

Минобрнауки Российской Федерации

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧЕРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ Н.Г.ЧЕРНЫШЕВСКОГО»

Кафедра компьютерной физики и метаматериалов на базе Саратовского
филиала Института радиотехники и электроники им. В.А. Котельникова РАН

**Средства автоматизации тестирования программного обеспечения
в задачах компьютерного моделирования**

**АВТОРЕФЕРАТ
ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ БАКАЛАВРСКОЙ РАБОТЫ**

Студента 4 курса 432 группы
направления 03.03.02 «Физика»
физического факультета

Колесова Артура Андреевича

Научный руководитель
доцент, к.ф.-м.н.

Ремизов А.С.

Зав. кафедрой:
профессор, д. ф-м.н

Аникин В.М.

Саратов 2019 г.

ОБЩАЯ ХАРАКТЕРИСТИКА РАБОТЫ

Актуальность. В настоящее время в сфере разработки программного обеспечения существует множество способов автоматизации на разных уровнях. В данной работе представлены способы применения автоматизированного тестирования в задачах компьютерного моделирования на различных этапах.

Для демонстрации были взяты 2 модели и к ним были написаны сценарии для автоматизированного тестирования. Первой такой моделью был клеточный автомат Конуэя. Для него было написано 3 теста, которые выполняют проверку корректности заполнения решетки, правил локальных взаимодействий, работы граничных условий.

В качестве второй модели был взят классификатор текстов на базе наивного классификатора Байеса. Для него было написано 4 теста, которые проверяют точность классификации текста не менее 90%, корректность работы с набором стоп-слов, правильность работы нормализатора, правильность работы токенизатора.

Цель работы заключалась в том, чтобы дать исследователю представление об устоявшихся технологиях тестирования и автоматизации в сфере разработки программного обеспечения, и предложить способы их применения в контексте реализации и развития компьютерных моделей каких-либо процессов, явлений или объектов.

Структура и объем работы. Выпускная квалификационная работа состоит из введения, основной части, состоящей из 4 глав, заключения, списка используемых источников и 2 приложений. Всего в работе 43 страницы.

КРАТКОЕ СОДЕРЖАНИЕ РАБОТЫ

Во введении описана практическая значимость и сформированы цели работы.

В первой главе представлено подробное описание видов, уровней и классификации тестирования программного обеспечения. На рисунке 1 представлена схема работы мутационного тестирования.

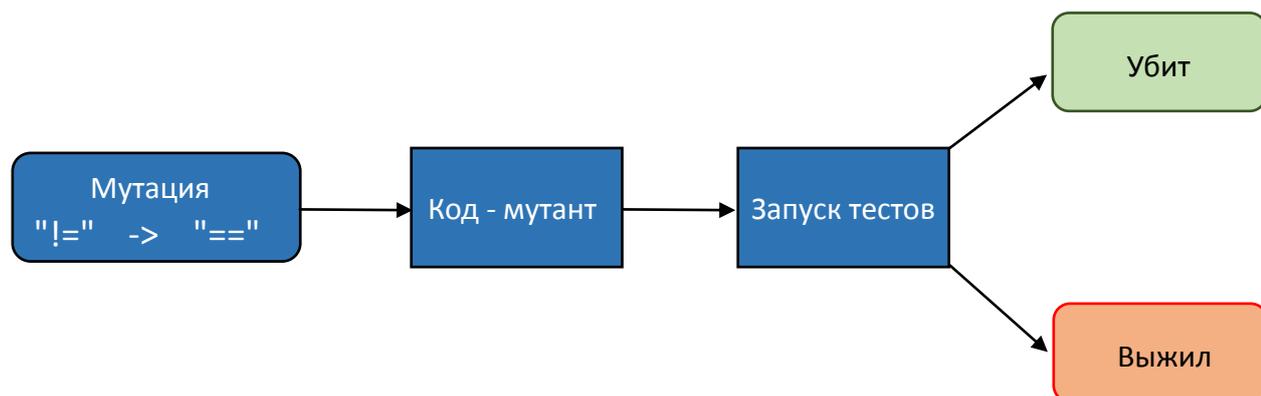


Рисунок 1 – Схема мутационного тестирования

Во второй главе говорится о автоматизированном тестировании и технологии непрерывной интеграции. Приведены преимущества автоматизации:

- Проверка состояния
- Быстрый отчёт
- Экономия человеческого ресурса
- Возможность разработчикам внести свой вклад в тесты

Также предоставлен список средств для автоматизации тестирования и интеграции.

Инструменты для CI/CD:

Локальные: GitLab CI, TeamCity, Bamboo, GoCD Jenkins, Circle CI.

Облачные: BitBucket Pipelines, Heroku CI, Travis, Codeship, Buddy CI, AWS CodeBuild.

Средства для проведения тестирования:

Python – Unittest, PyTest

C++ – GTest/Gmock

PHP – phpUnit, Codeception

Java – junit

В третьей главе рассказывается о компьютерном моделировании, необходимости автоматизации и вариантах применения автоматизации в задачах компьютерного моделирования.

Этап программной реализации и верификации разработанной ранее математической модели. Что возможно протестировать на этом этапе:

- консистентность и корректность исходных данных,
- модульные тесты для вычислительных алгоритмов,
- проверка граничных условий,
- проверка различного вида нормировок и интегральных характеристик

Этап постановки эксперимента и анализа полученных результатов.

Потенциальные сценарии для тестирования и автоматизации:

- автоматическая проверка корректности модели для шаблонных исходных данных, результат для которых известен из теоретических изысканий или математических оценок, тест задает начальные данные и параметры, запускает эксперимент, сверяет полученные результаты с ожидаемыми (это позволит контролировать корректность модели при последующих доработках)
- автоматическая подготовка исходных данных, пишется параметризованный генератор, а затем запускается тот или иной тест конкретного алгоритма или всей модели
- если модель использует внешние ресурсы (базы данных, различные программные API), осуществляется автоматическая проверка их доступности

- если модель использует аппаратуру, стоит протестировать ее доступность и готовность к обмену данными (датчики, радиометки, видекамеры и т.п.)

В четвертой главе приведены примеры автоматизированного тестирования для моделей клеточного автомата Конуэя и классификатора текстов. На рисунке 2 продемонстрирован внешний вид программной модели клеточного автомата Конуэя.

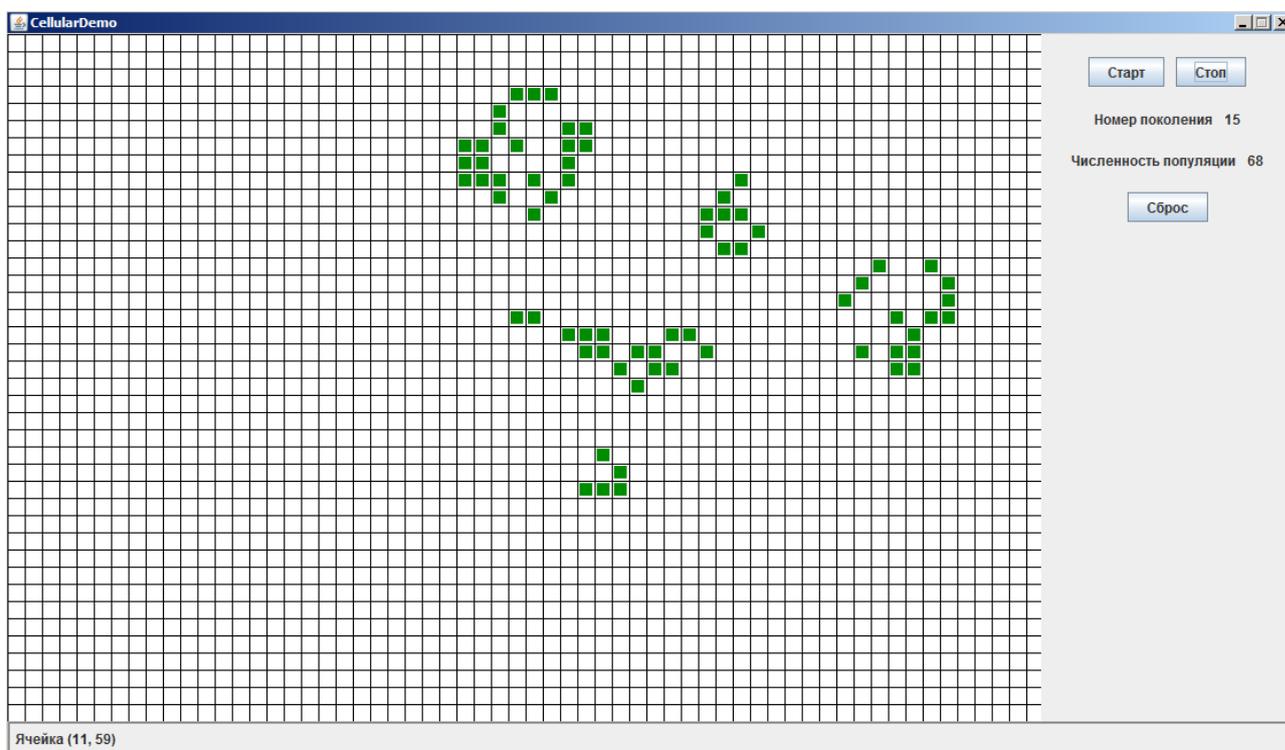


Рисунок 2 - вид программной модели клеточного автомата Конуэя

Для этой модели в ходе работы было написано 3 теста, которые проверяют следующее:

- корректность заполнения решетки,
- проверку правил локальных взаимодействий,
- корректность работы граничных условий

На рисунке 3 представлена схема основных компонент сервиса оценок.

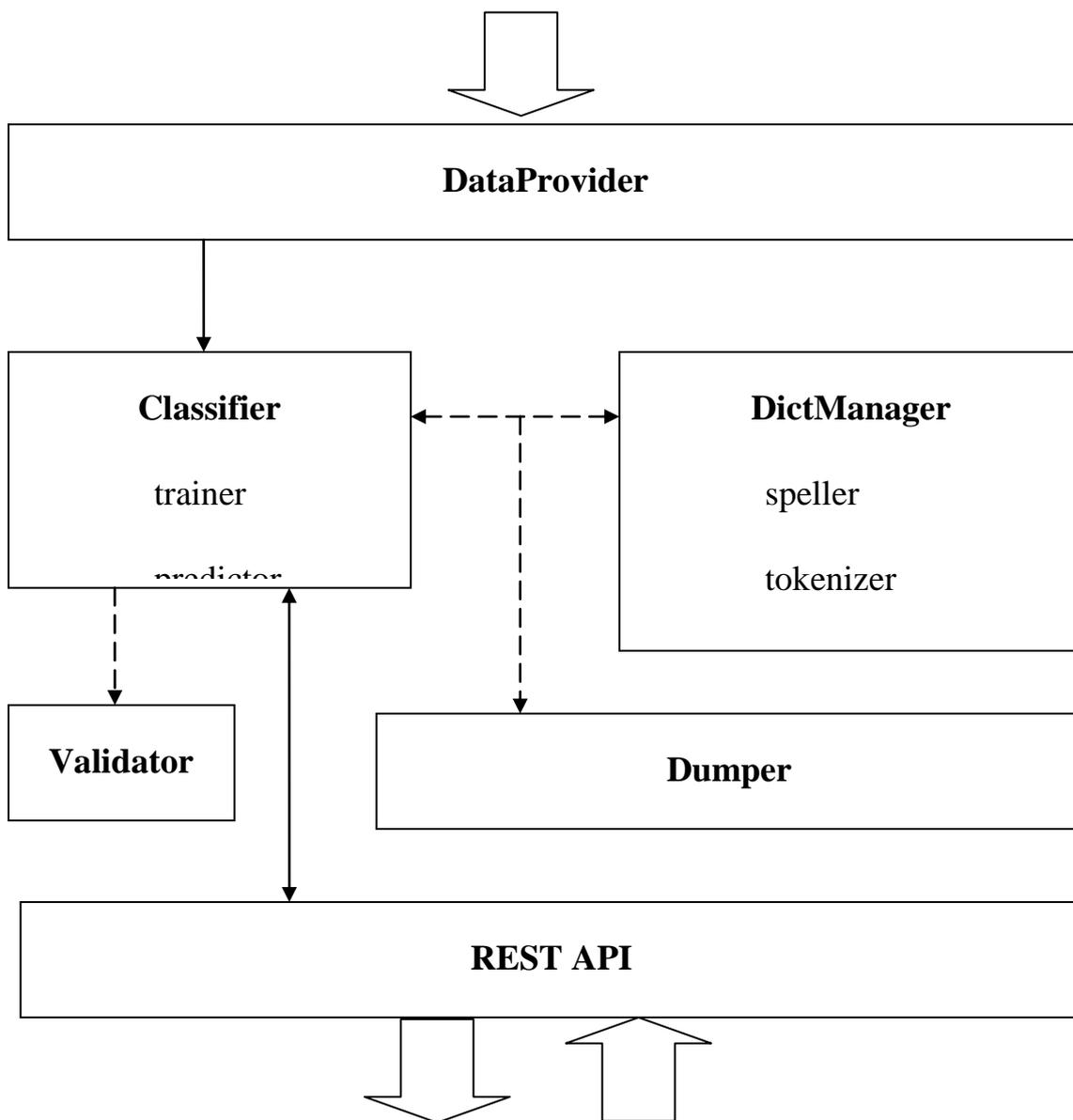


Рисунок 3 - схема основных компонент сервиса оценок

Validator - модуль валидации обученных моделей

DictManager - модуль для работы со словарями, именно он реализует нормализацию и токенизацию текста, проверку орфографии, экспорт/импорт сформированных словарей, векторизацию текста

Dumper - модуль для экспорта/импорта сформированных словарей и обученных моделей

REST API - модуль для общения с внешней средой, откуда поступают запросы на классификацию и куда возвращаются ответы, находится на стадии реализации

Для модели классификатора тестов было написано 4 теста, которые гарантируют следующее:

- проверку точности классификации не ниже 90% для предустановленной обученной модели, проверка проводится по несмещенной оценке,
- корректность работы с набором стоп-слов, проверяется их использование и отсутствие в итоговом словаре
- правильность работы нормализатора, который приводит слова к нормальной форме (ед. число им.падеж)
- правильность работы токенизатора, который нарезает предложения на токены для дальнейшей обработки

В **заключении** приводится перечень основных выводов, полученных в результате проведенной работы, и кратко суммируются основные преимущества автоматизации в задачах компьютерного моделирования.

В **приложениях** приведен код сценария автоматизированных тестов для обеих моделей.

ВЫВОДЫ

В работе сделана попытка применить опыт автоматизации и тестирования приложений к процессу разработки компьютерных моделей тех или иных процессов, явлений или объектов. Поскольку компьютерную модель можно рассматривать как разновидность специфического программного обеспечения, технически возможно применить устоявшийся стек технологий автоматизации.

Наибольший эффект от внедрения предлагаемых решений будет заметен в ситуации, когда компьютерную модель разрабатывает коллектив специалистов.

Применение предлагаемых решений гарантирует ожидаемое поведение модели в процессе ее доработки и развития, позволяет автоматизировать контроль возникновения ошибок и отклонений в работе, а также ряд других аспектов, что может быть полезно и актуально у разработчиков в этой сфере.

В практической части приведен пример автоматизации тестирования для модели классификатора текстов и модели клеточного автомата.

Список использованных источников

1. *Старолетов С.М.* Основы тестирования и верификации программного обеспечения. // Под ред.С.В. Макарова. М.: Лань, 2018.

2. *Джесз Х., Дейвид Ф.* Непрерывное развертывание ПО. Автоматизация процессов сборки, тестирования и внедрения новых технологий // Под ред. Гинзбург В.Р., перевод Сысолюк А.Г. М.: Вильямс, 2016.

3. *Антипов В.А., Бубнов А.А., Пылькин А.Н.* Введение в программную инженерию // ИНФРА-М 2019.

4. *Поль М. Дюваль, Стивен М., Эндрю Г.* Непрерывная интеграция. Улучшение качества программного обеспечения и снижение риска. // Перевод Коваленко В.А. М.: Вильямс, 2016.

5. *Заводинский В.Г.* Компьютерное моделирование наночастиц и наносистем. // Под ред. Игнатовой В.Р. М.: Физматлит, 2013.

6. *Подколзин А.С.* Компьютерное моделирование логических процессов. // Под ред. Легостаевой И.Л М.: Физматлит, 2008.

7. *Прудников В.В., Вакилов А.Н., Прудников П.В.* Фазовые переходы и методы их компьютерного моделирования // Под ред. Чернявского К.С. М.: Физматлит, 2009.

8. *Авдеев В.А.* Компьютерное моделирование цифровых устройств. // Под ред. Мовчан Д.А. М.: ДМК-Пресс, 2012.
9. *Себастьян Р., Вахид М.* Python и машинное обучение. М.: Вильямс, 2019.
10. *Мюллер А.П., Гвидо С.* Введение в машинное обучение с помощью Python // Перевод Груздев А. М.: Вильямс, 2017.
11. *Тоффоли Т., Марголюс Н.* Машины клеточных автоматов. М.: «Мир», 1991.
12. *Toffoli T.* Cellular automata machines. // The MIT Press; UK ed. edition, 1987.
13. *Bill L.* A friendly introduction to software testing. // Amazon Digital Services LLC, 2016.
14. *Gerald M. Weinberg.* Perfect software and other illusions about testing. // Dorset House, 2008.
15. *Седжвик Р., Уэйн К.* Computer science. Основы программирования на Java, ООП, алгоритмы и структуры данных. // Под ред. Римицан Н., перевод Матвеева Е., М.: Питер, 2018.
16. *Кузнецов М., Симдянов И.* Объектно-ориентированное программирование на PHP. // Под ред. Кондуковой Е. М.: БХВ-Петербург, 2007.
17. Автоматизация тестирования с нуля [Электронный ресурс] URL: <https://habr.com/ru/post/426487/>
18. Компьютерное моделирование физических процессов с многомерным временем [Электронный ресурс] URL: https://enotabene.ru/kp/article_20061.html

19. Тестирование программного обеспечения – основные понятия и определения [Электронный ресурс] URL: <http://www.protesting.ru/testing/>

20. Мутационное тестирование [Электронный ресурс] URL: <https://habr.com/ru/post/334394/>