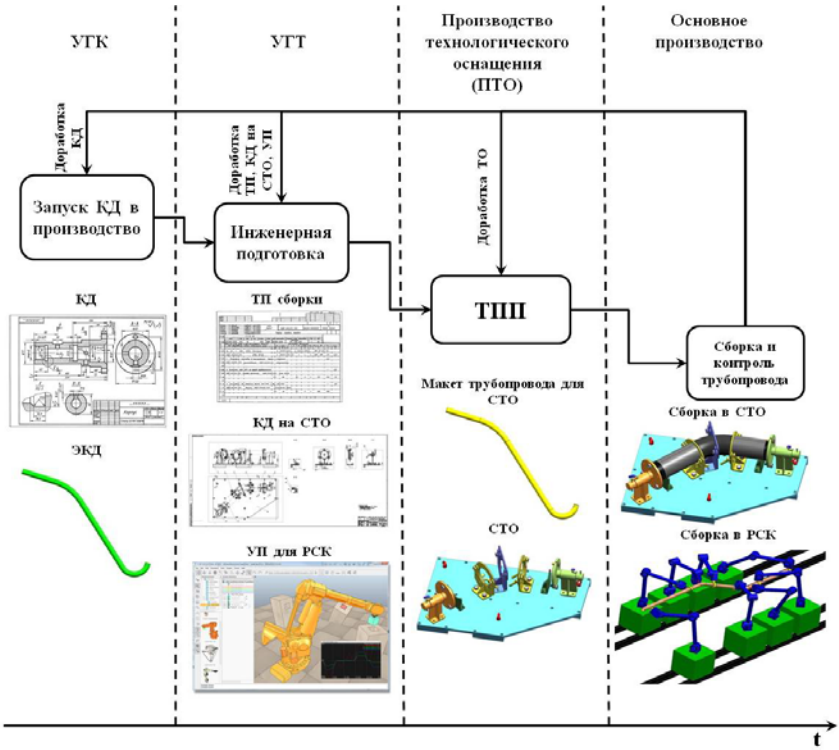
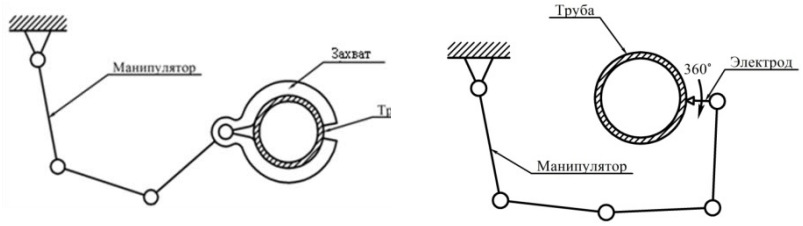


Рис. 2. Часть схемы *ЖЦ* трубопровода на авиационном предприятии

Роботизированные комплексы в сборке трубопроводов

РСК в общем виде представляет систему, состоящую из звеньев (стержней) [5, 13] на подвижном основании.

Схема работы манипуляторов представлена на рисунке 3.



а) б)

Рис. 3. Схема манипуляторов: а) с захватом; б) с электродом

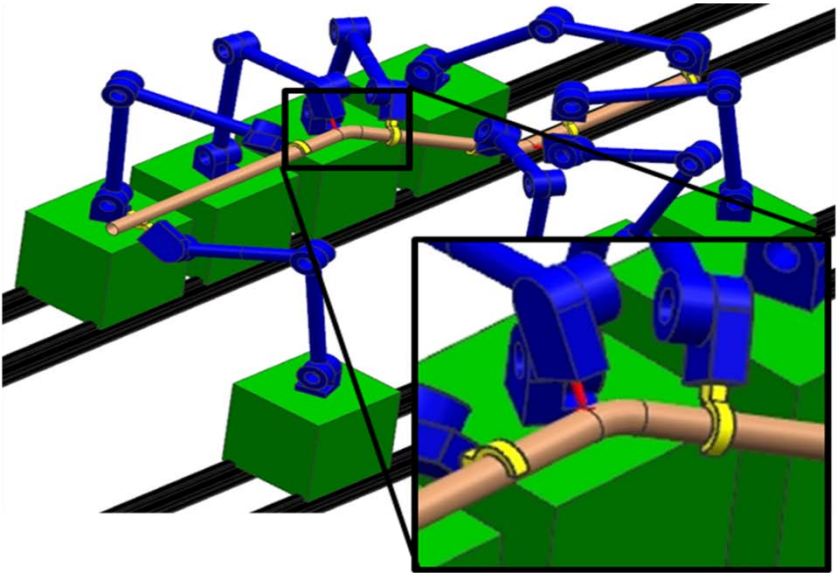


Рис. 4. Пример использования *РСК* для сварки трубопровода

Пригодность применения роботов для сварки фрагментов трубопроводов была проверена на производственной площадке в г. Тула, с участием представителей ООО «Нью Лайн Инжиниринг», ЗАО «Авиастар-СП», ОАО «ОАК».

Тестовая система состояла из четырех манипуляторов Fanuc 120iC (рис. 5), установленных на полу, захватов для труб, прикрепленным к фланцам роботов и системы управления. Прихватка труб осуществлялась вручную.

Рис. 5. Манипулятор Fanuc 120iC

Результат сварки трубопровода показан на рисунке 6.

Изготовленный трубопровод прошел приемку отделом технического контроля.

Рис. 6. Трубопровод после сварки

Прототипирование в роботизированной сборке трубопроводов

Одной из проблем, стоящих на пути роботизации, является высокая трудоемкость программирования робота с ручным обучением точек программной траектории сварки. С увеличением числа роботов и

номенклатуры выпускаемых трубопроводов затраты на программирование могут стать становятся недопустимо большими. Для уменьшения трудоемкости программирования часто используются системы автономного программирования (Off-Line Programming, Simulation) для роботов (Рис. 7).

Для дальнейшего уменьшения трудоемкости программирования *РСК* может быть использование прототипирование на уровне *CNC*-программ с повторным применением фрагментов кода.

Условно *УП* для *РСК* можно разделить на следующие типовые фрагменты:

– захват детали;
– ориентация деталей в пространстве;
– подвод инструмента в рабочую зону;
– сварка;
– вывод инструмента из рабочей зоны;
– освобождение трубы.

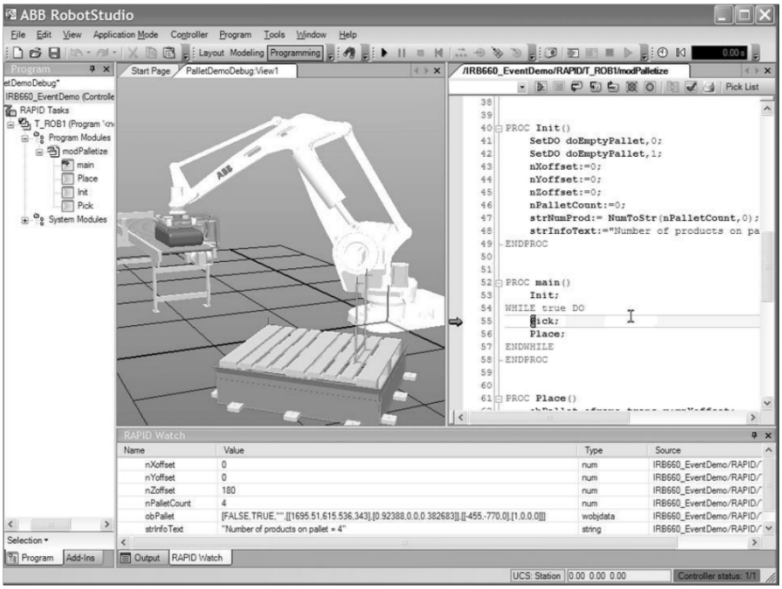


Рис. 7. Система автономного программирования RobotStudio (ABB)

Представленные фрагменты кода сохраняются в базе данных *(БД)* предприятия и, тем самым, происходит накопление опыта решений,

принятых инженерами-программистами (рис. 8). С заполнением *БД* могут быть выделены дополнительные типовые фрагменты кода.

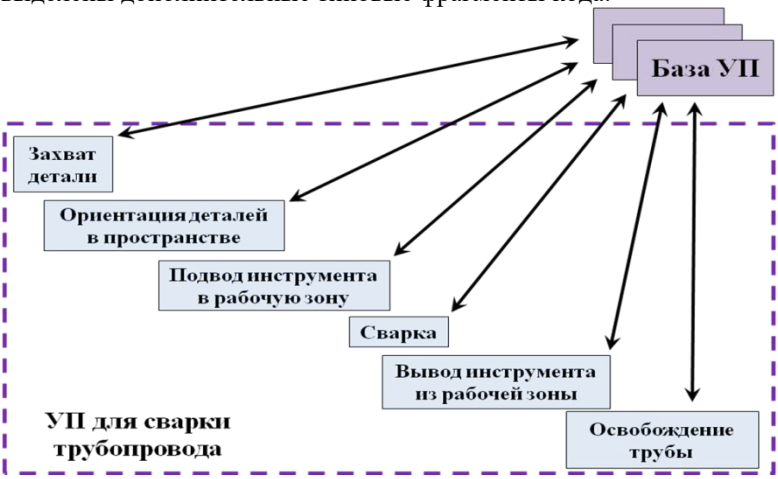


Рис. 8. Процесс формирования базы *УП* для *РСК*

Рассмотрим алгоритм работы инженера-программиста *РСК* с поддержкой ***WIQA.***

Инженер-программист выбирает решение из предлагаемых вариантов фиксаторов для *РСК*, выбора точек сварки, положения манипуляторов в пространстве и т. д. После чего производится симуляция сборки трубопровода с последующей материализацией. Затем проводится проверка принятых решений на противоречивость и выполнимость, и, если решения удовлетворяют требованиям, то они заносятся в *ТП* и *ТЗ*. Если же решения не удовлетворяют требованиям, то производится поиск решения, пока не будут выполнены требования по точности сборки.

Процесс разработки *УП* для *РСК* с поддержкой ***WIQA*** представлен на рисунке 9.

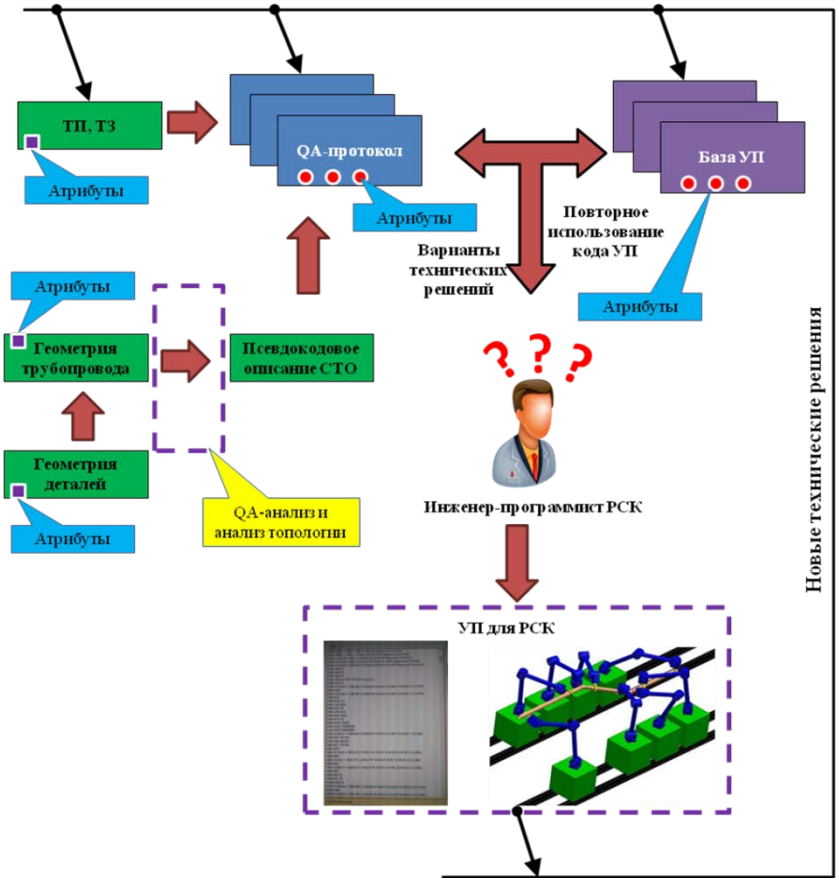


Рис. 9. Процесс разработки *УП* для *РСК* с поддержкой ***WIQA***

Принятые технические решения в виде кода *УП* записываются в *БД* для повторного использования кода и, так же, происходит формирования типовых фрагментов кода.

Заключение

В данной статье рассмотрен *ЖЦ* трубопроводов систем *ЛА* на этапе *КТПП* и предложен подход, основанный на использовании прототипирования на уровне *CNC*-программ с повторным применением фрагментов кода. Разработаны краткий алгоритм работы инженера-программиста РСК с использованием системы ***WIQA***, что позволит заложить основы для создания базы опыта в виде фрагментов кода для

повторного использования наиболее удачных решений и увеличить уровень естественности во взаимодействии человека и машины при разработке *УП* для *РСК*.

Использование базы фрагментов кода *УП* для *РСК* приведет к уменьшению вероятности принятия ошибочных решений на стадии инженерной подготовки за счет проверки на ошибки, увеличения производительности труда за счет повторного использования успешных фрагментов кода в качестве технических решений, что в конечном итоге приведет к снижению издержек, вызванных неверными или непродуманными и непроверенными решениями до стадии производства и уменьшению сроков *КТПП*.

Список использованной литературы

1. Боргест Н.М. Онтология проектирования. Теоретические основы. Ч. 1. Понятия и принципы. / Боргест Н.М., 2-е изд., Самара: СГАУ, 2010. 91 с.

2. Комаров В.А. и др. Концептуальное проектирование самолёта: учеб. пособие. / Комаров В.А. и др., 2-е изд., Самара: СГАУ, 2013. 120 с.

3. Коптев А.Н., Коптев А.А. Адаптивное взаимодействие в системе «разработчик-производство» // Известия Самарского научного центра РАН. Самара. 2014. № 1 (5) (16). С. 1422–1437.

4. Маклаев В. А. , Соснин П. И. Создание и использование автоматизированной базы опыта проектной организации / Маклаев В. А. , Соснин П. И., Ульяновск: УлГТУ, 2012. 360 с.

5. Накано Э. Введение в робототехнику. / Накано Э., под ред. А.М. Филатова к.т.н., перевод А.И. Логинова А.М.Филатова., М.: Мир, 1988. 334 с.

6. Павлов П.Ю., Соснин П.И. Онтологическая структуризация в параллельном инжиниринге проектирования сборочных приспособлений для летательных аппаратов // Известия Самарского научного центра РАН. Самара. 2016. № 1 (2) (18). С. 373–377.

7. Смирнов С.В. Онтологическая относительность и технология компьютерного моделирования сложных систем // Известия Самарского научного центра РАН. Самара. 2000. № 1 (2). С. 66–71.

8. Смирнов С.В. Опыт создания средств семантического моделирования и проектирования на массовой программной платформе // Матер. 5-й Междунар. конф. «Открытые семантические технологии проектирования интеллектуальных систем» (OSTIS-2015). Минск : БГУИР. 2015. С. 413–416.

9. Соснин П. И. Вопросно-ответное программирование человеко-компьютерной деятельности / Соснин П. И., Ульяновск: УлГТУ, 2010. 240 с.

10. Соснин П.И. Персональная онтология профессионального опыта // Матер. 4-й Междунар. конф. «Открытые семантические технологии проектирования интеллектуальных систем» (OSTIS-2014). Минск : БГУИР. 2014. С. 147–154.

11. СТП 687.06.0694-2011. Эталонирование трубопроводов гидрогазовых и топливных систем, обеспечение взаимозаменяемости. Технические требования

к эталонам труб и патрубков, порядок отработки, технического контроля и хранения. / СТП 687.06.0694-2011., 53 с.
12. СТП 687.07.0873-2004. Система качества. Технологическая подготовка производства. Изготовление и применение плазово-шаблонной оснастки. / СТП 687.07.0873-2004., 168 с.
13. Юревич Е.И. Основы робототехники. / Юревич Е.И., 2-е изд., С-Пб.: БХ13–Петербург, 2005. 416 с.
14. Allemang D., Hendler J. Semantic Web for the Working Ontologist: Effective Modeling in RDFS and OWL / Allemang D., Hendler J., 2-е изд., 225 Wyman Street, Waltham, MA 02451, USA: Morgan Kaufmann Publishers is an imprint of Elsevier, 2011. 369 с.
15. Staab S., Studer R. Handbook on Ontologies / Staab S., Studer R., Springer, 2009. 832 с.
16. Система контроля трубопроводов URL:http://www.hexagonmetrology.eu/rus/--_~~~_1815.htm#.Vpv22i_EPp0 (Дата обращения: 17.01.2016).
17. Sosnin P. Question-answer programming and modeling in expert systems. In Proceedings of the IASTED International Conference on Artificial 2009 IASTED International Conference on on Artificial Intelligence and Applications, AIA 2009, Innsbruck, 2009, pp. 208-213.
18. Sosnin P. Question-answer approach to human-computer interaction in collaborative designing. Cognitively Informed Intelligent Interfaces: Systems Design and Development 2012, pp. 157-176.

УДК 004.896

АНАЛИЗ РАЗЛИЧНЫХ ПОДХОДОВ К РЕШЕНИЮ И ПЕРСПЕКТИВ РЕАЛИЗАЦИИ ЗАДАЧ ФОРМИРОВАНИЯ И ИСПОЛЬЗОВАНИЯ ПРОЕКТНЫХ ОНТОЛОГИЙ

А.А. Пушкарева[42]

В данной статье анализируются существующие походы к решению задач формирования и использования проектных онтологий. Автор делает акцент на перспективах развития уже реализованных программных средств, позволяющих приблизиться к решению данных задач, а также обрисовывает основное проблемное поле данной области.

Введение

Разработка автоматизированных систем (АС), интенсивно использующих программное обеспечение, исключительно сложна, носит коллективный характер, осуществляется в корпоративных средах и часто приводит к срывам в сроках работ, превышениям их запланированного финансирования, уменьшению объёма реализованных функциональностей АС, а достаточно часто – прекращениям работ без достижения положительных результатов. При этом, отдельный вид работ, который необходимо выполнять качественно и в соответствии со стандартами и нормами – это разработка проектной документации, неотъемлемой частью которой является терминология проекта, которая варьируется в зависимости от предметной области.

В целях избегания ошибок в использовании терминологии применяется контролируемая лексика. Наиболее удобно формой представления такого словаря является онтологическая форма, поскольку она позволяет представлять знания в формальной структуре, которая упрощает их компьютерную обработку [1].

1. Проблемное поле и актуальность задач, связанных с разработкой проектных онтологий

[42] *432027, Ульяновск, ул. Северный Венец, 32, УлГТУ, e-mail: a.push1206@gmail.com*

Схематично проблемное поле, изложенное во введении, представлено на рис. 1.

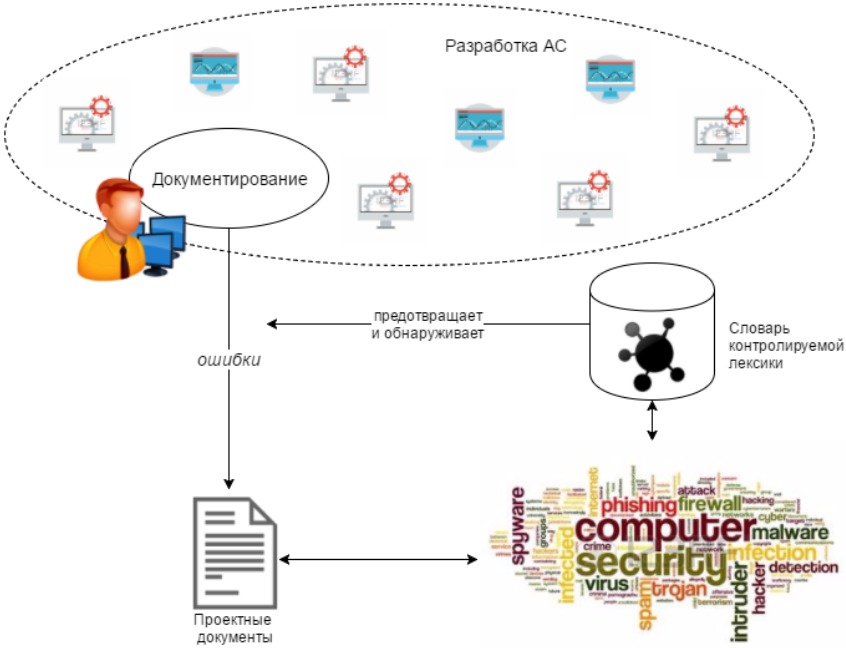


Рис.1. Проблемное поле

Включение прикладных онтологий в состав баз опыта для систематизации их содержания должно способствовать повышению эффективности оперативного взаимодействия проектировщиков с базами опыта в процессах разработки систем с ПО. Таким образом, наша главная и наиболее общая цель – повышение степени успешности разработки автоматизированных систем.

Для достижения этой цели необходимо решить две важнейшие задачи – сформировать онтологию проекта и создать средства, позволяющие её эффективно использовать.

2. Задача формирования проектных онтологий: основные этапы и механизмы

В рамках данной задачи мы рассматриваем процесс автоматизированного формирования проектных онтологии на основе текстов, написанных на естественном языке и принадлежащих к одной

предметной области. В качестве инструментально-моделирующей среды используем комплекс OwnWIQA, специфику которого определяет псевдопрограммирование, ориентированное на прецеденты. В данный комплекс включены различные плагины, одним из которых является плагин для взаимодействия с онтологией.

Задачу автоматизированного формирования онтологии можно разделить на две части:

- формирование словаря контролируемой лексики (словаря концептов);
- установление связей между концептами и формирование предположений относительно их типов (систематизация).

В ранее опубликованных работах [2] мы подробно останавливались на процессе формирования словаря концептов – обобщенно его можно проследить на рис. 2, который представляет собой внешний вид формы взаимодействия с уже реализованными программными агентами в среде OwnWIQA.

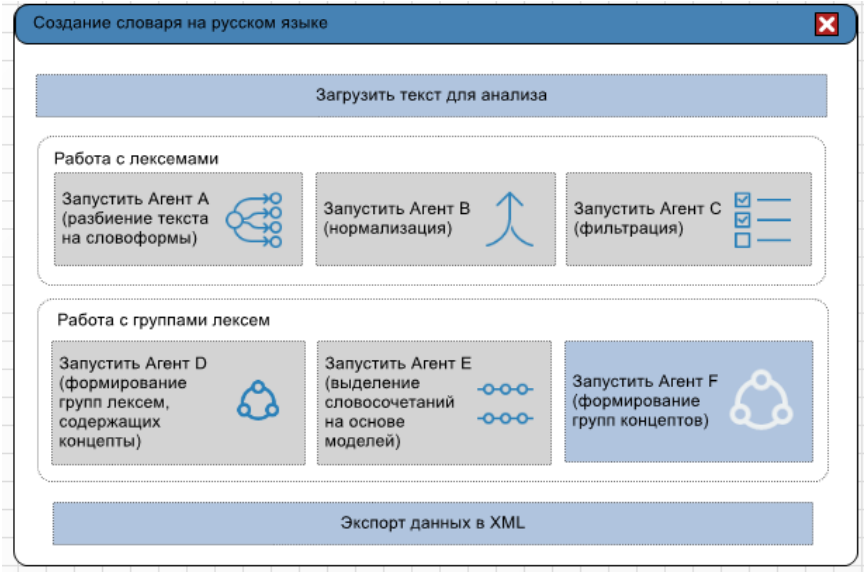


Рис.2. Общий вид совокупности средств формирования набора концептов

Отметим, что реализованная совокупность средств предназначена только для анализа текстов на русском языке и обладает следующим потенциалом для расширения функциональности:

- обработка текстов, написанных на других языках (английский, немецкий, французский и др.);
- реализация альтернативных механизмов фильтрации (в данном случае используется словарь стоп-слов) с учетом частеречной принадлежности слов и/или статистического подхода;
- автоматизированное формирование определений для концептов;
- создание дополнительной опции в плагине «Онтология» среды OwnWIQA, который связывал бы процесс формирования онтологии с плагином «Документация» и др.

Нашим коллективом, который также подробно занимался вопросом систематизации концептов в среде OwnWIQA, предлагаются **два альтернативных механизма установления связей между концептами готового словаря:**

- синтаксемный анализ;
- предикативный анализ.

Более подробно эти идеи описаны в статьях [3] и [4], однако отметим, что данные механизмы рассматриваются только для русского языка. При этом, предложенные алгоритмы активно используют формализацию синтаксической структуры предложения на естественном языке, поэтому при расширении совокупность средств формирования онтологии в пользу ещё одного языка (в нашем случае, английского), необходимо пересмотреть их эффективность, т.к. синтаксические зависимости в разных языках значительно различаются.

3. Задача использования проектных онтологий

Использование проектных онтологий – широкая задача, которая может рассматриваться с различных сторон. Онтологии проектирования могут применяться для следующих видов работ:

- поиск и предотвращение ошибок во вновь разрабатываемой проектной документации;
- обучение предметной области;
- поиск релевантной информации в базе знаний (поиск успешных прецедентов).

Что касается последнего, данная область является особенно актуально и обладает большим потенциалом для изучения, поскольку, как отмечено в [5], систем автоматической обработки документов с целью извлечения знаний в настоящее время не создано.

Инструментальная среда OwnWIQA является удобным инструментом для реализации средств, предназначенных для решения подобных задач. Она позволяет использовать визуальный редактор для наглядного

представления онтологии в виде графа, интегрироваться с плагином «документация» и осуществлять проверку документов в оперативном режиме, а также извлекать знания в форме прецедентов.

Заключение

Таким образом, задачи формирования и использования проектной онтологии не теряют свою актуальность. Каждый тип онтологий, в частности онтология проектирования, обладает своими особенностями, что позволяет обозначить широкое поле для исследований. Данные задачи обладают большим потенциалом для изучения, а реализация средств, обеспечивающих автоматизированное взаимодействие проектировщика с онтологиями, способна значительно упростить процесс разработки АС.

Список литературы

1. Документирование по ГОСТ 34* — это просто [Электронный ресурс]. – Режим доступа: http://habrahabr.ru/post/122700/
2. Pushkareva A., Sosnin P. Toolset to create a domain ontology: forming a set of concepts // Interactive systems: Problems of Human - Computer Interaction. – Collection of scientific papers. − Ulyanovsk: USTU, 2015. − p. 186-193
3. Васильева Т. В. Алгоритмизация процесса формирования онтологических связей на основе ближнего контекста // Информатика и вычислительная техника 2015. VII Всероссийская научно-техническая конференция аспирантов, студентов и молодых ученых ИВТ-2015 (Россия, г. Ульяновск. 25-27 мая 2015 г.) : сборник научных трудов / под общей ред. В.Н. Негоды. – Ульяновск : УлГТУ, 2015. – с. 71-77
4. Vasileva T., Sosnin P. Linguistic approach to define concept relations for domain ontology // Interactive systems: Problems of Human - Computer Interaction. – Collection of scientific papers. − Ulyanovsk: USTU, 2015. − p. 236-246
5. Титов В.В. Интерактивный справочник. [Электронный ресурс]. – Режим доступа: http://serendip.narod.ru/order/sprav/sprav0.htm
6. Норенков И.П. Интеллектуальные технологии на основе онтологий / Информационные технологии, 2010, № 1, с. 17-23.
7. Боргест Н.М. Онтология проектирования. Теоретические основы. Часть 1. Понятия и принципы. [Электронный ресурс]. – Режим доступа: http://www.ssau.ru/files/education/uch_posob/Онтология%20проектирования-Боргест%20НМ.pdf

УДК 004.896

Исследование разработки средств управления человеко-компьютерными прерываниями в вопросно-ответной моделирующей среды WIQA.NET

Т.В. Старшова [43]

В данной статье рассмотрен выбор наилучшего решения для управления человеко-компьютерной деятельностью на примере использования Agile-методов, реализованных в программной среде OWN.WIQA и WIQA.NET. Так же произведена качественная оценка реализуемых модулей.

Введение

В определенных условиях человек может выполнять несколько действий одновременно, однако человеческие возможности ограничены, и попытки одновременного выполнения нескольких задач могут привести к ошибкам

Если пользовательский интерфейс спроектирован с учетом особенностей человеческой деятельности, то пользователь может в определенный промежуток времени успешно выполнить несколько задач, переключаясь между ними [1]. Когда же пользовательский интерфейс спроектирован без учета этих человеческих ограничений, попытки одновременного выполнения нескольких задач могут привести к значительному снижению производительности работы пользователя и появлению ошибок в решении критически важных задач [2].

Для вопросно-ответного процессора WIQA.NET проблема прерываний является актуальной, т.к. данная система предназначена для работы в компьютерной сети и имеет подсистему коммуникации между проектировщиками;

Целью данной статьи является исследование разработки средств управления прерываниями при работе пользователя в средах автоматизированного проектирования в условиях большого количества прерываний, т.е. как внутренних, так и внешних.

[43] *432027, Ульяновск, ул. Северный Венец, 32, УлГТУ, tvstarshova@gmail.com*

Для достижения поставленной цели потребовалось решение следующих задач:

- Анализ ситуаций (сценариев) прерываний в процессе выполнения задачи пользователем;
- Разработка системы обработки прерываний в многозадачной вопросно-ответной среде OWN.WIQA;
- Организация GUI - интерфейса для реализации работы с заметками.

Основные бизнес-процессы

Реализуемый модуль имеет двухуровневую архитектуру (первый - введенные данные пользователя, второй - ответная реакция системы).

В системе предлагается выделить следующие функциональные подсистемы:

1. подсистема сбора, обработки и загрузки данных;

2. подсистема хранения данных, которая предназначена для хранения данных в структурах, нацеленных на принятие решений [3];

Нижеуказанная схема представляет собой бизнес-процессы работ с задачами в вопросно-ответной моделирующей среде WIQA [4] (см. рис.1).

1. Формирование карточки образа задачи;

2.Определение принадлежности карточки к определенной роли;

3.Определение постановки задачи (создание заметки)

4.Формирование очереди заметок;

5.Реализация области сохранения данных карточки для восстановления после процесса прерывания выполнения текущей задачи;

6.Сохранение карточки-прерванной задачи в очереди задач с установкой соответствующего приоритета;

7. Сохранение ранее созданной заметки в очередь заметки для последующего её редактирования;

8.Формирование заметки к текущей выполняемой задаче.

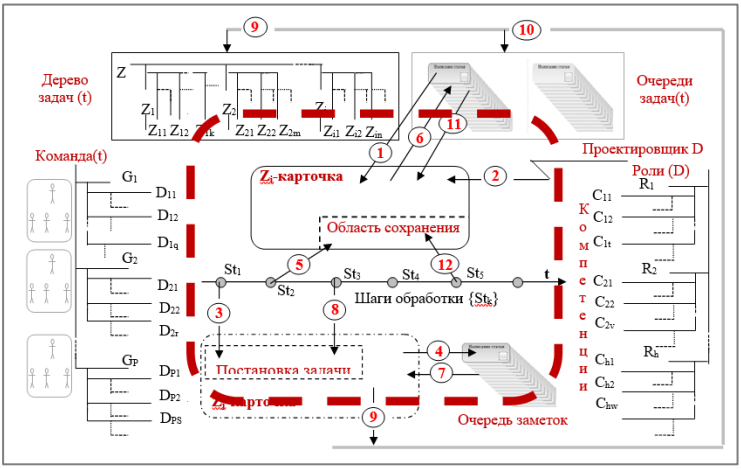


Рис. 1. «Жизненный цикл» обработки прерываний в вопросно-ответной среде WIQA

Пункты 3,4,5,7 относятся к реализации модуля «Заметки», т.е. пользователь может в любой момент времени создать необходимую ему заметку (3), которую впоследствии затем может перевести в задачу для конкретной группы пользователей (2).

Ниже описан пример реализации добавления заметок. Интерфейс добавления представлен в следующем виде (см. рис. 2):

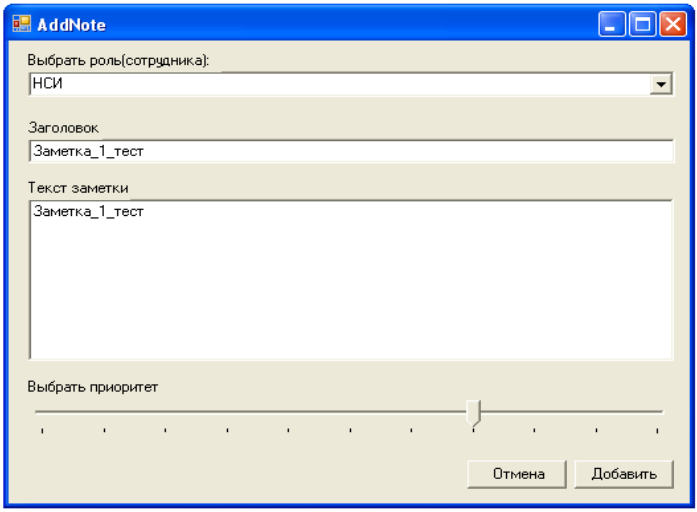


Рис. 2. Компонент WindowsForm для добавления заметки

При необходимости добавления заметки пользователем, вызвав форму, он заполняет все необходимые поля: выбирает роль, для которой будет добавлена заметка, заполняет заголовок и описание заметки, потом выбирает с помощью ползунка в трекбаре приоритет текущей заметки. После чего кликает по кнопке «Добавить» – и заметка затем будет добавлена в базу «БД поручений» и будет отображаться на форме отображения всех заметок.

Для того, чтобы отобразить все заметки, хранящиеся в БД, необходимо выполнить следующую последовательность действий (см.рис.3):

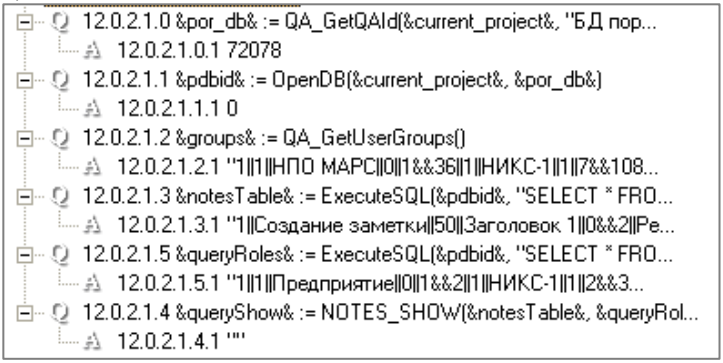


Рис. 3. Процедуры отображения заметок в WIQA.NET

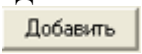
В вопросно-ответной среде необходимо выполнить последовательность следующих процедур:

1. Выполнить подключение к БД (в данном случае, подключение выполняется идентификатору ‘72078’);
2. Открыть БД по идентификатору;
3. Произвести запрос на получение списка групп пользователей;
4. Далее, каждой группе пользователей вывести соответствующий им список заметок;
5. Следующим шагом выполняется повторное получение списка ролей пользователей (для дальнейшего вывода в форму добавления заметок);
6. И, наконец, последним шагом, вызывается метод инициализации компонента WindowsForm для отображения формы со списком всех доступных заметок.

Моделирование тестовых случаев проверки работы системы

Для того, чтобы потом дать качественную оценку выполнения разработанной системы, необходимо эмулировать различные тестовые случаи для проверки системы. Данная процедура выполняется для того, чтобы позже можно было произвести выполнение поставленных задач для тестовых случаев поэтапно, то есть выполнить трассировку выполнения программы.

Таблица 1. Тестирование сценария на добавление заметки (позитивный тест)

Добавление заметки (позитивный тест)				
	Исходное состояние	**Действие**	**Ожидаемый результат**	**Код результата (passed/failed)**
1	Система работает стабильно, перед пользователем инициализирована и выведена форма с очередью заметок	Добавить новую заметку	-	…
1.1		Клик по кнопке «Добавить заметку»	Вывод формы для добавления заметки	passed
1.2		Заполнение полей необходимыми данными (выбор роли, заполнение заголовка и описания заметки, выбор приоритета)	-	passed
1.3		Клик по кнопке «Добавить» 	Заметка добавлена в базу	passed

Заключение

В силу специфики своей деятельности проектировщик тратит значительную часть своего рабочего времени на процессы коммуникации с другими проектировщиками, а также на другие отвлекающие его от

основной задачи процессы. Все эти процессы являются причиной прерываний.

В рамках исследования данной работы были выполнены следующие цели и задачи: была произведена разработка средств управления прерываниями при работе пользователя в средах автоматизированного проектирования в условиях большого количества прерываний.

Так же была произведена качественная оценка реализуемых модулей с помощью комбинированного тестирования – позитивных и негативных тестовых случаев.

Список литературы

1. Mary Czerwinski and Eric Horvitz. An Investigation of Memory for Daily Computing Events. Microsoft Research, One Microsoft Way, Redmond, WA 98052 USA 46. The effect of interruptions on flight crew performance: ASRS reports. Diane L. Damos, Ph.D. Damos Research Associates, Inc.]
2. Lapshov Y.A. Human interruptions management in corporate modeling environment WIQA.NET// Interactive Systems and Technologies: the Problems of Human-Computer interaction. Volume III. – Collection of scientific papers. – Ulyanovsk: UlSTU, 2009. – p..L. Erlbaum Associates Inc, USA. 2002.
3. Лорьер Ж.-Л. Системы искусственного интеллекта: Пер. с французского. – М.: Мир. – 1991. – 568 с., ил.
4. Соснин П.И. Вопросно-ответное программирование человеко-компьютерной деятельности. УлГТУ, Ульяновск – 2010.

УДК 004.896

ОБЗОР РАЗРАБОТКИ БИЗНЕС-ЭКСПЕРИМЕНТА

Т.В. Старшова [44]

В данной статье основные критерии методики построения бизнес-эксперимента. Так же выявлены и представлены основные постановочные пункты для запуска бизнес-эксперимента.

Введение

Многие крупные компании и корпорации не достигают существенного роста из-за того, что не вводят новшества или не реализуют новые идеи. В свою же очередь, введение инноваций несут риски следующего значения: новые допущения могут работать недостаточно хорошо, ключевую идею может не принимать заказчик и т.п. Например, такие крупные компании как Toronto-Dominion, CKE Restaurants, eBay, проводят внедрение инноваций далеко за пределами области применения, чтобы произвести качественную оценку для определения наибольшей выгоды внедрения [1] .

В данном случае проведение бизнес-экспериментов полезно: при анализе тестовых случаев и ошибок становится ясно, как будет «вести себя» та или другая инновация на крупном проекте, к тому же, появившиеся ошибки позволяют дополнительно оптимизировать целевой продукт и устранить обнаружившиеся ошибки.

В данном контексте, бизнес-эксперименты могут включать в себя как простые тесты (например, изменение информации основного контента сайта), так и боле сложные, комбинированные – к примеру, тестирование нового продукта в целевой аудитории и т.д.

Основные интеллектуальные риски

Проведение бизнес-экспериментов чреваты неопределенностью и риском, так как нет точной возможности узнать, насколько успешным будет данный эксперимент. Например, компания «Capital One»

[44] *432027, Ульяновск, ул. Северный Венец, 32, УлГТУ, tvstarshova@gmail.com*

Проводит десятки тысяч экспериментов каждый год, чтобы улучшить способы привлечения клиентов. При этом Capital One превратилась из небольшого подразделения в независимую компанию с капиталом в 19 млрд долларов.

Для того, чтобы поставить «бизнес-эксперимент» требуется две вещи: контрольная группа и механизм обратной связи. Многие компании пренебрегают тестированием нового продукта на целевой аудитории и запускают новые приложения сразу по всей клиентской базе. Наиболее лучшим решением является тот случай, когда рандомизированно выбрана контрольная группа.

Механизм обратной связи включает в себя два фактора обратной связи: поведенческий и восприятие. Поведенческие показатели напрямую зависят от использования инновации. Фактор восприятия чаще всего заключается в проведении опросов, исследований в специальной фокус-группе и т.д.

Правила проведения эксперимента

Как правило, проведение эксперимента начинается с простых и быстрых для реализации пилотных тестов (например, повышение или же повышение цены). Для запуска эксперимента были определены следующие критерии:

1. **Создание гипотезы.** Первичным процессом при формировании эксперимента является создание гипотезы – это вероятностное суждение того, что может произойти в случае успеха (провала). При создании гипотезы необходимо учесть следующие вопросы [3]:
   - Можно ли измерить то, что будет подвержено тестированию?
   - Подходит ли данная гипотеза под общие стратегии и цели?
   - Будет ли существенным вклад результатов тестирование данной гипотезы?
   - Как выглядит «успех» или же «провал» в данном контексте?
   - Поиск простых решений. Для данного случая необходимо произвести оценку, можно ли выполнить эксперимент с использованием текущих ресурсов. Когда какой-либо банк планирует внедрить новую технологию для клиента, это не приводит к переобучению персонала – чаще всего, внимание будет сосредоточено на действиях, которые могут быть автоматизированы с помощью информационных систем.
2. **Анализ выходных данных эксперимента.** Когда определена контрольная группа, можно провести несколько аналитических

экспериментов. Например, если в целевой группе присутствуют взрослые и дети, то можно провести эксперимент с каждой подгруппой по отдельности.

3. **Выход из целевой зоны**. Частой ошибкой многих компаний является то, что запущенные тесты лишь корректируют их основную идею работы. Хорошим примером в данном случае является компания Tesco (сеть супермаркетов Великобритании) – проведя эксперимент, они обнаружили, что будет выгодно отправить купоны на продажу органической продукции своим покупателям, что привело к повышению прибыли и, в свою очередь, является «внекоробочным мышлением».
4. **Измерение целевых значений.** Просмотр результатов пилотного тестирования в контрольной группе важен, т.к. результаты обратной связи в данном случае будут играть важную роль в дальнейшем проведении «пилотного» теста.
5. **Использование реальных условий.** Норвежский экономист Трюгве Хаавельмо, который выиграл в 1989 году Нобелевскую премию, отметил, что существует два вида экспериментов: «те, которые компании хотели бы сделать» и «поток экспериментов, где компании выступает сторонними-пассивными наблюдателями». Таким образом, цель состоит в том, что бы не «гнаться» за экспериментами с идеальными условиями, а принимать наилучшие решения, которые производятся в данный момент.
6. **Преодоление неблагоприятных факторов.** В данном случае, данные препятствия можно охарактеризовать двумя факторами: внешние и внутренние. Внутренних факторов-препятствии, как правило, намного больше, чем внешних. Экспериментирование требует более взвешенных принятий решений и готовности попробовать множество подходов, даже если некоторые из них не увенчаются успехом
7. **Анализ результатов.** После того как эксперимент выполнен, требуется значительное время для оценки результатов. Необходимо сравнить фактические результаты с предполагаемыми результатами созданной гипотезы.
   - Что должно было произойти и что произошло?
   - Был ли эксперимент успешным (неуспешным)?

Так же существует практическое ограничение на количество проводимых экспериментов. В данном случае, важную роль будут играть аналитики, которые и будут определять количество и тип проводимых тестов.

Хотелось бы отметить, что некоторые компании специально создают ПО для проведения экспериментов и тестов (Capital One, eBay). В качестве налога можно привести в пример такой инструмент как SAS [2]. Использование данного решения позволяет отслеживать поведение контрольных групп, хранить выводы экспериментов для просмотра в будущем.

Процесс тестирования (экспериментирования)

Основная идея тестирования («Put your Ideas to the Test») заключается в том, чтобы условия эксперимента были одинаковыми для всех. Процесс всегда начинается создания проверяемой гипотезы (должна обеспечиваться возможность пройти позитивный и негативный тест). После этого выбирается контрольная группа, ситуация, временной интервал для тестирования. После того, как эксперимент был проведен в течении указанного периода времени, данные анализируются для определения результатов и соответствующих действий. Результаты могут быть помещены «обучающие библиотеки» (learning library), которые в дальнейшем могут привести к проведению комбинированного тестирования или к пересмотру уже существующей гипотезы.

Заключение

Будь эксперименты комбинированными или простыми, естественными или же искусственно созданы, преследуется следующая цель: перейти от интуитивного принятия решений к принятию решений с использованием экспериментов. В свою очередь, интуиция будет играть важную роль в области инноваций, но с тем учетом, что она будет «подкреплена» пилотными тестами. При применении данного подхода открываются возможности для делегирования полномочий для выполнения простых пилотных тестов для низших уровней управления.

Бизнес-эксперименты помогут проверить внедряемые идеи и собрать дополнительную информацию, прежде чем тратить значительную часть ресурсов для крупного проекта. Они могут быть представлены как самыми простыми, так и сложными испытаниями на проекте.

Список литературы

1. Eric T. Anderson. Harvard Business Review, March 2011.
2. «Capital One Financial Corporation» HBS case no. 9-700-124.
3. Bisuness Experiments - https://www.mindtools.com/pages/article/business-experiments.htm
   http://www.holodilshchik.ru/index_holodilshchik_issue_6_2006_AHU.htm

УДК 621.372

ОПИСАНИЕ ПРОЦЕССА РАЗРАБОТКИ В ВИРУТАЛЬНОЙ СРЕДЕ OPENSIM[45]

А.С. Степанов[46]

В этой статье описывается платформа для работы в виртуальном мире – OpenSim. Так же описывается процесс работы в ней, на примере разработки рабочего места слесаря – сборщика.

Введение

Виртуальные миры набирают все большую популярность не только в области развлечений, но и в области обучения персонала, бизнеса и производства. Виртуальный мир на основе OpenSimulator разрабатывается для обучения персонала на АО «УМЗ». Такие меры позволяют снизить риск выхода дорогостоящего оборудование из строя из-за неправильных действий человека во время обучения техническим процессам. Обучение на виртуальных тренажёрах очень выгодно, а потому это крайне перспективная среда для работы. Тем более с появлением шлемов виртуальной реальности, которые позволят делать процесс обучения ещё более наглядным.

1. Краткое описание платформы OpenSim.

OpenSimulator это многопользовательский сервер, трёхмерной среды динамичных виртуальных миров, с открытым исходным кодом, прекрасно работающий на большинстве современных операционных систем. Используется для создания совместной виртуальной среды (или вселенной), взаимодействующей с большим семейством программ-клиентов, с возможностью выбора способа доступа к контенту (через выбор протоколов). OpenSimulator не ограничивает выбор технологий, позволяя создавать собственное решение, на технологиях, которые вы считаете наилучшими для своих целей; разработанные модули расширения позволяют создавать индивидуальные конфигурации сервера

[45] Исследование выполнено при финансовой поддержке РФФИ в рамках научного проекта № 15-07-08268 а.

[46] *432027, Ульяновск, ул. Северный Венец, 32, УлГТУ, e-mail:stepanovas@ido.ulstu.ru*

с расширенными возможностями. OpenSimulator написан на C#, работает как на Windows, используя .NET Framework, так и на Unix подобных машинах, используя Mono. OpenSimulator выпущен под лицензией BSD, которая делает его использование одинаково полезным как в продуктах с открытым исходным кодом, так и в закрытых решениях, в коммерческих целях.

Прямо из коробки OpenSimulator предоставляет возможности виртуальных миров аналогичных Second Life™, особенно учитывая то, что его работа основана на работе с одноименным ядром протокола обмена сообщениями SL. Этот факт дает неограниченные возможности работы со средой симулятора, используя альтернативные Second Life™ клиенты. Важно отметить, что OpenSimulator не должен рассматриваться как попытка скопировать функциональность сервера Second Life™, скорее, как аналог сосуществования современных ОС. Далеко не все возможности Second Life™ напрямую поддерживаются OpenSimulator, связано это с использованием лимитирующих лицензионных компонентов, применяемых в Second Life™ на их серверах.

Особенности:

- Поддержка множества виртуальных пространств на одном экземпляре симулятора, в т.ч. сплошные пространства в виде мега-регионов;
- Поддержка трехмерных пространств с переменным размером, в пределах одиночных экземпляров;
- Поддержка разных семейств программ для работы в виртуальном мире(клиентов) и возможность выбора методик доступа к виртуальному контенту (выбирая протокол взаимодействия со средой);
- Огромные возможности моделирования внешнего вида вашего аватара, одежды, поверхности тела, носимых аксессуаров;
- Моделирование физических процессов (отображаемое на физические предметы), в реальном времени, используя различные физические движки, включая ODE;
- Возможности создания разнообразного контента в среде симулятора, в реальном времени, используя встроенные инструменты;
- Возможность автоматизации процессов внутри среды симулятора, используя различные языки программирования, в том числе - LSL/OSSL, C#, JScript и VB.NET.
- Неограниченные возможности по настройке виртуального окружения, подключая дополнительные плагины к сцене симулятора

2. Описание процесса разработки.

В начале работы в вашем мире нет ничего кроме острова, на котором находится ваш аватар. Для создания объектов можно использовать стандартные примитивы, либо загружать 3D-модели из вне. Загружать можно только модели .DAE формата. После этого можно начинать писать скрипты для ваших моделей. Для этого нужно в нижнем меню нажать кнопку “Build”, выбрать нужный вам объект, в появившемся окне выбрать вкладку “content” и нажать кнопку “New Script”.

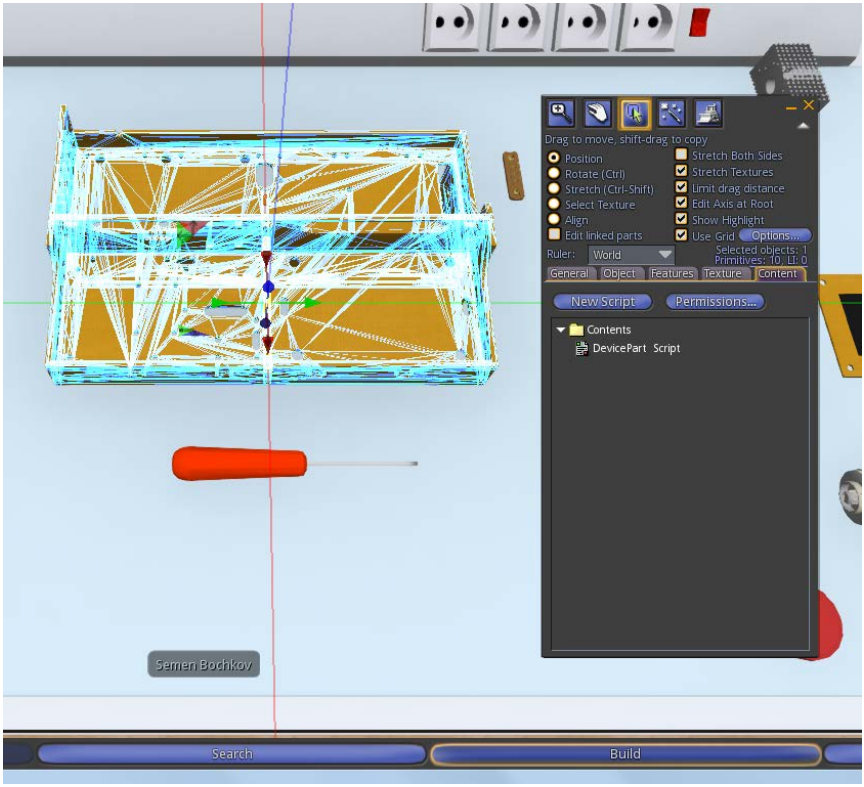


Рис.1. Создание скрипта

Появится новый скрипт, в котором ещё ничего нет. Кликнув по нему 2 раза откроется редактор. Код обычно пишется на языке LSL, можно использовать C# если в верхней строчке редактора написать “//c#”. Это сильно помогло при решении задачи движения объекта за курсором, так

как в слесаре-сборщике таких объектов весьма много. Эту проблему мы решили, использовав в скрипте C# библиотеку System.Drawing.

В процессе работы мы столкнулись с проблемой отсутствия глобальных переменных, видимых для каждого объекта мира. Для передачи данных от одного объекта к другому используются каналы. Это как чат между объектами, т.е. для вывода сообщения в общий чат нужно отправить сообщение на нулевой канал, и оно будет видно всем, для передачи же сообщений между объектами нужно выбрать номер канала, по которому будет передавать сообщение первый объект, а в скрипте второго объекта написать команду для прослушки этого канала. После этого нужно написать процедуру обработки пришедших сообщений.

Заключение

Конечно же здесь описаны далеко не все нюансы с работы с OpenSim, например, не были затронуты возможность отправки на сервер POST-запросов, или трансляция с сервера звуковых файлов, но тут были рассмотрены самые острые проблемы, с которыми мы столкнулись при разработке.

Список литературы

1. LSL Portal - Second Life Wiki [Электронный ресурс] – Режим доступа: http://wiki.secondlife.com/wiki/LSL_Portal. OpenSim [Электронный ресурс] – Режим доступа: www.opensimulator.org.
2. Виртуальные миры: практическое применение / Блог компании Гарс Телеком / Хабрахабр [Электронный ресурс] – Режим доступа: https://habrahabr.ru/company/garstelecom/blog/140081/.

УДК 621.372

ОБЗОР ГОТОВЫХ РЕАЛИЗАЦИЙ КОМПЬЮТЕРНЫХ ТРЕНАЖЕРНЫХ КОМПЛЕКСОВ

А.Н. Стратонов[47]

Данная статья предоставляет подробный обзор двух реализаций компьютерных тренажерных комплексов, успешно выступающих на данном рынке. В обзоре приведен, подробный обзор каждого из них, а также выявлены преимущества и недостатки каждого из рассмотренных тренажерных комплексов.

Введение

На рынке виртуальных тренажеров существует несколько решений, которые позволяют готовить специалистов для рабы на предприятии. Большинство этих систем, были реализованы на базе различных предприятий оборонного и нефтегазовго отрослей. В данном разделе представлен обзор наиболее распространенных из них. Обзору подвергаются те системы, некоторые идеи которые применяются в процессе решения задач.

Компьютерные тренажеры для обучения персонала по ПЛАС для химических и нефтеперерабатывающих производств подготовленные НПФ «КРУГ»

НПФ «КРУГ» разработан и внедрен ряд компьютерных тренажерных комплексов (КТК) для обучения персонала ликвидации аварийных ситуаций в нефтеперерабатывающей промышленности. Назначение обучающей системы повышение квалификации и обученности оперативного персонала технологических установок по ликвидации аварийных ситуаций. Обучающие системы создаются в качестве тренажеров, приближенных к действующим на установках АСУ ТП [1]. Пример эмитации аварийной ситуации для блока электрообессоливания представлен на рисунке 1.

[47] *432027, Ульяновск, ул. Северный Венец, 32,УлГТУ, e-mail:an.stratonov@gmail.com*

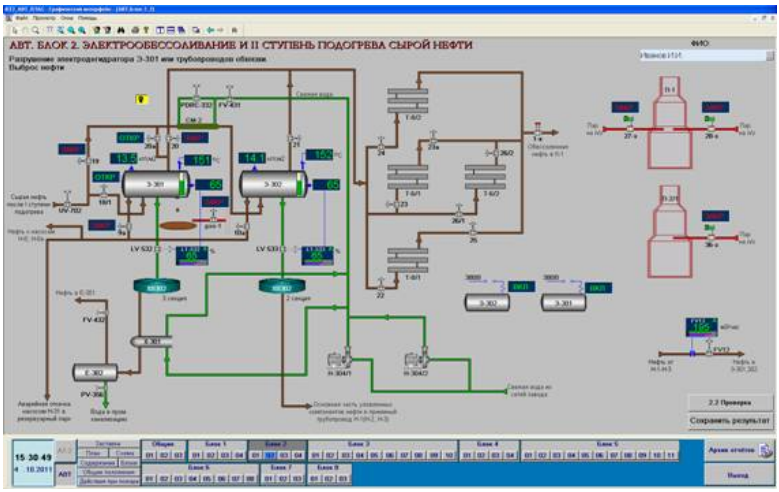


Рис. 1. Повреждение электродегидратора Э-301 и трубопроводов обвязки. Выброс нефти.

Обучающая система преследует следующие цели:

1. Обучение и приобретение практических навыков выполнения операций по предупреждению, локализации и ликвидации аварийных ситуаций
2. Непрерывный и периодический контроль и тестирование уровня знаний и навыков по ликвидации аварийных ситуаций
3. Повышение качества подготовки рабочих и ИТР, занятых ведением технологического процесса и эксплуатацией оборудования
4. Снижение вероятности возникновения аварийной ситуации по причине проявления человеческого фактора.

Данная система представляет собой программно-вычислительный комплекс, выполненный на базе персональных компьютеров. Она включает в себя перечень аварийных ситуаций в соответствии с документом «План ликвидации аварийных ситуаций» для технологической установки и моделирование действий по локализации и ликвидации конкретной аварийной ситуации.

Обучающая система имеет следующую структуру:

- Содержание-перечень аварийных ситуаций;
- Общие положения по ликвидации аварийных ситуаций;
- План расположения оборудования установки;
- Структура технологических блоков и их взаимосвязь;

- Видеокадры аварийных ситуаций;
- Инструкция по эксплуатации системы обучения.

Каждой аварийной ситуации соответствует определенный видеокадр. Видеокадр представляет собой часть технологической схемы с функциональной схемой автоматизации объекта. На видеокадрах аварийной ситуации реализованы функции управления технологическим объектом: открытие и закрытие ручной арматуры и отсечных клапанов, включение/отключение насосов, АВО, электродегидраторов, управление регулирующими клапанами. Действия обучающегося по ликвидации аварийной ситуации отображаются на мнемосхеме.

В компьютерном тренажере имитируются реальные средства управления технологическим процессом, что обеспечивает идентичность с промышленным автоматизированным рабочим местом (АРМ) оператора [2]. Конкретные средства управления оборудованием вызываются на видеокадр. Все управляющие действия фиксируются.

В тренажере предусмотрена функция «Помощь», которая позволяет оператору в ходе обучения сравнивать свои действия по ликвидации аварийной ситуации с правильными действиями. Обучающий тренажер может устанавливаться как непосредственно в операторных, так и в классах обучения. Установка тренажера в операторных позволяет эксплуатационному персоналу проводить обучение и закрепление знаний непосредственно на рабочем месте с последующей аттестацией его знаний в классах обучения. Обучающая система может быть реализована в виде отдельных рабочих мест, работающих независимо друг от друга, или объединена в общую локальную сеть по принципу преподаватель-обучающийся. Пользователи обучающей системы при необходимости могут самостоятельно вносить в нее изменения, например, при внесении изменений в ПЛАС [3].

Конструктор компьютерных тренажеров ООО "МТП"

Входящие в программный комплекс тренажеры могут быть использованы в теплоэнергетике, топливодобывающей и перерабатывающей промышленности, химических и нефтехимических предприятий, предприятий пищевой промышленности, систем теплоснабжения, газоснабжения, электроснабжения, водоснабжения, канализации и отопления.

Программный комплекс позволяет имитировать реальные процессы, происходящие в технологическом объекте, в том числе процесс управления этим объектом, а также позволяет контролировать процесс обучения [4]. Работающий в реальном масштабе времени комплекс математических моделей, технологических процессов, процессов ручного

и автоматического управления оборудованием решает с высокой точностью статическую и динамическую задачи для разнообразных режимов технологического оборудования, которое применяется в различных отраслях. В основу моделей положены уравнения баланса массы, энергии (тепла), количества движения, а также уравнения термодинамики для пара и воды и газов, уравнения тепло и массообмена.

При построении программного комплекса закладывались принципы интегральной оценки ступенчатых и плавных возмущений параметров теплового гидродинамического трактов, что полностью обеспечивает расчет в динамике на любом участке магистрали таких характеристик как давлений, расходов, энтальпий, температур, плотностей в средах, весовых и объемных соотношений в условиях движения многофазных сред. Каждая из моделей представляет собой отдельную задачу в мультизадачной программной системе тренажера; в каждой задаче имеется исполняемый программный модуль и специальным образом организованный набор данных (база данных). Совместное функционирование моделей обеспечивается сервером, который осуществляет обмен информацией между моделями, таким образом, необходимые значения переменных, рассчитанные в одной модели, передаются в другую в качестве граничных условий. Тренажер позволяет выполнить:

1. профессиональный отбор претендентов на право занятия оперативной деятельностью;
2. специальную подготовку эксплуатационного персонала различных предприятий;
3. квалифицированную аттестацию и переаттестацию персонала с выдачей соответствующего документа;
4. Программный комплекс основан на принципе модульного построения, что позволяет проводить поэтапную его модернизацию.

В настоящее время программный комплекс используется в качестве компьютерных тренажеров для подготовки оперативного персонала ТЭЦ, химических предприятий, учебных заведений (рис.2).

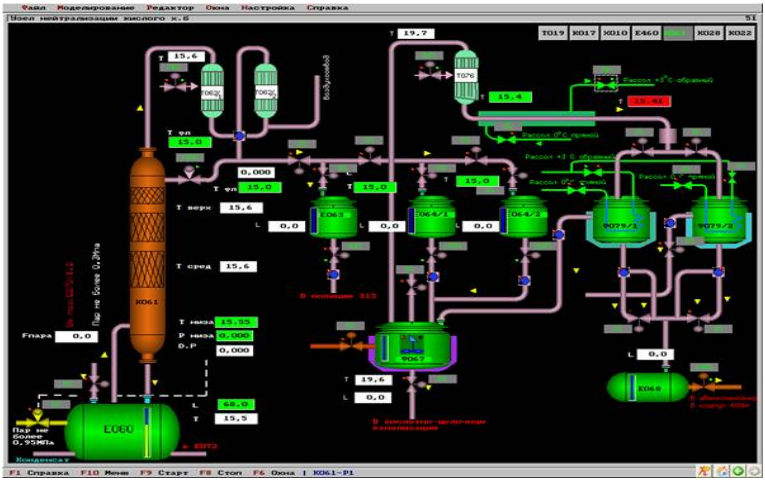


Рис 2. Пример интерфейса тренажерного комплекса производства хлорбензола (ректификация) на ОАО «Химпром»

Заключение

Каждая из рассмотренных компьютерных тренажеров систем, обладает как минимум тремя недостатками из приведенного списка: недостаточный универсализм, нагроможденность визуально-графического интерфейса, отсутствием возможностей по обучению. Приведенные причины легли в основу того, что современные конструкторы тренажеров наследуют модульную структуру и удобный визуально-графический интерфейс, что в значительной степени ускоряет разработку тренажеров.

Список литературы

1. Сайт компании НПФ «КРУГ». [Электронный ресурс]. — Электрон, дан.– Режим доступа: http://www.krug2000.ru/publications/512/1277.html свободный — Загл. с экрана. — Рус. яз.
2. Разработка ПЛА, ПЛАС [Электронный ресурс]. — Электрон, дан.– Режим доступа: http://www.ot-pb.ru/services/examination/43-razrabotka-pla-plas.html свободный — Загл. с экрана. — Рус. яз.
3. Компьютерный тренажерный комплекс автоматизированной холодильной установки, разработанный компанией «Транзас» [Электронный ресурс]. — Электрон, дан. Режим доступа: http://www.holodilshchik.ru/index_holodilshchik_issue_6_2006_AHU.htm — Загл. с экрана. — Рус. яз.
4. Программный комплекс ООО «МТС» [Электронный ресурс]. — Электрон, дан. Режим доступа: http://marinavanteeva.narod.ru/Progcomplex.htmlЗагл. с экрана. — Рус. яз.

УДК 621.372

ОБЗОР ДИАГРАММ ИШИКАВЫ. ШЕСТЬ ПРОСТЫХ ШАГОВ ИХ ПОСТРОЕНИЯ.

А.Н. Стратонов[48]

Данная статья посвящена обзору инструмента анализа причинно-следственных связей диаграмме Исикавы. Статья содержит краткую историю создания, способы применения, достоинства и недостатки, а также шаги для построения данной диаграммы.

Введение

Диаграмма Ишикавы (cause-effect diagram, fishbone diagram) – графический инструмент, позволяющий наглядно и систематизировано анализировать взаимосвязи следствий (effects) и причин (causes), которые порождают эти следствия или влияют на них. Еще эти диаграммы называют «диаграммами рыбного скелета» (fishbone diagram) за их внешнее сходство со скелетом рыб. Но какое бы имя не использовалось, необходимо помнить, что ценность этого метода состоит в способствовании категоризации и структуризации множества потенциальных причин, а так же, идентификации наиболее вероятной корневой причины изучаемого следствия. Метод применим при выполнении анализа как одним специалистом, так и группой специалистов.

Диаграмма Исикавы и ее автор

Этот инструмент анализа причинно-следственных связей между различными факторами и результатом разработал Каору Исикава, профессор Токийского университета, в 1952 (по другим источникам в 1943) году. Исикава известен в первую очередь как один из основателей японской школы менеджмента и японской системы качества.

Например, он утверждал, что в том случае, когда изменение процесса приводит к стабильно позитивному результату, этот новый процесс необходимо принять как стандартный и не отступать от него до тех пор, пока не пройдет последующий цикл улучшения этого процесса. При его

[48] *432027, Ульяновск, ул. Северный Венец, 32,УлГТУ, e-mail:an.stratonov@gmail.com*

наступлении, нужно проверить на практике новую версию процесса, снова утвердить его как стандарт и соблюдать его.
Также Каору Исикава разработал и ввел в практику на производстве концепцию кружков качества, а также разработал графическую диаграмму причинно-следственных связей. По имени разработчика этот метод назвали «диаграмма Исикавы», а по характерному внешнему виду «рыбий скелет» или «рыбья кость».

Диаграмма Исикавы - возможности

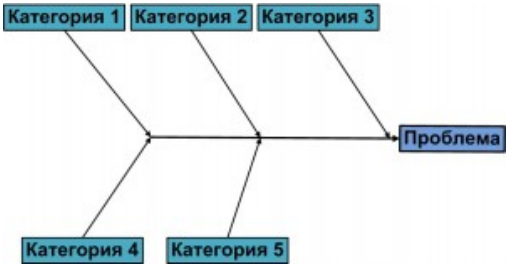
Диаграмма Исикавы позволяет прояснить и учесть все существенные факторы, влияющие на результат какой-либо деятельности. Таким результатом может быть изделие, услуга, завершенная работа, состояние оборудования — любой объект изучения или разработки. Применение диаграммы Исикавы позволяет выяснить причины каких-либо проблем в организации или, например, причины возникновения дефектов в изделиях.
Диаграмма Исикавы имеет ряд достоинств:

- помогает наглядно показать связи между полученным результатом и вызвавшими его причинами;
- позволяет провести анализ цепочки факторов, воздействующих на проблему.

Это сделало диаграмму довольно популярным инструментом анализа, особенно в системе управления качеством.

Как составить диаграмму Исикавы

Первый шаг. Необходимо максимально четко определить предмет изучения или проблему, лаконично



Метод построения диаграммы Исикавы — проблема и категории

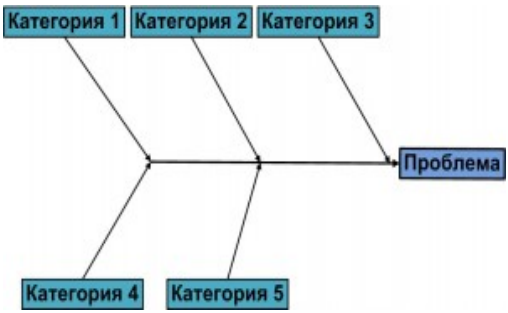
описать её и записать формулировку на рабочее поле в прямоугольнике с правой стороны. От прямоугольника влево нужно провести прямую горизонтальную линию.

Второй шаг. Над горизонтальной линией и под ней указываются группы факторов, влияющих на результат. Обычно в список групп входят так называемые «люди», «методы», «механизмы», «материалы», «контроль» и «среда»:

1. «люди» — группа причин, обусловленных состоянием и возможностями человека, то есть влиянием человеческого фактора. Например — уровень опыта, физические кондиции, способности, мотивация, состояние здоровья и прочие
2. «методы» — группа причин, обусловленных тем, каким образом делается работа, также сюда включают всё, что относится к точности или производительности отдельных операций процесса или его стадий
3. «механизмы» — группа причин, связанных с используемым оборудованием, оснасткой, которые использованы в процессе. Например, наличие и состояние рабочего и измерительного инструмента, применение каких-то дополнительных устройств
4. «материалы» включает в себя факторы, определяющие измеримые параметры и качества материала, участвующего в процессе. Например, температура, состав, упругость, влажность, вязкость или твердость
5. «контроль» включает в себя обстоятельства, влияющие на надежное и своевременное обнаружение ошибочных действий
6. «среда» — группа, включающая все значимые условия, обусловленные внешней средой и влияющие на процесс и результат действий. Например, давление, температура воздуха, воды, степень освещенности.

Необязательно использовать в рамках одного исследования все категории сразу, но нежелательно ограничиваться одной или двумя. Нужно исходить из соображений разумной достаточности.

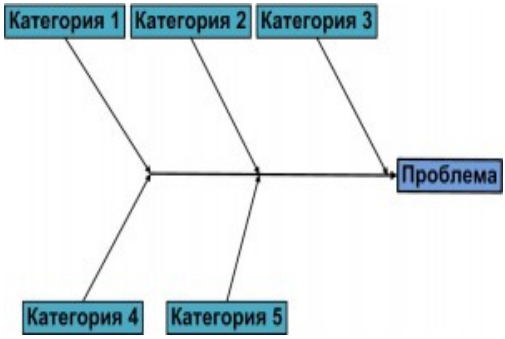
Третий шаг. Выбранные категории вписываются в прямоугольники, от них проводятся наклонные линии к уже имеющейся горизонтали.



Распределение причин по группам-категориям

Четвертый шаг. Причины изучаемой проблемы или действующие факторы распределяются по установленным категориям («люди», «механизмы» и т. д.) и указываются на диаграмме в виде линий, примыкающих к линиям категорий.

Пятый шаг. Необходимо разложить каждую из причин на составляющие. Для этого по поводу каждой причины нужно задать себе вопрос – «В связи с чем это случилось?»

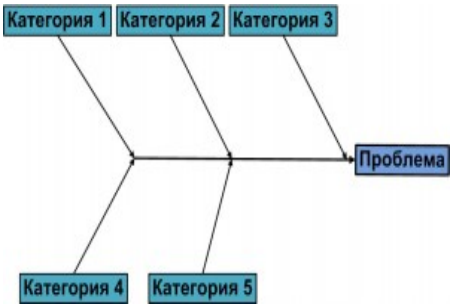


Второстепенные причины в диаграмме Исикавы

Ответы нужно фиксировать в виде ветвей более низкого уровня. Детализация причин должна продолжаться до тех пор, пока не будет найдена «корневая» причина.

Примечание. В некоторых источниках рекомендуется выделять факторы, позитивно влияющие на результат и негативные факторы. С этой целью линии, обозначающие их связь с категорией, проводят в разных направлениях, например, «позитив» налево либо изнутри от осевой линии, «негатив» — направо либо снаружи к оси — как на рисунке. Это

непринципиально, можно делать так как удобно, выделять факторы цветом линий или не выделять их вообще.



Изображение позитивных и негативных факторов в диаграмме Исикавы

Шестой шаг. Из числа первопричин выделяются наиболее значимые факторы, которые влияют на изучаемую проблему.

Несложный пример диаграммы Исикавы приведен на рисунке.

Недостатки диаграммы Исикавы

Из недостатков диаграммы можно назвать следующие:

- громоздкость графического представления (в конечном итоге для изображения всех «ветвей» может не хватить рабочей поверхности), правда это компенсируется наличием программного обеспечения, в том числе входящего в офисные пакеты;
- сложность применения и вероятность путаницы для тех случаев, когда проблема является комплексной;
- относительная сложность применения метода на практике. Есть методы, которые проще в воспринимать и применять на практике.

Заключение

Современные программные средства, обеспечивающие поддержку системы менеджмента качества, позволяют построить диаграмму Исикавы. Результаты экспертных оценок заносятся в систему, после чего автоматически производится ранжирование причин. Аналитик может наглядно увидеть те причины, которым следует уделить особое внимание и разработать корректирующие и предупреждающие мероприятия для их устранения или снижения влияния на конечный результат. Использование программных средств позволяет ускорить процесс анализа за счет отказа

от использования множества бумажных документов, а также накапливать базу знаний о проблемах и путях их решений.

Список литературы

1. Басовский Л.Е., Протасьев В.Б., Управление качеством: Учебник. М.: ИНФРА – М, 2002.

2. Виханский О.С. Стратегическое управление. М.: Гардарика, 1998.

3. Управление персоналом в условиях рыночной экономики/Под науч. ред. проф.,д-ра Штита. М.: Изд-во МГУ, 2003.

4. Шубенкова Е.В. Тотальное управление качеством: Учебное пособие/Под ред. Профессора Ю.Г. Одегова. – М.: Издательство «Экзамен», 2005.

УДК 621.372

МОДЕЛИРОВАНИЕ ПРОГНОЗА ПРОЕКТНЫХ МЕТРИК ПУТЕМ ИХ ИНТЕЛЛЕКТУАЛЬНОГО АНАЛИЗА

И.А. Тимина[49]

В статье предложена модель прогнозирования метрик проектной деятельности полученных посредством выделения нечетких тенденций данных проекта. Итогом исследования являются модели, алгоритмы и программы, решающие задачу организации автоматизированного метода прогнозирования проектных метрик.

Введение

Проблема постоянного обновления состояния всей проектной деятельности предприятия связана с высокой скоростью обновления больших массивов информации о текущих проектах. Для успешного ведения проектных работ руководителю необходимо отслеживать динамику совокупности проектов, идентифицировать проблемы и своевременно принимать корректирующие регулирующие воздействия, в том числе перераспределять ресурсы между проектами [7]. Решение данной задачи является создание способа анализа метрик проекта для получения прогноза значений, автоматизирующее процессы кластеризации по сходству всех имеющихся событий проектов предприятия.

1. Интеллектуальный анализ метрик проекта

Для прогнозирования числовых значений и нечетких тенденций состояния проекта организации используем метод моделирования нечетких элементарных тенденций [3], алгоритм которого заключается в следующем [1] [2]:

1) подсчет степени сходства поведения исследуемых временных рядов (ВР);
2) составление уравнения регрессии ВР1 по ВР2. Вычисление корреляции;
3) проверка гипотез [5].

[49] *432027, Ульяновск, ул. Северный Венец, 32, УлГТУ, e-mail: i.timina@ulstu.ru*

В качестве решения задачи прогнозирования было предложено три гипотезы прогноза:

1. **Гипотеза сохранения/смены тенденции.**

Рассматриваем тенденцию за предыдущий период. Прогнозируемое значение вычисляется по формуле:

$$\tau_{t+1} = \tau_t + \tau_p,$$

где τ_{t+1} – прогноз на следующий период времени;

τ_t – реальное значение в момент времени *t*;

τ_p – значение тенденции за предыдущий период времени.

2. **Гипотеза устойчивости/неустойчивости тенденции.**

Используем для прогнозирования скользящее среднее

$$\tau_{t+1} = \tau_t + G\tau_p,$$

где $G\tau_p$ – значение доминирующей нечеткой тенденции.

Рассматриваем тенденции за предыдущий выбранный период, выбираем преобладающий кластер тенденции и вычисляем прогноз по приведённой выше формуле, строим тенденцию. При одинаковых количествах вхождений тенденций используется оптимистичный прогноз, то есть выбирается наибольшее среднее значение тенденций.

Для нечеткой кластеризации тенденций используем FCM-алгоритм, основанный на минимизации целевой функции:

$$J = \sum_{j=1}^{C} \sum_{j=i}^{N} \mu_{ij}^{m} \|x_i - c_j\|,$$

где *N* – количество показателей;

C – количество кластеров;

μ_{ij} – степень принадлежности объекта *i* кластеру *j*;

m – любое действительное число, большее 1,

x_i – *i*-й объект набора объектов;

c_j – *j*-й кластер набора кластеров;

$|x_i - c_j|$ – норма, характеризующая расстояние от центра кластера *j* до объекта *i*.

Объектами кластеризации являются множество элементов ВР $A = \{a_1, \dots, a_n\}$. Задачей *FCM*-алгоритма является разбиение этого набора на заданное количество кластеров. Вводится окончательное множество свойств (тенденций) $P = \{p_1, \dots, p_q\}$, каждое из которых количественно показывает определенный признак частей исследуемой предметной области. Где под *n* подразумевается общее количество объектов данных, а *q* – общее количество измеримых свойств.

Далее для каждого из объектов кластеризации должны быть измерены все признаки множества *P* в некоторой количественной шкале. Тем

самым, каждому из элементов $a_i \in A$ ставится в соответствие некоторый вектор $x_i = \left(x_1^i, x_2^i, \dots, x_q^i\right)$, где x_j^i – количественное значение признака $p_j \in P$ для объекта $a_i \in A$. Все x_j^i принимают действительные значения. Эти количественные значения признаков тенденций и являются векторами признаков объектов кластеризации. Векторы значений признаков $x_i = \left(x_1^i, x_2^i, \dots, x_q^i\right)$ удобно представлять в виде матрицы данных *D* размерности $n \times q$, каждая строка которой равна значению вектора x_i.

3. Прогнозирование на заданный период на основе нечетких элементарных тенденций.

Алгоритм прогнозирования на период на основе тенденций работает следующим образом. Эксперт задает количество рассматриваемых тенденций за предыдущий период, например, за полгода – набор тенденций ***A***. Либо задает шаблон набора тенденций, зная предположительную тенденцию следующую за этим набором.

$$\left\{\tau_{t_{n-m}}, \dots, \tau_{t_{n-1}}, \tau_{t_n}\right\}$$

Далее идет поиск набора ***A*** тенденции во всех остальных предыдущих периодах.

$$\left\{\tau'_{t_{n-l-k}}, \dots, \tau'_{t_{n-l-(k-1)}}, \tau'_{t_{n-l}}\right\}$$

Найдя такой набор ***B***, учитываем какая тенденция ***C*** стоит после этого найденного набора ***B***. Строим прогноз равный тенденции ***C***.

$$\tau_{t+1} = \tau_t + \tau'_{t_{n-l+1}}$$

Если не существует такого набора ***B***, который совпадал бы с набором ***A***, поиск набора повторяется, но уже ищется не полное его совпадение, а с отклонением в одну тенденцию. Так повторяем до тех пор, пока не будет найден подходящий набор тенденций ***B***.

Выбор наилучшей гипотезы осуществляется с учетом наименьшей средней относительной ошибки (*MSE*).

Для реализации гипотез прогноза метрики проектных событий были представлены в виде нечетких временных рядов и временных рядов нечетких тенденций.

Каждая нечеткая тенденция представляется в виде кортежа отношения:

$$\tau = \langle \tilde{v}, \tilde{a}, \Delta t, \mu \rangle,$$

где τ – наименование нечеткой тенденции из множества $\mathfrak{I}$, $\tau \in \mathfrak{I}$;

$\tilde{v}$ – тип нечеткой тенденции (тип изменений) $\tilde{v} \in \tilde{V}$ выражает основные качественные зависимости временных рядов {*Падение, Рост, Стабильность*}.

$\tilde{a}$ – интенсивность нечеткой тенденции, $\tilde{a} \in \tilde{A}$, которое выражается лингвистически, например, значениями из множества {*Сильное, Среднее, Слабое*};

Δt – продолжительность нечеткой тенденции, $\Delta t \in \Delta T$;

μ – функция принадлежности участка нечеткого временного ряда, ограниченного интервалом Δt нечеткой тенденции τ.

Тенденция называется элементарной, если $\Delta t = 1$.

2. Алгоритм коррекции прогноза проектных метрик

Для решения задачи прогнозирования временного ряда $Y = \{t_i, x_i\}$, $(i = 1,2, \dots, n)$ с помощью нечеткого подобия, **при условии существования правдоподобности экспертной гипотезы**, что нечеткая тенденция временного ряда Z= $\{t_i, z_i\}$, $(i = 1,2, \dots, k)$ является **предиктором временного ряда** Y используем алгоритм коррекции гипотез прогноза [4], [6]. **Алгоритм 1** включает три фазы.

На первой фазе прогнозируются нечеткие элементарные тенденции временного ряда Y:

$$\tau_{t+1}^{Y} = f(\tau_t^{Y}),$$

здесь τ_{t+1}^{Y} – прогнозная нечеткая элементарная тенденция временного ряда Y;

τ_{t+1}^{Y} – текущая нечеткая элементарная тенденция временного ряда Y;

f – зависимость в нечетких элементарных тенденциях временного ряда Y.

На второй фазе прогнозная нечеткая элементарная тенденция временного ряда Y корректируется с учетом компонент основных тенденций исследуемого временного ряда $G\tau_Y$ и временного ряда предиктора $G\tau_Z$ соответственно:

$$\hat{\tau}_{t+1}^{Y} = r(\tau_{t+1}^{Y}, G\tau_Y, G\tau_Z),$$

где

τ_{t+1}^{Y} – прогнозная нечеткая элементарная тенденция временного ряда Y;

$\hat{\tau}_{t+1}^{Y}$ – прогнозная нечеткая элементарная тенденция временного ряда Y после корректировки;

$G\tau_Y$ – основная нечеткая тенденция временного ряда Y;

$G\tau_Z$ – основная нечеткая тенденция временного ряда Z;

r – правила корректировки.

Третья фаза используется для получения оценки прогнозного значения числового временного ряда Y.

3. Результаты исследований

Для исследований были взяты данные метрик событий проекта *MongoDB*, а именно главная его программная часть *Core Server*. Данные об ошибках, улучшениях и нововведенных функциях были взяты с ресурса *Jira.mongodb.org*.

MongoDB – документно-ориентированная система управления базами данных (СУБД) с открытым исходным кодом, не требующая описания схемы таблиц. Система *MongoDB* написана на языке *C++*.

В таблице 1 представлены результаты анализа.

Табл. 1. Анализ метрик событий проекта

Тип метки	Общая тенденция	Доминирующая тенденция	Мера сходства	Коррел яция	Интерпретация корреляции
Improvement	Рост	Рост Слабый	0,38	0,5419	Средняя
Task	Рост	Стабильность			
Bug	Падение	Рост Слабый	0,52	- 0,2377	Слабая
Improvement	Рост	Рост Слабый			

Табл. 2. Результаты прогнозирования метрик событий с учетом влияния ВР-предиктора

ВР зависимый	ВР-предиктор	Гипотеза 1	Гипотеза 2	Гипотеза 3
Improvement	Bug	Стабильность	Рост сильный	Стабильность
Improvement	Task	Стабильность	Рост сильный	Стабильность

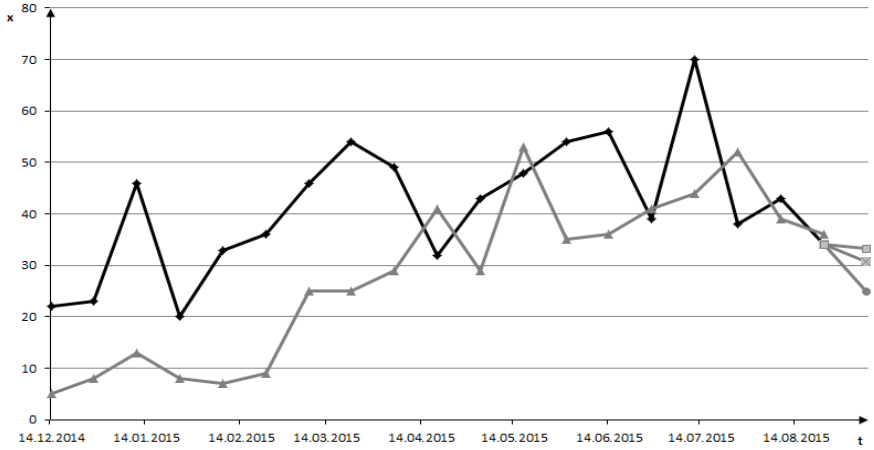


Рис. 1. Прогнозирование *Improvement* в проекте *MongoDB* с учетом влияния *Task* в проект (сохранения тенденции)

На рисунке 1 представлен вариант прогнозирования с использованием гипотезы, где ось x – значение ВР, ось t – время; « обозначает ВР Y; « » обозначает ВР Z; « » обозначает прогноз без учета корректировки; « » обозначает прогноз τ_{t+1}^{Y} после корректировки ВР

Y по основной тенденции ВР Y; « —■—» обозначает прогноз $\hat{\tau}_{t+1}^{Y}$ после корректировки ВР Y по основной тенденции ВР Z.

Полученные результаты показывают, что предложенный подход, реализующий модифицированный метод прогнозирования НЭТ может быть использован для краткосрочного прогнозирования ВР в ситуации, когда имеется экспертное предположение о существовании ВР предиктора.

Заключение

В представленной статье решена актуальная научно-техническая задача, имеющая важное значение для развития и применения в системах автоматизированного проектирования средств математического моделирования, численных методов, а именно: построение модели прогнозирования ВР метрик совокупности проектов крупной проектной организации на основе их анализа и моделирования.

Список литературы

1. Артюхов М.В., Ярушкина Н.Г., Романов А.А., Гуськов Г.Ю., Тимина И.А. Построение массива методов для прогнозирования временных рядов // VIII-ая Международная научно-практическая конференция «Интегрированные модели и мягкие вычисления в искусственном интеллекте. май 2015. Т. 1. С. 332-341

2. Артюхов М.В., Ярушкина Н.Г., Романов А.А., Гуськов Г.Ю., Тимина И.А. Прогнозирование временных рядов коллективами методов // Радиотехника. 2015. Т. 6. С. 48-54.

3. Тимина И.А. Нечеткая зависимость временных рядов в моделировании и прогнозировании состояния организации // Информатика и вычислительная техника (сборник научных трудов 5-й Всероссийской научно-технической. Май 2013. С. 228-234.

4. Тимина И.А. Прогнозирование зависимых нестационарных временных рядов на основе нечетких элементарных тенденций с учетом регулирования гипотезы // Сборник материалов 48й Научно-технической конференции «Вузовская наука в современных условиях». 2014. С. 182-185.

5. Ярушкина Н.Г., Воронина В.В., Тимина И.А., Эгов Е.Н. Прогнозирование состояния технической системы с применением меры энтропии для нечетких временных рядов // Автоматизация процессов управления. 2015. Т. 3(41). С. 49-58.

6. Ярушкина Н.Г., Тимина И.А. Алгоритм адаптации прогноза зависимых временных рядов // 2-ой Международный Поспеловский симпозиум «Гибридные и синергетические интеллектуальные системы ГИСИС’2014». 2014. С. 398-403.

7. Ярушкина Н.Г., Тимина И.А. Модель и средства управления проектированием автоматизированной системы на основе динамики метрик программного кода // Автоматизация процессов управления. 2015. Т. 3(41). С. 73-81.

УДК 621.372

ОПРЕДЕЛЕНИЕ БУДУЩИХ ПОСЛЕДСТВИЙ СОБЫТИЯ

В.А. Хомяков [50]

В данной статье представлены основные этапы определения будущих последствий события.

Введение

Если вам когда-либо было нужно изучить все последствия предлагаемого события, вы знаете, как трудно это может быть, найти все возможные последствия.

В подобных ситуациях, многие люди паникуют, и перечисляют первые последствия, которые им приходят в голову, в результате чего список получается не полный.

В этом случае может помочь «**Futures Wheel»**. Этот визуальный инструмент дает вам структурированный вид прямых и косвенных последствий решений, события и тенденций.

Об инструменте

Futures Wheel представляет собой простой, практичный инструмент, который поможет вам выявить прямые и косвенные последствия решения, события или тенденции.

Futures Wheel был создан Джером Гленн в 1972 году, с тех пор он стал признанным экспертом и спикером в этой области.

Инструмент создан, чтобы определить возможные последствия тенденций и событий, но вы также можете использовать его в процессе принятия решений и в управлении изменениями. На рисунке1 представлен пример использования данного инструмента.

50 *432027, Ульяновск, ул. Отрадная, 85, УлГТУ, vadim7394@gmail.com*

Рис. 1 - Futures Wheel

Этапы использования инструмента

Для определения последствий события Джером Гленн были определены следующие этапы использования:

1. **Определение События.** Напишите, то от чего необходимо отталкиваться центре куска бумаги или на доске. Это может быть событие, тенденция, проблемы, или возможное решение.
2. **Определение последствий первого порядка.** Определите что влияет на событие в первую очередь. Напишите каждое последствие в круге, и соединить его с центральной идеей стрелкой. Это последствие первого порядка.
3. **Определение косвенных последствий второго порядка**. Теперь вам нужно выяснить все возможные последствия влияющие на каждое последствие первого порядка, которые вы записали во втором этапе, и добавить их в диаграмму таким же образом.
   Затем повторите это путем выявления последствий третьего порядка, последствия четвертого порядка, и так далее.

4. **анализ последствий.** После того как вы выполнили все уровни Futures Wheel, вы будете иметь четкое представление о возможных прямых и косвенных последствий в результате события. Перечислите их.
5. **Определение действия.** Там, где возможны последствия, которые вы определили отрицательные, стоит подумать о том, как вы будете их минимизировать. Где последствия положительны, подумайте о том, что вы будете делать, чтобы в полной мере воспользоваться ими.

Так же, автор советует использовать разные цвета для каждого «уровня» колеса, как на рисунке 1, это позволит вам более наглядно анализировать последствия.

Заключение

Для событий, тенденций или решений, которые могут вести за собой большое количество последствий стоит использовать Futures Wheel, так как благодаря ему можно получить, полную наглядно полную информацию о событии.

Впоследствии, можно произвести анализ с использование полученных данных, и таким образом минимизировать нежелательные последствия.

Инструмент также можно использовать в процессе принятия решений и в управлении изменениями.

Список литературы

1. Futures Wheel. 5 мая, 2015. - https://en.wikipedia.org/wiki/Futures_wheel

2. Futures Research and the Strategic Planning Process – Implications for Higher Education. Chapter: The Stages of the Strategic Planning Process. By James L. Morrison, William L. Renfro, and Wayne I. Boucher. ASHE-ERIC Higher Education Research Reports. 1984. - http://horizon.unc.edu/projects/seminars/futuresresearch/stages.asp

3. The Futures Wheel - Decision-Making Skills Training From MindTools.com - https://www.mindtools.com/pages/article/futures-wheel.htm — Загл. с экрана. — Рус. яз.

УДК 658.512.26

ДЕКОМПОЗИЦИЯ ЭЛЕКТРОННОЙ МОДЕЛИ ИЗДЕЛИЯ ДО УРОВНЯ СТРУКТУРНО-ФУНКЦИОНАЛЬНЫХ ЭЛЕМЕНТОВ

Д.Э. Цыганков[51], А.Ф. Похилько[52]

Рассматривается процесс выделения структурно-функциональных элементов с фиксацией их семантической наполненности на основе системного анализа процесса формирования 3D-модели изделия.

Введение

Превращение САх-систем в основной инструмент проектировщика вместе со стремлением предприятий к повышению качества выпускаемой продукции [1], выводит на передний план проблемы сохранения и накопления проектных данных, получаемых в САПР для их модификации и повторного использования [2]. Ключевым является аспект фиксации проектных данных в виде электронных моделей, способных однозначно верно восприниматься не только самим автором проекта, но и другими проектировщиками подразделения организации [3].

Одно из решений данной проблемы – технология *Knowledge-based Engineering* (*KBE)* – проектирование на основе знаний, заключающаяся в применение систем, основанных на знаниях (*Knowledge-based Systems*) – программ, использующих базу знаний на основе причинно-следственных связей для решения задач в области проектирования [4]. Технологию KBE предлагается интегрировать с развивающейся в настоящее время в рамках объектно-ориентированного проектирования концепцией структурно-функционального элемента (СФЭ) [5].

1. Концепция структурно-функционального элемента

Структурно-функциональный элемент (англ. *Features, «фьючерс»*) – это составная часть информационного 3D-образа изделия, однозначно верно воспринимаемая в предметной области проекта. СФЭ может быть функциональным, конструктивным, технологическим, а также гибридным. В качестве наглядного примера на рисунке 1 представлена структурно-

[51] *432027, Ульяновск, ул. Северный Венец, 32, УлГТУ, e-mail: d.tsyg@mail.ru*
[52] *432027, Ульяновск, ул. Северный Венец, 32, УлГТУ, e-mail: paf_54@mail.ru*

функциональная декомпозиция на СФЭ 3D-модели гайки фиксирующей из состава соединителя коаксиального радиочастотного.

Как видно на рис. 1, каждый СФЭ имеет однозначно воспринимаемую семантическую наполненность (отверстия, резьба, канавка, фаска и др.), актуальную в предметной области изделия – соединителя коаксиального [6], а также конкретного проектируемого финального изделия (уровня «Сборочная единица»), в котором применяется рассматриваемая деталь. На основе физического смысла выделяются атрибуты СФЭ, качественно и количественно определяющие итоговый геометрический 3D-образ изделия (диаметр отверстия, количество граней под ключ и др.). Значения данных атрибутов являются характеристикой конкретного экземпляра СФЭ.

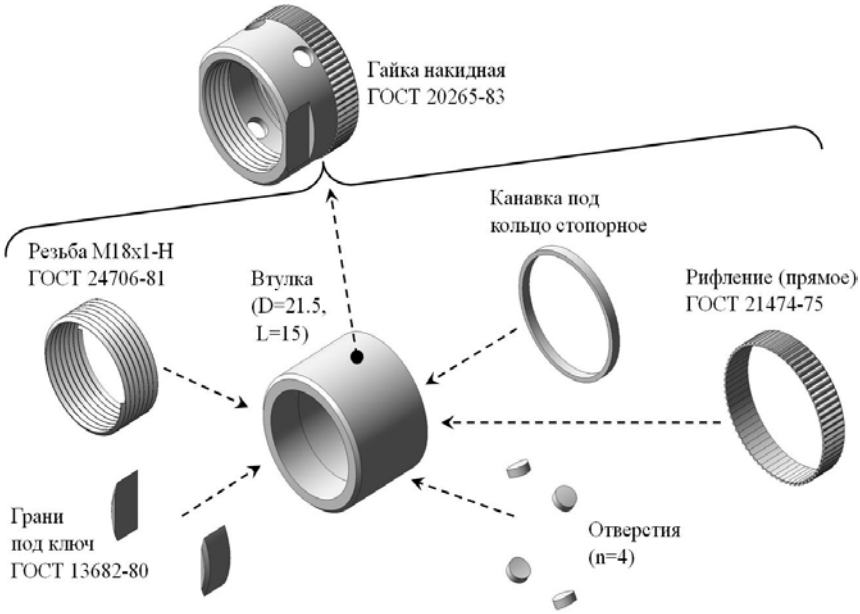


Рис. 1. Определение термина структурно-функционального элемента 3D-модели

Атрибуты базового тела-заготовки (3D-объекта СФЭ, составляющего структурную основу проектного решения) – в данном примере втулки, определяются в соответствии с ГОСТ 20265-83 и локальными проектными параметрами (внешний D и внутренний d диаметры, длина втулки L и др.). Атрибуты других СФЭ – в соответствии с параметрами и атрибутами базового тела, а также всего изделия и других компонент, сопрягаемых с гайкой накидной (кольцо стопорное, контактная цанга и пр.).

2. Процессное представление 3D-образа СФЭ

В книге [7] дано общее понятие системы как набора объектов и взаимосвязей между ними; в соответствии с чем, информационный 3D-образ изделия (уровня *«Деталь»*) Mod^{3D} можно представить как систему: набор СФЭ Fts_i, структурированный за счет их взаимосвязей $Int_{k,j}$ друг с другом:

$$Mod^{3D} = < Fts_i, Int_{i,(i+1)} >, \; i = \overline{1,n} \; . \qquad \textbf{(1)}$$

Представление 3D-модели изделия в соответствии с (1) позволяет подробно описывать ее внутреннюю структуру (что крайне удобно при рассмотрении сборочных единиц), однако не способно фиксировать в себе проектных знаний, закладываемых инженером-проектировщиком.

Для выделения логики принятия проектных решений, предлагается рассматривать каждый СФЭ процессом его формирования $Des.Proc^{Fts}$ – упорядоченной последовательности проектных процедур dp_i – операций, предоставляемых CAD-системой:

$$Des.Proc^{Fts} = < dp_i >, \; i = \overline{1,n} \; . \qquad \textbf{(2)}$$

$Des.Proc^{Fts}$ в терминах CAD-систем называется деревом построения. Взаимосвязь проектных процедур dp_i выполняется за счет того, что, результат выполнения *i*-й проектной процедуры dp_i может быть исходным для каждой из последующих за ней *(i+1 … n)* процедур. Следовательно, упорядоченность процесса $Des.Proc^{Fts}$ отображает иерархию построения СФЭ.

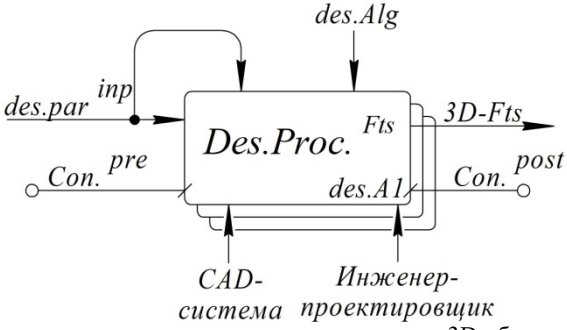


Рис. 2. Контекстная диаграмма процесса построения 3D-объекта СФЭ

Представление проектной информации и инженерных знаний в виде процесса наиболее информативно демонстрируется на базе методологии функционального моделирования IDEF0, что подробно описано в [8]. На рисунке 2 показана контекстная IDEF0-диаграмма процесса построения СФЭ $Des.Proc^{Fts}$. На данном рисунке: $des.par^{inp}$ – это исходные данные

(проектные параметры и их значения, а также геометрические данные); 3D-Fts – геометрический образ СФЭ; *des.Alg* – алгоритм проектирования (общее дерево класса проектных решений уровня СФЭ); Con^{pre}, Con^{post} – пред-, и постусловия соответственно (состояния формируемого 3D-объекта СФЭ, рассматриваемые как наборы условий для выполнения последующих проектных процедур dp_i).

Система $Des.Proc^{Fts}$ всех СФЭ входящих деталь, обобщенная объектной ориентацией, представляет собой сюръективное отображение информационного геометрического образа проектируемого изделия на основе семантической наполненности:

$$Sem.^{des} : Des.Proc^{Fts} \rightarrow Mod^{3D}. \quad \textbf{(3)}$$

Заключение

Представление совокупности СФЭ, описывающих 3D-модель изделия, процессом, в виде упорядоченной последовательностью проектных процедур $Des.Proc^{Fts}$ позволяет фиксировать логику их формирования, а, следовательно, и закладываемую проектную информацию с учетом знаний инженера-проектировщика и его опыта.

Список литературы

1. *Похилько А.Ф.* Технология представления проектной деятельности в интегрированной среде САПР // Вестник Ульяновского государственного технического университета, 2000. № 3. С. 32-37.
2. *Цыганков Д.Э.* Представление проектируемого изделия системой структурно-функциональных элементов / *Д.Э. Цыганков, А.Ф. Похилько* // Современные проблемы проектирования, производства и эксплуатации радиотехнических систем: Сборник научных трудов. – Ульяновск: УлГТУ, 2016. С. 207-210.
3. *Цыганков Д.Э.* Представление процесса проектирования на базе обобщения элементарных операций до уровня семантических единиц / *Д.Э. Цыганков, А.Ф Похилько* // Автоматизация процессов управления. 2015. № 3. С. 81-88.
4. *C.B. Chapman, M. Pinfold*, The Application of a Knowledge based Engineering Approach to the Rapid Design and Analysis of an Automotive Structure. Advances in Engineering Software, 2001. V. 32, № 12, pp. 903-912.
5. *D. Tsygankov, et al.*, The Design Process Structural & Logical Representation in the Concurrent Engineering Infocommunication Environment, *R. Curran et al.* Transdisciplinary Lifecycle Analysis of Systems, IOS Press, 2015, pp. 595-602.
6. ГОСТ 20265-83. Соединители радиочастотные коаксиальные. Присоединительные размеры. – М.: Изд-во стандартов, 1984. – 14 с.
7. *Евгенев Г.В.* Интеллектуальные системы проектирования: учеб. пособие. – М.: Изд-во МГТУ им. Н. Э. Баумана, 2009. 334 с.
8. *Похилько А.Ф.* CASE-технология моделирования процессов с использованием средств BPWin и ERWin: Учебное пособие / *А.Ф. Похилько, И.В. Горбачев*. Ульяновск: УлГТУ, 2008. 120 с.

УДК-004.853

АВТОМАТИЗАЦИЯ ОЦЕНКИ КАЧЕСТВА РЕАЛИЗАЦИИ ТАБЛИЦ КАСКАДНЫХ СТИЛЕЙ (CSS-MARK)

Г.И. Шайдуллова

В статье предлагается способ по автоматизации оценки качества владением CSS, стилем веб-сайта. Рассмотрена основная проблематика и идея автоматизации. Предлагается один из способов воплощения идеи, инструмент, позволяющий реализовать проверки, и механизм получения результата.

Введение

В конкурсах интернет-проектов оценка качества веб-дизайна обычно осуществляется вручную на основе набора критериальных параметров, по которым члены жюри проставляют оценки. Часть критериев имеет характер числовых параметров, например: объем текстового контента сайта и его страниц, число фотографий на сайте, число видеороликов на сайте, число посетителей сайта за определенный промежуток времени. Это обстоятельство создает основание для частичной автоматизации оценки качества. Возможности параметризации качества элементов веб-дизайна можно расширить, что позволяет повысить уровень автоматизации проверки конкурсных работ и тем самым уменьшить трудоемкость и субъективность оценки.

Назначением системы является:

- получение автоматизированной оценки качества использования (CSS);
- автоматизация ряда рутинных работ членов жюри;
- увеличение производительности экспертов, проверяющие большое количество работ участников турнира.

Система CSS-MARK разрабатывается для автоматизации большей части процесса проверки решений участников, используя один из способов проверки, основной идеей которого является оценивание параметров созданного сайта в сравнении с параметрами заданного образца, способ соответствия макету. В ходе автоматической проверки заданий извлекаются параметры созданных страниц и сопоставляются с параметрами образцов. Результатами проверки является оценка

соответствия созданного сайта с макетом в баллах (в процентах) по указанным параметрам.

Построение оценки основывается на проставлении баллов за выполненные работы.

1. Механизм оценки

Оценка на основе функции поощрения для каждого соответствия. Система оценивает уровень соответствия от требования и для каждого вида параметра существует определенный весовой коэффициент, который характеризует степень серьезности, значимости, кроме того учитываем контекст страницы, которая может быть строже оцениваться. Шкала коэффициентов от 0 до 1.

$$E = \sum_{i=1}^{N} w_i * g_i * p_i, \quad (1)$$

Где w_i - сложность элемента,

g_i - качество реализации элемента,

p_i - степень важности страницы.

В результате просчитывается процент соответствия макету.

2. Использование средств автоматизации тестирования

Для того чтобы не поставить участника в строгие рамки, выявляем более важные для проверки объекты, такие как текст, изображение, ссылки и т.д. Для каждого из объектов выделяем достаточные условия, например, для изображения: разность размеров не превышает определенную погрешность, расположение объектов должны попадать в одну зону позиционирования.

Для реализации данной проверки используется инструмент Selenium Webdriver, управляющий реальным браузером, который можно использовать как для автоматизации тестирования веб приложений, так и для выполнения других рутинных задач, связанных с работой в вебе.

Основными понятиями в Selenium Webdriver являются:

- Webdriver - самая важная сущность, ответственная за управление браузером. Основной ход скрипта/теста строится именно вокруг экземпляра этой сущности.
- Webelement - вторая важная сущность, представляющая собой абстракцию над веб элементом (кнопки, ссылки, инпута и др.). Webelement инкапсулирует методы для взаимодействия пользователя с элементами и получения их текущего статуса.

- By - абстракция над локатором веб элемента. Этот класс инкапсулирует информацию о селекторе (например, CSS), а также сам локатор элемента, то есть всю информацию, необходимую для нахождения нужного элемента на странице[1].

Пример использования Selenium в системе для проверки объекта типа «текст» представлен на Листинге 1., Рис.1. и Рис.2.:

```
case "text":
String fstyle = webEl.webElement.getCssValue("font-style");
String fsize = webEl.webElement.getCssValue("font-size");
String fcolor = webEl.webElement.getCssValue("color");
String fstyleCheck = webEl.webElementCheck.getCssValue("font-style");
String fsizeCheck = webEl.webElementCheck.getCssValue("font-size");
String fcolorCheck = webEl.webElementCheck.getCssValue("color");
/*Check this parametres and sum for result */
//Check fstyle
if(fstyle.equals(fstyleCheck)){
        result++;
}else{
        System.out.println("Note the font-style of the text: " +
webEl.webElementCheck.getText());
        result++;
}
//Check fsize
if(ConvertAndCheckFSize(fsize, fsizeCheck)){
        result++;
}else{
System.out.println("Note the font-size of the text: " +
webEl.webElementCheck.getText());
        result++;
}
//Check fcolor
if(ConvertAndCheckFColor(fcolor,fcolorCheck)){
        result++;
}else{
System.out.println("Note the font-color of the text: " +
webEl.webElementCheck.getText());
            result++;
}
break;
```

Листинг 1. Проверка для объекта типа «текст»

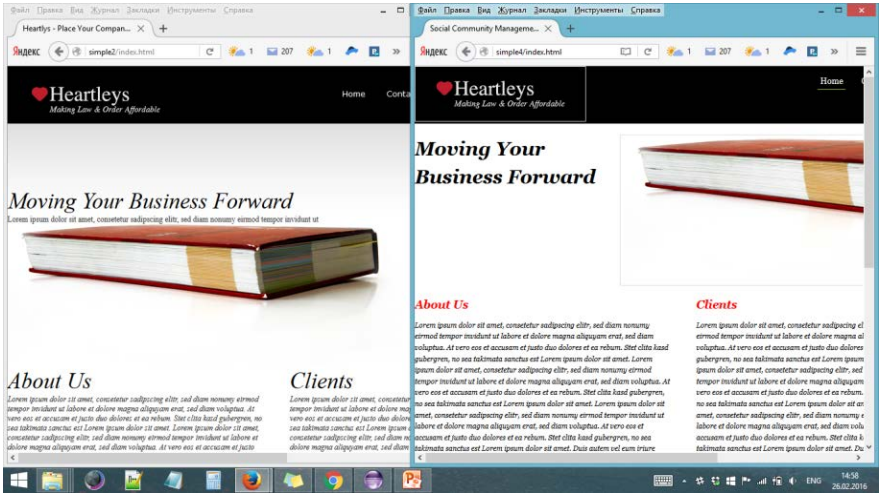


Рис.1. Сайт-образец (слева) и сайт-решение

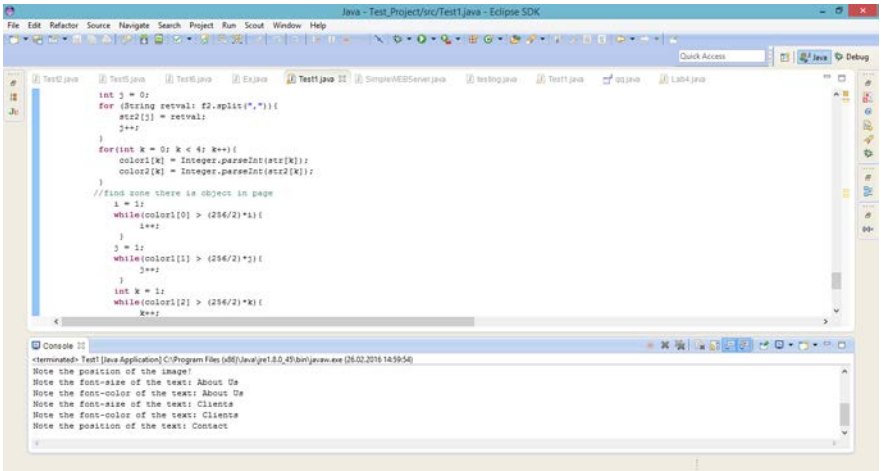


Рис.2. Результаты проверки объекта типа «текст»

Подробная информация о несоответствиях протоколируется (см. Рис.3.).

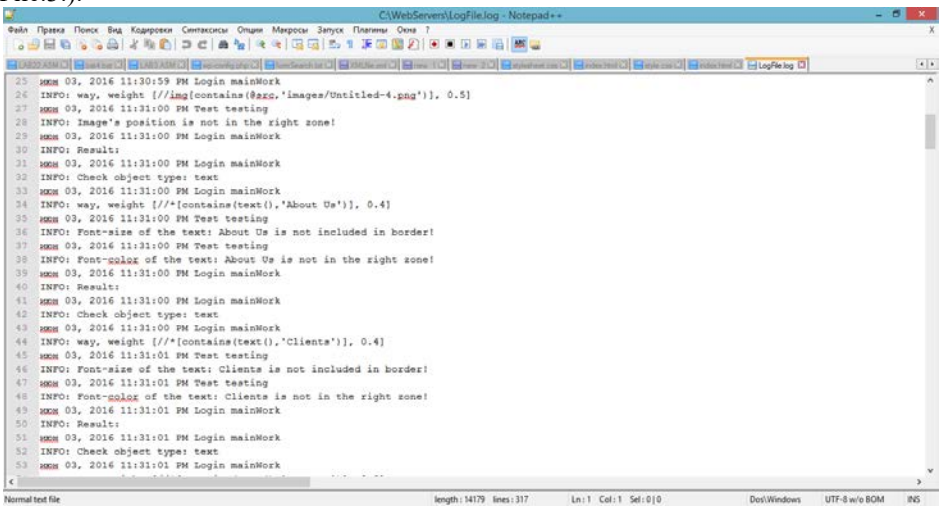


Рис.3. Протоколирование результатов проверки

4. Расширение системы

В качестве наиболее важных направлений дальнейших исследований и разработки следует указать:

- разработка общей платформы для автоматического сбора необходимой информации об участнике и запуск системы проверки;
- расширение списка типов проверяемых объектов;
- разработка более общего способа определения локаторов веб-элементов;
- автоматическое установление коэффициентов сложности элементам в зависимости от вводимых параметров, таких как свобода, аккуратность, идентичность и т.д.;
- модернизация формирования задания.

Список литературы

1. Selenium Webdriver[Электронный ресурс]: книжный онлайн ресурс GitBook - Режим доступа: https://www.gitbook.com/book/kreisfahrer/selenium-webdriver/details

УДК-004.048

ИСПОЛЬЗОВАНИЕ ДИАГРАММ СИСТЕМ

Г.И. Шайдуллова

В статье предлагается способ по моделированию работы системы по средствам построения диаграмм. Рассмотрено понятие система. Предоставлены виды связи между компонентами.

Введение

Под системой понимается множество элементов, находящихся в отношениях и связях друг с другом, которое образует определённую целостность, единство. На практике слово «система» может употребляться в различных значениях, как совокупность объектов природы и способ организации мыслительной деятельности, так и совокупность установившихся норм жизни и правил поведения.

Диаграммы систем представляют собой модели, которые используют для визуального отображения динамических сил, действующих на компоненты, взаимодействий между этими силами и взаимоотношений между различными видами деятельности, между причиной и следствием. В общем, характеризует поведение системы.

Во многих областях появляется необходимость в повышении производительности работы системы, поэтому крайне важно уметь прогнозировать непредвиденные последствия при принятии решении. Диаграмма систем позволяет моделировать процесс, включающий в себя петли обратной связи, усиливающие петли и внешние факторы, которые влияют результат. Построенные модели показывают, на какие компоненты следует обращать внимание для улучшения системы.

1. Связи между компонентами

В основе использования диаграмм заложена идея связывания компонентов.

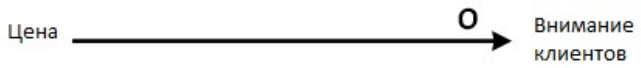


Рис.1. Обратная связь (O)

На рис. 1. показана обратная связь элементов цены и мнении клиентов. Увеличение цены приводит к уменьшению внимания клиентов.

Во многих случаях изменение одного компонента влияет на другой компонент, который будет влиять на первый. Обратная связь будет либо уменьшать влияние изменений, либо усиливать.

Там, где обратная связь уменьшает влияние изменения, обозначается балансирующей петлей. На рис. 2. приведен пример петли балансировки, где с ограниченными ресурсами компания пытается повысить качество продукта:

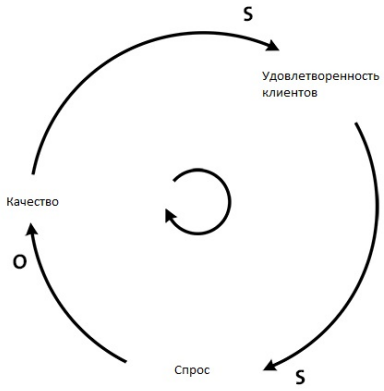


Рис.2. Пример балансирующей петли

Данную ситуацию можно представить в виде графика, представленном на рис. 3.

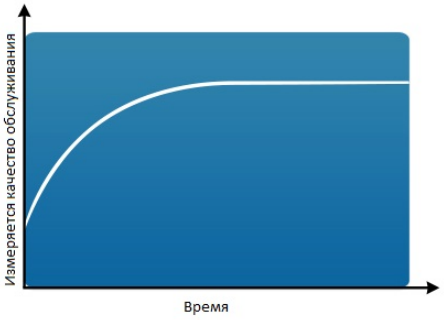


Рис.3. Зависимость уровня качества от времени

Если обратная связь увеличивает влияние изменений, то такая зависимость обозначается усиливающей петлей. На рис. 4. приведен пример театра, который пытается повысить свою рентабельность за счет инвестиций.

Рис.4. Пример усиливающей петли

Чем больше инвестиций вкладывается в рекламу театра, тем больше пьес с более известными актерами театр способен ставить. Хорошая игра должна принести большие просмотры, и, следовательно, более высокий уровень продаж билетов, что в свою очередь положительно влияет на прибыль театра.

График, показывающий, как продажи билетов могут меняться в зависимости от времени изображен на рис.5.:

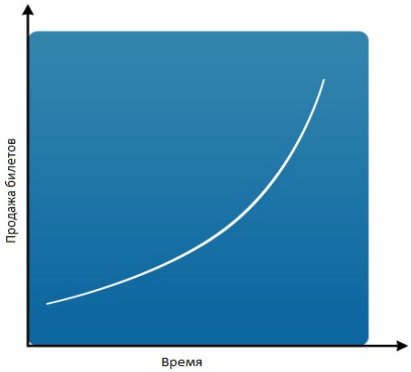


Рис.5. Зависимость продажи билетов от времени

Когда вносятся изменения в систему, она должна немедленно приспособиться к его новому состоянию. На практике всегда существует задержка по времени перед переходом на другой компонент. Такая задержка на диаграмме обозначается двойным слешем. Также на процесс влияют внешние факторы.

После того, как установлены взаимоотношения между компонентами на диаграмме, необходимо рассмотреть возможность задания количественного характера отношениям. Это позволит проектировать процессы в виде компьютерных моделей путем задания отношений через электронную таблицу.

В дальнейшем построенные в виде диаграмм модели, можно использовать для прогнозирования возможных следствий. Также позволит оценить вероятное влияние на систему внешних изменений, а также исследовать влияние изменений, которые могут произойти внутри системы.

Диаграммы систем фиксируют результаты проектной деятельности. Изучая диаграмму, можно обнаружить проблемы и недостатки рассматриваемой системы. На практике диаграммы моделируются во время проектирования системы, и используются в качестве основы для уточнения и конкретизации различных аспектов системы. Можно использовать карандаш и бумагу, или программное обеспечение, чтобы построить диаграмму системы.

Список литературы

1. Systems Diagrams. Understanding How Factors Affect One Another [Электронный ресурс]: https://www.mindtools.com/pages/article/newTMC_04.htm
2. Systems Diagrams. [Электронный ресурс]: http://deseng.ryerson.ca/dokuwiki/design:system_diagram

УДК 621.372

ОБЗОР ТЕХНОЛОГИЙ И РАБОЧИХ ПРОЦЕССОВ SHAREPOINT[53]

А.Н. Афанасьев[54], В.А. Гордеев[55], О.Г. Шаров[56]

Рассматриваются история создания технологий ShatePoint и описание платформы SharePoint 2013. Описание рабочих процессов SharePoint и их разработка в SharePoint Designer и Visio.

1. История SharePoint

В истории создания технологий SharePoint можно выделить несколько периодов.

1. В 2001 г. Microsoft выпустил сервер порталов SharePoint Portal Server 2001. Целью создания сервера было предоставление настраиваемой среды портала, предназначенной для совместной работы, управления документами и совместного использования знаний. Этот продукт содержал технологию "цифровой инструментальной панели" (Digital Dashboard). Также была создана служба SharePoint Team Services как отдельный продукт SharePoint Portal. Наличие двух отдельных продуктов с похожими названиями вносило путаницу. Часто говорили просто о "SharePoint", и не было понятно, идет речь о SharePoint Portal, о командной службе SharePoint или об обеих этих технологиях.

2. В версии 2003 продуктов SharePoint была разработана служба Windows SharePoint Services в качестве базового механизма для среды совместной командной работы.

3. Далее версия SharePoint Team Services была заменена версией Windows SharePoint Services 2.0, в которую были включены много новых и расширенных средств, часть из которых ранее входила в SharePoint Portal Server 2001. В то же время продукт Windows SharePoint Services 2.0

---

[53] *Исследование выполнено при финансовой поддержке РФФИ в рамках научного проекта № 15-07-08268 а*

[54] *432027, Ульяновск, ул. Северный Венец, 32, УлГТУ, e-mail:a.afanasev@ulstu.ru*

[55] *432027, Ульяновск, ул. Северный Венец, 32, УлГТУ, e-mail:gorde-vlad@yandex.ru*

[56] *432027, Ульяновск, ул. Северный Венец, 32, УлГТУ, e-mail:o.sharov@gmail.com*

появился как дополнительный компонент в составе Windows Server 2003. Сервер порталов SharePoint Portal Server 2003 остался отдельным серверным продуктом. Он был основан на технологии службы Windows SharePoint и задуман как решение «уровня предприятия» для объединения внутренних и внешних источников информации. Сервер порталов SharePoint Portal Server 2003 позволял производить поиск в нескольких сайтах и обеспечивал интеграцию бизнес-приложений в портал.

Эти версии SharePoint были более тесно интегрированы с продуктами Microsoft Office 2003, помогая пользователям работать со средствами SharePoint 2003, не теряя при этом возможности применять удобные функции приложений из пакета Office 2003. Например, пользователи могли создавать рабочие пространства встреч и документов непосредственно из продуктов Office 2003.

Большинство приложений из пакета Office 2003 включали также панель Shared Workspace, благодаря которой пользователи могли видеть информацию, хранящуюся на сайте, если был открыт документ, в котором они производили изменения.

4. Следующим продуктом был SharePoint 2007. Многие организации уже обладали опытом работы с первой и второй версиями продукта, и с нетерпением ждали версии "v3”, зная, что она станет еще более зрелой и в ней появится ряд новых средств. Начиная с семейства SharePoint 2007, продукт избавился от термина "Portal”, что избавила от путаницы, оно включает SharePoint Server 2007 и Windows SharePoint Services 3.0. В продуктах SharePoint "v3” также продолжена тенденция тесной интеграции с продуктами Office. Кроме того, хорошо работая с продуктами Office 2003, они оптимизированы для использования с продуктами Office 2007. В Microsoft также решили выделить этот компонент из серверного продукта и сделать его доступным отдельно в виде SharePoint Server 2007 for Search. Вдобавок в Microsoft был предложен набор средств, доступных только при условии активизации средств Enterprise во время или после процесса установки SharePoint Server 2007: в первую очередь это Excel Services, Business Data Catalog и Web Based InfoPath [4].

2. Что такое рабочие процессы SharePoint?

Рабочий процесс иногда рассматривается как некая последовательность задач, которая приносит результат. Говоря о рабочих процессах Microsoft SharePoint, выделяется более узкое определение, а именно — автоматизированное движение документов или элементов через

последовательность действий и задач, связанных с бизнес-процессом. Рабочие процессы могут быть использованы для постоянного управления общими бизнес-процессами предприятия, обеспечивая его возможностью включить бизнес-логику в документы и элементы в списке или библиотеке SharePoint [2]. В сущности, бизнес-логика представляет собой набор инструкций, которые задают и контролируют действия с документами или элементами.

Задача рабочих процессов уменьшить затраты и время, которые требуются, чтобы скоординировать бизнес-процесс. Такими могут являться:

- утверждение проекта,
- проверка документа,
- управление и отслеживание задач человека.

Например, на узле Office SharePoint Server 2007 рабочий процесс можно включить в библиотеку документов, из которой определенный документ направляется группе людей на утверждение. Когда автор документа запускает данный рабочий процесс для документа в указанной библиотеке, этот рабочий процесс создает задачи по утверждению документа, назначает их участникам рабочего процесса, а затем отправляет этим участникам уведомления электронной почтой, содержащие инструкции по выполнению задачи и ссылку на документ, подлежащий утверждению. В ходе выполнения рабочего процесса его владелец (в данном случае — автор документа) или участники могут на странице «Состояние рабочего процесса» узнать, кто из участников уже выполнил порученные задачи. По завершении всеми участниками своих задач рабочий процесс прекращается, а его владелец автоматически уведомляется об этом.

Действия рабочего процесса утверждения в данном примере показаны на следующем рисунке.

Рис1. Блок-схема рабочего процесса утверждения

Рабочие процессы не только поддерживают существующие процессы, выполняемые людьми, но и расширяют возможности по сотрудничеству и совместной работе с документами, списками и библиотеками. Пользователи узлов могут начинать рабочие процессы и участвовать в них при помощи настраиваемых форм, которые доступны из документов и элементов списка или библиотеки SharePoint. Кроме того, возможности рабочих процессов в Office SharePoint Server 2007 тесно интегрированы с выпуском 2007 набора Microsoft Office. Следующие задачи рабочего процесса могут быть выполнены в узле Office SharePoint Server 2007:

- просмотр списка рабочих процессов для документа или элемента,
- запуск рабочего процесса для документа или элемента,
- просмотр, изменение и переназначение задач рабочего процесса,
- выполнение задачи рабочего процесса.

3. Платформа рабочих процессов SharePoint 2013

Платформа рабочих процессов SharePoint 2013 использует диспетчер рабочих процессов (англ. Workflow Manager), который функционирует на базе рабочих процессов WF 4.0. В свою очередь, WF является частью .NET Framework 4.5.

Диспетчер рабочих процессов — это мощная служба, которая может использоваться множеством различных приложений, в том числе SharePoint Server 2013. Одна из основных идей разработки новой платформы заключается в том, чтобы предоставить полноценный функционал рабочих процессов пользователям SharePoint Server 2013 без жесткой привязки к самой платформе. Этот подход обеспечил возможность решать проблемы производительности, отказоустойчивости и масштабируемости SharePoint Server 2013 и диспетчера рабочих процессов независимо друг от друга [5].

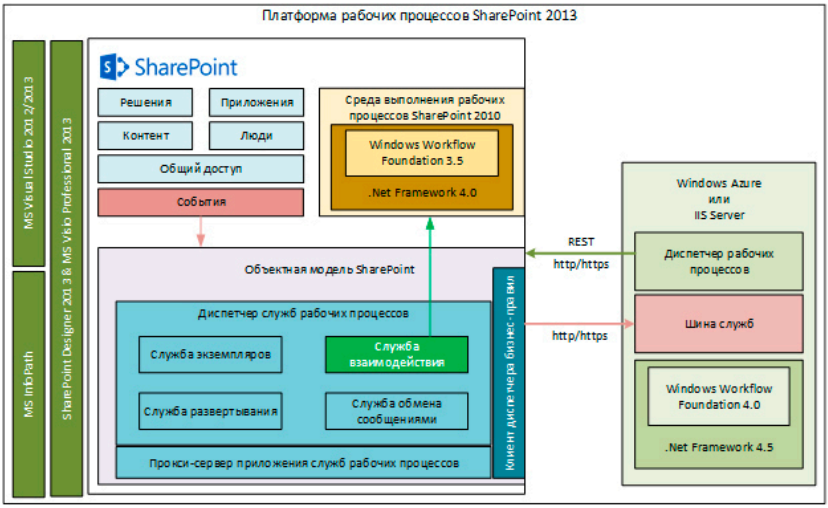


Рис2. Общая архитектура платформы рабочих процессов SharePoint 2013[5]

SharePoint 2013 использует диспетчер, отвечая только за поддержку моделей рабочих процессов и реализацию логики выполнения действий. Принципиальным в новой архитектуре является перенос процессов на сторону диспетчера, что позволяет вынести их за пределы SharePoint 2013, обеспечив полностью независимое использование серверных ресурсов.

Диспетчер рабочих процессов - это полностью самостоятельный программный продукт, который должен быть загружен, инсталлирован и

сконфигурирован, после чего будет обеспечено его подключение к SharePoint.

Для взаимодействия с диспетчером рабочих процессов SharePoint Server 2013 использует специализированный программный интерфейс (англ. Application Program Interface, API). Весь необходимый набор компонентов для этого включен в соответствующего клиента (англ. Workflow Manager Client). При этом все обращения реализуются через служебную шину.

Платформа рабочих процессов SharePoint 2013 базируется на модели облачных приложений. Диспетчер рабочих процессов при этом может быть установлен на отдельный сервер во внутренней сетевой инфраструктуре компаний, так и развернут в качестве сервиса в Windows Azure.

На практике компании часто в своей внутренней сетевой инфраструктуре разворачивают диспетчера рабочих процессов и шину служб для Windows Server на серверах SharePoint.

Важным моментом является то, что настройка рабочих процессов SharePoint 2013 доступна только в серверных редакциях, т.е. в SharePoint Server 2013 Standard и SharePoint Server 2013 Enterprise.

Перечислим основные преимущества, которые имеет новая архитектура рабочих процессов.

1. **Мульти-серверный хостинг**. Он обеспечивает высокую эффективность и скорость исполнения экземпляров рабочих процессов в сочетании с легкостью управления. Также он помогает повысить уровень безопасности, увеличивает гибкость масштабирования и улучшает распределения нагрузки между удаленными серверами диспетчера рабочих процессов.

2. **Масштабирование**. Управление разработкой рабочих процессов, реализацией бизнес-логики и формированием их экземпляров на стороне фермы SharePoint Server 2013 внутри сетевой инфраструктуры компании позволяет сохранять использование ресурсов серверов фермы на постоянном уровне, и в то же время выделять по требованию ресурсы, необходимые для непосредственного исполнения рабочих процессов.

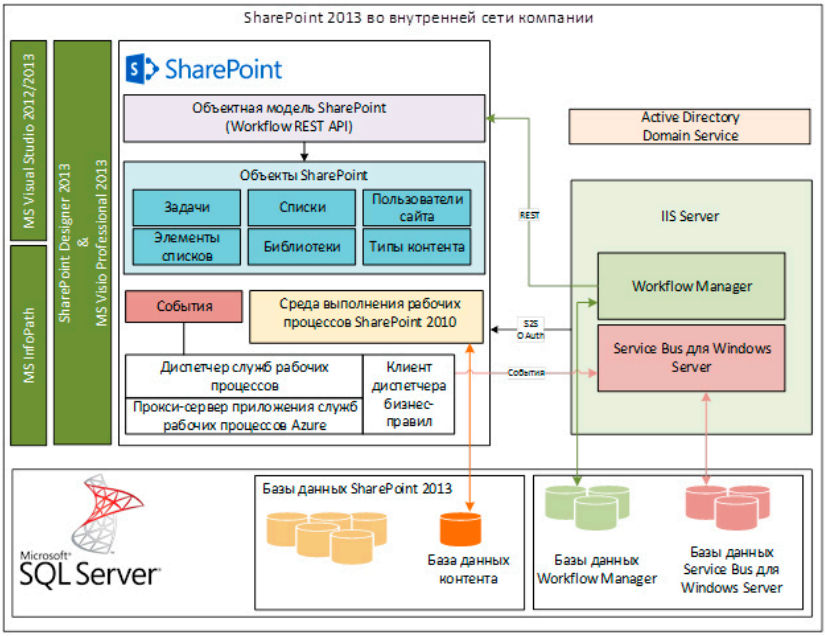


Рис3. Архитектура платформы рабочих процессов во внутренней сетевой инфраструктуре компании [5]

3. **Расширенные средства диагностики и мониторинга**. Новые функциональные возможности мониторинга и диагностики исполняющихся экземпляров рабочих процессов используют REST API и клиент диспетчера рабочих процессов. Системным администраторам предоставляется специализированный набор инструментов, в то время как клиентам доступен просмотр данных о состоянии и статусах запущенных рабочих процессов, гибкое управление их конфигурационными настройками.

4. **Полностью декларативная разработка рабочих процессов**. Это новая концепция впервые была использована на платформе SharePoint. Основными средством разработки для пользователей SharePoint Server 2013 становятся SharePoint Designer 2013, MS Visio 2013, MS InfoPath 2013. Для разработчиков Visual Studio 2012/2013 представлен набор специализированных шаблонов проектов и мощный инструментарий разработки действий рабочих процессов. Действия рабочих процессов, разрабатываемые в MS Visual Studio, ориентированы на последующее использование в SharePoint Designer 2013.

5. **Реализация в виде веб-сервисов**. Действия рабочих процессов SharePoint Server 2013 вызываются с помощью слабосвязанных веб-служб, использующих программный интерфейс для взаимодействия с SharePoint. Он базируется на функциях обмена сообщениями Windows Communication Foundation (WCF) и использовании REST-запросов. Поддержка среды исполнения, входящих и исходящих сообщений координируют все этапы выполнения рабочих процессов, гарантируя целостность и надежность.

4. Понимание процесса: Какие вопросы мы должны задавать?

Что мы должны знать о рабочем процессе, как минимум для того, чтобы иметь возможность автоматизировать или улучшить его с помощью SharePoint [3]?

- Технологические подробности
- Какое название процесса?
- Кто участники процесса?
- Какие системы, участвуют в этом процессе?
- Каковы предпосылки для этого процесса?
- Каковы условия после размещения процесса?
- Как долго длится процесс?
- Как часто этот процесс происходит?
- Подробный анализ:
- Какие существуют болевые точки в этом процессе?
- Примеры:
- Отслеживание и отчетность
- Обработка исключений
- Заполнение бланков
- Уведомлять участников
- И т.д
- Какие "Исключения" в этом процессе?
- Что такое процесс? (отображение процессов деятельности.)
- Какие права и соображения конфиденциальности на протяжении всего процесса?
- Потенциальные результаты:
- Карта процесса
- Все шаги
- Все формы / Документы
- Все исключения
- Все Актеры / Роли
- Все отчеты

5. Разработка рабочих процессов в ShaprePoint Designer и Visio

С помощью Visio 2013 и SharePoint Designer 2013 намного проще строить рабочие процессы для бизнес-аналитиков, консультантов и ИТ-специалистов. Так же Visio профессиональный 2013 и визуальный конструктор SharePoint Designer 2013 используют расширенное представление рабочих процессов в формате, которое понятно программистам и далеким от программирования пользователям.

При помощи Visio 2013 можно создать визуальный рабочий процессы SharePoint, после чего есть возможность экспортировать его в SharePoint Designer 2013, а затем опубликовать на сайте SharePoint.

После создания и экспортирования рабочего процесса, владелец сайта или ИТ-специалист добавляет параметры рабочего процесса, используя текстовый редактор и новый визуальный конструктор рабочих процессов, который представляет собой элемент управления ActiveX Visio 2013, размещенный в SharePoint Designer 2013.

Эти возможности идеально подходят для бизнес-аналитиков и консультантов по процессам, которые уже знакомы с блок-схемами в Visio, поскольку они позволяют создавать рабочие процессы, представляющие бизнес-логику. Пользователь, проектирующий рабочий процесс, может сосредоточиться исключительно на бизнес-аналитике, при этом не надо быть экспертом по декларативным рабочим процессам [1].

Список литературы

1. https://msdn.microsoft.com/ru-ru/library/office/jj163272.aspx
2. https://support.office.com/ru-ru/article/%D0%A0%D0%B0%D0%B1%D0%BE%D1%87%D0%B8%D0%B5-%D0%BF%D1%80%D0%BE%D1%86%D0%B5%D1%81%D1%81%D1%8B-%D0%B2-SharePoint-d74fcceb-3a64-40fb-9904-cc33ca49da56
3. http://www.rharbridge.com/?p=1134
4. Microsoft SharePoint 2007. Полное руководство (ИД "Вильямс”, 2008 г.).
5. https://habrahabr.ru/post/266943/

УДК 621.372

РАЗРАБОТКА ГЕНЕРАТОРА НЕГАТИВНЫХ BPMN ДИАГРАММ НА ОСНОВЕ RV-ГРАММАТИКИ[57]

А.Н.Афанасьев[58], С.Ю.Кириллов[59], Е.Ю.Воеводин[60]

Предложена автоматная грамматика для генерации диаграмм языка представления бизнес-процессов, получившая реализацию в программном продукте

Введение

После разработки RV-грамматики и её реализации в качестве плагина MS Visio для анализа диаграммного языка BPMN возникла задача тестирования. Был успешно применен метод ручного тестирования. К сожалению он потребовал много усилий и времени, поэтому необходимо было провести автоматизированное тестирование. В статье [1] предложена методика генерации правильных диаграмм. Такие диаграммы могут использоваться для нагрузочного тестирования, но полностью оценить корректность работы приложения не позволяют, необходимо разработать метод автоматической генерации негативных тестов для анализатора.

1. Методика генерации тестов

Приведем определения позитивного и негативного тестов.

Позитивный тест – последовательность квазитермов (диаграмма), являющаяся предложением целевого диаграммного языка. Негативный тест – последовательность квазитермов (диаграмма), не являющаяся предложением целевого диаграммного языка.

Часто на практике для тестирования используются методы динамической верификации [2], которые предполагают решение следующих задач [3]:

1) генерация (написание) тестов;

2) вынесение вердикта о прохождении теста, выполняемое с помощью так называемого тестового оракула, который представляет собой

[57] *Исследование выполнено при финансовой поддержке РФФИ в рамках научного проекта № 15-07-08268 а*

[58] *432027, Ульяновск, ул. Сев. Венец, 32, УлГТУ, e-mail: a.afanasev@ulstu.ru*

[59] *432027, Ульяновск, ул. Сев. Венец, 32, УлГТУ, e-mail:xayc73@gmail.com*

[60] *432027, Ульяновск, ул. Сев. Венец, 32, УлГТУ, e-mail:voevodineu@gmail.com*

процедуру, определяющую корректность работы тестируемой системы на данном тесте;

3) оценка качества тестов, выполняемая с помощью специальных метрик, получивших название критерии тестового покрытия;

Генерации негативных тестов посвящено достаточно малое количество работ. Если говорить о синтаксических правилах, то в работе [4] в качестве метода для генерации негативных тестов предлагается так называемый метод мутационного тестирования. Основой данного метода является предположение о том, что после внесения в исходную грамматику языка ряда изменений (мутаций) ее можно использовать для получения потенциально негативных тестов. Однако при использовании подобного подхода можно столкнуться со следующими проблемами:

• грамматика, полученная с помощью мутаций, может оказаться эквивалентной исходной грамматике;

• тесты, сгенерированные на основе грамматики-мутанта, не являющейся эквивалентной исходной, могут оказаться корректными.

В работе [5] описаны методы генерации позитивных и гарантированно негативных тестов, а также разработанные для них критерия покрытия. Практическую применимость описанных в работе методов подтвердили проведенные испытания по тестированию промышленных компиляторов.

Единственной работой, в которой помимо метода генерации негативных тестов, описан также и критерий покрытия для множества негативных тестов является работа [9].

Во всех описанных случаях используется видоизмененный метод мутации. Применим его для нашего генератора негативных диаграмм.

Идея мутации состоит в том, что в исходную грамматику вносятся изменения (мутации) для получения грамматик, описывающих близкие, но не эквивалентные исходному языки. Эти мутированные грамматики подаются на вход генератору тестов для получения потенциально негативных тестов. Рассмотрим его применимость для нашего случая, и, для начала, уточним алгоритм получения RV-грамматики.

Синтез RV – грамматики состоит из следующих этапов.

1. Определяется терминальный алфавит контролируемого графического языка, описывается расположение меток, выявляются семантические различия для связей, имеющих общее графическое представление, строится алфавит квазитермов.

2. Строится матрица допустимых паросочетаний для квазитерминального алфавита.

3. Определяются отношения над внутренней памятью, обеспечивающие эффективный контроль связности графических объектов.

4. По матрице допустимых паросочетаний, т.е. системе отношений, строится граф метаязыка RV – грамматик, вершинам которого поставлены в соответствие имена комплексов правил, а дугам – квазитермы и операции над внутренней памятью. Помимо графовой метаязык RV – грамматики может быть представлен также в табличной и аналитической формах.

После синтеза производится анализ, который осуществляется в два этапа.

1. Устраняются недетерминированности и неопределенности.
2. Производится минимизация RV – грамматики.

Исходя из вышеописанного алгоритма, для построения каждой негативной грамматики потребуется выполнять всё с первого пункта синтеза, поскольку уже в нем возможны изменения семантических особенностей квазитермов. Если предположить, что одна негативная грамматика будет содержать в себе лишь одну заложенную ошибку, то получается, что их необходимо как минимум 10. При дальнейшем анализе, увеличении элементов и других обстоятельств это количество будет только повышаться. Поскольку составление грамматики не автоматизировано и происходит вручную, необходимо затратить много времени, чтобы подготовить множество таких негативных грамматик, даже если увеличить количество заложенных в грамматику ошибок.

Рассмотрим альтернативные способы.

Одним из них является мутация уже сгенерированной диаграммы правильным генератором. Основными операциями являются: удаление токена из цепочки, вставка токена.

На их основе возможны более комплексные операции, такие как замена и перемещение. Также можно применять изменение некоторых свойств, присущие квазитерму.

Рассмотрим все возможные варианты

1. Удаление элемента из цепи (рис. 1).

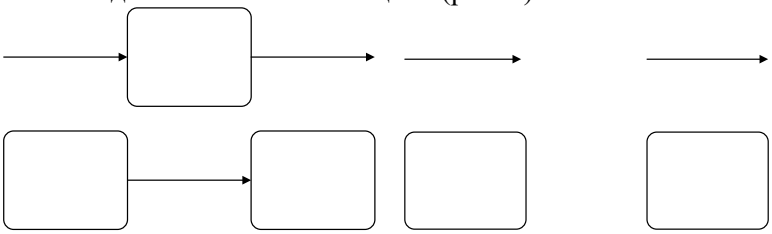


Рис. 1. Удаление элемента из цепи элементов

2. Вставка элемента (рис. 2).

В Visio структура диаграммы разделена на связи и элементы. Поэтому связки типа «элемент и элемент» и «связь и связь» невозможны, в таком случае выдаётся ошибка, т.е. имеется возможность лишь вставки дополнительной связи к элементу или целой цепочки.

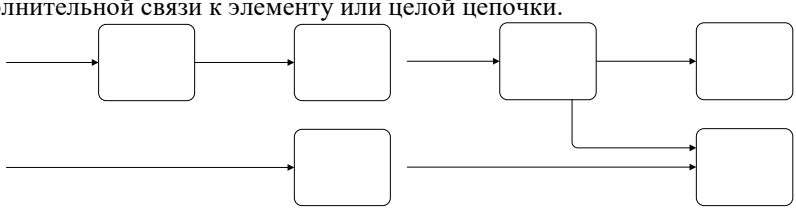


Рис. 2. Вставка элемента

3. Вставка цепочки связь + элемент (рис. 3)

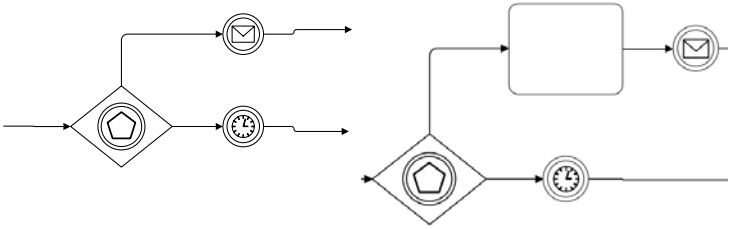


Рис. 3. Вставка цепочки связь + элемент

4. Замена элемента на элемент с похожими свойствами (рис. 4)

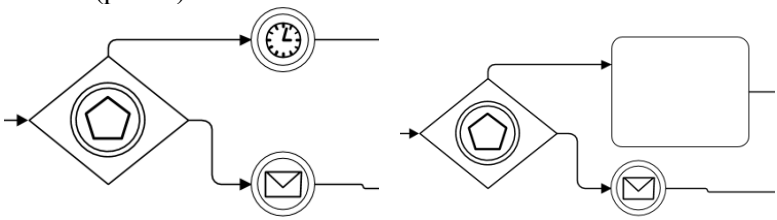


Рис. 4. Замена элемента на элемент с похожими свойствами

5. Замена элемента на элемент с другими свойствами

Несмотря на то, что синтаксически элементы событий ограничены по связям, Visio позволяет нам сделать замену. Рассмотрим все варианты:

- Заменить с сохранением всех связей (рис. 5),

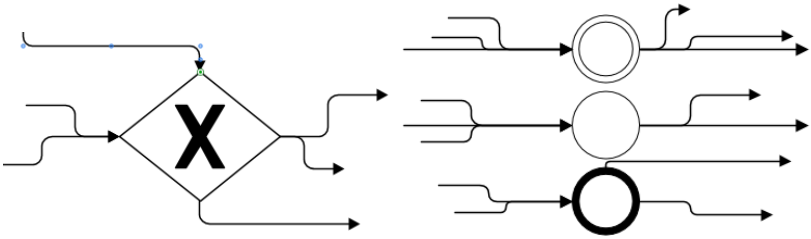


Рис. 5. Замена с сохранением всех связей

- Оставить лишние связи висящими (рис. 6),

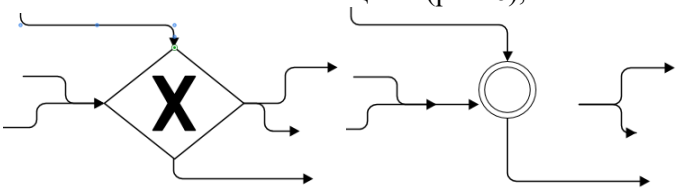


Рис. 6. Лишние связи остались висящими

- Соединить висящие связи с другими элементами, удовлетворяющими условиям (рис. 7),

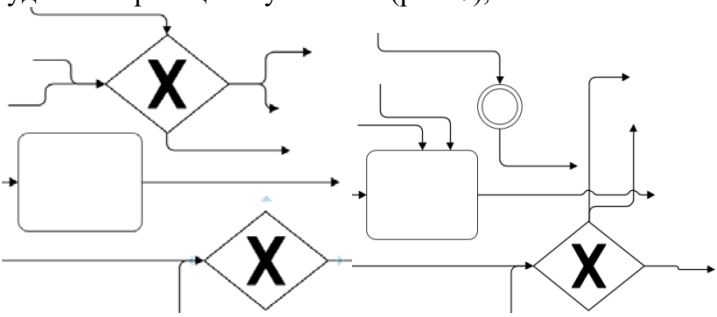
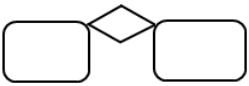
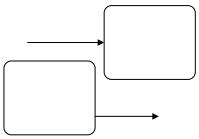
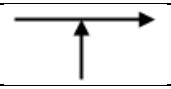
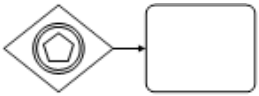
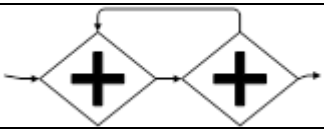


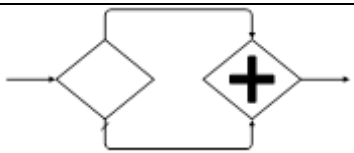
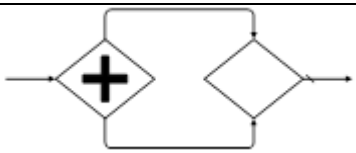
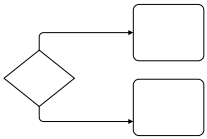
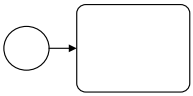
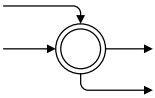
Рис. 7. Соединение висящих связей с другими элементами, удовлетворяющими условиям

- Часть связей оставить висящими, а часть соединить с другими элементами.

Список возможных ошибок показан в таблице 1.

Таблица 1. Возможные ошибки

Вид ошибки	**Пример**	**Графическое отображение**	**Есть ли возможность генерации**
Отсутствие связи	Эксклюзивный шлюз не соединен с действиями. Параллельный шлюз не соединен с действиями Другие		Да
Висячая связь	Связь, исходящая из элемента, но не входящая никуда. Связь, входящая в элемент, но не исходящая из другого элемента		Да
Ошибка передачи управления	Связь, входящая в другую связь		Да (в Visio нет)
Недопустимая последовательность элементов	После шлюза, основанном на событиях, следует действие, а не триггер события.		Да
Ошибка назначения	Параллельный шлюз не выполняет никакой функции		Да
«DeadLock»	Возвращение параллельной ветви в параллельный шлюз		Да

Зависание	Параллельный шлюз, идущий после эксклюзивного шлюза		Да
Неопределенность	Эксклюзивный шлюз, идущий после параллельного шлюза		Да
Отсутствие пути потока	Отсутствует поток, по умолчанию у эксклюзивного шлюза и все остальные условия равны «false»		Да
Старт/конец	Если есть хоть одно стартовое событие, то должно быть и конечное		Да
Ошибка кратности входящих и исходящих потоков	Событие имеет много входящих и исходящих ветвей		Да

Недостатком данного алгоритма является необходимость повторного прохода по полученной цепочке, что увеличивает время работы алгоритма.

Заключение

Разработка тестового генератора для анализатора диаграммных моделей языка BPMN является первым шагом на пути решения проблемы полноценного тестирования программных средств анализа и контроля диаграммных моделей графических спецификаций. Дальнейшим развитием созданного генератора является расширение алфавита, а также перенос описываемого подхода на другие диаграммные языки и создание универсального генератора моделей графических языков.

Список литературы

1. Шаров О.Г., Афанасьев А.Н. Синтаксически-ориентированная реализация графических языков на основе автоматных графических грамматик // Программирование. – 2005. - № 6. – С. 56-66.
2. Зеленов С.В., Пакулин Н.В. Верификация компиляторов - систематический подход // Труды Института системного программирования РАН. 2007.
3. Посыпкин М.А. Применение формальных методов для тестирования компиляторов // Труды Института системного программирования РАН. 2004.
4. Maurer P.M., "The design and implementation of a grammar-based data generator," Software: Practice and Experience, Vol. 22, No. 3, 1992. pp. 223–244.
5. Зеленов С.В., Зеленова С.А. Автоматическая генерация позитивных и негативных тестов для тестирования фазы синтаксического анализа // Труды Института системного программирования РАН. 2004. Т. 8
6. Шаров О.Г., Афанасьев А. Н. Нейтрализация синтаксических ошибок в графических языках // Программирование. – 2008. – №1. – С. 61-66.
7. Шаров О.Г., Афанасьев, А. Н. Методы и средства трансляции графических диаграмм // Программирование. – 2011. – №3. – С. 65-76.
8. Архипова М.В. Автоматическая генерация тестов для семантических анализаторов трансляторов. Диссертационная работа на соискание степени кандидата физико- математических наук. – Москва, 2006. ИСП РАН.

УДК 621.372

РАЗРАБОТКА ДИНАМИЧЕСКОГО ГЕНЕРАТОРА НЕГАТИВНЫХ BPMN ДИАГРАММ НА ОСНОВЕ RV-ГРАММАТИКИ[61]

А.Н. Афанасьев[62], С.Ю. Кириллов[63], Ю.А. Савичева[64]

Предложена автоматная грамматика для динамической генерации негативных диаграмм языка представления бизнес-процессов.

Введение

В статье [3] рассмотрены методы генерации неправильных диаграмм. Основным их недостатком является наличие дополнительного прохода для генерации. В данной статье предложен метод динамической генерации за один проход.

1. Методика генерации тестов

Рассмотрим исходные данные: имеется некоторое количество графических элементов, у которых есть определенные свойства, а также список ошибок, которые могут возникать, также мы имеем грамматику для генерации правильных диаграмм.

В реализации генератора мы используем вероятностный подход генерации квазитерма. Сам автомат имеет несколько состояний, в которых возможны операции только с некоторыми квазитермами. Это необходимо для формирования определенной последовательности квазитермов. Если вероятность нулевая, то это значит либо квазитерм вообще не обрабатывается в текущем состоянии, либо у него должен быть определенный предшественник. Вероятностное соответствие квазитермов и состояний автомата приведено в таблице 1.

---

[61] *Исследование выполнено при финансовой поддержке РФФИ в рамках научного проекта № 15-07-08268 а*

[62] *432027, Ульяновск, ул. Северный Венец, 32, УлГТУ, e-mail: a.afanasev@ulstu.ru*

[63] *432027, Ульяновск, ул. Северный Венец, 32, УлГТУ, e-mail:xayc73@gmail.com*

[64] *432027, Ульяновск, ул. Северный Венец, 32, УлГТУ, e-mail:jasintar@mail.ru*

Таблица 1 Вероятностное соответствие квазитермов и состояний автомата.

	Квазитерм1	Квазитерм2	…	КвазитермN
Состояние1	Вероятность11	Вероятность12		Вероятность1N
…				
СостояниеM	ВероятностьM1	ВероятностьM2		ВероятностьMN

Если поставить вероятность в заведомо ложное место как вручную, так и автоматически, то произойдет ошибка генерации. Можно попытаться нейтрализовать ошибку, автоматически перейдя в любое следующее состояние. Например, у каждого состояния предполагается квазитерм «default», который выполняется для всех неподдерживаемых состоянием квазитермов и в нем выполняется нужная функция randomState().

Здесь необходимо рассмотреть две проблемы.

1. Ошибка чтения из памяти при переходе на квазитерм, предполагающий предшественника. Поэтому необходимо внедрить условие $c(b^1) > 0$, которое означает, что в нужном магазине должен быть хотя бы один квазитерм. Если условие не выполняется, то продолжается переход в другое состояние.

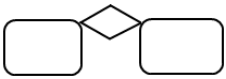
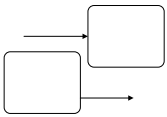
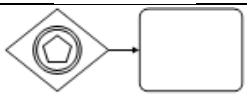
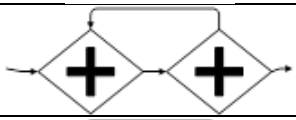
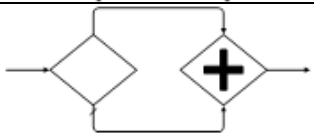
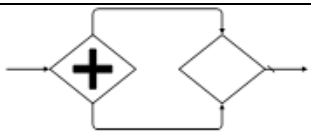
2. Невозможность определить, будет ли являться нейтрализованная цепочка негативной или будет соответствовать исходной грамматики языка. Частично это решается при автоматической генерации вероятностей переходов. Если заведомо известны правильные значения, возможно брать изначально неправильные. Тогда при случайном переходе и вынужденном пропуске состояния скорее всего будет изменена правильная последовательность квазитермов. Хотя этот критерий не дает полной уверенности в этом. Также встает вопрос, в какой же момент генератор должен отойти от правильных переходов и сгенерировать заведомо неправильный. Единственным способом является случайное появление этих событий, частоту таких событий можно регулировать разбросом границ возможных вероятностей, связанных, например, с количеством генерируемых элементов.

Можно заметить, что это не дает генерации всех видов ошибок. В добавок, встает вопрос об отслеживании генерации подобных ошибок. Поскольку алгоритм основан на работе автомата, который генерирует непрерывную выходную цепочку, то генерация ошибки «входящая висячая связь» возможна лишь одним образом. Для этого понадобится функция «обезличивания» квазитерма. При переходе на любой создающий квазитерм полностью соблюдаются операции над памятью и переходы присущие ему, но при записи готового элемента в цепочку не указывается его тип. Тем самым, мы не нарушаем правил и логики, но при последующей интерпретации элемента в графическое представление можем его пропустить и воспроизвести ошибку. Также для создания

ошибки кратности потоков придется расширить грамматику для всех простых элементов. Поскольку генерация происходит динамически, необходимо предполагать все возможные ситуации. Поэтому простой элемент «промежуточное событие», представленный квазитермом Ai, расширяем до дополнительных квазитермов, AiJ, AiF и AiA.

Благодаря вышеуказанному, возможна генерация ошибок для текущего алфавита. Список ошибок представлен в таблице 2.

Таблица 2. Список ошибок

Вид ошибки	**Пример**	**Графическое отображение**	**Возможность генерации**
Отсутствие связи	Эксклюзивный шлюз не соединен с действиями. Параллельный шлюз не соединен с действиями Другие		Да
Висячая связь	Связь, исходящая из элемента, но не входящая никуда. Связь, входящая в элемент, но не исходящая из другого элемента		Да
Ошибка передачи управления	Связь, входящая в другую связь		Да (в Visio нет)
Недопустимая последовательность элементов	После шлюза, основанном на событиях, следует действие, а не триггер события.		Да
Ошибка назначения	Параллельный шлюз не выполняет никакой функции		Да
«DeadLock»	Возвращение параллельной ветви в параллельный шлюз		Да
Зависание	Параллельный шлюз, идущий после эксклюзивного шлюза		Да
Неопределенность	Эксклюзивный шлюз, идущий после параллельного шлюза		Да

Отсутствие пути потока	Отсутствует поток, по умолчанию у эксклюзивного шлюза и все остальные условия равны «false»		Да
Старт/конец	Если есть хоть одно стартовое событие, то должно быть и конечное		Да
Ошибка кратности входящих и исходящих потоков	Промежуточное событие имеет много входящих и исходящих ветвей		Да

В таблице 3 приведена скорректированное множество квазитерминальных символов.

Таблица 3. Скорректированное множество квазитерминальных символов

Терминальный символ	**Квазитерминальный символ**	**Примечание**
	A0	Стартовое событие
	Ai	Промежуточное событие, лишь с одной входящей и одной исходящей связью
	AiJ	Промежуточное событие, выполняющее функцию слияния потоков (Join)
	_AiJ	
	AiF	Промежуточное событие, выполняющее функцию распараллеливания (Fork)
	AiA	Промежуточное событие, выполняющее обе вышеназванные функции одновременно. Предполагается, что элемент имеет не меньше двух входящих и исходящих связей
	_AiA	
	Ak	Конечное событие
	Ake	Конечное событие, у которого входящая связь принадлежит к потоку условия
	A	Действие, лишь с одной входящей и одной исходящей связью
	AM	Действие, выполняющее функцию объединения (Merge, «исключающее или»)
	_AM	
	AF	Действие, выполняющее функцию распараллеливания (Fork)
	EGC	Эксклюзивный шлюз, выполняющий функцию обычного условия (Condition)

	EGM	Эксклюзивный шлюз, выполняющий функцию объединения (Merge, «исключающее или»)
	_EGM	
	EGA	Эксклюзивный шлюз, выполняющий обе функции одновременно. Предполагается, что элемент имеет не меньше двух входящих и исходящих связей
	_EGA	
	PGF	Параллельный шлюз, выполняющий функцию распараллеливания потоков (Fork)
	PGJ	Параллельный шлюз, выполняющий функцию слияния потоков (Join)
	_PGJ	
	PGA	Параллельный шлюз, выполняющий обе вышеназванные функции одновременно. Предполагается, что элемент имеет не меньше двух входящих и исходящих связей
	_PGA	
	rel	Поток операций
	erel	Поток условий
	labelAiF	Метка промежуточного события
	labelAiA	Метка промежуточного события
	labelA0	Метка стартового события
	labelAF	Метка действия
	labelPGF	Метка паралл. шлюза
	labelPGA	Метка паралл. шлюза
	labelEGC	Метка экскл. Шлюза
	labelEGA	Метка экскл. Шлюза
	no_label	Метка завершения генерации

Видоизмененную грамматику можно увидеть в таблице 4.

Таблица 4. RV-граматика для динамической генерации негативных тестов

Пред. состояние	Квазитерм	След. состояние	Операции с памятью
r_0	A0	r1	$W_1(t^{1m^{(k-1)}})$
	A	r2	O
	AF	r1	$W_1(t^{2m^{(k-1)}})$
	PGF	r1	$W_1(t^{3m^{(k-1)}})$
	EGC	r9	$W_1(t^{5m^{(k-1)}})$
	default		randomState()
r_1	rel	r10	O

	default		randomState()
r_2	labelRel	r12	setVer($W_3(b^{12m})$)
	no_label	r13	*
	default		randomState()
r_3	labelEGC	r9	$W_2(b^{5m})/c(b^5) > 0$
	labelEGA	r9	$W_2(b^{6m}) /c(b^6) > 0$
	default		randomState()
r_4	labelEGC	r9	$W_2(b^{5m}) /c(b^5) > 0$
	labelEGA	r9	$W_2(b^{6m}) /c(b^6) > 0$
	labelRel	r12	setVer($W_3(b^{12m})$)
	no_label	r13	*
	default		randomState()
r_5	labelRel	r12	setVer($W_3(b^{12m})$)
	default		randomState()
r_6	labelA0	r1	$W_2(b^{1m}) /c(b^1) > 0$
	labelAF	r1	$W_2(b^{2m}) /c(b^2) > 0$
	labelPGF	r1	$W_2(b^{3m}) /c(b^3) > 0$
	labelPGA	r1	$W_2(b^{4m}) /c(b^4) > 0$
	no_label	r13	*
	default		randomState()
r_7	labelA0	r1	$W_2(b^{1m}) /c(b^1) > 0$
	labelAF	r1	$W_2(b^{2m}) /c(b^2) > 0$
	labelAiF	r1	$W_2(b^{13m}) /c(b^{13}) > 0$
	labelPGF	r1	$W_2(b^{3m}) /c(b^3) > 0$
	labelPGA	r1	$W_2(b^{4m}) /c(b^4) > 0$
	default		randomState()
r_8	labelA0	r1	$W_2(b^{1m}) /c(b^1) > 0$
	labelAF	r1	$W_2(b^{2m}) /c(b^2) > 0$
	labelPGF	r1	$W_2(b^{3m}) /c(b^3) > 0$
	labelPGA	r1	$W_2(b^{4m}) /c(b^4) > 0$

	labelRel	r12	setVer($W_3(b^{12m})$)
	no_label	r13	*
	default		randomState()
r_9	erel	r11	O
	default		randomState()
r_{10}	A	r2	O
	AF	r1	$W_1(t^{2m^{(k-1)}})$ && $W_1(rel^{12m^{(k)}})$
	Ai	r5	O
	AiJ	r7	$W_1(1^{t(13)}, k^{t(14)}, t^{14m^{(k-1)}})/W_2(e^{t(13)})$
	_AiJ	r8	$W_1(inc(m^{t(13)})/W_3(m^{t(13)} < k^{t(14)})$ || $(W_2(b^{13m}), W_2(b^{12m})$, changeState(r2))$/W_3(m^{t(13)} = k^{t(14)})$
	AiF	r1	$W_1(t^{13m^{(k-1)}})$ && $W_1(rel^{12m^{(k)}})$
	AiA	r7	$W_1(1^{t(15)}, k^{t(16)}, t^{15m^{(k-1)}})/W_2(e^{t(15)})$
	_AiA	r7	$W_1(inc(m^{t(15)})/W_3(m^{t(15)} < k^{t(16)})$ || $W_1(W_2(b^{16m})^{15m})$&&$W_1(rel^{12m^{(k)}})/W_3(m^{t(15)} = k^{t(16)})$
	Ak	r6	$W_2(b^{12m})/c(b^{12}) > 0$
	PGF	r1	$W_1(t^{3m^{(k-1)}})$ && $W_1(rel^{12m^{(k)}})$
	PGJ	r7	$W_1(1^{t(5)}, k^{t(6)}, t^{8m^{(k-1)}})/W_2(e^{t(5)})$
	_PGJ	r8	$W_1(inc(m^{t(5)})/W_3(m^{t(5)} < k^{t(6)})$ || $(W_2(b^{8m}), W_2(b^{12m})$, changeState(r2))$/W_3(m^{t(5)} = k^{t(6)})$
	PGA	r7	$W_1(1^{t(7)}, k^{t(8)}, t^{9m^{(k-1)}})/W_2(e^{t(7)})$
	_PGA	r7	$W_1(inc(m^{t(7)})/W_3(m^{t(7)} < k^{t(8)})$ || $W_1(W_2(b^{9m})^{4m})$&&$W_1(rel^{12m^{(k)}})/W_3(m^{t(7)} = k^{t(8)})$
	EGC	r9	$W_1(t^{5m^{(k-1)}})$ && $W_1(erel^{12m^{(k)}})$
	default		randomState()
r_{11}	A	r2	O
	AM	r3	$W_1(1^{t(9)}, kr^{t(10)}, t^{7m^{(k-1)}})/W_2(e^{t(9)})$
	_AM	r4	$W_1(inc(m^{t(9)})/W_3(m^{t(9)} < k^{t(10)})$ || $(W_2(b^{7m}), W_2(b^{12m})$, changeState(r2))$/W_3(m^{t(9)} = k^{t(10)})$
	Ai	r5	O
	PGF	r1	$W_1(t^{3m^{(k-1)}})$ && $W_1(rel^{12m^{(k)}})$
	EGC	r9	$W_1(t^{5m^{(k-1)}})$ && $W_1(erel^{12m^{(k)}})$
	EGM	r3	$W_1(1^{t(1)}, k^{t(2)}, t^{10m^{(k-1)}})/W_2(e^{t(1)})$

	_EGM	r4	$W_1(inc(m^{t(1)})/W_3(m^{t(1)} < k^{t(2)})$ $\|\| (W_2(b^{10m}), changeState(r2))/W_3(m^{t(1)} = k^{t(2)})$ && $W_2(b^{12m})$
	EGA	r3	$W_1(1^{t(3)}, k^{t(4)}, t^{11m^{(k-1)}})/W_2(e^{t(3)})$
	_EGA	r3	$W_1(inc(m^{t(3)})/W_3(m^{t(3)} < k^{t(4)})$ $\|\| W_1(W_2(b^{11m})^{6m})$ && $W_1(erel^{12m^{(k)}})/W_3(m^{t(3)} = k^{t(4)})$
	Ake	r14	$W_2(b^{12m}) /c(b^{12}) > 0$
	default		randomState()
r_{12}	rel	r10	O
	erel	r11	O
	default		randomState()
r_{13}	*		
r_{14}	labelEGC	r9	$W_2(b^{5m}) /c(b^{5}) > 0$
	labelEGA	r9	$W_2(b^{6m}) /c(b^{6}) > 0$
	no_label	r13	*
	default		randomState()

Пример работы автомата представлен на рисунке 1.

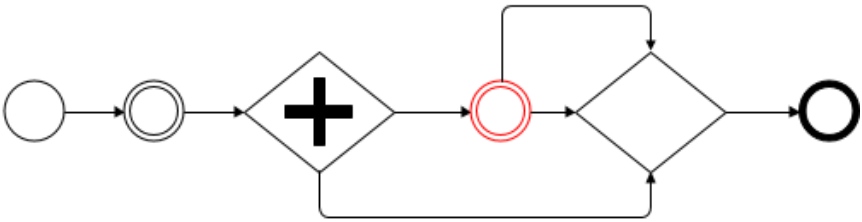


Рис. 1. Пример работы RV-автомата

При переходе с квазитерма rel в состояние r10 на 8 строке переходов случайным образом запускается негативная генерация, и автомат попадает на квазитерм AiF. Данный квазитерм невозможно получить для правильного языка. Выполняя необходимые операции над памятью, автомат генерирует ошибочную диаграмму.

Заключение

Указанный метод является более универсальным, а также позволяет генерировать негативные диаграммы динамически за один подход. Необходимо расширение текущего алфавита, а также перенос описываемого подхода на другие диаграммные языки.

Список литературы

1. Шаров О.Г., Афанасьев А.Н. Синтаксически-ориентированная реализация графических языков на основе автоматных графических грамматик // Программирование. – 2005. - № 6. – С. 56-66.
2. Шаров О.Г., Афанасьев, А. Н. Методы и средства трансляции графических диаграмм // Программирование. – 2011. – №3. – С. 65-76.
3. Афанасьев А.Н., Кириллов С.Ю., Савичева Ю.А. Разработка генератора негативных BPMN диаграмм на основе RV-грамматики // Настоящий сборник научных трудов конференции ИМАП-2016.

Научное издание

ИНФОРМАТИКА, МОДЕЛИРОВАНИЕ,
АВТОМАТИЗАЦИЯ ПРОЕКТИРОВАНИЯ
(ИМАП-2016)
VIII Всероссийская школа-семинар аспирантов,
студентов и молодых ученых
(Россия, г. Ульяновск, 25-26 октября 2016 г.)

Сборник научных трудов
Под редакцией А.Н. Афанасьева

ЛР № 020640 от 22.10.97.
Подписано в печать 01.12.2016. Тираж 50 экз.
Объем 16 Мб. Заказ ЭИ №1133.

Ульяновский государственный технический университет
432027, г. Ульяновск, ул. Сев. Венец, 32.
ИПК «Венец» УлГТУ, 432027, г. Ульяновск, ул. Сев. Венец, 32.