

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное бюджетное образовательное учреждение  
высшего образования

«УЛЬЯНОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

# **ИНФОРМАТИКА, МОДЕЛИРОВАНИЕ, АВТОМАТИЗАЦИЯ ПРОЕКТИРОВАНИЯ**

**(ИМАП-2017)**

IX Всероссийская школа-семинар аспирантов,  
студентов и молодых ученых  
(Россия, г. Ульяновск, 24-25 октября 2017 г.)

Сборник научных трудов

Ульяновск  
УлГТУ  
2017

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-2017). IX Всероссийская школа-семинар аспирантов, студентов и молодых ученых (Россия, г. Ульяновск, 24-25 октября 2017 г.): сборник научных трудов / под ред. А. Н. Афанасьева. – Ульяновск : УлГТУ, 2017. – 195 с.

В сборнике отражены материалы лекций ведущих ученых и докладов аспирантов, студентов и молодых ученых, представленные на IX Всероссийской школе-семинаре «Информатика, моделирование, автоматизация проектирования» (ИМАП-2017, г. Ульяновск, 24-25 октября 2017 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 978-5-9795-1707-0

© Коллектив авторов, 2017  
© Оформление. УлГТУ, 2017

## СОДЕРЖАНИЕ

<b>ЧАСТЬ I. МАТЕРИАЛЫ ЛЕКЦИЙ, ОРГАНИЗУЕМЫХ В РАМКАХ ПРОГРАММЫ ШКОЛЫ-СЕМИНАРА .....</b>	
П.И. Соснин ВОПРОСНО-ОТВЕТНЫЕ СЕТИ.....	7
В.Н. Негода ОЦЕНКА ЭФФЕКТИВНОСТИ РЕЗУЛЬТАТОВ МАГИСТЕРСКИХ РАБОТ В СФЕРЕ СОЗДАНИЯ АВТОМАТИЗИРОВАННЫХ СИСТЕМ.....	16
А.Н.Афанасьев, М.Е.Уханова, Н.Н. Войт ОБРАБОТКА КОНСТРУКТОРСКО-ТЕХНОЛОГИЧЕСКИХ ПОТОКОВ РАБОТ В УСЛОВИЯХ КРУПНОГО ПРЕДПРИЯТИЯ.....	27
<b>ЧАСТЬ II. МАТЕРИАЛЫ ДОКЛАДОВ УЧАСТНИКОВ ШКОЛЫ-СЕМИНАРА .....</b>	
А.Е.Абрамов АНАЛИЗ ПРОИЗВОДИТЕЛЬНОСТИ JAVA-SWING.....	44
А.Н. Афанасьев, Н.Н. Войт АНАЛИЗ ТЕМПОРАЛЬНЫХ ХАРАКТЕРИСТИК В ГИБРИДНЫХ ДИНАМИЧЕСКИХ ПОТОКАХ ПРОЕКТНЫХ РАБОТ.....	49
А.Н. Афанасьев, Н.Н. Войт, Р.Ф. Гайнуллин, С.Ю. Кириллов, В.А. Гордеев АНАЛИЗ СТРУКТУРНЫХ И СЕМАНТИЧЕСКИХ ОСОБЕННОСТЕЙ ДИАГРАММАТИЧЕСКИХ МОДЕЛЕЙ ДИНАМИЧЕСКИХ РАСПРЕДЕЛЕННЫХ ПОТОКОВ РАБОТ.....	56
А.Н. Афанасьев, Н.Н. Войт, В.С. Хородов ИССЛЕДОВАНИЕ МЕТОДОВ АНАЛИЗА И КОНТРОЛЯ ДИАГРАММАТИЧЕСКИХ МОДЕЛЕЙ ДИНАМИЧЕСКИХ РАСПРЕДЕЛЕННЫХ ПОТОКОВ РАБОТ.....	62
Т.В. Афанасьева, А.П. Курлов РАЗРАБОТКА И ОПИСАНИЕ СИСТЕМЫ КЛАССИФИКАЦИИ ВРЕМЕННЫХ РЯДОВ.....	68
Т.В. Афанасьева, А.Е. Полбин, Д.А.Максимов МОДЕЛИРОВАНИЕ КОМПЛЕКСНЫХ ЛИНГВИСТИЧЕСКИХ ОЦЕНОК КОМПОНЕНТ ОРГАНИЗАЦИОННО-ТЕХНИЧЕСКИХ СИСТЕМ С ИСПОЛЬЗОВАНИЕМ НЕЧЕТКИХ МНОЖЕСТВ.....	72

А.Д. Барт ВЫЯВЛЕНИЕ НАРУШЕНИЙ ПРИ МНОГОМЕРНОМ СТАТИСТИЧЕСКОМ КОНТРОЛЕ ПРОЦЕССА ОЧИСТКИ ПИТЬЕВОЙ ВОДЫ.....	76
С.И. Бочков ИССЛЕДОВАНИЕ САПР ЭЛЕКТРОННЫХ УСТРОЙСТВ DELTA DESIGN.....	82
С.И. Бочков ОСОБЕННОСТИ ГЕНЕРАЦИИ ЛАНДШАФТА В СРЕДЕ UNITY.....	87
С. И. Бригаднов АНАЛИЗ ЭКСПЕРТНЫХ СИСТЕМ.....	93
Е.Ю. Воеводин НЕЙРОННЫЕ СЕТИ В ЭЛЕКТРОННОМ ОБУЧЕНИИ АВТОМАТИЗИРОВАННОМУ ПРОЕКТИРОВАНИЮ.....	101
Н.Н. Войт, И.В. Беляева МАТЕМАТИЧЕСКОЕ МОДЕЛИРОВАНИЕ ПРОЦЕССА ЗАПОМИНАНИЯ МАТЕРИАЛА ПРОЕКТИРОВОЩИКОМ.....	106
Г.Ю. Гуськов, Т.В. Афанасьева, А.А. Романов РАЗРАБОТКА И РЕАЛИЗАЦИЯ АРХИТЕКТУРЫ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ДЛЯ ПРОГНОЗИРОВАНИЯ МЕТРИК ИТ ПРОЕКТОВ.....	112
А.В. Иванова ИССЛЕДОВАНИЕ СТАБИЛЬНОСТИ ВИБРАЦИЙ ГИДРОАГРЕГАТА С ПРИМЕНЕНИЕМ МЕТОДОВ СТАТИСТИЧЕСКОГО КОНТРОЛЯ.....	117
И.С. Ионова ИССЛЕДОВАНИЕ ПОТОКОВ РАБОТ ПО СОПРОВОЖДЕНИЮ ПРОИЗВОДСТВА СЛОЖНЫХ ТЕХНИЧЕСКИХ ИЗДЕЛИЙ.....	121
С.Ю.Кириллов, Н.Н.Войт, В.А.Гордеев РАЗРАБОТКА И ИССЛЕДОВАНИЕ ВРЕМЕННОЙ RV-ГРАММАТИКИ ДЛЯ АНАЛИЗА И КОНТРОЛЯ СТРУКТУРНЫХ ОСОБЕННОСТЕЙ ДИАГРАММАТИЧЕСКИХ МОДЕЛЕЙ ДИНАМИЧЕСКИХ РАСПРЕДЕЛЕННЫХ ПОТОКОВ РАБОТ.....	130
С.Ю. Кириллов, Н.Н. Войт, Р.С. Молотов, А.С. Степанов, Е.Ю. Воеводин, С.И. Бригаднов РАЗРАБОТКА И ИССЛЕДОВАНИЕ МЕТОДОВ АНАЛИЗА И КОНТРОЛЯ СЕМАНТИЧЕСКОЙ ЦЕЛОСТНОСТИ И СОГЛАСОВАННОСТИ ДИАГРАММАТИЧЕСКИХ МОДЕЛЕЙ ДИНАМИЧЕСКИХ РАСПРЕДЕЛЕННЫХ ПОТОКОВ РАБОТ НА ОСНОВЕ ВРЕМЕННОЙ RV-ГРАММАТИКИ.....	135

А.А. Куликова АВТОМАТИЗИРОВАННОЕ ИЗВЛЕЧЕНИЕ ПРИЧИННО-СЛЕДСТВЕННЫХ ОТНОШЕНИЙ ИЗ ТЕКСТОВ ПРОЕКТНЫХ ДОКУМЕНТОВ.....	140
С.А. Никулин РАЗРАБОТКА МОБИЛЬНОГО ПРИЛОЖЕНИЯ “ВЕЛОТРЕКЕР”.....	148
Е.П. Соснина, О.А. Утина ЛОКАЛИЗАЦИЯ ВЕБ-РЕСУРСОВ (НА ПРИМЕРЕ БАЗОВОГО ПРОЕКТА ПО ЛОКАЛИЗАЦИИ САЙТА ADVANTSHOP.NET).....	154
А.С. Степанов РАЗРАБОТКА РАБОЧЕГО МЕСТА СЛЕСАРЯ-СБОРЩИКА С РЕКОМЕНДАТЕЛЬНОЙ СИСТЕМОЙ НА ИГРОВОМ ДВИЖКЕ UNITY.....	158
А.Н. Стратонов ИСПОЛЬЗОВАНИЕ МОДЕЛИ БАЗЫ ПРЕЦЕДЕНТОВ ДЛЯ АВТОМАТИЗАЦИИ ПОИСКА РЕШЕНИЙ ДЛЯ ЗАДАЧ, ВОЗНИКАЮЩИХ В ПРОЦЕССЕ ПРОЕКТИРОВАНИЯ.....	163
И.А. Тимина РАЗРАБОТКА И ТЕСТИРОВАНИЕ КОНТРОЛЬНОГО ПРИМЕРА ДЛЯ РАЗРАБОТАННОГО МЕТОДА ПРОГНОЗИРОВАНИЯ.....	170
Д.Э. Цыганков, А.Ф. Похилько КОНСТРУКТОРСКОЕ ПОСТРОЕНИЕ 3D-МОДЕЛИ ИЗДЕЛИЯ В САД-СИСТЕМЕ.....	177
О.Э. Чоракаев МОДЕЛИРОВАНИЕ ПРОЕКТИРОВАНИЯ ИЗДЕЛИЙ СО СЛОЖНОЙ ГЕОМЕТРИЕЙ ДЛЯ ИЗГОТОВЛЕНИЯ НА 3D ПРИНТЕРЕ.....	182
Г.И. Шайдуллова МИКРОАНАЛИЗ ПРОИЗВОДИТЕЛЬНОСТИ СИСТЕМЫ ПРОВЕРКИ SQL-ЗАПРОСОВ.....	188
К.С. Ширкунова СТАТИСТИЧЕСКИЙ КОНТРОЛЬ МНОГОПАРАМЕТРИЧЕСКОГО ПРОЦЕССА ПРИ МАЛОМ ОБЪЕМЕ ДАННЫХ.....	192

# **ЧАСТЬ I**

## **МАТЕРИАЛЫ ЛЕКЦИЙ, ОРГАНИЗУЕМЫХ В РАМКАХ ПРОГРАММЫ ШКОЛЫ-СЕМИНАРА**

## ВОПРОСНО-ОТВЕТНЫЕ СЕТИ

П.И. Соснин<sup>1</sup>

В статье рассматриваются вопросно-ответные сети, регистрирующие результаты отображения операционного пространства проектирования на семантическую память. Основой таких сетей является дерево проектных задач в текущем состоянии их решения. В других версиях сетях представлены семантические модели проектировщиков и их групп, доступный опыт и другие компоненты операционного пространства. Важное место в семантической памяти занимает теория проекта и его концептуальное состояние, в том числе созданные документы.

### Введение

В статье основное внимание уделяется точке зрения на проект как на систему задач, работа с которой требует от проектировщиков их взаимодействия с доступным опытом в режиме реального времени. Эти взаимодействия являются основной причиной уникальности и основным источником новых непредсказуемых задач, возникающих в процессе проектирования.

Взаимодействия с доступным опытом имеет семантический характер. Поэтому их лучше отображать на семантическом уровне, как для контролируемого осуществления, так и для адекватной регистрации результатов.

В течение многих лет на кафедре «Вычислительная техника» УлГТУ исследовались процессы отображения и их результаты на концептуальном этапе проектирования автоматизированных систем (АС). Этот этап был выбран потому, что он непосредственно предполагает работу с семантическими (концептуальными) объектами, система которых материализуется в форме разрабатываемого концептуального проекта АС. Кроме того, на практике этот этап обычно является источником дорогостоящих семантических ошибок, обусловленных проблемами

---

<sup>1</sup> 432027, Ульяновск, ул. Северный Венец, 32, УлГТУ, e-mail: sosnin@ulstu.ru

персонального и коллективным пониманием, или, другими словами, этот этап подвержен негативному влиянию человеческих факторов, особенно когда проектировщики сталкиваются с новыми задачами [1].

По этим причинам в статье рассматриваются такие артефакты, как дерево проектных задач и окружающая их среда в состояниях их становления на концептуальной стадии проектирования определенной АС. В наших исследованиях освоено создание этих артефактов с использованием отображения операционного пространства проектирования на семантическую память вопросно-ответного типа [2]. Процесс и результаты отображений имеют унифицированные особенности, которые помогают использовать для артефактов общее название «вопросно-ответные сети» (QA-сети).

Такие сети (включая дерево задач) являются ценным источником информации, что приводит к новым положительным эффектам в обнаружении семантических ошибок и обеспечивает контролируемое достижение необходимых уровней персонального и коллективного понимания. Создание и использование QA-сетей осуществляется в специализированной инструментально-моделирующей среде WIQA (Working In Questions and Answers) [3].

## 1. Опыт создания и использования QA-сетей

В исследованиях возможностей использования контролируемого включения вопросно-ответных рассуждений (QA-рассуждений) проектировщиков в их работу на ранних этапах разработки АС был создан специализированный подход (вопросно-ответный подход, QA-подход) для концептуальной работы проектировщиков с проектными задачами. В своем применении к определенному проекту АС этот подход имеет следующие особенности:

1. Динамика QA-подхода представлена текущим состоянием дерева задачи  $TT(Z^*, t)$ , которое строится командой  $T^*$  проектировщиков  $\{D_{vs_i}\}$  с использованием поэтапного уточнения, применяемого к начальному оператору  $St(Z^*, t_0)$  корневой задачи  $Z^*$ .
2. Для каждого узла  $Z_i$ , представленного в дереве  $TT(Z^*, t)$  выражением  $St(Z_i, t)$  соответствующей задачи  $Z_i$ , конструктор (ответственный за эту задачу) выполняет анализ вопросно-ответный анализ (QA-анализ) в форме поэтапного уточнения, которое приводит к вопросно-ответной модели (QA-модель, QA( $Z_i, t$ )) этой задачи.



3. Дерево ТТ ( $Z^*$ ,  $t$ ) и результат QA-анализа для каждой задачи дерева записываются в специальной базе данных (вопросно-ответной базе, QA-базе) в иерархических формах, любая из которых визуальна доступна как в целом, так и на уровне её узлов (любой узел или группа узлов).
4. Задачи дерева ТТ ( $Z^*$ ,  $t$ ) распределяются между разработчиками команды, организационная структура которых также зарегистрирована в QA-базе в условиях, показанных на рисунке 1.

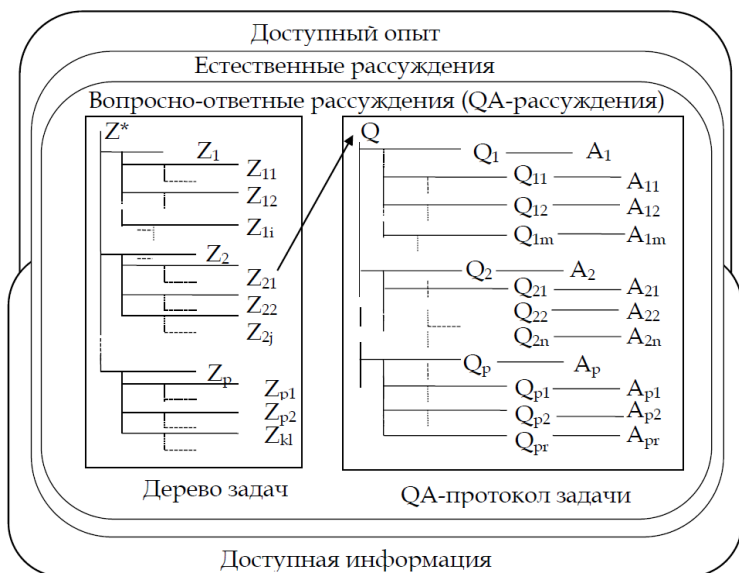


Рис. 1. Типовой результат QA-анализа

Схема подчеркивает, что вопросно-ответные структуры (QA-структуры) генерируются на основе естественных рассуждений проектировщиков, когда они конструктивно применяют QA-рассуждения при решении задач проекта. В этой работе проектировщики оперативно используют доступный опыт и соответствующую информацию. Проясним особенности QA-структур. Для определенного проекта  $P$  они представляют свое дерево задач ТТ( $Z^*$ ,  $t$ ) в текущий момент времени  $t$ . Другими словами, артефакт ТТ ( $Z^*$ ,  $t$ ) отражает состояние проекта  $P$  в

реальном времени с точки зрения его структуры задач [4]. Этот артефакт имеет следующие функции:

1. Артефакт формируется в семантической памяти инструментария WIQA.
2. Для любой задачи из  $TT(Z^*, t)$  отражение включает в себя вопросно-ответный протокол (QA-протокол,  $QA(t)$ ), который кодирует QA-рассуждения, используемые при решении этой задачи.
3. Все узлы  $TT(Z^*, t)$  визуальны доступны для проектировщиков.
4. Ячейки памяти помогают адекватно выразить семантику любой проектной задачи и любого компонента её QA-протокола.
5. Важное место среди атрибутов любой ячейки занимает поле, предназначенное для вербального описания загруженной единицы (описание для вопроса или ответа любого типа).
6. Проектировщики имеют возможность связывать группу узлов в структуре  $TT(Z^*, t)$  необходимым отношением.

## **2. Развитие и интерпретация QA-сетей**

Полезны две основных точки зрения на артефакт  $TT(Z^*, t)$ :

1. В соответствии с первым представлением артефакт имеет иерархическую структуру, которая формируется с использованием поэтапного уточнения.
2. Вторая точка зрения отражает структуру  $TT(Z^*, t)$  в условиях, когда проектировщики установили некоторые полезные отношения между узлами артефакта.

Таким образом, в поэтапном концептуальном проектировании дерево задач накапливает построенные и используемые описания вопросов и ответов любых типов. Более того, проектировщики имеют возможность проверять, корректировать, обрабатывать, изменять и использовать любое из этих описаний, если это будет необходимо или полезно. В своем текущем потенциале инструментарий WIQA поддерживает использование артефакта  $TT(Z^*, t)$  в приложениях, которые показаны на рисунке 2.

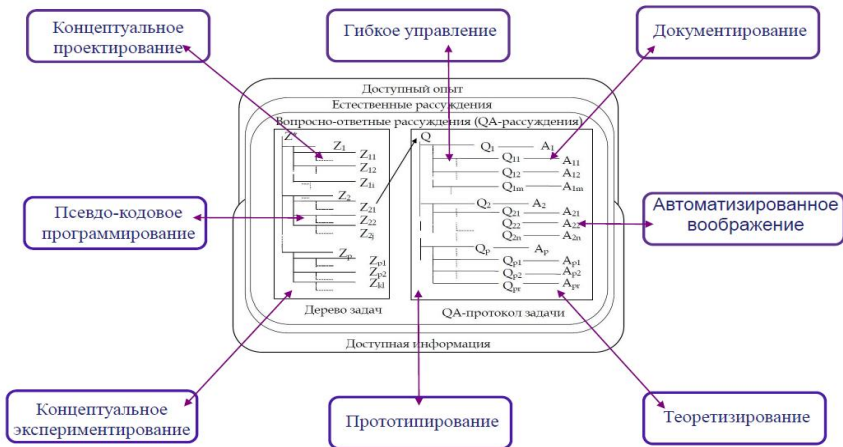


Рис. 2. Типовые применения QA-сетей

В любом из освоенных приложений происходит обращение к компонентам артефакта в соответствующих формах и интерпретациях. Это открывает возможность и право связывать любую из таких форм и ее интерпретацию с соответствующей версией QA-сети.

В текущем состоянии инструментарий WIQA поддерживает создание и использование, например, следующих версий QA-сетей:

1. Дерево задачи  $TT(Z^*, t)$ , некоторые особенности которого были описаны выше.
2. Организационная структура  $OS(T, t)$  команды проектировщиков  $T$ .
3. Онтология проекта  $OP(Z^*, t)$ , инструментальная поддержка которой встроена в ядро комплекса WIQA.
4. База моделей опыта  $BE(T, t)$  доступна для команды проектировщиков в их профессиональной деятельности.
5. База моделей прецедентов  $BP(Z^*, t)$ , созданных командой проектировщиков в текущем состоянии разработки определенной АС (с корневой задачей  $Z^*$ ).
6. Карточечная система для гибкого управления  $AM(Z^*, t)$ , ориентированная на использование механизмов Kanban и Scrum.
7. QA-сети для размещения псевдо-кодовых программ, включая их объединение в библиотеки.

### 3. Содержательно-эволюционная теория проекта

Среди новых приложений представим использование QA-сетей для теоретизирования в разработке проекта. С теоретической точки зрения [5] разработка любого АС-проекта - это уникальное явление, суть которого определяется, прежде всего, человеко-компьютерной деятельностью, которая реализуется командой проектировщиков, взаимодействующей с вовлеченными заинтересованными сторонами. В этом смысле теоретизация должна быть направлена на создание теории проекта, которая применяется, в основном, в процессе проектирования [6].

По этим причинам инструментарий WIQA поддерживает версию теоретизирования, ориентированную на создание для проектов теорий содержательно-эволюционного вида, типовая структура которых показана на рисунке 3.

Любая теория этого типа разрабатывается проектировщиками параллельно с другими видами деятельности и их продуктами, используемыми в процессе проектирования [7]. Теория проекта  $Th^P(t)$  является искусственным образованием, зависимость которого от времени  $t$  позволяет связать с этим образованием процесс становления теории, фазовая структура которого приведена на рисунке 3.

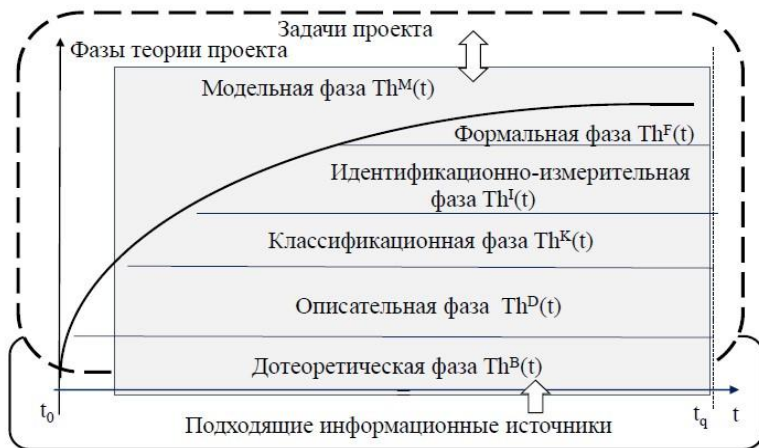


Рис. 3. Фазовая структура теории проекта

1. **Дотеоретическая фаза**  $Th^B(t)$  включает в себя сбор данных  $D(t)$  как набора фактов  $\{F_k\}$ , относящихся к соответствующей предметной области, а скорее к свойствам её сущностей и отношений между ними.
2. В **описательной фазе**  $Th^D(t)$  из зарегистрированных фактов конструируются тексты  $T(t) = \{T_i\}$ , связывающих факты в описании  $\{D_i=T_i\}$  сущностей, выделенных в предметной области, и эти конструкции уже вносят определенный порядок (системность) в совокупности фактов.
3. При построении теории особое внимание уделяется словарю и особенно развитию его части, представляющей систему понятий  $S^1(\{N_j\})$ , семантическое значение и порядок которых определяет определенный набор классификаций  $K=S^2(\{K_m\})$ . Разработка системы  $S^1(\{N_p\})$  на основе  $S^2(\{K_m\})$  является очень важной фазой становления теории. На рисунке 3 эта часть теории отмечена как **классификационная фаза**  $Th^C(\{K_m\})$ . В версии теории, формируемой в среде WIQA, эта фаза соответствует онтологии проекта [8].
4. В становление теории и ее использование, **идентификационно-измерительная фаза**  $Th^I(t)$  вводит возможность (эмпирической) интерпретации теоретических конструкций, в частности, возможность контролировать адекватность любого концептуального образования или понятия при их использовании в выбранном фрагменте описания или факта. Типичным подходом к идентификации является распознавание образов с использованием соответствующих средств, которые приводят к именам классов. Измерение помогает определить значения атрибутов для выделенных сущностей теоретизированной реальности. В нашем случае проектировщик может использовать автоматизированное умственное воображение и автоматизированные мысленные эксперименты.
5. **Формальная фаза** обычно выражается одним или несколькими формальными теориями  $Th^F(t) = \{Th^F_p\}$ , систематизирующими материал предыдущих фаз и переносящими решение задачи с уровня манипуляций с сущностями предметной области на уровень манипуляций с символическими конструкциями теории. Такой переход открывает возможность использовать вывод для предсказаний и применять доказательства для проверки предсказаний.
6. Следует отметить, что любая теория создается для ее использования, очень важным видом которого являются модели. Модели являются посредниками между теориями и реальностью, когда проектировщики

взаимодействуют с ними. Поэтому модели образуют полезную область теоретических приложений, которая показана на рисунке 3 как **модельная фаза**  $Th^M(t)$ .

Каждая АС является результатом работы, осуществляемой командой разработчиков, взаимодействующих с заинтересованными сторонами. В свою очередь, работа определяется как система действий, реализация каждого из которых приводит к соответствующему «продукту работы». В управлении проектами и программировании такие действия обычно называются задачами.

Как естественно-искусственные образования, задачи имеют следующие особенности:

1. Они ориентированы на достижение определенных целей, которые должны быть подтверждены полученными результатами и их проверками.
2. В общем смысле жизненный цикл задачи включает творческие действия, которые доводятся до их автоматизированного представления в повторных решениях задачи в будущем. В этих случаях задача должна быть решена, как правило, для повторного использования в будущем.
3. Описание задачи, которое обеспечивает повторное использование задачи, можно интерпретировать как модель соответствующего прецедента, представляющего единицу опыта.

Таким образом, любая решаемая задача  $Z_i$  оставляет текстовые и другие следы в теории проекта  $Th^P(t)$  как системе текстовых единиц  $S(\{T_i\}, t)$ .

### **Заключение**

В статье рассматривается отображение операционного пространства проектирования АС на семантическую память вопросно-ответного типа. Среди полезных результатов такого отображения необходимо отметить QA-сети, которые могут быть использованы для моделирования практически всех важных сущностей операционного пространства.

Исследованные версии QA-сетей демонстрируют, что это полезные источники ценной информации в основных составляющих процесса проектирования, таких как концептуальное проектирование, документирование, коммуникация, управление проектами, обучение и теоретизация.

В своих визуализированных формах эти сети выполняют роль посредника между проектировщиками и доступным опытом, включая в реализацию процессов естественно профессиональный опыт и его модели. QA-сети могут играть роль концептуального пространства, в котором для проектировщиков открывается возможность для таких действий, как автоматизированное умственное воображение и автоматизированное мысленное экспериментирование.

### Список литературы

1. Jacobson I., Ng P.-W., McMahon P., Spence I. and Lidman, S. (2012). The essence of software engineering: the SEMAT kernel, Queue, vol. 10 (10), pp. 1-12
2. P. Sosnin, "Precedent-Oriented Approach to Conceptually Experimental Activity in Designing the Software Intensive Systems," International Journal of Ambient Computing and Intelligence (IJACI) Vol. 7(1), 2016, pp. 69-93.
3. P. Sosnin "Conceptual Experiments in Automated Designing," Chapter in the book "Projective Processes and Neuroscience in Art and Design," Eds.: Rachel Zuanon, IGI-Global, 2016, pp. 155-181.
4. P. Ralph "Comparing two software design process theories," in Proceedings of the International Conference on Design Science Research in Information Systems and Technology, R. Winter, J.L. Zhao, and S. Aier, Eds. St. Gallen, Switzerland: Springer, 2010, pp. 139–153.
5. P.-W. Ng "Theory based software engineering with the SEMAT kernel: preliminary investigation and experiences." In Proceedings of the 3rd SEMAT Workshop on General Theories of Software Engineering., 2014, pp. 13-20.
6. D.I.K.I.K. Sjøberg, T. Dyba, Anda, B.C.D., and J.E., Hannay "Building Theories in Software Engineering. Guide to Advanced Empirical Software Engineering." Springer London. 2008. pp. 312—336.
7. D.E., Perry "Theories, theories everywhere." In Proceedings of the 5th International Workshop on Theory-Oriented Software Engineering, 2016.
8. P. Sosnin "An Application of Design Thinking in a Creation of a Project Theory Supporting a Conceptual Design of a Software Intensive System," In Proceedings of the 11th IEEE International Conference on Application of Information and Communication Technologies (AICT-2017), 2017, pp. 376-38

This work was supported by the Russian Fund for Basic Research (RFBR), Grant #15-07-04809a and the State Contract № 2.1534.2017/ПЧ

## ОЦЕНКА ЭФФЕКТИВНОСТИ РЕЗУЛЬТАТОВ МАГИСТЕРСКИХ РАБОТ В СФЕРЕ СОЗДАНИЯ АВТОМАТИЗИРОВАННЫХ СИСТЕМ

В.Н. Негода<sup>2</sup>

Рассматриваются механизмы оценки эффективности результатов магистерских исследований в условиях, когда созданные компоненты и подсистемы автоматизированных систем в своем большинстве находятся только на стадии отладки или альфа-тестирования.

### Введение

Главный вопрос к магистранту во время защиты выпускной работы: «Ну и что дают предложенные Вами методы и средства?». И в большинстве случаев ответ на этот вопрос не убедителен, более того, магистрант зачастую не владеет методами поиска убедительного ответа. Обусловлено такое положение дел тремя основными обстоятельствами.

Во-первых, образовательный стандарт магистерского направления «Информатика и вычислительная техника» предусматривает широкий спектр компетенций выпускника [1]. Это обстоятельство существенно сокращает время, которое магистрант может потратить на разработку реальных программных средств автоматизации. В фазе преддипломной практики и выполнения выпускной квалификационной работы он обычно разрывается между задачами оформления работы и отладки разработанных программ, функционирование которых должно подтвердить состоятельность проектно-аналитических результатов. Проведение достаточно объемной опытной эксплуатации в этих условиях чаще всего невозможно, а значит и невозможно получить достаточно убедительные оценки эффективности по результатам применения созданных средств автоматизации.

---

<sup>2</sup> 432027, Ульяновск, ул. Северный Венец, 32, УлГТУ, e-mail: [nvnulstu@gmail.com](mailto:nvnulstu@gmail.com)



Во-вторых, каждая дисциплина подчинена тому, чтобы обеспечить ряд компетенций и учебные задачи разных дисциплин обычно не связаны между собой. В то же время, задача предметной оценки экономического эффекта внедрения средств автоматизации обычно является междисциплинарной, поскольку позитивные эффекты и затраты многоаспектны.

В-третьих, большинство магистерских работ посвящено автоматизации деятельности людей и главный эффект от автоматизации заключается в повышении производительности труда. Оценка эффекта от производительности труда объективно является не простой, поскольку размер эффекта существенно зависит от многих факторов, характеризующих содержание и активность автоматизированных бизнес-процессов.

Наиболее зрелый процесс оценки эффективности строится на основе технологий имитационного моделирования и функционально-стоимостного анализа [2]. Автоматизация управления бизнес-процессами так или иначе сопряжена с бизнес-реинжинирингом, поскольку в ходе автоматизации бизнес-процессы меняются по своему содержанию в части аналитической работы менеджмента и принятия решений [3]. Проектные решения по автоматизации, материализованные в структурно-функциональных и поведенческих моделях бизнес-процессов, являются хорошей основой для интеграции трех видов проектных работ [4]:

- выбора наилучших решений автоматизации;
- функционально-стоимостного анализа автоматизируемых бизнес-процессов;
- имитационного моделирования.

Естественно, что в минимальном варианте оценка эффекта может базироваться на применении типовых экономических расчетов [5, 6]. В то же время, содержание учебных дисциплин, где организуется сравнительный анализ различных технических решений и изучается имитационное моделирование в части практических и лабораторных занятий целесообразно наполнять задачами оценки эффекта от автоматизации.

## **1. Использование типовых экономических расчетов для оценки ожидаемого экономического эффекта**

Наиболее простой механизм расчета экономического эффекта, активно используемый на практике, заключается в учете только основных затрат,

без их детализации. Следуя работе [6], изложим содержание основных расчетов.

Ожидаемый экономический эффект определяется по формуле:

$$\mathcal{E} = \mathcal{E}_r - E_n * K,$$

где  $\mathcal{E}_r$  – годовая экономия;

$E_n$  – нормативный коэффициент, принимаемый обычно равным 0.15;

$K$  – капитальные затраты на проектирование и внедрение, включая первоначальную стоимость покупных программных средств и сервисов.

Годовая экономия  $\mathcal{E}_r$  складывается из экономии эксплуатационных расходов и экономии в связи с повышением производительности труда пользователя:

$$\mathcal{E}_r = (P_{до} - P_{после}) + \Delta P_{пт},$$

где  $P_{до}$  и  $P_{после}$  – эксплуатационные расходы соответственно до и после внедрения разрабатываемой программы;

$\Delta P_{пт}$  – экономия от повышения производительности труда дополнительных пользователей.

Капитальные затраты на этапе проектирования  $K_k$  рассчитываются по формуле:

$$K_k = C + Z + M + H$$

где  $C$  – стоимость покупных программных средств и сервисов;

$Z$  – заработная плата специалистов на всех этапах проектирования и внедрения;

$M$  – затраты на использование ЭВМ на этапе проектирования и внедрения;

$H$  – накладные расходы на этапе проектирования и внедрения (80-120% от оплаты труда персонала).

В выше приведенной формуле могут фигурировать какие-то иные значимые прямые затраты, если они имеют место. В случае, когда разработку ведет сторонняя организация, капитальные затраты могут браться равными стоимости контракта на разработку.

Заработная плата специалиста рассчитывается по формуле:

$$Z = Z_d * T * (1 + A_c / 100) * (1 + A_n / 100)$$

где  $Z_d$  – дневная заработная плата разработчика на этапе проектирования;

$T$  – количество человеко-дней, затраченных на проектирование;

$A_c$  – процент отчислений на социальное страхование;

$A_n$  – процент премий.

Продолжительность работ определяет либо по нормативам, либо рассчитывают на основании экспертных оценок по эмпирической формуле:

$$T_0 = (3 * T_{\min} + 2 * T_{\max}) / 5$$

где  $T_0$  – ожидаемая продолжительность работ;

$T_{\min}$  и  $T_{\max}$  – соответственно наименьшая и наибольшая по мнению эксперта длительность работы.

Расчет затрат на машинное время:

$$M = t_d * C_d + t_n * C_n$$

где  $C_n$  и  $C_d$  – соответственно стоимости одного часа процессорного и дисплейного времени;

$t_d$  и  $t_n$  – соответственно процессорное и дисплейное время, необходимое для решения задачи (час).

Эксплуатационные расходы включают в себя следующее:

- Расходы на подготовку и сопровождение информации;
- Содержание персонала по обслуживанию комплекса технических средств;
- Расходы на функционирование программы;
- Расходы на содержание здания;
- Прочие расходы.

Первые две позиции могут рассчитываться как расходы на персонал по формуле, аналогичной приведенной выше формуле расчета зарплаты  $Z$ . При этом берутся зарплаты персонала, занятого эксплуатацией и трудоемкости соответствующих работ. Третья позиция рассчитывается аналогично затратам на машинное время  $M$  этапа проектирования. Четвертая и пятая позиции – по нормативам.

## **2. Потенциал уточнения оценок и наращивания эффекта**

Уточнение оценок и наращивание эффекта целесообразно базировать на интеграции проектного процесса и оценки экономического эффекта. Проектные решения содержат детальные спецификации

автоматизируемых процессов в формате различных диаграмм: диаграммы активности UML, спецификации бизнес-процессов BPMN (Business Process Model and Notation), событийные цепочки процессов eEPC (event-driven process chain), интегрированные спецификации IDEF. Не существует никаких принципиальных препятствия для разметки операций и подпроцессов оценками затрат времени и потребляемых ресурсов, что позволяет уточнять затраты по мере детализации проектных решений. В реализацию проектных решений достаточно легко встраиваются средства протоколирования, позволяющие фиксировать затраты времени в терминах работ и операций, фигурирующих в проектных диаграммах.

Наращивание эффекта целесообразно рассматривать как процесс модификации проектных диаграмм с целью уменьшения затрат времени исполнителей бизнес-процессов и потребляемых ресурсов. Иначе говоря, оценки эффекта должны играть активную роль в формировании проектных решений автоматизированной системы.

Указанный потенциал необходимо использовать, учитывая объективно существующие факторы сложности, основными из которых являются:

- Первоначальный расчет затрат времени и других ресурсов довольно трудоемок и в силу многообразия факторов, влияющих на эти затраты часто требует имитационного моделирования.
- Переход от моделей процессов, фигурирующих в диаграммах бизнес-процессов, к имитационным моделям требует квалификации и заметных усилий.
- Пакеты имитационного моделирования нужно выбрать, заполучить и установить.
- Руководства по системам имитационного моделирования содержат сотни страниц, а учебное время магистранта существенно ограничено.

Для снижения влияния выше указанных факторов сложности целесообразно использовать следующие организационно-методические решения:

- Накопление претендентов расчета затрат и имитационных моделей на сервере учебно-методической информации с организацией достаточно комфортного доступа всем студентам потока.
- Формирование информационных ресурсов, обеспечивающих продуктивную установку бесплатных студенческих (образовательных) версий пакетов имитационного моделирования:

AnyLogic Personal Learning Edition, GPSS World Student Version, MATLAB Student Version, The Arena Advantage for Students.

- Формирование программно-информационных ресурсов, поддерживающих быструю трансформацию учебных примеров, идущих в составе пакетов имитационного моделирования, в модели простейших бизнес-процессов.
- Формирование программно-информационных ресурсов с описанием исходных кодов имитационных моделей на языках программирования общего назначения. Модель простой системы массового обслуживания занимает порядка сотни операторов C++, что может быть легко интегрировано в прототипы программ автоматизации бизнес-процессов, например, через оператор `#ifdef SIMULATION`. Это позволяет ускорить реализацию моделей благодаря двум обстоятельствам: во-первых, студент использует хорошо освоенные средства программирования, во-вторых, существенно упрощается интерфейс между пространствами программных компонентов автоматизации и моделирования.
- Внедрение элементов оценки экономического эффекта в различные учебные дисциплины. Например, в дисциплине «Вычислительные системы» после работ по ускорению реализаций программных функций планируется работа по исследованию экономических эффектов, связанных с этим ускорением в различных условиях эксплуатации. В дисциплине «Технологии разработки программного обеспечения» наряду с работой, где исследуются решения по совершенствованию процесса разработки программ, планируется работа, в которой анализируются экономические аспекты такого совершенствования в различных условиях создания программного обеспечения.

Особое место в спектре учебных дисциплин магистратуры направления «Информатика и вычислительная техника» занимают дисциплины, нацеленные на формирование компетенций в сфере имитационного моделирования и планирования экспериментов. В этих дисциплинах максимальная результативность работы по оценке экономического эффекта от создаваемых в рамках магистерских исследований средств автоматизации достигается в случае, когда содержание дисциплин специальным образом ориентируется на задачи оценки экономического эффекта.

### **3. Ориентация изучения технологий имитационного моделирования на задачи оценки экономического эффекта**

Существенное влияние на процесс моделирование оказывает нацеленность проектно-аналитической задачи. Здесь следует выделить следующее:

- Моделирование в рамках технико-экономического обоснования проекта. При этом обычно создаются две модели: исходного автоматизируемого процесса и процесса, имеющего место после автоматизации. Предпроектное моделирование дает обоснование открытия финансирования, внутрипроектное позволяет улучшить соотношение «Затраты – Эффект» за счет выбора рациональных проектных решений, постпроектное моделирование дает оценку реально достигаемого эффекта от автоматизации, например, в формате годового экономического эффекта. Естественно, что в предпроектном моделировании в качестве базовых расчетных параметров используются нормативы, экспертные оценки и доступные информационные ресурсы, содержащие описания расчетов применительно к разработке программного и информационного обеспечения автоматизированных систем.
- Моделирование с целью оценить риски обеспечения параметрических ограничений, фигурирующих в техническом задании. Например, в случае ограничений реального времени сначала выделяется часть автоматизируемого процесса, для которой существуют риски. Затем строится модель временных затрат на операции автоматизируемого процесса, на основе которой уже строится имитационная модель. Для ее калибровки выполняется профилирование времязатратных программных функций. Затем формируется рабочая нагрузка на процесс и проводятся эксперименты с имитационной моделью. Ориентация на задачи экономического эффекта выполняется через включение в содержание соответствующих лабораторных работ
- Моделирование с целью выбора наилучшего проектного решения. Здесь либо создается несколько моделей – для каждого проектного решения своя модель, либо строится композитная модель, финишная операция которой обрабатывает ряды значений критериальных параметров для поддержки выбора.

Рассматриваемая ориентации изучения имитационного моделирования не меняет традиционной структуры этапов моделирования [7,8], поэтому порядок выполнения работ имеет такой вид:

- A. Формулировка проблемы и определение целей.
- B. Разработка концептуальной модели.
- C. Формализация имитационной модели.
- D. Программная реализация имитационной модели.
- E. Отладка и калибровка имитационной модели.
- F. Планирование экспериментов с имитационными моделями.
- G. Проведение экспериментов с имитационными моделями и анализ результатов.

Краткое содержание этапов работ рассмотрим применительно к случаю оценки эффекта от создания средств автоматизации.

#### ***A1. Описание автоматизируемого процесса***

Специфицируется автоматизируемый исходный процесс в тех условиях осуществления, для которых предполагается позитивный эффект от автоматизации. При этом выделяются критериальные параметры эффективности автоматизации (производительность труда, затраты времени персонала, объем материальных затрат и другие характеристики качества процесса), а также факторы, которые влияют на параметры качества исходного процесса.

#### ***A2. Описание создаваемых средств автоматизации***

Специфицируется функциональность создаваемых средств автоматизации и механизм ее влияния на характеристики качества автоматизируемого процесса.

#### ***A3. Определение целей и задач моделирования***

Формулируются цели и задачи имитационного моделирования, раскрывающие моделируемые сущности и замыслы экспериментов с моделями.

#### ***B1. Разработка концептуальной модели исходного автоматизируемого процесса***

Разрабатывается модель процесса с использованием наиболее подходящих средств, активно используемых при создании автоматизированных систем: диаграммы деятельности UML, структурно-функциональные диаграммы IDEF, событийно-функциональные

диаграммы eEPC и т.п. Элементы диаграмм дополнительно размечаются спецификациями параметров и функциональных зависимостей, влияющих на эффект.

### ***B2. Разработка концептуальной модели автоматизированного процесса***

Разрабатывается модель автоматизированного процесса с использованием тех же средств, что применены при разработке модели исходного процесса.

### ***C1. Формализация имитационной модели исходного автоматизируемого процесса***

Разрабатывается имитационная модель процесса терминах переменных и функциональных зависимостей, в которых фигурируют критериальные параметры и факторы, выделенные в п. А1. Сначала строится изоморфное отображение множества элементов спецификаций параметров и функциональных зависимостей, фигурирующих в разметке диаграмм бизнес-процессов, в множество элементов формальной спецификаций имитационной модели. Для модели определяются параметры генераторов случайных чисел, имитирующих рабочую нагрузку на автоматизируемый процесс и задержки исполнения рабочих операций, а также точки процесса, где фиксируются результаты, соответствующие критериальным параметрам.

### ***C2. Формализация имитационной модели автоматизированного процесса***

Разрабатывается модель автоматизированного процесса с использованием тех же средств, что применены при разработкой имитационной модели исходного процесса.

### ***D1. Разработка программной реализации имитационной модели исходного автоматизируемого процесса***

Реализация строится либо средствами специальных систем имитационного моделирования (GPSS Word, Eny Logic, MATLAB, Octave и т.п.), либо на языках общего уровня с использованием математических библиотек (Phyton, Java, C#, и т.п.).



## ***D2. Разработка программной реализации имитационной модели автоматизированного процесса***

Реализация строится с использованием тех же средств, что применены при программной реализации имитационной модели исходного процесса.

### ***E1. Отладка и калибровка имитационной модели исходного автоматизируемого процесса***

Отладка модели на контрольных примерах. Калибровка проводится с использованием данных экспертов и доступных результатов хронометража (протоколирования) операций исходного процесса.

### ***E2. Отладка и калибровка имитационной модели автоматизированного процесса***

Модели отлаживаются на контрольных примерах. Калибровка выполняется на основе экспертных оценок и результатов протоколирования хода автоматизированного процесса.

### ***F1. Планирование экспериментов с имитационной моделью исходного автоматизируемого процесса***

Разработка планов, содержащих спецификацию значений факторов, вовлекаемых в эксперименты, и механизмов их варьирования.

### ***F2. Планирование экспериментов с имитационной моделью автоматизированного процесса***

Аналогично предыдущему пункту, но для автоматизированного процесса.

### ***G1. Проведение экспериментов с имитационной моделью исходного автоматизируемого процесса***

Выполняется циклический процесс «Уточнение плана эксперимента – Организация рабочей нагрузки на моделируемый процесс – Проведение эксперимента – Анализ результатов».

### ***G2. Проведение экспериментов с имитационной моделью автоматизированного процесса***

Выполняется такой-же циклический процесс, как в предыдущем пункте.

### **Г3. Сравнительный анализ результатов**

Выполняется сравнительный анализ с целью характеристики поведения эффекта в пространстве значений факторов.

### **Заключение**

Для того, чтобы оценки экономического эффекта от создания средств автоматизации в магистерских работах были достаточно убедительными, целесообразно включать в спектр задач лабораторных исследований различных учебных дисциплин задачи перевода качественных и количественных эффектов от тех или иных технических решений в экономические показатели, а содержание дисциплин, связанных с имитационным моделированием и планированием экспериментов, специальным образом ориентировать на задачи получения оценок экономического эффекта.

### **Список литературы**

1. Федеральный государственный образовательный стандарт. Уровень высшего образования. Направление подготовки 09.04.01 Информатика и вычислительная техника. // URL: <http://fgosvo.ru/news/4/503>.
2. BUSINESS STUDIO. Имитационное моделирование деятельности. // URL: [www.businessstudio.ru/wiki/docs/current/doku.php/ru/simulation\\_fca](http://www.businessstudio.ru/wiki/docs/current/doku.php/ru/simulation_fca)
3. Ойхман Е.Г., Попов Э.В. Реинжиниринг бизнеса. Реинжиниринг организаций и и современные информационные технологии. – М.: Финансы и статистика., 2010. – 336 с.
4. Анисифоров А.Б., Анисифорова Л.О. Методики оценки эффективности информационных систем и информационных технологий в бизнесе. // Учебное пособие. – СПб.: 2014. – 97 с.
5. Тельнов Ю.В. Реинжиниринг бизнес-процессов. // Учебное пособие. – М.: ММИЭИФП, 2003, – 99 с.
6. Поддубный А. Расчет экономического эффекта от внедрения системы автоматизации. Antegra consulting. Новости/Экспертные мнения. // URL: [http://antegra.ru/news/experts/\\_det-experts/4/](http://antegra.ru/news/experts/_det-experts/4/)
7. Имитационное моделирование экономических процессов. Учебное пособие для слушателей программы eMBI. // Составитель: Н.Н.Лычкина – М: Государственный университет управления, 2005. – 164 с.
8. Шевченко Д.Н., Кравчяня И.Н. Имитационное моделирование на GPSS : учеб.-методич. пособие для студентов технических специальностей. – Гомель: БелГУТ, 2007. – 97 с.

## ОБРАБОТКА КОНСТРУКТОРСКО-ТЕХНОЛОГИЧЕСКИХ ПОТОКОВ РАБОТ В УСЛОВИЯХ КРУПНОГО ПРЕДПРИЯТИЯ<sup>3</sup>

А.Н.Афанасьев<sup>4</sup>, М.Е.Уханова<sup>5</sup>, Н.Н. Войт<sup>6</sup>

В статье рассматривается проблема проектирования конструкторско-технологических потоков работ в условиях крупного предприятия. В качестве примера рассматривалась авторская модель потока работ согласования конструкторской документации (КД) на основе сети Петри. Проведен анализ модели на возможные ошибки проектирования системы.

### Введение

В современном мире любое крупное предприятие, требует ускорения конструкторско-технологической подготовки производства для скорейшего вывода продукта на рынок, улучшения качества выпускаемой продукции и снижения себестоимости изготовления. Сегодня невозможно выполнить эти требования без внедрения технологии сквозного проектирования, которое основано на автоматизации этапов проектирования технической документации, систем управления проектами, электронного документооборота и организации единого информационного пространства. Для решения данных проблем разрабатывается много программных продуктов. Это разнообразные CAD\CAE, CAM\CAPP, PDM\PLM и ERP системы. Схема взаимоотношений между этими системами представлена на рис.1.

---

<sup>3</sup> Исследования поддержаны грантом Министерства образования и науки Российской Федерации, проект № 2.1615.2017/4.6.

Исследование выполнено при финансовой поддержке РФФИ и Правительства Ульяновской области в рамках научного проекта № 16-47-732152

Исследование выполнено при финансовой поддержке РФФИ в рамках научного проекта № 17-07-01417

<sup>4</sup> 432027, Ульяновск, ул. Сев. Венец, 32, УлГТУ, e-mail: a.afanasev@ulstu.ru

<sup>5</sup> 432027, Ульяновск, ул. Сев. Венец, 32, УлГТУ, e-mail:mari-u@inbox.ru

<sup>6</sup> 432027, Ульяновск, ул. Сев. Венец, 32, УлГТУ, e-mail:n.voit@ulstu.ru

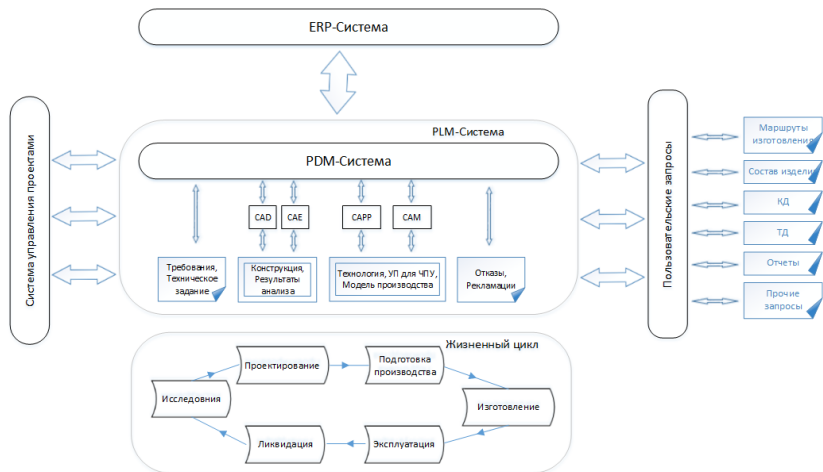


Рис. 1. Схема взаимоотношений CAD, CAM, PLM и ERP систем

Современные САПР включают в себя несколько взаимосвязанных компонентов. Прежде всего, это средства для проектирования конструкторской документации CAD (Computer Aided Design) и средства инженерного расчета CAE (Computer Aided Engineering). Типичными представителями CAD\CAE систем для российского рынка являются Компас-3D, T-Flex CAD, T-FLEX Анализ, Solidworks, Altium Designer, Autodesk Simulation, Unigraphics NX CAE и пр.

Несколько иными являются средства для проектирования технологической документации, которые включают в себя средства автоматизированной разработки техпроцессов CAPP (Computer Aided Production Planning) и средства автоматизированного производства – CAM (Computer Aided Mechanical). Типичными представителями CAPP систем для российского рынка являются Вертикаль, T-FLEX Технология, ADEM TDM, Sprut TP и пр., а CAM систем - Unigraphics NX, Siemens NX, CAM350 и пр.

PDM-системы (Product Lifecycle Management) выступают в качестве единого информационного пространства, в котором хранится разработанная техническая документация и электронная структура

изделия. Технология PDM позволяет управлять структурой изделия и проектами, автоматизировать получение отчетов, организовать многопользовательский доступ к документам в режиме реального времени.

PLM-системы (Product Data Management) представляют технологию управления жизненным циклом изделий. Это программное решение, которое управляет инженерными данными и информацией об изделии, а также обеспечивает управление всеми связанными с изделием процессами на всем протяжении жизненного цикла, начиная от проектирования и производства до завершения эксплуатации. В настоящее время разработано немало таких систем как российского, так и зарубежного производства: Лоцман-PLM, T-Flex-PLM, Siemens-PLM, Lotsia PLM и пр. Подобные PLM-системы очень эффективны для сквозного проектирования и позволяют производить интеграцию конструкторской разработки 3D модели изделия в технологическую подготовку производства, что приводит к уменьшению времени конструкторско-технологической подготовки производства (КТПП).

ERP-системы (Enterprise Resource Planning System) – это системы планирования и управления производством продукции. Такие информационные системы, используются для контроля и планирования ресурсов, которые применяются на предприятии, закупок и учета сырья, управления производством изделия, планирования заданий цехам. В качестве примера ERP-систем можно привести такие продукты, как Альфа, Lotsia ERP, 1C ERP, SAP.

Ускорение процесса согласования конструкторской и технологической документации основано на создании потоков работ. Внедрение на предприятии технологии потоков работ позволяет формализовать структуру и последовательность процедур прохождения документации. В настоящее время, практически у каждого производителя PLM-системы содержит встроенный блок управления потоками работ, который предназначен для моделирования рабочих процессов и автоматизации управления потоками заданий. Статистика бизнес-эффектов от внедрения технологии потоков работ представлена в таблице 1 [1-14].

Однако, системы управления жизненным циклом продуктов не всегда отвечают потребностям многих крупных предприятий. Поток работ в PLM-системах часто слишком тесно связан с потоком информации и с малой гибкостью для принятия альтернативных решений в режиме реального времени. Эти факторы всегда приводят к чрезмерно

продолжительным и дорогостоящим циклом внедрения workflow-систем, реинжиниринга процессов и реконфигурирования системы. Процессы реинжиниринга не только медленные и дорогостоящие, но часто и невозможные.

Таблица 1. Статистика бизнес-эффектов от внедрения технологии потоков работ сквозного проектирования

<b>Бизнес-эффекты</b>	<b>Доля улучшения</b>
Сокращение количества ошибок в технической документации	На 70 %
Сокращение длительности процессов КТПП	На 20-60 %
Сокращение затрат на подготовку и выпуск технической документации	На 40 %
Сокращение времени на поиск информации	На 40 %
Сокращение времени на согласование документации	В 1,5-7 раз
Сокращение времени вывода изделия на рынок	На 25-75 %
Сокращение доли брака	На 40 %

Более того, большинство workflow-систем позволяют моделировать потоки работ, но в них отсутствуют эффективные методы верификации диаграмматических моделей потоков работ и связанных с ними семантическими составляющими в виде текстов и программных модулей. Данные проблемы будут рассмотрены ниже.

Целью данной статьи является разработка модели согласования конструкторской документации в виде потока работ и анализа этой модели. В качестве основы представления, разработки и сопровождения использованы инструменты проектирования компании АСКОН и РЦ АСКОН-Волга.

## **1. Особенности конструкторско-технологических потоков работ на крупном предприятии**

Производство продукции на каждом предприятии имеет свои особенности [15-20]. Для крупных предприятий характерно:

- большое количество номенклатурных позиций для изготовления;
- сжатые сроки подготовки производства;
- наличие конструкторской документации, как собственной разработки, так и сторонних организаций;
- длительный цикл согласования КД и изменений;
- изготовление продукции по 2D-чертежам;

- большая организационная структура предприятия;
- сложное взаимодействие между подразделениями.

К сожалению, в крупных промышленных организациях, особенно производящих технику специально назначения, не удастся обойтись без 2D-проектирования. В этом случае конструкторами рисуется 3D-модель чисто формально, только для понимания конструкции. Технологи, получая документацию в 2D, разрабатывают собственные 3D-модели, чтобы описать технологию или разработать управляющую программу для станков с ЧПУ. Что приводит к увеличению времени разработки.

Следует отметить, что необходимо как можно ранее обнаружить ошибки на этапе проектирования. Как известно, чем позднее будет обнаружена ошибка, тем она дороже обойдется. Самые дорогие ошибки – это те, которые возникают на этапе производства. Но и на этапе разработки технологии изготовления изделия, ошибки в конструкции приводят к неминуемой задержке конструкторско-технологической подготовки производства, поскольку конструктору потребуется не только разработать изменение изделия, но еще его и согласовать, а это занимает немалое время.

Поэтому все больше возрастает тенденция сквозного проектирования с использованием 3D-моделей и систем управления потоками работ. Пример схемы движения 3D-модели шестерни от проектирования к технологии представлена на рис. 2 [14].

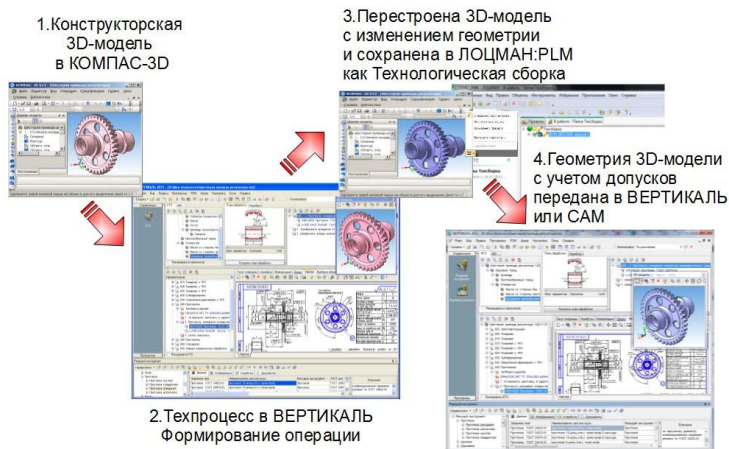


Рис. 2. Схема движения 3D модели от проектирования к технологии

## **2. Этапы проектирования конструкторско-технологических потоков работ**

Проектирование конструкторско-технологических потоков работ предлагается разбить на несколько этапов:

- формальное описание основных бизнес-процессов конструкторско-технологической подготовки производства;
- построение модели потока работ, основанном на формальном описании бизнес процессов;
- анализ свойств разработанной модели конструкторско-технологического потока работ;
- анализ ошибок обработки данных.

Ниже будет более подробно описаны этапы проектирования конструкторско-технологической подготовки производства.

## **3. Структурно-функциональная схема конструкторско-технологических потоков работ**

Одним из условий результативной работы предприятия является эффективное взаимодействие всех составляющих его подразделений и структур. Информационные потоки, отражающие суть производственного процесса, движутся по цепочке.

Для описания конструкторско-технологических потоков работ предприятия выделим основные процессы конструкторско-технологической подготовки производства. На рис. 3 представлена разработанная схема процессов конструкторско-технологической подготовки производства, которая является одним из примеров нормативных потоков проектных работ.

На схеме можно выделить два основных потока работ, отвечающие за конструкторские и технологические процессы подготовки производства. Эти процессы решают разные задачи, но нацелены на успешное освоение нового изделия. Рассмотрим их подробнее.

Конструкторские процессы потоков работ подготовки производства, включают в себя решение следующих задач:

1. Определение требований к новому изделию.
2. Анализ номенклатуры нового изделия.
3. Разработка графика выполнения работ по освоению или разработке нового изделия.
4. Анализ рекламаций от покупателей.



5. Определение требований для устранения неисправности.
6. Разработка и согласование технического задания на проведение работ.
7. Разработка и согласование КД.
8. Разработка и согласование изменений КД.

По завершению конструкторской подготовки производства начинается технологическая подготовка производства. Технологические потоки работ подготовки производства, включают в себя решение следующих задач.

1. Анализ возможности изготовления нового изделия.
2. Анализ оборудования для изготовления нового изделия.
3. Принятие решения о закупке оборудования.
4. Разработка графика выполнения работ по разработке технологии изготовления нового изделия.
5. Разработка маршрута изготовления номенклатуры состава изделия.
6. Разработка технологического процесса изготовления, в том числе и управляющих программ.
7. Определение трудоемкости изготовления изделия.
8. Нормирование основных материалов номенклатуры изделия.
9. Разработка технологической оснастки.

Нетрудно заметить, что потоки работ связаны между собой.

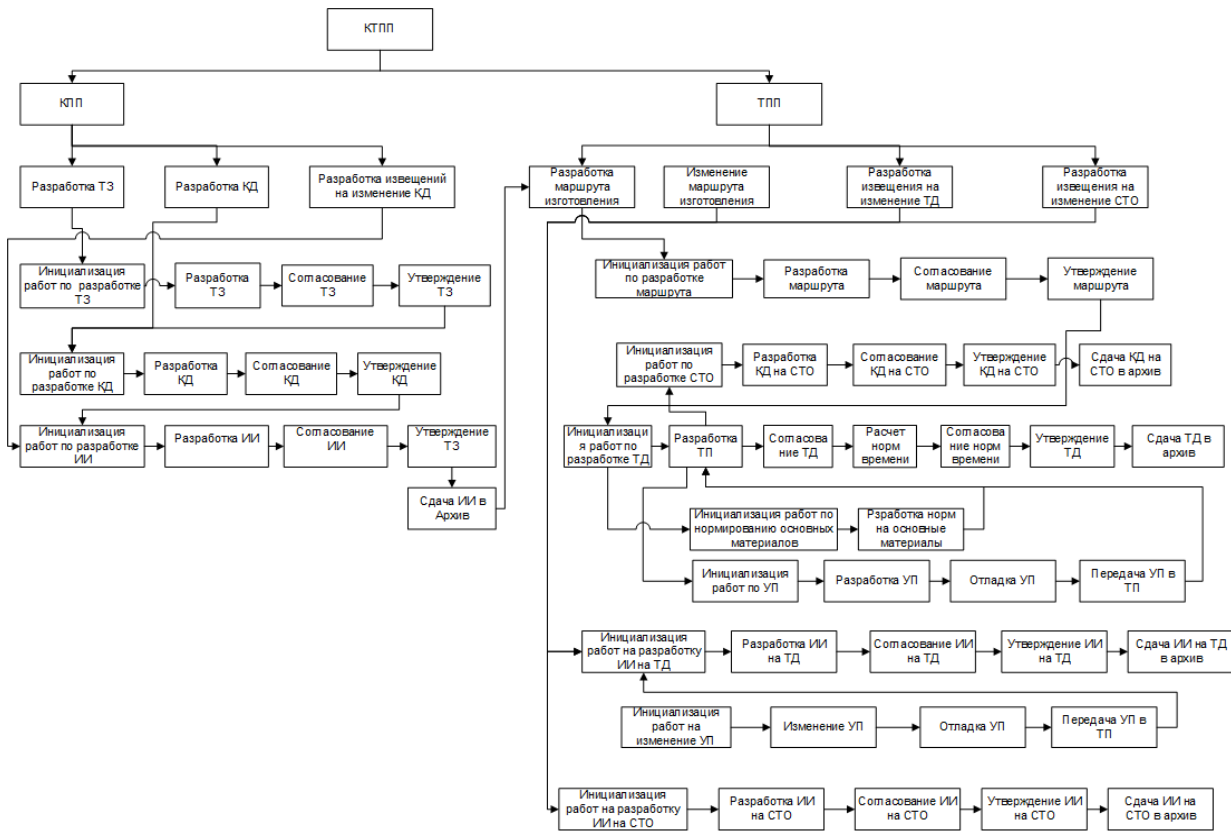


Рис. 3. Схема процессов конструкторско-технологической подготовки производства

#### **4. Моделирование типового потока работ согласования конструкторской документации**

Для построения модели конструкторско-технологических потоков работ выделим одну из задач конструкторской подготовки производства: рассмотрим типовую процессную модель согласования конструкторской документации детально.

Чтобы построить модель мы спроектировали бизнес-процесс разработки и согласования КД, определили правила формирования комплекта КД, выделили задания и определили их исполнителей. В качестве инструмента для проектирования модели потока работ, использовалось специализированное программное обеспечение Workflow Designer Системы управления проектами работ (СУПР), разработанное РЦ Аскон-Волга.

Мы построили последовательность прохождения заданий, разработали скрипты, для изменения состояний документов в процессе согласования и заполнения атрибутов согласования в Лоцман:PLM. На рис. 4 представлена модель потока работ разработки и согласования КД на специализированном языке СУПР.

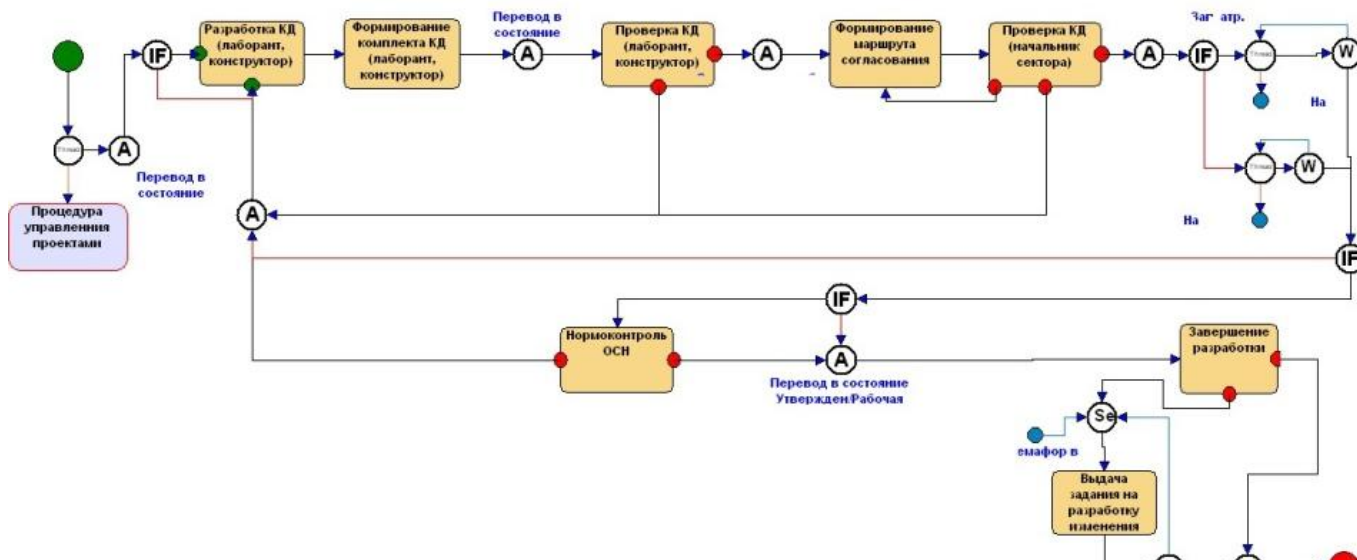


Рис. 4. Модель конструкторского согласования КД на специализированном языке компании АСКОН-Волга



$$m_i = -e,$$

где  $m_i$  – новый этап прохождения согласования,  $e$  – ошибки и замечания. Тогда весь процесс согласования КД можно рассмотреть в виде:

$$S = \sum_1^n m_i,$$

где  $S$  – процесс согласования КД,  $n$  – количество этапов согласования.

**Пример.** Начальник конструкторского сектора проверяет КД на наличие ошибок в документации, полноту комплекта документов для согласования, наличие согласующих подписей, необходимых для текущего этапа согласования. По результатам проверки, начальник сектора принимает решение: отдать документацию на доработку или отправить на следующий этап согласования. Формально запись выглядит так:

$$r = d \vee m_i \cup m_t,$$

где  $r$  – решение начальника сектора,  $d$  – доработка КД,  $m_i$  – очередной этап прохождения согласования,  $m_t$  – текущий этап прохождения согласования

## 5. Анализ свойств модели потока работ согласования КД

Разработанная модель согласования КД была проверена в специализированном программном обеспечении Visual Object Net++. Модель проверялась на следующие свойства:

- достижимость (reachability) – которое устанавливает, что конечное состояние системы будет достигнуто при любой последовательности переходов от позиции  $i$ . Данное свойство также подразумевает, что при достижении конечной позиции данной сети отсутствуют фишки в промежуточных позициях;
- безопасность (safety) – устанавливает, что в процессах отсутствуют зависания (deadlocks), заикливания, тупики;
- живость (liveness) – устанавливает, что система не содержит лишних позиций, которые никогда не будут выполнены. Отсутствие живости означает либо избыточность бизнес-процесса в проектируемой системе, либо свидетельствует о возможности возникновения заикливаний, тупиков, блокировок.

Проведенный анализ модели потока работ согласования КД, представленный на рис. 5, показал, что некоторые свойства сети зависят от квалификации проектировщика. Итак,

- модель обладает не очень хорошим свойством достижимости, поскольку возможно скапливание фишек в позиции  $m_1$ . Число фишек, которые могут скопиться в указанной позиции, зависит в первую очередь от квалификации конструктора, разработавшего КД. Чем ниже квалификация конструктора, тем больше количество раз документация будет возвращаться на доработку, а значит, тем большее количество фишек будет скапливаться в позиции  $m_1$ ;
- модель обладает хорошим свойством живости, поскольку отсутствует избыточность бизнес-процесса. Это свойство также зависит от квалификации конструктора. Чем выше квалификация разработчика КД, чем меньше «живость» потока работ, поскольку многие операции, связанные с доработкой КД не будут выполнены;
- модель обладает хорошим свойством безопасности, поскольку отсутствуют зависания, заикливание, тупики и блокировки.

В результате проведенного анализа был обнаружен парадокс свойства живости. Традиционно считается, что чем выше показатель свойства живости, тем лучше модель системы была разработана. Однако, в действительности, чем выше квалификация конструктора, тем меньше ошибок он совершает, и документация реже отправляется на доработку. Следовательно, это лучше для конструкторско-технологической подготовки производства. В результате блоки модели, отвечающие за отправку КД на доработку, используются реже. И именно это приводит к ухудшению свойства живости.

## **6. Анализ ошибок обработки данных**

Следует обратить внимание, что любой процесс согласования документации, как конструкторской, так и технологической содержит блоки программного кода, которые позволяют оперировать объектами, их состояниями и их атрибутами в процессе согласования. Например, смена состояния объекта в процессе согласования, для управления его жизненным циклом, или заполнения удостоверяющей подписи на этапах согласования.

Как известно, любой программный код может содержать ошибки. И чем больше строк программного кода, тем большее количество ошибок может допустить программист. Ошибки делятся на два вида – это синтаксические ошибки или семантические ошибки. Синтаксические ошибки выявить достаточно легко, это делают любые компиляторы программного кода или специально написанные unit-тесты. Большинство таких ошибок выявляются в момент написания программного кода. И если синтаксические ошибки находятся и исправляются достаточно легко и

быстро, то семантические ошибки выявить очень сложно, поскольку чаще всего они возникают на этапе выполнения приложения.

Семантические ошибки делятся на две группы.

#### 1. Логические ошибки.

Ошибки данного типа можно выявить, построив модель логики программного кода. Анализ модели позволит выявить тупиковые ситуации (deadlock), выполнимость программы. Этот вопрос рассматривается подробно во многих источниках, поэтому методы выявления таких ошибок в этой статье описываться не будут.

#### 2. Ошибки целостности данных.

Большинство ошибок данного типа можно избежать, если разработчик архитектуры информационной системы грамотно проектирует ограничения целостности данных на уровне баз данных: определяет, может ли значение атрибута быть пустым, устанавливает внешние ключи, накладывает ограничение уникальности, устанавливает проверку корректность введенного значения. В ряде случаев это не представляется возможным.

Чтобы выявить ошибки данного типа требуется разработка специального программного приложения, проверяющего возможности заполнения атрибута допустимыми значениями. В качестве метода определения таких ошибок предлагается использование Чеклиста, в котором перечислены все необходимые атрибуты процесса согласования. Например, при согласовании конструкторской документации заполняются перечень атрибутов у документа. Для корректной работы программы требуется разработать Чеклист, в котором будут перечислены перечень недопустимых ошибок.

Чеклист ошибок согласования конструкторской документации.

- проверка наличия указанных атрибутов у всех возможных типов конструкторской документации.
- проверка наличия обязательных для заполнения атрибутов.
- проверка наличия файлов документации.

## Заключение

Проанализированы особенности конструкторско-технологической подготовки производства продукции в условиях крупного предприятия. Разработана структурно-функциональная схема процессов типовых потоков проектных работ, а также определен перечень задач, решаемых в процессах конструкторско-технологической подготовки производства. Авторы разработали модели потока работ разработки и согласования конструкторской документации на базе специализированного языка РЦ Аскон-Волга и сетей Петри, а также произвели анализ модели потока



работ согласования КД, которая обладает свойствами живости, безопасности и достижимости. В результате работы был произведен анализ типов возникающих ошибок верификации диаграмматических потоков работ, связанных с ними семантическими составляющими в виде текстов и программных модулей. Будущие направления работ авторов будут связаны с разработкой методов исключения ошибок диаграмматических потоков работ, используя классификацию типов допустимых структурных ошибок.

### Список литературы

1. Афанасьев А.Н., Войт Н.Н., Бочков С.И., Уханова М.Е., Ионова И.С. Назработка и исследование виртуальных рабочих мест в среде OPENSIM // Вестник Ульяновского государственного технического университета. 2016. № 4 (76). С. 43-47.
2. Афанасьев А.Н., Войт Н.Н., Уханова М.Е., Ионова И.С., Епифанов В.В. Анализ конструкторско-технологических потоков работ в условиях крупного радиотехнического предприятия // Радиотехника. 2017. № 6. С. 49-58.
3. Карпов Ю. Г. MODEL CHECKING. Верификация параллельных и распределенных программных систем. – БХВ-Петербург, 2010.
4. Гончарук Ю. О. Проблемы создания и внедрения модели бизнес-процессов предприятия в форме Workflow System // Ресурсоэффективным технологиям-энергию и энтузиазм молодых: сборник научных трудов VI Всероссийской конференции, г. Томск, 22-24 апреля 2015 г.—Томск, 2015. – С. 263-272.
5. Awad A., Puhlmann F. Structural detection of deadlocks in business process models // Business Information Systems. – Springer Berlin Heidelberg, 2008. – С. 239-250.
6. Aguilar J. C. P. et al. Model Checking of BPMN Models for Reconfigurable Workflows // arXiv preprint arXiv:1607.00478. – 2016.
7. Janssen W. et al. Model checking for managers // International SPIN Workshop on Model Checking of Software. – Springer Berlin Heidelberg, 1999. – С. 92-107.
8. Kheldoun A., Barkaoui K., Ioualalen M. Specification and verification of complex business processes-a high-level petri net-based approach // International Conference on Business Process Management. – Springer International Publishing, 2015. – С. 55-71.
9. A.N. Afanasyev, N.N. Voit, M.E. Ukhanova, I.S. Ionova. Treatment design-engineering workflows in large enterprises.
10. И. Трохалиню. Лоцман: PLM 2017: Было и стало. Обзор новых возможностей в управлении данными об изделиях машиностроения. САПР и графика, май 2017.
11. И. Кочан. T-FLEX PLM на международном рынке. САПР и графика, май 2017.
12. П. Егоров. Проект "Развитие САПР" на базе Комплекса АСКОН. САПР и графика, апрель 2017.
13. П. Ведмидь, В. Власов. PLM системы менеджмента качества. САПР и графика, апрель 2017.
14. Сквозная 3D-технология АСКОН для предприятий машиностроения. ct3d.ru
15. 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 // В сборнике: 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.
16. Афанасьев А.Н., Войт Н.Н. Компонентная автоматизированная обучающая система САПР на основе гибридной нейронной сети // Автоматизация. Современные технологии. 2009. № 3. С. 14-18.

17. 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.
18. Афанасьев А.Н., Войт Н.Н. Реализация конструктора сценария обучающих курсов // Вестник Ульяновского государственного технического университета. 2011. № 1 (53). С. 54-59.
19. Афанасьев А.Н., Войт Н.Н. Разработка методов нечеткой параметрической адаптивной диагностики обучаемого инженера // Автоматизация процессов управления. 2009. № 3. С. 51-56. 11
20. 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.

## **ЧАСТЬ II**

### **МАТЕРИАЛЫ ДОКЛАДОВ УЧАСТНИКОВ ШКОЛЫ-СЕМИНАРА**

## АНАЛИЗ ПРОИЗВОДИТЕЛЬНОСТИ JAVA-SWING.

А.Е.Абрамов<sup>7</sup>

Одним из основных способов взаимодействия пользователя с программными функциями являются слушатели (англ. Listeners), которые добавляются в компонент интерфейса и реагируют на определенные события. Также, диспетчер ввода-вывода работает с разной скоростью в зависимости от ОС. Все эти факторы влияют на потребляемые ресурсы и увеличивают время исполнения программы, что может оказывать влияния на подсчет времени исполнения решения.

### Введение

Одним из важных критериев «юзабилити» любой программы является производительность приложения и пользователя, который использует данное приложение. В контексте конкурсных работ немаловажным критерием является способность оценивать время проверки каждого из заданий, присылаемых участником. Ведь по завершению турнира, часто необходимо знать, через сколько будут проверены все присланные решения и сформирована итоговая таблица турнира. Для решения этой проблемы необходимо определить основополагающие факторы, которые влияют на проверки тестовых заданий и вообще графического интерфейса пользователя (ГПИ) в библиотеке Java Swing.

### 1. Локализация проблемных мест

Одним из основных способов взаимодействия пользователя с программными функциями являются слушатели (англ. Listeners [1]), которые добавляются в компонент интерфейса и реагируют на определенные события. Они потребляют ресурсы системы и увеличивают время исполнения программы, что может оказывать влияния на подсчет времени исполнения решения, иными словами, их нужно учитывать при вычислении ограничения по памяти для проверяемого решения участника.

---

<sup>7</sup> 432027, Ульяновск, ул. Северный Венец, 32, УлГТУ

## Время обработки событий

В статье под названием «Measuring the Performance of Interactive Applications with Listener Latency Profiling» [2] был проведен анализ скорости передачи события при клике мышью. В указанной статье получились сравнительные значения, показанные на рис. 1. Для вычисления времени использовался обычный таймер.

```
...
dispatchStart 14 296380441352951
dispatchEnd   14 296380443289230
dispatchStart 14 296380665901043
listenerCall  14 296380666001335 java.awt.Toolkit$SelectableAWEventListener eventDispatched
listenerCall  14 296380666030878 java.awt.LightweightDispatcher eventDispatched
listenerReturn 14 296380666055671 java.awt.LightweightDispatcher eventDispatched
listenerReturn 14 296380666080605 java.awt.Toolkit$SelectableAWEventListener eventDispatched
listenerCall  14 296380666120135 ch.unisi.inf.performance.test.awt.MouseButtonLatencyCanvas$1 mouseReleased
listenerReturn 14 296381569377812 ch.unisi.inf.performance.test.awt.MouseButtonLatencyCanvas$1 mouseReleased
dispatchEnd   14 296381569494865
...
```

Рис. 1. Сравнительные значения времени вызова методов

Таким образом, исходя из полученных значений, время передачи события клика кнопкой мышью заняло около 2 мкс. Что сравнительно мало от общего количества времени выполнения, но довольно много при наличии большого количества действий с диалоговыми компонентами на экране. Схематично, работа метода на шкале времени изображена на рис. 2.

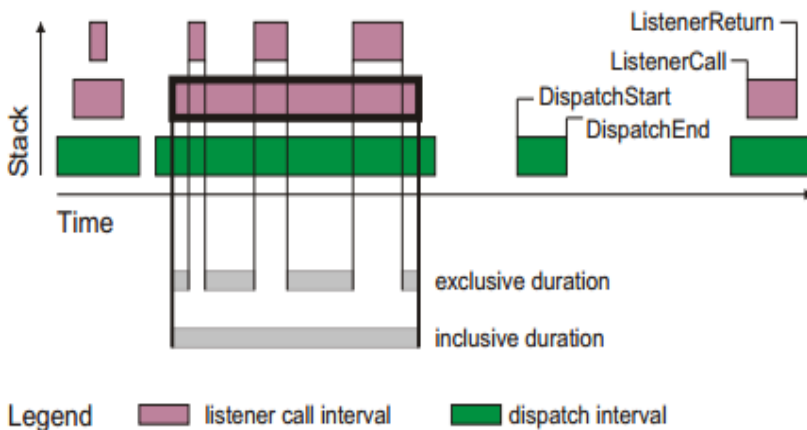


Рис. 2. Время выполнения события

## **Распределение чекера**

Проблема в том, что чекер проводит проверки последовательно и иногда можно распараллеливать задачи, необходимо выяснить, в каких случаях это возможно. Суть оптимизации в попытке представить задание и список компонентов в виде ориентированного графа, где вершинами являются компоненты-контейнеры. При этом, можно запускать параллельно несколько чекеров, что позволит ускорить время проверки. При таком подходе необходимо однозначно определять, есть ли в ветви модифицирующие компоненты и свойства, которые могут изменить поведение программы, так же, нужно определить поведение программы в случае ошибки (проверяем остальные ветви дальше или останавливаем процесс).

## **Эмуляция действий пользователя**

Одним из самых затратных факторов, влияющих на время проверки является эмуляция действий пользователя. Одной из причин этого является то, что настоящий пользователь не сможет за доли секунд производить действия с элементами интерфейса. Допустим, при разработке тех же анимаций переходов или времени инициализации компонентов, необходимо в чекере добавлять время ожидания перехода компонентов в работоспособное состояние. Некоторые события (hover), тоже требуют времени на проверку.

## **2. Формирование тестовой нагрузки**

В качестве задания для измерения времени выполнения всех проверок чекера будет использоваться задание «Калькулятор». В данном задании необходимо реализовать одноразрядный калькулятор, который будет поддерживать функции сложения, вычитания, умножения, деления и очистки. Пример реализации показан на рис. 3. Данный пример содержит в себе набор самых популярных диалоговых компонентов, и программа проверки содержит в себе 10 тестов свойств компонентов.

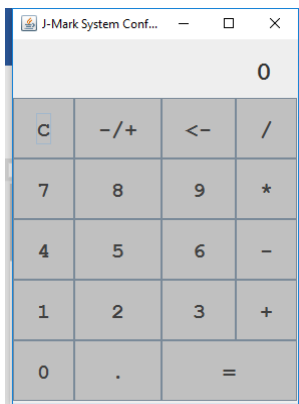


Рис. 3. Диалоговое окно калькулятора

### 3. Измерение затрат времени

В табл. 1 показано время, затраченное на проверку. Можно заметить, что действия для эмуляции действий пользователя занимают больше всего времени. Также много времени занимает инициализация диалогового окна JFrame.

Таблица 1. Время проверки множества компонентов

Компонент-свойство	Время проверки, мсек
JButton-Click	88
JButton-Color	33
JButton-Text	35
JLabel-Text	11
JLabel-Color	9
JTextField-Color	8
JTextField-Text	14
...	...
Input text	726
Move mouse	133
Click	242
JPanel-Background	13
JFrame	847

Время проверки задания зависит от множества факторов, таких как: тип компонента, сложность компонента, вид свойства, которое необходимо проверить, свойства проверяющей системы. При разработке

задания целесообразно эмпирическим путем выяснять время проверки всего задания на конкретной машине.

Целесообразно было бы оптимизировать время эмуляции действий пользователя.

## **Заключение**

Исходя из полученных данных в сравнительных таблицах, полученных опытным путем, становится понятно, что время проверки задания зависит от множества факторов, таких как: тип компонента, сложность компонента, вид свойства, которое необходимо проверить, свойства проверяющей системы. При разработке задания целесообразно эмпирическим путем выяснять время проверки всего задания на конкретной машине.

Проверка свойств компонентов занимает минимальное время, которым всецело можно пренебречь. Также, можно заметить, что наибольшее количество времени при проверке занимают компоненты-контейнеры и эмуляция действий пользователя. В особенности, инициализация класса диалогового окна JFrame

Целесообразно было бы изучить возможность и реализовать проверку каждого JFrame параллельно, как предложено в последнем подзаголовке первой главы. Также, необходимо изучить возможность ускорения эмуляции действий пользователя без захвата управления над мышкой и клавиатурой, т.к. это занимает очень много времени и ресурсов.

## **Список литературы**

1. Шилдт Г. Swing: Руководство для начинающих //Издательский дом" Вильямс. – 2007.
2. Jovic M., Hauswirth M. Measuring the performance of interactive applications with listener latency profiling //Proceedings of the 6th international symposium on Principles and practice of programming in Java. – ACM, 2008. – С. 137-146.



## АНАЛИЗ ТЕМПОРАЛЬНЫХ ХАРАКТЕРИСТИК В ГИБРИДНЫХ ДИНАМИЧЕСКИХ ПОТОКАХ ПРОЕКТНЫХ РАБОТ<sup>8</sup>

А.Н. Афанасьев<sup>9</sup>, Н.Н. Войт<sup>10</sup>

Сроки исполнения, длительность выполнения, очередность проектных процедур являются темпоральными параметрами при информационном сопровождении жизненного цикла изделий в гибридных динамических потоках проектных работ, которые предлагается структурно-параметрически анализировать с помощью временной автоматной RVT-грамматики с магазинной памятью и семантически анализировать с помощью формализации спецификаций данных потоков проектных работ в базисе темпоральных операторов "До", "В течение", "После".

### Введение

Потоки проектных работ являются мощным инструментом анализа бизнес-процессов предприятия и содержат проектные задачи, которые адресованы к конкретным подразделениям и исполнителям предприятия. Такие проектные работы могут производиться одновременно в разных структурных подразделениях и разными исполнителями, поэтому вопросы синхронизации, блокировки ресурсов, тупиков (deadlocks), узкого горлышка (bottlenecks) и т.п., встречающиеся в теории информатики, возникают и в области управления бизнес-процессами предприятия. Такие потоки проектных работ можно представить в темпоральном базисе "До", "В течение", "После" поставив в соответствие параметру срок временной регламент исполнения работ предприятия.

Как правило, системы управления потоками проектных работ представлены фреймворками, которые обеспечивают аккумуляцию,

---

<sup>8</sup> Исследования поддержаны грантом Министерства образования и науки Российской Федерации, проект № 2.1615.2017/4.6.

Исследование выполнено при финансовой поддержке РФФИ и Правительства Ульяновской области в рамках научного проекта № 16-47-732152

Исследование выполнено при финансовой поддержке РФФИ в рамках научного проекта № 17-07-01417

<sup>9</sup> 432027, Ульяновск, ул. Северный Венец, 32, УлГТУ, e-mail: a.afanasev@ulstu.ru

<sup>10</sup> 432027, Ульяновск, ул. Северный Венец, 32, УлГТУ, e-mail: n.voit@ulstu.ru

диспетчеризацию и представление потоков проектных работ в виде диаграмматических моделей таких как BPMN, UML, IDEF0, WPDL и т.п. Подобные системы разработаны по разным парадигмам [1], их методы анализа потоков проектных работ делятся на обеспечивающие (моделирующие) и не обеспечивающие (не моделирующие) имитацию бизнес-процессов во времени [2]. К не моделирующим схемам относятся диаграммы активности UML, WPDL, BPMN, в которых нет механизмов и свойств для анализа. К моделирующим (динамическим) – сети Петри (как наиболее яркий представитель),  $\pi$ -исчисления ( $\pi$ -calculus). В настоящее время  $\pi$ -исчисление является перспективной, но еще очень молодой и развивающейся теорией, в ней много открытых вопросов и нерешенных проблем. Сети Петри имеют следующие ограничения: нет универсального фрейворка для моделирования и анализа потоков проектных работ на базе сетей Петри; нет механизма, который помог бы проектировщику при моделировании и обеспечил успешное завершение задачи с необходимыми требованиями (свойствами).

Достаточно широкое применения анализа потоков работ при разработке безошибочных систем на этапе концептуального проектирования нашел метод Model Checking. Однако, он предназначен для опытных ученых и инженеров, так как сложен в понимании и оперировании, а также анализирует модель, а не объект исследования [2].

При проектировании и разработке сложных автоматизированных систем разработчики выделяют оркестрацию (orchestration) и хореографию (choreography) в архитектуре системы. Языками моделирования для описания оркестровки выступают BPEL (Business Process Executive Language), XPD (XML Process Definition Language), UML (Unified Model Checking). Соответственно, языками моделирования для описания хореографии выступают WS-CDL (Web-services Choreography Description Language) и ebXML (Electronic Business using eXtensible Markup Language). Можно констатировать, что в проектной практике используется множество различных (гибридных) языков описания бизнес-процессов в оркестровке и хореографии, а также при моделировании, анализе, контроле и оптимизации потоков работ предприятия.

Динамические системы управления потоками проектных работ, такие как ARIS Business Architect (IDS Scheer), LSPS Agile Change Management Application (WhiteStein Technologies), WBI Modeler (IBM), учитывают темпоральные особенности бизнес-процессов предприятия и представляют их в виде параметров в BPMN-, eEPC-моделях. Однако, они не имеют методов анализа темпоральных характеристик в этих диаграмматических моделях в плане сроков исполнения, длительности выполнения и очередности проектных процедур.

Таким образом, в работе разработана автоматная RVT-грамматика с магазинной памятью и метод спецификации гибридных динамических потоков проектных работ на базе темпоральной алгебры, обеспечивающие анализ гибридных динамических потоков проектных работ при информационном сопровождении жизненного цикла изделия на крупном проектном предприятии.

## **1. Анализ состояния научных исследований в области управления производством**

Анализом и контролем потоков проектных работ занимаются такие ученые как Афанасьев А.Н. [3-8], Карпов Ю.Г. [9], Соснин П.И., Лифшиц Ю. [10], Ярушкина Н.Г., Калянов Г.Н. [11], Курейчик В.В., Бурдо Г. Б., Конев Б. Ю., Шалыто А. А., Савенков К., Кулямин В. В., Neda Saeedloei, Gopal Gupta [12], Кларк Э.М., Буч Г., Yuan Wang, Yushun Fan [13].

Карпов Ю.Г. в работе [9] представляет подход Model Checking к анализу, контролю, моделированию и реинжинирингу бизнес-процессов, в котором главным недостатком является исследование модели, а не самой системы, поэтому возникает вопрос об адекватности модели к системе, а также сложность решения перечисленных выше задач является экспоненциальной.

В работе Neda Saeedloei and Gopal Gupta [12] применен темпоральный автомат реализующий темпоральную контекстно-свободную грамматику  $\omega$ -grammars к анализу кибер-физических систем с последующим переводом этой грамматики в ко-индуктивное логическое программирование для интерпретатора Prolog.

Yuan Wang and Yushun Fan [13] предлагают использовать темпоральную логику действий для описания потоков работ в графовой форме, что требует описание всех маршрутов графа в формулах темпоральной логики действий и доказательства от противного предположения. Используется линейная темпоральная логика к формализации маршрута из задач, разветвлений AND, OR и схождения JOIN, однако, вопрос адекватности построения описания потоков работ в графовой форме остается не решенным.

База инструментальных средств анализа и контроля потоков проектных работ доступна по адресу [14]. Кроме этого, имеются инструменты CPN Tools [15], «Roméo - A tool for Time Petri Nets analysis» [16], TimesTool [17], Tina Toolbox [18], Visual Object Net++ [19].

## 2. Темпоральная автоматная RVT-грамматика с магазинной памятью

RVT-грамматикой языка  $L(G)$  называется упорядоченная восьмерка непустых множеств  $G = (V, \Sigma, \tilde{\Sigma}, C, E, R, T, r_0)$ , где

- $V = \{v_e, e = \overline{1, L}\}$  – вспомогательный алфавит (алфавит операций над внутренней памятью);
- $\Sigma = \{a_l, l = \overline{1, T}\}$  – терминальный алфавит графического языка, являющийся объединением множеств его графических объектов и связей (множество примитивов графического языка);
- $\tilde{\Sigma} = \{\tilde{a}_t, t = \overline{1, \tilde{T}}\}$  – квазитерминальный алфавит, являющийся расширением терминального алфавита. Алфавит  $\tilde{\Sigma}$  включает: квазитермы графических объектов; квазитермы графических объектов, имеющих более одного входа; квазитермы связей-меток с определенными для них семантическими различиями. Связью-меткой является каждая связь, исходящая из графического объекта, содержащего более одного входа или выхода. Допустимо одну из связей такого графического объекта не обозначать как связь-метка. Механизм меток используется для возврата к не проанализированным частям графического образа; квазитерм, определяющий отсутствие связей-меток. Квазитерм *no\_label*, определяющий отсутствие связей-меток, используется для прекращения анализа диаграммы при условии выполнения операции(ий) над внутренней памятью;
- $C$  – множество идентификаторов часов;
- $E$  – множество темпоральных отношений “До”, “В течение”, “После” (инициализация часов  $\{c := 0\}$ , отношения вида  $\{c \sim x\}$ , где  $c$  переменная (идентификатор часов),  $x$  является константой,  $\sim \in \{=, <, \leq, >, \geq\}$ );
- $R = \{r_i, i = \overline{0, I}\}$  – схема грамматики  $G$  (множество имен комплексов продукций, причем каждый комплекс  $r_i$  состоит из подмножества  $P_{ij}$  продукций  $r_i = \{P_{ij}, j = \overline{1, J}\}$ );
- $T \in \{t_1, t_2, t_3, \dots, t_n\}$  является множеством временных меток, причем есть отображение  $FT\Sigma: \Sigma \times C \times E \rightarrow T$  и соответственно  $F\tilde{T}\tilde{\Sigma}: \tilde{\Sigma} \times C \times E \rightarrow T$ .
- $r_0 \in R$  – аксиома RVT-грамматики (имя начального комплекса продукций),  $r_k \in R$  – заключительный комплекс продукций.

Продукция  $P_{ij} \in r_i$  имеет вид  $a_l^{[t_i]} \xrightarrow{W\gamma(\gamma_1, \dots, \gamma_n)} r_m$ , где

- $W_v(\gamma_1, \dots, \gamma_n)$  – n-арное отношение, определяющее вид операции над внутренней памятью в зависимости от  $v \in \{0,1,2,3\}$ ;  
 $r_m \in R$  – имя комплекса продукции-приемника.

### 3. Анализ темпоральных характеристик гибридных динамических потоков проектных работ

Спецификация свойств гибридных динамических потоков проектных работ описывает требования к поведению системы (исполнителя) и может быть задана формулой темпоральной алгебры в виде дизъюнктивно-нормальной формы следующим образом:

$$\varphi ::= p | \neg \varphi | \varphi \vee \varphi | \varphi \circ \varphi | \varphi \cup \varphi, \quad (1)$$

где  $p$  – темпоральный атомарный предикат (atomic predicate),  $\neg$  – отрицание,  $\vee$  – логическое ‘ИЛИ’, оператор  $\circ$  – темпоральный оператор NextTime (X),  $\cup$  – темпоральный оператор Until. Причем, темпоральный атомарный предикат  $p$  принимает значение ‘0’ или ‘1’ в конкретные временные интервалы и, в отличие от предиката логики первого порядка, не зависит от своей структуры.

*Пример.* Начальник сектора конструкторов проверяет конструкторскую документацию (КД) к изделию на наличие ошибок, принимает решение: отдать ее на доработку при наличии ошибок или, в случае отсутствия ошибок, передать инженерам-конструкторам в отдел конструкторского бюро – ОКБ (Design); а начальник сектора лаборатории проверяет конструкторскую документацию (КД) к изделию на наличие ошибок, принимает решение: отдать ее на доработку при наличии ошибок или, в случае отсутствия ошибок, передать лаборантам в Лабораторию (Laboratory). Формальная запись выглядит так:

$$\varphi = \text{Design} \vee \text{Laboratory} \cup (\text{No Errors in Problem Definition}), \quad (2)$$

где Design – темпоральный атомарный предикат: признак передачи КД в ОКБ, Laboratory – темпоральный атомарный предикат: признак передачи КД в Лабораторию, (No Errors in Problem Definition) – темпоральный атомарный предикат: признак отсутствия ошибок в КД.

Для дальнейшего упрощения темпоральных алгебраических выводов определим  $\varphi$  через темпоральный оператор  $\circ$  как:

$$\varphi = NE \vee (D \vee L) \wedge \circ \varphi, \quad (3)$$

где NE – No Errors in Problem Definition, D – Design, L – Laboratory.

Используя теорему [9] о том, что для любого логико-алгебраического выражения можно построить конечный автомат с памятью, следовательно,

можно построить конечный автомат с памятью также для  $\varphi$ . Предложим следующую теорему.

*Теорема “Семантический анализ”.* Содержание эластичной ленты RVT-грамматики, построенной по  $\varphi$ , должно является подмножеством содержания  $(v_1, \dots, v_n)$  RVT-грамматики, построенной к диаграмматической модели, иначе в гибридных динамических потоках проектных работ существует семантическая ошибка, т.е.

$$\exists \varphi \rightarrow w_2(v_1, \dots, v_n) / w_3(v_1, \dots, v_n \neq \emptyset) \quad (3)$$

Доказательство.

Используем метод доказательства от обратного, докажем, что в таком случае у нас будет две разных грамматики. Пусть содержание эластичной ленты RVT-грамматики к  $\varphi$  не является подмножеством содержания эластичной ленты RVT-грамматики к диаграмматической модели. Следовательно, существует продукция в RVT-грамматики к  $\varphi$  отличная от продукций в RVT-грамматики к диаграмматической модели, что приводит к неэквивалентности двух грамматик. Что и требовалось доказать.

Таким образом, семантический анализ гибридных динамических потоков проектных работ реализуется с помощью определения не пустого множества пересечения содержания эластичных лент двух грамматик – RVT-грамматики к  $\varphi$  и RVT-грамматики к диаграмматической модели.

## Заключение

Разработаны темпоральная автоматная RVT-грамматики с магазинной памятью, обеспечивающая структурно-параметрический анализ временных характеристик гибридных динамических потоков проектных работ, метод семантического анализа темпоральных характеристик гибридных динамических потоков проектных работ, использующий спецификацию гибридных динамических потоков проектных работ в виде темпоральной алгебры.

В будущих работах будет рассмотрен класс структурно-параметрических и семантических ошибок гибридных динамических потоков проектных работ.

## Список литературы

1. D. Georgakopoulos, M. Hornick, and A. Sheth, An overview of workflow management: From process modeling to infrastructure for automation. Journal on Distributed and Parallel Database Systems, 3(2): 119-153, April 1995.
2. Yuan Wang, Yushun Fan Using Temporal Logics for Modeling and Analysis of Workflows // Proceedings of E-Commerce Technology for Dynamic E-Business, 2004. IEEE International Conference on, 2004. DOI: 10.1109/CEC-EAST.2004.72
3. 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 // В сборнике: 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.
4. Афанасьев А.Н., Войт Н.Н. Компонентная автоматизированная обучающая система САПР на основе гибридной нейронной сети // Автоматизация. Современные технологии. 2009. № 3. С. 14-18.
  5. 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.
  6. Афанасьев А.Н., Войт Н.Н. Реализация конструктора сценария обучающих курсов // Вестник Ульяновского государственного технического университета. 2011. № 1 (53). С. 54-59.
  7. Афанасьев А.Н., Войт Н.Н. Разработка методов нечеткой параметрической адаптивной диагностики обучаемого инженера // Автоматизация процессов управления. 2009. № 3. С. 51-56. 11
  8. 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.
  9. Карпов Ю. Г. MODEL CHECKING. Верификация параллельных и распределенных программных систем. — СПб.: БХВ-Петербург, 2010. — 560 с.
  10. Верификация программы и темпоральные логики. URL: <http://logic.pdmi.ras.ru/~yura/modern/034.pdf>
  11. Калянов Г.Н. Моделирование, анализ, реорганизация и оптимизация бизнес-процессов / Учебное пособие. — М.: Финансы и статистика, 2006. — 240 с. URL: <http://www.twirpx.com/file/2204790/>
  12. Saeedloei, Neda, and Gopal Gupta. "Timed definite clause omega-grammars." LIPICs-Leibniz International Proceedings in Informatics. Vol. 7. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2010.
  13. Wang, Yuan, and Yushun Fan. "Using temporal logics for modeling and analysis of workflows." E-Commerce Technology for Dynamic E-Business, 2004. IEEE International Conference on. IEEE, 2004.
  14. YAHODA. URL: <http://web.archive.org/web/20120220001353/http://anna.fi.muni.cz/yahoda/>
  15. CPN Tools. URL: <http://cpntools.org/>
  16. Roméo. URL: <http://romeo.rts-software.org>
  17. TimesTool. URL: <http://www.timestool.com/documentation.shtml>
  18. Tina Toolbox. URL: <http://projects.laas.fr/tina/>
  19. Visual Object Net++. URL: <https://www.techfak.uni-bielefeld.de/~mchen/BioPNML/Intro/VON.html>

## АНАЛИЗ СТРУКТУРНЫХ И СЕМАНТИЧЕСКИХ ОСОБЕННОСТЕЙ ДИАГРАММАТИЧЕСКИХ МОДЕЛЕЙ ДИНАМИЧЕСКИХ РАСПРЕДЕЛЕННЫХ ПОТОКОВ РАБОТ<sup>11</sup>

А.Н. Афанасьев<sup>12</sup>, Н.Н. Войт<sup>13</sup>, Р.Ф. Гайнуллин<sup>14</sup>,  
С.Ю. Кириллов<sup>15</sup>, В.А. Гордеев<sup>16</sup>

Рассматриваются особенности диаграмматических моделей, представленных в широко применяемых на практике артефактах визуальных языков UML, IDEF, BPMN, SharePoint.

### **1. Использование диаграмматических моделей в задачах проектирования, сопровождения и реинжиниринга сложных автоматизированных систем**

Использование диаграмматических моделей в практике проектирования сложных автоматизированных систем (САС), особенно на концептуальных этапах, позволяет повысить эффективность проектных решений, избежать «дорогих» ошибок, улучшить понимание проекта исполнителями, организовать взаимодействие между заказчиками и исполнителями проекта, в ряде случаев автоматизировать процесс получения программного кода. Решение этих задач связано с проблемой «успешности» создания САС. По данным исследований компании Standish Group [1] только около 40% проектов отвечают требованиям заявленной

---

<sup>11</sup> Исследование выполнено при финансовой поддержке РФФИ в рамках научного проекта № 17-07-01417 а

<sup>12</sup> 432027, Ульяновск, ул. Северный Венец, 32, УлГТУ, e-mail: a.afanasev@ulstu.ru

<sup>13</sup> 432027, Ульяновск, ул. Северный Венец, 32, УлГТУ, e-mail: [n.voit260883@gmail.com](mailto:n.voit260883@gmail.com)

<sup>14</sup> 432027, Ульяновск, ул. Северный Венец, 32, УлГТУ, e-mail: [r.gainullin@gmail.com](mailto:r.gainullin@gmail.com)

<sup>15</sup> 432027, Ульяновск, ул. Северный Венец, 32, УлГТУ, e-mail: [s.kirillov@php73.ru](mailto:s.kirillov@php73.ru)

<sup>16</sup> 432027, Ульяновск, ул. Северный Венец, 32, УлГТУ, e-mail: [gordevlad@yandex.ru](mailto:gordevlad@yandex.ru)



функциональности, реализованы в заданные сроки и в рамках заданной сметы.

При сопровождении программно-информационных систем решаются две основные задачи: исправление ошибок и наращивание функциональности. Решение этих задач также связано с диаграмматическими моделями.

Проблема реинжиниринга деятельности крупных предприятиях в условиях активного применения информационных и автоматизированных проектных технологий на концептуальном этапе определяется необходимостью визуализации новых бизнес-процессов, при этом основой визуализации служат диаграмматические модели, представленные в артефактах визуальных языков.

Таким образом, исследование структурных (в первую очередь топологических особенностей) и семантических (связность диаграмм, возможно представленных на различных визуальных языках, в части текстовой составляющей) является актуальной задачей, имеющей большое практическое значение в контексте рассматриваемых проблем проектирования, сопровождения и реинжиниринга САС.

## **2. Язык UML**

Визуальный язык UML является базовым в мастер-технологии RUP в проектировании САС.

Интегрированная модель САС в нотации UML может быть представлена в виде совокупности диаграмм (рис. 1) [2].

Основной структурной особенностью UML-диаграмм являются их комплексность. Например, вершина диаграммы использования может развертываться в диаграмму деятельности, отдельные блоки которой могут соответствовать диаграммам классов и т.д. В диаграмме деятельности могут быть использованы сложные конструкции, связанные с применением разветвителей типа «AND», «OR», «XOR», причем разветвители могут «стоять» на значительном удалении друг от друга и образовывать конструкции с удаленным контекстом.

Диаграммы не должны содержать противоречивой информации.

Противоречивость модели может служить причиной серьезных проблем при ее реализации и последующем использовании на практике. Например, наличие замкнутых путей при изображении отношений агрегирования или композиции приводит к ошибкам в программном коде, который будет реализовывать соответствующие классы. Наличие элементов с одинаковыми именами и различными атрибутами свойств в одном пространстве имен также приводит к неоднозначной интерпретации и может быть источником ошибок.



Рис. 1. Интегрированная модель САС в нотации UML-моделей

В коллективной разработке САС работа проектировщиков открывает дополнительные источники сложно диагностируемых, распределенных по множеству диаграмм ошибок..

При работе в коллективе важно контролировать согласованность понятий, используемых в комплексных диаграммах. Для анализа семантической составляющей диаграмматики в [3] предложен метод анализа и контроля семантических ошибок диаграмматических нотаций бизнес-процессов в составе комплексной диаграммы, созданной в процессе коллективного проектирования, на основе автоматных графических RV-грамматик, отличающийся использованием графовой модели отношений понятий семантической текстовой информации диаграмм и позволяющий расширить класс ошибок, диагностируемых в процессе проектирования САС, и, тем самым, сократить время проектирования.

Семантический контроль комплексных диаграмм позволяет контролировать ошибки, распределенные по различным диаграммам. При семантическом контроле создаваемое комплексное представление об АС становится согласованным и непротиворечивым.

Предлагаемый метод к анализу семантического наполнения диаграмматики бизнес-процессов позволяет диагностировать ошибки несогласованности понятий на ранних этапах разработки САС. Ранняя диагностика таких ошибок позволяет сократить время разработки за счет уменьшения итераций, связанных с их исправлением.

### 3. Язык IDEF

Общая методология IDEF состоит из трех частных методологий моделирования, основанных на графическом представлении систем:

- IDEF0 используется для создания функциональной модели, отображающей структуру и функции системы, а также потоки информации и материальных объектов, связывающие эти функции;
- IDEF1X применяется для построения информационной модели, отображающей структуру и содержание информационных потоков, необходимых для поддержки функций системы;
- IDEF3 позволяет построить динамическую модель меняющихся во времени поведения функций, информации и ресурсов системы.

Структурными особенностями диаграмм семейства IDEF являются: правильность использования связей блоков, означающих входы/выходы определенной функции, управление и механизмы реализации; корректное использование перекрестков в IDEF3, позволяющее избежать зависимостей, неоднозначностей и дедлоков.

С семантической точки зрения наиболее полезной является модельное представление онтологий на визуальном языке SL (Schematic Language), составляющем основу методологии IDEF 5.

Используются четыре типа схем.

1. Диаграмма классификации.
2. Композиционная схема, предназначенная для визуализации представления состава классов онтологии.
3. Схема взаимосвязей.
4. Диаграмма состояния объекта, позволяющая документировать процесс с точки зрения изменения состояния объекта.

Таким образом, стандарт IDEF 5 предоставляет собой структурированную методологию, с помощью которой разрабатывать и динамически поддерживать онтологическую модель системы. Алгоритмы сопоставления объектов этой модели и соответствующих связей с текстовой составляющей и учетом следования блоков диаграмм IDEF 0 и

IDEF 3 обеспечивают семантическую согласованность и целостность комплекса моделей методологии IDEF.

#### **4. Язык BPMN**

Основной целью создания языка BPMN (Business Process Model and Notation) является обеспечение доступной нотацией описания бизнес-процессов всех участников разработки автоматизированных систем: бизнес-аналитиков, создающих схемы процессов вместе с заказчиками, прикладных разработчиков, ответственных за внедрение технологий выполнения бизнес-процессов, руководителей и исполнителей, управляющих бизнес-процессами и отслеживающих их выполнение.

Диаграммы на BPMN могут быть транслированы в ряд исполняемых языков, например, WSBPEL – Web Services Business Process Execution Language. При создании BPMN были рассмотрены графические нотации и методики моделирования следующих формализмов: UML Activity Diagram, UML EDOC Business Processes, IDEF, ebXMLBPSS, Activity-Decision Flow (ADF) Diagram, RosettaNet, LOVeM и Event-ProcessChains (EPCs). Поэтому можно констатировать, что BPMN вобрал в себя лучшие функциональные, алгоритмические, объектные особенности перечисленных инструментов.

Новым в языке BPMN является наличие связей и блоков, имеющих временные метки. Контроль временных параметров этих артефактов является актуальной и важной теоретической и практической задачей, которая решается в рамках разработанной временной RV-грамматики [4].

#### **5. Язык SharePoint**

Визуальные артефакты, методика и инструментарий SharePoint ориентированы на описание и реализацию рабочих процессов - автоматизированное движение документов или элементов через последовательность действий и задач, связанных с бизнес-логистикой.

Рабочие процессы используются для постоянного управления общими бизнес-процессами предприятия, обеспечивая его возможностью включить бизнес-логику в документы. Бизнес-логика представляет собой набор инструкций, которые задают и контролируют действия с документами. Задача рабочих процессов заключается в уменьшении затрат и времени на координирование бизнес-процессов. Основными рабочими процессами являются: утверждение проекта, проверка документа, управление и отслеживание задач сотрудника.

С точки зрения визуализации рабочих процессов графическими средствами SharePoint основной отличительной особенностью является

вложенность процессов, определяющая, в том числе, их иерархическую структуру. Эффективным механизмом контроля такой вложенности является стековый. Он и реализован в качестве внутренней памяти в RV-грамматике языка.

С точки зрения семантической согласованности рабочих процессов для анализа текстовой составляющей могут быть использованы авторские алгоритмы, применяющиеся для решения аналогичной задачи для UML.

### Список литературы

1. <http://www.standishgroup.com/outline>
2. <https://cyberpedia.su/5x389d.html>
3. Afanasyev A., Voit N., Gaynullin R. THE ANALYSIS OF DIAGRAMMATIC MODELS OF WORKFLOWS IN DESIGN OF THE COMPLEX AUTOMATED SYSTEMS // Advances in Intelligent Systems and Computing. 2016. Т. 450. С. 227-236.
4. A. N. Afanasyev, N. N. Voit and S. Y. Kirillov. Development of RYT-grammar for analysis and control dynamic workflows // International Conference on Computing Networking and Informatics (ICCNI), Lagos, Nigeria, 2017, pp. 1-4. doi: 10.1109/ICCNI.2017.8123797

# ИССЛЕДОВАНИЕ МЕТОДОВ АНАЛИЗА И КОНТРОЛЯ ДИАГРАММАТИЧЕСКИХ МОДЕЛЕЙ ДИНАМИЧЕСКИХ РАСПРЕДЕЛЕННЫХ ПОТОКОВ РАБОТ<sup>17</sup>

А.Н. Афанасьев<sup>18</sup>, Н.Н. Войт<sup>19</sup>, В.С. Хородов<sup>20</sup>

Приведены исследования в области актуальности разработки и использования динамических потоков работ, методов и средств их создания. Обозначены основные задачи, возникающие при синтаксическом и семантическом анализе диаграмматических моделей потоков работ

## 1. Введение

Разработка динамических проектных процессов стала одним из новейших направлений в науке «Управление бизнес-процессами (BPM)». Исследователи и практики совершенствуют инструменты, методы и теорию гибкой разработки проектных процессов [1-4]. Идея гибкой разработки проектных процессов была передана в BPM от разработки программного обеспечения, где гибкая разработка программного обеспечения стала установленным термином и методом. В работе [5] приводится базовое определение динамического потока работ при проектировании автоматизированных систем как потока проектных работ, приспособленного к изменениям в окружающей среде.

За последние 20 лет в области управления разработок автоматизированных систем были решены важные проблемы создания, организации, анализа и контроля потоков работ, а также их реализации в практике разработки настоящих систем. В работе [6] дается подробный обзор методов и инструментов для организации потоков работ, которые включают следующие фазы: моделирование потоков работ «как есть»; их

---

<sup>17</sup> Исследование выполнено при финансовой поддержке РФФИ в рамках научного проекта № 17-07-01417 а

<sup>18</sup> 432027, Ульяновск, ул. Северный Венец, 32, УлГТУ, e-mail: a.afanasev@ulstu.ru

<sup>19</sup> 432027, Ульяновск, ул. Северный Венец, 32, УлГТУ, e-mail: n.voit260883@gmail.com

<sup>20</sup> 432071, Ульяновск, ул. Карла Маркса, д. 71, ООО РИТГ, e-mail: v.khorodov73@gmail.com

оптимизация посредством создания модели процесса «чтобы быть»; изменение организационной структуры; «прокрутка» потока работ; усовершенствование процесса. Обсуждается использование систем на этапах разработки потоков работ, таких как ERP-системы, Системы управления Технологическими процессами (WfMS), инструменты моделирования процесса.

## 2. Динамические потоки работ

Динамическая разработка сложных автоматизированных систем связана с адаптацией потоков работ к изменениям в требованиях к системе. Под динамичностью предприятия принято понимать его свойство, обеспечивающее функционирование в динамическом мире [7], и касается двух способностей: 1) приспособиться к изменениям в окружающей среде; 2) обнаружить новые возможности, постоянно появляющиеся в динамическом мире для запуска абсолютно новых продуктов (служб). Становление динамичности требует разработки подхода, который позволяет обнаруживать изменения и воздействовать на них. Потребность в разработке такого подхода появилась как реакция на увеличение степени изменений в требованиях к разработке. Например, в работе [8] автор говорит: “требование изменений на этапах разработки при традиционных методах погружает успешность проекта в болото”.

Ряд известных компаний, например, Whitestein Technologies, Magenta Technologies, SkodaAuto, Volkswagen, Saarlust AG отмечают, что первое поколение статических (монолитных) систем управления жизненным циклом изделия и потоками проектных работ [8] больше не удовлетворяет потребностям многих компаний, подход и автоматизированные средства первого поколения стандартизации процессов потоков проектных работ исчерпали свой ресурс и, как следствие, появляются плохо формализованные (неподходящие, содержащие семантические ошибки) процессы, стимулирующие рост затрат на их исправление, развитие и улучшение.

При разработке сложных автоматизированных систем с помощью подходов Scrum, LSPS [9] и других стратегическое соответствие (полнота, адекватность и непротиворечивость) проекта задуманному часто теряется из-за большого объема информации. Чтобы проект соответствовал требованиям заказчика и был успешным, в помощь разработчику предлагается концептуальная модель, дающая общее представление о структуре организации проектного процесса с описанием акторов (сущностей) и их роли, функциях и связи друг с другом.

Выделяют концептуальные модели организации динамических распределенных потоков проектных работ, которые имеют значительные

преимущества перед монолитными (традиционными): SECI [10], схемы Дж. Захмана [11], Rational ADS [11].

К традиционным системам управления потоками работ относят ProBis [12].

К динамическим системам управления потоками проектных работ, согласно работам [13, 14, 15], относят YAWL (Yet Another Workflow Language), iPB.

Во всех подобных системах используются диаграмматические представления потоков работ. При этом решаются задачи анализа структуры (синтаксиса) и смысла (семантики) диаграмм. Так, в работе [16] используются цветные сети Петри для динамического семантического анализа потоков работ, а в работе [17] подход  $\pi$ -Calculus, формализующий потоки работ в алгебраические высказывания логики первого порядка.

### 3. Графические грамматики

В современной теории графических визуальных языков для представления диаграмм используется логическая модель, содержащая графические объекты и связи между ними. Для обработки таких моделей используются графические грамматики. John L. Pfaltz и Azreil Rosenfeld предложили веб-грамматику [18]. Zhang и Costagliola [19, 20] разработали позиционную графическую грамматику, относящуюся к контекстно-свободной грамматике. Wittenberg и Weitzman [21] разработали реляционную графическую грамматику. Zhang и Orgun [22] описали сохраняющую графическую грамматику в своих работах. Однако упомянутые графические грамматики имеют следующие недостатки.

1. Позиционные грамматики, развиваясь на базе плекс-структур, не предполагают использование областей соединения и не могут применяться для графических языков, объекты которых имеют динамически изменяемое количество входов/выходов.

2. Авторы реляционных грамматик оговариваются о несовершенстве механизма нейтрализации в плане неполноты формируемого списка ошибок.

3. Отсутствует контроль семантической целостности (текстовой атрибутики комплексных диаграмматических моделей, представленных на различных визуальных языках), а также семантической согласованности в плане структурных вопросов диаграмм между собой и концептуальной моделью в целом.

4. Общими недостатками вышеописанных грамматик являются: увеличение числа продукций при построении грамматики для неструктурированных графических языков (при неизменном количестве



примитивов графического языка для описания всех вариантов неструктурированности происходит значительное увеличение количества (продукций), сложность построения грамматики, большие временные затраты анализа (анализаторы, построенные на базе рассмотренных грамматик, обладают полиномиальным или экспоненциальным временем анализа диаграмм графических).

В работах [23, 24] для обработки визуальных языков предлагается синтаксически-ориентированный подход на базе семейства RV-грамматик. Однако механизмы анализа и контроля структурных и семантических особенностей диаграмм в плане их целостности и согласованности между собой и концептуальной моделью, в том числе по текстовой атрибутике, отсутствуют.

#### **4. Нейтрализация ошибок**

Задача нейтрализации ошибок и ее решение хорошо отражено в классических работах по компиляторам, например, [25]. Предложен также метод нейтрализации ошибок для RV-грамматик [26]. Однако вопросы нейтрализации для диаграмматических моделей динамических распределенных потоков работ в них не решены.

#### **5. Метакомпиляция**

Метакомпиляция языков программирования и специализированных языков является классической задачей в теории формальных языков и грамматик. Ее решение связано с описанием лексики, синтаксиса и семантики с помощью специальных нотаций. Авторами предлагается аналогичное решение: синтаксис диаграмматических моделей визуальных языков предлагается описать с помощью текстовых правил, учитывающих специфику указанных взаимосвязанных моделей. Отличительной особенностью алгоритма метатрансляции является реализация возможности генерации операций с памятью RV-анализатора.

Трансляция моделей визуального языка в другой целевой язык на основе RV-грамматик решается в работе [27]. Однако задача трансляции нескольких взаимосвязанных диаграмматических моделей потоков работ, представленных на различных языках, в целевой язык не рассматривается.

#### **6. Инструментальные средства**

В наиболее распространенных инструментальных средствах создания и обработки диаграмматических моделей, таких как Microsoft Visio [28], Visual paradigm for UML [29], Aris Toolset [30], IBM Rational

Software Architect (RSA) [31] анализ и контроль диаграмматических моделей производится прямыми методами, требует нескольких «проходов» в зависимости от контролируемого типа ошибки, отсутствует контроль структурных особенностей комплексных диаграмматических моделей, а также семантический контроль целостности и согласованности структурной и текстовой атрибутики связанных диаграмматических моделей динамических потоков работ.

Таким образом, для решения вышеуказанных проблем необходима разработка комплексной системы новых методов, моделей и средств анализа и контроля диаграмматических моделей динамических распределенных потоков работ, обеспечивающей решение поставленных в проекте задач.

### Список литературы

1. Thiemich, C. & Puhlmann, F., 2013. An Agile BPM Project Methodology. In BPM Conference. van der
2. Seethamraju, R. & Seethamraju, J., 2009. Enterprise systems and Business Process Agility- A Case Study. In Proceedings of the 42nd Hawaii International Conference on System Sciences., pp.1-12.
3. Bider, I., Johannesson, P. & Perjons, E., 2010. In Search of the Holy Grail: Integrating social software with BPM. Experience report. In Enterprise, Business-Process and Information Systems Modeling, LNBP, Vol. 50. Springer, pp.1-13.
4. Kindermann, H., 2013. Empowering process participants - the way to a truly agile business process management. [Online] Available at: <http://www.onthemove-conferences.org/index.php/keynotes2013/2013keynotekindermann> [Accessed 15 Augustus 2013].
5. Agile Business Process Development: Why, How and When - Applying Nonaka's theory of knowledge transformation to business process development. <https://www.researchgate.net/publication/266078141>
6. Becker, J., Kugeler, M. & Rosemann, M., eds., 2011. Process Management: A Guide for the Design of Business Processes. 2nd ed. Springer.
7. Sherehiy, B. W., K. & J.K., L., 2007. A review of enterprise agility: Concepts, frameworks, and attributes. International Journal of Industrial Ergonomics, 37, pp.445-60.
8. Highsmith, J., Orr, K. & Cockburn, A., 2000. E-Business Application Delivery, pp. 4-17. [Online] Available at: [www.cutter.com/freestuff/ead0002.pdf](http://www.cutter.com/freestuff/ead0002.pdf).
9. A global Swiss company offering advanced intelligent application software for multiple business sectors. <http://whitestein.com/>
10. Gram Consulting, 2009. "Ba" for Management Development. [Online] Available at: <http://gramconsulting.com/2009/04/ba-for-management-development/>
11. Концептуальное моделирование компьютеризованных систем: учебное пособие / П.И. Соснин. – Ульяновск: УлГТУ, 2008. – 198 с.
12. Andersson, T., Andersson-Ceder, A. & Bider, I., 2002. State flow as a way of analyzing business processes-case studies. Logistics Information Management, 15(1), pp.34-45.
13. YAWL Foundation, 2004. YAWL. [Online] Available at: <http://www.yawlfoundation.org/> [Accessed 05 February 2016].
14. Bider, I., 2014. Analysis of Agile Software Development from the Knowledge Transformation Perspective. In Johansson, B., ed. To appear in 13th International

- Conference on Perspectives in Business Informatics Research (BIR 2014). Lund, Sweden. Springer, LNBIP
15. IbisSoft, 2009. iPB Reference Manual. [Online] Available at: <http://docs.ibissoft.se/node/3> [Accessed 05 February 2016].
  16. Jalali, A., Wohed, P. & Ouyang, C., 2012. Aspect Oriented Business Process Modelling with Precedence. In Business Process Model and Notation, LNBIP, Vol. 125. Springer, pp.23-37.
  17. Hasso Plattner Institut. <http://bpt.hpi.uni-potsdam.de>
  18. Fu K. Structural methods of pattern recognition. – Moscow: Mir, 1977. – P.319.
  19. Zhang D. Q., Zhang K. Reserved graph grammar: A specification tool for diagrammatic VPLs //Visual Languages. Proceedings. 1997 IEEE Symposium on. – IEEE. – pp. 284-291 (1997)
  20. Costagliola G., Lucia A.D., Orece S., Tortora G. A parsing methodology for the implementation of visual systems. [Online] Available at: <http://www.dmi.unisa.it/people/costagliola/www/home/papers/method.ps.gz> [Accessed 05 February 2016].
  21. Wittenburg K., Weitzman L. Relational grammars: Theory and practice in a visual language interface for process modeling (1996). [Online] Available at: <http://citeseer.ist.psu.edu/wittenburg96relational.html> [Accessed 05 February 2016].
  22. Zhang K. B., Zhang K., Orgun M. A. Using Graph Grammar to Implement Global Layout for A Visual Programming Language Generation System. (2002)
  23. Шаров О.Г., Афанасьев А. Н. Синтаксически-ориентированная реализация графических языков на основе автоматных графических грамматик // Программирование. – 2005. – №6. – С. 56-66.
  24. Шаров О.Г., Афанасьев А. Н. Методы и средства трансляции графических диаграмм // Программирование. – 2011. – №3. – С. 65-76.
  25. Ахо А.В., Сети Р. Ульман Дж.Д. Компиляторы: Принципы, технологии и инструменты. - М.: Издательский дом "Вильямс", 2003.
  26. Шаров О.Г., Афанасьев А. Н. Нейтрализация синтаксических ошибок в графических языках / О. Г. Шаров, А. Н. Афанасьев // Программирование. – 2008. – №1. – С. 61-66.
  27. Шаров О.Г., Афанасьев А.Н. Методы и средства трансляции графических диаграмм // Программирование. – 2011. – Т. 37. – № 3. – С. 65-75. <http://elibrary.ru/item.asp?id=16777695>
  28. Roth C. Using Microsoft Visio 2010. – Pearson Education, 2011.
  29. Paradigm V. Visual paradigm for uml //Hong Kong: Visual Paradigm International. Available at: <http://www.visual-paradigm.com/product/vpuml/>. Accessed April. – 2010. – Т. 15. – С. 2010.
  30. Santos Jr P. S., Almeida J. P. A., Pianissolla T. L. Uncovering the organisational modelling and business process modelling languages in the ARIS method //International Journal of Business Process Integration and Management. – 2011. – Т. 5. – №. 2. – С. 130-143.
  31. Hoffmann H. P. Deploying model-based systems engineering with IBM® rational® solutions for systems and software engineering //Digital Avionics Systems Conference (DASC), 2012 IEEE/AIAA 31st. – IEEE, 2012. – С. 1-8. <http://www.standishgroup.com/outline>

## РАЗРАБОТКА И ОПИСАНИЕ СИСТЕМЫ КЛАССИФИКАЦИИ ВРЕМЕННЫХ РЯДОВ<sup>21</sup>

Т.В. Афанасьева, А.П. Курлов<sup>22</sup>

В работе рассматривается архитектура и функциональное наполнение системы многоклассовой классификации временных рядов. Приведено описание разработанного ПО и результаты экспериментов, показывающих результативность классификации.

### Введение

При анализе поведения организационно-технических систем (ОТС) часто применяют алгоритмы machine learning для оценивания класса поведения всей системы или ее компонент. К таким алгоритмам относятся широко известные алгоритмы классификации временных рядов (ВР) (алгоритмы обучения с учителем)[1]: метод k ближайших соседей (где k задается через кросс валидацию), наивный байесовский классификатор, дерево решений C4.5, метод опорных векторов с линейной и квадратичной базовой функцией ядер, Random forest, Rotation Forest, байесовская сеть, нечеткая нейронная сеть и др.

В 2009 году был разработан метод шейплетов для многоклассовой классификации ВР, который основан на извлечении свойств ВР, представленных в виде так называемых шейплетов [2]. Метод шейплетов выделяет в своём классе ВР представителя, что позволяет значительно повысить точность, поэтому он был выбран для программной реализации задачи многоклассовой классификации множества ВР [3].

### Проектирование системы многоклассовой классификации ВР

Под шейплетом понимают часть ВР, которая наилучшим образом характеризует весь класс ВР. Это означает, что расстояние между ВР с очень похожими шейплетами мало, а между ВР с непохожими шейплетами велико.

---

<sup>21</sup> Работа выполнена при финансовой поддержке РФФИ проект № 16-07-005353

<sup>22</sup> 432027, Ульяновск, ул. Северный Венец, 32, УлГТУ, e-mail: tv.afanasjeva@gmail.com

Итерационный процесс классификации заключается в переборе кандидатов на шейплеты, для каждого из которых вычисляется мера качества, называемая «информационная прибыль». По окончании итераций, алгоритм возвращает лучший шейплет и соответствующую информационную прибыль. Алгоритм останавливает работу после того, как информационная прибыль шейплета перестаёт меняться.

Для определения представителя класса или лучшего шейплета среди ВР одного класса необходимо осуществить поиск данного шейплета во всех ВР этого класса с помощью скользящего окна.

Обычно в алгоритмах классификации сначала проводится его обучение на обучающей выборке, а затем обученный алгоритм проверяется на тестовой выборке. В предлагаемом проекте сначала строятся модели классов и модели ВР с использованием понятия «лучший шейплет». Эти модели являются входами не только для классификации тестовой выборки, но и полноценным выходом системы, так как по этим моделям можно оценить исследуемую совокупность (см. рис. 1).

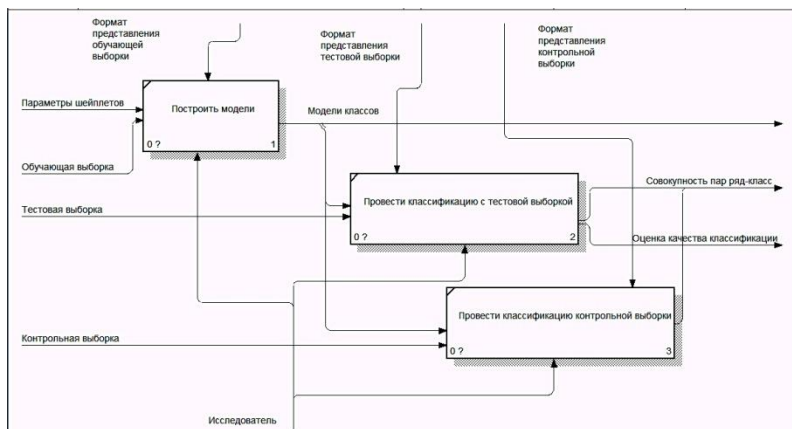


Рис. 1. Архитектура системы классификации

## Описание ПО системы классификации ВР

На основе метода шейплетов было разработано ПО в соответствии с архитектурой, представленной выше. При разработке данной системы использовалась среда Microsoft Visual Studio 2015. Для создания интерфейса взаимодействия с пользователем использовалась графическая система создания приложений Windows Presentation Foundation, которая является подсистемой библиотеки NET.Framework с языком XAML для

создания элементов интерфейса. Языком программирования является объектно-ориентированный язык программирования C#.

ПО способно принимать на вход три вида файлов с временными рядами: обучающая выборка, тестовая выборка и контрольная выборка. Работа разработанного ПО для многоклассовой классификации ВР приведена на рисунке 2.

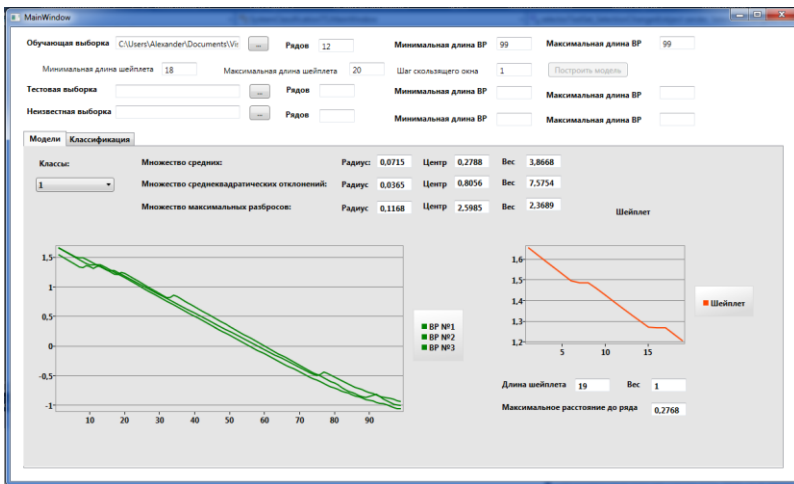


Рис.2. Пример работы ПО классификации ВР

Для того чтобы выполнить непосредственно алгоритм классификации необходимо загрузить в программу либо файл с тестовой выборкой, либо файл с контрольной выборкой. Для запуска классификации необходимо перейти на вкладку «Классификация» и нажать кнопку «Запустить классификацию». После чего на вкладке появится информация о классификации, а именно таблица с номером ряда, определённым классом, и в случае с тестовой выборкой также будет указан настоящий класс. Также отобразится значение характеристики Ассигасу. Для того чтобы более подробно посмотреть информацию по каждому ряду, необходимо нажать на строку в таблице с соответствующим рядом. В таблице «Информация по ряду» отобразятся все параметры классификации по каждому классу, исходя из которых можно понять, почему ряд был отнесён именно к этому классу. Также на графике есть возможность увидеть графическое представление исследуемого ряда и рядов класса, к которому данный ряд был отнесён.

## Заключение

Разработанная система предназначена для многоклассовой классификации и ориентирована на работу с временными рядами, которые имеют следующие свойства:

- Одномерные временные ряды (при необходимости данный алгоритм может быть модернизирован для работы с многомерными временными рядами);
- Стационарные временные ряды.

Для оценивания точности классификации использовалась база данных ВР В качестве источника данных временных рядов для проекта выбрана база «The UCR Time Series Classification Archive» ([http://www.cs.ucr.edu/~eamonn/time\\_series\\_data/](http://www.cs.ucr.edu/~eamonn/time_series_data/)), содержащая временные ряды различных предметных областей от биологии до техники. Для оценки качества классификации временных рядов использовалось 10 выборок, длина рядов которых варьировалась от 100 до 142, а количество классов от 2 до 5. Полученные результаты применения метода шейплетов показали высокую точность. Так, для ВР с одинаковой длиной была достигнута точность классификации 88.13%, для ВР с разной длиной 86.23%. Общая точность алгоритма классификации ВР составила 87.18%.

## Список литературы

1. Bagnall A., Lines J., Hills J., and Bostrom A. Time-series classification withCOTE: The collective of transformation-based ensembles. IEEE Transactions on Knowledge and Data Engineering, 27:2522–2535, 2015 [Электронный ресурс]. Доступен по адресу: <http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=7069254>. [Дата обращения: 25.05.17].
2. Ye L., Keogh E. Time series shapelets: a new primitive for data mining //Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining (KDD'09). ACM Publ., 2009, pp. 947-956.
3. Карпенко А. П., Кострубин М. С., Чернышев А. С. Эффективность классификации многомерных временных рядов с помощью шейплетов // Наука и Образование: научное издание МГТУ им. Н.Э. Баумана. Электрон. журн. 2015. № 11. С. 382–405. [Электронный ресурс]. Режим доступа: <http://technomag.edu.ru/jour/article/view/127/129>. [Дата обращения: 25.05.17].

## МОДЕЛИРОВАНИЕ КОМПЛЕКСНЫХ ЛИНГВИСТИЧЕСКИХ ОЦЕНОК КОМПОНЕНТ ОРГАНИЗАЦИОННО-ТЕХНИЧЕСКИХ СИСТЕМ С ИСПОЛЬЗОВАНИЕМ НЕЧЕТКИХ МНОЖЕСТВ<sup>23</sup>

Т.В. Афанасьева, А.Е. Полбин, Д.А.Максимов<sup>24</sup>

В работе рассматривается моделирование комплексных лингвистических оценок компонент организационно-технических систем по набору показателей, позволяющих группировать их в кластеры с одинаковым уровнем качества.

### Введение

При анализе организационно-технических систем (ОТС) в задачах управления часто применяют модели оценивания таких сложных систем, выраженные лингвистическими терминами [1,2]. Такие термины могут быть предварительно сформированы экспертно или в результате оценивания числовых показателей и отражать качество ОТС. Полученные лингвистические оценки позволяют оперативно, без привлечения дополнительных сложных вычислений, группировать компоненты ОТС в кластеры с одинаковым уровнем качества. Как правило, оценочные лингвистические термины получаемые по отдельным показателям качества первоначально «вычисляются» для отдельных компонент ОТС. Под компонентами ОТС понимается типовые процессы, персонал и ресурсы. В настоящей статье рассматривается применение нечетких систем логического вывода для интеграции показателей качества отдельных компонент ОТС и получения лингвистических оценок.

### Модель компоненты ОТС

Пусть модель компоненты ОТС описывается в виде:  $S = \langle A, B, D \rangle$ , где  
A - вектор, определяющий критерий «качество типовых процессов»;  
B - вектор, определяющий критерий «качество персонала»;  
D - вектор, определяющий критерий «качество ресурсов».

---

<sup>23</sup> Работа выполнена при финансовой поддержке РФФИ проект № 16-07-005353

<sup>24</sup> 432027, Ульяновск, ул. Северный Венец, 32, УлГТУ, e-mail: tv.afanasjeva@gmail.com



Предположим, что имеются результаты оценивания компонента ОТС с использованием указанных критериев в абсолютной шкале от 0 до 5, где 0 означает полное отсутствие качества для оценки, 5 означает что компонент удовлетворяет всем требованиям качества.

Тогда по некоторым исходным данным применяя средневзвешенную сумму, учитывающую важность отдельного критерия можно получить комплексную оценку компонента ОТС, выраженную в виде числа (см. Таблицу 1).

Таблица 1. Результат оценивания компонент ОТС

Компонент ОТС	A	B	D	Всего	Ранг
<u><a href="#">К1</a></u>	4,60	4,40	5,00	4,68	1
<u><a href="#">К2</a></u>	4,20	4,60	5,00	4,68	1
<u><a href="#">К3</a></u>	3,60	4,20	5,00	4,4	2
<u><a href="#">К4</a></u>	4,00	3,00	5,00	4	3
<u><a href="#">К5</a></u>	2,80	3,60	4,00	3,6	4
<u><a href="#">К6</a></u>	4,00	3,40	3,33	3,5	5
<u><a href="#">К7</a></u>	3,00	1,60	2,33	2,17	6

По результатам из Таблицы 1 видно, что компоненты К1 и К2 относятся к кластеру с наивысшим уровнем качества, заняв первое место в рейтинге. Последним в рейтинге оказался компонент К7, продемонстрировавший низкий уровень качества. Остальные компоненты ОТС показали средние результаты.

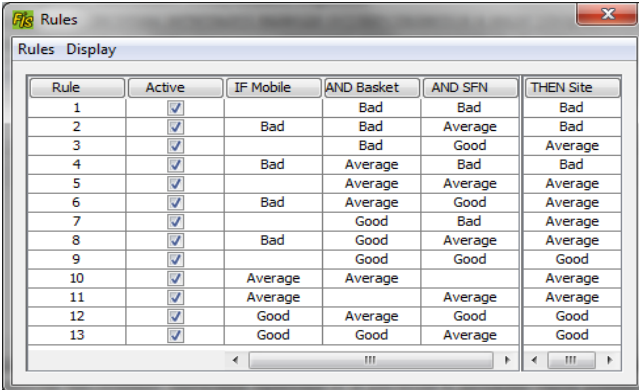
Для получения комплексной лингвистической оценки компоненты ОТС необходимо решить ряд задач. В общем случае разработка нечеткой системы (в дальнейшем – модели) заключается в описании лингвистических переменных системы, разработке структуры системы, выработке стратегии логического вывода, задания базы правил и, наконец, выборе наиболее приемлемого способа дефазификации полученных результатов. Лингвистические переменные модели составляют «словарь», используемый при логическом выводе на основании заданных правил, структура системы описывает информационные потоки, связывающие отдельные элементы модели, а база правил является «мозгом» нечеткой системы, в котором сконцентрированы знания экспертов.

## Комплексное оценивание компонент ОТС

Таким образом, при проектировании нечеткой логической модели комплексного оценивания компоненты ОТС можно выделить следующие этапы:

- Этап определения лингвистических переменных моделей;
- Этап выбора функции принадлежности;
- Этап создания базы правил;
- Этап выбора способа дефаззификации.

Ниже рассмотрим особенности проектирования нечеткой модели комплексного оценивания компоненты ОТС. На первом шаге определяется по одному правилу для каждой комбинации термов входных переменных, т.е. переменных, находящихся в части «ЕСЛИ» правила. Для этих правил устанавливается степень их поддержки равная 1. Далее, для каждого заданного правила, выбирается наиболее правдоподобный терм выходной переменной (формируется часть «ТО» правила). В общем случае размер базы правил (число правил в ней) нечеткой системы будет равен произведению количества термов входных переменных. В данном случае с тремя входными переменными и одной выходной переменной, каждая из которых содержит по 3 термина, база правил будет состоять из 27 правил, которая в дальнейшем была оптимизирована до 13 правил (см. рис.1).



Rule	Active	IF Mobile	AND Basket	AND SFN	THEN Site
1	<input checked="" type="checkbox"/>		Bad	Bad	Bad
2	<input checked="" type="checkbox"/>	Bad	Bad	Average	Bad
3	<input checked="" type="checkbox"/>		Bad	Good	Average
4	<input checked="" type="checkbox"/>	Bad	Average	Bad	Bad
5	<input checked="" type="checkbox"/>		Average	Average	Average
6	<input checked="" type="checkbox"/>	Bad	Average	Good	Average
7	<input checked="" type="checkbox"/>		Good	Bad	Average
8	<input checked="" type="checkbox"/>	Bad	Good	Average	Average
9	<input checked="" type="checkbox"/>		Good	Good	Good
10	<input checked="" type="checkbox"/>	Average	Average		Average
11	<input checked="" type="checkbox"/>	Average		Average	Average
12	<input checked="" type="checkbox"/>	Good	Average	Good	Good
13	<input checked="" type="checkbox"/>	Good	Good	Average	Good

Рис.1. Правила нечеткой модели генерации комплексной лингвистической оценки компонента ОТС

Применяя системы логического вывода по Мамдани и по Сугено были получены лингвистические термы, выражающие комплексную оценку состояния компоненты ОТС. Для валидации нечетких моделей комплексного оценивания было проведено сравнение

аппроксимированных нечётких оценок по сравнению с числовыми оценками, представленными в таблице 1. Результаты сравнительной оценки представлены в таблице 2.

Таблица 2. Результаты сравнительного оценивания

Компонент ОТС	Мамдани		Суждено	
	Оценка	Ранг	Оценка	Ранг
<a href="#">К1</a>	4,460	1	4,600	2
<a href="#">К2</a>	4,460	1	4,680	1
<a href="#">К3</a>	4,460	1	4,400	3
<a href="#">К4</a>	4,460	1	4,000	4
<a href="#">К5</a>	3,189	2	3,600	5
<a href="#">К6</a>	3,000	3	3,492	6
<a href="#">К7</a>	2,043	4	2,172	7

## Заключение

Полученные результаты демонстрируют, что погрешность в нечетком оценивании по Мамдани в среднем составляет 8%, при наименьшем различии менее чем в 1% и наибольшем в 14%. По полученным данным можно утверждать, что в конкретном случае нечеткая модель оценивания компонент ОТС была составлена с достаточной степенью точности для использования.

Однако наиболее точной является модель, полученная в системе нечеткого вывода по Суждено. Данная модель наиболее точно отражает комплексную оценку компонент ОТС, полученную в результате применения нечетких множеств.

## Список литературы

1. Афанасьева Т.В. и др. Моделирование лингвистических оценок на основе ACL-шкалы. / Труды V Международной научно-практической конференции “Интегрированные модели и мягкие вычисления (Коломна, 2009 г.)”, В 2-х т. Т.1. М.: Физматлит, 2009. С.298-304.
2. Ярушкина Н.Г., Афанасьева Т.В. Метод нечеткого моделирования и анализа тенденций временных рядов // Интеллектуальные системы управления: коллективная монография, под ред. Академика РАН С.Н. Васильева. - М.: Машиностроение, 2010. – С. 301-305.

## ВЫЯВЛЕНИЕ НАРУШЕНИЙ ПРИ МНОГОМЕРНОМ СТАТИСТИЧЕСКОМ КОНТРОЛЕ ПРОЦЕССА ОЧИСТКИ ПИТЬЕВОЙ ВОДЫ

А.Д. Барт<sup>25</sup>

Разработана методика оценки стабильности физико-химических показателей качества при очистке питьевой воды. Предлагается осуществлять мониторинг показателей с помощью методов статистического управления, используемых при контроле технологических процессов. Стабильность зависимых показателей оценивается с помощью многомерных карт Хотеллинга. Предложены способы выявления показателей качества, с которыми связано нарушение стабильности процесса. Практические расчеты проведены на примере данных Санкт-Петербургского водоканала.

### Введение

При превышении предельно допустимых значений некоторых параметров питьевой воды она становится непригодной для использования. Качество воды зависит от состояния водоисточника и применяемых методов очистки. Необходимо проконтролировать и обеспечить стабильность физико-химических показателей воды, поступающей к потребителю.

Мониторинг показателей может проводиться статистическими методами, аналогичными тем, что используются при статистическом контроле и управлении технологическими процессами [1-3].

При исследовании стабильности показателей качества питьевой воды использовались данные Санкт – Петербургского водоканала. Один раз в сутки проводились измерения восьми физико-химических показателей [4-5]: мутности, цветности, содержания алюминия и хлоридов, рН, содержание остаточного хлора, окисляемость и щелочность.

Для оценки стабильности физико-химических параметров качества питьевой воды была предложена следующая методика:

1. По результатам мониторинга за предшествующий период проверяется нормальность распределения показателей, при необходимости проводится нормализация данных.

---

<sup>25</sup>432027, Ульяновск, ул. Северный Венец, 32, УЛГТУ, e-mail:nasty.a.bart.1995@mail.ru

2. Для оценки взаимосвязи между показателями используется корреляционная матрица, при этом выявляются значимые корреляции.

3. Для оценки стабильности процесса по независимым показателям применяются карты Шухарта для индивидуальных наблюдений и скользящих размахов, для контроля коррелированных показателей используется карта Хотеллинга.

4. Если при мониторинге на карте появляется точка вне контрольных границ, необходимо внесение изменений в процесс очистки с целью обеспечения стабильности процесса.

## 1. Построение карты Хотеллинга

Анализ матрицы корреляций показал, что многомерный контроль с применением карты Хотеллинга необходим при мониторинге пары коррелированных показателей «содержание алюминия» и «окисляемость», а также для группы показателей рН, «содержание остаточного хлора» и «щелочность».

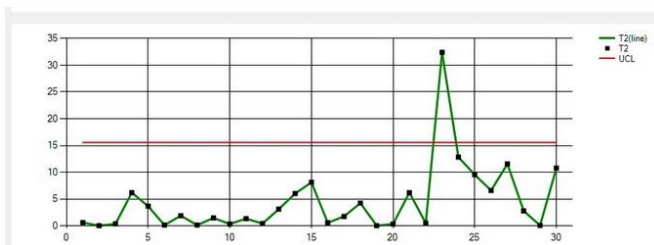


Рис.1. Карта Хотеллинга для группы показателей «содержание алюминия» и «окисляемость» (апрель)

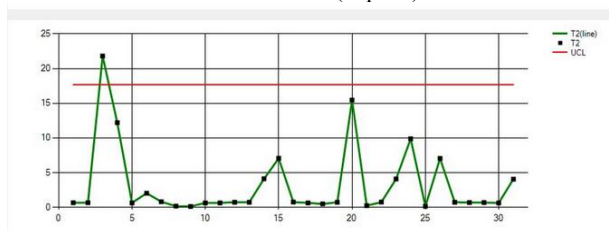


Рис.2. Карта Хотеллинга для группы показателей рН, «содержание остаточного хлора» и «щелочность» (август)

При проведении многомерного статистического контроля по двум группам физико-химических показателей качества питьевой воды, оказалось (рис.1, 2), что в некоторые периоды времени показатели ведут

себя нестабильно. Однако, карта Хотеллинга не показывает непосредственно, с каким из показателей (или совместным влиянием показателей) связан выход статистики Хотеллинга за контрольную границу. В соответствии с этим возникает задача выявления показателя, по какому именно произошло нарушение процесса.

## 2. Выявление нарушений с помощью карт Шухарта

Одним из способов выявления показателя качества, по которому произошло нарушение – построение карт Шухарта по каждому из показателей отдельно [3]. Применяя данный подход к группе показателей «содержание алюминия» и «окисляемость» за апрель месяц, можно сделать вывод, что ответственность за нарушение процесса несут оба показателя. Это объясняется тем, что, как и на построенных картах Шухарта (рис. 3, 4) так и на карте Хотеллинга (рис. 1), нарушение процесса произошло в 23 наблюдении.

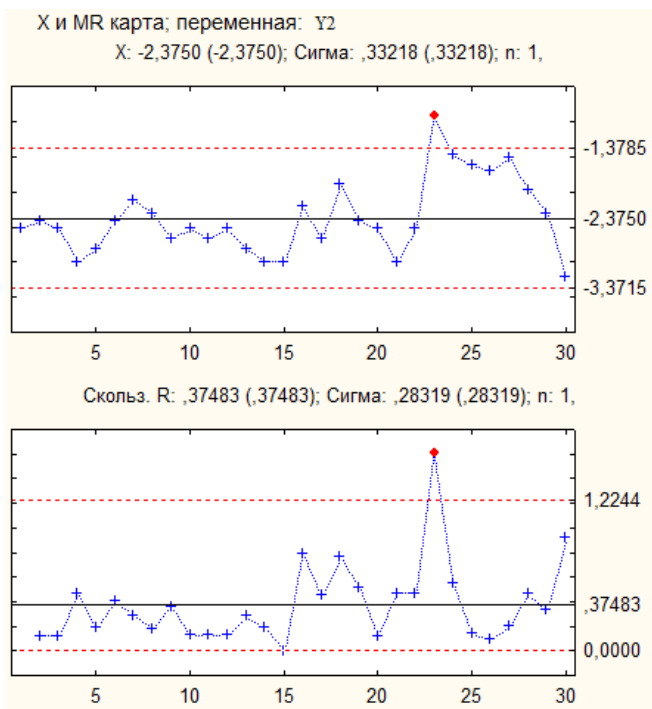


Рис.3. Карта Шухарта по показателю «содержание алюминия»

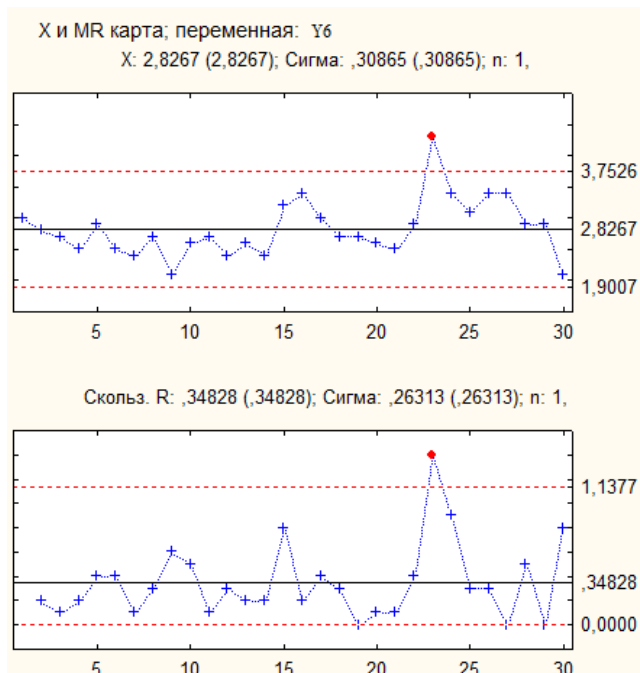


Рис.4. Карта Шухарта по показателю «окисляемость»

### 3.Понижение размерности при использовании алгоритма Хотеллинга

Еще один подход, который позволит выявить показатель, по которому произошло нарушение процесса – это поочередное удаление данных по одному из показателей и построение карты Хотеллинга не по  $p$ , а по  $(p-1)$  показателю качества очистки питьевой воды, т.е. снижение размерности алгоритма Хотеллинга [6-7].

Анализируя карту Хотеллинга для группы показателей по рис. 2, отметим, что в третьем наблюдении зафиксирован выброс статистики Хотеллинга за критическую границу. Для выявления показателя, по которому произошло нарушение процесса, последовательно удалялось по одному из трех показателей и строились карты Хотеллинга по двум оставшимся. Рассматривая построенные карты (рис. 5-7), видно, что выброс связан с показателем «щелочность».

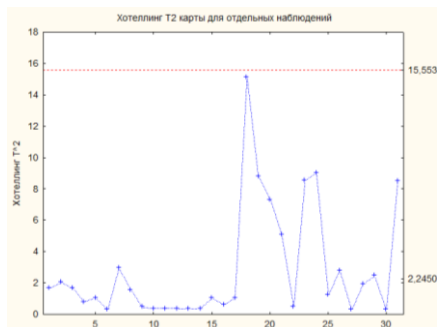


Рис. 5. Карта Хотеллинга по группе показателей pH и «содержание остаточного хлора»

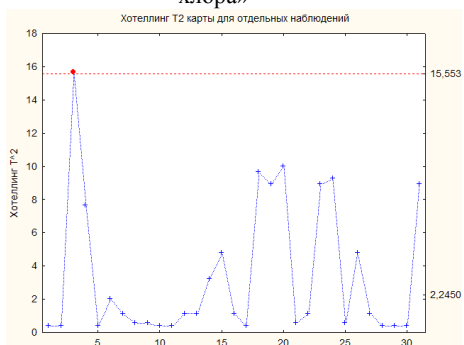


Рис. 6. Карта Хотеллинга по группе показателей «содержание остаточного хлора» и «щелочность»

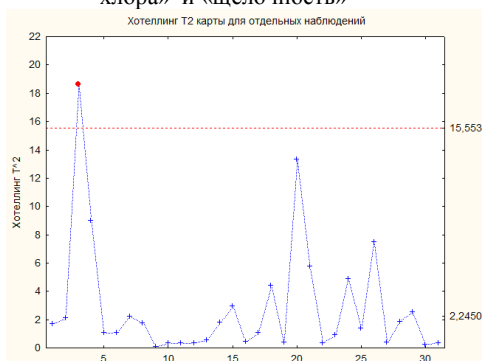


Рис. 7. Карта Хотеллинга по группе показателей pH и «щелочность»

Отметим, что выход точки за контрольную границу означает лишь нарушение статистической стабильности, но не выход за границу допуска, поэтому своевременно принятые меры могут обеспечить необходимое



качество питьевой воды: контрольная карта заранее предупреждает о возможном нарушении [8-10].

### Список литературы

1. Клячкин В.Н. Статистические методы в управлении качеством: компьютерные технологии – М.: Финансы и статистика, ИНФРА-М, 2009. – 304 с.
2. Клячкин В.Н., Кувайскова Ю.Е., Алексеева В.А. Статистические методы анализа данных. – М.: Финансы и статистика, 2016. – 240 с.
3. Уилер, Д. Статистическое управление процессами/ Оптимизация бизнеса с использованием контрольных карт Шухарта; пер. с англ. / Д. Уилер, Д. Чамберс. – М.: Альпина Бизнес Букс, 2009. – 409 с.
4. Барт А.Д. К вопросу об оценке ковариационной матрицы при статистическом контроле по индивидуальным наблюдениям // Материалы научно-практической всероссийской конференции молодых ученых «Прикладная математика и информатика: современные исследования в области естественных и технических наук». – Тольятти, 2017. – С.68-71.
5. Клячкин В.Н., Барт А.Д. Оценка стабильности показателей качества при очистке питьевой воды // Экологические системы и приборы, 2017. №8. С.14-20.
6. Клячкин В.Н. Модели и методы статистического контроля многопараметрического технологического процесса. – М.: ФИЗМАТЛИТ, 2011. – 196 с.
7. Боровиков В. Statistica: Искусство анализа данных на компьютере. – СПб: Питер, 2001. – 656 с.
8. Клячкин В.Н., Карпунина И.Н. Анализ стабильности функционирования технического объекта по критерию многомерного рассеяния контролируемых параметров // В сборнике: Информационные технологии и нанотехнологии. Самарский национальный исследовательский университет имени академика С.П. Королева. 2017. С. 1751-1754.
9. Бубырь Д.С., Булыжев Е.М., Клячкин В.Н., Крашенинников В.Р. Регрессионные модели оценки качества питьевой воды по физико-химическим показателям водоисточника // Известия высших учебных заведений. Поволжский регион. Технические науки. 2016. № 4 (40). С. 46-54.
10. Клячкин В.Н. Многомерный статистический контроль процесса // В сборнике: Инновационные технологии в металлообработке Сборник научных трудов. Ульяновск. 2017. С. 219-226.

## ИССЛЕДОВАНИЕ САПР ЭЛЕКТРОННЫХ УСТРОЙСТВ DELTA DESIGN<sup>26</sup>

С.И. Бочков<sup>27</sup>

В статье представлен обзор схемотехнической САПР Delta Design, рассмотрены особенности модулей САПР и их функциональные возможности.

### Введение

Автоматизация проектирования занимает особое место среди информационных технологий. Знание основ проектной деятельности, умения работать со средствами САПР требуются инженеру (проектировщику) для решения современных проектных задач.

Одной из ключевых отраслей, использующей САПР для решения задач, является электронная промышленность, в частности, занимающаяся проектированием печатных плат.

Существует множество специализированных систем, направленных на автоматизацию проектирования в схемотехнике [3, 4]. В рамках статьи даётся обзор отечественной САПР Delta Design, структурная схема которой [1] приведена на рис. 1.

### Менеджер библиотек

*Менеджер библиотек* обеспечивает ведение базы электрорадиоизделий в среде Delta Design. База данных РЭК построена на основе совокупности библиотек - отдельное функционально завершенное хранилище данных о компонентах.

Использование библиотек позволяет четко структурировать базу данных электрорадиоизделий и позволяет легко осуществлять импорт и

---

<sup>26</sup> Исследования поддержаны грантом Министерства образования и науки Российской Федерации, проект № 2.1615.2017/4.6.

Исследование выполнено при финансовой поддержке РФФИ и Правительства Ульяновской области в рамках научного проекта № 16-47-732152

Исследование выполнено при финансовой поддержке РФФИ в рамках научного проекта № 17-07-01417

<sup>27</sup> 432027, Ульяновск, ул. Северный Венец, 32, УлГТУ, e-mail: bochkovsam1@rambler.ru

экспорт различных групп компонентов: библиотек, частей библиотек, отдельных компонентов, отдельных частей описаний компонентов.

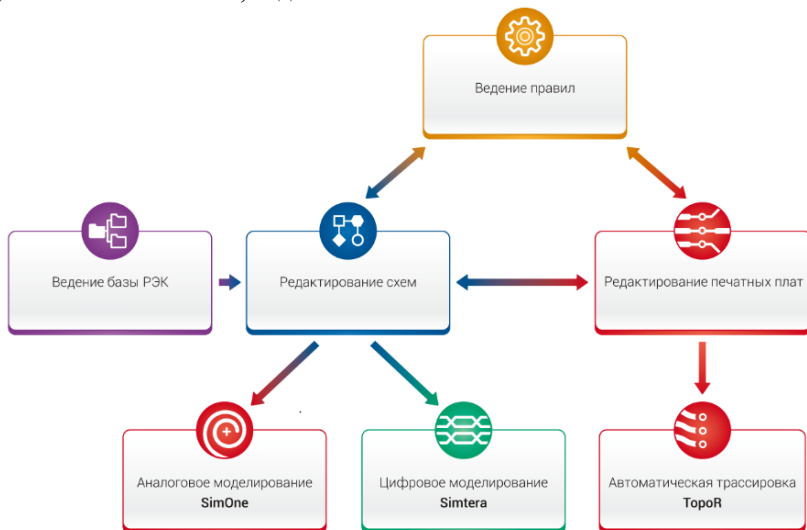


Рис. 1. Структурная схема модулей САПР Delta Design

Для описания компонента используются следующие данные:

- **Набор топологических посадочных мест (ТПМ).** ТПМ является отображением компонента на печатной плате, представленным в виде участка платы, на котором расположен компонент. Посадочные места могут быть заданы в строгом соответствии со стандартом IPC-7351. Для оптимизации работ по созданию ТПМ в программе предусмотрен мастер создания посадочных мест для основных типов корпусов
- **Набор условных графических обозначений (УГО)**
- **Описание контактов компонента.** Представление в виде таблицы, объединяющей информацию о всех контактах компонента и указывающая на структурное разделение компонента, для случаев, когда компонент содержит в одном корпусе несколько модулей (секций). С помощью данной таблицы задаются свойства контактов компонентов и определяется соответствие между контактом компонента, контактной площадкой на ТПМ и выводом на УГО.
- **Набор SPICE-моделей.** Программа позволяет задать для компонента SPICE-модель в виде текста и установить соответствие его контактов с элементами модели. В дальнейшем на баз схемы

может быть сформирован SPICE-нетлист для моделирования в модуле SimOne, вызываемом непосредственно из САПР с запуском моделирования, или другой программы, реализующей SPICE-моделирование с последующим сохранением файла на диск.

- **Технические характеристики компонента.** Список конкретных параметров определяется семейством, к которому принадлежит компонент. Значения параметров задаются для каждого конкретного компонента. Значения параметров конкретной технической характеристики (например, рабочее напряжение, номинал и т.д.) могут переданы для отображения на УГО и ТПМ.
- **Дополнительные данные.** Система обеспечивает повторное использование описания частей компонента для ускорения работы.

## Схемотехнический редактор

Функционал *схемотехнического редактора* (рис.2) позволяет проектировать электрические схемы в САПР Delta Design на основе выбираемых стандартов проектирования и оформления, что позволяет совместить процесс проектирования и выпуск документации. Для реализации данного подхода рабочее поле схемотехнического редактора представляется в виде отдельных листов, на которых выполняется построение схемы. Кроме того, в процессе создания схемы организован автоматический сбор данных для формирования документов вида «Спецификация», «Перечень элементов», «Ведомость покупных изделий».

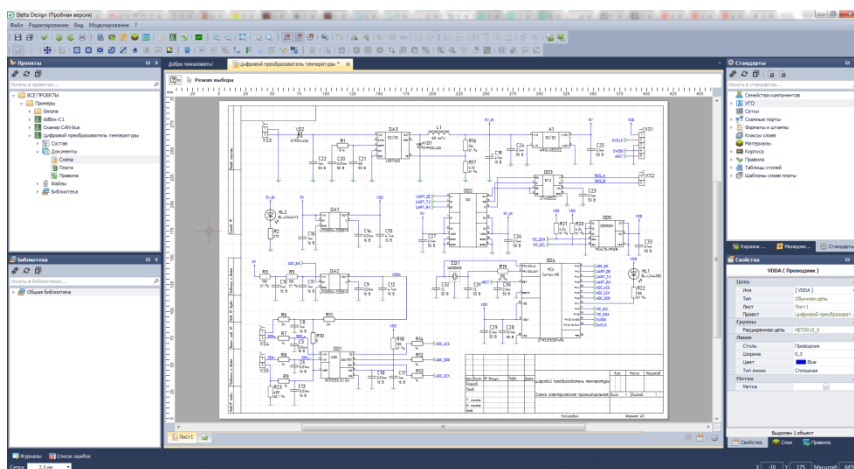


Рис. 2. Схемотехнический редактор в САПР Delta Design

Оформление листов схемы и документов осуществляется в строгом соответствии со стандартами, заданными для реализуемого проекта. В

системе доступны стандарты ГОСТ и ANSI, при этом проектировщик имеет возможность задать свои стандарты оформления, которые будут использоваться для выпуска документации. Такой подход способствует разделению схемы на функциональные части, позволяет улучшить восприятие разрабатываемых схем и уменьшить вероятность ошибки в выпускаемой документации.

САПР Delta Design предоставляет разработчику набор инструментов, обеспечивающий моделирование как аналоговых, так и цифровых процессов, проходящих в электрических схемах (функционалы SimOne и Simtera соответственно).

### Редактор правил

В *редакторе правил* осуществляется управлением правилами и технологическими ограничениями разрабатываемой печатной платы. Дерево иерархии правил формируется в зависимости от сложности проекта. Ведение правил осуществляется с использованием принципа наследования заданных параметров по иерархии цепей, слоев и регионов (областей) на плате. Каждый из уровней правил представляет собой отдельную таблицу с едиными принципами ведения данных.

Правила проектирования в САПР Delta Design состоят из нескольких групп:

- **Физические.** Определяют физические параметры треков: ширину, сужение, расположение T-соединения.
- **Зазоры.** Определяют предельные расстояния между различными объектами, размещаемыми на плате.
- **Трассировка.** Определяют разрешение на размещение цепей в различных зонах платы.

### Редактор плат

Функционал *редактора плат* позволяет создавать в САПР Delta Design печатные платы любого уровня сложности, при этом используется готовый справочник материалов. Размещение компонентов и прокладка трекгов осуществляются с непрерывным контролем правил проектирования. Для трассировки дифференциальных пар применяется дополнительный контроль зазоров и длин проводников. Такой подход позволяет точно определять свойства проектируемого изделия и в целом повысить эффективность разработки многослойных плат. Трассировка печатной платы может осуществляться в интерактивном или полностью автоматическом режиме.

## Автотрассировщик

*Автотрассировщик TopoR*, входящий в состав САПР Delta Design, обеспечивает существенное снижение протяженности параллельных трасс и уменьшает уровень перекрестных электромагнитных помех. Разводка проводников выполняется под произвольными углами и обеспечивает более экономичное использование коммутационного пространства. Эффективность автотрассировки превосходит результаты аналогов.

Система обеспечивает *выпуск конструкторской и производственной документации*. Формат производственной документации позволяет использовать ее на автоматических производственных линиях. Конструкторская документация выпускается в соответствии с выбираемыми проектировщиком стандартами (доступны стандарты оформления ГОСТ и ISO), помимо этого разработчик имеет возможность задать собственные стандарты, тем самым обеспечить любые требования, предъявляемые к конструкторской документации.

## Заключение

Приведённый обзор показал, что САПР Delta Design подходит для решения задач сквозного цикла проектирования печатных плат, разработки принципиальных электрических схем, моделирования работы аналоговых и цифровых устройств, разработки конструкции печатных плат, размещения компонентов и осуществления полуавтоматической и автоматической трассировки печатных плат, проектирования сложных электронных устройств в промышленных масштабах[2], а также формирования и выпуска конструкторской документации.

## Список литературы

1. DD - Delta Design [Электронный ресурс] – Режим доступа: <http://dd.ru/>, свободный.
2. Delta Design [Электронный ресурс] – Режим доступа: [http://cxem.net/software/delta\\_design.php](http://cxem.net/software/delta_design.php), свободный.
3. Войт Н.Н., Канев Д.С. Обзор современных САПР: P-CAD 2002, OrCAD 9.2, Quartus II 7.2, AutoCAD 2008, SolidWorks 2008, КОМПАС-3D V10 Plus // Информатика, моделирование, автоматизация проектирования: сборник научных трудов / под ред. Н. Н. Войта. – Ульяновск : УлГТУ, 2010. – С. 137-141.
4. Исследование и анализ возможностей программной среды SimOne, Delta Design, Symica, LTspice/SwitcherCAD, gEDA для разработки учебных виртуальных электронных лабораторий // Вестник КГУСТА – 2015. – №1(40). [Электронный ресурс] – Режим доступа: [https://elibrary.ru/download/elibrary\\_26723104\\_55459725.pdf](https://elibrary.ru/download/elibrary_26723104_55459725.pdf), свободный.

## ОСОБЕННОСТИ ГЕНЕРАЦИИ ЛАНДШАФТА В СРЕДЕ UNITY<sup>28</sup>

С.И. Бочков<sup>29</sup>

Для моделирования ситуаций на больших виртуальных участках зачастую требуется высокая точность расположения типовых объектов: деревьев, зданий и т.д. В статье рассматривается процесс автоматизированного воссоздания реального участка земли в виртуальном пространстве на примере игрового движка Unity.

### Введение

В настоящее время в процессах обучения, принятия решений активно используются виртуальные пространства, в которое добавляются имитационные модели объектов предметной области, с которыми взаимодействует обучающийся [3] или оператор.

Существуют программы, в которых используются полигоны огромных размеров, где природные факторы – рельеф, растительность, уровень воды – имеют критическое значение, например, при моделировании средств РЭБ. Большинство графических движков поддерживают редактирование ландшафта в режиме дизайна, однако с увеличением размеров полигона многократно возрастает сложность создания ландшафта и любая неточность может стоить дорого.

В игровом движке Unity 3D предусмотрены инструменты по автоматической генерации ландшафта по заданным параметрам. За отрисовку ландшафта в нём отвечает компонент Terrain, операции по изменению ландшафта связаны с объектом TerrainData, хранящем данные о ландшафте.

---

<sup>28</sup> Исследования поддержаны грантом Министерства образования и науки Российской Федерации, проект № 2.1615.2017/4.6.

Исследование выполнено при финансовой поддержке РФФИ и Правительства Ульяновской области в рамках научного проекта № 16-47-732152

Исследование выполнено при финансовой поддержке РФФИ в рамках научного проекта № 17-07-01417

<sup>29</sup> 432027, Ульяновск, ул. Северный Венец, 32, УлГТУ, e-mail: bochkovsam1@rambler.ru

## Построение рельефа

Данные о высотах ландшафта хранятся в двумерном массиве размером `heightmapWidth * heightmapHeight`, заполненном числами с плавающей запятой от 0 до 1; каждая ячейка означает долю от наибольшей заданной высоты `Terrain Height`. Наиболее распространённым и удобным способом генерации рельефа является использование карты высот. Данный вид карты представляет собой матрицу оттенков серого цвета; чем темнее пиксель, тем ниже данная точка. Пример карты высот показан на рис. 1.

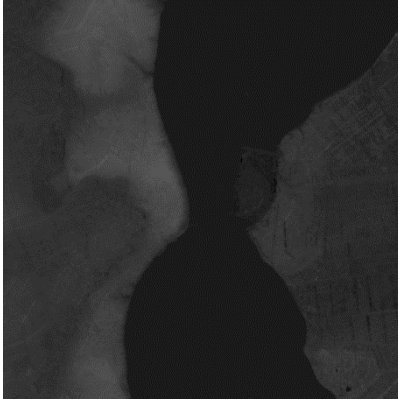


Рис. 3. Карта высот г. Ульяновска и окрестностей.

Традиционный размер карты высот  $2^n - 1, n \in \mathbb{Z}$ . Дело в том, что разрешение карты высот задаётся параметром `Heightmap Resolution`.

Для устранения лесенок используется простейший алгоритм сглаживания [5]. В листинге 1 показан процесс генерации рельефа.

Листинг 1. Генерация рельефа по карте высот.

```
// подгрузка карты высот
string filePath = "Assets/heightsmap2.png";
Texture2D poster = new Texture2D(2049, 2049, TextureFormat.RGB565,
false);
byte[] fileData = File.ReadAllBytes(filePath);
poster.LoadImage(fileData);
poster.Apply();
// сетка высот
float[,] heights = terrainData.GetHeights(0, 0, terrainData.heightmapWidth,
terrainData.heightmapHeight);
// чтение карты высот
for (int i = 0; i < poster.height; i++)
{
```



```

    for (int j = 0; j < poster.width; j++)
    {
        Color col = poster.GetPixel(i, j);
        heights[j, i] = col.grayscale;
    }
}
/* сглаживание */
// установка высот
terrainData.SetHeights(0, 0, heights);

```

## Генерация растительности

У компонента Terrain существует два вида объектов растительности: деревья и трава, причём последний означает не только траву в узком смысле, но и любой другой объект, представленный в виде растрового изображения (спрайта). В редакторе Unity такой объект называется «деталью», далее в статье понятие «деталь» будет вкладываться именно этот смысл. Следует также учесть, что при генерации растительности необходимо, чтобы в редактор ландшафта были загружены образцы деревьев и деталей, которые необходимо отобразить на полигоне.

Отличия одного вида растительности от другого состоят в следующем. Деревья в TerrainData представляют из себя объекты структуры TreeInstance[2], в которой указываются размеры дерева, позиция, поворот, порядковый номер из списка загруженных образцов и т.д. Они хранятся в массиве TreeInstance[].

Для каждого вида детали предусмотрены слои размером detailWidth \* detailHeight. Характеристики объекта детали можно задавать как непосредственно в редакторе ландшафта, так и в коде с помощью класса DetailPrototype[1]. В каждой ячейке слоя указывается целочисленная плотность детали.

Генерация деревьев и деталей приведена на листингах 2 и 3 соответственно.

Листинг 2. Генерация деревьев по карте растительности.

```

// Загрузка карты деревьев
string filePath = "Assets/map73.png";
Texture2D tMap = new Texture2D(950, 950);
byte[] fileData = File.ReadAllBytes(filePath);
tMap.LoadImage(fileData);
tMap.Apply();
// Создание шаблона дерева
TreeInstance ti = new TreeInstance();
ti.prototypeIndex = 1;
ti.heightScale = 1f;

```

```

ti.rotation = 0;
ti.widthScale = 1f;
// Создание списка деревьев
List<TreeInstance> instances = new List<TreeInstance>();
// Чтение карты деревьев
for (int i = 0; i < tMap.height; i++)
{
    for (int j = 0; j < tMap.width; j++)
    {
        if (/* условие */)
        {
            /* вычисление позиции дерева */
            ti.position = new Vector3(x, y, z);
            instances.Add(ti);
        }
    }
}
// Помещение списка в TerrainData
terrainData.treeInstances = instances.ToArray();

```

Листинг 3. Генерация деталей по карте растительности.

```

// получение слоя деталей
int[,] detailsGrass = terrainData.GetDetailLayer(0, 0,
terrainData.detailWidth, terrainData.detailHeight, 0);
// загрузка карты деталей
string filePath = "Assets/map73.png";
Texture2D tMap = new Texture2D(950, 950);
byte[] fileData = File.ReadAllBytes(filePath);
tMap.LoadImage(fileData);
tMap.Apply();
// чтение карты
float GrassmapToDetailsX = terrainData.detailWidth / tMap.width;
float GrassmapToDetailsZ = terrainData.detailHeight / tMap.height;
for (int i = 0; i < tMap.height; i++)
{
    for (int j = 0; j < tMap.width; j++)
    {
        Color pixel = tMap.GetPixel(i, j);
        if (/* условие */)
        {
            int x = Mathf.RoundToInt(i * GrassmapToDetailsX);
            int z = Mathf.RoundToInt(j * GrassmapToDetailsZ);

```

```

                                detailsGrass[z, x] = 16; // установка плотности
растительности
                                }
                                }
                                }
/* сглаживание */
terrainData.SetDetailLayer(0, 0, 0, detailsGrass);

```

## Нанесение текстур

Нанесение текстур на ландшафт осуществляется тем же способом, что и генерация деталей, с учётом некоторых особенностей. Данные о степени нанесения текстур хранятся в трёхмерном массиве размером  $\text{alphanapWidth} * \text{alphanapHeight} * \text{alphanapLayers}$ , где  $\text{alphanapLayers}$  – число текстур, загруженных в редактор ландшафта. Значение в каждой ячейке находится в отрезке  $[0; 1]$  и соответствует прозрачности той или иной текстуры.

Результат генерации ландшафта показан на рис. 2.

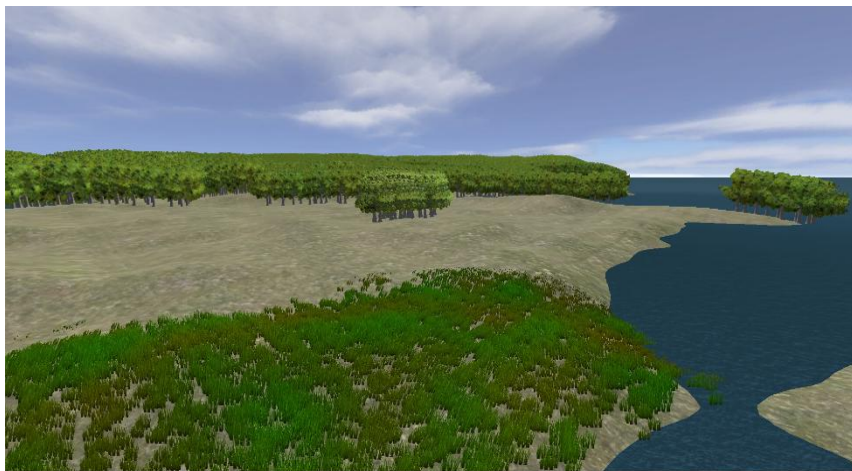


Рис. 4. Сгенерированный ландшафт, район Нижней Террасы.

## Заключение

Рассмотренные процедуры и возможности позволяют автоматизировать процесс генерации ландшафта на примере игрового движка Unity 3D.

Дополнительное преимущество такого способа состоит в том, что можно сгенерировать данные из исходных карт и сохранить их в файле на

стадии отладки, а при запуске скомпилированной версии проекта просто импортировать данные из этого файла.

Данный процесс можно усовершенствовать, дополнив его алгоритмом генерации гидрографических объектов и зданий.

### Список литературы

1. Unity - Scripting API: DetailPrototype [Электронный ресурс] – Режим доступа: <https://docs.unity3d.com/ScriptReference/DetailPrototype.html>, свободный.
2. Unity - Scripting API: TreeInstance [Электронный ресурс] – Режим доступа: <https://docs.unity3d.com/ScriptReference/TreeInstance.html>, свободный.
3. Афанасьев А. Н., Войт Н. Н., Молотов Р. С. Использование виртуальных симуляторов при формировании и проверке практических навыков обучающегося // Информатизация образования – 2017: сборник материалов Международной научно-практической конференции (Чебоксары, 15 июня – 17 июня 2017 года) / отв. ред. Н. В. Софронова. – Чебоксары : Чуваш. гос. пед. ун-т, 2017. – С. 95-99.
4. Афанасьев А. Н., Гульшин В. А. Войт Н. Н., Молотов Р. С. Концептуальное проектирование масштабируемой тренажерной системы специального назначения // Электронное обучение в непрерывном образовании 2017. IV Международная научно-практическая конференция (Россия, Ульяновск, 12 – 14 апреля 2017 г.): сборник научных трудов. – Ульяновск : УлГТУ, 2017. – С. 35– 42.
5. Генерация карты высот / Хабрахабр [Электронный ресурс] – Режим доступа: <https://habrahabr.ru/post/183986/>, свободный.

## АНАЛИЗ ЭКСПЕРТНЫХ СИСТЕМ<sup>30</sup>

С. И. Бригаднов<sup>31</sup>

В работе рассмотрен обзор современных экспертных систем, произведен их анализ, рассмотрена структура и функционирование наиболее распространенных экспертных систем.

### Введение

Экспертные системы с середины 1980-х годов стали широко разрабатываться и практически использоваться для решения трудно формализуемых и плохо структурируемых задач в различных сферах деятельности человека: медицина, военное дело, вычислительная техника, промышленность и др.

Экспертные системы развивались как новая информационная технология, позволяющая при решении задач преобразовывать как данные, так и знания.

Экспертные системы (ЭС) явились результатом практической реализации теории искусственного интеллекта (ИИ), применение которых существенно расширило область использования вычислительной техники.

ЭС состоят из следующих частей: процедур логического вывода, принятия решений и базы знаний. Последняя отличается от базы данных тем, что она является активной, т.е. она передаёт не только знания, но и опыт экспертов. И здесь выявляется основное требование при создании ЭС: правильно выбранный эксперт, передающий свои знания. Именно он и метод формализации знаний определяют 90% качества системы [1].

Далее будут рассмотрены типовая структура экспертной системы и наиболее распространенные примеры экспертных систем.

---

<sup>30</sup> Исследования поддержаны грантом Министерства образования и науки Российской Федерации, проект № 2.1615.2017/4.6.

Исследование выполнено при финансовой поддержке РФФИ и Правительства Ульяновской области в рамках научного проекта № 16-47-732152

Исследование выполнено при финансовой поддержке РФФИ в рамках научного проекта № 17-07-01417

<sup>31</sup> 432027, Ульяновск, ул. Северный Венец, 32, УлГТУ, e-mail: sergbrig@yandex.ru

# 1. Структура экспертной системы

Типовая структура экспертной системы представлена на рис. 1.

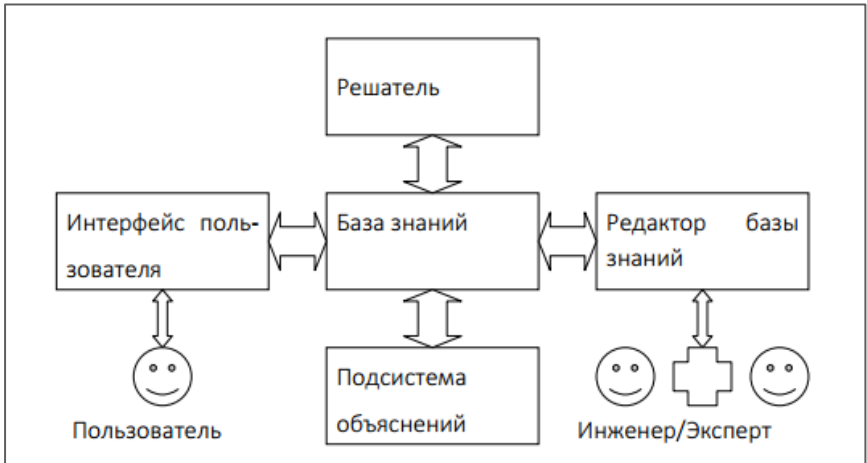


Рис. 1. Типовая структура ЭС

База знаний содержит в себе правила анализа информации по проблеме, полученной от пользователя. ЭС анализирует эту информацию и дает рекомендации по разрешению конкретной проблемы.

База знаний состоит из двух составляющих:

- факты — статические сведения о предметной области;
- правила — набор инструкций, который позволяет выводить новые факты, исходя из уже известных.

В рамках логической модели базы знаний формируются на языке Пролог с помощью предикатов для описания фактов и правил логического вывода [2].

Факты в базе знаний описывают те явления, которые являются постоянными для данной предметной области. Данные, зависящие от условий конкретной задачи, ЭС получает от пользователя в процессе работы и сохраняет в рабочей памяти.

База знаний ЭС создается при помощи трех групп людей:

- эксперты той проблемной области, к которой относятся задачи, решаемые ЭС;
- инженеры по знаниям, являющиеся специалистами по разработке ИИС;
- программисты, осуществляющие реализацию ЭС.

## 2. Обзор распространенных экспертных систем

**CODES.** Экспертная система помогает разработчику базы данных, желающему использовать подход IDEF1 для определения концептуальной схемы базы данных. Хотя в качестве подхода IDEF1 полезна, сложность ее правил часто сдерживает ее применение. Разработчик описывает, какие свойства и взаимосвязи желательны в базе данных, под руководством системы CODES, осуществляемым в форме диалога. Затем система применяет свои знания в виде правил и эвристик IDEF1 для построения концептуальной схемы разрабатываемой базы данных. Знания в CODES представлены в виде правил с применением обратной цепочки рассуждений в качестве стратегии управления. CODES реализована на языке UCI LISP. Она была разработана в Университете штата Южная Калифорния и доведена до уровня демонстрационного прототипа [3].

**MIXER.** Экспертная система оказывает помощь программистам в написании микропрограмм для разработанной Texas Instruments СБИС TI990. По заданному описанию микропрограммы система получает оптимизированные микропрограммы для TI990. MIXER содержит знания по микропрограммированию для TI990, взятые из руководства и из анализа микропрограммы управляющего ПЗУ TI990. Сюда относятся знания о том, как преобразовывать введенные описания в наборы промежуточных операций, как выделить соответствующие регистры под переменные и как преобразовать промежуточные операции в наборы микроопераций. MIXER использует эти знания, чтобы определить, какие микрооперации являются лучшими для реализации микропрограммы. Система представляет знания в виде правил и данных, обладает унификацией, управляемой механизмом вывода, и динамическим возвратом. MIXER реализована на языке Пролог. Она была разработана в Токийском университете и доведена до уровня демонстрационного прототипа [3].

**ACE.** Экспертная система определяет неисправности в телефонной сети и дает рекомендации по необходимому ремонту и восстановительным мероприятиям. Система работает без вмешательства пользователя, анализируя сводки-отчеты о состоянии, получаемые ежедневно с помощью CRAS, программы, следящей за ходом ремонтных работ в кабельной сети. ACE обнаруживает неисправные телефонные кабели и затем решает, нуждаются ли они в планово-предупредительном ремонте и выбирает, какой тип ремонтных работ вероятнее всего будет эффективным. Затем ACE запоминает свои рекомендации в специальной базе данных, к которой у пользователя есть доступ. Система принимает решения, применяя знания относительно телефонных станций, сообщения

системы CRAS и стратегии анализа сетей. Представление знаний в системе основано на правилах, используется схема управления посредством прямой цепочки рассуждений. ACE реализована на языках OPS4 и FRANZ LISP и работает на микропроцессорах серии AT&T 3B-2, размещенных в подстанциях наблюдения состояния кабеля. Она разработана в Bell Laboratories. ACE прошла опытную эксплуатацию и доведена до уровня коммерческой экспертной системы [3].

**Экспертная система MATHLAB 68.** Данная экспертная система помогает математикам, ученым и инженерам выполнять символьные преобразования алгебраических выражений, встречающиеся в задачах математического анализа [4]. Система выполняет дифференцирование, разложение многочленов на множители, вычисление неопределенных интегралов, прямое и обратное преобразования Лапласа, а также решает линейные дифференциальные уравнения с постоянными (числовыми и символьными) коэффициентами. Она содержит математические знания в виде отдельных модулей, каждый из которых специализирован на выполнении определенной функции. Данные пользователя классифицируются на три категории: выражения, уравнения и функции. Правила алгебраических преобразований данных меняются в зависимости от категории. Система реализована на языке Лисп и послужила краеугольным камнем при разработке системы MACSYMA. MATHLAB 68 разработана в Массачусетском технологическом институте и доведена до уровня исследовательского прототипа.

**RUBRIC.** Экспертная система помогает пользователю получить доступ к базам данных, содержащим неформатированные тексты. Например, когда пользователь называет какую-нибудь тему, RUBRIC автоматически разыскивает все документы, содержащие тексты, связанные с этой темой. В системе RUBRIC взаимоотношения между темами, подтемами и фразами, содержащими ключевые слова, выражены в виде правил. Правила также определяют другие варианты терминов, выражений и способов написания одной и той же темы или понятия. Пользователь может сформулировать запрос в виде правила, задающего критерий поиска, например эвристический вес, определяющий насколько сильно образец правила указывает на наличие темы правила. В ходе поиска RUBRIC предоставляет пользователю документы, которые лежат в кластере, содержащем по крайней мере один документ с весом выше заданного пользователем порога. Это предотвращает ситуацию, в которой произвольно выбранный порог мог бы разделить близкие по рангу документы. Система реализована на языке FRANZ LISP, разработана компанией «Advanced Information & Decision Systems» и доведена до уровня исследовательского прототипа [3].



**OpenCyc (open source version of the Cyc technology).** OpenCyc является сокращенным открытым вариантом базы знаний Cyc. Может использоваться для создания экспертных систем. В БД OpenCyc содержится 47000 понятий и 300000 фактов. Система несомненно будет уместна, вместе с появлением в сети Интернет семантического уровня (обработки запросов, системы взаимосвязи контента и т.д.) В 2008 интерес к данной экспертной системе был проявлен фирмой Google. Применение OpenCyc многообразно: создание обучающих игр, создание БЗ в wikipedia, подключение в качестве модуля к другим экспертным системам и т.д.

**CLIPS.** Суть технологии CLIPS заключается в том, что язык и среда CLIPS предоставляют пользователям возможность быстро создавать эффективные, компактные и легко управляемые экспертные системы. При этом пользователь применяет множество уже готовых инструментов (встроенный механизм управления базой знаний, механизм логического вывода, менеджеры различных объектов CLIPS и т. д.) и конструкций (упорядоченные факты, шаблоны, правила, функции, родовые функции, классы, модули, ограничения, встроенный язык COOL и т. д.). Так же CLIPS предоставляет возможность разбиения базы данных и решения задачи на отдельные независимые модули. Экспертная система CLIPS применяется как в системах, ориентированных на различные игры, так и в нечётких системах для формализации знаний.

**ExpSystem PC 1.4.2** - Эта экспертная система поможет людям не столь хорошо разбирающимся в комплектующих для персонального компьютера подобрать себе приемлемую конфигурацию будущего компьютера. В программу вводятся данные о требовании к системе и максимальной сумме денег, предполагаемой для покупки. Так же данная экспертная система пригодится для менеджеров компьютерных салонов что бы автоматизировать свою работу и сделать автоматизированное рабочее место для своих покупателей. Для подбора комплектующих используется два алгоритма расчета: для "богатого" и "бедного" покупателя. Данная ЭС была разработана в декабре 2006 года.

**APM WinMachine.** Инструментально-экспертная система APM WinMachine представляет собой энциклопедию по машиностроению, включающую инструменты и программы для автоматизированного расчета и проектирования деталей машин, механизмов, элементов конструкций и узлов. Кроме этого, она имеет современные графические средства, встроенные базы данных, необходимую информационную базу знаний, разветвленную систему подсказок и фундаментальный электронный учебник по основам проектирования машин. APM

WinMachine содержит современные, эффективные и надежные программы для расчета:

- энергетических и кинематических параметров;
- прочности, жесткости и устойчивости;
- выносливости при переменных режимах нагружения;
- вероятности, надежности и износостойкости;
- динамических характеристик.

Кроме того, в APM WinMachine имеется набор инструментальных средств расчета и анализа [5]. Эти средства, а также проектируемые детали, в зависимости от назначения разделены на подсистемы (модули), которые могут функционировать как в составе системы, так и самостоятельно:

1. **APM Graph** - плоский чертежно-графический редактор с инструментом расчета размерных цепей
2. **APM Studio** - модуль твердотельного и поверхностного моделирования
3. **APM Mechanical Data** - модуль хранения стандартных деталей и узлов, расчетных данных по машиностроению
4. **APM Material Data** - модуль хранения и редактирования параметров материалов
5. **APM Construction Data** - модуль хранения графической информации по стандартным деталям и элементам строительных конструкций
6. **APM Drive** - модуль автоматизированного проектирования привода вращательного движения произвольной структуры
7. **APM Trans** - модуль проектирования механических передач вращения
8. **APM Shaft** - модуль проектирования валов и осей
9. **APM Bear** - модуль проектирования подшипниковых узлов качения с учетом их класса точности
10. **APM Joint** - модуль проектирования соединений элементов машин
11. **APM Spring** - модуль проектирования упругих элементов машин

12. **APM Cam** - модуль проектирования кулачковых механизмов
13. **APM Slider** - модуль проектирования плоских механизмов произвольной структуры
14. **APM Plain** - модуль расчета подшипников скольжения
15. **APM Screw** - модуль расчета неидеальных винтовых передач (скольжения, шарико- и планетарно-винтовых)
16. **APM Structure3D** - модуль расчета напряженно-деформированного состояния, устойчивости, собственных и вынужденных колебаний деталей и конструкций с использованием метода конечных элементов
17. **APM Beam** - модуль расчета балочных элементов конструкций
18. **APM FEM2D** - модуль конечно-элементного анализа плоских деталей
19. **APM Truss** - модуль расчета ферменных конструкций
20. **MDM (Основы проектирования машин)** - электронный учебник

Каждый модуль предоставляет пользователю интегрированную среду, которая в общем случае включает в себя:

- специализированный графический редактор;
- встроенные базы данных;
- полный цикл вычислений;
- разнообразные средства представления результатов расчета.

**Экспертная система EXSYS.** Данная экспертная система используется в различных отраслях промышленности, т.к. является программной оболочкой, которая может быть как диалоговой (непосредственные ответы на вопросы пользователя), так и вложенной, где все исходные данные поступают от другой программы. Диапазон проблем, которые могут быть решены экспертной системой, обширен [6].

Экспертные системы могут быть разработаны в EXSYS для любой проблемной области, в которой для решения какой-либо задачи необходимо сделать выбор среди определенного набора вариантов, а процесс достижения этого решения основан на логических шагах. Любая проблемная область, где человек или группа людей имеют специальные экспертные знания, необходимые другим, является возможной областью применения ПО EXSYS . Экспертные системы могут помочь

автоматизировать выполнение сложных инструкций, выбрать какое-либо изделие из группы изделий или диагностировать оборудование.

Инструменты программного обеспечения EXSYS используются для создания вероятностных, основанных на знаниях, экспертных систем. Компания EXSYS полагает, что наиболее эффективный путь построения экспертных систем состоит в том, чтобы обеспечить экспертов средствами, которые позволили бы им самим создавать эти системы. Результаты такой политики говорят за себя – ПО EXSYS успешно работает в широком диапазоне реальных проблем, с более чем 11000 пользователей во всем мире. EXSYS-программы рассчитаны на то, что экспертные системы будут создаваться экспертами проблемной среды совместно с профессионалами в области построения инженерных знаний

### **Заключение**

Современные технологии на производстве не могут обойтись без использования персональных компьютеров, соответственно применение экспертных систем становится все более востребованным, начиная с простейших систем диагностирования и заканчивая сложными системами диагностики, контроля и наладки оборудования.

### **Список литературы**

1. Брызгалин В.В. Вечкина А.В. Грачева Е.В. СОВРЕМЕННЫЕ ЭКСПЕРТНЫЕ СИСТЕМЫ // Успехи современного естествознания. – 2012. – № 6 – С. 85-86
2. Дошина А. Д. Экспертная система. Классификация. Обзор существующих экспертных систем // Молодой ученый. — 2016. — №21. — С. 756-758.
3. Примеры экспертных систем [Электронный ресурс] – Режим доступа: <http://www.aiportal.ru/articles/expert-systems/examples-expsys.html> , свободный.
4. Нейлор К. Как построить свою экспертную систему: пер. с англ. М.:ЭНЕРГОАТОМИЗДАТ, 1991
5. Прочностной расчет и проектирование конструкций, деталей машин и механизмов (APM WinMachine) [Электронный ресурс] – Режим доступа: [http://apm.ru/produkti/programmie\\_kompleksi/APM\\_WinMachine](http://apm.ru/produkti/programmie_kompleksi/APM_WinMachine) , свободный.
6. Корухова Л.С., Любимский Э.З., Мальшко В.В. Программные средства реализации ассоциативного планирования: препринт ИПМ им. М.В.Келдыша РАН №10. М., 2002.

# НЕЙРОННЫЕ СЕТИ В ЭЛЕКТРОННОМ ОБУЧЕНИИ АВТОМАТИЗИРОВАННОМУ ПРОЕКТИРОВАНИЮ<sup>32</sup>

Е.Ю. Воеводин<sup>33</sup>

В статье описывается подход к повышению эффективности электронного обучения автоматизированному проектированию используя нейронные сети.

## Введение

Улучшение параметров освоения материала всегда являлось важным мотивирующим фактором в образовательном процессе[1]. На итоговую успеваемость обучаемого влияют многие факторы, такие как:

- Вовлеченность в образовательный процесс. На этот фактор могут влиять такие параметры как текущие самочувствие студента, его настроение, мотивация в изучении предмета и другие психоэмоциональные величины.
- Дисциплинированность. На этот фактор влияют особенности характера обучаемого.
- Взаимодействие с другими обучаемыми в группе. На этот фактор влияют как характер обучаемого, так и его взаимоотношения с другими участниками группы
- Опыт преподавателя и участие во время обучения со студентами.

## 1. Описание нейронной сети и набор данных

Согласно Эдельштейну [2], существует пять распространенных видов информации: последовательности, ассоциации, классификации, кластеры и прогнозирование. Таким образом, общая цель нейронных сетей заключается в предоставлении аппарата решений для задач ассоциирования, классификации, кластеризации и прогнозирования.

---

<sup>32</sup> Исследования поддержаны грантом Министерства образования и науки Российской Федерации, проект № 2.1615.2017/4.6.

Исследование выполнено при финансовой поддержке РФФИ и Правительства Ульяновской области в рамках научного проекта № 16-47-732152

Исследование выполнено при финансовой поддержке РФФИ в рамках научного проекта № 17-07-01417

<sup>33</sup> Ульяновск, ул. Северный Венец, 32, УлГТУ, e-mail:voevodineu@gmail.com

Более того, нейронные сети обладают возможностью использовать собственный опыт с целью улучшения своих показателей с помощью адаптации.

Нейронные сети также способны распознавать неточную информацию (неполные или шумные данные) и могут быть очень эффективными, в частности, когда невозможно определить шаблоны или шаги, которые необходимо предпринять для решения задачи. Таким образом, отличительной чертой нейронной сети является способность обучения выполнению задачи, в случае, если метод ее решения не представляется явным или не поддается математическому описанию.

В данном исследовании для изучения эффективности процесса обучения используется нормализованная матрица из 25 наиболее часто используемых категоризированных переменных (признаков или предикторов). Для достижения более высоких уровней вычислительной мощности требуется чрезвычайно сложная структура нейронных сетей. В данной статье описывается опыт использования многослойной нейронной сети на основе перцептрона, которая отображает набор входных переменных на набор выходных данных. Он состоит из нескольких слоев узлов ориентированного графа, на котором каждый слой подключен к следующему. Это один из самых популярных и успешных методов для многих задач таких как анализ содержимого, распознавание образов, прогнозирование временных рядов, диагностика и др.

## **2. Категоризированные предикторы**

Обучение сети производилось методом обратного распространения ошибки. Оценка ошибки производилась методом обратного среднего. Количество нейронов среднего слоя: 50. Категоризированные предикторы, предлагаемые к использованию, представлены в таблице 1:

Таблица 1. Категории и предикторы

Категория	Предиктор	Категория	Предиктор
Вовлечение	Аутентификация	Использование контента	Количество использований поиска и ключевые слова
	Участие в модулях совместной активности		Количество комментирований
	Участие в модулях автоматической оценки (тест)		Контекст и типы вопросов теста
	Баллы модулей автоматической оценки по попыткам		Среднее время ответа на вопрос
Раздражение	Количество сообщений внутр. почты		
Самооценка	Скука	Содержимое курса	Количество модулей (по типам)
	Беспомощность		Количество вопросов
	Эффективность		Уровень персонализации
	Уровень амбиций	Внешние воздействия	Количество полученных сообщений
	Организованность		Уровень аптайма сервера
Предпосылки	Уровень претеста	Внешние воздействия	Ширина канала
	Средний уровень оценок		Общее число затраченных часов
	Скорость изменения оценок (стабильность)		

### 3. Результаты обучения сети

Обучение сети производилось на выборке из 1315 наборов параметров. Валидация и тестирование сети производилось на 282 разных наборах параметров. На 18 эпохе обучения изменение ошибок валидации перестало быть значительным, как показано на рис. 1. Средняя ошибка на этапе обучения составила 0.18, на этапе валидации: 0.33, на этапе тестирования: 0.22.

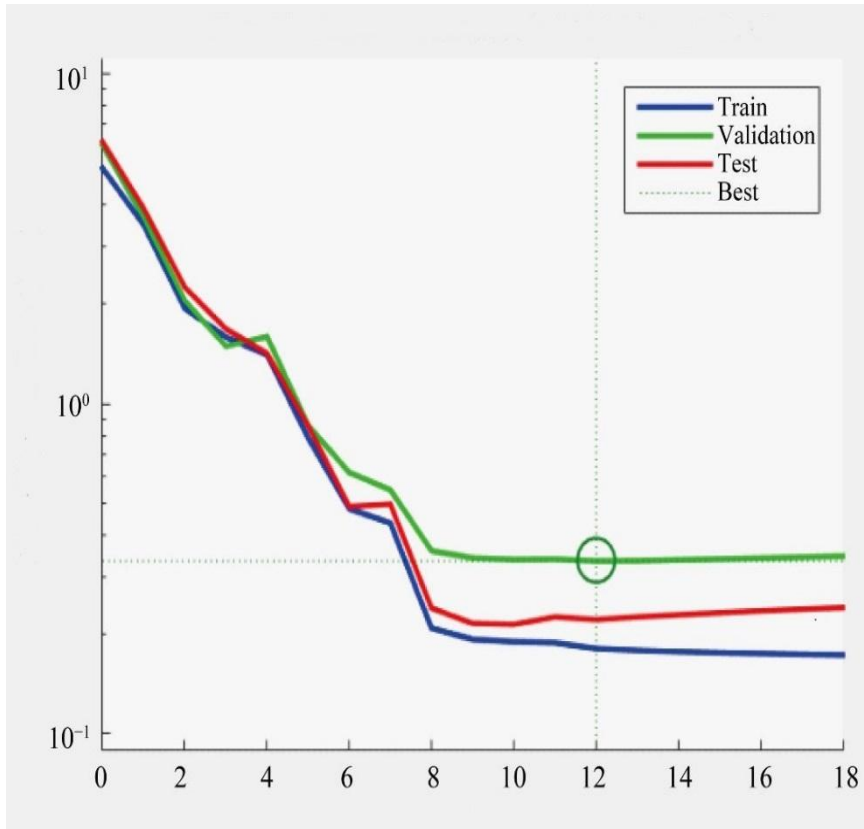


Рис. 1. Изменение ошибки (стоимость сети) при обучении, валидации и тестировании



## Заключение

Метод обратного распространения ошибки дает слабые результаты настройки весов даже на небольших сетях без использования автокодировщика или других обучающих подсетей или является слишком ресурсозатратным. Следующим этапом проектирования станет подбор и сравнительный анализ обучающих подсетей для первоначальной настройки весов.

## Список литературы

1. Афанасьев А.Н., Войт Н.Н. [РАЗРАБОТКА КОМПОНЕНТНО-СЕРВИСНОЙ ПЛАТФОРМЫ ОБУЧЕНИЯ: АНАЛИЗ И РАЗРАБОТКА КОМПОНЕНТА МЕТОДА ДИАГНОСТИКИ ПРОЕКТНЫХ ХАРАКТЕРИСТИК ОБУЧАЕМОГО ИНЖЕНЕРА С ПОМОЩЬЮ ДИАГРАММ UML](#) // [Вестник Ульяновского государственного технического университета](#). 2012. № 4 (60). С. 43-46.
2. Edelstein, H.A. (1999) Introduction to Data Mining and Knowledge Discovery. 3rd Edition, Crows Corporation, Potomac.

УДК 004.946

## МАТЕМАТИЧЕСКОЕ МОДЕЛИРОВАНИЕ ПРОЦЕССА ЗАПОМИНАНИЯ МАТЕРИАЛА ПРОЕКТИРОВЩИКОМ<sup>34</sup>

Н.Н. Войт<sup>35</sup>, И.В. Беляева<sup>36</sup>

Выполнено моделирование процесса накопления знаний в интервале времени. Предложен метод расчета объема освоенного материала проектировщиком в определенный момент времени.

### Введение

Готовность проектировщика в определенный момент времени к решению поставленных задач (оперативных единиц деятельности) является значимой информацией для руководства структурного подразделения с целью распределения этих задач в виде проектных потоков между проектировщиками, а также важным фактором, влияющим на качество и сроки выполнения работ [1-5].

### Разработка концептуальной модели процесса обучения проектировщика

Интенсивность познания intensityCognizable определяет интенсивность изучения нового материала проектировщиком (оперативных единиц деятельности – ОЕД). Интенсивность забывания intensityFoget определяет интенсивность забывания изученных ОЕД. Предлагается их оценить количественно с помощью разработанной концептуальной модели процесса обучения (рис. 1).

---

<sup>34</sup> Исследование выполнено при финансовой поддержке РФФИ и Правительства Ульяновской области в рамках научного проекта № 16-47-732152  
Исследования поддержаны грантом Министерства образования и науки Российской Федерации, проект № 2.1615.2017/4.6

<sup>35</sup> 432027, Ульяновск, ул. Северный Венец, 32, УлГТУ, e-mail: n.voit@ulstu.ru

<sup>36</sup> 432027, Ульяновск, ул. Северный Венец, 32, УлГТУ, e-mail: belyaeva\_iv@mail.ru

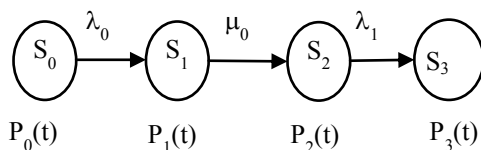


Рис. 1. Концептуальная модель процесса обучения проектировщика

Интерпретация представленного графа следующая. Проектировщик с вероятностью  $P_0(t)$  находится в состоянии  $S_0$ , в котором ему нужно изучить новые ОЕД с интенсивностью познания  $\nu_0$ . Когда ОЕД изучены, проектировщик с вероятностью  $P_1(t)$  находится в состоянии  $S_1$ . Далее проектировщик с интенсивностью  $\mu_0$  начинает забывать изученные ОЕД и с вероятностью  $P_2(t)$  переходит в состояние  $S_2$ . Повторение изученного ОЕД с целью восстановления забытых знаний проходит с интенсивностью повторения  $\nu_1$  и проектировщик с вероятностью  $P_3(t)$  переходит в состояние  $S_3$ . Степень готовности проектировщика в момент времени  $t$  определена через функцию  $READY(t) = P_1(t) + P_3(t)$ , т.к. именно состояния  $S_1$  и  $S_3$  представляют вероятности знания ОЕД в момент времени  $t$ , а  $S_0$ ,  $S_2$  – «начальное» состояние и остаточность знаний ОЕД после времени забывания.

### Вывод функции оценки готовности проектировщика к решению задач

Разработанную графовую модель процесса обучения можно представить в динамике дифференциальными уравнения от  $t$  следующим образом (1): производная вероятности любого состояния равна сумме потоков вероятностей, переводящих систему в это состояние, минус сумма всех потоков вероятностей, выводящих систему из этого состояния [6].

$$\begin{cases} \frac{dP_0(t)}{dt} = -\lambda_0 P_0(t), \\ \frac{dP_1(t)}{dt} = \lambda_0 P_0(t) - \mu_0 P_1(t), \\ \frac{dP_2(t)}{dt} = \mu_0 P_1(t) - \lambda_1 P_2(t), \\ \frac{dP_3(t)}{dt} = \lambda_1 P_2(t). \end{cases} \quad (1)$$

Систему (1) можно решить с помощью преобразований Лапласа (прямых и обратных), представленных в таблице 1.

Таблица 1. Преобразование Лапласа для системы уравнений [7]

Функция	Изображение функции
$P_i(t)$	$F_i(s)$
$\frac{dP_i(t)}{dt}$	$s \cdot F_i(s) - F_0(s)$
$e^{-\lambda_0 t}$	$\frac{1}{s + \lambda_0}$
$\lambda_0 \frac{e^{-\lambda_0 t} - e^{-\mu_0 t}}{\mu_0 - \lambda_0}$	$\frac{\lambda_0}{(s + \lambda_0)(s + \mu_0)}$
$\lambda_0 \mu_0 \left( -\frac{e^{-\lambda_1 t}}{(\lambda_0 - \lambda_1)(\lambda_1 - \mu_0)} + \frac{e^{-\lambda_0 t}}{(\lambda_0 - \lambda_1)(\lambda_0 - \mu_0)} - \frac{e^{-\mu_0 t}}{(\mu_0 - \lambda_1)(\lambda_0 - \mu_0)} \right)$	$\frac{\lambda_0 \mu_0}{(s + \lambda_0)(s + \mu_0)(s + \lambda_1)}$
$\lambda_0 \lambda_1 \mu_0 \left( \frac{1}{\lambda_0 \lambda_1 \mu_0} + \frac{e^{-\lambda_1 t}}{(\lambda_0 - \lambda_1) \lambda_1 (\lambda_1 - \mu_0)} - \frac{e^{-\lambda_0 t}}{\lambda_0 (\lambda_0 - \lambda_1) (\lambda_0 - \mu_0)} + \frac{e^{-\mu_0 t}}{\mu_0 (\mu_0 - \lambda_1) (\lambda_0 - \mu_0)} \right)$	$\frac{\lambda_0 \mu_0 \lambda_1}{s(s + \lambda_0)(s + \mu_0)(s + \lambda_1)}$

Пусть  $F_i(0) = \begin{cases} 1, & \text{если } i = 0, \\ 0, & \text{если } i > 0. \end{cases}$  Система уравнений (1) после преобразований примет следующий вид (2).

$$\begin{cases} sF_0(s) - 1 = -\lambda_0 F_0(s), \\ sF_1(s) = \lambda_0 F_0(s) - \mu_0 F_1(s), \\ sF_2(s) = \mu_0 F_1(s) - \lambda_1 F_2(s), \\ sF_3(s) = \lambda_1 F_2(s). \end{cases} \quad (2)$$

Переносим  $F_i(s)$  в левую часть от равенства в системе (2), получим систему относительно  $F_0(s), F_1(s), F_2(s), F_3(s)$  (3).

$$\begin{cases} F_0(s) = \frac{1}{s + \lambda_0}, \\ F_1(s) = \frac{\lambda_0}{(s + \lambda_0)(s + \mu_0)}, \\ F_2(s) = \frac{\lambda_0 \mu_0}{(s + \lambda_0)(s + \mu_0)(s + \lambda_1)}, \\ F_3(s) = \frac{\lambda_0 \mu_0 \lambda_1}{s(s + \lambda_0)(s + \mu_0)(s + \lambda_1)}. \end{cases} \quad (3)$$

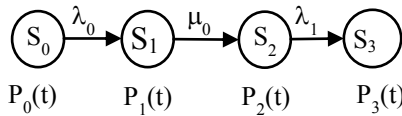
Применяя к системе (3) обратные преобразования Лапласа (таблица 1), получим  $P_0(t)$ ,  $P_1(t)$ ,  $P_2(t)$ ,  $P_3(t)$ , представленные в системе (4).

$$\left\{ \begin{array}{l} P_0(t) = e^{-\lambda_0 t}, \\ P_1(t) = \lambda_0 \frac{e^{-\lambda_0 t} - e^{-\mu_0 t}}{\mu_0 - \lambda_0}, \\ P_2(t) = \lambda_0 \mu_0 \left( -\frac{e^{-\lambda_1 t}}{(\lambda_0 - \lambda_1)(\lambda_1 - \mu_0)} + \frac{e^{-\lambda_0 t}}{(\lambda_0 - \lambda_1)(\lambda_0 - \mu_0)} - \frac{e^{-\mu_0 t}}{(\mu_0 - \lambda_1)(\lambda_0 - \mu_0)} \right), \\ P_3(t) = \lambda_0 \lambda_1 \mu_0 \left( \frac{1}{\lambda_0 \lambda_1 \mu_0} + \frac{e^{-\lambda_1 t}}{(\lambda_0 - \lambda_1) \lambda_1 (\lambda_1 - \mu_0)} - \frac{e^{-\lambda_0 t}}{\lambda_0 (\lambda_0 - \lambda_1) (\lambda_0 - \mu_0)} + \frac{e^{-\mu_0 t}}{\mu_0 (\mu_0 - \lambda_1) (\lambda_0 - \mu_0)} \right). \end{array} \right. \quad (4)$$

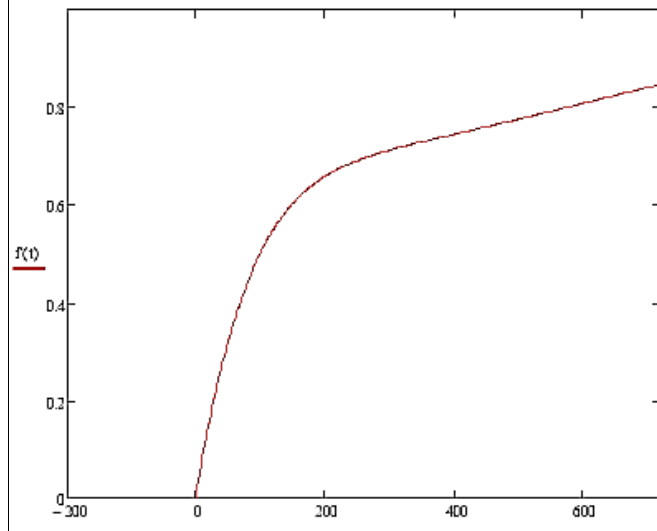
Следовательно, функция готовности проектировщика к решению задачи в момент времени  $t$  имеет следующий вид

$$\text{READY}(t) = \lambda_0 \frac{e^{-\lambda_0 t} - e^{-\mu_0 t}}{\mu_0 - \lambda_0} + \lambda_0 \lambda_1 \mu_0 \left( \frac{1}{\lambda_0 \lambda_1 \mu_0} + \frac{e^{-\lambda_1 t}}{(\lambda_0 - \lambda_1) \lambda_1 (\lambda_1 - \mu_0)} - \frac{e^{-\lambda_0 t}}{\lambda_0 (\lambda_0 - \lambda_1) (\lambda_0 - \mu_0)} + \frac{e^{-\mu_0 t}}{\mu_0 (\mu_0 - \lambda_1) (\lambda_0 - \mu_0)} \right) \quad (5)$$

Пример.



Дано	Решение
$\lambda_0 = \frac{1}{120} \text{ день}^{-1}$ $\lambda_1 = \frac{1}{180} \text{ день}^{-1}$ $\mu_0 = \frac{1}{365} \text{ день}^{-1}$ $t \in [0; 720]$	$\text{READY}(t) = P_1(t) + P_3(t)$ . Подставляя значения $t$ , $\lambda_0$ , $\lambda_1$ , $\mu_0$ в формулу (5) [8], получаем: $\text{READY}(t) =$ $\frac{1}{120} * \frac{e^{-\frac{1}{120}t} - e^{-\frac{1}{365}t}}{\frac{1}{365} - \frac{1}{120}} + \frac{1}{120} * \frac{1}{180} * \frac{1}{365} \left( \frac{1}{\frac{1}{120} * \frac{1}{180} * \frac{1}{365}} + \frac{e^{-\lambda_1 t}}{(\lambda_0 - \lambda_1) \lambda_1 (\lambda_1 - \mu_0)} - \frac{e^{-\lambda_0 t}}{\lambda_0 (\lambda_0 - \lambda_1) (\lambda_0 - \mu_0)} + \frac{e^{-\mu_0 t}}{\mu_0 (\mu_0 - \lambda_1) (\lambda_0 - \mu_0)} \right)$ $e^{-1180t} 1120 - 1180 * 1180$ $*(1180 - 1365) - e^{-1120t} 1120 * (1120 - 1180) * (1120 - 1365) + e^{-1365t} 1365 * (1365 - 1180) * (1120 - 1365)$
Построить график $\text{READY}(t)$	График функции $\text{READY}(t)$ имеет следующий вид



## Заключение

Разработанный метод может быть использован при разработке систем обучения в плане оценки (прогноза) качества обучения проектировщиков в условиях крупного проектного предприятия.

## Список литературы

1. Афанасьев А.Н., Войт Н.Н. Разработка компонентно-сервисной платформы обучения: диаграммы классов программного компонента сценария на UML-языке // Вестник Ульяновского государственного технического университета. - 2012. - № 2 (58). - С. 32-36.
2. Войт Н.Н. Разработка методов и средств адаптивного управления процессом обучения в автоматизированном проектировании // диссертация на соискание ученой степени кандидата технических наук / Ульяновский государственный технический университет. Ульяновск, 2009.
3. Афанасьев А.Н., Войт Н.Н. Разработка компонентно-сервисной платформы обучения: анализ и разработка компонента метода диагностики проектных характеристик обучаемого инженера с помощью диаграмм UML // Вестник Ульяновского государственного технического университета. - 2012. - № 4 (60). - С. 43-46.

4. Афанасьев А.Н., Войт Н.Н. Разработка методов нечеткой параметрической адаптивной диагностики обучаемого инженера // Автоматизация процессов управления. - 2009. - № 3. - С. 51-56.
5. 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 // В сборнике: 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.
6. [http://studopedia.ru/4\\_134710\\_uravneniya-kolmogorova.html](http://studopedia.ru/4_134710_uravneniya-kolmogorova.html)
7. [http://primat.org/news/prjamoe\\_i\\_obratnoe\\_preobrazovanie\\_laplasa/2014-11-30-871](http://primat.org/news/prjamoe_i_obratnoe_preobrazovanie_laplasa/2014-11-30-871)
8. [http://www.wolframalpha.com/input/?i=d%2Fdx+Si\(x\)%5E2&lk=3](http://www.wolframalpha.com/input/?i=d%2Fdx+Si(x)%5E2&lk=3)

## РАЗРАБОТКА И РЕАЛИЗАЦИЯ АРХИТЕКТУРЫ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ДЛЯ ПРОГНОЗИРОВАНИЯ МЕТРИК ИТ ПРОЕКТОВ<sup>37</sup>

Г.Ю. Гуськов<sup>38</sup>, Т.В. Афанасьева<sup>39</sup>, А.А. Романов<sup>40</sup>

В статье рассмотрена архитектура программного обеспечения, позволяющего осуществлять прогнозирование временных рядов, специфичных для ИТ проектов. Результатом исследования является гибкое архитектурное решение, позволяющее использовать большинство моделей, алгоритмов и программ, реализованных для прогнозирования временных рядов и позволяющее получать данные из систем контроля версий и систем отслеживания ошибок.

### Введение

Анализ большого количества уже реализованных проектов позволяет крупной организации увеличить точность оценок прогнозирования метрик проекта влияющих на его успешность и оптимизировать затраты организации. К задаче оценки состояния проекта возможно подойти только оттолкнувшись от данных, полученных из системы контроля версий и системы отслеживания ошибок. Данные из этих систем агрегируют в себе состояние проекта. В случае выявления схожих значений по набору метрик модуль кластеризации отнесёт данные проекты в один кластер и можно будет перенести поведение уже завершённого проекта с проектов в состоянии разработки [1,2,3]. Получив представление о поведении схожего проекта можно попытаться спрогнозировать возможные проблемы и избежать их или же попытаться оптимизировать выделяемые на проект ресурсы. Решение проблемы мониторинга проектов предложено посредством применения

---

<sup>37</sup> Работа выполнена при поддержке РФФИ, проекты №16-47-732070, 16-47-730715.

<sup>38</sup> 432027, Ульяновск, ул. Северный Венец, 32, УлГТУ, e-mail: guskovgleb@gmail.com

<sup>39</sup> 432027, Ульяновск, ул. Северный Венец, 32, УлГТУ, e-mail: tv.afanasjeva@gmail.com

<sup>40</sup> 432027, Ульяновск, ул. Северный Венец, 32, УлГТУ, e-mail: romanov73@gmail.com



средств анализа состояния метрик проектов программного обеспечения, загруженных в систему контроля версий.

Подобное прогнозирование показателей проекта относится преимущественно к крупным организациям начиная от нескольких сотен специалистов. В малых организациях данный подход будет неприменим, так как слишком высока роль каждого специалиста.

## **1. Общая концепция программного обеспечения и его особенности, сформулированные по его целям и задачам**

Программное обеспечение предусматривает несколько вариантов использования. Основными из них являются:

- Проведение прогнозирования;
- Добавления данных из нового источника;
- Нормирование данных выбранным алгоритмом;
- Изменение критерия оценки точности;

Помимо необходимых функций с точки зрения пользователя не меньшую важность имело выделение наиболее часто расширяемых мест в программе. Разработка научно-исследовательского программного обеспечения определяет частые изменения в его структуре и расширении его возможностей. Разработчики, которые модифицируют проект в идеале должны интегрировать свою часть программы с основным решением без изменения основного решения иначе будет порождён круговорот ошибок в разных частях программы. Подобную проблему можно решить разработкой через тестирование и отчасти она решена подобным образом.

Основными местами для расширения функциональности следует считать:

- Увеличение числа типов временных рядов (временные ряды, содержащие дополнительную информацию, или информацию в нетривиальном формате);
- Увеличение числа алгоритмов нормировки данных временных рядов
- Увеличение числа алгоритмов предобработки данных временных рядов
- Добавление новых критериев оценки качества временного ряда (можно использовать расчёт необходимых метрик, не обязательно ошибок);
- Расширение числа алгоритмов прогнозирования временных рядов;
- Агрегирование результатов прогнозирования временных рядов разными методами;

## 2. Базовый алгоритм работы программного обеспечения прогнозирования

Для выявления наиболее эффективного применения гипотез прогноза на основе его корректировки, предложен алгоритм:

- 1) Поиск временных рядов со схожим поведением исходя из данных полученных из системы контроля версий используя один из критериев оценки [4,5,6];
- 2) Выбор проекта среди уже реализованных. наиболее близкого по поведению к разрабатываемому проекту [7];
- 3) Расчёт прогноза по трем гипотезам, учитывая схожесть, корректируем полученный прогноз [8,9];
- 4) Вычисление ошибки прогнозирования одним из внесённых в систему критериев.
- 5) Формирование отчёта для эксперта предметной области с указанием качества точности оценки.

В приведённом алгоритме используются критерии оценки точности нечеткого моделирования числовых уровней ВР в виде мер качества MSE и RMSE:

Квадрат среднеквадратичной ошибки:

$$MSE = \frac{1}{n} \sum_{i=1}^n (e_i^2) = \text{mean}_{i=\overline{1,n}}(e_i^2), \quad (1)$$

Средняя квадратичная ошибка:

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (e_i^2)} = \sqrt{\text{mean}_{i=\overline{1,n}}(e_i^2)}; \quad (2)$$

Данные критерии являются наиболее распространёнными, но далеко не единственными. При решении разных задач прогнозирования могут быть адекватны и применяться разные критерии.

## 3. Диаграмма классов прилагаемого программного обеспечения

Наиболее распространённым стандартом описания архитектурных решений является набор UML-диаграмм. В свою очередь диаграмма классов является наиболее распространённой диаграммой из этого набора. Существует мнение о том, что исходный код не может быть описан лучше, чем с помощью самого исходного кода. Но данное мнение является скорее заблуждением, основанным на незнании широкого спектра UML-диаграмм. Исходный код понятен только специалистам, несёт в себе следы технологии реализации кроме того по нему довольно сложно быстро оценить систему в целом. В целом практика проектирования с

использованием скелета проекта, состоящего из исходного кода не оправдана и содержит очень много спорных решений.

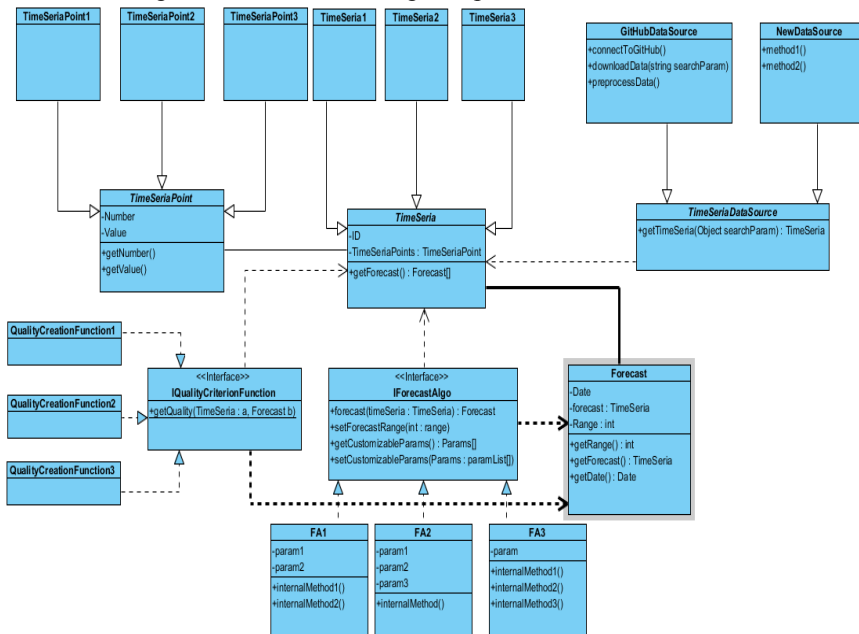


Рис.1 UML-диаграмма разработанной архитектуры программного обеспечения.

Из данной диаграммы видно, что необходимая гибкость достигается использованием абстракций на всех уровнях, где исследователь может предложить своё оригинальное решение. Примером оригинального решения может служить уникальный метод прогнозирования или тип временного ряда.

На схеме показана часть архитектуры. Отсутствуют базовые характерные для большинства проектов возможности по работе с базой данных, сохранением и дальнейшим учётом уже рассчитанных прогнозов и т.д. Кроме базовых возможностей на схеме отсутствуют классы отвечающие за агрегирование оценок, предобработку данных и нормирование данных. Гибкость в алгоритмах неуказанных на схеме обеспечивается тем же способом, что и для приведённых на схеме.

Для получения данных по временным рядам используется специальный класс, который работает с открытым репозиторием исходного кода GitHub.

## Заключение

Предложенная архитектура программного обеспечения позволит решить основные задачи, возникающие как перед разработчиком, так и перед исследователем. Проведенные вычислительные эксперименты на базе данной архитектуры позволят сделать вывод о её пригодности в дальнейшем применении. Предлагаемая архитектура учитывает специфику временных рядов характерных для ИТ проектов, но так же позволяет масштабировать решение на новые области

## Список литературы

1. Pedrycz W., Chen S.M. Time Series Analysis, Modeling and Applications: A Computational Intelligence Perspective (e-book Google) // Intelligent Systems Reference Library. 2013. Vol. 47. P. 404.
2. 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.
3. Мошкин В.С., Пирогов А.Н., Тимина И.А., Шишкин В.В., Ярушкина Н.Г. Интеллектуальный анализ данных и онтологический подход в управлении проектами // Автоматизация процессов управления. 2016. Т. 4(46). С. 84-92.
4. Herbst G., Bocklish S.F. Online Recognition of fuzzy time series patterns // 2009 International Fuzzy Systems Association World Congress and 2009 European Society for Fuzzy.
5. Kacprzyk J., Wilbik A. Using Fuzzy Linguistic summaries for the comparison of time series // 2009 International Fuzzy Systems Association World Congress and 2009 European Society for Fuzzy Logic.
6. Pedrycz W., Chen S.M. Time Series Analysis, Modeling and Applications: A Computational Intelligence Perspective (e-book Google) // Intelligent Systems Reference Library. 2013. Vol. 47. P. 404.
7. Тимина И.А. Корректировка гипотез прогноза для извлечения знаний о временных рядах // Четырнадцатая национальная конференция по искусственному интеллекту с международным участием. КИИ-2014 Труды конференции. 2014. С. 68-76.
8. Ярушкина Н.Г., Тимина И.А. САПР на основе модели прогнозирования нечетких тенденций метрик // Прикладные информационные системы Сборник научных трудов. 2016. С. 191-199.
9. Ярушкина Н.Г., Тимина И.А. Автоматизированное проектирование на основе модели прогнозирования нечетких тенденций метрик // Пятнадцатая национальная конференция по искусственному интеллекту с международным участием Труды конференции: в 3 томах. Российская ассоциация искусственного интеллекта. 2016. С. 156-163.

# ИССЛЕДОВАНИЕ СТАБИЛЬНОСТИ ВИБРАЦИЙ ГИДРОАГРЕГАТА С ПРИМЕНЕНИЕМ МЕТОДОВ СТАТИСТИЧЕСКОГО КОНТРОЛЯ

А.В. Иванова<sup>41</sup>

Рассмотрена методика оценки стабильности вибраций гидроагрегата. При однопараметрическом статистическом контроле применяются стандартные карты Шухарта. Для контроля среднего уровня многопараметрического процесса используется карта Хотеллинга, для контроля многопараметрического рассеяния – карта обобщенной дисперсии.

## Введение

Стабильность работы гидроагрегата во многом зависит от его вибрационного состояния. Повышенная вибрация является причиной быстрого износа и отказов ответственных элементов и узлов. Поэтому, согласно разработанному в России стандарту [1] использование гидроагрегатов, не оснащенных системами виброконтроля, запрещено.

Система вибромониторинга гидроагрегата включает показания распределенной сети датчиков относительной и абсолютной вибрации различных элементов агрегата. Замер вибраций проводился в 10 точках по два измерения в секунду. Для повышения эффективности управления гидроагрегатом необходимо иметь возможность предупреждения об аварийной ситуации по множеству его характеристик, в частности по критерию нарушения стабильности вибраций [1-4].

Для оценки стабильности вибрационного состояния агрегата возможно применение контрольных карт используемых при статистическом управлении технологическими процессами [5-7].

## 1. Контроль независимых показателей процесса

В рассматриваемом гидроагрегате контролируется биение вала и вибрация. Состояние характеризуется десятью показателями. Для определения наличия корреляций между анализируемыми параметрами выбирается этап стабильной работы гидроагрегата и с помощью специальной программы строятся матрицы корреляций.

---

<sup>41</sup> 432027, Ульяновск, ул. Северный Венец, 32, УлГТУ, e-mail: age-89@mail.ru

На рис. 1 представлена корреляционная матрица, значимые корреляции выделены красным цветом.

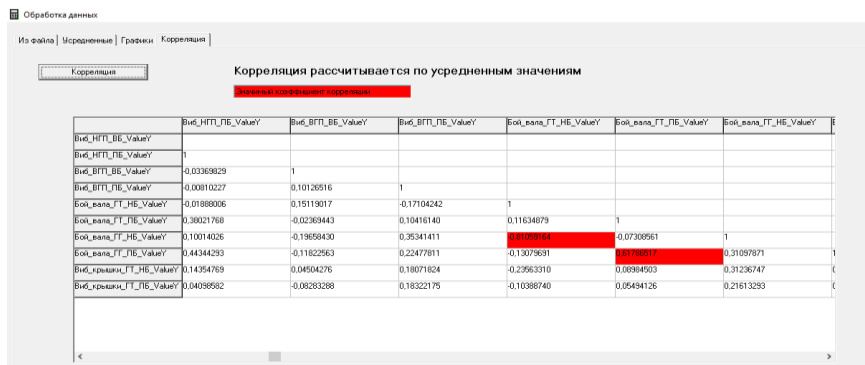


Рис.1. Корреляционная матрица

Анализ корреляций свидетельствует о том, что стабильность некоррелированных параметров могут быть оценены с помощью стандартных карт Шухарта [8-9], например, для среднего уровня и стандартного отклонения. На рис. 2 эти карты показаны для показателя  $X_1$ , построенные в системе Statistica.

Процесс считается стабильным, так как об этом свидетельствуют обе карты и для среднего уровня, и для рассеяния (отсутствуют точки, выходящие за контрольные границы).

Таким образом анализируется каждый некоррелированный параметр.

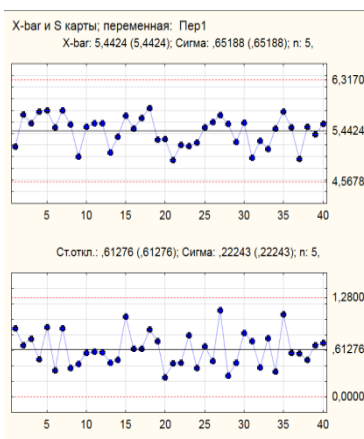


Рис.2. Карта Шухарта для показателя  $X_1$

## 2. Многомерный статистический контроль

Для контроля среднего уровня коррелированных параметров может применяться многомерная карта Хотеллинга [5-6]. На рис. 3 представлена карта Хотеллинга для коррелированной пары параметров X5-X7.

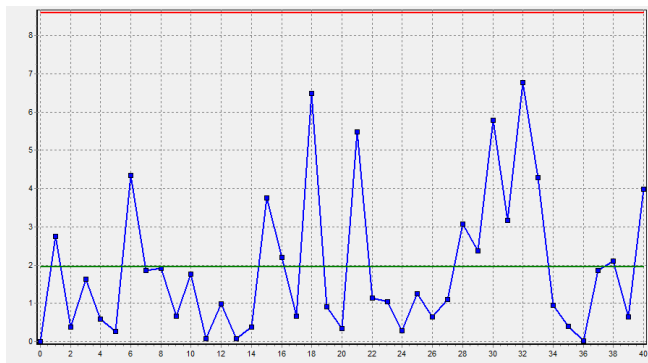


Рис.3. Карта Хотеллинга для группы коррелированных показателей X5-X7

Контролировать стабильность многомерного рассеяния процесса можно с помощью карты обобщенной дисперсии. Карта обобщенной дисперсии для той же пары коррелированных параметров X5-X7 представлена на рис. 4.

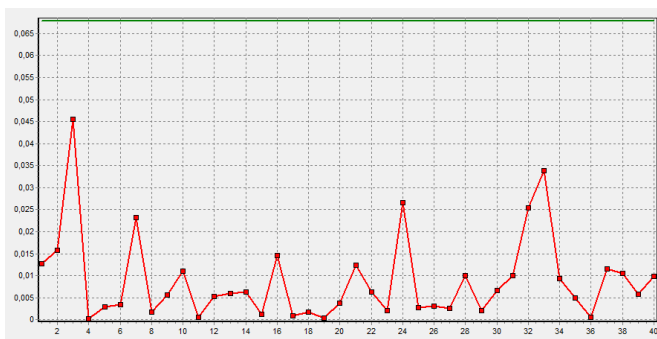


Рис.4. Карта обобщенной дисперсии для группы коррелированных показателей X5-X7

Как и в случае однопараметрического контроля процесса, в многопараметрическом статистическом контроле о нарушении стабильности процесса свидетельствует выход хотя бы одной из точек на карте за контрольные границы. Таким образом, как карта Хотеллинга, так

и карта обобщенной дисперсии свидетельствуют о стабильности процесса по группе коррелированных показателей X5-X7.

Проведенное исследование позволяет сформулировать алгоритм проведения анализа стабильности вибраций гидроагрегата.

### Список литературы

1. Стандарт организации ОАО РАО «ЕЭС России». Методика оценки технического состояния основного оборудования гидроэлектростанций. СТО 17330282.27.140.001-2006.
2. Иванова А.В., Клячкин В.Н. Статистический анализ данных о вибрациях гидроагрегата // Материалы научно-практической всероссийской конференции молодых ученых «Прикладная математика и информатика: современные исследования в области естественных и технических наук». – Тольятти, 2017.
3. Клячкин В.Н., Кувайскова Ю.Е., Алёшина А.А. Моделирование вибраций гидроагрегата на основе адаптивных динамических регрессий // Автоматизация. Современные технологии. – 2014. – № 1. – С.30-34.
4. Клячкин В.Н., Кувайскова Ю.Е., Алёшина А.А. О возможности совместного описания характеристик гидроагрегата адаптивными регрессиями // Вектор науки Тольяттинского государственного университета. – 2013. – № 1 (23). – С.40-41.
5. Клячкин В.Н. Статистические методы в управлении качеством: компьютерные технологии. – М.: Финансы и статистика, ИНФРА-М, 2009. – 304 с.
6. Клячкин В.Н. Модели и методы статистического контроля многопараметрического технологического процесса. – М. : ФИЗМАТЛИТ, 2011. – 196 с.
7. Клячкин В.Н., Кувайскова Ю.Е., Алексеева В.А. Статистические методы анализа данных. – М. : Финансы и статистика, 2016. – 240 с.
8. Уилер, Д. Статистическое управление процессами/ Оптимизация бизнеса с использованием контрольных карт Шухарта; пер. с англ. / Д. Уилер, Д. Чамберс. – М. : Альпина Бизнес Букс, 2009. – 409 с.
9. Клячкин В.Н. Многомерные статистические методы в прикладных задачах // В сборнике: [IN MEMORIAM: Султан Галимзянович Валеев](#), сборник памяти С. Г. Валеева. Ульяновск, 2016. С. 88-104.
10. Кравцов Ю.А., Клячкин В.Н. [Обнаружение нарушения процесса по наличию скачков на многомерной карте](#) // В сборнике: [Информатика, моделирование, автоматизация проектирования](#) Сборник научных трудов VII Всероссийской школы-семинара аспирантов, студентов и молодых ученых (ИМАП-2015). 2015. С. 149-152.



## ИССЛЕДОВАНИЕ ПОТОКОВ РАБОТ ПО СОПРОВОЖДЕНИЮ ПРОИЗВОДСТВА СЛОЖНЫХ ТЕХНИЧЕСКИХ ИЗДЕЛИЙ<sup>42</sup>

И.С. Ионова<sup>43</sup>

Для формального анализа проектных потоков работ предлагается моделирование логистических процессов, на примере частных задач расчета себестоимости и нормирования деталей и сборочных единиц, используя представление потоков работ в виде графов. В работе были разработаны и проанализированы модели двух уровней: структурно-функциональная и потоковая модели.

### Введение

Парадигмой анализа деятельности предприятия является исследование потоков работ, протекающих на этом предприятии. Данные потоки можно определить, как совокупность задач, решение которых направлено на достижение конкретных целей предприятия. Вот почему внедрение их формального описания позволяет построить системы управления качеством предприятия, решить проблему построения эффективной структуры управления, оптимизировать деятельность на основе ключевых показателей.

С потоками работ связана производственная деятельность сотрудников, они могут реализовываться несколькими работниками в пределах одного подразделения предприятия, охватывать несколько его подразделений или даже несколько подразделений различных предприятий. Поэтому они представляются как параллельно протекающие активности многих исполнителей, последовательно выполняющих свои функции. Здесь возникают известные проблемы доступа к ресурсам, блокировки, ограничения живости, и поэтому результаты и решения в

---

<sup>42</sup> Исследования поддержаны грантом Министерства образования и науки Российской Федерации, проект № 2.1615.2017/4.6.

Исследование выполнено при финансовой поддержке РФФИ и Правительства Ульяновской области в рамках научного проекта № 16-47-732152

Исследование выполнено при финансовой поддержке РФФИ в рамках научного проекта № 17-07-01417

<sup>43</sup> 432027, Ульяновск, ул. Северный Венец, 32, УлГТУ, e-mail: epira@mail.ru

области верификации параллельных процессов являются актуальными с практической точки зрения.

Наличие большого числа взаимодействующих потоков ставит задачу формальной проверки их правильности. Однако потоки работ специфицируются работниками, которые не имеют подготовки в области формальных моделей и информатики. Для их формального анализа необходимо детальное представление модели процесса на формальном языке, которое трудно построить и которое нелегко понять обычным работникам [1-8].

## **1. Проблема**

Обработка проектно-конструкторских потоков работ крупного промышленного предприятия требует решения следующих задач:

- 1) анализа, систематизации и разработки нормативных диаграммных моделей потоков работ;
- 2) анализа и контроля свойств диаграммных моделей, в первую очередь топологических;
- 3) анализа целостности бизнес-процессов;
- 4) интерпретация потоков работ, включая внедрение систем управления потоками работ на основе разработанного бизнес-процесса.

Отсутствие современных инструментов, методов и средств для обработки потоков работ определяет предмет исследования и его эффективность.

## **2. Связанные работы**

Анализом и контролем проектных потоков работ занимаются научные школы ГУВШЭ, МГТУ СТАНКИН, МВТУ им. Н.Э. Баумана, УлГТУ, ПОМИ АН РФ, ВМК МГУ имени М.В. Ломоносова, Институт системного программирования РАН (Россия), университет Карнеги-Меллон (США), лаборатории VERIMAG (Франция), такие ученые как Афанасьев А.Н., Карпов Ю.Г., Соснин П.И., Лифшиц Ю., Ярушкина Н.Г., Калянов Г.Н., Конев Б. Ю., Шальто А. А., Савенков К., Кулямин В. В. (Россия), а также Teda Saeedloei, Gopal Gupta, Э.М. Кларк, Г. Буч (США), Yuan Wang, Yushun Fan (Китай), а также технологии Rational Unified Process (RUP), методологии PBWD, языков моделирования Unified Model Language (UML), extended Event Driven Process Chain (eEPC), BPMN, IDEF0, IDEF3, Amber, Promela, YAWL.

## **3. Потоки работ в условиях крупного предприятия**

Потенциал развития отрасли информационных технологий в России очень высок. Важным этапом развития в этой области явилось формирование в стране цифровой экономики, ориентированной на

повышении эффективности всех отраслей за счет использования информационных технологий. В связи с этим, Председатель правительства РФ Дмитрий Медведев подписал распоряжение от 28 июля 2017 года №1632-р, утверждающее программу «Цифровая экономика Российской Федерации». Данная программа предлагает адаптацию нормативной правовой базы к новым видам отношений, новым объектам и субъектам цифровой экономики, создание инфраструктуры, технологий и платформ, обеспечивающих получение, хранение и обработку того объема данных, которые создаются в условиях цифровой экономики, обеспечение недискриминационного доступа к данным при условии обеспечения законных прав и интересов субъектов и владельцев данных, подготовку квалифицированных кадров, обеспечение информационной безопасности, развитие прикладных решений для нового качества использования информационно-телекоммуникационных технологий в сфере государственного и муниципального управления, здравоохранения, образования, управления городским хозяйством и других отраслях экономики. Это тот стандарт, к которому должны стремиться и крупные производственные предприятия.

Предприятие «Ульяновский механический завод» занимается проектированием и производством радиоэлектронной аппаратуры и приборов, и других технических изделий. Сложившиеся структуры предприятия имеют сложную логистическую систему, включающую операции с материальными, финансовыми и информационными потоками. Данные потоки работ протекают в условиях, сформированных спецификой предприятия и особенностями производства сложных технических изделий:

- 1) многономенклатурность выпускаемой продукции;
- 2) преобладание технологической специализации рабочих мест, участков, цехов;
- 3) использование универсального оборудования и технологической оснастки;
- 4) наличие большого объема ручных сборочных и доводочных операций;
- 5) существенная длительность производственного цикла;
- 6) возможное изменение маршрутов изготовления определенных изделий;
- 7) неавтоматизированный складской учет;
- 8) применение смешанных технологий и различных систем и пр.

В этих условиях решаются задачи анализа, выявления узких мест и оптимизации логистических процессов. Одним из таких процессов является ценообразование. Без данного процесса невозможна финансовая деятельность предприятия, заключение контрактов и получение прибыли.

В состав изделий входят как готовые покупные материалы и комплектующие, так и детали, и сборочные единицы собственного производства. В связи с этим корректный расчет себестоимости изделия является важным финансовым потоком работ в рамках всей логистической системы.

При расчете себестоимости применяются смешанные технологии (бухгалтерский учет ведется в 1С: Предприятие, а складской учет ведется вручную). Это может приводить к возникновению ошибок, нестыковке данных, сложностям доступа к данным, более длительным расчетам и пр. Вот почему необходим переход на качественно новый уровень использования информационно-телекоммуникационных технологий в сфере социально-экономической деятельности предприятия.

В связи с этим возникает проблема реинжиниринга на предприятии. Должно быть, осуществлено максимальное внедрение средств автоматизации на всех этапах жизненного цикла сложных мобильных изделий. Необходимо провести фундаментальное переосмысление и радикальное перепроектирование логистических процессов, сопровождающих производство, для достижения максимального эффекта производственно-хозяйственной и финансово-экономической деятельности. Вот почему в настоящее время на предприятии АО «Ульяновский механический завод» происходит активное внедрение системы «Alfa», которая призвана решить выше поставленные задачи.

#### **4. Исследование потоков работ по сопровождению производства**

Для исследования потоков работ представим их в виде графов. В качестве частных задач для анализа были выбраны потоки работ по сопровождению производства, связанные с формированием себестоимости продукции при изготовлении технических изделий и нормированием деталей и сборочных единиц, входящих в состав изделий. В работе разработаны модели двух уровней: структурно-функциональная и потоковая модели. Рассмотрим особенности каждой из них.

Структурно-функциональная модель «Расчет себестоимости» представлена на рис. 1.

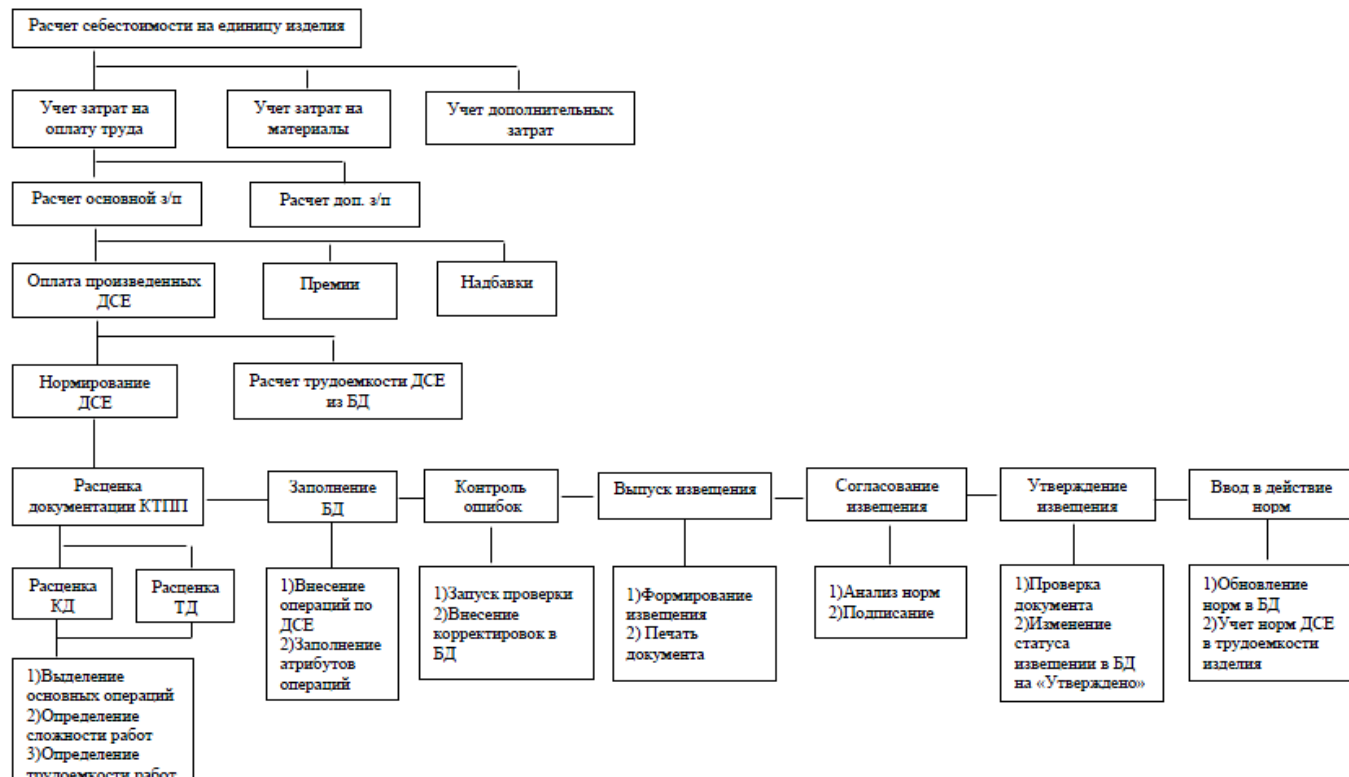


Рис. 1. Структурно-функциональная модель «Расчет себестоимости»

При расчете себестоимости на единицу изделия (единицей изделия является конкретный шифр затрат по изделию) учитываются затраты на оплату труда, которые рассчитывает Отдел труда и заработной платы, затраты на материалы, рассчитываемые Планово-экономический отдел, и дополнительные затраты, которые считает Отдел бухгалтерского учета. Все затраты прописываются в статьях калькуляции по изделию. К затратам на материалы относятся затраты на сырье и основные материалы, вспомогательные материалы, покупные полуфабрикаты, возвратные отходы, комплектующие изделия, работы и услуги сторонних организаций производственного характера, транспортно-заготовительные расходы, топливо и энергия на технологические цели, тара и упаковка. К дополнительным затратам можно отнести страховые взносы на обязательное социальное страхование, затраты на подготовку и освоение производства, затраты на специальную технологическую оснастку, специальные затраты, общепроизводственные затраты, общехозяйственные, прочие производственные затраты, внепроизводственные затраты. Все вышеперечисленные виды дополнительных затрат считаются процентом от затрат на оплату труда. Затраты на оплату труда основных производственных рабочих делятся на основную заработную плату и дополнительную. К дополнительной заработной плате относятся некоторые виды надбавок, несвязанные с произведенными деталями и сборочными единицами (ДСЕ), к основной – оплата произведенных деталей и сборочных единиц, премии, и некоторые надбавки, рассчитываемые от оплаты за произведенные ДСЕ. Оплата произведенных ДСЕ происходит по установленным трудозатратам (трудоемкости ДСЕ), рассчитываемых из базы данных по ДСЕ. Данную трудоемкость определяют и вносят в базу данных инженеры по нормированию и организации труда. При производстве новых деталей и сборочных единиц, на которые еще не установлена трудоемкость, необходимо их отнормировать и ввести в действие эти нормы.

Нормирование ДСЕ начинается с получения новой документации – это либо пакет конструкторской документации (сборочный чертеж, схема электрическая принципиальная и т.д.), либо пакет технологической документации (технологический процесс). Уже на этом этапе просматриваются проблемы организации рабочего процесса – для точного определения трудоемкости необходим полный пакет документов (то есть вся конструкторская и технологическая документация), на деле же, это в лучшем случае только часть конструкторской документации. Технологический процесс составляется уже намного позже на основе операций, определенных нормировщиком. После получения документации, нормировщик выделяет основные операции при производстве ДСЕ, определяет профессии производственных рабочих,

которые будут изготавливать ДСЕ, определяет сложность работ (разряд работ), вредность работ, и трудоемкость по каждой операции. После этого в базе данных находят ДСЕ, создают ее новую редакцию, в которой заполняют все операции и атрибуты по ним. На данном этапе возможны ошибки из-за «человеческого фактора» – можно неправильно заполнить базу данных, пропустить какие-то атрибуты, присвоить не тот тип атрибутам и т.п. Для контроля данных ошибок в базе данных существует возможность формирования отчета «Контроль ошибок». При запуске отчета выводится список ДСЕ, в которых были сделаны ошибки, а также номер строки и тип атрибута, которые уточняют, где именно эти ошибки были сделаны. После того, как все ошибки будут найдены и исправлены, в базе данных создается извещение и ему присваивается статус «Сформировано». Затем «Извещение» печатается для согласования норм с другими службами. Извещение должно быть согласовано с начальником отдела технологов, начальником цеха, начальником отделом труда и заработной платы. При согласовании с каждой службой осуществляется анализ норм. Если в ходе анализа будут найдены какие-либо ошибки, или несоответствие трудоемкости ДСЕ, то вносятся изменения в базу данных, и выпускается повторное извещение. После согласования со всеми службами, извещение передается в отдел труда и заработной платы. После проверки документа в базе данных меняется статус извещения на «Утверждено», после чего происходит обновление норм ДСЕ.

На рис. 2 представлена потоковая модель «Нормирование ДСЕ». Разработанная модель нормирования ДСЕ была проверена в специализированном программном обеспечении Visual ObjectNet++.

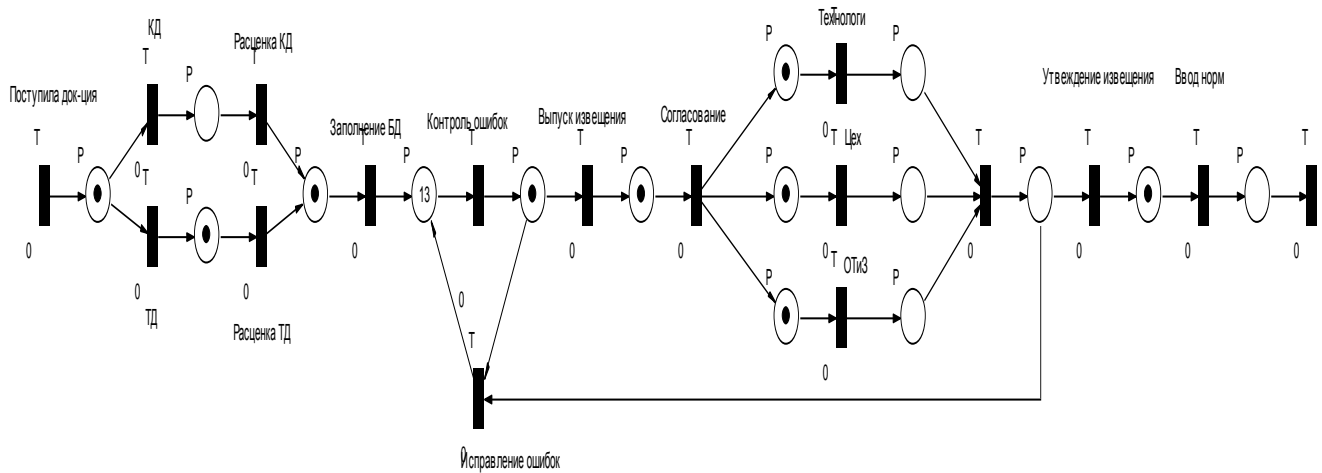


Рис. 2. Поточковая модель «Нормирование ДСЕ»



## Заключение

В результате моделирования было большое скопление фишек в позиции «Контроль ошибок» (рис. 2), что говорит о том, что модель не является безопасной, и поэтому потоки работ при нормировании ДСЕ также значительно скапливаются при контроле. Необходимо «разгрузить» этап контроля ошибок с помощью структурного изменения графа потоков работ. Также модель обладает свойствами достижимости, живости и справедливости.

## Список литературы

1. Афанасьев А.Н., Войт Н.Н., Бочков С.И., Уханова М.Е., Ионова И.С. Назработка и исследование виртуальных рабочих мест в среде OPENSIM // Вестник Ульяновского государственного технического университета. 2016. № 4 (76). С. 43-47.
2. Афанасьев А.Н., Войт Н.Н., Уханова М.Е., Ионова И.С., Елифанов В.В. Анализ конструкторско-технологических потоков работ в условиях крупного радиотехнического предприятия // Радиотехника. 2017. № 6. С. 49-58.
3. 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 // В сборнике: 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.
4. Афанасьев А.Н., Войт Н.Н. Компонентная автоматизированная обучающая система САПР на основе гибридной нейронной сети // Автоматизация. Современные технологии. 2009. № 3. С. 14-18.
5. 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.
6. Афанасьев А.Н., Войт Н.Н. Реализация конструктора сценария обучающих курсов // Вестник Ульяновского государственного технического университета. 2011. № 1 (53). С. 54-59.
7. Афанасьев А.Н., Войт Н.Н. Разработка методов нечеткой параметрической адаптивной диагностики обучаемого инженера // Автоматизация процессов управления. 2009. № 3. С. 51-56.
8. 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.

**РАЗРАБОТКА И ИССЛЕДОВАНИЕ ВРЕМЕННОЙ RV-  
ГРАММАТИКИ ДЛЯ АНАЛИЗА И КОНТРОЛЯ  
СТРУКТУРНЫХ ОСОБЕННОСТЕЙ  
ДИАГРАММАТИЧЕСКИХ МОДЕЛЕЙ  
ДИНАМИЧЕСКИХ РАСПРЕДЕЛЕННЫХ ПОТОКОВ  
РАБОТ<sup>44</sup>**

С.Ю.Кириллов<sup>45</sup>, Н.Н.Войт<sup>46</sup>, В.А.Гордеев<sup>47</sup>

Предложено определение временной RV-грамматики и её возможная реализация с примером.

**Введение**

Применение временного автомата в проектировании, спецификации, контроле и анализе систем реального времени является известной практикой [1]. Временные и гибридные автоматы используются для анализа и управления в киберфизических системах (КФС) [2]. Время является неотъемлемой характеристикой большинства процессов. Очень важно при описании потоков работ учитывать временной фактор. Это может быть заранее определенное значение выполнения операции или же таймаут, после которого ход потока работ продолжался бы принудительно. Также бывают ситуации, когда необходимо уложиться с выполнением всех работ в определенный срок, либо распределить по времени определенный общий ресурс. Невыполнение может привести к срыву сроков или простоя, что критически скажется на ожидаемом результате. За основу будем использовать RV-грамматику [3].

---

<sup>44</sup> Исследование выполнено при финансовой поддержке РФФИ в рамках научного проекта № 17-07-01417 а

<sup>45</sup> 432027, Ульяновск, ул. Северный Венец, 32, УлГТУ, e-mail: xayc73@gmail.com

<sup>46</sup> 432027, Ульяновск, ул. Северный Венец, 32, УлГТУ, e-mail: n.voit@ulstu.ru

<sup>47</sup> 432027, Ульяновск, ул. Северный Венец, 32, УлГТУ, e-mail: gordevlad@yandex.ru

## Определение временной RV-грамматики

Автоматная временная RV-грамматика является расширением RV-грамматики [4] и представлена шестеркой следующих компонентов:

$$G = (V, \Sigma, C, E, R, r_0), \quad (1)$$

где  $V = \{v_e, e = \overline{1, H}\}$  – вспомогательный алфавит (алфавит операций над внутренней памятью);  $\Sigma = \{a_l, l = \overline{1, L}\}$  – терминальный алфавит языка;  $C$  – конечное множество идентификаторов часов;  $E$  – множество временных выражений, определенных на  $C$  (ограничение часов и сброс часов), ограничено следующими выражениями: с начала  $\{c := 0\}$  и далее  $\{c \sim x\}$ , причем  $c$  является переменной, а  $x$  является константой,  $\sim \in \{=, <, \leq, >, \geq\}$ ;  $R = \{r_i, i = \overline{0, I}\}$  – схема грамматики  $G$  (множество имен комплексов продукций, причем каждый комплекс  $r_i$  состоит из подмножества  $P_{ij}$  продукций  $r_i = \{P_{ij}, j = \overline{1, J}\}$ );  $r_0 \in R$  – аксиома RT-грамматики (имя начального комплекса продукций),  $r_k \in R$  – заключительный комплекс продукций. Продукция  $P_{ij} \in r_i$  имеет вид  $P_{ij}: a_l \frac{W_v(\gamma_1, \dots, \gamma_n)}{E} r_m$ , где  $W_v(\gamma_1, \dots, \gamma_n)$  –  $n$ -арное отношение, определяющее вид операции над внутренней памятью в зависимости от  $v \in \{0, 1, 2\}$  (соответственно 0 – операции не производится, 1 – запись, 2 – чтение);  $\gamma_1, \dots, \gamma_n \in V$ ;  $r_i \in R$  – имя комплекса продукции-источника;  $r_m \in R$  – имя комплекса продукции-приемника.

### Пример

Пример диаграммы, использующей временную характеристику, изображен на рис. 1.

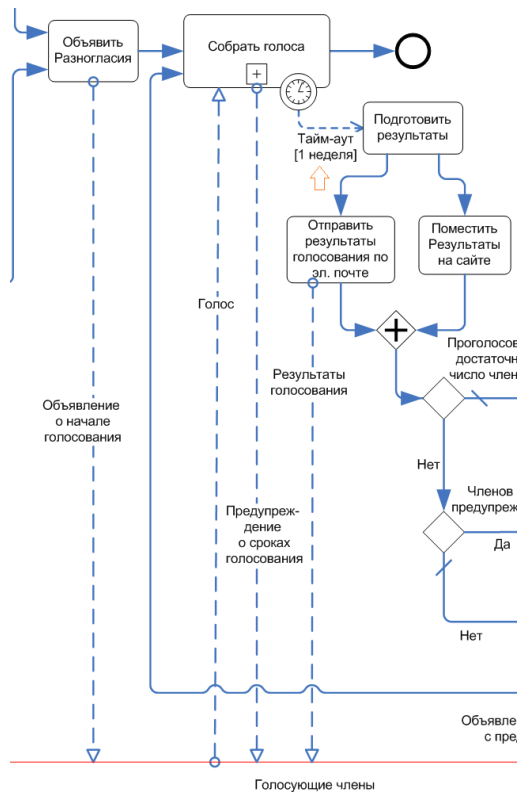


Рис. 1. Пример использования таймаута в нотации BPMN

Здесь описан процесс проведения выборов в совет директоров. Для них определена дата проведения, или же конечная временная точка, когда выбранный человек должен приступить к выполнению обязанностей. Предположим, что начальная дата тоже определена и находится в диапазоне текущего времени и даты завершения. Поскольку факторов, влияющих на время прохождения процесса много, а в данном примере присутствует и самый значимый - человеческий фактор, то необходимо определить таймауты, в результате которых будет изменяться поведение потока. Самым плохим исходом событий, при котором можно не уложиться в срок, является ситуация срабатывания всех возможных таймаутов. Она и будет рассмотрена. Для операций, у которых не указано временное значение или таймаут, будет присвоено значение по умолчанию  $t_d$ .

RV-грамматика представляет собой автомат с внутренней памятью, при этом анализ диаграммы выполняется за линейное время в зависимости от длины графической цепочки. При наличии условий или параллелизма возникают сложности с подсчетом временных промежутков. Для решения этой задачи вводится дополнительная лента, которая будет содержать информацию о суммарном времени для текущего элемента. При возврате по связь-метке label кроме операции чтения из магазина элемента необходимо будет также получать значение из ленты, относящейся к нему. Предположим, что затраты времени происходят только при выполнении операции. Добавим операцию  $W_1(t_s^{t(5)})$ , в которой  $t_s$  - предварительно высчитанная сумма двух чисел: текущей суммы времени для элемента и его временной характеристики. При возврате анализа также выполняется операция  $W_2(b^{1m}, b^{t(6)})$ , то есть кроме чтения из магазина самого элемента, происходит чтение информации из соответствующей ленты его временной суммы, которая определяет  $t_s$ .

Таблица 1. Пример временной RV-грамматики

№ п/п	Комплекс	Квазитерм	Приемник	Операция с памятью
1	r0	A0	r1	o
2	r1	rel	r3	o
3	r2	labelEG	r3	$W_2(b^{1m}, b^{t(6)})$
4		labelPG	r3	$W_2(b^{2m}, b^{t(6)})$
5	r3	Ai	r1	o
6		Aim	r1	o
7		Ait	r1	$W_1(t_s^{t(6)})$
8		Akl	r2	$W_3(e^{1m}, e^{2m})$
9		Ak	r4	o
10		A	r1	$W_1(t_s^{t(6)})$
11		EGc	r1	$W_1(t^{1m^{(n-1)}})/W_3(k=1)$
12		EG	r2	$W_1(1^{t(1)}, k^{t(2)})/W_3(e^{t(2)}, k \neq 1)$
13		_EG	r2	$W_1(\text{inc}(m^{t(1)}))/W_3(m^{t(1)} < k^{t(2)})$
14		_EGe	r1	$W_1(t^{1m^{(n-1)}})/W_3(m^{t(1)}=k^{t(2)}, p \neq 1)$
15		_EGme	r1	$o/W_3(m^{t(1)}=k^{t(2)}, p=1)$
16		PGf	r1	$W_1(t^{2m^{(n-1)}})/W_3(k=1)$
17		PG	r2	$W_1(1^{t(3)}, k^{t(4)})/W_3(e^{t(3)}, k \neq 1)$
18		_PG	r2	$W_1(\text{inc}(m^{t(3)}))/W_3(m^{t(3)} < k^{t(4)})$

19		_PGe	r1	$W_1(t^{2m^{(n-1)}})/W_3(m^{t(3)}=k^{t(4)}, p \neq 1)$
20		_PGje	r1	$W_1(t^{2m^{(n-1)}})/W_3(m^{t(3)}=k^{t(4)}, p = 1)$
21	r4	no_label	r5	*
22	r5			

Символ "\*" обозначает операции вида:

$$* = W_2(e^{1m}) \&\& W_2(e^{2m}) \&\& W_3(m^{ai(1)} = k^{ai(2)}) \&\& W_3(m^{ai(3)} = k^{ai(4)}) \&\& W_3(m^{ai(6)} < t_{max})$$

То есть проверяем магазины на пустоту, сравниваем количество проанализированных ветвей с их общим количеством и временную сумму на каждом элементе с её максимальным заданным значением.

### Заключение

Разработан метод анализа диаграммы на основе RV-грамматики, который позволяет учитывать временные ограничения. Это может предотвращать логические ошибки планирования при разработке и проектировании автоматизированных систем. В дальнейшем возможна доработка метода на проверку временных пересечений для различных ресурсов и исполнителей.

### Список литературы

1. Karpov Yu. G. MODEL CHECKING. Verifikaciya parallel'nyh i raspredeennyh programmnyh system. - SPb.: BHV-Peterburg, 560 s, 2010 (in Russian).
2. Edward A. Lee. Cyber-physical systems: Design challenges // ISORC, 2008.
3. Шаров О.Г., Афанасьев А.Н. Автоматная графическая грамматика // Вестник Ульяновского государственного технического университета. 2005. № 1 (29). С. 54-56.
4. Афанасьев А.Н. Методология графо-аналитического подхода к анализу и контролю потоков работ в автоматизированном проектировании сложных компьютерных систем // Вестник Ульяновского государственного технического университета. 2011. № 3 (55). С. 48-52

**РАЗРАБОТКА И ИССЛЕДОВАНИЕ МЕТОДОВ  
АНАЛИЗА И КОНТРОЛЯ СЕМАНТИЧЕСКОЙ  
ЦЕЛОСТНОСТИ И СОГЛАСОВАННОСТИ  
ДИАГРАММАТИЧЕСКИХ МОДЕЛЕЙ  
ДИНАМИЧЕСКИХ РАСПРЕДЕЛЕННЫХ ПОТОКОВ  
РАБОТ НА ОСНОВЕ ВРЕМЕННОЙ RV-  
ГРАММАТИКИ<sup>48</sup>**

С.Ю. Кириллов<sup>49</sup>, Н.Н. Войт<sup>50</sup>, Р.С. Молотов<sup>51</sup>,  
А.С. Степанов<sup>52</sup>, Е.Ю. Воеводин<sup>53</sup>, С.И. Бригаднов<sup>54</sup>

Предлагается временная RV-грамматика, правило которой дополнено процедурой извлечения и анализа семантической (текстовой) информации. В основу метода анализа семантической целостности и согласованности текстовой составляющей диаграмм при коллективном проектировании положен подход объединения онтологий, сформированных на основе модифицированного способа лексико-синтаксических шаблонов

### **Введение**

При коллективном проектировании сложных автоматизированных систем (САС) на концептуальном этапе разработки возникает проблема семантической целостности и согласованности диаграмматических моделей (одинаковое понимание и использование понятий и их обозначений, иерархическая и параметрическая согласованность и др.).

---

<sup>48</sup> Исследование выполнено при финансовой поддержке РФФИ в рамках научного проекта № 17-07-01417 а

<sup>49</sup> 432027, Ульяновск, ул. Северный Венец, 32, УлГТУ, e-mail: хаус73@gmail.com

<sup>50</sup> 432027, Ульяновск, ул. Северный Венец, 32, УлГТУ, e-mail: n.voit@ulstu.ru

<sup>51</sup> 432027, Ульяновск, ул. Северный Венец, 32, УлГТУ, e-mail: step\_al\_ul@mail.ru

<sup>52</sup> 432027, Ульяновск, ул. Северный Венец, 32, УлГТУ, e-mail: r.molotov@yahoo.com

<sup>53</sup> 432027, Ульяновск, ул. Северный Венец, 32, УлГТУ, e-mail: voevodineu@gmail.com

<sup>54</sup> 432027, Ульяновск, ул. Северный Венец, 32, УлГТУ, e-mail: sergbrig@yandex.ru

Наличие таких ошибок, которые являются самыми «дорогими», резко снижает показатель успешности проектирования САС. Проведение семантического анализа всех разработанных диаграммных моделей потоков работ позволяет выявлять логические ошибки. Для решения задачи предлагается доработанная временная RV-грамматика, использующая модифицированный метод лексико-синтаксических шаблонов.

## 1. Онтологический анализ диаграмм на основе временной RV-грамматики

Для проведения онтологического анализа можно выделить два подхода:

1. В течении процесса анализа диаграммы.
2. При полном построении онтологии по завершению анализа диаграммы.

При первом подходе появляется возможность сразу указать на конкретный элемент, который привел к ошибке. Однако придется затрачивать большее количество времени, так как анализ онтологий должен проводиться на каждом элементе.

При втором подходе процесс объединения онтологий необходимо провести лишь один раз.

Далее будет рассмотрен второй вариант. Общая схема онтологического анализа диаграмм представлена на рис.1 [1].

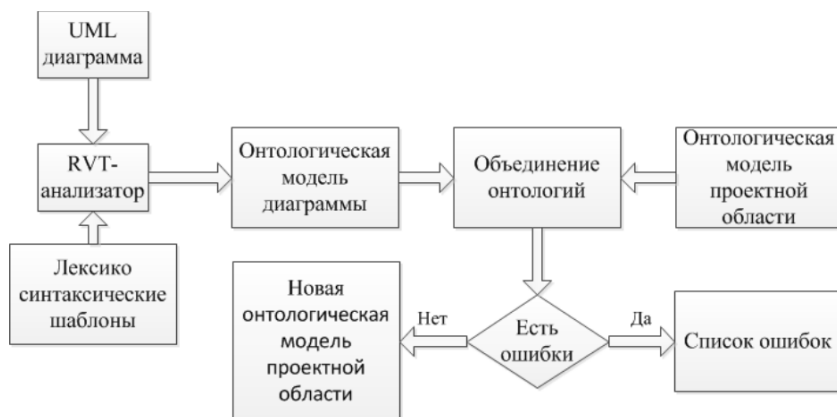


Рис. 1. Общая схема онтологического анализа UML диаграмм



Временная RV-грамматика позволяет контролировать временные факторы, влияющие на выполнение потоков работ. Исходя из этого, применение данной грамматики возможно лишь для диаграмм, предполагающих описание процессов.

Для того, чтобы дополнительно контролировать согласованность различных типов диаграмм, внесем изменения в правило RV-грамматики (определение автоматной RV-грамматики и ее пример для анализа языка BPMN приведены в [2]):

$$a(\chi, \zeta) \xrightarrow{W_V(\chi_1, \dots, \chi_n)} r_m$$

где  $\chi$  - процедура извлечения семантической информации, а  $\zeta$  - процедура извлечения временной информации.

Процедура  $\chi$  заключается в поиске соответствующего правила в списке. Правило состоит из двух частей – replacement и pattern. Часть pattern описывает лексико-синтаксический шаблон. Часть replacement определяет местоположение данной текстовой единицы в частичном семантическом дереве диаграммы. Для языка UML в работе [ ] предложена многоуровневая грамматика.

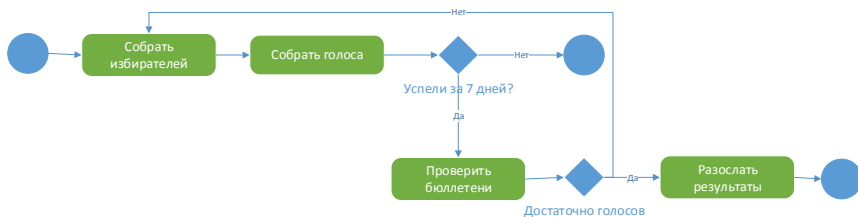


Рис. 2. Пример UML диаграммы активности

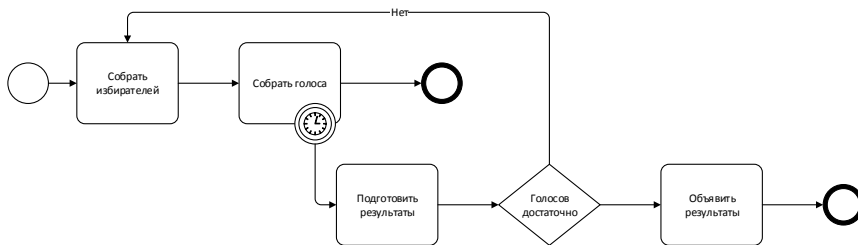


Рис.3. Пример BPMN диаграммы

На рис. 2 и рис. 3 отображены упрощенная модель проведения выборов с помощью диаграммы активности UML и BPMN. Предполагается, что разработкой моделей занимались разные проектировщики, специализирующиеся на конкретном языке.

К указанным диаграммам применяются следующие шаблоны:

$\text{block}\{\text{type: action}\} \ \& \ \text{block}\{\text{value: name}\} \ == \ \text{verb} \ \rightarrow \ \text{Action}[\text{value: name}]$

$\text{block}\{\text{type: action}\} \ \& \ \text{block}\{\text{value: name}\} \ == \ \text{noun} \ \rightarrow \ \text{Resource}\langle\text{Action}[\text{value: name}]\rangle$

В результате получаем схожие, но тем не менее различающиеся деревья онтологии (рис. 4 и рис. 5).

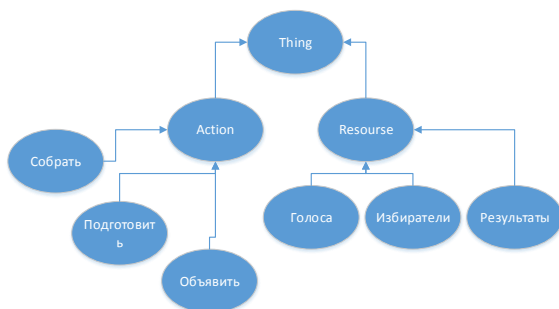


Рис.4. Онтологическая модель, построенная по диаграмме активности UML

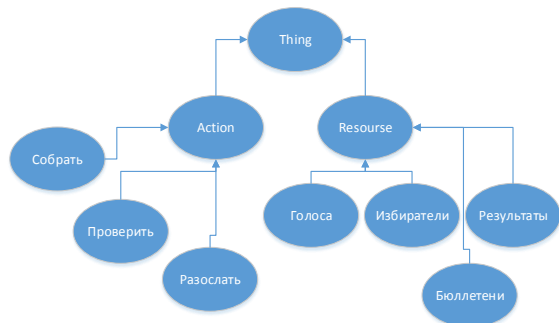


Рис.5. Онтологическая модель, построенная по BPMN диаграмме

В данном случае, объединение произойдет успешно, так как явно выделяются опорные элементы. Но сразу заметно, что каждая модель содержит уникальные понятия, и при дальнейшем анализе будет выявлена ошибка.

Диаграммы не согласованы между собой. Такой результат может быть получен вследствие использования диаграммы из другого проекта, из-за неправильно составленной диаграммы или из-за отсутствия связующей диаграммы. В каждом из этих случаев разработчику выдается сообщение

об ошибке, и вторая диаграмма помечается как ошибочная. Проектировщик может отказаться использовать вторую диаграмму или исправить ее, чтобы диаграммы стали семантически согласованными.

### **Заключение**

Использование предложенной модификации временной RV-грамматики позволяет уменьшить количество «дорогих» ошибок при проектировании САС. Дальнейшее направление работ связано с интеграцией предложенного метода с онтологиями, построенными на основе документальных артефактов проектного предприятия (отчеты, текущие документы, нормативные документы, методики и др.)

### **Список литературы**

1. Афанасьев А.Н., Гайнуллин Р.Ф., Афанасьева Т.В. Семантический анализ диаграмматических моделей в проектировании сложных автоматизированных систем // Открытые семантические технологии проектирования интеллектуальных систем. 2015. № 5. С. 385-388.
2. Афанасьев А.Н., Кириллов С.Ю., Бригаднов С.И. Разработка RV-грамматики для диаграммного языка BPMN // Информатика, моделирование, автоматизация проектирования. Сборник научных трудов VII Всероссийской школы-семинара аспирантов, студентов и молодых ученых (ИМАП-2015). 2015. С. 68-75.

## АВТОМАТИЗИРОВАННОЕ ИЗВЛЕЧЕНИЕ ПРИЧИННО-СЛЕДСТВЕННЫХ ОТНОШЕНИЙ ИЗ ТЕКСТОВ ПРОЕКТНЫХ ДОКУМЕНТОВ

А.А. Куликова<sup>55</sup>

В данной статье предлагается механизм извлечение пар текстовых конструкций из текстов проектных документов, связанных отношением типа «причина – следствие». Данный механизм порождает инструментальное средство, которое может быть применено на этапе концептуального проектирования с целью построение модели задачи, а также достижения её более глубокого понимания.

### Введение

Вопрос автоматического анализа текста на естественном языке на предмет извлечения из него различных типов связей является актуальной задачей, возникшей на стыке прикладной лингвистики и вычислительной техники.

Кроме того, задача анализа проектных документов имеет принципиальное значение в процессе проектирования автоматизированных систем, поскольку именно в проектных документах содержится наиболее полная информация о проекте, которая может быть использована для принятия решений. Извлечение же из таких документов семантической информации в виде концептов, связанных отношением типа «причина – следствие», способствует достижению проектировщиком более глубокого понимания стоящей перед ним задачи, а также является основой для последующего построения моделей прецедентов – интеллектуально обработанных способов решения конкретных задач, накопленных в соответствующей базе опыта.

Таким образом, исследование, представленное в данной статье, направлено на облегчение процесса концептуального проектирования и снижение количества ошибок и неточностей, возникающих на данном

---

<sup>55</sup> 432027, Ульяновск, ул. Северный Венец, 32, УлГТУ, e-mail: a.push1206@gmail.com

этапе и, как следствие, повышение успешности разработок автоматизированных систем.

В связи с тем, что извлечение причинно-следственных отношений из текстов на естественном языке используется нами в прикладных целях, основным критерием успешности нашего исследования является не столько точность выявления таких отношений, сколько применимость разработанного нами механизма в рамках комплекса средств онтологического управления проектами. Подробно структура данного комплекса описывается в [1] и [2], другие аспекты и функциональности комплекса представлены в [3] и [4]. Онтологическое представление семантической информации, извлеченной из текстов проектных документов, используется нами, как наиболее удобная форма представления знаний о проекте, которая позволяет не только накапливать их, но также анализировать и использовать их в прикладных целях.

В данной работе мы рассматриваем алгоритм автоматизированного извлечения причинно-следственных отношений как для русского, так и для английского языков.

## **1. Обобщённый механизм извлечения причинно-следственных связей из проектных документов**

Причинно-следственными отношениями связываются множество вершин именных, глагольных и других синтаксических групп, а также единичных лексем [5] – назовём их текстовыми единицами. Пары таких текстовых единиц предлагается извлекать с помощью программных агентов (для английского и русского языков соответственно).

Результат анализа текстов используется специальным инструментом онтологического контроля, который позволяет проектировщику получить дополнительную (скрытую, неочевидную) информацию о проекте и сделать шаг на пути к построению модели прецедента (см. рис. 1).

Обозначим отношение типа «причина – следствие» следующим образом:

$$REASON(X,Y), \quad (1)$$

где  $X$  является причиной  $Y$ , а вместо  $REASON$  может стоять соответствующий тег или предикат (если он есть).

Ключевыми компонентами в процессе извлечения связей подобного типа являются агент, который обеспечивает автоматическое извлечение таких связей на основе набора маркеров (тегов) причинно-следственных связей, сигнализирующих о наличии связи в тексте, – для русского и английского языков соответственно, – и процедура онтологического контроля. Рассмотрим каждый из них подробнее.

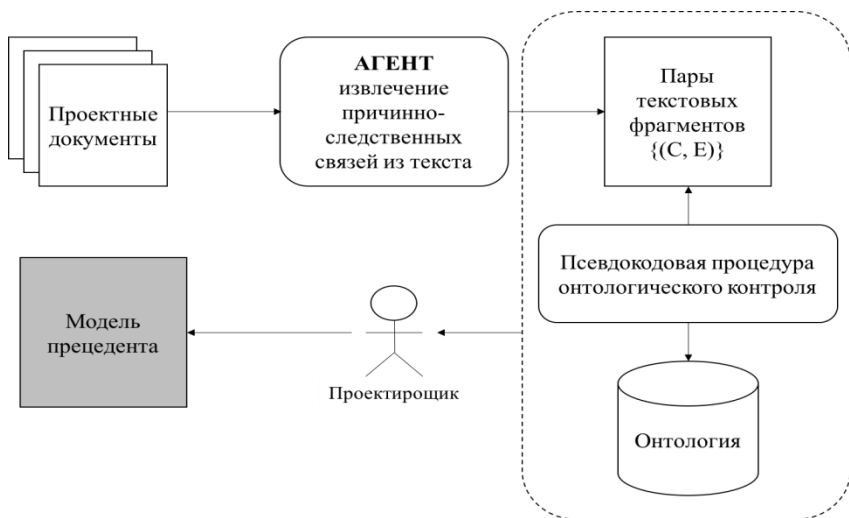


Рис. 1. Обобщенная схема извлечения причинно-следственных связей

## 2. Маркеры выражения причинно-следственных отношений в языке

Некоторые лингвисты разделяют семантические типы высказываний, содержащих причинно-следственные отношения, на два типа (в зависимости от типа причины):

- высказывания с предметной причиной (cause-and-effect);
- высказывания с логической причиной (reason-and-result).

Также выделяют несколько типов маркеров, выражающих причинно-следственные отношения:

- 1) предлоги (в простых предложениях), союзы и союзные слова (как правило, разделяют две части сложноподчиненного предложения);
- 2) устойчивые выражения с участием существительного или прилагательного, а также существительные, сигнализирующие о наличии такой связи (зачастую имплицитно в связи с их многозначностью);
- 3) каузативные глаголы (выражают причинно-следственные отношения в некоторых случаях имплицитно, т.е. неявно).

Среди маркеров первого типа:

- союзы: *because, as, since, inasmuch as, now that, as long as, such ... that, so ... that, so, for this reason, потому что, поскольку, так как* и т.д.;

- предлоги: *because of, due to, owing to, on account of, in order to, из-за, в результате, благодаря* и т.д.;
- вводные слова: *therefore, consequently, as a result, accordingly, for this reason, hence, thus, nevertheless, that's why, следовательно, таким образом, по этой причине, соответственно* и т.д.

Среди маркеров второго типа:

- эксплицитные (явные): *to be an influence on, to be the effect of, the result is, as a result of, to be the reason for, to be the cause of, to be the factor of, to have an impact on, to have an influence on, to be responsible for, быть/стать причиной, быть/стать следствием, оказать влияние на* и т.д.;
- имплицитные (неявные): *antecedent, ground, justification, motivation, motive, necessity, occasion, start* и т.д.

Среди маркеров третьего типа:

- эксплицитные (как правило, употребляются как в активном, так и в пассивном залоге): *to result in/from, to cause, to produce, to lead to, to affect, to contribute to, to determine, to influence, to bring about, to generate, to induce, to occasion, to secure, to set off, to stir up, to touch off, to trigger, to force, to produce, вызывать, приводить к, обеспечивать, следовать из* и т.д.;
- имплицитные: *let, get, make, keep, put* (в некоторых значениях), *order, ask, persuade, mean* и т.д.

Мы проанализировали несколько теоретических работ [5-12], посвященных способам выражения причинно-следственных отношений и сформировали следующую классификацию, представленную в таблице 1.

Таблица 1. Классификация тегов, выражающих причинно-следственные отношения в тексте

	Тип тега	Правила выделения фрагментов	Примеры тегов
A	Маркирует причину	Чаще всего стоит между следствием и причиной. В качестве причины следует рассматривать все предложение до точки, запятой или другого знака препинания. В качестве следствия – начало предложения (если тег встречается посередине предложения), или оставшуюся часть предложения между запятой и точкой (если тег стоит в начале предложения).	<i>because, as, since, inasmuch as, as long as, now that</i>  <i>потому что, поскольку, так как</i>
		Y, <A> X. или <A> X, Y.	

Таблица 1. Продолжение

	Тип тега	Правила выделения фрагментов	Примеры тегов
<b>B</b>		<p>Чаще всего стоит между следствием и причиной. В качестве причины следует рассматривать лексическую группу после тега. В качестве следствия – начало предложения (если тег встречается посередине предложения), или оставшуюся часть предложения между запятой и точкой (если тег стоит в начале предложения).</p> <p>Y, &lt;B&gt; [X] (...). или &lt;B&gt; [X] (...), Y.</p>	<p><i>because of, due to, owing to, on account of, as a result of</i></p> <p><i>из-за, в результате, благодаря</i></p>
<b>C</b>		<p>Сказуемое (предикат), относящееся к причине. Подлежащее следует рассматривать как причину. Именную группу после маркера – как следствие.</p> <p>&lt;C&gt; (X, Y)</p> <p>В пассивном залоге (при наличии формы пассивного залога) – см. тип F.</p>	<p><i>be/have an influence on, be the reason for, be the cause of, be the factor of, have an impact on, be responsible for, result in, cause, produce, lead to, affect, contribute to, determine, influence, bring about, generate, induce, occasion, secure, set off, stir up, touch off, trigger, force, produce</i></p> <p><i>вызывать, приводить к, быть/стать причиной, обеспечивать</i></p>
<b>D</b>	Маркирует следствие	<p>Чаще всего стоит в начале предложения и маркирует следствие (до знака препинания). При этом причина может содержаться в предыдущем предложении (если предложение, промаркированное тегом, простое), или в начале предложения (если тег стоит в середине).</p> <p>X. &lt;D&gt; Y(, ...). или X, &lt;D&gt; Y.</p>	<p><i>for this reason, so, therefore, consequently, as a result, accordingly, for this reason, hence, thus, nevertheless, that's why, the result is</i></p> <p><i>следовательно, таким образом, по этой причине, соответственно</i></p>



Таблица 1. Продолжение

	Тип тега	Правила выделения фрагментов	Примеры тегов
<b>E</b>		Чаще всего стоит между причиной и следствием. В качестве следствия следует рассматривать лексическую группу после тега. В качестве причины – начало предложения (если тег встречается посередине предложения), или оставшуюся часть предложения между запятой и точкой (если тег стоит в начале предложения).  X, <E> [Y] (...). или <E> [Y] (...), X.	<i>in order to</i>  <i>чтобы, для того чтобы</i>
<b>F</b>		Сказуемое (предикат), относящееся к следствию. Подлежащее следует рассматривать как следствие. Именную группу после маркера – как причину.  <F> (Y, X)	<i>be the effect of, result from</i>  <i>следовать из</i>
<b>G</b>	Маркирует и причину, и следствие	Парные теги. Первый тег маркирует причину, второй – следствие  (...) <G1> X, <G2> Y (...).	<i>such ... that, so ... that</i>  <i>так... что...</i>

После выполнения агентом своей работы мы получим **N** пар текстовых конструкций (фрагментов) типа **(C, E) = ([причина], [следствие])**. Условно обозначим фрагмент, относящийся к причине, как **F<sup>C</sup>**, а к следствию – как **F<sup>E</sup>**. Каждый фрагмент состоит из последовательности лексических единиц (слов): **F = {W}**.

### 3. Процедура онтологического контроля

Процедура онтологического контроля использует онтологию с целью извлечения дополнительной информации о проекте на основе полученных пар текстовых конструкций, с одной стороны, а с другой – пополняет онтологию новыми найденными в проектных документах отношениями.

Ниже представлен алгоритм процедуры онтологического контроля:

- 1) Выбрать фрагмент **F<sub>i</sub>** из множества **{F}**
- 2) **Если** множество **{F}** не пусто, **то** перейти к (3), **иначе конец**
- 3) Выбрать слово **W<sub>i</sub>** из множества **{W}**

- 4) *Если* множество  $\{W\}$  не пусто, *то* перейти к (5), *иначе* перейти к (11)
- 5) *Если* слово  $W_i$  входит во множество концептов онтологии  $\{C\}$ , *то* перейти к (6), *иначе* перейти к (11)
- 6) *Если* слово  $W_i$  связано отношением типа «причина – следствие» с каким-либо концептом онтологии  $R^{CE}(W_i, C^{CE})$ , *то* перейти к (7), *иначе* перейти к (8)
- 7) Вывести пару  $R^{CE}(W_i, C^{CE})$  на экран
- 8) *Если* слово  $W_i$  связано отношением типа «синоним» с каким-либо концептом онтологии  $R^S(W_i, C^S)$ , *то* перейти к (9), *иначе* перейти к (11)
- 9) *Если* слово  $W_i$  связано отношением типа «причина – следствие» концептом  $C^S$  онтологии  $R^{CE}(W_i, C^S)$  и ( $F_i = F^C$  и слово  $W_i$  является главным) *или* ( $F_i = F^E$  и слово  $W_i$  является зависимым), *то* перейти к (11), *иначе* перейти к (10)
- 10) Добавить отношение  $R^{CE}(W_i, C^S)$  в онтологию
- 11) Перейти к (3)

### Заключение

Отметим, что в качестве потенциала для развития данного алгоритма может использоваться схема на основе анализа англоязычных и русскоязычных тезаурусов и корпусов текстов, используемая группой учёных из США [13] для извлечение из текстов отношений типа «часть – целое», также основанная на лексико-синтаксических паттернах и тегах. Перечень таких паттернов формируется в несколько этапов:

- 1) Извлечение пар концептов, связанных отношением типа «часть – целое», из электронного тезауруса (семантической сети) WordNet для английского языка и РуТез – для русского.
- 2) Составление классификации и собственной иерархии данных пар.
- 3) Поиск данных пар в большом корпусе текстов общей тематики, извлечение предложений, содержащих такие пары, и выявление лексико-семантических паттернов, выражающих отношение заданного типа.
- 4) Составление классификации паттернов и их иерархии в зависимости от частотности и других факторов.

Решение о целесообразности выполнения данной работы будет приниматься после получения экспериментальных данных о работе алгоритма, описанного выше.

## Список литературы

1. Sosnin P., Pushkareva A., Negoda V. (2018) Ontological Support of Design Thinking in Developments of Software Intensive Systems. In: Abraham A., Kovalev S., Tarassov V., Snasel V., Vasileva M., Sukhanov A. (eds) Proceedings of the Second International Scientific Conference "Intelligent Information Technologies for Industry" (IITI'17). IITI 2017. Advances in Intelligent Systems and Computing, vol 679
2. П.И. Соснин, А.А. Пушкарева. Средства онтологического сопровождения проектного мышления // Труды Конгресса по интеллектуальным системам и информационным технологиям «IS&IT'17». Научное издание в 3-х томах. – Таганрог: Изд-во Ступина С.А., 2017. – Т. 2. – с. 366-374
3. P. Sosnin, A. Pushkareva. Ontological Controlling the Lexical Items in Conceptual Solution of Project Tasks; LNCS, volume 10409, pp 31-46
4. Anna Kulikova. Automated Discovery of Part-Whole Relations to Assist Semantic Modelling of a Project. In: Interactive systems: Problems of Human - Computer Interaction. – Collection of scientific papers. – Ulyanovsk: USTU, 2017. – p. 118-123
5. Маслов П.П. Обнаружение и извлечение причинно-следственных закономерностей из текста на естественном языке. [Электронный ресурс] – Режим доступа: <http://math.nsc.ru/conference/zont09/reports/24MaslovPP.pdf>
6. Mary Nell Sorensen. Logical connectors. URL: <https://staff.washington.edu/marynell/grammar/logicalconnectors.html>
7. Шапиро Э.Д. Выражение причинно-следственных отношений в сложноподчиненных предложениях различных типов. [Электронный ресурс] – Режим доступа: <http://cyberleninka.ru/article/n/vyrazhenie-prichinno-sledstvennyh-otnosheniy-v-slozhnopodchinennyh-predlozheniyah-raznyh-tipov>
8. Бессалов А.Ю. Каузативные глаголы как средство выражения причинно-следственных отношений в английском и французском языках. [Электронный ресурс] – Режим доступа: <http://www.vestnik-mgou.ru/Articles/Doc/1448>
9. Cause and Effect. E-reading Worksheets. URL: <http://www.ereadingworksheets.com/text-structure/patterns-of-organization/cause-and-effect/>
10. Cause & Effect Word List and Notes. URL: [https://www.iuj.ac.jp/iep/inside/IEPweb/htdocs/Documents/TS\\_VocabSelfAccess/VocabModule3-2012/a.CauseandEffectWordList&Notes.pdf](https://www.iuj.ac.jp/iep/inside/IEPweb/htdocs/Documents/TS_VocabSelfAccess/VocabModule3-2012/a.CauseandEffectWordList&Notes.pdf)
11. Argumentative Strategy: Cause & Effect // [https://www.ln.edu.hk/eng/rhetoric/Argumentative/Cause\\_and\\_Effect.html](https://www.ln.edu.hk/eng/rhetoric/Argumentative/Cause_and_Effect.html)
12. Васильева Э.П. Состав и структура поля каузальности в английском газетном дискурсе. [Электронный ресурс] – Режим доступа: <http://engphil.samsu.ru/laniuscms/docs/grammatical/vasilieva4.pdf>
13. Roxana Girju, Adriana Badulescu, Dan Moldovan. Automatic Discovery of Part-Whole Relations. URL: [http://www.hlt.utdallas.edu/~adriana/Publications/PartWhole\\_CL2006.pdf](http://www.hlt.utdallas.edu/~adriana/Publications/PartWhole_CL2006.pdf)

Данная работа поддержана грантом Российского фонда фундаментальных исследований (РФФИ), грант № 15-07- 04809а, и выполнена в рамках государственного контракта № 2.1534.2017/ПЧ.

## РАЗРАБОТКА МОБИЛЬНОГО ПРИЛОЖЕНИЯ “ВЕЛОТРЕКЕР”

С.А. Никулин<sup>56</sup>

Данная статья посвящена разработке мобильных приложений. Анализу требований при проектировании и реализации мобильных приложений. Данный анализ был проведен на примере разработки мобильного приложения “Велотрекер” для мобильных операционных систем Android.

### Введение

В настоящее время все больше пользователей уже перестают пользоваться настольными компьютерами, отдавая свое предпочтение персональным мобильным средствам, которые позволяют быть на связи со всем миром в режиме 24/7. В отличии от настольного компьютера, мобильный телефон или планшет - это более личностное устройство. В них хранятся фотографии пользователя, номера его кредитных карт, с которых выполняются платежи, аккаунты социальных сетей. Некоторые приложения уже способны помогать нам в планировании действий: стоит только запустить приложение, приняв предварительно соглашение, и через некоторое время оно вычислит местоположение вашего офиса и дома; после чего будет подсказывать сколько времени добираться до работы и за сколько необходимо проснуться, чтобы не пропустить важное совещание или встречу.

Целью данной работы является разработка мобильного приложения для ОС Android для эффективного взаимодействия пользователя и серверной базы данных. Разрабатываемое приложение основной функцией содержит ведение и сохранение статистики для велопутешествий. Кроме того, приложение будет позволять делиться своими достижениями с друзьями в социальных сетях.

В соответствии с целью в работе определены следующие задачи:

- Анализ методов проектирования информационных систем и мобильных приложений;

---

<sup>56</sup> 432027, Ульяновск, ул. Северный Венец, 32, УлГТУ, e-mail: san2008ni@mail.ru

- Исследование методов реализации клиент-серверной архитектуры;
- Проектирование мобильного приложения, его реализация и описание.

## 1. Анализ предметной области.

Мобильный велотрекер – это приложения для смартфона, позволяющее отслеживать активность велосипедиста с возможностью построения маршрута и другими возможностями. Рассмотрим некоторые из существующих мобильных приложений такого рода.

В онлайн-магазинах iOS и Android существует большое количество разнообразных приложений для поклонников велоспорта. Но всех их можно разделить на несколько типов:

- Измерители – вычисляют скорость при помощи GPS и считают преодоленное расстояние;
- Планировщики – помогают составить маршрут поездки;
- Сервисные программы – включают в себя напоминания, справочники и мануалы по ремонту;

Социальные программы – позволяют пригласить друзей для совместной поездки и позволяют делиться с ними своими достижениями.

Интернет Вещей - это сложная концепция включающая в себя большое количество технологий. В самом простом представлении это идея взаимодействия между умными устройствами, встроенными во все окружающее нас - здания, одежду оборудование и т.д. Использование IoT позволит сделать жизнь более безопасной, позволит более эффективно контролировать здоровье, позволит повысить эффективность производств.

В последнее время наблюдается значительный рост интереса пользователей к данной теме. Количество запросов в поисковых системах за последние два года начало быстро возрастать (рис. 1 и 2).

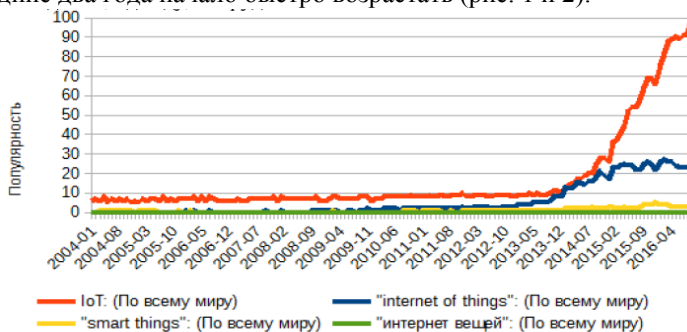


Рис.1. Статистика поисковых запросов Google (в мире)

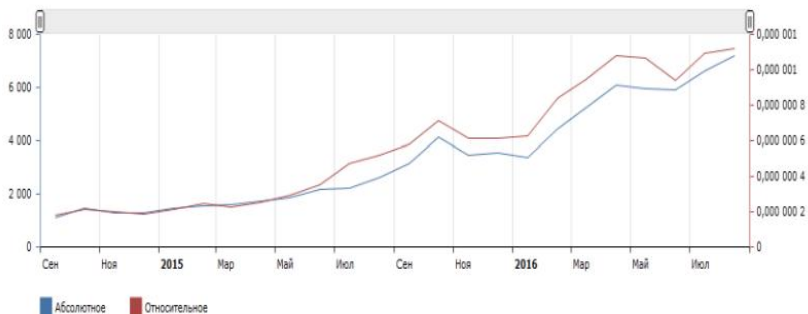


Рис. 2 История показов по фразе "IoT" Яндекса

Сферы использования IoT определяют требования к качеству связи, которые могут определяться как экономическими, так и техническими характеристиками. В некоторых случаях может возникнуть необходимость использования «уникальных» технологий для организации связи с целью обеспечения наибольшей эффективности. Требования варьируются довольно широко от приложения к приложению.

## 2. Платформа для разработки.

Android OS являются оптимальной платформой для реализации предлагаемого мобильного приложения. На Android OS работает более 85% устройств. Смартфоны на Android принадлежат к широкому спектру ценовых категорий, что позволяет, при разработке под них, охватить широкую аудиторию.

IntelliJ IDEA является популярной IDE поддерживающая широкий спектр возможностей для разработки, в том числе и разработку под Android OS. Так как для программирования на Android необходимо компилировать под JVM, а IntelliJ IDEA является удобным и популярным инструментом для работы с JAVA, то было принято решение использовать эту среду разработки.

Был проведен выбор шаблонов разработки мобильных приложений. В результате был выбран паттерн MVC – широко распространенный шаблон разработки мобильных и web-приложений. Строение данного паттерна представление-контроллер-модель данных. Основная идея этого паттерна в том, что и контроллер и представление зависят от модели, но модель никак не зависит от этих двух компонент.

Были рассмотрены принципы реализации пользовательских интерфейсов Android приложений и учтены соответствующие потребности в их реализации.

### 3. Реализация.

При разработке некоторой системы ее функциональные возможности, то есть поведение, которым она должна обладать с точки зрения заказчика, документируются с помощью модели прецедентов. Модель прецедентов показывает, какие функции должна быть способна выполнять система, а также в какой среде она должна работать. Модель прецедентов разрабатывается на ранних этапах проектирования и дает возможность заказчику четко сформулировать свои требования, а инженеру-разработчику – понять, что нужно заказчику.

Основными элементами модели прецедентов являются актеры и прецеденты.

Субъекты может только вводить информацию в систему, только получать информацию из системы или делать и то, и другое. В роли субъекта может выступать также и другая система, если она будет взаимодействовать с разрабатываемой системой, и не будет являться ее неотъемлемой частью.

Прецеденты предназначены для моделирования диалога между системой и ее субъектами. Они представляют собой возможности, которые система может обеспечить конкретному субъекту.

На рисунке 3 представлена диаграмма прецедентов, которая представляет функции пользователя по работе с системой.

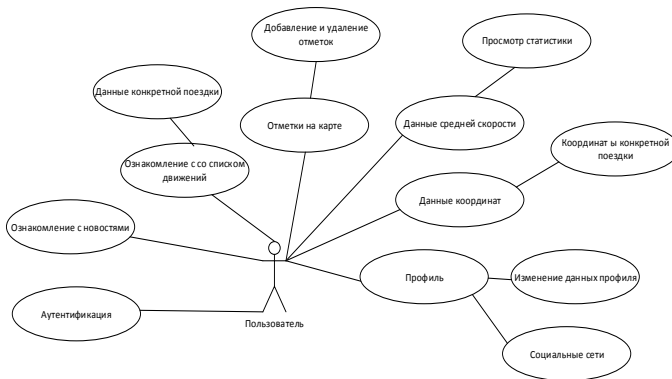


Рис. 3. Диаграмма прецедентов пользователя системы

Диаграмма взаимодействия это диаграмма, на которой представлено взаимодействие, состоящее из множества объектов и отношений между ними, включая и сообщения, которыми они обмениваются. Этот термин применяется к видам диаграмм с акцентом на взаимодействии объектов.

Диаграммы состояний используются для описания поведения, реализуемого в рамках варианта использования, или поведения экземпляров класса (объектов), компонента, узла или системы в целом. Поведение моделируется через автомат (англ. state machine), описывающий возможные последовательности состояний экземпляра сущности и переходы между ними на протяжении его жизненного цикла, начиная от создания и заканчивая уничтожением.

Диаграмма состояний (автомат) представляет собой связный ориентированный граф, вершинами которого являются состояния, а дуги служат для обозначения переходов из состояния в состояние. Под состоянием (англ. state) понимается ситуация в ходе жизни экземпляра сущности, когда эта ситуация удовлетворяет некоторому условию, экземпляр выполняет некоторые операции или ждет наступления некоторого события. Например, для объекта его состояние может быть задано в виде набора конкретных значений атрибутов, при этом изменение этих значений будет отражать изменение состояния моделируемого объекта. Диаграмма состояний приведена на рисунке 4.

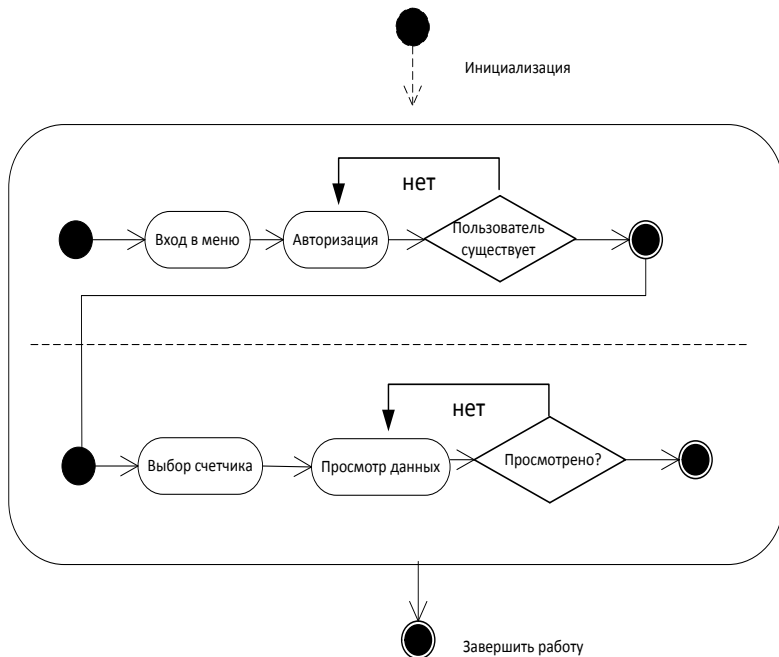


Рис. 4 Диаграмма состояний авторизации пользователя



## Заключение

Было разработано мобильное приложение “Велотрекер”. Основным функционалом, которого является отслеживание местоположения велосипедиста в пространстве, отображение этого местоположения и маршрута на карте. Разработанное приложение имеет поддержку Интернета вещей, которая заключается в возможности подключения носимых устройств и их работе совместно с приложением. Одним из таких устройств является пульсометр. Так же возможна передача информации собранной в ходе работы приложения на другие устройства.

## Список литературы

1. Android Studio - [Электронный ресурс] - Режим доступа: <http://developer.android.com/sdk/index.html> — Англ. яз. (Дата обращения: 25.05.2017).
2. Android Studio - обзор возможностей // Android helper. Освоим android вместе? URL: <http://android-helper.com.ua/android-studio/> — Англ. яз. (дата обращения: 02.09.2016).
3. Васвани В.А.: MySQL: использование и администрирование, СПб, Питер, 2011 г., 368 с.
4. Вигерс, Битти: Разработка требований к программному обеспечению, - М., Издательство: BHV, 2014 г., 736 стр.
5. Гурвиц Г.Е.: Microsoft Access 2011. Разработка приложений на реальном примере, - М., BHV, 2011 г., 424 с.
6. Исаев Г.А.: Проектирование информационных систем. Учебное пособие, - М., Омега-Л, 2015 г., 424 с.
7. Курлов А.А., Петров Е.А.: Методология информационной аналитики, - М., Проспект, 2014 г., 384 с.

**ЛОКАЛИЗАЦИЯ ВЕБ-РЕСУРСОВ  
(НА ПРИМЕРЕ БАЗОВОГО ПРОЕКТА ПО  
ЛОКАЛИЗАЦИИ САЙТА ADVANTSHOP.NET)**

Е.П. Соснина<sup>57</sup>, О.А. Утина<sup>58</sup>

Рассматривается задача локализации веб-приложений на примере реального проекта. Предметом исследования настоящей работы являются способы, проблемы и возможные решения при локализации веб-ресурса.

Область исследования нашей работы связана с современными задачами прикладной лингвистики, направленными на решение практических вопросов по локализации и интернационализации веб-ресурсов организаций. Востребованность такого рода технологий для различного рода веб-проектов и в сфере информационно-лингвистического обеспечения программных продуктов возрастает с каждым годом в связи с глобализацией электронного бизнеса. Важной задачей здесь также является установление международных контактов и освоение новых рынков сбыта продукции и услуг, поэтому актуальность вопросов, связанных, в том числе, с межкультурной коммуникацией, не вызывает сомнений.

В настоящее время термин «локализация» используется достаточно часто не только в сфере информационных технологий, но и в теории и практике перевода, компьютерной лингвистике, межкультурной коммуникации. Вопросы локализации веб-ресурсов обсуждаются в работах Б. Эсселинка [1, 2], Э. Пима [3], Сосниной Е.П. [4], Сухаревой Е.Е., Шуриной О. В.[5] и других.

Локализация текстов подразумевает часто не только профессиональный перевод, и техническую адаптацию контента и кода, но и передачу лингвокультурных аспектов (описывающих какой-либо продукт, изделие, услугу) к особенностям определенной страны, региона или группы населения, национального языка [4]. Большое значение также придается рекламно-маркетинговой кампании на рынке продвижения

---

<sup>57</sup> 432027, Ульяновск, ул. Северный Венец, 32, УлГТУ, e-mail: ling@ulstu.ru

<sup>58</sup> 432027, Ульяновск, ул. Северный Венец, 32, УлГТУ, e-mail: ling@ulstu.ru

продукта, при проведении которой учитываются особенности менталитета конкретного народа. Итогом локализации должен стать продукт, воспринимаемый целевой аудиторией не как переведенный, а как созданный на родном языке потребителей.

Цель и задачи нашего исследования непосредственно направлены на рассмотрение современных технологий локализации, выявление проблем при локализации на конкретных проектах и рассмотрение возможных путей решения выявленных проблем. Для примера был выбран ресурс - [www.advantshop.net](http://www.advantshop.net).

В ходе проекта мы разделили процесс локализации веб-ресурса Advantshop на следующие этапы:

- 1) рассмотрение технического задания компании на разработку;
- 2) анализ интернациональных веб-ресурсов;
- 3) локализация базовых элементов веб-ресурса [www.advantshop.net](http://www.advantshop.net);
- 4) seo-оптимизация и верстка базовой локализованной версии веб-ресурса.

Языком локализации был выбран английский язык, потому что компания планирует выход на западный рынок, где население по большей части является англоговорящим.

Данный проект выполнялся на базе компании AdvantShop, которая занимается разработкой специализированной платформы для создания интернет-магазинов. Однако, для данной работы при локализации веб-ресурса было принято решение расширить сферу услуг, предоставляемых для западного рынка – помимо платформы для работы интернет-магазина в компании разрабатывается решение для бизнеса (решение для планирования ресурсов предприятия). В качестве веб-ресурса указанной компании выступает новая версия официального сайта AdvantShop (<https://www.advantshop.net>).

При локализации веб-ресурса необходимо соблюдать требования по:

1. Адаптивному переводу.
2. Корректному переводу терминологии в соответствии с глоссариями для данной сферы.
3. Контролю за служебными тегами, не подлежащими переводу, но присутствующими в файлах веб-ресурса (при нарушении может пострадать функциональность продукта).
4. Подбору переводческих эквивалентов с учетом SEO-задач.

На этапе анализа веб-ресурсов конкурентов было выявлено, что структура интернациональной версии веб-ресурсов компаний не всегда совпадает со структурой их русскоязычных версий и имеет ряд отличий. То есть необходимо придерживаться стратегии *адаптивного перевода*. Например, в подразделах меню исходного российского сайта нашей

компания представлено больше пунктов, и наша задача была, в том числе, связана с локализацией базового проекта интернационального сайта с учетом его развития в перспективе и опущением нерелевантного контента. Например, раздел *Note* представляет выборочный перевод главной страницы русской версии сайта. В переводной версии дается краткий обзор услуг компании: изъята информация об обучающем курсе, который записывался на русском языке. Также отсутствует блок *Готовые интеграции*, так как информация об интеграциях платформы с российскими компаниями является нерелевантной для англоговорящих пользователей.

Этап локализации элементов веб-ресурса включает в себя работы с навигационной структурой сайта, текстовым контентом, медиа-материалами на страницах.

Несмотря на кажущуюся простую структуры базового проекта, нами было локализовано более двухсот элементов и модулей, включая информационные сообщения, графические и медиа-материалы. Перевод производился с учетом устоявшихся переводческих соответствий, а также применялась терминологическая база Microsoft.

На этапе локализации контента переводился и адаптировался текстовый контент сайта для базовых страниц: заголовки, тексты блоков, контакты, включая также все графические сообщения. Например, на сайте компании AdvantShop в блоке *Примеры магазинов* в русской версии сайта представлены скриншоты клиентской части интернет-магазинов, где, соответственно, представлена информация на русском языке. Мы заменили данные графические файлы на новые, созданные специально для локализованной версии данного раздела локализованные версии интернет-магазинов клиентов.

При локализации ресурса необходимо также учитывать требования по SEO. На этапе поисковой оптимизации был произведен подбор семантического ядра для интернациональной версии сайта компании (более 600 ключевых слов и словосочетаний), составление заголовков к страницам, составление тегов <title> и мета-тегов description и keywords, с учетом анализа их конкуренции и частотности по GoogleAdwords. Например:

- <meta name="description" content="All-in-one system for business management. Platform for creating and running an online store. Free support and user-friendly interface. Create an online store on the AdvantSale platform in 5 minutes. Try it for free." />
- <meta name="keywords" content="all software in one website, all in one management software, running online store, online store builder, build ecommerce website, ecommerce system, ecommerce solutions

company, erp crm, erp system management, small business erp solutions" />

В процессе локализации исследуемого нами веб-ресурса исходный сайт подвергался динамическому преобразованию разработчиками (*agile-разработка*), что частично создавало трудности при выполнении поставленных нами задач и верстке, так как для качественной локализации сложного проекта необходимо четкое техническое задание, утвержденное у заказчика. Поэтому, для локализации на первом этапе был выбран базовый фреймворк сайта. При этом, общее число сверстанных строк даже этой базовой версии составило около полутора тысяч.

Выполненная базовая версия проекта по локализации веб-ресурса AdvantShop (<https://www.advantshop.net>) представлена в сети в виде отдельного ресурса <http://advantsale.com/index.html>. Проект нацелен на выход компании на западный рынок и привлечение иностранных клиентов. После запуска проекта в интернете необходима дальнейшая динамическая локализация остальных модулей исходного сайта и развитие самого продукта по требованиям и с учетом отзывов клиентов и компании, а также выработка SMM на иностранном языке, в том числе англоязычные ресурсы в соцсетях (Твиттер, Фейсбук и других).

### Список литературы

1. Cadieux P., Esselink B. GILT: Globalization, Internationalization, Localization, Translation //Globalization Insider. 2004. Vol. XI. Iss. 1.5. P. 1-5.[Электронный ресурс]// Режим доступа: <http://www.translationdirectory.com/article127.htm>
2. Esselink B. A Practical Guide to Localization / B. Esselink. – Amsterdam/Philadelphia : John Benjamins Publishing Company, 2000. – 488p.
3. Пым А. Website localization / А. Пым // The Oxford Handbook of Translation Studies / ed. by K. Malmkjaer, K. Windle. – Oxford : Oxford University Press, 2011. – P. 410–424.
4. Соснина Е. П. Локализация текстов как задача прикладного переводоведения // Язык. Культура, речевое общение: материалы международной научной конференции: в 2-х частях. - часть 2. - Москва: МПГУ, 2015 - с. 276-281.
5. Сухарева Е. Е., Шурлина О. В. Анализ качества перевода и локализации интернет-представительств компаний. // Воронежский государственный университет. ВЕСТНИК ВГУ. Серия: Лингвистика и межкультурная коммуникация. 2016. №1

**УДК 004.94**

## **РАЗРАБОТКА РАБОЧЕГО МЕСТА СЛЕСАРЯ-СБОРЩИКА С РЕКОМЕНДАТЕЛЬНОЙ СИСТЕМОЙ НА ИГРОВОМ ДВИЖКЕ UNITY<sup>59</sup>**

**А.С. Степанов<sup>60</sup>**

В этой статье описывается игровой движок Unity, процесс разработки в нём, на примере рабочего места слесаря-сборщика, а также рекомендательная система интегрированная в данное рабочее место.

### **Введение**

Графический пользовательский интерфейс слесаря-сборщика РЭАиП в тренажере обеспечивает компьютерное моделирование (имитацию) реального рабочего места слесаря-сборщика РЭАиП и технологического процесса сборки изделия (источника смещения). Внешний вид его представлен на рис. 1 и имеет следующую структуру: инвентарь, рабочее поле, панель подсказок.

---

<sup>59</sup> *Исследования поддержаны грантом Министерства образования и науки Российской Федерации, проект № 2.1615.2017/4.6.*

*Исследование выполнено при финансовой поддержке РФФИ и Правительства Ульяновской области в рамках научного проекта № 16-47-732152*

*Исследование выполнено при финансовой поддержке РФФИ в рамках научного проекта № 17-07-01417*

<sup>60</sup> 432027, Ульяновск, ул. Северный венец, 32, УлГТУ, e-mail: stepanovas@ido.ulstu.ru



Рис. 1. Экранный снимок тренажера

Инвентарь представляет собой панель в левом нижнем углу и содержит информацию о том надет ли браслет, какие предметы находятся в руках, или какие действия совершаются руками.

Панель подсказок – это тёмное поле справа от инвентаря, при наведении курсора на деталь, в нём отобразится более подробное описание детали, взятое из спецификации.

Рабочее поле представляет виртуальное пространство, виртуальное рабочее место и является в центре внимания обучающегося. В нём выполняются все функции тренажера (перемещение аватара, надевание антистатического браслета, работа на рабочем месте, все технологические операции при сборке изделия, работа с инструментом и т. п.). Управление движением аватаром выполняется с помощью клавиш клавиатуры: А – влево, S – назад, D – вправо, W – вперед. Выбор предмета осуществляется нажатием ЛКМ на нем.

Обзор аватара (вид) меняется при нажатии на карточку источника смещения, с помощью камеры обеспечивается хороший обзор при работе на виртуальном рабочем месте. Причем активация камеры происходит при нажатии ЛКМ, а деактивация камеры и перевод в общий вид происходит при нажатии на неё же, кроме того можно приблизить и отдалить вид с помощью колёсика мыши.

## 1. Краткое описание платформы Unity.

Unity — это инструмент для разработки двух- и трёхмерных приложений и игр, работающий под операционными системами Windows, Linux и OS X. Есть возможность создавать приложения для запуска в браузерах с помощью специального подключаемого модуля Unity (*Unity Web Player*), а также с помощью реализации технологии WebGL.

Приложения, созданные с помощью Unity, поддерживают DirectX и OpenGL. Активно движок используется как крупными разработчиками (Icepick lodge, InXile Entertainment, Harebrained Schemes, Moon Studios, Obsidian Entertainment), так и разработчиками Indie-игр (например, Inside, Slender: The Arrival, Surgeon Simulator 2013, Firewatch, Besiege и т. п.) в силу наличия бесплатной версии, удобного интерфейса и простоты работы с движком.

Редактор Unity имеет простой Drag&Drop интерфейс, который легко настраивать, состоящий из различных окон, благодаря чему можно производить отладку игры прямо в редакторе. Движок поддерживает три сценарных языка: C#, JavaScript (модификация). Расчёты физики производит физический движок PhysX от NVIDIA.

Одно из преимуществ заключается в том, что Unity основан на C# — языке программирования высокого уровня. Это важно, потому что в отличие от игровых движков, основанных на C++, у C# есть много уже внедрённых элементов и методов, которые можно просто брать и использовать — не тратя время на их воспроизведение».

Проект в Unity делится на сцены (уровни) — отдельные файлы, содержащие свои игровые миры со своим набором объектов, сценариев, и настроек. Сцены могут содержать в себе как, собственно, объекты (модели), так и пустые игровые объекты — объекты, которые не имеют модели («пустышки»). Объекты, в свою очередь содержат наборы компонентов, с которыми и взаимодействуют скрипты. Также у объектов есть название (в Unity допускается наличие двух и более объектов с одинаковыми названиями), может быть тег (метка) и слой, на котором он должен отображаться. Так, у любого объекта на сцене обязательно присутствует компонент Transform — он хранит в себе координаты местоположения, поворота и размеров объекта по всем трём осям. У объектов с видимой геометрией также по умолчанию присутствует компонент Mesh Renderer, делающий модель объекта видимой.



Также Unity поддерживает физику твёрдых тел и ткани, а также физику типа Ragdoll (тряпичная кукла). В редакторе имеется система наследования объектов; дочерние объекты будут повторять все изменения позиции, поворота и масштаба родительского объекта. Скрипты в редакторе прикрепляются к объектам в виде отдельных компонентов.

Редактор Unity поддерживает написание и редактирование шейдеров. Редактор Unity имеет компонент для создания анимации, но также анимацию можно создать предварительно в 3D-редакторе и импортировать вместе с моделью, а затем разбить на файлы.

Unity 3D поддерживает систему Level Of Detail (сокр. LOD), суть которой заключается в том, что на дальнем расстоянии от игрока высокодетализированные модели заменяются на менее детализированные, и наоборот, а также систему Occlusion culling, суть которой в том, что у объектов, не попадающих в поле зрения камеры не визуализируется геометрия и коллизия, что снижает нагрузку на центральный процессор и позволяет оптимизировать проект. При компиляции проекта создается исполняемый (.exe) файл игры (для Windows), а в отдельной папке — данные игры (включая все игровые уровни и динамически подключаемые библиотеки).

Модели, звуки, текстуры, материалы, скрипты можно запаковывать в формат .unityassets и передавать другим разработчикам, или выкладывать в свободный доступ. Этот же формат используется во внутреннем магазине Unity Asset Store, в котором разработчики могут бесплатно и за деньги выкладывать в общий доступ различные элементы, нужные при создании игр. Чтобы использовать Unity Asset Store, необходимо иметь аккаунт разработчика Unity. Unity имеет все нужные компоненты для создания мультиплеера. Также можно использовать подходящий пользователю способ контроля версий. К примеру, Tortoise SVN или Source Gear.

## **2. Рекомендательная система.**

Главная задача данного тренажёра это - обучение персонала, в связи с чем обучаемым позволено допускать ошибки в процессе обучения. Для отслеживания этих ошибок была разработана рекомендательная система, которая взаимодействует с тренажёром. В данном случае рекомендательная система представляет собой клиент-серверное приложение.

Почти каждое действие студента вызывает процесс формирования текущего состояния тренажёра, основанного на его логико-алгебраической

модели, и дальнейшей отправки сформированного состояния рекомендательной системе. Рекомендательная система содержит в себе список правил, которые могут создаваться, удаляться или изменяться учителем. Каждому правилу соответствует свой текст рекомендации. Состояние сравнивается с каждым правилом в списке и, если хотя бы одно правило соответствует состоянию, рекомендательная система отправляет обратно текст рекомендаций для каждого работавшего правила.

Пример рекомендаций:

- Вы использовали неверный конденсатор, внимательно изучите сборочный чертёж и спецификации;
- Вы не надели антистатический браслет;
- Вы использовали не те винты закрепляя трансформатор;
- Вы не установили дроссель, внимательно изучите сборочный чертёж.

## Заключение

Данная статья представляет использование информационных технологий в обучении. Данный подход позволяет упростить процесс обучения не только для обучающихся, но и для преподавателей, так как берёт часть их работы на себя (Поиск ошибок и путей их решения). Конечно тренажёр и рекомендательная система имеют ряд недостатков, например, для составления новых правил для рекомендательной системы необходимо обладать хотя бы базовыми навыками программирования и знать, как формируется состояние рабочего места в тренажёре.

## Список литературы

1. 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.
2. 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.
3. 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.
4. ИНОБР > Проекты > Компьютерные тренажерные системы [Электронный ресурс] – Режим доступа: [http://misis.ru/inobr/kts\\_inobr](http://misis.ru/inobr/kts_inobr), свободный.
5. Афанасьев А.Н., Войт Н.Н., Бочков С.И., Уханова М.Е., Ионова И.С. Разработка и исследование виртуальных рабочих мест в среде OpenSim. [Электронный ресурс] – Режим доступа: <http://elibrary.ru/item.asp?id=28092779>, свободный.
6. Технологии виртуальных миров в электронном обучении. Афанасьев А.Н., Егорова Т.М. [Электронный ресурс] – Режим доступа: <http://elibrary.ru/item.asp?id=23109124>, свободный.

## ИСПОЛЬЗОВАНИЕ МОДЕЛИ БАЗЫ ПРЕЦЕДЕНТОВ ДЛЯ АВТОМАТИЗАЦИИ ПОИСКА РЕШЕНИЙ ДЛЯ ЗАДАЧ, ВОЗНИКАЮЩИХ В ПРОЦЕССЕ ПРОЕКТИРОВАНИЯ

А.Н. Стратонов<sup>61</sup>

Данная статья предоставляет собой пример проектирования персональной базы прецедентов на базе программного комплекса WIQA. В рамках статьи будет рассмотрена комбинаторика прецедентов, а также представлена модель прецедента.

### Введение

Как известно любой человек должен стараться использовать те инструменты, которые он ранее использовал и чаще всего останавливает свой выбор на похожих инструментах. Персональная база прецедентов, позволит использовать ранее накопленный опыт индивида в его новых начинаниях, что существенно ускорит его работу. Скорость обработки информации человеком повысится, что также скажется на его продуктивности и качестве выполнения работ.

### Задачи персональной базы прецедентов

Наиболее важной частью любой работы является объявление задач. При проектировании персональной базы прецедентов необходимо решить следующие задачи:

- *Для повышения эффективности (Q1) персональной работы проектировщика в ситуациях (Q2) обнаружения необходимости внедрения при проектировании новых задач, необходимо разработать базу моделей прецедентов единицы которой, построены с использованием интеллектуального обработчика новых задач;*
- *База прецедентов должна обеспечивать свое развитие за счет очередных моделей прецедентов, реальных новых задач (Q3), в*

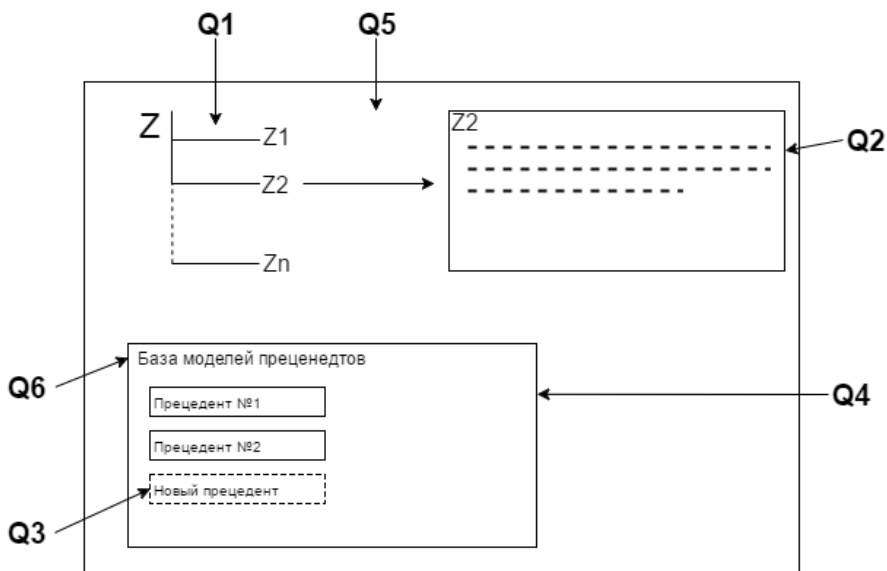
---

<sup>61</sup> 432027, Ульяновск, ул. Северный Венец, 32, УЛГТУ, e-mail: an.stratonov@gmail.com

построении которых активно используется онтологическое сопровождение, включая доступ к самой базе прецедентов (Q4);

- *База прецедентов должна быть составной частью* предметной базы опыта (Q5) и должна быть согласованной с компонентами программного комплекса WIQA, включая согласованность с онтологиями;

Наиболее важной частью базы моделей прецедентов, является его рациональное описание на естественном (или естественно-профессиональном) языке.



В итоге получаем следующие вопросы:

- Q.1. При каких условиях, при решении задачи, появляется новый прецедент?
- Q.2. В какой форме появляется новая задача?
- Q.3. Как будет осуществляться решение новой задачи? (design thinking)
- Q.4. Как будет формироваться модель прецедента? (интеллектуальное образование)
- Q.5. Как новая модель будет встраиваться в базу моделей прецедентов
- Q.5.1. Что из себя представляет база моделей прецедентов?

Q.5.2. Каков механизм встраивания нового прецедента в базу моделей прецедентов?

Q.6. Какие характеристики лежат в основе базы прецедентов?

За правильность употребления естественного языка отвечает его логика. А значит, в построениях и применениях естественно-языковых моделей прецедентов разумно использовать подходящие логики. В выборе полезных логик следует различать описания условных и реактивных составляющих прецедентов, ориентируясь при этом на «силы», оправдывающие, активизирующие и осуществляющие их повторное применение.

Для пояснения сказанного, введем следующую обобщенную логическую схему прецедента:

**Схема прецедента  $r_i$ :**

	<b>так как</b> [мотивы $M$ ]
<b>в</b>	<b>поскольку</b> [цели $C$ ]
<b>ы</b>	<b>если</b> [предусловия $U'$ ],
<b>б</b>	<b>то</b> реакция $r_q(t)$ ,
<b>о</b>	<b>из-за чего</b> [постусловия $U''$ ]
<b>р</b>	-----
	<b>есть альтернативы</b> $\{r_p\}$ .

Рис. 1. Схема прецедента

Описания мотивов и целей в модели (рис. 1) представляют контекст повторного применения прецедента с позиции «сил», оправдывающих целесообразность его применения и перехода к состоянию с ожидаемым постусловием.

Постусловие также связано с целями, требующими прецедент, но отражает их в форме состояния положения дел, к которому приведет применение образца прецедента. А цели определяют причины, по которым необходимо или полезно оказаться в этом состоянии.

Мотивы в оправданиях относятся к причинам либо личного, либо общественного характера, побуждающим субъекта к активности, в рамках которой потребовался прецедент. Перейдем к кодированию реакции в модели прецедента.

Такой код должен представлять рациональное естественно-языковое описание схемы действий, реализация которой приводит к повторному осуществлению прецедента.

## Логическое представление прецедента

В построениях логических моделей прецедентов, по сути дела, следует определиться с тем, какие виды логических формул и в каких случаях рационально использовать в логических моделях прецедентов, например, следующего вида:

**имя прецедента  $P_i$ :**

**в** так как [логическая формула (ЛФ) для мотивов  $M = \{M_k\}$   
**ы** поскольку [ЛФ для целей  $C = \{C_j\}$  ]  
**б** если [ЛФ для предусловия  $U' = \{U'_n\}$ ],  
**о** то реакция  $r_q(t)$ ,  
**р** из-за чего [ЛФ для постусловия  $U'' = \{U''_m\}$ ]  
 -----  
 есть альтернативы  $\{P_i(r_p)\}$ .

Рис. 2 Модель прецедента

Мотивы и цели являются составляющими, которые относятся к активности субъекта. Для логической композиции по совокупностям таких составляющих достаточны операции «И», «ИЛИ» и «НЕ». Построение каждой из таких композиций в существенной мере зависит от содержания прецедента и контекста его осуществления.

Кроме того, мотивы и цели учитываются в рациональном формировании модели прецедента, в основном, не для поиска подходящего прецедента, а для ориентации субъекта в повторном использовании прецедента и для его интеллектуального освоения. В поиске прецедента эти составляющие его модели могут выполнять только «грубую» функцию, но в выборе среди множества альтернатив роль этих составляющих важна.

А.4. Под интеллектуальном освоением прецедента понимается процесс ориентации субъекта для его повторного использования. Под моделью же прецедента понимается логическое описание с использованием логических операций «И», «ИЛИ» и «НЕ»

Перейдем к логическим формулам предусловия и постусловия. В этих логических конструктах связность признаков, выбранных для их спецификации, зависит от среды осуществления прецедента. В последующих рассуждениях будем различать только природные и компьютерные среды существования прецедентов.

Для адекватного описания (в предусловиях и постусловиях прецедентов) природных объектов и связей между ними может потребоваться (в оценках возможностей повторного использования) включение в условия составляющих, отражающих природные закономерности.

Более того, в природных средах и предусловия и постусловия могут включать составляющие, выражающие участие субъекта в осуществлении прецедента. Отмеченное указывает на то, что для логических представлений предусловий и постусловий прецедентов природных сред их логическая структура (по сравнению с формулами для отражения мотивов и целей) усложняется и может включать не только признаки, но и вычислимые числовые отношения между ними.

В числе прецедентов компьютерных сред будем выделять два их типа, к одному из которых относятся материализованные с помощью компьютерных средств природные прецеденты, а ко второму прецеденты, в осуществлении которых используются действия с компьютерными конструктами (например, активизация программных единиц, запись в файл, человеко-компьютерные диалоговые действия).

С построениями и использованиями прецедентов связано много задач, в число которых входят выявление прецедентов, создание их моделей для различных целей (например, для обеспечения эффективного поиска прецедента и решении задачи выбора при наличии альтернатив, интеллектуальное и практическое освоение прецедента), рационализация моделей и многие другие задачи.

### **Комбинаторика прецедентов**

Напомним, что интересы монографии связаны с реагированием человека на сложившуюся ситуацию с использованием образцов прецедентов, извлекаемых им из доступного опыта. Такой интерес переносится на методы решения задач, для построения которых используются образцы прецедентов, за каждым из которых стоит решенная ранее задача.

**А.2** Новая задача возникает как образ прецедента, решение и доказанность которого будет проведена в будущем.

Ответ на этот вопрос дает теория и практика решения задач в предметной области искусственного интеллекта.

**А.3.** решение задачи сводится на декомпозицию последней на множество более мелких задач, связанных между собой отношением И/ИЛИ, решение которых не вызывает затруднений.

Во-первых, к числу таких методов относятся переборные методы на И/ИЛИ-графах, в основу которых положена редукция (сведение) задач к подзадам.

На рисунке отражена ИЛИ-зависимость между альтернативными прецедентами, регистрируемая в их моделях, и И-зависимость между прецедентами, подчиненными реакциям  $\gamma$ , вложенным в прецеденты.

Из теории переборov на И/ИЛИ-графах, нацеленных на решение задач, через сведение их подзадачам, решение которых уже найдено, известно, что если решение у задачи  $Z$ , в базисе доступных прецедентов, есть, то оно будет построено. Напомним, что для задач, по их определению в сравнении с проблемами, всё, что необходимо для построения решения в доступном опыте имеется.

В комбинаторных методах на И/ИЛИ-графах, применяются различные версии переборov, в том числе и использующие эвристики, сокращающие в поиске решений задач объем переборov. Еще одним подходом к решению задач с использованием прецедентов является активность субъекта, осуществляемая по образцу экспертных систем. В таком подходе новая задача может быть решена:

- либо за счет адаптации к её условиям известного прецедента, образец которого хранится в базе знаний (базе опыта) экспертной системы;
- либо за счёт пошаговой сборки решения из прецедентов, каждый очередной из которых выбирается с учётом постусловий предшественников, что также можно квалифицировать как комбинаторику прецедентов.

### **Адаптация задачной ситуации**

В решении задач на базе прецедентов не следует забывать, что субъект  $S_{bi}$  является активной силой, которую, при необходимости и возможностям, можно использовать для адаптации задачной ситуации к условиям использования имеющихся образцов прецедентов.

A5.1. База прецедентов представляет собой систему хранения уникальных моделей прецедентов, персонализированных для конкретного проектировщика.

Такая практика распространена практически во всех видах деятельности, альтернативой чему является адаптация известного прецедента к новым условиям, существующим в задачной ситуации.

Разумеется, адаптацию к задачной ситуации следует использовать и в тех случаях, когда процесс решения задачи ещё не завершён и когда ещё не принято решение о связывании решаемой задачи с будущим прецедентом.

A.5.2 Если в задачной ситуации нечто отсутствует, что может оказать позитивное воздействие на построение решения задачи, а субъекту  $S_{bi}$  по



силам это нечто отсутствующее в ситуацию ввести, то следует предположить, что изменения задачной ситуации (в реальности) будут произведены, и продолжить построение решения.

Другими словами, постановку задачи в определённых пределах, допускаемых возможным и рациональным использованием активной силы субъекта, можно изменить, введя в неё информацию об изменённой задачной ситуации.

А.6. По итогу база прецедентов представляет собой уникальный, персонализированный набор прецедентов, доступ к которым осуществляется посредством WIQA, а поиск осуществляется каталогом ключевых слов, в ядре которых лежит онтологическое описание прецедента.

## Заключение

Выше неоднократно (явно или неявно) подчеркивалось, что любой деятельностью, управляют определенные мотивы, цели и условия, к которым необходимо обращаться оперативно с разными целями, что позволит перечислить основные цели работы в мотивационно-целевой форме.

**Основной мотив**, принимаемый в расчет, был связан с «повышением коэффициента полезного действия индивида в его персональной деятельности за счет рационального использования и хранения его персонализированного опыта в виде персональной базы прецедентов». **Мотивационная установка** исходила из того, что любой человек более ответственно и продуктивно использует те инструменты, которые он ранее использовал и чаще всего останавливает свой выбор на похожих инструментах. Персональная база прецедентов, позволит использовать **ранее накопленный опыт индивида в его новых начинаниях**, что существенно ускорит его работу.

## Список литературы

1. Ульяновск:–Соснин П.И. Человеко-компьютерная диалогика / П.И. Соснин УлГТУ.
2. Соснин П.И. Инструментарий вопросно-ответных рассуждений в корпоративной среде автоматизированного проектирования / П.И.Соснин №3.– 2004. –// Программные продукты и системы.
3. Карпушин А.Н., Соснин П.И. Моделирование мотивационно-целевых отношений в системах продукции. // А.Н. Карпушин. Труды девятой национальной конференции по искусственному интеллекту с международным участием. – М.: Физматлит, 2004.
4. . Соснин П.И., Касапенко Д.В., Принятие решений в экспертных вопросно- ответных средах // П.И.Соснин. Труды международной конференции «Интеллектуальные системы», – М.: Физматлит, –2008.

## РАЗРАБОТКА И ТЕСТИРОВАНИЕ КОНТРОЛЬНОГО ПРИМЕРА ДЛЯ РАЗРАБОТАННОГО МЕТОДА ПРОГНОЗИРОВАНИЯ<sup>62</sup>

И.А. Тимина<sup>63</sup>

В статье рассмотрен контрольный пример предложенной модели прогнозирования метрик проектной деятельности полученных посредством выделения нечетких тенденций данных проекта. Результатом исследования является выявление наилучшей гипотезы согласно моделям, алгоритмам и программам, решающим задачу организации автоматизированного метода прогнозирования проектных метрик.

### Введение

Одним из факторов рентабельной деятельности проектных организаций является постоянный анализ множества проектов крупной проектной организации на протяжении всего жизненного цикла. Задачу измерения характеристик проектной деятельности необходимо рассматривать как зависимость от создания средства для управления проектом, которое автоматизирует процессы кластеризации по сходству всех имеющих события проектов предприятия для последующего прогноза значений [1,2,3]. Решение проблемы мониторинга проектов предложено посредством применения средств анализа состояния метрик проектов программного обеспечения, загруженных в систему контроля версий.

При масштабах современных предприятий, занимающихся крупными проектами, отслеживание проектной деятельности занимает много времени, средств и человеческих ресурсов. Автоматизация данной деятельности главного инженера проекта обеспечит сокращение затрат, снижение нагрузки, упростит управление. Это достигается за счет снижения времени обработки данных по событиям хода проектирования, точности анализа, получению возможного прогноза посредством гипотез и минимизации его ошибки.

---

<sup>62</sup> Работа выполнена при поддержке РФФИ, проекты №16-47-732070, 16-47-730715.

<sup>63</sup> 432027, Ульяновск, ул. Северный Венец, 32, УлГТУ, e-mail: i.timina@ulstu.ru

## 1. План экспериментов

Для того чтобы выявить эффективность применения гипотез прогноза на основе корректировки, предложен следующий порядок действий:

- 1) анализируем временные ряды событий системы контроля версий [4,5,6];
- 2) выявляем пары с подобными тенденциями развития сущностей событий проекта [7];
- 3) прогнозируем по трем гипотезам, учитывая схожесть, корректируем полученный прогноз [8,9];
- 4) высчитываем ошибки прогнозирования.

Для этого будем использовать критерии оценки точности нечеткого моделирования числовых уровней ВР в виде мер качества MSE и RMSE:

Квадрат среднеквадратичной ошибки:

$$MSE = \frac{1}{n} \sum_{i=1}^n (e_i^2) = \text{mean}_{i=\overline{1,n}}(e_i^2),$$

Средняя квадратичная ошибка:

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (e_i^2)} = \sqrt{\text{mean}_{i=\overline{1,n}}(e_i^2)};$$

- 5) эксперт на основе своих знаний и полученных данных прогноза и ошибок по каждой гипотезе – делает вывод о выборе гипотезы и об адекватности предложенной модели и метода.

## 2. Исследование метрик открытых проектных

Открытый проект «FreeNAS10». Взяты ВР «закрытых» «Bug» и «Feature» (FreeNAS 10) [10]. В таблицах 1, 2 представлены результаты анализа и прогнозирования метрик проектной деятельности.

Табл. 1. Анализ данных проекта

Тип метки	Общая тенденция	Доминирующая тенденция	Мера сходства	Корреляция	Интерпретация корреляции
Bug	Падение	Стабильность	0,82	0,9736	Сильная
New Feature	Падение	Стабильность			

Табл. 2. Результаты прогнозирования с учетом влияния ВР-предиктора

ВР зависимый	ВР-предиктор	Гипотеза 1	Гипотеза 2	Гипотеза 3
Bug	New Feature	Стабильность	Рост средний	Падение сильное

В таблице 3 представлен прогноз появления «Bug». На рисунке 1 прогноз показан графически.

Табл. 3. Значения метрики «Bug» по результатам прогноза

Гипотеза	Значение метрики «Bug»			Тенденция, интенсивность ( $\hat{\tau}_{t+1}^Y$ )
	$\tau_{t+1}$	$\tau_{t+1}^Y$	$\hat{\tau}_{t+1}^Y$	
Гипотеза сохранения тенденции	7	5	3	Стабильность
Гипотеза устойчивости тенденции	15	13	11	Рост средний
Гипотеза прогноза на заданный период	5	3	1	Падение сильное

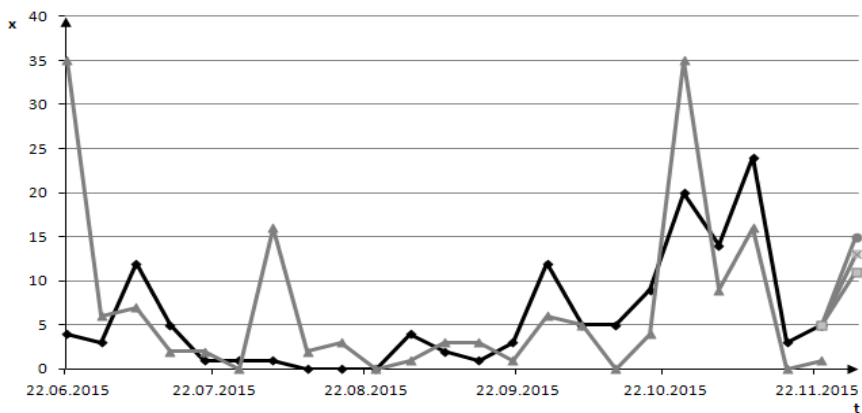


Рис. 1. Прогнозирование появления ошибок в проекте «FreeNAS10» с учетом влияния добавления новых функциональных возможностей в проект (гипотеза учета устойчивости тенденции)

Результаты оценки качества прогнозирования ошибок в разработке проекта с учетом влияния добавленных новых функций в проект представлены в таблице 4.

Табл. 4. Оценка качества прогноза («Bug-New Feature»)

Гипотеза	MSE			RMSE		
	$\tau_{t+1}$	$\tau_{t+1}^Y$	$\hat{\tau}_{t+1}^Y$	$\tau_{t+1}$	$\tau_{t+1}^Y$	$\hat{\tau}_{t+1}^Y$
Гипотеза сохранения тенденции	0,7656	0,765	1,5625	0,875	0,8747	1,25
Гипотеза устойчивости тенденции	0,1914	0,2104	0,006	0,4375	0,4587	0,0774
Гипотеза прогноза на заданный период	0,0017	0,0018	0,1151	0,0417	0,042	0,3392

Открытый проект «FreeNAS9». Взяты закрытые ВР «Bug» и «Feature» (FreeNAS 9) [11]. В таблицах 5, 6 представлены результаты анализа и прогнозирования метрик проектной деятельности.

Табл. 5. Анализ данных проекта

Тип метки	Общая тенденция	Доминирующая тенденция	Мера сходства	Корреляция	Интерпретация корреляции
Bug	Рост	Стабильность	0,75	0,9387	Сильная
New Feature	Рост	Стабильность			

Табл. 6. Результаты прогнозирования с учетом влияния ВР-предиктора

ВР зависимый	ВР-предиктор	Гипотеза 1	Гипотеза 2	Гипотеза 3
Bug	New Feature	Падение среднее	Рост сильный	Падение среднее

В таблице 7 представлен прогноз появления «Bug». Графически прогноз изображен на рисунке 2.

Табл. 7. Значения метрики «Bug» по результатам прогноза

Гипотеза	Значение «Bug»			Тенденция, интенсивность ( $\hat{r}_{t+1}^Y$ )
	$r_{t+1}$	$r_{t+1}^Y$	$\hat{r}_{t+1}^Y$	
Гипотеза сохранения тенденции	0	0	0	Падение среднее
Гипотеза устойчивости тенденции	567	584	600	Рост сильный
Гипотеза прогноза на заданный период	128	145	161	Падение среднее

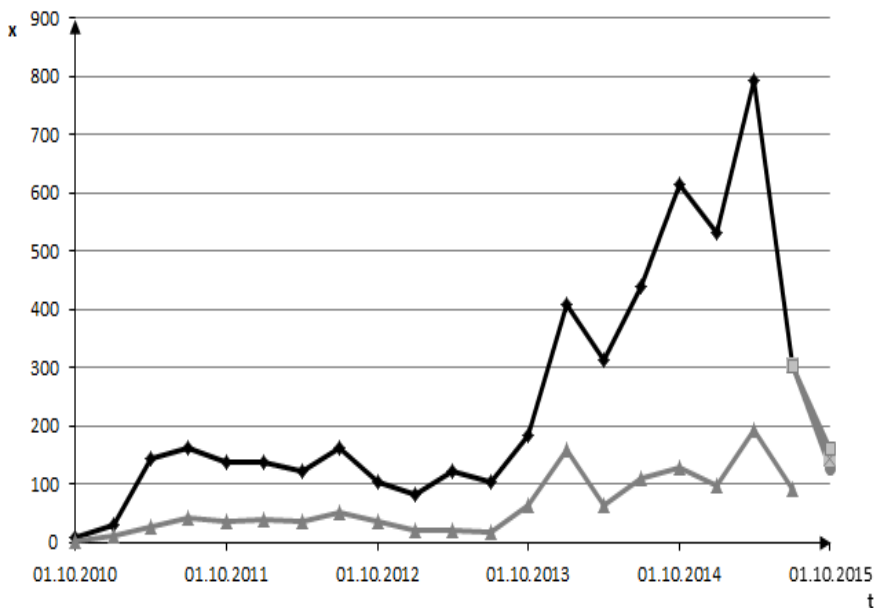


Рис. 2. Прогнозирование появления ошибок в проекте «FreeNAS9» с учетом влияния добавления новых функциональных возможностей в проект (гипотеза прогноза на заданный период)

Результаты оценки качества прогнозирования ошибок в разработке проекта с учетом влияния добавленных новых функций в проект представлены в таблице 8.

Табл. 8. Оценка качества прогноза («Bug - New Feature»)

Гипотеза	MSE			RMSE		
	$\tau_{t+1}$	$\tau_{t+1}^Y$	$\hat{\tau}_{t+1}^Y$	$\tau_{t+1}$	$\tau_{t+1}^Y$	$\hat{\tau}_{t+1}^Y$
Гипотеза сохранения тенденции	0,1097	0,0937	0,1042	0,3312	0,3061	0,3227
Гипотеза устойчивости тенденции	0,1097	0,0937	0,1042	0,3312	0,3061	0,3227
Гипотеза прогноза на заданный период	0,0003	0,0001	0,0001	0,0164	0,0087	0,0079

### 3. Эффективность предложенного метода прогнозирования

Согласно проведенным экспериментам открытых проектов по метрикам «Bug» и «New Feature», в проекте «FreeNAS10» была выявлена сильная корреляция и сходство между метриками, что свидетельствует о влиянии на прогноз ВР-предиктора. Так как основная доминирующая НТ ВР1 Y «Стабильность» (табл. 1) и доминирующая НТ ВР2 Z («New Feature») «Стабильность» при общей тенденции «Падение», было получено меньшее значение оценки качества MSE – не более 2% от среднего значения метрик при корректировке прогноза гипотезы устойчивости тенденции (табл. 4), что говорит о значимости предложенного подхода прогнозирования. В остальных случаях результаты исследования показали, что MSE составила более 2 % от среднего значения метрик.

В эксперименте открытого проекта «FreeNAS9» между метриками «Bug» и «New Feature» была выявлена сильная корреляция и сходство, что свидетельствует о влиянии на прогноз ВР-предиктора. Поскольку основная доминирующая НТ ВР1 Y «Стабильность» (табл. 5) и доминирующая НТ ВР2 Z («New Feature») «Стабильность» при общей тенденции «Рост», было получено меньшее значение оценки качества MSE – не более 2% от среднего значения метрик при корректировке прогноза гипотезы на заданный период (табл. 8), что говорит о значимости предложенного подхода прогнозирования.

### Заключение

Проведенные вычислительные эксперименты по исследованию эффективности предложенных методов и алгоритмов показали, что в половине случаев предложенный подход показывает улучшение качества прогнозирования по критерию MSE, который составляет не более 1-2% от среднего значения метрик прогноза.

### Список литературы

1. Pedrycz W., Chen S.M. Time Series Analysis, Modeling and Applications: A Computational Intelligence Perspective (e-book Google) // Intelligent Systems Reference Library. 2013. Vol. 47. P. 404.
2. 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.
3. Мошкин В.С., Пирогов А.Н., Тимина И.А., Шишкин В.В., Ярушкина Н.Г. Интеллектуальный анализ данных и онтологический подход в управлении проектами // Автоматизация процессов управления. 2016. Т. 4(46). С. 84-92.
4. Herbst G., Bocklish S.F. Online Recognition of fuzzy time series patterns // 2009 International Fuzzy Systems Association World Congress and 2009 European Society for Fuzzy.

5. Kacprzyk J., Wilbik A. Using Fuzzy Linguistic summaries for the comparison of time series // 2009 International Fuzzy Systems Association World Congress and 2009 European Society for Fuzzy Logic.
6. Pedrycz W., Chen S.M. Time Series Analysis, Modeling and Applications: A Computational Intelligence Perspective (e-book Google) // Intelligent Systems Reference Library. 2013. Vol. 47. P. 404.
7. Тимина И.А. Корректировка гипотез прогноза для извлечения знаний о временных рядах // Четырнадцатая национальная конференция по искусственному интеллекту с международным участием. КИИ-2014 Труды конференции. 2014. С. 68-76.
8. Ярушкина Н.Г., Тимина И.А. САПР на основе модели прогнозирования нечетких тенденций метрик // Прикладные информационные системы Сборник научных трудов. 2016. С. 191-199.
9. Ярушкина Н.Г., Тимина И.А. Автоматизированное проектирование на основе модели прогнозирования нечетких тенденций метрик // Пятнадцатая национальная конференция по искусственному интеллекту с международным участием Труды конференции: в 3 томах. Российская ассоциация искусственного интеллекта. 2016. С. 156-163.
10. FreeNAS 10 [Электронный ресурс] // Bug Tracking System: [сайт]. URL: <https://bugs.pcbssd.org/projects/freenas-10> (дата обращения: 30.09.2017).
11. FreeNAS 9 [Электронный ресурс] // Bug Tracking System: [сайт]. URL: <https://bugs.pcbssd.org/projects/freenas> (дата обращения: 30.09.2017).



## КОНСТРУКТОРСКОЕ ПОСТРОЕНИЕ 3D-МОДЕЛИ ИЗДЕЛИЯ В САД-СИСТЕМЕ

Д.Э. Цыганков<sup>64</sup>, А.Ф. Похилько<sup>65</sup>

Описан переход от оперирования абстрактными конструктивными элементами геометрии при формировании информационного 3D-образа изделия к реальным элементам конструкции. Построение таких объектов, несущих фиксированный смысл в предметной области, обеспечивается рассматриваемыми в настоящей работе семантическими макрооперациями.

### Введение

Развитие САХ-систем под эгидой методологии *Concurrent Engineering* (СЕ) упрочнило [1] положение 3D-моделей в жизненном цикле изделия (ЖЦИ) [2,3], прежде всего, на стадии опытно-конструкторских работ (ОКР) [4], вследствие чего, последние являются **отображением** изделия в плане его конструкции – важнейшей проектной информации на этапе технического проектирования. *Конструкция изделия*, определяемая конструкторской документацией (КД) согласно [5], отображается в САД-системах следующим образом:

$$САД : Констр(Изд) \rightarrow Мод_{Изд}^{3D}, \quad (1)$$

где  $Мод_{Изд}^{3D}$  и  $Констр(Изд)$  – это 3D-модель проектируемого изделия и его конструкция соответственно.

### 1. Базовая операция САД-системы

Конструктивный элемент геометрии (англ. *Features*), присутствующий во всех без исключения САПР, представляет собой составную часть 3D-модели изделия:

$$Мод_{Изд}^{3D} = \sum_{q=1}^m KЭ_q, \quad (2)$$

где  $KЭ_q$  – это  $q$ -ый (из  $n$ ) конструктивный элемент.

---

<sup>64</sup> 432027, Ульяновск, ул. Северный Венец, 32, УлГТУ, e-mail: d.tsyg@mail.ru

<sup>65</sup> 432027, Ульяновск, ул. Северный Венец, 32, УлГТУ, e-mail: paf\_54@mail.ru

Построение каждого КЭ обеспечивается т.н. «*базовыми операциями*» – простейшими действиями, определяемыми функционалом используемой САД-системы. Базовые операции (БО), выполнение которых в каждой САПР сильно различается, имеют семантически подобное содержание – «*построение конкретного конструктивного элемента (КЭ)*», т.е.:

$$CAD: BO_i < par_j^{BO_i} \rightarrow KЭ_q < attr_k^{KЭ_q} \rangle, \quad (3)$$

где  $par_j$  и  $attr_k$  –  $j$ -ый параметр  $i$ -ой БО и  $k$ -ый атрибут сформированного  $q$ -го КЭ соответственно.

Как видно из выражения (3), геометрические атрибуты формируемых конструктивных элементов определяются параметрами базовых операций. При этом для пользователя базовая операция является абстракцией, так как описывающие ее параметры актуальны исключительно внутри САД-системы. Последующее задание соответствий между конструктивными элементами 3D-модели и конструкцией изделия детально рассматривается в работе [6]; оно происходит «*в уме*» инженера, что исключает однозначно верное воспроизведение логики построения проектного решения для другого конструктора, – не являющегося автором проекта.

## 2. Семантическая макрооперация

В работе [7] впервые описан механизм обобщения базовых операций до уровня семантических единиц. Главной целью является повышение информативности процесса построения 3D-модели в САПР за счет перехода от абстрактных базовых операций к семантическим процедурам («*семантические макрооперации*»), несущим реальный конструкторский смысл в предметной области проектируемого изделия.

Семантическая операция (СМО) представлена на рисунке 1.

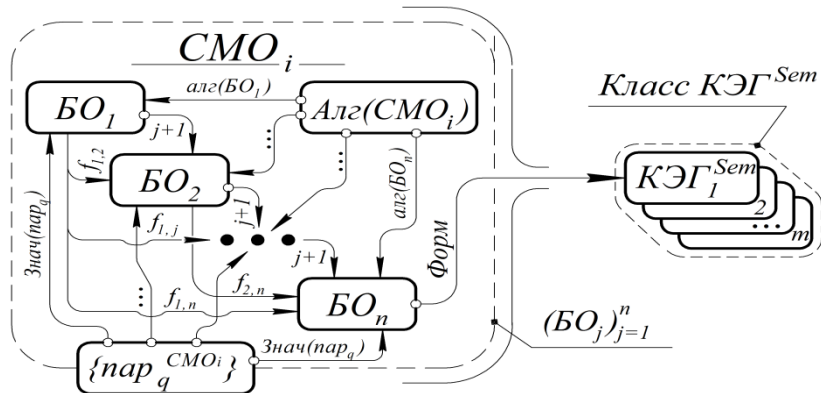


Рис. 1. Структура семантической макрооперации

Как показано на данном рисунке 1, СМО обеспечивает построение семантического конструктивного элемента, отличающегося тем, что он уже не является абстракцией, отображая реальные конструкторские элементы изделия как составные части 3D-модели.

Представляя собой композицию БО, СМО описывается как:

$$СМО_s < пар_j^{СМО_s} \succ \approx \bigcup_{i=1}^{n(s)} БО_i < пар_j^{БО_i} > \quad (4)$$

где символ объединения подразумевает обобщение всех используемых БО в рамках одной процессной единицы.

Ключевую роль в СМО играет алгоритм ее реализации. В зависимости от исходных данных, базовые операции выстраиваются в определенную последовательность, необходимую для формирования конструкторского решения, им удовлетворяющего.

Такая последовательность записывается в виде:

$$КЭ_q^{Sem} = БО_1 \cup БО_2 \cup \dots \cup БО_n \quad (5)$$

Но она обеспечивает формирование КЭ только как экземпляра класса. Критерием такого класса является **семантическое подобие** – основанное на конструктивно-функциональных свойствах реального устройства, а это, подразумевает и **структурную (топологическую) и геометрическую (параметрическую) вариативность** конструкторского решения.

Реализовать такие типы вариативность в стандартном функционале САД-системы не представляется возможным. Т.о., в рамках одной СМО обобщаются не только последовательности базовые операции, называемые далее «маршрут конструирования», но их различные комбинации (исполнения), отличающиеся как количеством БО, так и их составом, и, что очевидно, результирующими выходными данными.

Для обеспечения структурно-геометрической вариативности (но с сохранением смыслового содержания) в алгоритм СМО закладываются правила и условия. **Условия** обеспечивают корректность входных данных, не нарушающих конструкторские параметры в плане их заданного смысла в предметной области

**Правила** выполнения СМО имеют более сложную структуру; именно они обеспечивают древовидность СМО. Г.Б. Евгеньев в работе [8] называет такие правила выполнения терминами «*продукционные правила*» или модули знаний, они представляют собой условную связку «**ЕСЛИ – ТО**».

Тогда конструкторское решение формируется следующим образом:

$$КЭ_q^{Sem} = БО_1^0 \cup (БО_2^1 \cap \dots \cap БО_2^m) \cup \dots \cup (БО_n^1 \cap \dots \cap БО_n^m) \quad (6)$$

где  $m$  – номер конструкторского маршрута, определяющий формируемый КЭГ как экземпляр класса.

Множество таких условий позволяет реализовать в рамках одной СМО класс КЭ, имеющих общее конструктивно-функциональное назначение.

### 3. 3D-модель изделия как последовательность СМО

Сформированная 3D-модель, как отмечается в книге А.А. Варакина [9], полностью определяется деревом построения – структурой, содержащей в себе БО в порядке их выполнения. Дерево построения также называется «*историей построения*». Традиционное дерево построения предоставляет информацию лишь о том, как была построена данная 3D-геометрия, и внесение в нее изменений касается только геометрии.

3D-модель, формируемая последовательностью СМО имеет вид:

$$Mod_{IBo}^{3D} = \bigcup_{i=1}^n CMO_i < id, Имя, пар_j^{CMO_i} > , \quad (7)$$

так как каждая СМО характеризуется наименованием в виде «*построение q-го конструктивно-функционального элемента*», то их композиция в дереве построения обеспечивает предоставление подробной информации о конструкции проектируемого изделия.

Параметры СМО соответствуют атрибутам формируемого ею КЭГ, которые, в свою очередь, описывают конструктивно-функциональные характеристики реального устройства. То есть дальнейшая работа с 3D-моделью: редактирование, модификация и др., сводится к оперированию именно такими «смысловыми» параметрами, позволяя конструктору точно задавать и определять требуемую конфигурацию изделия.

Описанное представление 3D-модели относится к уровню «*Деталь*», при этом, атрибуты формируемых СМО «смысловых» конструктивных элементов доступны для работы при формировании 3D-модели уровня «*Сборочная единица*», что позволяет однозначно определять взаимное расположение деталей-компонент, сохраняя структурную целостность и конструкторскую грамотность проектируемого узла.

### Заключение

Предлагаемый авторами способ формирования и представления 3D-модели в САПР нацелен на повышение информативности последней с одновременным снижением трудоемкости ее построения и требуемых временных ресурсов. Дерево построения 3D-модели является важнейшей характеристикой, а следовательно, закладывание в него конструкторско-функциональной информации, ассоциативно связанной с 3D-геометрией, позволит значительно повысить ее информативность, предоставляя пользователю атрибуты с реальным конструкторским смыслом.

Следующим шагом является разработка библиотеки семантических макроопераций, предоставляющей функции создания, хранения, вызова, редактирования и удаления СМО. Если с точки зрения пользователя-конструктора такие СМО являются целостными (неделимыми) единицами, то в данной библиотеке будет выполняться определение их внутренней структуры как системы БО согласно рисунку 1.

*Исследование выполнено при поддержке РФФИ, грант №16-47-732138.*

### Список литературы

1. Tsygankov, D. et al. The Design Process Structural & Logical Representation in the Concurrent Engineering Infocommunication Environment, R. Curran et al. (eds.) Transdisciplinary Lifecycle Analysis of Systems – Proceedings of the 22nd ISPE Inc. International Conference on Concurrent Engineering, July 20-23, 2015, IOS Press, Amsterdam, 2015, pp. 595-602.
2. Ахтулов, А.Л. Задачи геометрического моделирования в создании систем автоматизации конструирования обводообразующих поверхностей сложных объектов / А.Л. Ахтулов, Л.Н. Ахтулова // Вестник Сибирской государственной автомобильно-дорожной академии. 2011. № 22. С. 43-47.
3. Калинин, В.И. Применение шаблонов Knowledge based Engineering в САПР CATIA V5 для моделирования сотовых панелей / В.И. Калинин, М.В. Лихачев // Решетневские чтения. 2015. Т. 2. № 19. С. 220-222.
4. Антипин, А.В. Интеграция САПР при конструировании электронной аппаратуры / А.В. Антипин, Е.Е. Носкова // Актуальные проблемы авиации и космонавтики. 2013. Т. 1. № 9. С. 192.
5. ГОСТ 2.102-2013. Единая система конструкторской документации (ЕСКД). Виды и комплектность конструкторских документов. Взамен ГОСТ 2.102-68. Введ. 2014-06-01. М. : Стандартинформ, 2014. II, 12 с.
6. Похилько, А.Ф. Отображение функциональной структуры проектируемого изделия в дереве построения его 3D-модели / А.Ф. Похилько, Д.Э. Цыганков // Известия Самарского научного центра Российской академии наук. 2017. Т. 19, № 1 (2). С. 424-427.
7. 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.
8. Евгеньев, Г.Б. Метод генерации 3D моделей в продукционных базах знаний / Г.Б. Евгеньев, А.А. Кокорев, М.В. Пиримяшкин // Известия высших учебных заведений. Машиностроение. 2015. № 4 (661). С. 38-48.
9. Варакин, А.А. Использование САПР SolidWorks в конструкторско-технологическом проектировании электронных средств : метод. указания. Ч. 1. Основы создания трехмерных моделей / А.А. Варакин. – Владимир : Изд-во Владим. гос. ун-та, 2009. 52 с.

## МОДЕЛИРОВАНИЕ ПРОЕКТИРОВАНИЯ ИЗДЕЛИЙ СО СЛОЖНОЙ ГЕОМЕТРИЕЙ ДЛЯ ИЗГОТОВЛЕНИЯ НА 3D ПРИНТЕРЕ

О.Э. Чоракаев<sup>66</sup>

Рассматриваются вопросы изготовления изделий методами 3D печати, производными аддитивной технологии производства изделий. Выделяются параметры, негативно влияющие на результирующий продукт и являющиеся базовыми при моделировании производственных процессов.

### Введение

В наши дни Additive Manufacturing-технологии используются повсеместно: научно-исследовательские организации с их помощью создают уникальные материалы и ткани, производственные компании используют 3D принтеры для ускорения прототипирования новой продукции, архитектурные и конструкторские бюро нашли в 3D печати нескончаемый строительный потенциал, в то время как дизайн-студии буквально вдохнули новую жизнь в дизайнерский бизнес благодаря аддитивным машинам.

Аналитики отрасли сходятся во мнении, что технологии послойного синтеза фактически оккупировали современную промышленность. Уже сегодня научно-исследовательские центры, занимающиеся аддитивными-разработками, получают огромные финансовые вливания от оборонного комплекса и медицинских государственных институтов.

### 1. Виды аддитивных технологий и их краткое описание

Существует множество технологий, которые можно назвать аддитивными, их объединяет принципиально новый подход к построению модели путем добавления материала в отличие от традиционных технологий, где создание детали происходит путем удаления избыточного материала.

---

<sup>66</sup> 432027, Ульяновск, ул. Северный Венец, 32, УлГТУ,  
e-mail: olegchorakaev@yandex.ru

Вне зависимости от конкретной технологии общую схему аддитивного производства можно изобразить в виде следующей последовательности (рис. 1).



Рис.1. Последовательность операций при аддитивном производстве.

Наиболее точной аддитивной технологией считается стереолитография (SLA), которая основывается на послойном отверждении жидкого фотополимера лазером. SLA принтеры используются преимущественно для изготовления прототипов, макетов и дизайнерских компонентов повышенной точности с высоким уровнем детализации. Точность изготовления моделей порядка 0.05-0.13 мм.

Усовершенствованной версией SLA печати стало селективное лазерное спекание (SLS). Эта технология позволяет в качестве чернил использовать порошкообразные материалы, такие как керамическая глина, металлический порошок, цемент и сложные полимеры.

Работу с металлическим материалом в лучшей степени поддерживает инновационная технология производства сложных изделий посредством лазерного наплавления металлического порошка (Selective laser melting - SLS). Технология требует мощных лазеров для плавления металлических порошков до образования гомогенной массы, из которой образуются трехмерные физические объекты. Данный процесс успешно заменяет традиционные методы производства, так как физико-механические свойства изделий, построенных по технологии SLM, зачастую превосходят свойства изделий, изготовленных по традиционным технологиям.

Установки лазерного наплавления позволяют получать изделия с физико-механическими параметрами, зачастую превосходящими свойства изделий, изготовленных по традиционным технологиям, что помогает решать сложные производственные задачи промышленных предприятий, работающих в авиакосмической, энергетической, машиностроительной и приборостроительной отраслях.

В особую группу выделяются недавно появившиеся MultiJet-, PolyJet-, InkJet-, ColorJet-аппараты, которые работают по классической аддитивной-технологии. Они оборудованы струйными печатными

головками, заправленными быстро-застывающим материалом (например, фотополимерный или восковой материал, гипс или пластик). На сегодняшний день подобные устройства нешироко распространены, однако не исключено, что уже через несколько лет трехмерная струйная печать станет столь же распространена, как и классические печатные устройства.

Наиболее массовыми и поэтому самыми широко распространенными остаются FDM-принтеры – устройства, создающие трехмерные объекты путем послойного наплавления филамента (расплавленной пластиковой нити), которые могут оснащаться одной или несколькими печатными головками, внутри которых находится нагревательный элемент с фильерами (экструдер), который расплавляет тонкую пластиковую нить (леску) и послойно укладывает ее согласно данным математической 3D-модели. Более подробно схема работы классического FDM-принтера раскрывается на рисунке 2.

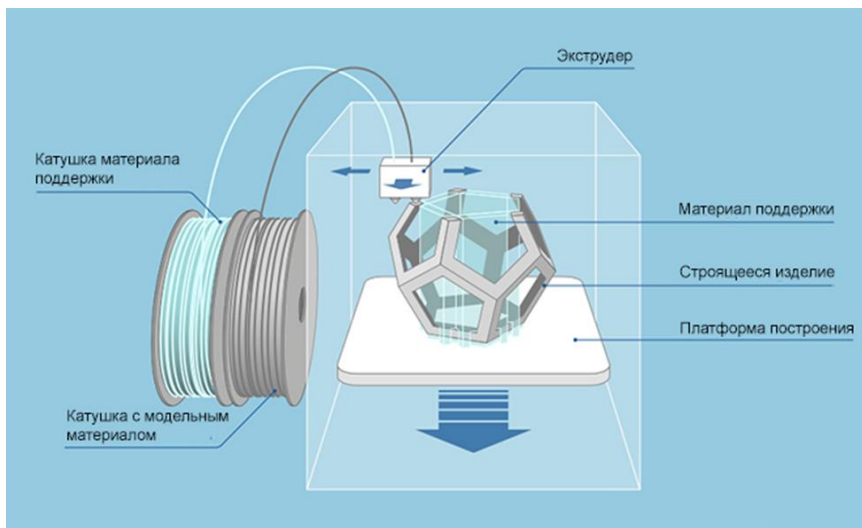


Рис.2. Схема исполнения FDM-принтера

## 2. Основные параметры 3D – печати и последствия нарушения технологии

Параметры, характеризующие процесс FDM –печати могут быть сгруппированы по следующим признакам:



- настройки принтера (размер рабочего объема по трем координата, количество экструдеров, диаметр сопла и т.д.);
- настройки прутка (диаметр, температура плавления, температура подогрева платформы, необходимое охлаждение и т.д.);
- настройки процесса печати, сведенные в таблицу 1.

Таблица 1. Параметры настройки печати

№	Наименование	Описание
1	Высота слоя	один из самых важных параметров, влияющих на качество печати. При его установке необходимо ориентироваться на геометрию печатаемой модели и время, за которое планируется напечатать готовое изделие. Чем меньше значение параметра высоты слоя, тем дольше идет печать. Высота слоя может варьироваться от 0.05 до 0.25 для сопла диаметром 0,3 мм. Главное правило при выставлении этого параметра: он не может превышать толщину используемого сопла (желательно ставить не больше $\frac{3}{4}$ от этой толщины).
2	Высота первого слоя	как правило, задается немного больше параметра Layer height для лучшего закрепления модели на столе.
3	Минимальное количество слоев контура модели	сначала экструдер очерчивает контур слоя. Затем заполняет его с определенной плотностью заполнения. Так вот, этот показатель равен количеству контуров, которое экструдер сделает перед тем, как начать заполнение. Чем больше этот параметр, тем прочнее внешняя поверхность детали. Но с увеличением слоев есть и паразитный эффект – скручивание (загиб) детали. Чем меньше число, тем меньше толщина стенки модели, а значит, модель будет более хрупкая.
4	Сплошные слои	количество слоев с верхней и нижней закрывающих плоскостей модели (рекомендуется 4 для низа и верха).

Основные дефекты трехмерной печати возникают при нарушении вышеописанных параметров, например приведенные ниже:

- Деформация - отклеивание первого слоя (причиной может быть неравномерный прогрев или неправильно подобранная температура подогрева, недостаточная адгезия со столом, неправильная калибровка или недостаточная площадь контакта модели);
- Смещение первого слоя - основа модели смещается из-за веса распечатки, который давит на первый слой, когда нижние слои еще не успели остыть (необходимо подобрать баланс температуры между нагревом сопла и внешним охлаждением, подобрать уровень платформы и подъема сопла, формирование наиболее плавных контуров детали);
- Смещение слоев в модели (причиной являются проблемы в механике принтера – натяжение ремней, жесткость конструкции)
- Отсутствующие слои (также зачастую проблема с механикой – изношенные подшипники, направляющие или неправильная смазка, проблемы с экструдером)

### 3. Язык G-кодов для управления 3D принтером и примеры калибровочных кодов

3D принтер является простым ЧПУ станком, который просто выполняет команды на специальном языке под названием G-код. В целом команды для управления аддитивными устройствами претерпели изменения, но общий стиль остался неизменным. Порядок команд в программе строго не оговаривается, но традиционно предполагается, что первыми указываются подготовительные команды, затем команды перемещения, затем выбора режимов обработки и технологические команды.

Рассмотрим специфичные команды для 3D – принтеров:

*Команды экструдера:*

- M82 - Установить экструдер в абсолютную систему координат
- M83 - Установить экструдер в относительную систему координат
- M104 - Ожидание нагрева экструдера до определенной температуры [M104 S190]
- M105 - Получить текущую температуру экструдера [M105 S2]
- M106 - Включение вентилятора обдува детали [M106 S127] - мощность 50%
- M107 - Выключение вентилятора обдува детали [M 107]
- M109 - Нагреть экструдер и удерживать температуру [M109 S215]

*Команды подогреваемого стола:*

- M140 - Установить температуру стола [M140 S65]
- M190 - Нагреть стол и удерживать температуру [M190 S60]

*Команды свойств филамента:*

- M200 - Задать диаметр филамента [M200 D1.128]
- M600 – Замена филамента

В качестве дополнительных подпрограмм приводятся подпрограммы калибровки высоты стола (листинг 1) и смены пластика при печати одним соплом (листинг 2).

```
G28 X0 Y0 Z0
G90
G1 X10 Y10 F1000
G4 S15
G1 X10 Y190
G4 S15
G1 X190 Y190
G4 S15
G1 X190 Y10
G4 S15
G1 X100 Y100
G4 S15
M300 S5000 P280
```

Листинг 1. Подпрограмма калибровки стола

Сопло перемещается по пяти точкам, ожидает там пятнадцать секунд, в течение которых есть возможность откалибровать регулятор высоты нагреваемой платформы.

```
G91
G1 Z5
G90
G1 Y10 X10 F1000
G4 S60
G91
G1 E100 F100
G92 E0
G90
```

Листинг 2. Подпрограмма смены филамента

Первоначально вычисляем высоту слоя, который желаем печатать другим цветом. Умножаем номер слоя на высоту слоя, получаем высоту в миллиметрах. Находим в G-коде строку с Z равной вычисленной высоте.

### Заключение

В работе рассмотрены основные технологии аддитивного производства, используемые в промышленности, и подробно рассмотрены параметры этого процесса и их влияние на конечный продукт. Данные параметры будут положены в основу процесса моделирования послойного изготовления, для концептуального экспериментирования до вывода изделия на принтер.

### Список литературы

1. Тарасова Т. В., Гвоздева Г. О., Тихонова Е. П. Перспективы использования лазерного излучения для поверхностной обработки цветных сплавов//Вестник МГТУ СТАНКИН. 2012. № 2. С. 140-143.
2. Мархадаев Б. Е. Новая аддитивная технология изготовления пластиковых и металлических объектов//Проблемы механики современных машин: мат-лы VI междунар. конф. Т. 2. -Улан-Удэ, 2015. -С. 216-221.
3. Григорьев С. Н., Смуров И. Ю. Перспективы развития инновационного аддитивного производства в России и за рубежом//Инновации. 2013. № 10. С. 2-8.
4. Шишковский И. В., Холпанов Л. П., Закиев С. Е. Моделирование и послойный синтез объемных изделий из нитрида титана методом селективного лазерного спекания. //Физика и химия обработки материалов, 2005, № 3, с. 71-78.
5. Смуров И. Ю., Ядройцев И. А., Мовчан И. А., Окунькова А. А., Черкасова Н. Ю., Антоненкова Г. В., Аддитивное производство с помощью лазера. Проведение экспериментальных работ. //Вестник МГТУ СТАНКИН. 2012. Т. 1. № 1. С. 36-38.

## МИКРОАНАЛИЗ ПРОИЗВОДИТЕЛЬНОСТИ СИСТЕМЫ ПРОВЕРКИ SQL-ЗАПРОСОВ

Г.И. Шайдуллова

В данной статье рассматривается подход к проведению экспериментального исследования производительности программы, автоматически тестирующей правильность sql-запросов. Сформулированы гипотезы о проблемах с производительностью, выявлены параметры и факторы, влияющие на производительность проверки.

### Введение

В настоящее время SQL является единственным механизмом связи между прикладным программным обеспечением и базой данных. В связи с распространением использованием баз данных, повышается необходимость в подготовке людей владеющий SQL. Освоить многообразие средств и возможностей можно только через множественное решение конкретных задач. Задача по созданию системы, в автоматическом режиме проверяющая данные решения, является, на сегодняшний день, актуальной.

В данной работе предлагается рассмотреть способ проведения исследования производительности программы, автоматически тестирующей правильность sql-запросов.

Пространство проблем с производительностью может быть широким в связи с широким пространством решений: параллелизм, приостановка или ограничение на параллелизм для уменьшения влияния на время выполнения одного запроса и так далее. В этой связи оценка производительности нас интересует не столько с позиции оптимизации, сколько с позиции выявления зависимостей от значимых параметров:

- Время выполнения запроса. Этот параметр необходим для определения ограничения time limit, которое имеет место в задании.
- Время выполнения всего теста, содержащего возможно довольно много проверок, причем для каждой проверки требуется новое содержимое базы данных.
- Общее время обработки всех заданий на тестирование из очереди заданий, которая образуется из решений многих участников турнира.

## Тестовая рабочая нагрузка

Для измерения времени выполнения запроса рассматривается тестовая рабочая нагрузка в зависимости от сложности выборки. Главным фактором, влияющим на время выполнение запроса, является уровень сложности запроса. Выявить данный фактор возможно через оценку сложности алгоритма выполнения исследуемых запросов. Поскольку анализ проходил средствами СУБД PostgreSQL есть возможность использовать оператор EXPLAIN ANALYZE <sql-request>, возвращающий детальную информацию о работающих алгоритмах при выполнении данного запроса.

В таблице 1 зафиксированы полученные по функции анализа алгоритмы для выбранного набора тестовых запросов к общедоступной базе данных «geosample» сервера «gis-lab.info» о геоданных Российской Федерации[1].

Где T – время для «не разогретой» системы; n – глубина дерева, прохождение по которому занимает для данной таблицы 43 ms (определяется по поиску несуществующих объектов).

Таблица 1. Оценка сложности выполнения запросов

№	Формулировка задания	Эталонный SQL-запрос	Выполнение EXPLAIN ANALYZE	Оценка
1 5	Определить район с населенным пунктом Комсомольск	select rayon from settlements where name = 'Комсомольск';	<b>QUERY PLAN</b> text Seq Scan on settlements (cost=0.00..28.56 r Filter: ((name)::text = 'Комсомольск')::t Total runtime: 0.237 ms	O(T+log(n))
1 6	Вывести ширину и долготу центра населенного пункта Боярка	select lat, long from settlements where name = 'Боярка';	<b>QUERY PLAN</b> text Seq Scan on settlements (cost=0.00..28.56 r Filter: ((name)::text = 'Боярка')::text Total runtime: 0.187 ms	O(T+log(n))
1 2	Вывести список населенных пунктов типа Город.	select * from settlements where type = 'r';	<b>QUERY PLAN</b> text Seq Scan on settlements (cost=0.00..28.56 Filter: ((type)::text = 'r')::text Total runtime: 0.297 ms	O(T+log(n))

Таблица 1. Продолжение

№	Формулировка задания	Эталонный SQL-запрос	Выполнение EXPLAIN ANALYZE	Оценка
13	Вывести населенные пункты с долгой центра от 76 до 86. Названия субъектов должны быть отсортированы в порядке убывания	select * from settlements where long between 76 and 86 order by subject desc;	<pre> <b>QUERY PLAN</b> text Sort (cost=41.87..42.62 rows=301 width=1522) Sort Key: subject Sort Method: quicksort Memory: 223kB -&gt; Seq Scan on settlements (cost=0.00..29. Filter: ((long &gt;= 76)::double precision) Total runtime: 2.222 ms </pre>	$O(T+n^2 \log(n))$
17	Вывести 30 населенных пунктов, начиная с 5 записи, кроме населенных пунктов района «Новосибирская». Данные сортируются по широте центра в убывающем порядке.	select * from settlements where rayon != 'Новосибирская' order by lat desc offset 5 limit 30;	<pre> <b>QUERY PLAN</b> text Limit (cost=39.49..39.56 rows=30 width=1522) (actual rows=30) -&gt; Sort (cost=39.47..40.36 rows=356 width=1522) Sort Key: lat Sort Method: top-N heapsort Memory: 49kB -&gt; Seq Scan on settlements (cost=0.00..29. Filter: ((rayon)::text &lt;&gt; 'Новосибирская')) Total runtime: 1.545 ms </pre>	$O(T+n \log^2(n))$

В таблице 2 представлены измерения времени выполнения исследуемых запросов.

Таблица 2. Измерения времени выполнения запросов

№ в каталоге	Время выполнения запроса, ms. (среднее из трех попыток)
15	70
16	73
17	176
12	261
13	533

В ходе исследований, по собранным данным прослеживается зависимость уровня сложности запроса от количества возвращаемых записей. Время увеличивается относительно количества выводимых записей и применения в запросе операторов обработки данными, такие как сортировка и ограничения определенных данных по экспоненциальной зависимости, которую можно построить на основе измерений и представить в виде уравнения. Если переходить на числовое представление, по итогам нашего исследования получаем уравнение вида:

$$y = 33,457 e^{0,5334x} \quad (1)$$

### **Заключение**

Исследование позволило выявить зависимость сложности запроса от количества возвращаемых записей. Предполагается использовать данную зависимость при генерации заданий для турнира по проверки правильности sql-запросов. При создании задания необходимо учитывать сложность алгоритмов, которые будут использоваться для выполнения запроса, и выставлять time limit задания с помощью выявленной зависимости.

### **Список литературы**

1. Доступ к ресурсу: <http://gis-lab.info/qa/geosample.html>

## СТАТИСТИЧЕСКИЙ КОНТРОЛЬ МНОГОПАРАМЕТРИЧЕСКОГО ПРОЦЕССА ПРИ МАЛОМ ОБЪЕМЕ ДАННЫХ

К.С. Ширкунова<sup>67</sup>

При производстве печатных плат в сточных водах имеют место органические и неорганические загрязняющие примеси, что наносит серьезный ущерб окружающей среде из-за наличия тяжелых металлов. При этом, контроль загрязненности сточных вод является дорогостоящим и требует дополнительного обучения персонала. Рассмотрены особенности статистического контроля стабильности уровня загрязнения в условиях, когда количество наблюдений мало.

### Введение

Основными источниками загрязнения водоемов являются недостаточно очищенные сточные воды предприятий. Производство печатных плат наносит серьезный ущерб окружающей среде из-за наличия тяжелых металлов[1].

Имеются данные по химическому составу сточных вод при производстве печатных плат на предприятии с частотой снятия показаний один раз в 4-7 дней. Контролируется стабильность процесса очистки сточных вод. Для решения задачи могут быть использованы методы статистического контроля процессов [2,3]. В этом случае необходимый для исследования объем выборки получается только при поквартальном анализе (15-20 индивидуальных наблюдений за квартал).

Часть контролируемых показателей независимы, а некоторые из них коррелированы между собой. Для показателей первой группы в принципе могут быть использованы стандартные контрольные карты Шухарта[3,4], однако при использовании индивидуальных наблюдений необходимо проверить нормальность их распределения. Учитывая малость выборок, используется критерий Шапиро-Уилка. Для коррелированных показателей применяют многомерные методы, но

---

<sup>67</sup> 432027, Ульяновск, ул. Северный Венец, 32, УлГТУ, e-mail: kristinashirkunova19@mail.ru



проблема нарушения нормальности сохраняется, и возникает необходимость проведения преобразований данных[5].

## 1. Контроль независимых показателей

Для независимых показателей (при незначимости корреляций) в случае, когда критерий Шапиро-Уилка подтвердил нормальность, используют стандартные карты Шухарта по индивидуальным наблюдениям и скользящим размахам. Такие карты показаны на рис. 1 для показателя аммиак. Карты свидетельствуют о стабильности процесса: на них нет точек, выходящих за контрольные границы.

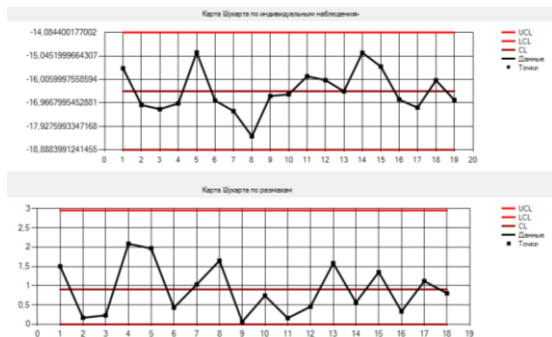


Рис. 1. Карты Шухарта для индивидуальных наблюдений и скользящих размахов

Если же гипотеза о нормальности не подтверждается, можно попытаться нормализовать данные с использованием преобразования Джонсона. Однако такой подход не всегда приводит к нужным результатам. Иногда целесообразно воспользоваться картами на основе распределения Грама-Шарлье [6-7].

Первые четыре слагаемых ряда Грама-Шарлье имеют вид:

$$\varphi(z) = f(z) - \frac{\mu_3}{3!\sigma^3} f^{(3)}(z) + \frac{\mu_4}{4!\sigma^4} f^{(4)}(z) \quad (1)$$

где  $f^{(r)}(z)$  –  $r$ -ая производная функции плотности стандартного нормального распределения,  $\mu_i$  –  $i$ -ый центральный момент выборки. Используя разложения производных в ряд Маклорена, после преобразования получим формулу

$$\varphi(z) = f(z) \left[ 1 - \frac{\mu_3}{3!\sigma^3} (3z - z^3) + \frac{\mu_4}{4!\sigma^4} (3 - 6z^2 + z^4) \right] \quad (2)$$

определяющую значения, откладываемые на карте.

Контрольные границы для карт на основе распределения Грама-Шарлье при заданном уровне значимости  $\alpha$  определяются как квантили соответствующего распределения порядка  $\alpha/2$  и  $1 - \alpha/2$ . Они вычисляются с использованием формулы:

$$F(z) = \frac{1}{\sqrt{2\pi}} \left( z - \frac{z^3}{6} + \frac{z^5}{40} - \frac{z^7}{336} + \frac{z^9}{3456} + \right. \\ \left. + \left[ 1 - \frac{z^2}{2} + \frac{z^4}{8} - \frac{z^6}{48} + \frac{z^8}{384} \right] \cdot \left[ -\frac{\mu_3}{3!\sigma^3} (z^2 - 1) + \frac{\mu_4}{4!\sigma^4} z(3 - z) \right] \right) \quad (3)$$



Рис.2. Карта индивидуальных наблюдений на основе распределения Грама-Шарлье

В качестве примера на рис. 2 показана карта на основе распределения Грама-Шарлье для показателя рН (кислотно-щелочной баланс). Карта показывает наличие нарушений стабильности процесса в 2, 7-10, 12 и 20 точках.

## 2. Контроль коррелированных показателей

В рассматриваемой ситуации коррелированными оказались две пары показателей: нитриты и фосфаты, железо и медь. Для контроля коррелированных показателей используются специальные многомерные методы [6-8]. Контроль среднего уровня процесса при условии соблюдения нормальности распределения возможен с помощью карт Хотеллинга по индивидуальным наблюдениям.

В случае нарушения нормальности возможен такой подход. Проводится преобразование данных к некоррелированным главным компонентам [9], и для каждой главной компоненты строятся карты на основе распределения Грама-Шарлье.

Для контроля многомерного рассеяния применяются карты обобщенной дисперсии, построение которых возможно только по мгновенным выборкам. Используя выборки объемом по три наблюдений (это

допустимо, учитывая, что коррелированы между собой по два показателя) получим за квартал 5-7 выборки, что явно недостаточно. На рис. 3 в качестве примера показана карта обобщенной дисперсии за 3 квартала для пары показателей нитриты и фосфаты. Карта говорит о стабильности данных показателей в рассматриваемой группе кварталов.

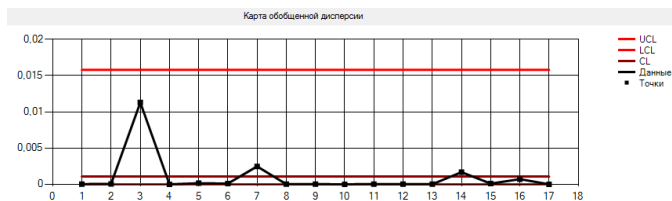


Рис.3. Карта обобщенной дисперсии

## Список литературы

1. Алексеев М.И., Рублевская О.Н. Очистка сточных вод. – СПб.: Новый журнал, 2013. – 496 с.
2. Клячкин В.Н. Статистические методы в управлении качеством: компьютерные технологии – М.: Финансы и статистика, ИНФРА-М, 2009. – 304 с.
3. Уилер, Д. Статистическое управление процессами/ Оптимизация бизнеса с использованием контрольных карт Шухарта; пер. с англ. / Д. Уилер, Д. Чамберс. – М. : Альпина Бизнес Букс, 2009. – 409 с.
4. Ширкунова К.С. Мониторинг стабильности химического состава сточных вод при производстве печатных плат / К.С. Ширкунова // IN MEMORIAM: Султан Галимзянович Валеев: сборник памяти С. Г. Валеева. – Ульяновск, 2016. – С. 151-158.
5. Ширкунова К.С. Многомерный статистический контроль процесса при нарушении нормальности распределения показателей. / Прикладная математика и информатика: современные исследования в области естественных и технических наук– Тольятти, 2017. – С.664-666.
6. Клячкин В.Н., Михеев А.Ю. Идентификация режима статистического контроля многопараметрического технологического процесса. Автоматизация. Современные технологии. 2011. № 12.– С. 27-31.
7. Клячкин В.Н. Статистический контроль технологического рассеяния в многопараметрическом процессе / В.Н. Клячкин, Т.И. Святова // Автоматизация. Современные технологии. – 2013. – № 12. – С. 22-25
8. Кувайскова Ю.Е., Клячкин В.Н., Бубырь Д.С. Прогнозирование состояния технического объекта на основе мониторинга его параметров / XII Всероссийское совещание по проблемам управления: Труды. М.: Институт проблем управления им. В.А. Трапезникова РАН. – 2014.– С.7616-7626.
9. Клячкин В.Н., Кувайскова Ю.Е., Алексеева В.А. Статистические методы анализа данных. – М. : Финансы и статистика, 2016. – 240 с.

Научное электронное издание

**ИНФОРМАТИКА, МОДЕЛИРОВАНИЕ,  
АВТОМАТИЗАЦИЯ ПРОЕКТИРОВАНИЯ**  
(ИМАП-2017)

IX Всероссийская школа-семинар  
аспирантов, студентов и молодых ученых  
(Россия, г. Ульяновск, 24-25 октября 2017 г.)

Сборник научных трудов

Под редакцией А.Н. Афанасьева

ЛР № 020640 от 22.10.97.

ЭИ № 1028. Объем 4,5 Мб.

Подписано в печать 01.12.2017. Заказ №878.

Ульяновский государственный технический университет  
432027, г. Ульяновск, ул. Сев. Венец, 32.  
ИПК «Венец» УлГТУ, 432027, г. Ульяновск, ул. Сев. Венец, 32.  
Тел.: (8422) 778-113  
E-mail: [venec@ulstu.ru](mailto:venec@ulstu.ru)  
[venec.ulstu.ru](http://venec.ulstu.ru)