

Программная инженерия



Пр
ИН
7
2022
Том 13

MONETEC 2022



The International Science and Technology Conference
«Modern Network Technologies, MoNeTec -2022»

CALL FOR PAPERS

<https://monetec.ru/>

October 27-29, 2022

Moscow, Russia

CONFERENCE

The conference gathers specialists of the international scientific community, research departments of corporations, start-ups, representatives of industry and business, development institutions, and public authorities to discuss promising and relevant technologies in the field of computer networks, virtualization of network resources, and cloud computing.

TOPICS

- Data Communication Infrastructure
- QoS Control
- Cloud Computing
- Optimization Methods, Tools, and Technologies in Cloud Computing
- Network Function Virtualization and Services
- Edge Computing
- 5G & 6G Wireless Technologies, Applications and Services
- Future Networking

PAPER SUBMISSION

The following paper categories are welcome:

- **Full papers** (from 6 to 12 pages) must describe original and unpublished research results.
- **Short papers** (up to 6 pages) must describe original and unpublished work-in-progress.
- **Posters** must describe the student work or work-in-progress.

Accepted and presented papers will be published in the MoNeTec-2022 Conference Proceedings and IEEE Xplore® library. All accepted papers written in Russian will be published in the Proceedings volume indexed in RSCI (Russian Science Citation Index).



ОТДЕЛЕНИЕ
МАТЕМАТИЧЕСКИХ
НАУК



IMPORTANT DATES

- **June 23, 2022**
Abstract Submission
- **July 15, 2022**
Paper Submission
- **August 01, 2022**
Preliminary Acceptance Notification
- **September 01, 2022**
Acceptance Notification
- **September 15, 2022**
Camera-ready Version
- **October 01, 2022**
Ready-to-print Poster Submission



GENERAL SPONSOR



SPONSORS



ORGANIZERS



SUPPORTED BY



Программная инженерия

Том 13
№ 7
2022
пр
ин

Учредитель: Издательство "НОВЫЕ ТЕХНОЛОГИИ"

Издается с сентября 2010 г.

DOI 10.17587/issn.2220-3397

ISSN 2220-3397

Редакционный совет

Садовничий В.А., акад. РАН
(председатель)
Бетелин В.Б., акад. РАН
Васильев В.Н., чл.-корр. РАН
Макаров В.Л., акад. РАН
Панченко В.Я., акад. РАН
Стемпковский А.Л., акад. РАН
Ухлинов Л.М., д.т.н.
Федоров И.Б., акад. РАН
Четверушкин Б.Н., акад. РАН

Главный редактор

Васенин В.А., д.ф.-м.н., проф.

Редколлегия

Антонов Б.И.
Афонин С.А., к.ф.-м.н.
Бурдонов И.Б., д.ф.-м.н., проф.
Борзов Ю., проф. (Латвия)
Гаврилов А.В., к.т.н.
Галатенко А.В., к.ф.-м.н.
Корнеев В.В., д.т.н., проф.
Костюхин К.А., к.ф.-м.н.
Махортов С.Д., д.ф.-м.н., доц.
Манцивода А.В., д.ф.-м.н., доц.
Назиров Р.Р., д.т.н., проф.
Нечаев В.В., д.т.н., проф.
Новиков Б.А., д.ф.-м.н., проф.
Павлов В.Л. (США)
Пальчунов Д.Е., д.ф.-м.н., доц.
Петренко А.К., д.ф.-м.н., проф.
Позднеев Б.М., д.т.н., проф.
Позин Б.А., д.т.н., проф.
Серебряков В.А., д.ф.-м.н., проф.
Сорокин А.В., к.т.н., доц.
Терехов А.Н., д.ф.-м.н., проф.
Филимонов Н.Б., д.т.н., проф.
Шапченко К.А., к.ф.-м.н.
Шундеев А.С., к.ф.-м.н.
Щур Л.Н., д.ф.-м.н., проф.
Язов Ю.К., д.т.н., проф.
Якобсон И., проф. (Швейцария)

Редакция

Чугунова А.В.

Журнал издается при поддержке Отделения математических наук РАН,
Отделения нанотехнологий и информационных технологий РАН,
МГУ имени М.В. Ломоносова, МГТУ имени Н.Э. Баумана

СОДЕРЖАНИЕ

Змеев Д. О., Змеев О. А., Иванова Л. С. Практика работы с антипаттернами для Essence Practice Library	311
Gaydamaka K. I., Ognyanovich P. A. Applying Supervised Machine Learning Methods to Determine the Atomicity of Requirements for Complex Technical Systems	322
Alexandrov D. V. Overview of Face Recognition Algorithms for Person Identification	331
Ступина М. В. Архитектура программной системы поддержки учета посещаемости в вузе	344
Ермилов Е. Е., Постный А. В. Персонализация учебного плана посредством применения алгоритмов обучения с подкреплением	354

Журнал зарегистрирован
в Федеральной службе
по надзору в сфере связи,
информационных технологий
и массовых коммуникаций.
Свидетельство о регистрации

ПИ № ФС77-38590 от 24 декабря 2009 г.

Журнал распространяется по подписке, которую можно оформить в подписных агентствах (индекс по Объединенному каталогу "Пресса России" — 22765) или непосредственно в редакции (для юридических лиц).
Tel.: (499) 270-16-52.

[Http://novtex.ru/prin/rus](http://novtex.ru/prin/rus) E-mail: prin@novtex.ru

Журнал включен в Российский индекс научного цитирования (РИНЦ) и Russian Science Citation Index (RSCI).

Журнал входит в Перечень научных журналов, в которых по рекомендации ВАК РФ должны быть опубликованы научные результаты диссертаций на соискание ученой степени доктора и кандидата наук.

© Издательство "Новые технологии", "Программная инженерия", 2022

SOFTWARE ENGINEERING

PROGRAMMAYA INGENERIA



Published since September 2010

DOI 10.17587/issn.2220-3397

ISSN 2220-3397

Editorial Council:

SADOVNICHY V. A., Dr. Sci. (Phys.-Math.),
Acad. RAS (*Head*)
BETELIN V. B., Dr. Sci. (Phys.-Math.), Acad. RAS
VASIL'EV V. N., Dr. Sci. (Tech.), Cor.-Mem. RAS
MAKAROV V. L., Dr. Sci. (Phys.-Math.), Acad.
RAS
PANCHENKO V. YA., Dr. Sci. (Phys.-Math.),
Acad. RAS
STEMPKOVSKY A. L., Dr. Sci. (Tech.), Acad. RAS
UKHLINOV L. M., Dr. Sci. (Tech.)
FEDOROV I. B., Dr. Sci. (Tech.), Acad. RAS
CHETVERTUSHKIN B. N., Dr. Sci. (Phys.-Math.),
Acad. RAS

Editor-in-Chief:

VASENIN V. A., Dr. Sci. (Phys.-Math.)

Editorial Board:

ANTONOV B.I.
AFONIN S.A., Cand. Sci. (Phys.-Math)
BURDONOV I.B., Dr. Sci. (Phys.-Math)
BORZOVS JURIS, Dr. Sci. (Comp. Sci), Latvia
GALATENKO A.V., Cand. Sci. (Phys.-Math)
GAVRILOV A.V., Cand. Sci. (Tech)
JACOBSON IVAR, Dr. Sci. (Philos., Comp. Sci.),
Switzerland
KORNEEV V.V., Dr. Sci. (Tech)
KOSTYUKHIN K.A., Cand. Sci. (Phys.-Math)
MAKHORTOV S.D., Dr. Sci. (Phys.-Math)
MANCIVODA A.V., Dr. Sci. (Phys.-Math)
NAZIROV R.R. , Dr. Sci. (Tech)
NECHAEV V.V., Cand. Sci. (Tech)
NOVIKOV B.A., Dr. Sci. (Phys.-Math)
PAVLOV V.L., USA
PAL'CHUNOV D.E., Dr. Sci. (Phys.-Math)
PETRENKO A.K., Dr. Sci. (Phys.-Math)
POZDNEEV B.M., Dr. Sci. (Tech)
POZIN B.A., Dr. Sci. (Tech)
SEREBRJAKOV V.A., Dr. Sci. (Phys.-Math)
SOROKIN A.V., Cand. Sci. (Tech)
TEREKHOV A.N., Dr. Sci. (Phys.-Math)
FILIMONOV N.B., Dr. Sci. (Tech)
SHAPCHENKO K.A., Cand. Sci. (Phys.-Math)
SHUNDEEV A.S., Cand. Sci. (Phys.-Math)
SHCHUR L.N., Dr. Sci. (Phys.-Math)
YAZOV Yu. K., Dr. Sci. (Tech)

Editors: CHUGUNOVA A.V.

CONTENTS

Zmeev D. O., Zmeev O. A., Ivanova L. S. Antipattern Practice for Essence Practice Library	311
Gaydamaka K. I., Ognyanovich P. A. Applying Supervised Machine Learning Methods to Determine the Atomicity of Requirements for Complex Technical Systems	322
Alexandrov D. V. Overview of Face Recognition Algorithms for Person Identification	331
Stupina M. V. The Architecture of the Software System for Supporting Attendance Records at the University	344
Ermilov E. E., Postnyi A. V. Curriculum Personalization through the Application of Reinforcement Learning Algorithms	354

Д. О. Змееv, ассистент, denis.zmeev@accounts.tsu.ru,
О. А. Змееv, д-р физ.-мат. наук, проф. кафедры, ozmeyev@gmail.com,
Л. С. Иванова, ассистент, lidiya.ivanova@persona.tsu.ru,
Томский государственный университет

Практика работы с антипаттернами для Essence Practice Library

Представлено расширение для библиотеки практик языка *Essence* в виде практики работы с антипаттернами. Для представления антипаттернов в системе предложены субальфа *Antipattern*, ее состояния и контрольные точки. Для фиксации данных об антипаттерне предложены рабочий продукт *Antipattern Report* и его уровни детализации. Для проведения анализа архитектуры системы предложена активность *Inspect Architecture*. Активность *Fix Architecture* представляет действия по устранению недостатков архитектуры. Анализ кода представлен в виде активности *Review the Code*, а исправление найденных в ходе анализа недостатков — в виде активности *Refactor the Code*. Проанализировано влияние субальфы *Antipattern* на состояние альфы *Software System*. Даны рекомендации о проведении предложенных активностей.

Ключевые слова: антипаттерн, код-ревью, недостаток кода, практика, разработка, рефакторинг, управление проектом, *Essence*, *Practice Library*, SEMAT

Введение

Отрасль разработки программного обеспечения (ПО) является одной из самых быстроразвивающихся, массовый переход на удаленную работу в условиях пандемии ускорил это развитие [1]. За последние годы был предложен ряд методологий по управлению проектами в сфере разработки ПО. Однако статистика, представленная Standish Group в новом отчете CHAOS Report [2], показывает, что только около 31 % проектов завершаются успешно, 50 % вызывают трудности, а 19 % проектов являются провальными, причем данные значения существенно не изменились с 2010 г. [2–4]. Поскольку статистика успешности проектов не улучшается, необходимо рассмотреть различные негативные факторы, отрицательно влияющие на успешность проектов. Такими факторами могут быть риски управления, меняющиеся требования, а также различные проблемы, возникающие в процессе разработки ПО — баги (ошибки) и антипаттерны [5] — неудачные архитектурные решения часто возникающих задач проектирования и/или реализации.

Инициатива SEMAT (*Software Engineering Method And Theory*) [6] является новой попыткой систематизировать имеющиеся методологии и практики программной инженерии, создав единое Ядро и язык описания *Essence* [7], позволяющий описать новые практики понятным практически лю-

бому человеку способом. Создание такого языка по сути является созданием метамодели для описания ранее придуманных и новых методологий. Авторы языка *Essence* опубликовали Библиотеку Практик (*Practice Library*) [8], в которую вошли такие известные методологии (в терминах *Essence* они носят название "методы"), как Унифицированный процесс [9] и гибкая методология разработки Agile [10].

Тема расширения языка *Essence* и Библиотеки Практик уже была затронута в литературе [11–14]. Например, авторы работы [12] предприняли попытку описать подход TOGAF (*The Open Group Architecture Framework* [15]) в терминах *Essence*. Авторы инициативы SEMAT продолжают дорабатывать Библиотеку, например, работа [11] расширяет Библиотеку практикой применения вариантов использования. Интерес вызывает работа [16], авторы которой предложили систему измерений для альфы *Software System* (Программная Система) для оценки качества продукта и получения данных о здоровье и прогрессе проекта. В работе [13] авторы предлагают усилить стандарт с помощью унификации используемой терминологии.

Авторы работ [12, 14] отмечают необходимость расширения Ядра *Essence* и Библиотеки Практик, кроме того, автор книги [17] отмечает необходимость рассмотрения косвенных убытков и причин их возникновения (которыми являются и анти-

паттерны) в рамках результатов работы инициативы SEMAT.

Поскольку Библиотека Практик по-прежнему крайне ограничена, а большинство работ предлагают лишь расширения существующих элементов, возникает необходимость добавления новых методов и практик. Работа с антипаттернами и другими недостатками кода в рамках процессов по разработке ПО может быть представлена в виде практики, которая может использоваться как отдельно, так и в составе новых или уже описанных в Библиотеке методов. Поскольку антипаттерны оказывают на систему отрицательное воздействие, препятствуют ее развитию и могут стать косвенной причиной гибели проекта, разработка данной практики представляет не только практический, но и исследовательский интерес.

Целью работы, результаты которой представлены в статье, является разработка практики работы с антипаттернами в терминах языка Essence. Актуальность темы состоит в необходимости расширения Библиотеки Практик в связи с высоким негативным влиянием антипаттернов на проект.

Научная новизна выполненного авторами исследования определяется отсутствием в Библиотеке Практик каких-либо упоминаний антипаттернов и недостатков кода.

При разработке практики использовался опыт авторов книг по работе с антипаттернами и недостатками кода [5, 18, 19].

1. Элементы языка Essence

Ядро Essence состоит из практик — способов организации специализированных работ в рамках процесса разработки программного продукта. Практики, использованные командой для проекта, объединяются в методы. В табл. 1 приведены краткий обзор элементов языка и их графическая нотация. Описание практики работы с антипаттернами с использованием данных элементов представлено в разд. 3.

В Ядро Essence также входят конкретные Альфы, например, Software System — система, состоящая из ПО, аппаратного обеспечения и данных. Полный перечень альф Ядра представлен на рис. 1.

Таблица 1

Основные элементы языка Essence

Пиктограмма	Название	Определение
	Альфа (<i>Alpha</i>)	Изначально — акроним от английского Abstract Level Progress Health: семантически фиксирует некоторую бизнес-область метода или практики, состояние которой необходимо отслеживать в процессе реализации проекта
	Состояние Альфы (<i>Alpha State</i>)	Сущность, фиксирующая прогресс изменений Alpha (альфы) в ходе реализации метода или практики. Состоит из набора истинных/ложных утверждений, называемых Checkpoint
	Пространство Активностей (<i>Activity Space</i>)	На высоком абстрактном уровне описывает виды деятельности, необходимые в процессе реализации метода или практики (с точки зрения объектно-ориентированного программирования в некотором смысле являются абстрактными классами или интерфейсами)
	Активность (<i>Activity</i>)	Моделирует детальные и конкретные действия, которые необходимо совершить в процессе выполнения проекта, чтобы добиться прогресса, предписываемого практикой (с точки зрения объектно-ориентированного программирования в некотором смысле являются аналогами объектов или классов, реализующих интерфейсы)
	Рабочий продукт (<i>Work Products</i>)	Моделирует сущности, необходимые для описания результатов, которые создаются в ходе выполнения активностей или задач в тех или иных практиках
	Уровень детализации (<i>Level of detail</i>)	Сущность, фиксирующая прогресс, достигнутый рабочим продуктом. Обычно выделяются два подвида: обязательный — со сплошной рамкой — и необязательный — со штриховой рамкой. Фактически выполняют роль, аналогичную состояниям Alpha

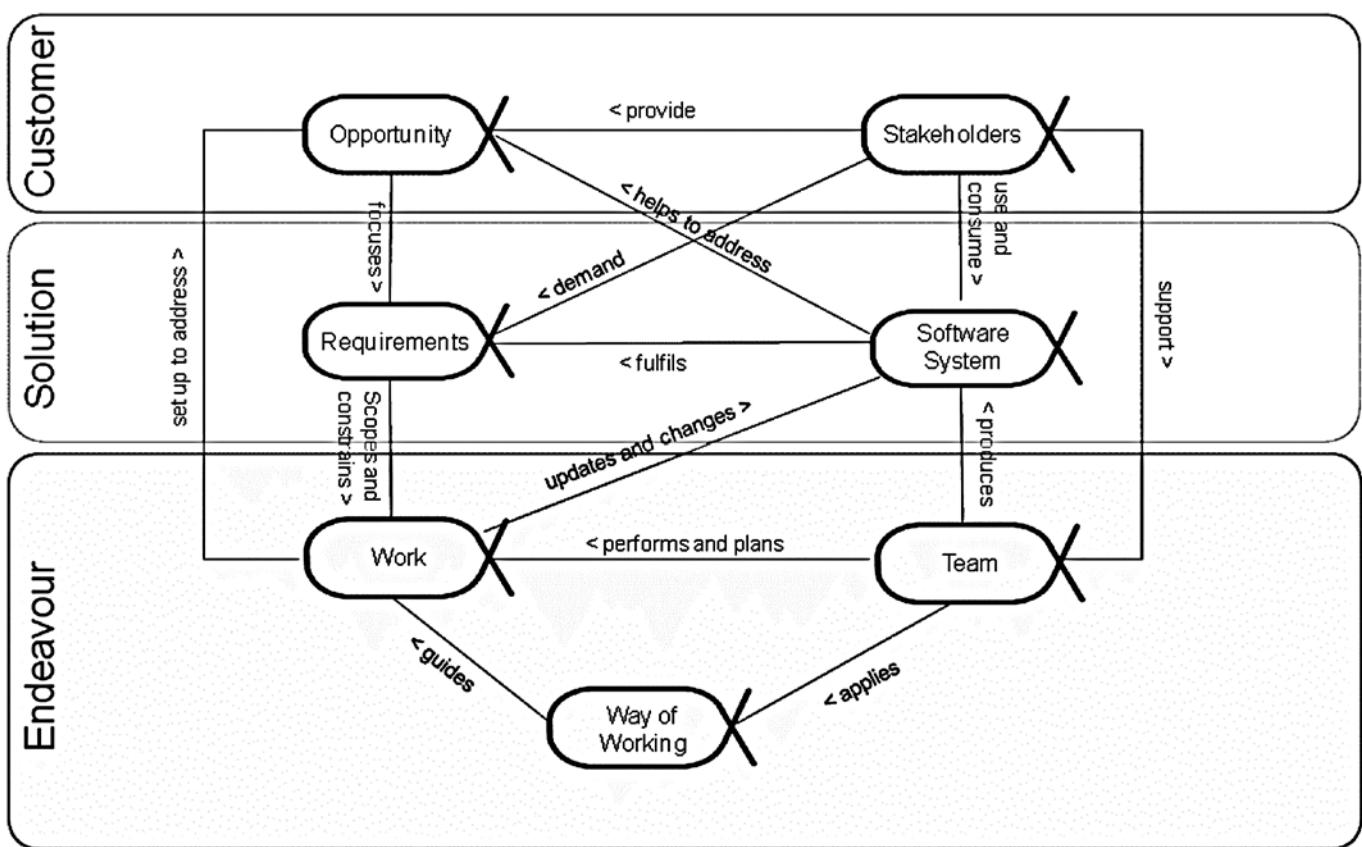


Рис. 1. Альфа Ядра Essence [3]

С точки зрения работы с антипаттернами наибольший интерес среди активностей Ядра представляет активность Implement the System — создать систему, внедрив, протестировав и интегрировав один или несколько ее элементов; включает в себя исправление ошибок и модульное тестирование.

Опираясь на Ядро Essence и Библиотеку Практик, авторами данной статьи была составлена собственная практика работы с антипаттернами в рамках процесса разработки программного продукта.

2. Работа с антипаттернами

В рамках данной статьи термином антипаттерн объединены различные понятия, обозначающие недостатки, возникшие на этапах проектирования и/или реализации программного продукта (antipattern [5], code smell [18]). Примерами антипаттернов являются как "классические" Божественный объект [5], Стрельба дробью [18] и Полтергейст [5], так и недостатки исходного кода, которые рассмотрены в посвященной рефакторингу литературе: Длинный метод [5], Условная сложность [5] и др. Следует уточнить, что ошибки ("баги" и другие примеры некорректного поведения ПО) в контексте данной работы не рассматриваются.

В процессе работы над программным продуктом антипаттерны могут быть обнаружены как во время различных обзоров исходного кода и документации (аудит, код-ревью), так и при добавлении нового или изменения существующего функционала. Следует отметить, что тема автоматизации процесса поиска антипаттернов актуальна [20], поскольку процесс поиска таких недостатков является достаточно трудоемким, особенно, если речь идет о крупных программных продуктах.

Найденный антипаттерн становится частью так называемого технического долга [21]. В дальнейшем, если затраты на устранение недостатка не превышают возможные финансовые потери от его присутствия в системе, найденный антипаттерн устремляется с помощью процедуры рефакторинга. Следует также учитывать, что, по мнению авторов книги [18], "важным предварительным условием проведения рефакторинга является наличие надежных тестов".

3. Практика работы с антипаттернами

Практика работы с антипаттернами представляет собой рекомендации по работе с различными недостатками проектирования и реализации, которые могут быть найдены в диаграмме или в ис-

ходном коде программной системы. В нее вошли субальфа Антипаттерн, рабочий продукт Отчет об Антипаттерне и активности Проверить архитектуру, Исправить архитектуру, Проверить код, Провести рефакторинг кода.

3.1. Субальфа Антипаттерн

Для представления антипаттерна и работы с ним в рамках процесса разработки программного продукта авторами настоящей статьи предлагается субальфа (альфа второго уровня) Антипаттерн (*Antipattern*) — распространенное неэффективное решение часто возникающей задачи проектирования и/или реализации. Антипаттерн не является причиной некорректной работы приложения, однако может быть причиной будущих проблем модификации и повторного использования. Далее в работе при использовании термина Антипаттерн в смысле субальфы термин будет употреблен с заглавной буквы, в контексте антипаттерна какой-то системы — с прописной.

Антипаттерн является субальфой, представленной в Ядре альфы Software System (рис. 2), он препятствует ее развитию, поскольку наличие в Software System антипаттернов может отрицательно сказаться на ее способности к прогрессу. Степень отрицательного влияния оценить достаточно сложно.

На протяжении всего жизненного цикла в ПО могут возникать антипаттерны — неудачные решения, отрицательно влияющие на повторное использование и расширяемость системы, а также удобочитаемость кода. Такие недостатки не обнаруживаются в результате тестирования и являются техническим долгом, который нужно устранять.

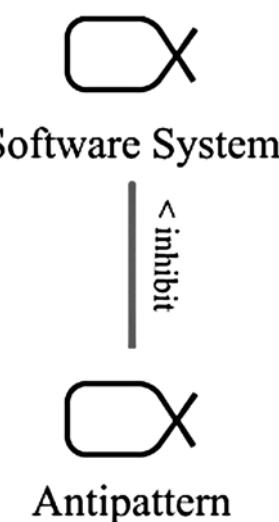


Рис. 2. Субальфа Антипаттерн

Антипаттерны могут препятствовать развитию Software System, в которой они присутствуют. Они могут быть обнаружены и идентифицированы во всех состояниях Software System, кроме Retired, поскольку в данном состоянии работа с разрабатываемой системой полностью прекращается. В состоянии Architecture Selected антипаттерны могут быть обнаружены в документации к проекту (диаграммах). В состояниях Demonstrable — Operational антипаттерны могут быть обнаружены в документации и исходном коде системы. Своевременное устранение антипаттернов увеличивает скорость разработки и исправления ошибок, т. е. способствует более быстрому переходу в следующее состояние.

Состояния субальфы Антипаттерн и соответствующие им контрольные точки представлены в табл. 2 (поскольку большинство опубликованных в настоящий момент описаний практик существуют только на английском языке, в тексте данной работы приведены русские описания, но в расширенной английской аннотации приведен английский вариант).

Как показано в табл. 2, Антипаттерн проходит через состояния подозревается, идентифициирован, метод рефакторинга определен, исправлен, закрыт. Если детально описывать жизненный цикл Антипаттерна, складывается описанная ниже последовательность.

Сначала, в результате ревью, исправления бага, разработки нового функционала может возникнуть подозрение на антипаттерн. Область, в которой обнаружено подозрение на антипаттерн, помечается для анализа. В результате анализа антипаттерн идентифицируется: определяются его границы, тип, зависимости. Определяется степень влияния антипаттерна на систему. В случае, если влияние незначительно (устранение антипаттерна дороже, чем отрицательный урон от его наличия в системе), работу по устранению можно прекратить. В зависимости от типа антипаттерна определяется способ его устранения. Затем проводится процедура рефакторинга, антипаттерн устраняется. С помощью тестирования и ревью определяется, что антипаттерн устранен, новых недостатков не возникло, следовательно, антипаттерн можно считать закрытым.

Состояние субальфы Антипаттерн не зависит от состояния Software System. Если поддержка и расширение функционала системы не планируется — антипаттерны могут остаться в состоянии подозревается или идентифицирован. Аналогично для отдельных элементов системы: если не планируется их поддержка, расширение или повторное использование — найденные в данных областях антипаттерны можно не устранять.

Состояния субальфы Антипаттерн

Состояние	Описание	Контрольные точки
Обнаружен	Выделена часть системы, которая содержит антипаттерн	Обнаружены признаки некачественного проектирования или реализации системы. Определены границы области низкого качества в системе. Определены рабочие продукты, связанные с низкокачественной частью системы
Идентифицирован	Определены местоположение антипаттерна в системе и его тип	Определены элементы системы, входящие в антипаттерн. Определен тип антипаттерна. Выявлены зависимости системы от элементов, составляющих антипаттерн. Проведена оценка влияния антипаттерна на систему
Метод рефакторинга определен	Проведен выбор наиболее подходящего метода исправления (рефакторинга) антипаттерна	Определены возможные варианты рефакторинга. Проведена оценка стоимости выявленных вариантов рефакторинга. Выбран наиболее подходящий вариант рефакторинга
Исправлен	Антипаттерн удален из системы	Элементы системы, входящие в антипаттерн, полностью покрыты тестами. Проведен рефакторинг антипаттерна. Тесты подтверждают, что работоспособность системы была сохранена. Обновлены Рабочие продукты, связанные с измененной частью системы
Закрыт	Отсутствие антипаттерна в системе подтверждено	Были проведены тесты, обзоры или другие соответствующие действия, чтобы гарантировать, что антипаттерн исправлен или не является фактическим дефектом или недостатком. Завершены мероприятия по управлению антипаттерном. Технический долг уменьшился

3.2. Рабочий продукт Отчет об Антипаттерне

Для работы с информацией об Антипаттерне в рамках проекта предлагается рабочий продукт (*Work Product*) Отчет об Антипаттерне (*Antipattern Report*). Отношения между рабочим продуктом и альфами представлены на рис. 3.

Уровни детализации данного артефакта представлены в табл. 3. Отчет об Антипаттерне описывает местонахождение, тип антипаттерна, а также содержит информацию о работах, проведенных по его устранению. При использовании различных программных продуктов для управления проектом Отчет об Антипаттерне может быть представлен в виде задачи (*Task*, *Issue*), записи в *Wiki* или *Readme*-файлах проекта.

В качестве "частей системы" могут выступать: класс, совокупность связанных классов, слой, выделяемый в системе, и другие элементы, являющиеся составными элементами антипаттерна.

3.3. Активности

Для формализации процедур по изменению состояния субальфы Антипаттерн предлагаются следующие Активности, представленные на рис. 4.

Рассмотрим отдельные активности, представленные на рис. 4.

1. *Inspect Architecture* (Проверить архитектуру) — проверить архитектуру системы на наличие антипаттернов (рис. 5).

Данную активность следует проводить для всех частей системы.

Проверить архитектуру необходимо для того, чтобы:

- проверить качество проектирования системы;

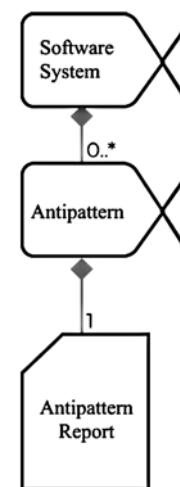


Рис. 3. Альфы и рабочий продукт Antipattern Report (Отчет об Антипаттерне)

Уровни детализации рабочего продукта Отчет об Антипаттерне

Уровень детализации	Описание	Контрольные точки
Обнаружено наличие антипаттерна	В системе обнаружен недостаток	Определены части системы, входящие в антипаттерн
Определены характеристики антипаттерна	Обнаруженный в системе недостаток локализован, описан и проанализирован	Определен тип антипаттерна
Определено влияние антипаттерна	Проведена оценка влияния антипаттерна и принято решение о целесообразности его устранения	Проведена оценка стоимости исправления антипаттерна. Проведена оценка возможных потерь, связанных с наличием антипаттерна. Антипаттерн признан обязательным к исправлению
Антипаттерн устранен (необязателен)	Недостаток, найденный в системе, полностью устранен, все связанные работы успешно завершены	Проведен рефакторинг. Тестирование показало корректность работы частей системы, входящих в антипаттерн. Деятельность, связанная с устранением антипаттерна, завершена

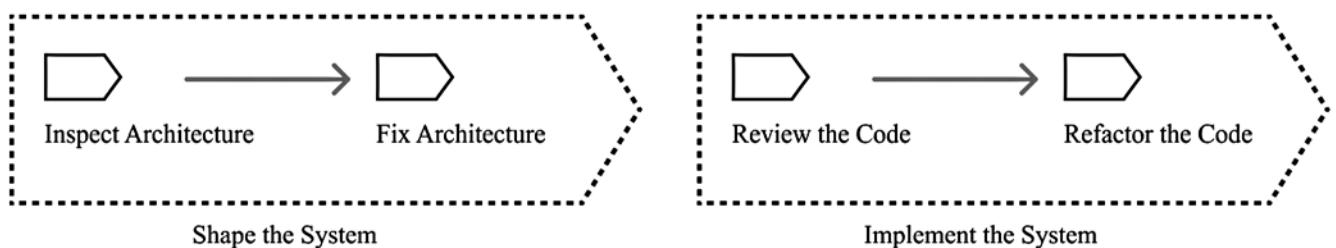


Рис. 4. Активности по работе с субальфой Антипаттерн

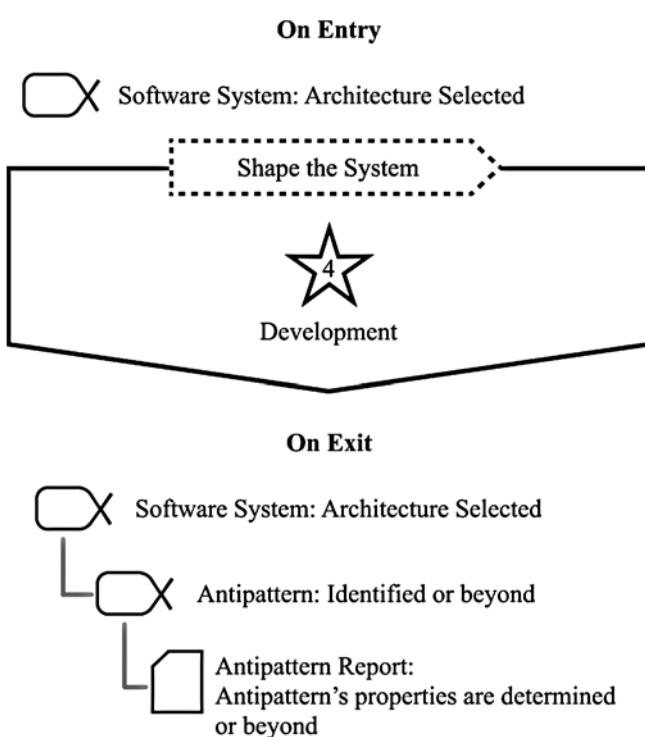


Рис. 5. Активность Inspect Architecture

- проверить отсутствие антипаттернов;
- проверить отсутствие багов;
- увеличить осведомленность команды о проекте.

Критерии завершения активности: система проверена полностью. Проверяющие члены команды согласны, что найденные проблемные вопросы устраниены, либо признаны незначительными.

Описываемая в данной работе практика может быть использована в составе методов, например, при включении практики антипаттернов в метод *Essential Unified Process*, активность *Inspect Architecture* будет взаимодействовать с субальфами Компонент (*Component*) и Архитектура (*Architecture*) (рис. 6). В результате активности также могут быть выявлены баги (расхождения с требованиями к системе).

2. *Fix Architecture* (*Исправить архитектуру*) — устраниТЬ антипаттерны в архитектуре системы (рис. 7).

Данная активность позволяет устраниТЬ существующий антипаттерн, а также улучшить качество архитектуры системы, что положительно скажется на повторной используемости и расширяемости.

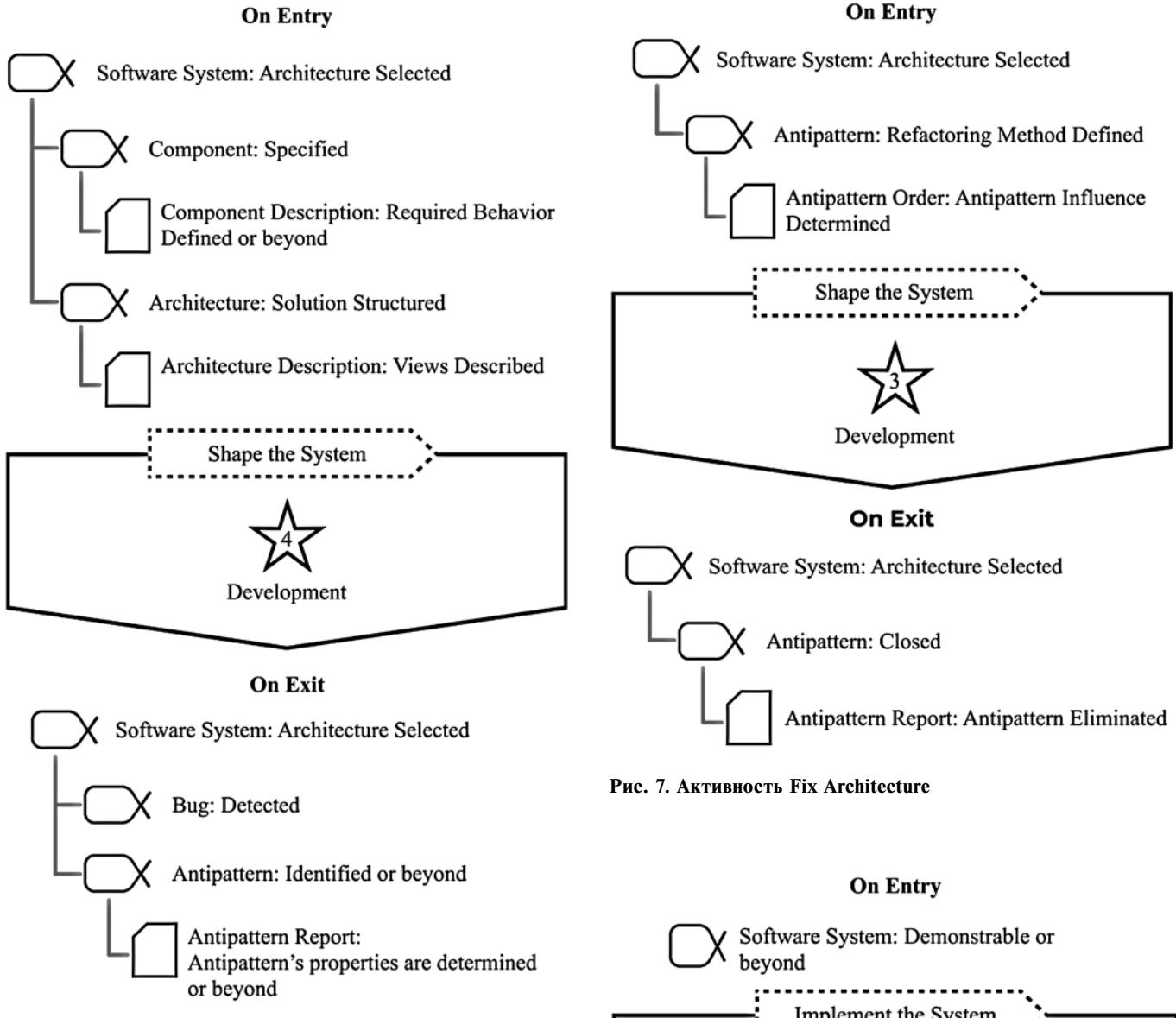


Рис. 6. Активность Inspect Architecture в методе Essential Unified Process

Критерий завершения данной активности — Антипаттерн, найденный ранее, устранен. Проверяющие члены команды согласны, что найденные проблемные вопросы устраниены, либо признаны незначительными.

В составе метода *Essential Unified Process* данная активность может использовать и изменять рабочие продукты *Component Description* и *Architecture Description*. В результате ее проведения может измениться состояние субальфы Баг на Fixed (ошибка может быть исправлена).

3. *Review the Code* (Проверить код) — провести код-ревью исходного кода системы (рис. 8).

Код-ревью необходимо, чтобы:

- проверить качество исходного кода;
- проверить отсутствие антипаттернов;

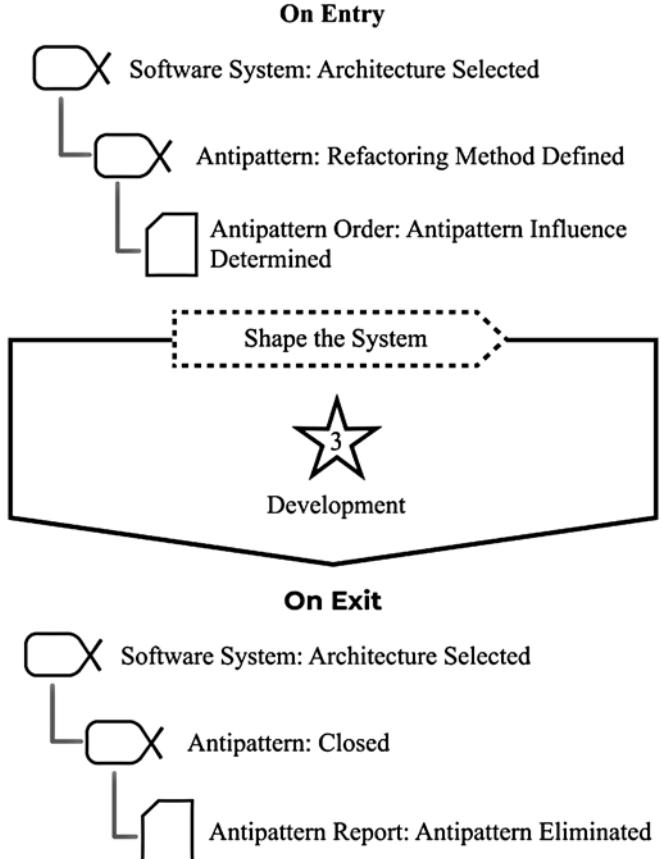


Рис. 7. Активность Fix Architecture

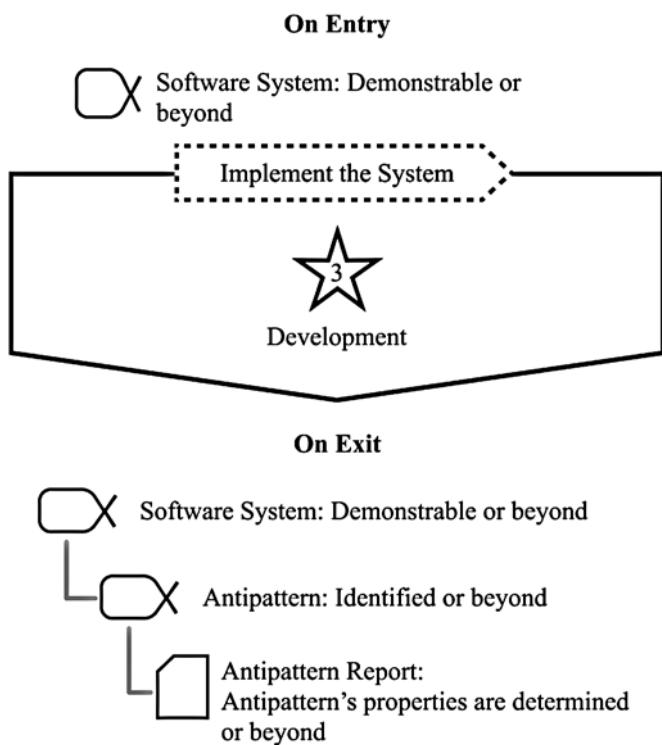


Рис. 8. Активность Review the Code

- проверить отсутствие багов, обнаруживаемых в исходном коде;
- увеличить осведомленность команды о проекте.

Данную активность следует проводить для всех компонентов системы.

Критерии завершения активности: система проверена полностью. Проверяющие члены команды согласны, что найденные проблемные вопросы устранены, либо признаны незначительными.

Чаще всего методика проведения код-ревью определяется правилами, установленными в конкретной команде разработчиков. В литературе [22, 23] предлагаются перечисленные далее варианты проведения процедуры код-ревью.

• Проверка кода (*Code Inspections*) [23]. Проверки кода проводятся после завершения проектирования участка программной системы, после завершения реализации участка системы и после создания модульных тестов для участка системы. Данный подход подробно описан в статье [23].

• Современный метод (*Modern Code Review*) [22]. Для автоматизации процесса могут быть использованы различные инструменты, такие как Gerrit [24], CodeFlow [25], а также встроенные средства GitHub [26], GitLab [27]. Один или несколько ревьюеров проверяют изменения кодовой базы, проведенные членом команды перед добавлением изменений в репозиторий. Процесс можно разделить на несколько итераций, которые продолжаются, пока не будут устранены все значимые из найденных недостатков и все обнаруженные баги. Подробное описание варианта реализации подхода в компании Google можно найти в работе [22].

4. *Refactor the Code* (*Провести рефакторинг кода*) — провести рефакторинг исходного кода системы, не изменяя ее поведение (рис. 9).

Проведение рефакторинга необходимо, чтобы:

- устранить технический долг;
- улучшить читаемость кода;
- улучшить расширяемость системы;
- улучшить способность кода к повторному использованию.

Данная активность позволяет устраниТЬ существующий антипаттерн, а также улучшить качество исходного кода компонента, что положительно скажется на повторной используемости, читаемости, расширяемости исходного кода. В процессе рефакторинга могут быть найдены и устранены ошибки системы.

Критерий завершения данной активности — антипаттерн устранен без нарушения работоспособности системы: недостаток, найденный ранее,

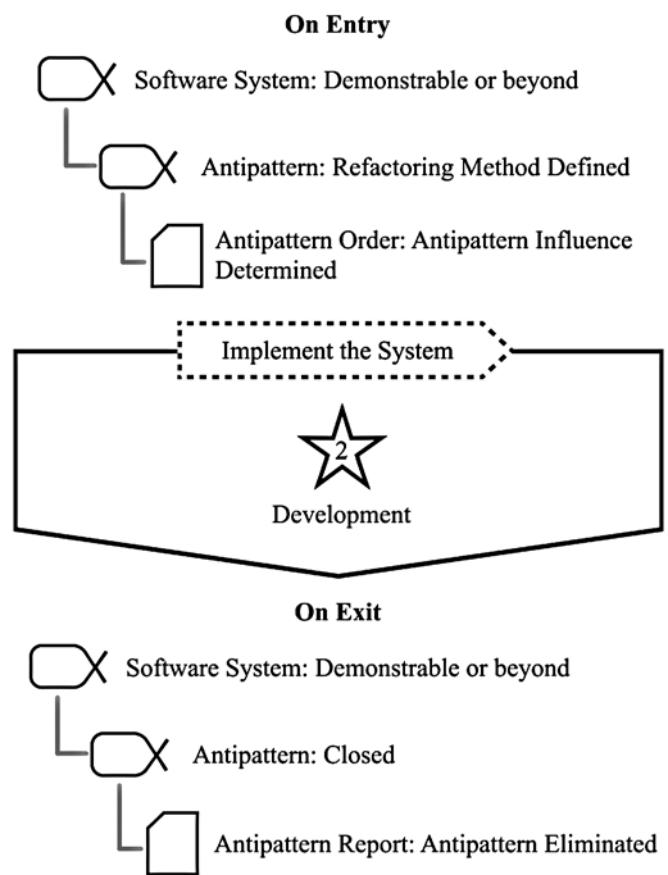


Рис. 9. Активность Refactor the Code

устранен. Тесты, покрывающие данный фрагмент системы, успешно завершаются. Внешнее поведение системы не изменилось. Качество кода данного фрагмента системы улучшилось.

Применимые способы рефакторинга зависят от вида найденного недостатка кода. Различные способы рефакторинга подробно описаны в литературе [18, 19]. Согласно классической процедуре проведения рефакторинга [18], для проверки успешности проведения рефакторинга необходимо наличие автоматических тестов. Для работы с тестами в Ядре Essence присутствует пространство активностей Протестировать Систему (*Test the System*).

Заключение

Предложенная в настоящей статье практика позволяет формализовать процедуру по работе с антипаттернами при разработке ПО. Возможна комбинация предложенного подхода с другими практиками.

Предложенная субальфа Антипаттерн является одним из примеров субальфы, которая может оказывать отрицательное влияние на развитие проекта. Рабочий продукт, подтверждающий на-

личие в Software System неудачных решений — Отчет об Антипаттерне, — является свидетельством того, что в будущем могут появиться различные проблемные вопросы, связанные с расширением и развитием разрабатываемой системы. Это обстоятельство может стать основанием для переработки существующего исходного кода или переосмысления принятых архитектурных решений.

Результаты работы планируется использовать в расширении системы поддержки принятия решений [28], поскольку сам факт существования антипаттернов в системе может являться флагом того, что менеджер ошибается в своих предположениях относительно текущего состояния проекта.

Список литературы

1. Digital Economy Report 2021. Cross-border data flows and development: For whom the data flow. URL: https://unctad.org/system/files/official-document/der2021_en.pdf
2. Portman H. Review Standish Group — CHAOS 2020: Beyond Infinity. URL: <https://hennyportman.wordpress.com/2021/01/06/review-standish-group-chaos-2020-beyond-infinity/>
3. Wojewoda S., Hastie S. Standish Group 2015 Chaos Report — Q&A with Jennifer Lynch. URL: <https://www.infoq.com/articles/standish-chaos-2015/>
4. The Standish Group — CHAOS Report. Project Smart. URL: <https://www.projectsmart.co.uk/white-papers/chaos-report.pdf>
5. Brown W. J., Malveau R. C., McCormic III H. W., Mowbray T. J. AntiPatterns: Refactoring Software, Architectures, and Projects in Crisis. New York: John Wiley & Sons, 1998.
6. Пак Дж., Якобсон И., Майбург Б., Джонсон П. SEMAT вчера, сегодня и завтра: перспективы промышленного использования // Программная инженерия. 2014. № 11. С. 6–16.
7. Jacobson I., Ng P.-W., McMahon P. E., Spence I., Lidman S. The Essence of Software Engineering: Applying the SEMAT Kernel. Addison-Wesley, 2013. 224 p.
8. Practice Library. URL: <https://practicelibrary.ivarjacobson.com/start>
9. Scott K. The Unified Process Explained, 1st edition. Addison-Wesley Professional, 2001. 208 p.
10. Beck K., Beedle M., van Bennekum A. et al. Agilemanifesto разработки программного обеспечения. URL: <https://agilemanifesto.org/iso/ru/manifesto.html>
11. Jacobson I., Spence I., Bittner K. USE-CASE 2.0: The Guide to Succeeding with Use Case. Ivar Jacobson International, 2011. 55 p.
12. Munera D., Villa G. F. A brief TOGAF description using SEMAT Essence Kernel. URL: https://www.researchgate.net/publication/335855005_A_brief_TOGAF_description_using_SEMAT_Essence_Kernel
13. Zapata-Jaramillo C., Henao-Roqueme A. A proposal for improving the Essence standard by using terminology unification // Ingenieria. 2021. Vol. 26, No. 2. P. 213–232. DOI: 10.14483/23448393.16428.
14. Simonette M., Spina E. Software & Systems Engineering Interplay and the SEMAT Kernel // The Voice of the Systems - The Journal Of The Israeli Systems Engineers. 2018. Issue 22. P. 6–20.
15. The TOGAF Standard, Version 9.2 Overview. URL: <https://www.opengroup.org/togaf>
16. Perdomo Charry W., Zapata C. Software quality measures and their relationship with the states of the software system alpha // Ingeniare. 2021. Vol. 29. P. 346–363. DOI: 10.4067/S0718-33052021000200346.
17. Jones C. Software Development Patterns and Antipatterns. CRC Press, 2022. 512 p. DOI: 10.1201/9781003193128.
18. Фаулер М., Бек К., Брант Д. и др. Рефакторинг: улучшение проекта существующего кода. СПб: Альфа-книга, 2017. 448 с.
19. Кириевски Д. Рефакторинг с использованием шаблонов. М.: Вильямс, 2016. 400 с.
20. Змеев О. А., Иванова Л. С. Поиск артефактов проектирования. Обзор подходов // Вестник Томского государственного университета. Управление, вычислительная техника и информатика. 2015. № 2 (31). С. 81–90.
21. Cunningham W. The WyCash Portfolio Management System // Addendum to the proceedings on Object oriented programming systems, languages, and applications. 1992. P. 29–30.
22. Sadowski C., Söderberg E., Church L., Sipko M., Bacchelli A. Modern code review: a case study at google // Proceedings of the 40th International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP'18). Association for Computing Machinery, New York, NY, USA, 2018. P. 181–190. DOI: 10.1145/3183519.3183525.
23. Fagan M. Design and code inspections to reduce errors in program development // IBM Systems Journal. 1976. Vol. 15, No. 3. P. 182–211.
24. Gerrit Code Review. URL: <https://www.gerritcodereview.com>
25. CodeFlow. URL: <https://www.getcodeflow.com>
26. GitHub. URL: <https://github.com>
27. GitLab. URL: <https://about.gitlab.com>
28. Змеев Д. О., Иванова Л. С., Рафикова Р. Р. О представлении прогресса проекта по разработке программного обеспечения в форме динамической байесовской сети // Информационные технологии и математическое моделирование (ИТММ-2020). Материалы XIX Международной конференции имени А. Ф. Терпугова. Томск, 2021. С. 291–297.

Antipattern Practice for Essence Practice Library

D. O. Zmeev, denis.zmeev@accounts.tsu.ru, O. A. Zmeev, ozmeyev@gmail.com,
L. S. Ivanova, lidiya.ivanova@persona.tsu.ru,
Tomsk State University, Tomsk, 634050, Russian Federation

Corresponding author:

Lidiya S. Ivanova, Assistant, Tomsk State University, 634050, Tomsk, Russian Federation
E-mail: lidiya.ivanova@persona.tsu.ru

Received on April 03, 2022

Accepted on May 26, 2022

The Essence graphical representation language allows to describe various project management practices in software development. At the moment, the Practice Library describes the most popular development methodologies, but work with various risks, such as code smells or antipatterns, which may be cause of future problems, is not represented.

This article presents an extension for the Practice Library of the Essence language in the form of a practice for working with antipatterns. To represent antipatterns in the system, the Antipattern subalpha, its states and checkpoints are proposed. Antipattern's states and checkpoints:

1. Detected:

- Signs of poor-quality design or implementation of the Software System have been found.
- The boundaries of the low-quality area in the Software System have been defined.
- Work Products associated with the inferior part of the Software System have been defined.

2. Identified:

- The Software System Elements included in the Antipattern have been determined.
- The type of antipattern has been determined.
- Dependencies of the Software System on the antipattern elements have been determined.
- The impact of the antipattern on the Software System has been estimated.

3. Refactoring Method Defined:

- Possible options for refactoring have been identified.
- The cost of the refactoring options has been assessed.
- The most suitable refactoring method has been selected.

4. Fixed:

- The Software System elements included in the antipattern are completely covered by tests.
- The antipattern has been refactored.
- Tests confirm that the Software System remains operational.
- The Work Products related to the changed part of the Software System have been updated.

5. Closed:

- Tests, reviews or other appropriate activities have been undertaken to ensure that the antipattern has been corrected or shown not to actually be a fault or flaw.
- The antipattern management has been finalized.
- Technical debt has decreased.

To record data about an antipattern, the work product Antipattern Report and its levels of details with checkpoints are proposed. Levels of details and checkpoints are:

1. Antipattern detected:

- The components of the Software System included in the antipattern are defined.

2. Antipattern's properties are determined:

- An antipattern type has been defined.

3. Antipattern influence determined:

- The cost of antipattern fixing was estimated.
- An assessment of possible losses associated with the presence of an antipattern was carried out.
- The antipattern is recognized as mandatory for elimination.

4. Antipattern eliminated — optional:

- Refactoring completed.
- Testing showed the correct work of the components included in the antipattern.
- Antipattern elimination activity completed.

To analyze the architecture of the system, the Inspect Architecture activity is proposed. The Fix Architecture activity represents actions to fix architecture flaws. Code analysis is presented as a Review the Code activity, and correction of deficiencies found during the analysis is presented as a Refactor the Code activity. The influence of subalpha on the state of the Software System alpha is analyzed. Recommendations were given on the proposed activities. Information about activities is presented in the form of diagrams in the Essence language.

The proposed practice allows to record information about the found flaws in the code, process them correctly and avoid problems with the project in the future. It is an example of working with entities that negatively affect the progress of the project.

Keywords: Antipattern, Code Review, Code Smell, Development, Essence, Practice, Practice Library, Project Management, Refactoring, SEMAT

For citation:

Zmeev D. O., Zmeev O. A., Ivanova L. S. Antipattern Practice for Essence Practice Library, *Programmnaya Ingeneria*, 2022, vol. 13, no. 7, pp. 311–321.

DOI: 10.17587/prin.13.311-321

References

1. **Digital Economy Report** 2021. Cross-border data flows and development: For whom the data flow, available at: https://unctad.org/system/files/official-document/der2021_en.pdf
2. **Portman H.** Review Standish Group — CHAOS 2020: Beyond Infinity, available at: <https://hennyportman.wordpress.com/2021/01/06/review-standish-group-chaos-2020-beyond-infinity/>
3. **Wojewoda S., Hastie S.** Standish Group 2015 Chaos Report — Q&A with Jennifer Lynch, available at: <https://www.infoq.com/articles/standish-chaos-2015/>
4. **The Standish Group** — CHAOS Report, available at: <https://www.projectsmart.co.uk/white-papers/chaos-report.pdf>
5. **Brown W. J., Malveau R. C., McCormic III H. W., Mowbray T. J.** *AntiPatterns: Refactoring Software, Architectures and Projects in Crisis*, New York, John Wiley & Sons, 1998.
6. **Park J. S., Jacobson I., Myburgh B., Johnson P.** SEMAT Yesterday, Today and Tomorrow. An Industrial Perspective, *Программная Инженерия*, 2014, no. 11, pp. 6–16 (in Russian).
7. **Jacobson I., Ng P.-W., McMahon P. E., Spence I., Liddman S.** *The Essence of Software Engineering: Applying the SEMAT Kernel*, Addison-Wesley, 2013, 224 p.
8. Practice Library, available at: <https://practicelibrary.ivarjacobson.com/start>
9. **Scott K.** *The Unified Process Explained*, 1st edition, Addison-Wesley Professional, 2001, 208 p.
10. **Beck K., Beedle M., van Bennekum A.** et al. Manifesto for Agile Software Development, available at: <https://agilemanifesto.org>
11. **Jacobson I., Spence I., Bittner K.** USE-CASE 2.0: The Guide to Succeeding with Use Case. Ivar Jacobson International, 2011. 55 p.
12. **Munera D., Villa G. F.** A brief TOGAF description using SEMAT Essence Kernel. available at: https://www.researchgate.net/publication/335855005_A_brief_TOGAF_description_using_SEMAT_Essence_Kernel
13. **Zapata-Jaramillo C., Henao-Roqueme A.** A proposal for improving the Essence standard by using terminology unification, *Ingenieria*, 2021, vol. 26, no. 2, pp. 213–232. DOI: 10.14483/23448393.16428.
14. **Simonette M., Spina E.** Software & Systems Engineering Interplay and the SEMAT Kernel, *The Voice of the Systems* —
15. **The TOGAF Standard**, Version 9.2 Overview, available at: <https://www.opengroup.org/togaf>
16. **Perdomo Charry W., Zapata C.** Software quality measures and their relationship with the states of the software system alpha, *Ingeniare*, 2021, vol. 29. pp. 346–363. DOI: 10.4067/S0718-33052021000200346.
17. **Jones C.** Software Development Patterns and Antipatterns. CRC Press, 2022. DOI: 10.1201/9781003193128.
18. **Fowler M., Beck K., Brant J., Opdyke W., Roberts D.** *Refactoring: Improving the Design of Existing Code*, Boston: Addison-Wesley Professional, 1999.
19. **Kerievsky J.** *Refactoring to Patterns*, Boston, Addison-Wesley Professional, 2004, 367 p.
20. **Zmeev O. A., Ivanova L. S.** Design artifacts detection. Review of the approaches, *Vestnik Tomskogo gosudarstvennogo universiteta. Управление, вычислительная техника и информатика*, 2015, no. 2 (31), pp. 81–90 (in Russian).
21. **Cunningham W.** The WyCash Portfolio Management System, *Addendum to the proceedings on Object oriented programming systems, languages, and applications*, 1992, pp. 29–30.
22. **Sadowski C., Söderberg E., Church L., Sipko M., Bacchelli A.** Modern code review: a case study at google, *Proceedings of the 40th International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP '18)*. Association for Computing Machinery, New York, NY, USA, 2018, pp. 181–190. DOI: 10.1145/3183519.3183525.
23. **Fagan M.** Design and code inspections to reduce errors in program development, *IBM Systems Journal*, 1976, vol. 15, no. 3, pp. 182–211.
24. **Gerrit** Code Review, available at: <https://www.gerritcodereview.com>
25. **CodeFlow**, available at: <https://www.getcodeflow.com>
26. **GitHub**, available at: <https://github.com>
27. **GitLab**, available at: <https://about.gitlab.com>
28. **Zmeev D. O., Ivanova L. S., Rafikova R. R.** Presenting the progress of a software development project in the form of a dynamic Bayesian network, *Информационные технологии и математическое моделирование (ITMM-2020). Материалы XIX Международной конференции имени А. Ф. Терпугова*, Tomsk, 2021, pp. 291–297 (in Russian).

ИНФОРМАЦИЯ

Продолжается подписка на журнал "Программная инженерия" на второе полугодие 2022 г.

Оформить подписку можно через подписные агентства

или непосредственно в редакции журнала.

Подписной индекс по Объединенному каталогу

"Пресса России" — 22765

Сообщаем, что с 2020 г. возможна подписка
на электронную версию нашего журнала через:

ООО "ИВИС": тел. (495) 777-65-57, 777-65-58; e-mail: sales@ivis.ru,

ООО "Урал-Пресс округ". Для оформления подписки (индекс 013312)
следует обратиться в филиал по месту жительства — <http://ural-press.ru>

Адрес редакции: 107076, Москва, Матросская Тишина, д. 23, оф. 45,

Издательство "Новые технологии",
редакция журнала "Программная инженерия"

Тел.: (499) 270-16-52. E-mail: prin@novtex.ru

Applying Supervised Machine Learning Methods to Determine the Atomicity of Requirements for Complex Technical Systems¹

K. I. Gaydamaka, k.gaydamaka@gmail.com, Department of Systems Engineering,
"MIREA — Russian Technological University", Moscow, 119454, Russian Federation,
P. A. Ognyanovich, Pasha2la71@gmail.com,
National Research Nuclear University "MEPhI"

Corresponding author:

Kirill I. Gaydamaka, Senior Lecturer, Department of Systems Engineering, "MIREA — Russian Technological University", Moscow, 119454, Russian Federation
E-mail: k.gaydamaka@gmail.com

Received on November 16, 2021

Accepted on June 06, 2022

The article is devoted to the problem of determining the atomicity of requirements for complex technical systems. The purpose of this article is to apply supervised machine learning methods, namely classification, to determine the atomicity of requirements. It is assumed that feature engineering based on the linguistic features of requirements statements will make it possible to classify requirements into atomic and non-atomic ones with high accuracy. The article describes the use of the following methods for generating features: fastText, BERT, customFeature. Requirements are classified using a logistic classifier, decision trees, random forest, and gradient boosting. The best combination of methods turned out to be using customFeature with gradient boosting.

Keywords: gradient boosting, BERT, requirements management, machine learning, requirements atomicity, requirements quality

For citation:

Gaydamaka K. I., Ognyanovich P. A. Applying Supervised Machine Learning Methods to Determine the Atomicity of Requirements for Complex Technical Systems, *Programmnaya Ingeneriya*, 2022, vol. 13, no. 7, pp. 322—330.

УДК 004.852

К. И. Гайдамака, ст. преподаватель, k.gaydamaka@gmail.com,
РТУ МИРЭА,
П. А. Огнянович, студент, Pasha2la71@gmail.com,
Национальный исследовательский ядерный университет "МИФИ"

Применение методов машинного обучения с учителем для определения атомарности требований к сложным техническим системам

Статья посвящена проблеме определения атомарности требований к сложным техническим системам. Цель этой статьи — исследовать возможность применения методов машинного обучения с учителем, в частности, классификации, для определения атомарности требований. Предполагается, что разработка признаков, основанная на лингвистических особенностях формулировок требований, позволит с высокой точностью классифицировать требования

¹ The article is based on the materials of the report at the Seventh International Conference "Actual problems of Systems and Software Engineering" APSSE 2021.

на атомарные и неатомарные. Описано использование следующих методов формирования признаков: *fastText*, *BERT*, *customFeature*. Требования классифицируются с использованием логистического классификатора, метода деревьев решений, случайного леса и градиентного бустинга. Экспериментальное исследование показало, что наилучшей комбинацией методов является сочетание *customFeature* с градиентным бустингом.

Ключевые слова: градиентный бустинг, *BERT*, управление требованиями, машинное обучение, атомарность требований, качество требований

Introduction

The development of high-quality system requirements plays a decisive, key role in the process of working on a project, regardless of its scale [1]. Time, development cost and other indicators of the project's effectiveness directly depend on how accurately and correctly, the system requirements describe the system that must be created. Taking into account the fact, how complex, large-scale projects can be, it would be logical to assert that the number of requirements in them can amount to several thousand [2]. And since the development of requirements is entirely on the shoulders of the requirements engineers, it would be logical to assume also that it should not depend on the human factors: inattention, negligence or haste. But in cases where huge sums of money are at stake, and due to the imperfection of the human resources, the project executor may suffer significant losses, it is necessary to minimize the human factor.

The requirements make it possible to determine what the interested parties want to get from the system under development and what properties the system should have in order to satisfy their needs.

The quality of the requirements can be judged by the presence of several quality criteria. There are certain criteria that every statement of requirement should meet. These are summarised as follows [3]:

- *Atomic*: each statement carries a single traceable element;
- *Unique*: each statement can be uniquely identified;
- *Feasible*: technically possible within cost and schedule;
- *Legal*: legally possible;
- *Clear*: each statement is clearly understandable;
- *Precise*: each statement is precise and concise;
- *Verifiable*: each statement is verifiable, and it is known how;
- *Abstract*: does not impose a solution of design specific to the layer below.

In addition, there are other criteria that apply to the set of requirements as a whole:

- *Complete*: all requirements are present;
- *Consistent*: no two requirements are in conflict;

- *Non-redundant*: each requirement is expressed once;
- *Modular*: requirements statements that belong together are close to one another;
- *Structured*: there is a clear structure to the requirements document;
- *Traceable*: the appropriate degree of traceability coverage has been achieved.

The article is devoted to the problem of determining the atomicity of requirements for complex technical systems. The purpose of this article is to apply supervised machine learning methods, namely classification, to determine the atomicity of requirements.

Description of the dataset

The initial data is a table of 287 records with the following columns: "ID", "Requirement" and "Singular", where "ID" is the unique identification number of the requirement, "Requirement" is the textual wording of the requirement statement, and "Singular" is the tag of membership of each requirement in the class of atomic or non-atomic (1 and 0, respectively). All requirements statements in dataset are in Russian.

Feature engineering

There is one very interesting principle in computer science: garbage in, garbage out (GIGO). That is, if incorrect data is fed to the algorithm, the results will be incorrect, even if the algorithm is the correct and accurate. Supervised machine learning algorithms are no exception [4]. Therefore, data preprocessing and feature engineering are very important steps in machine learning. Consider two neural network-based natural language processing models: fasttext and *BERT*, and custom created features.

FastText is a library developed by Facebook that contains pre-trained ready-made vector representations of words [5]. The algorithm uses both CBOW (continuous bag of words) and skipgram models. For us, the main interest is that FastText does not require training. There are ready-made models for 157 languages, including Russian. By default FastText uses 300 word vectors, but this can be changed if desired. However, FastText

converts a word into a vector, and we have the text of the requirement as the initial data. The main idea of using FastText to transform a textual formulation of a requirement into a vector representation is that the requirement is cleared of all special characters and signs, tokenized and lemmatized [6], after which a list of tokens is fed to the FastText model input. Further, the values of the obtained set of vector representations of individual words of the requirement formulation are averaged.

The BERT model was announced and made public in 2018 [7]. Initially, the BERT model is trained on a huge amount of text taken from books, Wikipedia, etc., and in several languages. This technology is based on such algorithms developed by the NLP (natural language processing) community as learning with partial involvement of a teacher, ELMo models (Embeddings from Language Models) [8], ULMFiT (Universal Language Model Fine-Tuning) [9], OpenAI Transformer and others.

BERT is based on the Transformer, in fact, BERT is a trained stack of Transformer encoders. Encoders are objects of the same structure, but with different weights. Each encoder consists of two parts — an inner understanding layer and a propagation layer. The encoder receives as input numerical vectors obtained from words, processes the vectors at the level of the internal understanding layer and transmits them to the feedforward neural network. The result of the output of one encoder level is the input for the next encoder. Encoders form a stack of 12 (24 for deeply trained models) elements, an encoder stack, and is the backbone of BERT, as mentioned. Each sent word in BERT goes through a chain of encoders and as a result, for each word, a numeric vector of dimension 768 is formed for the base BERT implementation. The obtained vectors are the required ones in the framework of this problem. Further, it is possible to perform various operations with them: cluster, use as input data in supervised machine learning algorithms, for example, for a logistic regression problem.

The customFeature method is based on the use of the language features of the requirements statement. How can an atomic requirement differ from a non-atomic requirement? The first thing that comes to mind is the size of the requirement [10]. It is logical that it is unlikely that a large requirement in most cases will be atomic. On the other hand, you can pay attention to the number of words in the requirement, since the number of characters alone cannot provide enough information. You can also count the number of offers in a request. In most cases, a requirement that consists of more than one clause is not atomic.

Next, it's worth remembering what an atomic requirement is and taking a closer look at the differences

between an atomic and a non-atomic requirement. A requirement is considered atomic if each statement of the requirement must be one element of the hierarchy of requirements, and the subsequent division of the proposal into two or more requirements is not possible. Consider the following examples.

- The service life of the phone must be at least 1 year.
- The service life of the phone and its individual parts must be at least 1 year.

Obviously, the second requirement is non-atomic, in contrast to the first, because it can be divided into two requirements: "The service life of the phone must be at least 1 year" and "The service life of individual parts of the phone must be at least 1 year". So what are the fundamental differences between them in terms of language features?

Let's take a look at the INCOSE Requirements Writing Guide [11]. It contains rules for statements and sets of requirements that will be very useful for the feature engineering process. It clearly states: avoid conjunctions in requirements. In the example of a non-atomic requirement, the conjunction "and" is just used, which means it is logical to make a sign that determines the total number of unions in the checked requirement.

In Russian, the enumeration can be specified not only with the help of conjunctions, but also with the use of punctuation marks. The presence of a large number of special characters may indicate a possible non-atomicity of the requirement, so it is worth counting either their total number, or each special character separately.

The next rule that should definitely be used to form a feature is to formulate a requirement in one sentence. Indeed, if it was necessary to split the text into two sentences, then it is worth thinking about splitting the requirement itself into two. Therefore, you should define a binary sign that signals the fulfillment of this rule.

In addition, the INCOSE guidance prescribes not to use brackets in the wording of requirements, not to indicate clarifying text in brackets. Often, information in brackets indicates information that is redundant, deprives the requirement of unambiguity, or contains enumerations. To determine whether the rule described above is followed, it is worth introducing a binary feature that takes the value of one if there are round, square or angle brackets in the requirement.

Also in the INCOSE manual it is written that when writing requirements, you should use the active voice. That is, the requirement "The identity of the client must be verified" is unacceptable. In a simple case, as in the example, the passive voice can be determined by the presence in the text of the requirement of the con-

struction "must be", but due to the peculiarities of the Russian language, one should also parse the sentence and check that the predicate is in the active voice. To check this rule, it is necessary to select a binary sign that signals the presence of a passive voice in a demand.

Another interesting rule is to avoid imprecise terms such as "several", "any", "acceptable", "many", etc. Adverbs are necessary to clarify an action, and not vice versa. In addition, you should not use ambiguous adjectives, the use of which introduces ambiguity: effective, relevant, sufficient, adequate, etc. To implement this rule, you can enter a list of inaccurate words and for each requirement calculate the proportion of such words.

Also, in the formulation of requirements, it is necessary to avoid using the particle "not". Consider an example of a requirement: "The system must not release hazardous substances into the atmosphere." The problem with the formulation of this requirement is that it is impossible to verify its implementation in a finite time. The existence of this rule leads us to the formation of a binary feature that determines the presence of the "not" particle in the checked requirement.

The presence of a slash "/" in the statement of requirements is also undesirable. A typical example — and/or, allows for ambiguity in interpretation. An exception is the forward slash in SI units, for example, km/h, m/s.

Another interesting rule is to avoid pronouns. The use of pronouns in order to avoid repetitions in the formulation of requirements is unacceptable, since it entails the appearance of ambiguity, and therefore problems when verifying the implemented requirement. This rule can be interpreted as a quantitative feature, the value of which will be the proportion of pronouns in the wording of the requirement under consideration.

In addition, absolute phrases that denote an unattainable goal should not be used in the statement of requirements. The indicators of non-fulfillment of this rule can be the presence in the formulation of the requirement of such words as "all", "always", "never", etc. Such requirements cannot be verified. Hence, a quantitative feature appears, which will be equal to the fraction of words of absolute revolutions in the formulation of the requirement under consideration.

Thus, the customFeature feature space is formed, onto which you can map any text formulation of the requirement and obtain a vector from this feature space.

Atomicity check

In the general case, the classification problem can be described as follows: there is a certain sample of objects for which the belonging to the class is determined by labels [12]. The set of classes is finite and known.

It is required to implement an algorithm that will be able to determine the class of the entire set of objects without labels.

The problem of determining the atomicity of a requirement formulation can be formulated as a classification problem: the set of classes is represented by two elements: atomic, not atomic. Adapting the formulation of the classification problem to the problem under consideration, we obtain the following statement: it is necessary to build an algorithm that will perform the binary classification of the requirement.

Since the classification problem is a special case of supervised learning, it becomes necessary to compose and mark up a training sample. For this, a requirements specification of 287 elements was formed and marked up. The ratio of elements of different classes in the sample is 43 to 57, which makes it possible to assert that it is balanced. The specification of requirements was divided in a ratio of 3 to 7 — into test and training samples, respectively.

Let's take a closer look at gradient boosting. It is a composition of N basic algorithms, constructed into a composition [13]. They are built sequentially, one after the other, so that the i -th algorithm corrects the composition errors of the previous $i-1$ pieces. Because of this method of building a composition, it becomes possible to use relatively simple basic algorithms, for example, shallow trees. Sometimes a constant classifier can be taken as the first basic algorithm, which assigns all objects to one class.

Let us consider in order all variants of combinations of classifiers and requirements preprocessing. Let's start with a logistic classifier [14] with various options for preprocessing the initial data in text format.

As an evaluation metrics we will use Accuracy and the area under the ROC curve [15].

Figure 1 shows the results of evaluating the logistic classifier and customFeature.

The figure shows that the logistic classifier trained on the requirements converted into vectors using customFeature showed an accuracy of 86 % on the test sample, and the area under the ROC curve was 0.853. Next, consider the results obtained for the logistic classifier with FastText preprocessing (fig. 2).

The results turned out to be much more modest than those of the previous data preprocessing method. Only 75 % accuracy and 0.741 area under the AUC ROC curve. The next in line will be a logistic classifier using BERT. The results of the assessment are shown in fig. 3.

It can be seen that the accuracy when using BERT is comparable to the first option, 83 %. The area under the ROC curve is 0.832.

Next, consider classifiers based on decision trees. Let's start by combining the decision tree and the customFeature (fig. 4).

	precision	recall	f1-score	support
0	0.84	0.91	0.87	45
1	0.89	0.79	0.84	39
accuracy			0.86	84
macro avg	0.86	0.85	0.86	84
weighted avg	0.86	0.86	0.86	84

ROC AUC: 0.8529914529914528

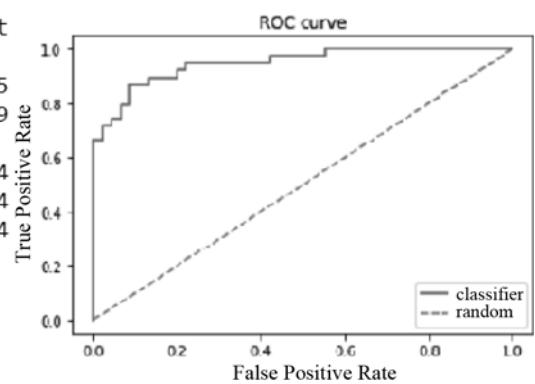


Fig. 1. Logistic + customFeature Results

	precision	recall	f1-score	support
0	0.72	0.87	0.79	45
1	0.80	0.62	0.70	39
accuracy			0.75	84
macro avg	0.76	0.74	0.74	84
weighted avg	0.76	0.75	0.75	84

ROC AUC: 0.7410256410256411

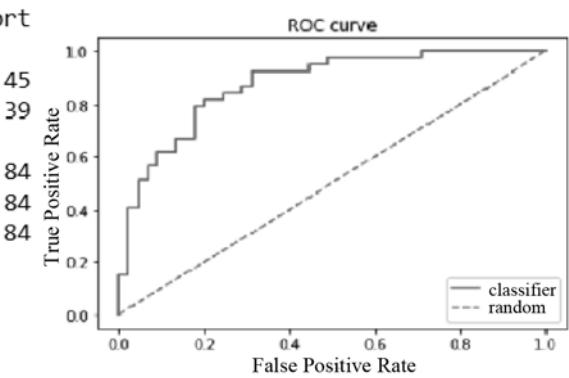


Fig. 2. Results of Logistic + FastText

	precision	recall	f1-score	support
0	0.84	0.84	0.84	45
1	0.82	0.82	0.82	39
accuracy			0.83	84
macro avg	0.83	0.83	0.83	84
weighted avg	0.83	0.83	0.83	84

ROC AUC: 0.8324786324786325

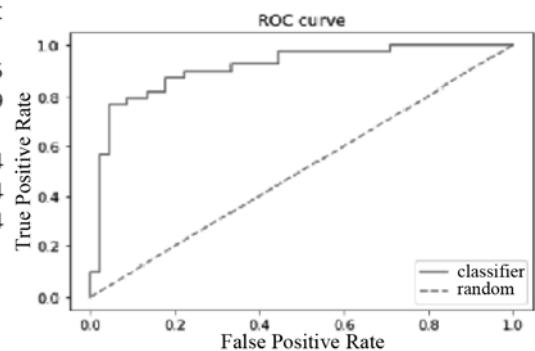


Fig. 3. Results of Logistic + BERT

	precision	recall	f1-score	support
0	0.83	0.87	0.85	45
1	0.84	0.79	0.82	39
accuracy			0.83	84
macro avg	0.83	0.83	0.83	84
weighted avg	0.83	0.83	0.83	84

ROC AUC: 0.8307692307692308

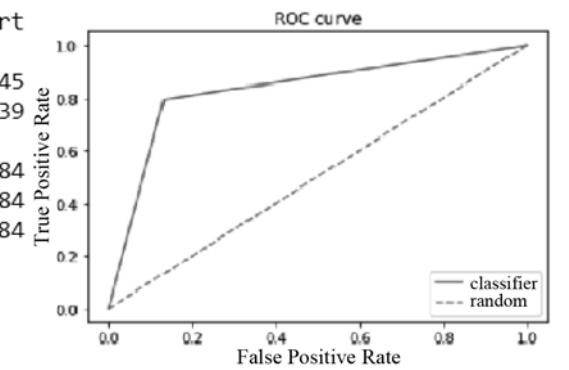


Fig. 4. Decision Tree + customFeature Results

The decision tree using customFeature showed a slightly more modest result than the logistic classifier: the accuracy is 83 % and the area under the ROC curve is 0.831. Let's move on to the next combination - decision tree and FastText. The results are shown in fig. 5.

The result is the worst result among all those tested. Only 69 % accuracy and an area under the ROC curve is 0.684. Next is the decision tree and BERT (fig. 6).

This combination received an accuracy rating of 76 % and an area under the ROC curve of 0.764. At the moment, we would like to recall the peculiarity of decision trees to unnecessarily adjust to the training set, and as a result, have poor generalizing ability, which we saw in

the last two combinations due to the large dimension of the vector space (300 for FastText and 768 for BERT).

Let's move on to considering a random forest. Let's start with the customFeature for preprocessing the initial data (fig. 7).

So far we get the highest scores: 88 % accuracy and 0.875 area under the ROC curve. Let's look at more complex preprocessing methods, first FastText. The results are shown in fig. 8.

Again we run into rather low results when using fastText: 80 % accuracy and 0.784 area under the ROC curve. Next up is a random forest combined with BERT. The results are shown in fig. 9.

	precision	recall	f1-score	support
0	0.69	0.78	0.73	45
1	0.70	0.59	0.64	39
accuracy			0.69	84
macro avg	0.69	0.68	0.68	84
weighted avg	0.69	0.69	0.69	84
ROC AUC:	0.6837606837606839			

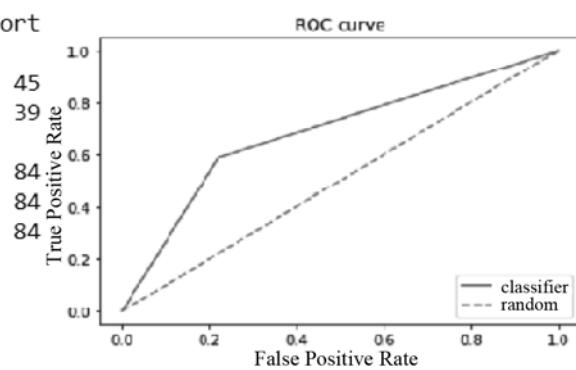


Fig. 5. Decision Tree + FastText Results

	precision	recall	f1-score	support
0	0.80	0.73	0.77	45
1	0.72	0.79	0.76	39
accuracy			0.76	84
macro avg	0.76	0.76	0.76	84
weighted avg	0.77	0.76	0.76	84
ROC AUC:	0.764102564102564			

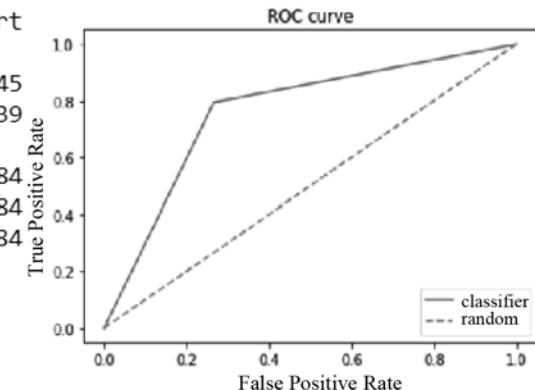


Fig. 6. Results of Decision Tree + BERT

	precision	recall	f1-score	support
0	0.84	0.96	0.90	45
1	0.94	0.79	0.86	39
accuracy			0.88	84
macro avg	0.89	0.88	0.88	84
weighted avg	0.89	0.88	0.88	84
ROC AUC:	0.8752136752136751			

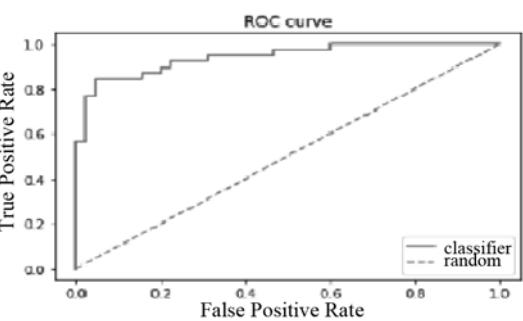


Fig. 7. Random Forest + customFeature Results

	precision	recall	f1-score	support
0	0.73	0.98	0.84	45
1	0.96	0.59	0.73	39
accuracy			0.80	84
macro avg	0.85	0.78	0.78	84
weighted avg	0.84	0.80	0.79	84
ROC AUC:	0.7837606837606839			

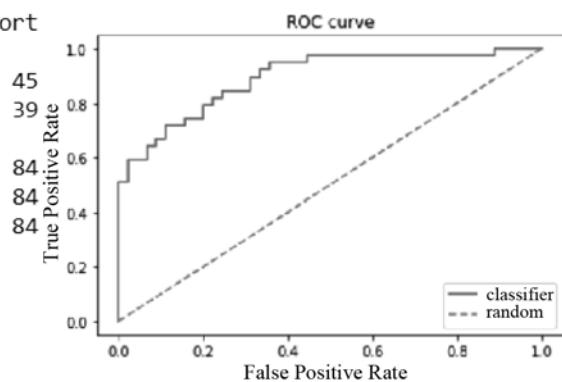


Fig. 8. Results of Random Forest + FastText

	precision	recall	f1-score	support
0	0.83	0.96	0.89	45
1	0.94	0.77	0.85	39
accuracy			0.87	84
macro avg	0.88	0.86	0.87	84
weighted avg	0.88	0.87	0.87	84
ROC AUC:	0.8623931623931624			

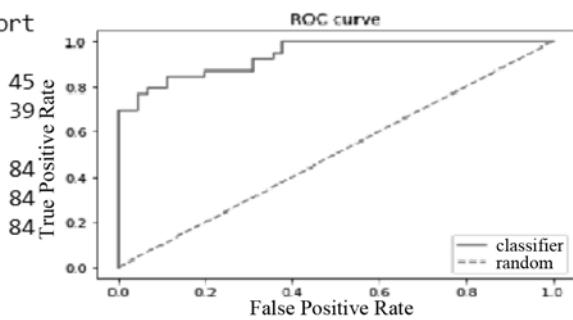


Fig. 9. Results of Random Forest + BERT

Random forest in combination with BERT showed fairly high scores: 87 % accuracy and an area under the ROC curve of 0.862. Continuing the consideration of the results, the next step is a classifier based on gradient boosting over decision trees. First, let's take a look at the customFeature results (fig. 10).

Again we see excellent results for customFeature: 88 % accuracy combined with 0.874 area under the ROC curve. The next step is the results of preprocessing using FastText (fig. 11).

The FastText results are also modest this time against the background of other data preprocessing methods: an accuracy of 81 % and an area under the ROC curve of 0.803. Let's consider the last option — gradient boosting over decision trees in combination with BERT. The results of this option are shown in fig. 12.

The last method of checking the requirements for atomicity showed the following results: an accuracy of 83 % and an area under the ROC curve of 0.829.

	precision	recall	f1-score	support
0	0.83	0.98	0.90	45
1	0.97	0.77	0.86	39
accuracy			0.88	84
macro avg	0.90	0.87	0.88	84
weighted avg	0.89	0.88	0.88	84
ROC AUC:	0.8735042735042734			

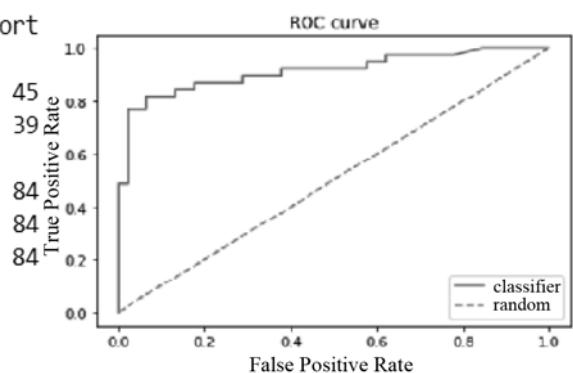


Fig. 10. Gradient Boosting + customFeature Results

	precision	recall	f1-score	support
0	0.78	0.89	0.83	45
1	0.85	0.72	0.78	39
accuracy			0.81	84
macro avg	0.82	0.80	0.81	84
weighted avg	0.81	0.81	0.81	84
ROC AUC:	0.8034188034188035			

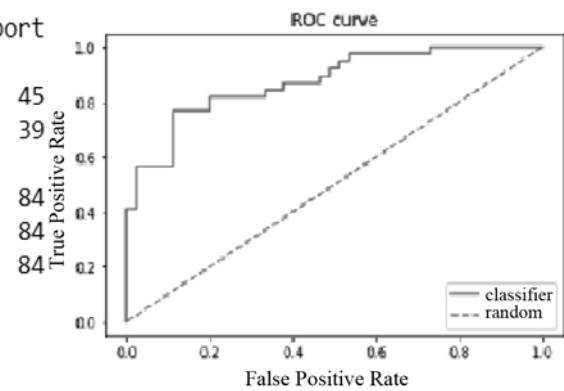


Fig. 11. Gradient Boosting + FastText Results

	precision	recall	f1-score	support
0	0.82	0.89	0.85	45
1	0.86	0.77	0.81	39
accuracy			0.83	84
macro avg	0.84	0.83	0.83	84
weighted avg	0.84	0.83	0.83	84
ROC AUC:	0.8290598290598289			

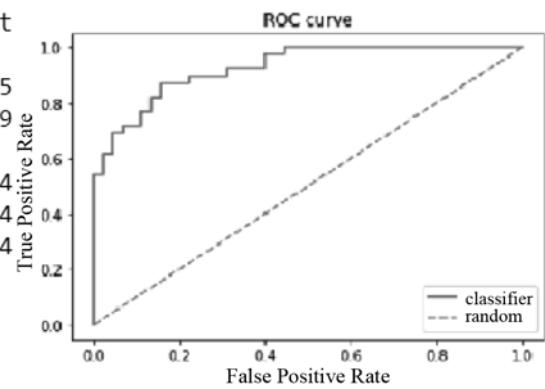


Fig. 12. Gradient Boosting + BERT Results

General results of combinations of requirements verification methods

Learning method	customFeature		Fast Text		BERT	
	Accuracy, %	ROC	Accuracy, %	ROC	Accuracy, %	ROC
Logistic classifier	86	0.853	75	0.741	83	0.832
Decision tree	83	0.831	69	0.684	76	0.764
Random forest	88	0.875	80	0.784	87	0.862
Gradient boosting	88	0.874	81	0.803	83	0.829

Conclusion

To summarize, what is the best combination, what is better to use when checking sets of requirements for atomicity? Consider the results of all combinations of methods in table.

Comparison of the results obtained when testing combinations of data preprocessing methods and algorithms for classifying requirements for atomicity allow us to draw several conclusions:

Using FastText as a way to vectorize a requirement statement has proven to be the least accurate option.

BERT's accuracy is comparable to customFeature using any of the possible classification methods.

Maximum accuracy of 88 % is achieved when using customFeature in combination with random forest or gradient boosting.

The results obtained allow us to make the assumption that models based on a small number of meaningful features, which are important precisely in determining the atomicity of requirements when classifying requirements, turn out to be no worse than those that take into account the semantics of the proposal (customFeature in comparison with BERT).

Thus, the best way to classify requirements for atomicity is gradient boosting over decision trees using the customFeature vector representation of requirements.

The result obtained can be used to create systems for automatic quality control of requirements. The requirements engineer will be able to get quick feedback on the quality of the requirements and not have to wait for the requirements review. The use of machine learning allows to avoid setting up rules for checking the quality of requirements. Instead, it is enough to have a labeled dataset.

References

1. Chatzipetrou P., Unterkalmsteiner M., Gorscheck T. Requirements' Characteristics: How do they Impact on Project Budget in a Systems Engineering Context? *2019 45th Euromicro Conference on Software Engineering and Advanced Applications (SEAA)*, 2019 Aug. 28, IEEE, 2019, pp. 260–267.
2. Batovrin V., Gaydamaka K. Requirements engineering – key factor for project success, *The Project Management Journal*, 2017, no. 1 (49), pp. 6–20.
3. Hall E. *Requirements Engineering*, US, Kent, Gray Publishing, 2005, 239 p.
4. Muhamedyev R. Machine learning methods: An overview, *Computer modelling & new technologies*, 2015, vol. 19, no. 6, pp. 14–29.
5. Joulin A., Grave E., Bojanowski P., Douze M., Jégou H., Mikolov T. Fasttext. zip: Compressing text classification models. 2016. arXiv preprint arXiv:1612.03651.
6. Chowdhury G. G. Natural language processing, *Annual review of information science and technology*, 2003, vol. 37, no. 1, pp. 51–89.
7. Devlin J., Chang M. W., Lee K., Toutanova K. Bert: Pre-training of deep bidirectional transformers for language understanding. 2018 Oct 11. arXiv preprint arXiv:1810.04805.
8. Ethayarajh K. How contextual are contextualized word representations? comparing the geometry of BERT, ELMo, and GPT-2 embeddings. 2019. arXiv preprint arXiv:1909.00512.
9. Howard J., Ruder S. Universal language model fine-tuning for text classification. 2018. arXiv preprint arXiv:1801.06146.
10. Génova G., Fuentes J. M., Llorens J., Hurtado O., Moreno V. A framework to measure and improve the quality of textual requirements, *Requirements Engineering*, 2013, vol. 18, iss. 1, pp. 25–41, DOI: 10.1007/s00766-011-0134-z.
11. INCOSE, Guide for writing requirements INCOSE TP-2010-006-02, USA, San Diego, International Council on Systems Engineering, 2015, 73 p.
12. Kotsiantis S. B., Zaharakis I., Pintelas P. Supervised machine learning: A review of classification techniques, *Emerging artificial intelligence applications in computer engineering*, 2007, vol. 160, no. 1, pp. 3–24.
13. Ke G., Meng Q., Finley T. et al. Lightgbm: A highly efficient gradient boosting decision tree, *Advances in neural information processing systems* 30, 2017, pp. 3146–3154.
14. Nasteski V. An overview of the supervised machine learning methods, *Horizons*, 2017, vol. 4, pp. 51–62.
15. Hossin M., Sulaiman M. N. A review on evaluation metrics for data classification evaluations. *International journal of data mining & knowledge management process*, 2015, vol. 5, no. 2, pp. 1–11. DOI: 10.5121/ijdkp.2015.5201.

ИНФОРМАЦИЯ



4–6 октября 2022 г. в Санкт-Петербурге
на базе АО "Концерн" ЦНИИ "Электроприбор" состоится
15-я мультиконференция по проблемам управления
(МКПУ-2022)

Мультиконференция включает пять локальных конференций:

- ♦ XXXIII конференция памяти выдающегося конструктора гироскопических приборов Н. Н. Острякова
- ♦ Конференция "Информационные технологии в управлении" (ИТУ-2022)
- ♦ Конференция "Математическая теория управления и ее приложения" (МТУиП-2022)
- ♦ Конференция "Управление в аэрокосмических системах" имени академика Е. А. Микрина (УАКС-2022)
- ♦ Конференция "Управление в морских системах" (УМС-2022)

В рамках мультиконференции пройдет Семинар по закрытой тематике

Информация для связи:

ГНЦ РФ АО "Концерн" ЦНИИ "Электроприбор",
Тел.: +7 (812) 499-82-10 — Истомина Елена Анатольевна,
+7 (812) 499 82 67 — Тарановский Дмитрий Олегович,
E-mail: icins@eprib.ru

Overview of Face Recognition Algorithms for Person Identification¹

D. V. Alexandrov, dmalex-m2@yandex.ru, School of Software Engineering,
National Research University Higher School of Economics, Moscow, 109028, Russian Federation,
Department of Corporate Information Systems "MIREA — Russian Technological University", Moscow,
119454, Russian Federation

Corresponding author:

Dmitriy V. Alexandrov, Professor, National Research University Higher School of Economics, Moscow,
109028, Russian Federation, MIREA — Russian Technological University, Moscow, 119454, Russian Federation
E-mail: dmalex-m2@yandex.ru

Received on July 13, 2021

Accepted on March 28, 2022

Trends in computer vision and pattern recognition and capabilities of modern computers contributed to a considerable amount of research of these areas application in facial recognition systems. The purpose of this paper is to investigate the most significant methods of face recognition. In the first two sections of current paper, the methods of face recognition and identification are presented. The analysis of these methods covers the most important features of the pattern recognition area. An application of groups of methods is considered for different purposes. This paper contains comments for capabilities of algorithms under observation. The third section reveals result of the algorithms testing using real-world datasets and examples.

Keywords: person identification, face recognition, neural networks, machine learning

For citation:

Alexandrov D. V. Overview of Face Recognition Algorithms for Person Identification, *Programmnaya Ingeneria*, 2022, vol. 13, no. 7, pp. 331—243.

УДК 004.93'1

Д. В. Александров, д-р техн. наук, проф., dmalex-m2@yandex.ru,
НИУ ВШЭ, РТУ МИРЭА

Обзор алгоритмов распознавания лиц для идентификации личности

Тенденции в областях компьютерного зрения и распознавания образов, а также возможности современных компьютеров способствовали значительному объему прикладных исследований в сфере создания систем распознавания лиц. В статье представлено исследование наиболее значимых методов распознавания лиц. В первых двух разделах статьи представлены методы распознавания и идентификации лиц. Анализ этих методов охватывает наиболее важные особенности сферы распознавания образов. Рассмотрено также применение групп методов для различных целей. Даны комментарии по применимости используемых алгоритмов. В разд. 3 представлены результаты тестирования алгоритмов с использованием реальных наборов данных и примеров.

Ключевые слова: идентификация личности, распознавание лиц, нейронные сети, машинное обучение

¹ The article is based on the materials of the report at the Seventh International Conference "Actual problems of Systems and Software Engineering" APSSE 2021.

Introduction

Since the beginning of time people were aware of keeping their goods safe, and that is the main reason for the invention of locks. Through thousands of years people were changing the mechanism of door locks and standalone locks in order to create a better construction that will not let people without a valid token — usually, a key, — get any further. For some time, people were only saving money under lock and key. Treasures or other worthy goods could be traded or exchanged for making a living. Shortly afterwards people decided to keep their passport, personal mail, under lock and key, bank clerks started locking their records — since the moment information found a value.

Previously it was possible and convenient to present information only on paper, or a close substitute. The situation remained unchanged until the end of 19th century, when technical revolutions led to various important inventions, like the phone and the telegraph, which consequently gave birth to the first electronic computer. Since the creation of the first portable personal computer the market has grown and changes at a staggering speed — the quantity of information, that could be stored on a hard drive, increased from 240 bytes to 4–5 terabytes, which is more than around 10^{11} times. However, the quality of the information changed as well — the first computers stored only numbers to make calculations, nowadays it stores photos, videos, documents, programs and much more. The first electronic password was implemented by MIT (Massachusetts Institute of Technology) researchers in the 1960's, when there was a need to distribute files between different users, so that each file had a ‘lock’, that could be opened with a ‘key’ — password. Soon it evolved in a huge variety of alternatives — 2-factor password verification, physical USB key, graphical representations instead of numbers etc. These alternatives have no leader as all of them have vulnerabilities — if a thief accesses the phone of his victim, two factor authorization process does not work, password or finger drawings are easy to steal. The first idea of an application of human identity verification in electronic systems came from police methods — fingerprints. Soon scientists started researching methods of implementation of various biometric characteristics for identification.

A reasonable impetus to development of biometric identification systems gave the distribution of modern smartphones with its capabilities. Fingerprint sensors are now built-in and being used for identification, purchases and many other purposes in most of middle-class or higher smartphones, as well as a frontal camera, which in most cases permits the production of a high-resolution photo of the person who is holding the phone.

However, facial recognition systems that are in the market today are often a subject to hacking as there are various methods of corrupting the systems. For example, modern Facial Recognition System (FRS) should distinguish the person who is holding the phone and the picture that is being held below the camera. In order to replace current authorization methods with FRS it is crucial to obtain the same or better error rate, which is very low in the case of standard verification systems, having, despite that fact different problems, it is easy to forget the password or, in case of low password complexity — to crack it.

Modern FRS use different types of neural networks to accomplish their aim. It is interesting that most of theoretical basics, needed to build such systems, were presented in a previous century but they could not find their application in wide areas until recently due to the lack of technology and computational power. Since that moment some new areas were defined such as computer vision, machine vision, pattern recognition or graphics visualization, all of them have different aims and problem solving methods, which are, nevertheless, mostly based on an image process and some other data, that is being gathered by different sensors.

One of the most recent and promising methods of identifying a person is called a 3D recognition. In existing solutions an identification system analyses position of eyes, lips, ears, jaw and nose shape and distance ratio between these objects; some of them also count complex eye characteristics — eyes and eyebrow shape and color. In case of 3D recognition the contour of eye sockets, nose and chin is also being analyzed. An important benefit of the 3D method is that it can recognize the face from different angles. Another advantage is stabilization of light unlike other methods. In the case of a badly lit environment, attempts to recognize a face, are very likely to have a lower precision rate due to possible shadow presence, which always spoils calculations.

It is vital to mention, that the above-described advantages of a 3D recognition method can be achieved only in cases of using either three different cameras in an array, or using specific sensors for obtaining a 3D model of the head. Even though in terms of this work it will not include any of these hardware methods of facial recognition, multi-dimensional models are still going to be judged in order to verify if they still can be useful for facial recognition.

Based on an experience of existing solutions development it is fair to expect, that in most cases it is going to be used in combination with different methods in order to achieve a considerable result. This approach is now being used in a huge number of distributed systems to increase an accuracy of problem solving and to prevent system break-in. As it was mentioned before, it is crucial

to realize, that the system should not only identify a person by his face, but also should be aware of a possible corruption with different existing techniques.

The goal of current research is to implement a facial recognition module for person identification by using the most appropriate stack of technologies and the most efficient and suitable algorithm of pattern recognition available.

The above aim will be accomplished by fulfilling the following research objectives:

- to observe existing methods of solving the face recognition problem and lying underneath theoretical basics;
- to implement the most relevant and accurate ones;
- to test it in a real-world circumstance;
- to select the most appropriate one or few of them and to build the FRS on top of it.

This paper consists of an introduction, three parts and a conclusion. The introduction reveals the relevance, an object, purpose and methods of research, reveals the theoretical and practical significance of the work. The first and the second parts deal with theoretical basics of facial recognition patterns, they describe an analysis of existing methods, and results of testing each method on real world examples. The third part is devoted to technology stack and program implementation itself, it mostly consists of practical results, obtained through the process of program development. In conclusion the results of study are summarized and final conclusions on the topic under consideration are formed.

1. Face recognition basics

Face recognition is a challenging problem in the field of computer vision and image processing. Through the history of technology development various algorithms were proposed and implemented as classifiers, that provide a sufficient precision rate and a high potential for a wide area of tasks.

The process of face recognition consists of multiple steps, that the system needs to go through to accomplish its aim:

- getting the image/video source;
- face detection;
- face normalization;
- feature extraction;
- descriptor feature matching;
- face identification.

For each step different methods of solving the problem may be used. Some older FRS did not follow exact steps, for example they could analyze the whole picture instead of an extracted frame with detected face. It could significantly increase execution time and accuracy.

Firstly, it needs to obtain a media file containing an object, that is going to be elaborated on. Media file may represent either a local video or image file stored in typical containers (png, jpg, mp4, flv) or a live stream from a camera, which is usually also represented in one of a wide known standards (RTSP, RTMP, HLS). Once the file or stream is acquired, the system tries to detect the face on each frame of presented media file. It may be mentioned that simple algorithms of face detection are not tied to specifications of an exact object "face", thus it can also recognize any kind of object — license plates, pedestrians, cars. More complicated groups of algorithms may pay attention to inner features of human behavioral — emotions, speaking qualities.

In order to detect a face, the system usually uses genetic algorithm and Eigenface technology [1—4]. The genetic algorithm is a search method based on the concept of natural evolution. In such an algorithm each chromosome (individual) is a possible solution to a given problem. Each individual chromosome may mate with others in case of a good fit during one generation. Thus, the least fitting individuals disappear from the population because no other chromosomes are mating with them. Chromosomes that survive through one generation then breed a new generation of individuals by crossing-over and mutating. The process is iterative, and the number of generations varies depending on the quality of the result needed. At the end of the process, it is expected to obtain an optimal solution to the problem. In case of detecting a face, genetic algorithm may be applied by the following steps:

- normalize an image (change the range of pixel intensity values);
- produce RGB extraction to obtain a gray-scale image;
- perform preprocessing and encode a sub-window of an initial image;
- using generated chromosomes, perform a search on an obtained sub-view using Eigenface (the template of human face) being rotated and moved around the given frame;
- applying a genetic algorithm to calculate parameters of template position on a given frame (center, scaling, rotation and match rate);
- in case of low fitness values — proceed to the next iteration of the loop and continue with a next sub-window.

Fitness function is a function of the difference between the intensity value of the input image and that of the template image measured for an expected chromosome value. Fitness function is defined as:

$$f(n) = 1 - \frac{\sum_{(x,y) \in W} |f(x,y) - f_{n,t}(x,y)|}{B_{\max} \times xSize \times ySize}, \quad (1)$$

where B_{\max} is a maximal brightness of an image; $xSize$ and $ySize$ are width and height of the template image; f and $f_{n,t}$ are the intensity values justified for the n -th position of the chromosome for an initial image and template respectively.

On a good match an algorithm stops and fixes a possible face region. It may be represented as an oval but usually the smallest frame that contain the face is being used. Genetic algorithm flow includes the following steps.

- Getting the original color image.
- Preprocessing.
- Image enhancement.
- Filtering.
- Generic searching.
- Creating the face region.

Modern facial detection systems first detect possible human eye regions. Then using the genetic algorithm system, they generate all possible face regions which should include eyebrows, nose and mouth comers. Each sub-window of an initial image is normalized to avoid the shirring effect due to head movement or illumination effect. The fitness value of each frame is measured based on its projection on Eigenfaces.

To create a set of Eigenfaces (template of human face) it is necessary to:

- using a training set of face images where eyes and mouths should be aligned, lighting conditions should be similar and one resolution value is specified, create an image matrix where each column of the matrix is an image;
- subtract the mean;
- calculate eigenvalues and eigenvectors of the covariance matrix, which can also be called eigenfaces due to the possibilities presented as an image;
- choose principal components by sorting eigenvectors in descending order of its eigenvalues λ_i and setting a threshold ε on the total variance of $v = (\lambda_1 + \lambda_2 + \dots + \lambda_n)$ and evaluate the number of principal components k which is the smallest number that satisfies to:

$$\frac{(\lambda_1 + \lambda_2 + \dots + \lambda_k)}{v} > \varepsilon, \quad (2)$$

where n is a number of components.

Eigenfaces may represent both existing and new faces. It allows to calculate the distance of a new face from the mean face. Eigenvalues represent the scale of difference between the mean image and the current eigenvector value. The information is lost when the image is projected on a subset of eigenvectors. However, selecting eigenvectors with the largest eigenvalues discards most of the available options and will grant a desirable output. Eigenfaces is based on the Principal Component

Analysis (PCA) and remains one of the most robust and applicable methods of face detection [1, 4, 6].

Another possible option to identifying a face is the Fisherfaces algorithm, which is based on Linear Discriminant Analysis (LDA). Its usage may be efficient in cases of problems of recognition of faces under different illumination. Instead of presenting the whole variety of possible options with given weights (eigenfaces) Fisherfaces is trying to lower the distance between inner-class samples of data and maximize — between different classes, for example two images of one person under different lighting conditions or in a different pose may differ more than pictures of different people made under the same conditions. LDA may be used in order not only to detect a face but also to recognize it, meanwhile PCA has a poor performance in face recognition.

After feature extraction the system prepares the recognition stage and matches with descriptor data. From this point of flow, it depends on the aim of the system, in cases of trying to detect emotion on a human's face the system will analyze existing datasets of people with this emotion. In cases of face recognition, it will proceed to the whole trained data in order to find the closest match. There are various databases that can be used to train and test the facial recognition system with datasets containing various numbers of people, emotions and conditions affecting pictures.

Obviously, the facial recognition system may be constructed on a different flow or algorithms. However, while no new algorithm is proposed, the described scheme remains one of the most efficient, due to possibility to extract a module and replace it with another one. Some simple FRSs use only one method to detect and recognize a face, some complicated ones use a huge variety of different methods depending on a situation and aim of the system.

2. Face recognition methods

Keen interest to the technology of face recognition brought a huge variety of different algorithms of solving the problem. Some groups of methods, that are going to be observed below may appear as a key technique for the whole process of FRS flow, e. g., former systems used only PCA to detect and identify a face on a given picture or, inversely, modern systems may utilize mostly 3D based algorithms to construct an object, measure it and compare it with another one. Algorithms are going to be classified in groups by main feature of analysis core.

2.1. Classical methods of pattern recognition

This group of methods was named due to using mostly traditional approaches of image analysis and face detection and recognition. It may be divided into

methods utilizing local features and methods utilizing holistic features, that can also be subdivided into linear and nonlinear methods.

Linear projection appearance-based methods have proved themselves as a good solution in cases of a "perfect" situation — all objects are accurately aligned; the illumination influence is minimal and facial emotions should not be presented. The main explanation for such a bad performance is that faces are projected on a complex nonlinear manifold in high dimensional space, thus the algorithms cannot efficiently extract desired features. Principal component analysis, linear discriminant analysis and linear regression classifier are the most well-known algorithms of classical group methods.

2.1.1. Linear holistic methods

2.1.1.1. PCA (Principal Component Analysis)

One of the eldest algorithms initially proposed by Karl Pearson in 1901, is being used in a wide variety of areas including data compression, econometrics and bioinformatics. The mean is being subtracted, covariance matrix on its base constructed and eigenvectors and eigenvalues are calculated. In general, main components are eigenvectors sorted by descending eigenvalues. Eigenvectors, that are above a specified threshold form a feature vector. This vector is multiplied by the initial dataset and transposition brings the result of PCA — new dataset that is constructed solely on vectors we have chosen. The main idea of PCA is to find the straight, that will ensure the lowest sum of squared distances to this straight from the corresponding points of an original dataset.

When applied to image processing, firstly, the image is being transformed from an N^2 dimensional matrix to a one-dimensional vector. This vector contains an intensity of grayscale image pixels row by row. To detect patterns in a set of images, a matrix of such vectors is constructed and PCA is applied. The data based on chosen eigenvectors allows to calculate the distance between a new image and the initial dataset in axes derived from PCA analysis [7].

2.1.1.2. LDA (Linear Discriminant Analysis)

The algorithm is based on Discriminant function analysis, which was originally developed by Sir Ronald Fisher in 1936 and may be named as a generalization of an original one. As well as PCA, it has a lot of applications — bankruptcy prediction, earth science, biomedical studies. The goal of an approach is to reduce the dimension of a dataset by projecting it on a less-dimensional subspace. First, d-dimensional mean vectors for different classes and scatter matrices are computed both for in-between class and within class. Then eigenvectors and eigenvalues of scatter matrices are evalu-

ated and sorted just like in PCA. A $n \times d$ -dimensional matrix of samples multiplied by an obtained matrix of k eigenvectors transform data onto the new subspace.

The main difference between PCA and LDA approaches, is that in the first one it is needed to build component axes that will maximize the variance whereas in the second one, component axes are being maximized to separate different classes and find similarity in single classes. In general, reducing the dimension of an initial dataset significantly reduces computational costs for a given classification task and meanwhile minimizes an error occurrence possibility in parameter estimation [8, 9].

2.1.1.3. LRC (Linear Regression Classifier)

Linear classifier is aiming to identify which class or group does the object belong to. Response vectors for each class are being constructed, and the vector y representing the image under observation is being evaluated via logistic regression as follows in (3),

$$y = \frac{e^{X_i \beta_i}}{1 + e^{X_i \beta_i}}, \quad (3)$$

where X_i represents a class-specific model — matrix of one-class dataset and β_i is a vector of regression parameters. The weight of each feature derives from the gradient descent method. The minimal distance between original and predicted response vector will point to the class which is the most likely the object belonging.

The main disadvantage of such an approach is that in typical FRS it is unpredictable, if an image dataset contains "perfect" samples or they may contain occluded data, e. g., in cases that 3 out of 4 parts of an image are shadowed, then by the majority vote, even if the last one contains perfectly distributed facial features, it is very likely to have face portions mixed with an occlusion. When applied in FRS, before using LRC image dataset is usually being cleaned for potentially contaminated pixels using other methods like PCA or other simple linear methods [10, 11].

2.1.1.4. ICA (Independent Component Analysis)

The method is analyzing the multi-dimensional data to find underlying factors — components, that are both statistically independent and non-Gaussian. It is mostly applied to solve the problem of "the cocktail party" to extract the noise from a signal and obtain a clean desired source. The covariance matrix for an image vector is calculated (4), where E is a prediction operator, factorization of analysis of X brings an equation on (5), where Δ is a diagonal real positive and F transforms the original data X into Z in such a way that components

of Z are statistically independent. The whitening operation performs the transformation of vector X into U (6), that will have a unit covariance matrix, where Φ is an orthonormal eigenvector matrix and Λ is a diagonal eigenvalue matrix, that come from equation (7). The rotation operation then consequently extracts sources by minimizing the mutual information, normalizing the operation, then evaluates final independent components [12].

$$\Sigma_X = E\{[X - E(X)][X - E(X)]'\}, \quad (4)$$

$$\Sigma_X = F\Delta F^t, \quad (5)$$

$$X = \Phi\Lambda^{1/2}U, \quad (6)$$

$$\Sigma_X = \Phi\Lambda\Phi^t. \quad (7)$$

ICA identifies the independent source components from their linear mixtures (the observables). ICA thus provides a more powerful data representation than PCA as its goal is that of providing an independent rather than uncorrelated image decomposition and representation.

2.1.1.5. 2DPCA (2D Principal Components Analysis)

2DPCA differs from its ancestor PCA with an approach of treating images like 2D matrices instead of creating vectors of the image. An image covariance matrix is being constructed using an original matrix and its eigenvectors are evaluated for further feature definition. It allows an extracted feature from an object to be classified. Firstly, it obtains a mean matrix between all objects presented in class and evaluates a scatter matrix. Next, it computes a projection subspace by constructing orthonormal vectors to corresponding largest eigenvalues of the scatter matrix. Finally, projecting images on each of the obtained vectors derives the principal components vectors. Computing the sum of Euclidian distances between principal component vectors and tested ones and specifying a threshold value allows us to get an object prediction. This method is computationally lower than PCA and brings the same accuracy rate of identifying objects [13].

Reconstructive approaches (such as PCA and ICA) are reported to be robust for the problem of contaminated pixels, whereas discriminative approaches (such as LDA) are known to yield better results in clean conditions. In order to deal with disadvantages of linear methods some nonlinear extensions were proposed — KPCA and KLDA [14].

2.1.2. Nonlinear holistic methods

2.1.2.1. KPCA (Kernel Principal Component Analysis)

The method preserves simplicity of PCA and uses kernel methods carrying standard linear operations of

PCA to reproduce kernel Hilbert space. The underlying idea is that it is almost always possible to linearly separate N points in $(d - 1) \geq N$ dimensions. This method grants a high level of accuracy, however, remains time-consuming to be implemented in real-world applications. There exist some methods that are involved in the process feature extraction in order to lower execution time costs.

2.1.2.2. KLDA (Kernel Linear Discriminant Analysis)

An approach also known as generalized discriminant analysis is also an extension of linear method LDA, which due to performing in a new feature space allows nonlinear mapping to be examined. Despite a strong theoretical validity KLDA, KPCA and other extensions over linear methods do not produce significant improvements when applied in the real-world environment. There are various methods based on GDA (General Discriminant Analysis) and being used for face recognition purposes, KLDA may be classified as one of them [14, 15].

2.1.2.3. LLE (Locally Linear Embedding)

LLE is a nonlinear dimension-reductive technique that maps data onto a single global coordinate system of lower dimension. Projected data is being analyzed to identify closest neighbors by any method (Euclidean distance, Pearson coefficients etc.). This approach shows its usefulness in case of applying it to data where objects are presented in a not-proper way, e. g. a half-turned face may be reconstructed as a fully presented one and successfully compared with a proper facial object.

Another approach called Locality Preserving Projection is behaving in the same way as LLE using slightly different methods, both are considered as possible solutions for training purposes, but questionable in case of projecting new data items [16].

Observed approaches above, both linear and nonlinear, may be attributed as holistic based. Below, local features-based groups of methods are going to be observed. The main difference is that features in the holistic-based groups of methods represent the optimal variances of pixel data, this data is used to identify humans, in case of local features-based methods — data, which is being analyzed, already contains facial feature content which has characteristics which allow us to identify one individual from another [17].

2.1.3. Local feature-based methods

2.1.3.1. LBP (Local Binary Patterns)

Unlike in previously observed groups of methods once the face on the image is detected, for each pixel an LBP is constructed and then the image is being divided into small regions from which LBP histograms

are extracted and formed into a single feature vector. This approach allows to identify people and to deal efficiently with facial emotional states. In further research, after initial proposition of an algorithm it has been determined that LBP combined with the HOG descriptor (Histogram of oriented gradients) in terms of facial recognition may significantly improve the computing time and precision rate [18, 19, 29].

2.1.3.2. iSVM (*improved Support Vector Machine*)

The performance of an ancestor — SVM, based on the kernel function is lacking in efficiency, it was proposed to enlarge a spatial resolution around the margin by a conformal mapping, and so the distance between classes is growing, increasing the identification precision rate. The algorithm preserves a good computational speed with a high precision rate while working with datasets containing occluded images [19, 20].

2.1.3.3. LDP (*Local Directional Pattern*)

LDP method is a texture pattern that differs from its close opponent LBP with a detailed analysis of features; by applying LBP it is only possible to find typical one-level objects like curves and edges, in case of LDP it analyzes different types of edges and curves therefore going deeper. It is possible due to usage of Kirsch masks and evaluating eight directions on each pixel. Hence, the analysis of data goes deeper, and can find some features of objects of one class. Each pixel goes through the procedure of applying a certain threshold on obtained direction numbers, transforms above values to 1, below — to 0 and evaluates the number that identifies the information about one pixel. After the LDP application the image is represented by an LDP descriptor (histogram).

The approach uses nearly the same algorithm for computing histograms that will represent sample data.

However, the possibilities of LDP are not limited with detection of low-level only objects, the method is able to work with contrast and those features observed mostly, that the human eye would have caught the first time [21].

2.1.3.4. SIFT (*Scale Invariant Feature Transform*)

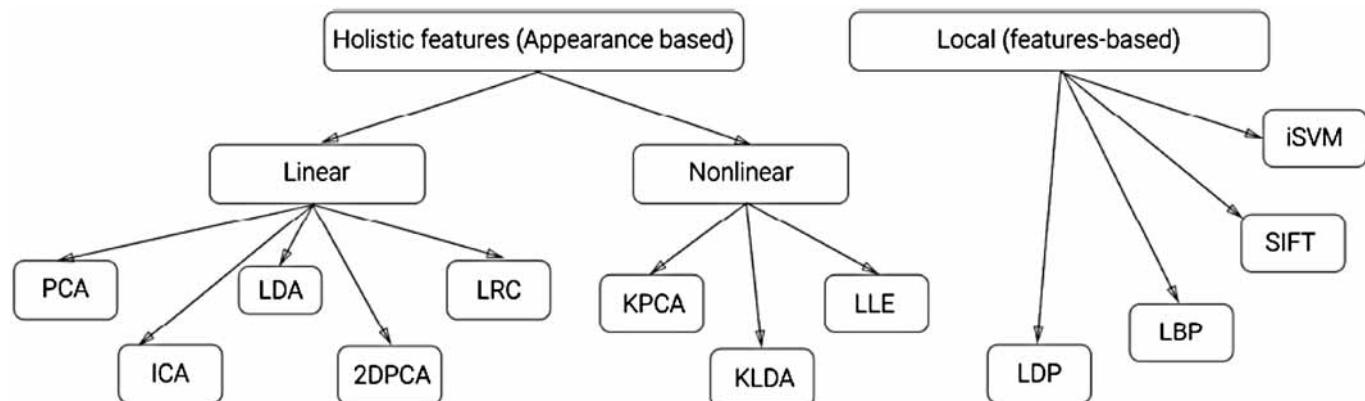
The method was initially proposed to extract some features from image objects, but soon was applied for object identification. SIFT does not pay attention to rotation and scaling and it also works well for pose and emotion invariants. This method is not computationally intensive and proves itself to be a good competitor to linear methods of face recognition [22, 23].

Thus, a number of classical face recognition algorithms were described (figure). The holistic (appearance-based) group include linear and nonlinear methods using slightly different steps of solving the problem, but still quite similar in terms of working with data. Local feature-based group differs in extracting feature descriptors and classifying them with multiple possible characteristics.

Classical methods are used in any kind of FRS in our days, but not only because of robustness of this group of methods but due to their stabilities and ability to obtain desired data while applying other algorithms just as one step of the process.

2.2. Applied methods

In the previous section algorithms and their execution steps were observed, in order to improve efficiency of different phases of face recognition — face detection, feature detection and extraction, analysis etc. — some additional methods may be used. In this section its underlying mechanics are described and their application in the real-world environment is considered.



Classical face recognition methods

2.2.1. Artificial Neural Networks (ANN)

Neural networks are being used widely in completely different problem-solving processes — its relevance may be explained by robustness and efficiency of neural network application. Particularly, in face recognition, neural networks are usually used to compare a recognized face from an image with the data stored on a database — the data which was used to train neural network. ANN is a type of neural network that uses "nodes" of three types — input, hidden and output nodes. Gathering initial data from input neurons and using the transfer function to apply NN's knowledge and passing values through neurons multiple times, in the end it reflects the meaningful in output nodes.

In order to significantly improve a precision rate of a neural network it is better to train it firstly on a 'clear' data — images that were taken in perfect conditions and are not spoiled with any kind of occlusion. Thus, obtaining a low sum-squared error neural network may further consume "noisy" images and still make high-rated predictions [6, 24].

One of ANN's applications is Self-Organizing Map (SOM), which the application usually considers for dimensionality reduction and feature extraction. It projects data to one or two-dimensional spaces placing the nearest points close to each other on a new space, therefore, providing partial invariance to rotation and image deformation. After the SOM application a soft nearest neighbor decision method may be applied in order to identify unlabeled subjects [26].

A multilayer perceptron (MLP) is a type of artificial neural network that maps an input data on a set of appropriate outputs, it consists of multiple layers, obligatorily connected to each other, and uses the supervised learning techniques, e. g. Back Propagation Algorithm (BPA). BPA, which is basically a method for training the weights in a multilayer neural network. Its principle is to model a given transfer function in the way that we expect the output to be presented. It is applied in the middle of the training process in order to correct obtained weights and improve the output of neural network nodes. Recently it was nearly replaced with more efficient Support Vector Machines, but now it is being under observation for further research due to successes in deep learning. The significant disadvantage of ANN application in facial recognition is a necessity of a solid amount of data to produce a satisfactory prediction [25].

2.2.2. Gabor wavelets

Gabor filters is one of the available methods to extract local features from an image. Mostly it can be applied on the step of image processing — to detect and represent features on an image by applying a filter — a function, which convolves an image to a Gabor representation of an image, it captures the properties of orientation selectivity and spatial localization, optimizing space localization and frequency domain. The drawback of this method is in high dimensionality of feature vectors that are extracted by Gabor filters. In order to solve this problem usually some typical methods of dimensionality reduction are applied — PCA, LDA or their descendants.

Gabor wavelets' characteristics are similar to human visual system characteristics; they are able to work with samples of one person presented in different pose or illumination. When dealing with face recognition in Gabor transformation needs to be used over 40 filters — typically 5 scales and 8 orientations.

This approach proves to be robust even when dealing with one or a small amount of sample data. However, selecting the most important list of features that were extracted by Gabor wavelets is a time-consuming and computationally intensive method, so it is not useful when creating a real-time system of face recognition. The Simplified Gabor Wavelets (SGW) proved their unstableness to pose and facial expressions changes, thus losing all meaning of applying such approach [27].

Elastic Bunch Graph Matching (EBGM) is one of feature-based algorithms that uses Gabor Wavelets transformation of each feature point obtained with standard methods. The method is based on dynamic link structures. A set of fiducial points on an image represents a fully connected graph and is labeled with the Gabor filters applied to a window around initial points. A set of graphs of fiducial points is combined in a stack structure, which is called a Face Bunch Graph. In order to recognize a new face image, its image graph is compared with sample data and the similarity is evaluated by the lowest distance between obtained values [5].

2.2.3. Face descriptor-based methods

Descriptor-based methods were already observed earlier, this approach is used in Local Binary Patterns, Scale Invariant Feature Transform and other species of image description methods. Unlike in other global-based methods where the entire image is observed for

features detection, in FDB methods, local features of the image are evaluated in neighbor pixels and then form a global description of the whole image. The image is being divided onto sub-images, each of them present a local feature histogram, which afterwards is being concatenated into a global descriptor which can identify an image and its local features.

Methods that use such an approach are in general, robust and competitive even to completely new techniques of face recognition. It may be used as well for facial expression changes or to identify some inner characteristic of an object under observation — e. g., skin color, nationality and psychological predisposition [28].

2.2.4. 3D methods

Three-dimensional facial recognition is obviously a more accurate and robust approach to solving the problem of person identification. It does not depend on the illumination changes, facial expressions or poses. Thus, the model of a person's face gives the rather appropriate estimation of a person's identity. This group of methods is more likely to develop further in the close future due to a necessity of human recognition in an uncomfortable context with a high precision rate. Modern algorithms cannot provide an acceptable efficiency yet.

There are several drawbacks to this approach. First, in order to obtain a desired quality of model construction and respective human identification, it is necessary to use specific hardware — sensors, specific cameras or just several usual cameras connected in an array. Second, in most FRS it is not possible to obtain an object's 3D model because an aim of the system is to identify a possible intruder for example. However, for specific needs it is an acceptable approach — e. g., for foreign passport issue, hardware is used to capture both 2D and 3D images of an object for different needs. A 2D image is for person identification by an employee and 3D is for FRS verification. Also, it may be used in the case when the person significantly changes his outlook — e. g., cuts his beard or hair.

Methods exist for obtaining a 3D model of a person without using specific hardware — by taking several pictures of an object from different angles and then through object reconstruction, obtaining the model, which can then be used for identification. In the fields of 3D model construction there are also 3D Generic Elastic Models (3D GEM) and 3D Morphable Model (3DMM) which can construct a model just from one single frontal image. Its algorithm consists of the approaches mentioned before but aimed specifically for 3D reconstruction. Hereby, PCA is being used for

minimizing the difference between principal components of 3D model's 2D acquired images of an object at different angles [30, 33].

2.2.5. Video-based methods

These groups of methods are the least explored in the field of face recognition and, nevertheless, one of the most potentially efficient [31]. Instead of a single image capture and its further analysis, video may grant a significant size array of images with a variety of object representations from different angles with different facial expressions and under different conditions [32]. An approach was not relevant until recent times due to a small number of cameras that can capture needed data of a desired quality. In the last few years, the amount of video content in the global network increased dramatically — people are publishing video content from their smartphone cameras, the number of cameras in cities is also significantly raised due to the concept of Smart city — safety reasons, control reasons, e. g., track how the municipal services do their job etc.

The K-nearest neighbors or Gaussian mixture model may be used to identify and compare obtained image sets from several video sources. The drawback of a modern perception of the video-based approach lies primarily in an incompleteness of possible data extracted from video sources. Instead of analyzing people's behavior, manners and emotional states, modern systems just create several images and compare them utilizing existing methods of face recognition.

One of the most recent developments in these respective areas permitted capture of some facial expressions, e. g., an eye blink. Development of technologies will continue to improve and raise a functional part of such recognition, because most modern facial recognition systems may be corrupted just with a printed image of another person.

The main concept of FRS may be represented by the following. The system should act like a person who knows every other person that he may meet in his daily routine. Modern technologies and, particularly, the Neural networks concept are consistently developing to the perception of a human's natural systems features. This approach may lead to a more-than-human precision rate of person identification but containing significantly more information about people. Capabilities of modern computers allow to store an enormous amount of data, which can be processed and analyzed, modern trends in Big Data are also aimed to solve the problem of large data parts analysis. The basic overview of methods that were observed in this section is presented in the table 1.

Table 1

Summary overview

Method	Advantages	Disadvantages
Classical face recognition algorithms	These methods project a face onto a linear subspace spanned by the eigenface images. The distance from face space is orthogonal to the plane of the mean image, so it may be easily turned to Mahalanobis distances with probabilistic interpretation	In case of varying degrees of luminosity of images represented in dataset it may lead to lack of a precision rate. The problem is that it is still unclear how to select the neighborhood size or assign optimal values for them
Artificial neural networks	A radial basis function for an artificial neural network is naturally integrated with a non-negative matrix factorization. Also, other approaches for process simplification regarding ANNs native linearization feature and computation speed up. It may be a perfect solution for FRS which will work with partially distorted objects	This group of methods require a large set of images in order to produce an adequate estimation. It is inaccurate in the same way like other statistically based methods
Gabor wavelets	Gabor wavelets work very well with spatial orientation issues. Various biometric applications are based on this approach	This approach requires the construction of a relatively high-dimensional Gabor feature space. Due to the high computational power needed it is not the best solution for a real-time. FRS Also, it is sensitive to changes in illumination
Face descriptor-based methods	This involves extraction and comparison of the most discriminant local features in order to minimize the difference between the same object and maximize between different people. These methods are discriminative and robust to illumination and expression changes. They offer compact, easy to extract and highly discriminative descriptors	These group of methods require a huge computational power while extracting descriptors. However, it is simple once this is done
3D-based facial recognition	This is significantly more accurate compared to the traditional 2D capturing process. It is independent to pose and illumination of the object, which makes this approach more robust	At the moment, it is not applicable to real-world applications due to the high computational requirements. It also requires very strict calibration in order to get a viable output
Video-based recognition	The potential and main advantage of this approach is the possibility of capturing more information from a video frame, choosing a more appropriate image set, catching human actions — blinks etc.	Video-based recognition is not fully investigated, but it has a high potential for developing real-time applications. It needs to implement other groups of methods to obtain the distance between images and measure its similarity

3. Application of observed algorithms in practice

In the previous section, various methods of face recognition were examined. In this section, the development part of current research is going to be described.

3.1. Experimental results

The aim of this section is to describe how to build a face recognition system for person identification, which will possibly replace or substitute a typical password authorization, therefore, it is not needed to care about head position or occlusion in images — user of such a system would be interested in correct recognition and will prevent bad sample data perception. In order to estimate each algorithm's efficiency sample datasets are going to be used; there exist various large datasets for face recognition, each of them may be used for different purposes — estimating precision rate under different

lighting conditions, with a huge number of people who look similar or containing a differently labeled pictures with facial expressions labeled.

In terms of this research, it is not needed to explore efficiency of facial expressions influence on face recognition precision rate, it is only needed to grant a sufficient precision rate for a person who can try to authorize in different lighting conditions. The FERET database containing around 2,400 images, the Yale database with around 150 images and LFW database with more than 13,000 images were considered to provide a sample data for further testing in terms of current research. All these datasets are distributed for free to improve quality of an overall face recognition technology and to obtain independent results while testing a FRS.

It is decided to conduct a testing part for the most stable appearance-based linear methods — Principal Component Analysis and Linear Discriminant Analysis — and for the most accurate local features-based methods — improved

Support Vector Machine, Local Binary Patterns and Scale Invariant Feature Transform methods. PCA and LDA proved themselves to be robust and efficient algorithms of face recognition, compared to modern iSVM and SIFT it will indicate the quality growth in facial recognition technologies, LBP represents a texture-based approach.

In order to conduct current experiments, the following scenarios were considered — 6 images available for training, 4 images to identify a person for each class and 8 images available for training, 2 images for person identification — test samples. Also, it was decided to create two subspaces — in open subspace training data consists of images under various illumination conditions and images to identify differed from training data; in closed subspace training data consists of images under different illumination conditions as well, but test samples were taken under the same circumstances. For testing purposes SciKit open-source Python library was used to implement a range of machine learning algorithms.

In all methods under observation an increase of training data led to an accuracy improvement, as well as a presence of similar images in training set and in sample image. Local feature-based methods proved themselves to be stricter than appearance-based methods. Improved Support Vector Machine proved itself as the most precise algorithm (table 2).

Table 2

Precision rates for different ratios of training/test datasets

Method	Open subspace		Close subspace	
	60/40	80/20	60/40	80/20
PCA	71.5	75	86.5	90
LDA	76.5	77	89.5	93
iSVM	79	84	91	95
LBP	78.6	81.3	90	94.5
SIFT	81	83	90	94

3.2. Implementation of facial recognition system

Current research involves the construction of the prototype of Face Recognition System. The system for security reason is divided into backend and frontend parts. The iOS application represents the client side of it, and Node.js + Python is responsible for server-side calculations. The prototype itself is implemented for authorization needs of any service that wants to provide facial verification. The system also provides an API that is used on client side.

The system is built on widely distributed products and frameworks. Nginx is one of the most efficient web

servers. FFmpeg library deals with media files handling and converting. Python scripts are used to access OpenCV functionality that has proven itself as one of the best free solutions for commercial projects. Node.js logical center is a core of the system, that handles the data and grants user to access his account. All the data between client and server is transmitted via secure web-sockets — one of the most reliable communication channels.

After registration, the evaluated classifier for each person is stored in a database and then will be retrieved when the person will try to authenticate. In case of a small precision rate at authorization process user will be proposed to make an additional training session.

3.2.1. Useful features of different algorithms

Through analysis of the algorithms its interesting features were revealed. There exists a giant number of features that can be extracted from a person's face to analyze his behavior, his intentions, his emotions and many others. All the data may be used further with different purposes — from psychological analysis to capturing a suspicious person.

Some holistic linear and nonlinear methods may also be used for finding a person wearing glasses or emotion identification, e. g., ICA and KLDA. There exist various applications of these methods, they are mostly entertaining: "Guess your nationality" or "The most similar Hollywood superstar", that are distributed widely for fun. However, there also exist various important applications of the methods, such as crowd analysis, behavioral patterns and computer vision.

Local based algorithms may be good when analyzing shape and texture. It may reveal various details about the person — skin qualities, nationality belonging, what kind of accessories he is wearing. These methods work well with detecting shapes, lines, edges or angles; therefore, it can be applied in other various areas. In current project iSVM utilize a detection of lines that were presented in datasets, and by this, defend the system from breaking when a user places a photo instead of himself. In order to test the capabilities of the system a registered person wearing sunglasses tried to authenticate but did not hit the threshold. However, it is vital to say that the system gave a 0.67 precision rate to identify a right person.

The amounts of information that can be collected with existing methods consequently lead to the topic of Big Data as only computational expenses may stop the analysis of all people that can be caught by Closed Circuit Television (CCTV) systems all over the world. It is possible to catch gender, age, ethnicity, clothes, style, eye color — all of this is potentially priceless information that can be used for direct marketing, government or medical control including during the COVID-19 pandemic.

3.2.2. Prototype implementation

The implemented program allows registration and authorization using a created classifier for the user. It may be applied to any kind of existing applications — private photo storage, messenger, e-banking etc.

Implemented algorithm of suspicious movement detection catches strange device orientation changes and protects the initial authorization module from breaking it with an image of a person. The algorithm involves usage of a motion coprocessor built-in to iPhones.

Authentication process takes approximately 5–6 seconds to gather all needed data and requires the user just to look at the camera, it is permitted to tilt the head not more than 45 degrees.

At the end of authentication process the user will be notified about the result obtained by a classifier, in case of successful and correct process handling (shaking the phone is not permitted) and if the obtained precision rate is over the specified threshold the user is granted access to the account where photos are stored. For the process of registration nearly the same screens are being used with different content though, the user will spend around 10 seconds sending data, he will be asked to face the camera if he moves too much.

All the data is stored and processed on the server. Once the user completes his photo session, an array of photos is delivered to the server, each picture is analyzed, cropped, normalized and modified for better recognition purposes. Based on the linear Support Vector Machine approach it evaluates the classifier, using the benefits of this approach it constructs two classifiers — one for person recognition and one for person differentiation, so-called "within-class" and "in-class" classifiers. Then the training stage is over. However, it is possible to train classifiers multiple times, growing the size of the datasets and making the predictions better.

The application is able to analyze not only photos but videos, so each user's registration grants a 25 fps video of around 8 seconds, which contains around 40 suitable images that can be analyzed to compare emotional features of each person, features of the skin and the eyes under different circumstances, it is possible to require actions from user to approve his identity by shaking his head or blinking.

Conclusion

In terms of current research, the analysis of existing technologies was conducted, the most significant methods were tested and afterwards implemented in the FRS prototype. The analysis of technologies included the structure description and evaluation of existing facial recognition systems, groups of methods and different

approaches to face detection, analyzing, processing and recognition were reviewed. The testing stage included a practical test of different algorithms applied on the same standard datasets in order to determine the most efficient algorithm. In the implementation the Support Vector Machine method, which proved itself the one of most precise, was built in the server-side logic of the system. A client-side iOS application allows simplified access to the classifier, it allows its modification — to increase the amount of new data — or its use to verify a person's identity.

A huge stack of technologies being used for the implementation of facial recognition systems. It is important to notice that with the modern growth rates of technologies the number of alternatives that might be used for direct or indirect application in FRS will grow dramatically.

This paper contains information collected, interpreted and structured for facial recognition purposes. It may help to choose an area of further investigation for either creating new approaches, or building a system based on existing methods. Video-based methods of analyzing data should be of interest as it is a wide area of possible investigation. It is also recommended to pay attention to 3D vision methods, technologies that allow a volumetric object to be built within the camera of a phone are already presented and being implemented and distributed in modern devices.

Nowadays, biometric methods of authentication are replacing standard methods of authorization and sooner will cover multiple different areas of human-machine interaction. The implementation and development of an algorithm for video analysis of human behavior patterns may be proposed as a possible continuation of this research.

The author expresses gratitude to Alexey Ryabov, Konstantin Kudelkin and Kirill Babich for fruitful cooperation in creating existing prototypes of the face recognition system, as well as in research on this topic.

References

1. Singh A., Singh S. K., Tiwari S. Comparison of face Recognition Algorithms on Dummy Faces, *The International Journal of Multimedia & Its Applications (IJMA)*, 2012, vol. 4, no. 4, pp. 121–135.
2. Sukhija P., Behai S., Singh P. Face Recognition System Using Genetic Algorithm, *Procedia Computer Science*, 2016, vol. 85, pp. 410–417.
3. Tareeq S. M., Parveen R., Rozario L. J., Bhuiyan A. A. Robust Face Detection using Genetic Algorithm, *Information Technology Journal*, 2007, vol. 6, pp. 142–147.
4. Panigrahy M. P., Kumar N. Face Recognition using Genetic Algorithm and Neural Networks, *International Journal of Computer Applications*, 2012, vol. 55, no. 4, available at: <http://research.ijcaonline.org/volume55/number4/pxc3882613.pdf>.
5. Wiskott L., Fellous J. M., Kruger N., Malsburg C. V. D. Face Recognition by Elastic Bunch Graph Matching, *Intelligent Biometric Techniques in Fingerprint and Face Recognition* / eds. L. C. Jain et al., CRC Press. 1999. Chapter 11, pp. 355–396.

6. **Le T. H.** Applying Artificial Neural Networks for Face Recognition, *Advances in Artificial Neural Systems*, 2011, vol. 2011, pp. 1–16.
7. **Smith L. I.** A tutorial on Principal Components Analysis (Computer Science Technical Report No. OUCS-2002-12), available at: <https://ourarchive.otago.ac.nz/handle/10523/7534>
8. **Welling M.** Fisher Linear Discriminant Analysis, available at: <https://www.cs.huji.ac.il/w~csip/Fisher-LDA.pdf>
9. **Raschka S.** Linear Discriminant Analysis, *Bit by Bit 2014*, available at: https://www.researchgate.net/publication/264825166_Linear_Discriminant_Analysis_bit_by_bit
10. **Naseem I., Togneri R., Bennamoun M.** Linear Regression for Face Recognition, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2010, vol. 32, no. 11, pp. 2106–2112.
11. **Lai J., Jiang X.** Robust Face Recognition Using Trimmed Linear Regression, *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, 2013, pp. 2979–2983.
12. **Liu C., Wechsler H.** Comparative Assessment of Independent Component Analysis (ICA) for Face Recognition, available at: <http://www.face-rec.org/algorithms/ica/liu99comparative.pdf>
13. **Das D. K.** Comparative Analysis of PCA and 2DPCA in Face Recognition, *International Journal of Emerging Technology and Advanced Engineering*, 2012, vol. 2, issue 1, pp. 330–336.
14. **Yang M. H.** Face Recognition Using Kernel Methods, NIPS, 2001.
15. **Ji S., Jieping Ye. J.** Generalized Linear Discriminant Analysis: A Unified Framework and Efficient Model Selection, *IEEE Transactions on Neural Networks*, 2008, vol. 19, no. 10, pp. 1768–1782.
16. **Kadoury S.** Face Detection Using Locally Linear Embedding, 2005, available at: <https://pdfs.semanticscholar.org/830d/35f4a9d8df629a35a05c514a6c45c64f4f62.pdf>
17. **Tseng S.** Comparison of Holistic and Feature Based Approaches to Face Recognition. 2003, available at: <https://www.semanticscholar.org/paper/COMPARISON-OF-HOLISTIC-AND-FEATURE-BASED-APPROACHES-Tseng/84bc48c85b2389f700779c1bda8d808dc4972451>
18. **Rahim A., Hossain N., Wahid T., Azam S.** Face Recognition using Local Binary Patterns (LBP), *Global Journal of Computer Science and Technology Graphics & Vision*, 2013, vol. 13, issue 4, available at: https://globaljournals.org/GJCST_Volume13/1-Face-Recognition-using-Local.pdf
19. **Jia H. and Martinez A. M.** Support Vector Machines in Face Recognition with Occlusions, *CVPR*, 2009, pp. 136–141.
20. **Li Z., Tang S.** Face Recognition Using Improved Pairwise Coupling Support Vector Machines, available at: <https://www.csie.ntu.edu.tw/~cjlin/courses/slt2002/cr1288.pdf>
21. **Ishraque S. Z., Jabid T., Chae O.** Face Recognition Based on Local Directional Pattern Variance (LDPV), *Proc. International Conference on Computer Science and Technology*, Jeju, Korea, June 25–27, 2012, pp. 101–106.
22. **Gupta T., Garg L.** Face Recognition Using SIFT, *International Journal of Emerging Technology and Advanced Engineering*, 2014, vol. 4, issue 5, pp. 358–363.
23. **Sadeghipour E., Sahragard N.** Face Recognition Based on Improved SIFT Algorithm, *International Journal of Advanced Computer Science and Applications*, 2016, vol. 7, no. 1, pp. 548–551.
24. **Reda A., Aoued B.** Artificial Neural Network-Based Face Recognition, *First International Symposium on Control, Communications and Signal Processing*, 2004, available at: <https://eurasip.org/Proceedings/Ext/ISCCSP2006/defevent/papers/cr1117.pdf>
25. **Kasar M. M., Bhattacharyya D., Kim T.** Face Recognition Using Neural Network, *A Review — International Journal of Security and Its Applications*, 2016, vol. 10, no. 3, pp. 81–100.
26. **Chen Q., Kotani K., Lee F., Ohmi T.** Face Recognition Using Self-Organizing Maps, available at: <http://cdn.intechopen.com/pdfs/10464.pdf>
27. **Yi Jin, Qiu-Qi Ruan.** Face Recognition Using Gabor-Based Improved Supervised Locality Preserving Projections, *Computing and Informatics*, 2009, vol. 28, pp. 81–95.
28. **Wolf L., Hassner T., Taigman Y.** Descriptor Based Methods in the Wild, *Faces in Real-Life Images workshop at the European Conference on Computer Vision (ECCV)*, Oct 2008, available at: https://talhassner.github.io/home/publication/2008_ECCV
29. **Ahonen T., Hadid A., Pietikainen M.** Face Description with Local Binary Patterns: Application to Face Recognition, *IEEE Trans Pattern Anal Mach Intell.*, 2006, vol. 28, no. 12, pp. 2037–2041.
30. **Tan J. S., Venkat I., Liao I. Y., Wilde P. D.** General Human Traits Oriented Generic Elastic Model for 3D Face Reconstruction, *BMVC 2016*, available at: https://www.semanticscholar.org/paper/305768198_General_Human_Traits_Oriented_Generic_Elastic_Model_for_3D_Face_Reconstruction/links/57f6116108ae886b89816c4c/General-Human-Traits-Oriented-Generic-Elastic-Model-for-3D-Face-Reconstruction.pdf?origin=publication_detail
31. **Stallkamp J., Ekenel H. K., Stiefelhagen R.** Video-based Face Recognition on Real-World Data, *2007 IEEE 11th International Conference on Computer Vision*. 14–21 Oct. 2007, pp. 1–8.
32. **Burgos-Artizzu X. P., Perona P., Dollar P.** Robust Face Landmark Estimation under Occlusion, *2013 IEEE International Conference on Computer Vision*, Sydney, Australia, 1–8 December 2013, pp. 1513–1520.
33. **Wojciek W., Gromaszek K., Junisbekov M.** Face Recognition: Issues, Methods and Alternative Applications, 2016, available at: <https://www.intechopen.com/chapters/51031>

М. В. Ступина, канд. пед. наук, доц., masamvs@bk.ru,
Донской государственный технический университет, Ростов-на-Дону

Архитектура программной системы поддержки учета посещаемости в вузе

В соответствии с модульно-рейтинговой системой, принятой в вузах, а также в силу смещения акцентов в сторону самостоятельной работы обучающихся и построения индивидуальных образовательных траекторий, актуализируется задача учета посещаемости. В статье рассмотрен проект системы, основной целью которой является автоматизация процесса подтверждения присутствия обучающихся на учебном занятии за счет сканирования QR-кода. Представлены результаты визуального моделирования, структура системы, организация хранения данных, а также алгоритмы работы основных программных модулей. Приведены тестовые примеры работы веб-приложения и мобильного приложения как основных компонентов программной системы учета посещаемости.

Ключевые слова: учет посещаемости, электронный журнал, QR-код, диаграмма прецедентов, архитектура системы, мобильное приложение, веб-приложение, сервер обработки, СУБД MySQL, Swift, PHP

Введение

Стремительное развитие ИТ-сфера и средств информационно-коммуникационных технологий в последние годы привели к автоматизации множества областей человеческой деятельности, в том числе и системы высшего образования. Необходимость обеспечения конкурентоспособности университета и его соответствия мировым стандартам в области просвещения в современных условиях требуют цифровизации всей инфраструктуры вуза и реинжиниринга следующих его процессов в области автоматизации: административно-хозяйственной деятельности, управленического учета, управления учебным процессом и т. д. [1, 2]. С этой целью вузы переходят на комплексные системы управления, позволяющие автоматизировать различные информационные процессы. К таким процессам можно отнести хранение, обработку и анализ информации в рамках деятельности вуза в различных областях, таких как поступление, обучение и оплата за обучение; расчет и распределение нагрузки; управление научной работой и инновациями; выполнение основных операций в личных кабинетах пользователей; деятельность на рабочем месте структурных подразделений (приемная комиссия, деканаты, отдел магистратуры и аспирантуры, диссертационные советы, управление дополнительным образованием и т. д.). При этом

все возрастающие требования к вузам со стороны государства требуют постоянной модернизации программно-аппаратного обеспечения, дополнения его новыми функциональными возможностями, а также создания специализированных интернет-сервисов и мобильных приложений.

Действующими Федеральными государственными образовательными стандартами, в соответствии с которыми осуществляется подготовка в рамках образовательных программ, определяются необходимость наличия в вузе системы оценки качества подготовки обучающихся. В связи с этим в вузах широкое применение получила модульно-рейтинговая система как форма контроля, которая, в частности, учитывает и посещаемость занятий обучающимися. Сказанное выше определяет необходимость решения задач хранения, представления, учета, анализа и т. д. информации о посещаемости обучающихся в электронном виде.

В то же время, как показывает практика, подтверждение присутствия обучающихся на занятии осуществляется путем проведения преподавателем переклички с последующим занесением информации в электронный журнал. Этот процесс является довольно рутинным, может занимать продолжительное время в случае большой аудитории слушателей, а также может сопровождаться ошибками, связанными с человеческим фактором. Например,

обучающийся может не услышать свою фамилию, преподаватель может пропустить обучающегося в списке группы или внести некорректные данные и т. п.), что актуализирует процесс поиска программных решений автоматизации процесса учета посещаемости в вузе.

Проблема исследования

Современные программные решения, существующие на рынке, предлагают ряд комплексных систем для вузов класса CRM (*Customer Relationship Management*), отвечающих за управление взаимоотношениями с клиентами и автоматизацию бизнес-процессов. В качестве примеров подобных CRM можно назвать ListOkCRM, Отмечалка, Таллант, BasePlan и др. Возможности CRM позволяют формировать журналы посещаемости, распределять учебную нагрузку, автоматически подготавливать необходимые статистические отчеты и т. д. Однако в CRM не в полной мере реализованы программные модули, непосредственно автоматизирующие процесс подтверждения присутствия обучающихся на учебном занятии. Кроме того, переход на CRM требует значительных материальных вложений и трудовых затрат на изменения структуры программно-аппаратных комплексов вуза.

На настоящее время все большее распространение получают технологии управления мобильными устройствами для создания мобильной образовательной среды в вузе. Так, например, в работе [3] представлены возможности использования мобильных устройств в целях обеспечения контроля посещаемости в автоматическом режиме на основании фактического местоположения обучающегося.

Особенности применения беспроводных технологий, таких как Bluetooth, Wi-fi, NFC и др. нашли отражений в работах [4–7]. Процесс внедрения комплекса мобильных средств поддержки учебного процесса (мобильные приложения "Расписание", "Распределение", "Успеваемость", "Посещаемость" и др.) рассмотрен в работе исследователей на базе Санкт-Петербургского политехнического университета [4]. Так, разработанное мобильное приложение "Посещаемость" позволяет автоматически регистрировать присутствие обучающегося на занятии на основании беспроводных технологий. Как отмечают авторы, это требует наличия в аудиториях специализированного оборудования.

В настоящее время автоматический учет посещаемости может быть реализован на базе алгоритмов детектирования и распознавания лиц.

В работе [8] представлено решение задачи детектирования с использованием алгоритма Виолы–Джонса и перспективы данного подхода в настройке эталонов студентов, учете посещаемости и его анализа. В исследовании [9] авторами подробно рассмотрены различные алгоритмы, методы и подходы к распознаванию лиц, представлена архитектура системы учета посещаемости и реализация системы с использованием искусственных нейронных сетей.

В качестве еще одного из наиболее перспективных направлений в области автоматизации процесса учета посещаемости учебных занятий можно выделить использование QR-кодов, получивших распространение во время пандемии коронавирусной инфекции COVID-19. Так, в работе [10] рассмотрены вопросы генерации QR-кода в форме PNG с использованием библиотеки PHP QR Code, сканирования QR-кода мобильным устройством обучающимися с использованием API, занесения данных о присутствии обучающегося на занятии в электронный журнал. Отмеченные автором основные направления использования QR-кодов в учебном процессе свидетельствуют о перспективах данного подхода к автоматизации процесса учета посещаемости.

Постановка задачи исследования

Необходимость разработки программной системы учета посещаемости обусловлена потребностью в автоматизации процессов сбора, передачи и хранения данных о посещаемости обучающихся.

С учетом современных тенденций в области распространения QR-кодов, обеспечивающих кодирование любой информации и возможностей ее оперативного распознавания с использованием камер мобильных устройств, оптимальным видится создание программной системы, включающей в себя:

- мобильное приложение для обучающихся, позволяющее выполнить сканирование QR-кода;
- веб-приложение для ведения электронного журнала преподавателем.

Основные функциональные возможности программной системы включают:

- хранение данных о посещаемости обучающихся;
- предоставление данных о посещаемости в табличном виде всем субъектам учебного процесса;
- оперативный доступ преподавателя к веб-приложению с любого устройства;
- настройка электронного журнала посещаемости и импорт оперативной информации об учебных дисциплинах и обучающихся;

- генерация QR-кода учебного занятия в консоли преподавателя;
- фиксация присутствия обучающегося на учебном занятии путем сканирования QR-кода камерой мобильного устройства.

Для разработки предложенной системы необходимо решить перечисленные далее задачи.

1. Разработать архитектуру программной системы учета посещаемости.
2. Выполнить проектирование базы данных электронного журнала посещаемости.
3. Разработать веб-приложение, обеспечивающее доступ к информации о посещаемости и позволяющее генерировать QR-код учебного занятия.
4. Разработать мобильное приложение, позволяющее выполнять сканирование QR-кода.

Архитектура программной системы учета посещаемости

Программная система учета посещаемости представляет собой совокупность трех компонентов (рис. 1):

- веб-приложение (используется для генерации QR-кода и получения доступа к электронному журналу посещаемости);
- мобильное приложение (используется для выполнения сканирования QR-кода и передачи

данных в электронный журнал, необходима авторизация);

- сервер обработки (используется для получения запроса от мобильного приложения через API).

Основными информационными потоками в системе являются: данные для авторизации преподавателя; данные об учебных дисциплинах; данные об обучающихся; id учебной дисциплины; данные для авторизации обучающегося; аутентификационные данные обучающегося; данные о присутствии на занятии; данные о посещаемости.

Следует отметить, что с сервером обработки пользователи программной системы напрямую не контактируют. Данный компонент программной системы обеспечивает работу разработанного API. В разработанной программной системе API умеет обрабатывать три следующих ключевых запроса.

1. Запрос на авторизацию обучающегося в мобильном приложении. В этом запросе сервер обработки получает логин и пароль пользователя, а возвращает ответ в виде JSON-документа, содержащий статус авторизации обучающегося в программной системе.

2. Запрос на внесение отметки о присутствии обучающегося на учебном занятии. В этом запросе сервер обработки получает аутентификационные данные обучающегося и уникальный иден-

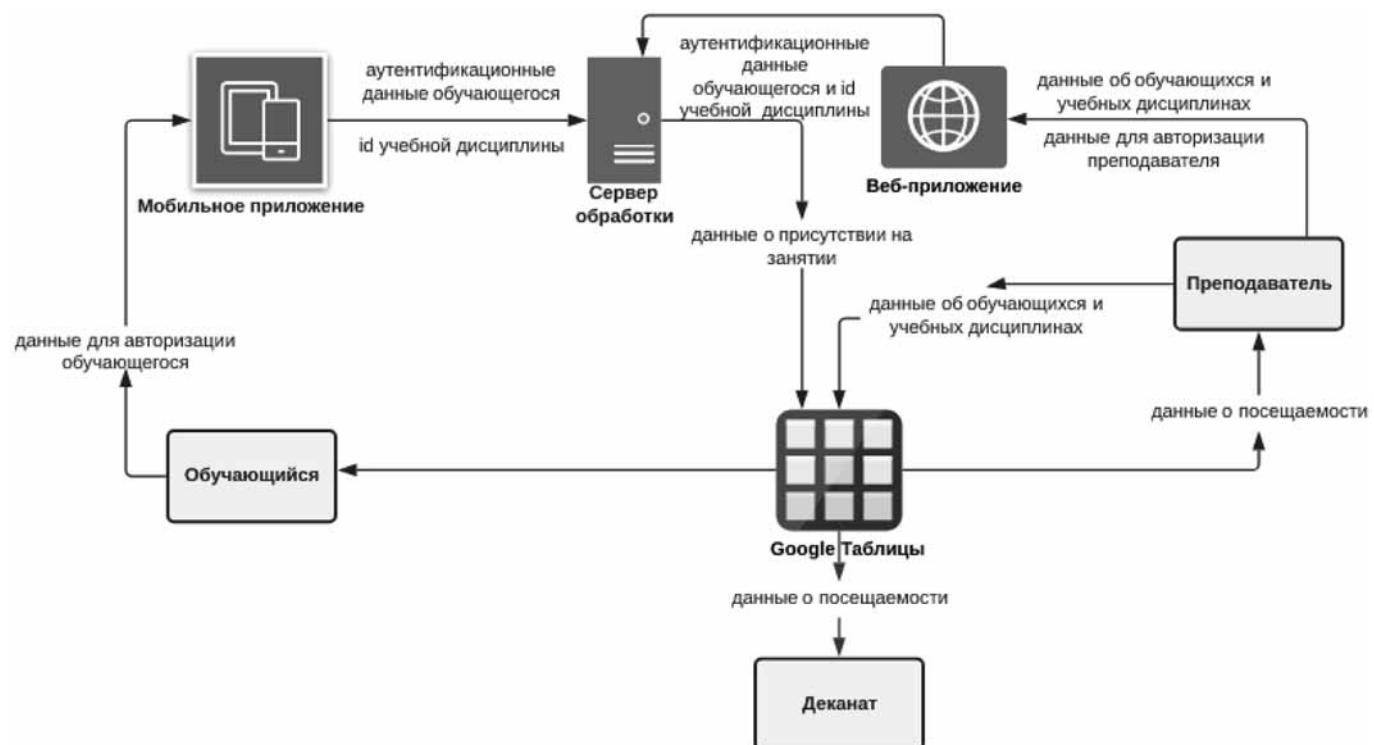


Рис. 1. Архитектура программной системы учета посещаемости

тификатор учебной дисциплины (соответствует spreadsheet id в Google Sheets API), полученный в результате сканирования QR-кода. Ответ сервера обработки данных содержит результат операции по внесению данных о присутствии.

3. Запрос на получение данных обучающегося. При данном запросе сервер обработки получает только аутентификационные данные обучающегося, а в качестве ответа возвращает массив данных (в виде JSON-документа), в котором указаны ФИО и учебная группа обучающегося.

Программная реализация системы

Моделирование системы. Проектирование программной системы выполнено на основе принципов объектно-ориентированного программирования с помощью языка моделирования UML. Для описания системы на концептуальном уровне была применена диаграмма вариантов использования (рис. 2).

Основные акторы: обучающийся, преподаватель, деканат. Варианты использования (прецеденты) представлены в таблице.

Хранение данных. Структура базы данных веб-приложения построена на основе реляционной модели данных (рис. 3).

Выделены четыре отношения:

1) Student_auth_data — авторизационные данные обучающихся:

- id — уникальный идентификатор;
- username — логин;
- password — пароль;

2) Student_personal_data — персональные данные об обучающихся:

- id — уникальный идентификатор;
- firstname — имя;
- lastname — фамилия;
- middlename — отчество;
- stud_group — учебная группа;

3) Teacher_auth_data — авторизационные данные преподавателей:

- id — уникальный идентификатор;
- username — логин;
- password — пароль;
- FIO — ФИО преподавателя;

4) Lessons — учебные дисциплины:

- id — уникальный идентификатор;

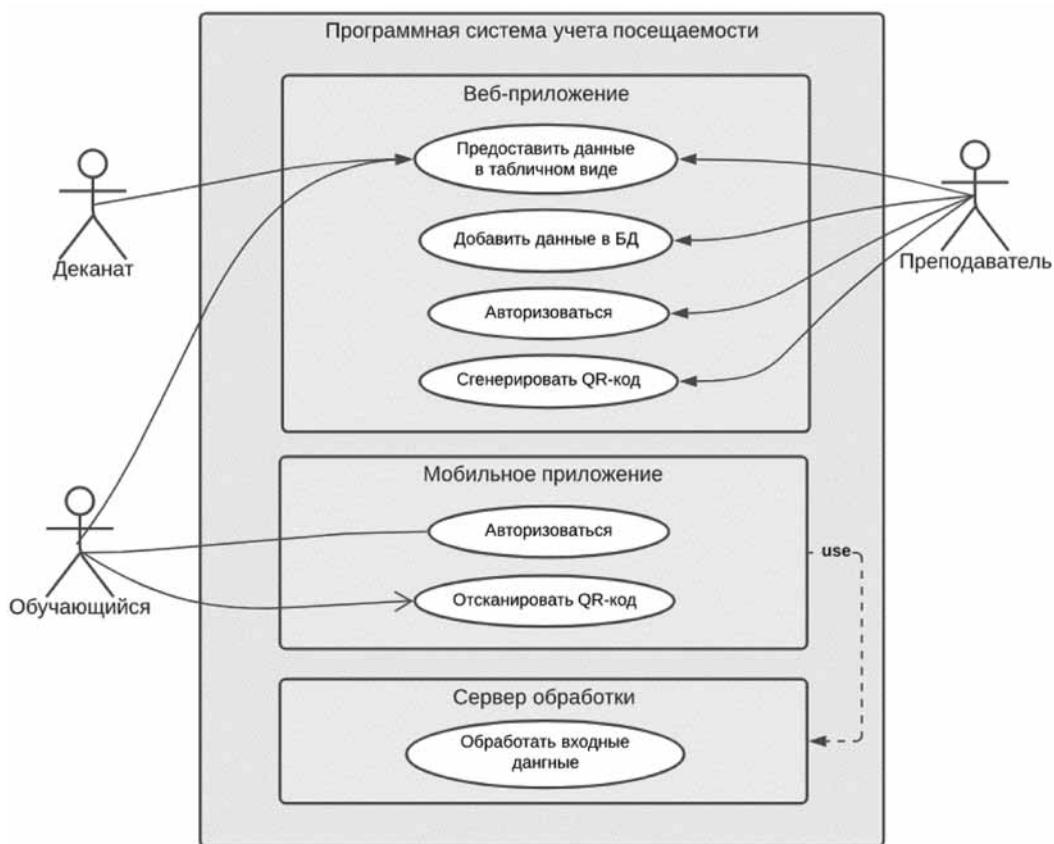


Рис. 2. Диаграмма вариантов использования

Описание прецедентов

Вариант использования	Краткое описание
Предоставить данные о посещаемости в табличном виде	Запускается обучающимся, преподавателем, деканатом. Позволяет просмотреть информацию о посещаемости в электронном журнале
Авторизоваться в веб-приложении	Запускается преподавателем. Позволяет преподавателю авторизоваться в веб-приложении для дальнейшего взаимодействия с ним
Внести данные в базу данных	Запускается преподавателем. Позволяет преподавателю внести в базу данных данные об обучающихся, дисциплинах
Сгенерировать QR-код	Запускается преподавателем. Позволяет представить spreadsheetid учебной дисциплины в графическом виде (QR-код)
Авторизоваться в мобильном приложении	Запускается обучающимся. Позволяет обучающемуся авторизоваться в мобильном приложении для дальнейшего взаимодействия с ним
Отсканировать QR-код	Запускается обучающимся. Позволяет считать информацию с QR-кода и отправить ее на сервер обработки
Обработать входные данные	Используется мобильным приложением. Позволяет серверу обработки обработать переданные мобильным приложением данные

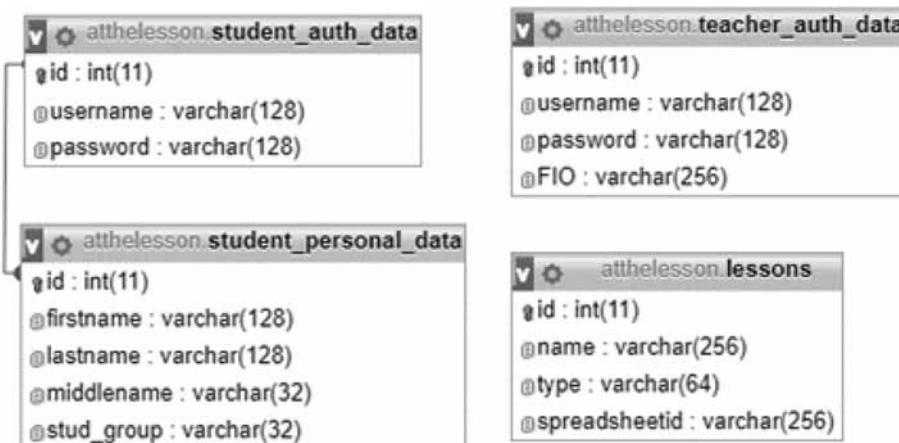


Рис. 3. Схема базы данных

- name — название дисциплины;
- type — тип учебного занятия;
- spreadsheetid — идентификатор Google Таблицы.

В качестве хранилища данных о посещаемости используются электронные таблицы Google Таблиц. Для учета посещаемости преподаватель создает несколько таких таблиц, а внутри таких таблиц — листы, в которые и заносятся данные о посещаемости. Каждый такой лист соответствует учебной группе, а каждая таблица — учебной дисциплине.

Модули веб-приложения. В качестве программных инструментальных средств разработки веб-приложения были выбраны: язык разметки HTML, язык описания стилей CSS, серверный язык программирования PHP и система управления базами данных MySQL. В качестве среды разработки использовался текстовый редактор Microsoft Visual

Studio Code с установленными расширениями для работы с HTML, CSS и PHP.

Электронный журнал, в котором отражена посещаемость обучающихся, представляет собой веб-приложение. Для работы с электронным журналом преподавателю необходимо авторизоваться, после чего станет доступна возможность просмотра и редактирования данных в специальной консоли. После настройки журнала, импорта данных о расписании занятий и учебных группах преподаватель имеет возможность сгенерировать QR-код, уникальный для каждого занятия.

На листинге 1 представлен фрагмент программного кода на языке PHP, отвечающий за генерацию QR-кода на соответствующей странице. В рассматриваемой программной системе используется генерация QR-кодов третьей версии, что означает создание QR-кода размером 29×29 модулей (точек).

```

if (isset($_REQUEST['data'])) {
    if (trim($_REQUEST['data']) == '')
        die ('Данные пустые, исправьте это!');
}
else {
    $filename = $PNG_TEMP_DIR.'test'.md5($_REQUEST['data'].'|
        .$errorCorrectionLevel.'|'.$matrixPointSize).'.png';
    QRcode::png($_REQUEST['data'],$filename,
        $errorCorrectionLevel, $matrixPointSize, 2);
    echo '<hr/>';
    QRcode::png('Base Error Code Generation',$filename,
        $errorCorrectionLevel, $matrixPointSize, 2);
}

```

Листинг 1. Генерация QR-кода

Выбор размера при генерации отвечает за масштабирование генерируемого QR-кода в целях удобного вывода как на большие экраны, так и на экраны мобильных телефонов.

Модули мобильного приложения. Разработка мобильного приложения осуществлялась на языке Swift. В качестве среды разработки был выбран инструмент Xcode.

Обучающимся необходимо авторизоваться в мобильном приложении программной системы, после чего для них станет доступен функционал по сканированию QR-кода.

На листинге 2 представлен фрагмент программного кода обработки мобильным приложением ответа от сервера обработки на языке Swift.

Сканирование QR-кода, который был сгенерирован преподавателем, позволяет идентифицировать присутствие обучающегося на занятии. После сканирования QR-кода мобильное приложение выполняет запрос к серверу обработки данных через API. Ответ сервера обработки данных в JSON-формате будет содержать сообщение об успешном внесении данных о присутствии на занятии, либо сообщение об ошибке.

```

var qrData: QRData? = nil {
    didSet {
        if qrData != nil {
            MarkMe(onDone: { (response) in
                self.result = response.summary
                if (self.result! == "successfully"){
                    self.showDialogue(text: "Вы отметили
                    свое присутствие на паре!", title: "Успешно!")
                } else {
                    self.showDialogue(text: "Что-то пошло не так. Попробуйте еще раз.", title: "Ошибка!")
                }
            },
            onError: { (error) in
                self.showDialogue(text: "Отсканированный код
                не является корректным.\nОбратитесь к
                преподавателю.", title: "Ошибка!") },
            spreadsheetId: qrData!.codeString!,
            nickname: UserDefaults.standard.string(forKey: "nickname")!,
            password: UserDefaults.standard.string(forKey: "password")!
        })
    }
}
?>

```

Листинг 2. Обработка ответа от сервера

Результаты исследования

На рис. 4 представлена главная страница веб-приложения, позволяющая преподавателю авторизоваться в веб-приложении программной системы учета посещаемости.

После авторизации становится доступной консоль преподавателя (рис. 5), где можно внести из-

менения в электронный журнал, выполнить импорт учебных дисциплин и списка обучающихся учебной группы, сгенерировать QR-код, а также сделать ряд других настроек.

На рис. 6 представлен фрагмент консоли, где преподаватель может добавлять, редактировать и удалять учебные дисциплины.

Консоль преподавателя также содержит страницу генерации QR-кода. Для того чтобы сгенерировать QR-код, который должны будут отсканировать обучающиеся, преподавателю нужно перейти на страницу генерации QR-кода, выбрать учебную дисциплину, вид занятия и размер генерируемого кода, а затем нажать на кнопку "Сгенерировать QR-код". Это действие преподаватель выполняет перед каждым учебным занятием.

Пример генерации для одной из лекций по дисциплине "Информационные технологии" QR-кода размером 10×10 мм — наименьшим размером, воспринимаемым камерой мобильного устройства, — представлен на рис. 7.

Обучающиеся проходят авторизацию в мобильном приложении. В случае неверного ввода данных будет открыто всплывающее окно, которое сообщает об ошибке. При этом аутентификационные данные обучающегося сохраняются, что при повторном запуске мобильного приложения позволяет сразу открыть страницу-сканер QR-кода.

Страница-сканер QR-кода является главным экраном мобильного приложения, позволяюще-

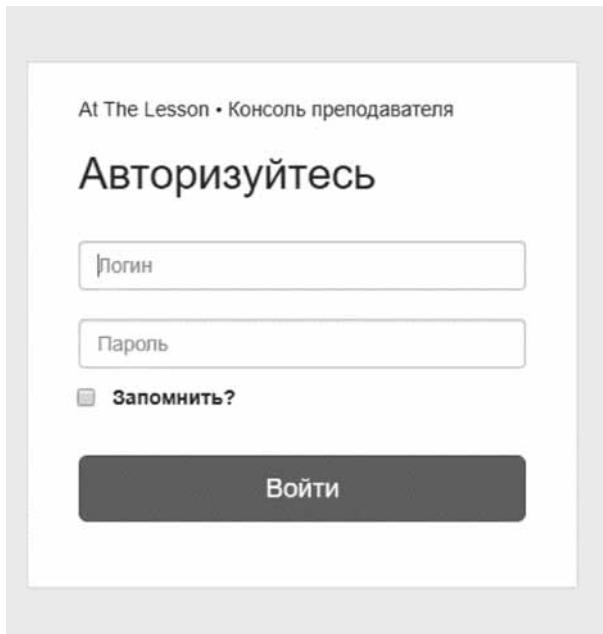


Рис. 4. Экран авторизации веб-приложения

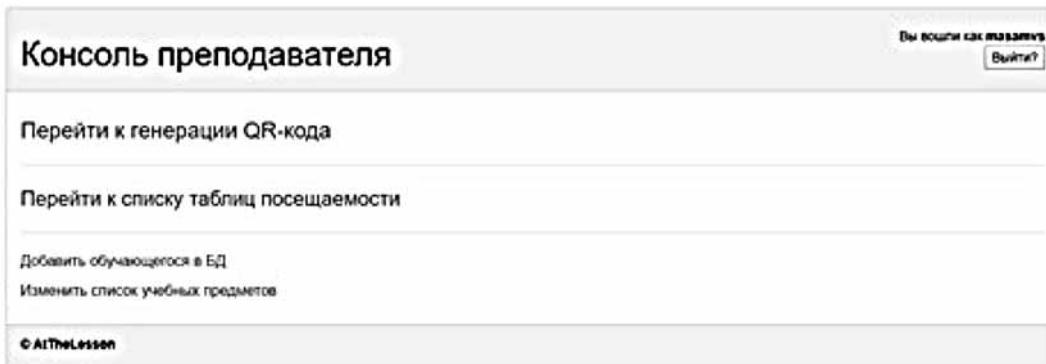


Рис. 5. Консоль преподавателя

Lessons						Вернуться в консоль	Добавить новую запись
						Поиск	X
#	<input type="checkbox"/>	Название дисциплины	Тип занятия	ID листа Google Sheets	Действия		
1	<input type="checkbox"/>	Информационные технологии	Лекция	1G_CKvWY5FWWlnKjnDcXF47xGyOMzGxq5Kg0x0n7U04E	<input type="button" value=""/>	<input type="button" value=""/>	
2	<input type="checkbox"/>	Информационные технологии	Практика	114rX1xT3T5k4FFYR0bwqv4jat_43-uaTaGHferh4YuM	<input type="button" value=""/>	<input type="button" value=""/>	
#	<input type="checkbox"/>	Название дисциплины	Тип занятия	ID листа Google Sheets	Действия		
						Select	

Рис. 6. Отображение списка учебных дисциплин в консоли преподавателя



Рис. 7. Сгенерированный QR-код

го отметить свое присутствие на занятии. Чтобы отсканировать QR-код, сгенерированный преподавателем и выведенный на экран проектора, обучающийся должен навести камеру мобильного устройства, на котором установлено мобильное приложение, на QR-код. Затем мобильное приложение автоматически отсканирует QR-код и запустит процесс отправки данных на сервер обработки. В случае, если обучающийся отсканирует недействительный QR-код, приложение выведет соответствующее всплывающее окно с ошибкой. Если же обучающийся отсканирует корректный QR-код, и он является обучающимся группы, для которой был сгенерирован QR-код в рамках учебной дисциплины, то мобильное

приложение выведет соответствующее всплывающее окно, которое сообщит об успешной отметке о присутствии на учебном занятии (рис. 8).

В дальнейшем преподавателю доступна информация о посещаемости обучающихся (персональные данные скрыты в соответствии с Федеральным законом "О персональных данных" от 27.07.2006 № 152-ФЗ) на учебном занятии в электронном журнале (рис. 9).

Электронный журнал на базе облачного сервиса Google Таблицы позволяет формировать отчеты и статистику посещения учебных занятий, а сто-

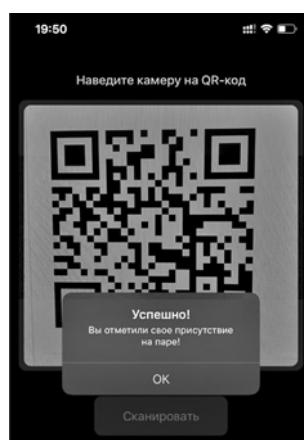


Рис. 8. Экран с успешной отметкой присутствия

№	ФИО	07.02	14.02	21.02	28.02	07.03	14.03	21.03
1	#####	присутствует	присутствует	присутствует	присутствует	присутствует		присутствует
2	#####				присутствует	присутствует	присутствует	присутствует
3	#####							
4	#####	присутствует		присутствует	присутствует	присутствует	присутствует	
5	#####	присутствует		присутствует	присутствует	присутствует	присутствует	
6	#####	присутствует						
7	#####	присутствует	присутствует				присутствует	присутствует
8	#####		присутствует	присутствует	присутствует	присутствует	присутствует	присутствует
9	#####		присутствует	присутствует	присутствует	присутствует		присутствует
10	#####		присутствует	присутствует	присутствует	присутствует	присутствует	присутствует
11	#####			присутствует	присутствует	присутствует	присутствует	присутствует
12	#####			присутствует	присутствует	присутствует	присутствует	присутствует
13	#####	присутствует						присутствует
14	#####							присутствует
15	#####	присутствует	присутствует	присутствует	присутствует	присутствует		присутствует
16	#####	присутствует				присутствует		присутствует
17	#####	присутствует	присутствует	присутствует	присутствует	присутствует		присутствует
18	#####					присутствует	присутствует	присутствует
19	#####	присутствует	присутствует	присутствует	присутствует	присутствует		присутствует
20	#####	присутствует	присутствует	присутствует	присутствует	присутствует		присутствует
21	#####	присутствует	присутствует	присутствует	присутствует	присутствует	присутствует	
22	#####							
23	#####		присутствует					
24	#####	присутствует	присутствует	присутствует	присутствует	присутствует	присутствует	

Рис. 9. Фрагмент электронного журнала посещаемости

ронние дополнения и поддержка скриптов делают возможным расширение базового функционала в соответствии с потребностями субъектов образовательного процесса.

Заключение

Необходимость контроля освоения обучающимися образовательной программы, выполнение ими индивидуальных планов, в том числе и посещение учебных занятий, предусмотренных учебным планом, определяет необходимость со стороны преподавателя ведения учета посещений учебных занятий обучающимися.

Процесс учета посещаемости одной группы или потока при использовании программной системы учета посещаемости требует минимальных временных затрат, включая время на запуск веб-приложения и генерацию QR-кода, доступного для сканирования обучающимися.

Следует отметить, что несмотря на возможность передачи и распространения в электронном виде QR-кода учебного занятия среди студентов, фальсификация данных о присутствии на занятиях может быть проведена и при других способах учета посещаемости. Например, при проведении потоковой лекции в аудитории более 60 человек, перекличка со стороны преподавателя может сопровождаться откликом одного студента на чужую фамилию — фамилию отсутствующего студента. Процесс учета посещаемости путем рукописной записи в списке присутствующих группы может быть сфальсифицирован за счет изменения почерка. Журналы посещаемости, заполняемые старостами, также могут быть исправлены в пользу отсутствующих студентов. Таким образом, при указанных способах учета посещаемости данные могут быть недостоверны, а их проверка с последующим заполнением электронного журнала требует дополнительного аудиторного времени или личного времени преподавателя. В то же время, использование QR-кодов, несмотря на возможность фальсификации данных о посещаемости со стороны

некоторых студентов, позволяет в наибольшей степени ускорить и автоматизировать процесс учета посещаемости.

Дальнейшее развитие программной системы учета посещаемости связано с разработкой кроссплатформенного мобильного приложения, а также интеграцией данного программного комплекса в единую информационно-образовательную среду вуза.

Список литературы

1. Новгородова Н. Г., Редькина Б. А. Автоматизация образовательной деятельности профессионально-педагогического вуза // Международный журнал экспериментального образования. 2014. № 3-2. С. 15–17.
2. Зафиевский А. В. Автоматизация управления учебным процессом в вузе // Успехи современного естествознания. 2010. № 1. С. 115–117.
3. Колосков С. С., Бабешко В. Н., Самочадин А. В., Кошич Ю. А. Методика апробации внедрения мобильных сервисов на платформе управления мобильными устройствами в вузах // Экономика. Информатика. 2015. № 19 (216). С. 152–159.
4. Райчук Д. Ю., Самочадин А. В., Носицын С. М., Хмельков И. А. Комплекс мобильных средств поддержки учебного процесса // Информатика, телекоммуникации и управление. 2014. № 6 (210). С. 7–15.
5. Bhalla V., Singla T., Gahlot A., Gupta V. Bluetooth Based Attendance Management System // IJIET. 2013. Vol. 3, Iss. 1. P. 524–527.
6. More V., Nayak S. Attendance Automation using Near Field Communication (NFC) Technology // Internat. J. of Scientific & Engineering Research. 2013. Vol. 4, Iss. 12. P. 572–575.
7. Ayu M. A., Ahmad B. I. TouchIn: An NFC Supported Attendance System in a University Environment // Internat. J. of Information and Education Technology. 2014. Vol. 4, No. 5. P. 448–453.
8. Синецкий Р. М., Гавриков М. М. Система учета посещения занятий студентами на основе алгоритмов распознавания лиц // Известия вузов. Северо-Кавказский регион. Серия: Технические науки. 2016. № 3 (191). С 24–30.
9. Иванова Е. В., Струева А. Ю. Система учета посещаемости студентов на основе распознавания лиц // Вестник ЮУрГУ. Серия: Вычислительная математика и информатика. 2021. Том 10, № 4. С. 60–73.
10. Stupina M. V., Anistratenko K. V., Pazina L. O. Using the QR code as a means of automating the process of accounting for attendance at educational classes // Journal of Physics: Conference Series. 2021. Vol. 2131. Article 022077. URL: <https://iopscience.iop.org/article/10.1088/1742-6596/2131/2/022077/pdf>

The Architecture of the Software System for Supporting Attendance Records at the University

M. V. Stupina, masamvs@bk.ru,
Don State Technical University, Rostov-on-Don, 344003, Russian Federation

Corresponding author:

Mariya V. Stupina, Associate Professor,
Don State Technical University, Rostov-on-Don, 344003, Russian Federation
E-mail: masamvs@bk.ru

Received on April 07, 2022

Accepted on May 11, 2022

Nowadays, in accordance with the modular rating system adopted in universities, as well as due to the shift in emphasis towards the independent work of students and the construction of individual educational trajectories, the task of accounting for attendance at classes is being updated. The purpose of this paper is to develop the architecture of a software system that allows you to automate the process of confirming the presence of students in the classroom. Based on the analysis of modern works and the technical capabilities of current software solutions, the method of scanning the unique QR code of each training session with the camera of the student's mobile device was chosen as the most optimal means of solving this problem. The paper presents the results of visual modeling of the software system, as well as the structure of the system, which includes the teacher's console in the form of a web application in PHP and a mobile application for the iOS platform developed in Swift. The organization of data storage in the MySQL DBMS, the structure of relations are described. The main algorithms for the operation of the main software modules for scanning and generating a QR code are considered. Test examples of the operation of a web application and a mobile application as the main components of the system are given. With a high degree of confidence, we can assume that the presented system will be effective for solving attendance accounting problems.

Keywords: attendance records, electronic journal, QR code, precedent diagram, system architecture, mobile application, web application, processing server, MySQL DBMS, Swift, PHP

For citation:

Stupina M. V. The Architecture of the Software System for Supporting Attendance Records at the University, *Programmnaya Ingeneria*, 2022, vol. 13, no. 7, pp. 344–353.

DOI: 10.17587/prin.13.344-353

References

1. Novgorodova N. G., Redkina B. A. Automation of educational activities of a vocational pedagogical university, *Mezhdunarodnyj zhurnal eksperimental'nogo obrazovaniya*, 2014, no. 3-2, pp. 15–17 (in Russian).
2. Zafievskij A. V. Automation of educational process management at the university, *Uspekhi sovremennoego estestvoznaniya*, 2010, no. 1, pp. 115–117.
3. Koloskov S. S., Babeshko V. N., Samochadin A. V., Koschlich Yu. A. Methodology for approbation of the implementation of mobile services on the mobile device management platform in universities, *Economy. Informatics*, 2015, no. 19 (216), pp. 152–159 (in Russian).
4. Rajchuk D. Yu., Samochadin A. V., Nosnicyn S. M., Hmeil'kov I. A. A set of mobile means of supporting the educational process, *Informatika, telekommunikacii i upravlenie*, 2014, no. 6 (210), pp. 7–15 (in Russian).
5. Bhalla V., Singla T., Gahlot A., Gupta V. Bluetooth Based Attendance Management System, *IJIET*, 2013, vol. 3, iss. 1, pp. 524–527.
6. More V., Nayak S. Attendance Automation using Near Field Communication (NFC) Technology, *Internat. J. of Scientific & Engineering Research*, 2013, vol. 4, iss. 12, pp. 572–575.
7. Ayu M. A., Ahmad B. I. TouchIn: An NFC Supported Attendance System in a University Environment, *Internat. J. of Information and Education Technology*, 2014, vol. 4, no. 5, pp. 448–453.
8. Sineckij R. M., Gavrikov M. M. Student attendance system based on face recognition algorithms, *Izvestiya vuzov. Severo-Kavkazskij region. Seriya: Tekhnicheskie nauki*, 2016, no. 3 (191), pp. 24–30 (in Russian).
9. Ivanova E. V., Strueva A. Yu. Student attendance system based on face recognition, *Vestnik Yuurgu. Seriya: Vychislitel'naya matematika i informatika*, 2021, vol. 10, no. 4, pp. 60–73 (in Russian).
10. Stupina M. V., Anistratenko K. V., Pazina L. O. Using the QR code as a means of automating the process of accounting for attendance at educational classes, *Journal of Physics: Conference Series*, 2021, vol. 2131, article 022077, available at: <https://iopscience.iop.org/article/10.1088/1742-6596/2131/2/022077/pdf>

Е. Е. Ермилов, вед. аналитик данных, egor.e.ermilov@gmail.com,
А. В. Постный, зам. руководителя методического отдела, alekh.po@gmail.com,
ООО "Интерда", Москва

Персонализация учебного плана посредством применения алгоритмов обучения с подкреплением

Рассмотрено применение марковского процесса принятия решений и рекомендательных алгоритмов для решения проблемы составления индивидуального расписания ученика. Персонализация индивидуального учебного плана способствует повышению качества образования, помогает выстраивать программу с учетом особенностей каждого ученика.

Рассмотренный алгоритм позволяет учитывать накопленную историю оценок учеников за прохождение урока как для выявления максимально эффективных цепочек уроков, так и для составления индивидуальных цепочек уроков.

Ключевые слова: образование, персонализация, искусственный интеллект, машинное обучение, расписание, учебный план, обучение с подкреплением

Введение

В последние десятилетия в сфере образования проявляется все большее стремление к индивидуализации обучения. Проблемам индивидуализации и дифференциации процесса обучения посвящен ряд педагогических работ И. Э. Унт, А. А. Кирсанова, Г. Ф. Суворовой, С. Д. Шевченко и других авторов [1]. Российская педагогическая энциклопедия определяет термин "индивидуализация обучения" как "организация учебного процесса с учетом индивидуальных особенностей учащихся, которая позволяет создать оптимальные условия для реализации потенциальных возможностей каждого ученика". Индивидуализация обучения осуществляется в условиях коллективной учебной работы в рамках общих задач и содержания обучения [2].

Таким образом, с одной стороны, индивидуализация обучения предполагает подход к каждому ученику с учетом его особенностей в целях повышения эффективности обучения. С другой стороны, предполагаются некие рамки общих задач и содержания обучения. В случае общего образования такими рамками являются Федеральные государственные образовательные стандарты (ФГОС) [3–5].

Когда процесс обучения происходит один на один с педагогом (частные учителя, репетиторы,

тьюторы), индивидуализация обучения не составляет технических сложностей. Результат и эффективность обучения определяются квалификацией педагога. Он сам определяет параметры обучения — учебный план ученика, темп прохождения материала, порядок изложения тем.

В случае групповых занятий, в том числе в классах обычной очной школы, возникает проблема с индивидуализацией. Невозможно подобрать такие параметры обучения, которые бы учитывали все индивидуальные особенности каждого ученика. В этом случае чаще всего индивидуальное обучение сочетается с дифференцированным обучением [1]. То есть ученики объединяются в группы и для этих групп подбирается свой учебный план, который по возможности максимально учитывает особенности учеников данной группы.

В онлайн-образовании, когда ученикам предоставляется набор материалов для обучения (видеоматериалы, конспекты, тесты и пр.), существует возможность для индивидуализации обучения. Несмотря на то что набор материалов ограничен, можно регулировать учебный план ученика следующими способами:

- добавлять или убирать некоторые темы из учебного плана;
- изменять количество материалов по данной теме (например, демонстрировать только неко-

торые видеоматериалы или давать сокращенный набор тестов);

- изменять порядок изучения тем.

С учетом числа обучающихся составление таких учебных планов для каждого ученика вручную потребует большого количества ресурсов. В связи с этим возникает необходимость разработки алгоритма, который на основе некоторых данных о каждом ученике мог бы составить индивидуальный план обучения.

Цель нашего исследования — разработка алгоритма, который может составлять индивидуальный учебный план или программу обучения автоматически, без привлечения дополнительных человеческих ресурсов.

Необходимо, чтобы алгоритм умел учитывать накопленную историю оценок учеников и методические рекомендации. История учеников должна учитываться при автоматической персонализации наряду с дополнительными условиями или ограничениями, например, такими как ФГОС.

В рамках данной работы индивидуализацию учебного плана будем связывать только с изменением порядка изучения уроков, а также с включением или исключением некоторых тем из программы обучения. Алгоритм не будет учитывать изменение объема материалов в рамках одного урока.

Обзор работ по тематике

В настоящий момент персонализация учебного плана не очень глубоко исследована, хотя и предпринимались некоторые попытки это сделать.

Во многих работах исследуется проблема составления оптимального расписания для учебных заведений [6–11]. В рассмотренных работах оптимизация учитывает больше ресурсы учебного заведения (число преподавателей, наличие аудиторий и т. п.), а не успеваемость учеников.

В некоторых работах все же учитываются возможности студентов [12, 13], но не делается персональный учебный план.

В работе [14] подробно рассказано о необходимости участия алгоритмов машинного обучения и искусственного интеллекта в учебном процессе. Доказано, что с темпами современного мира нужно не просто автоматизировать рутинную работу, но и делать это с помощью более прогрессивных инструментов. Однако в работе [14] не дано ни-

каких конкретных примеров реализации алгоритмов.

В статье [15] дан метод, похожий на тот, что будет рассмотрен в настоящей работе. Но стоит заметить, что хоть авторы работы [15] и предлагают использовать обучение с подкреплением для составления индивидуального расписания, они не применяют данный подход на реальных данных. Они лишь провели симуляцию на 20 вымышленных учениках со случайно сгенерированными параметрами.

В настоящей работе рассматривается применение обучения с подкреплением на реальных данных реальных учеников, проходящих обучение на платформе школьного онлайн-образования ИнтернетУрок [16].

Основная часть исследования

Для моделирования было условлено, что процесс обучения (последовательного прохождения уроков) — это марковский процесс принятия решения [17]. Пример марковского процесса представлен на рисунке.

Для описания марковского процесса нужно определить следующие множества:

- S — множество состояний;
- A — множество действий;
- R — множество вознаграждений.

Пусть имеется N уроков, тогда принимаем, что множество уроков — это и есть множество состояний S .

Любое действие — это решение, какой урок пройти следующим. Это и составляет множество действий A .

Получаем, что размер множеств S и A совпадает и равен числу уроков в библиотеке платформы ИнтернетУрок [16].

Вознаграждения в нашем случае — оценки от 2 до 5, нормализованные относительно средней оценки по классу и предмету с помощью формулы:

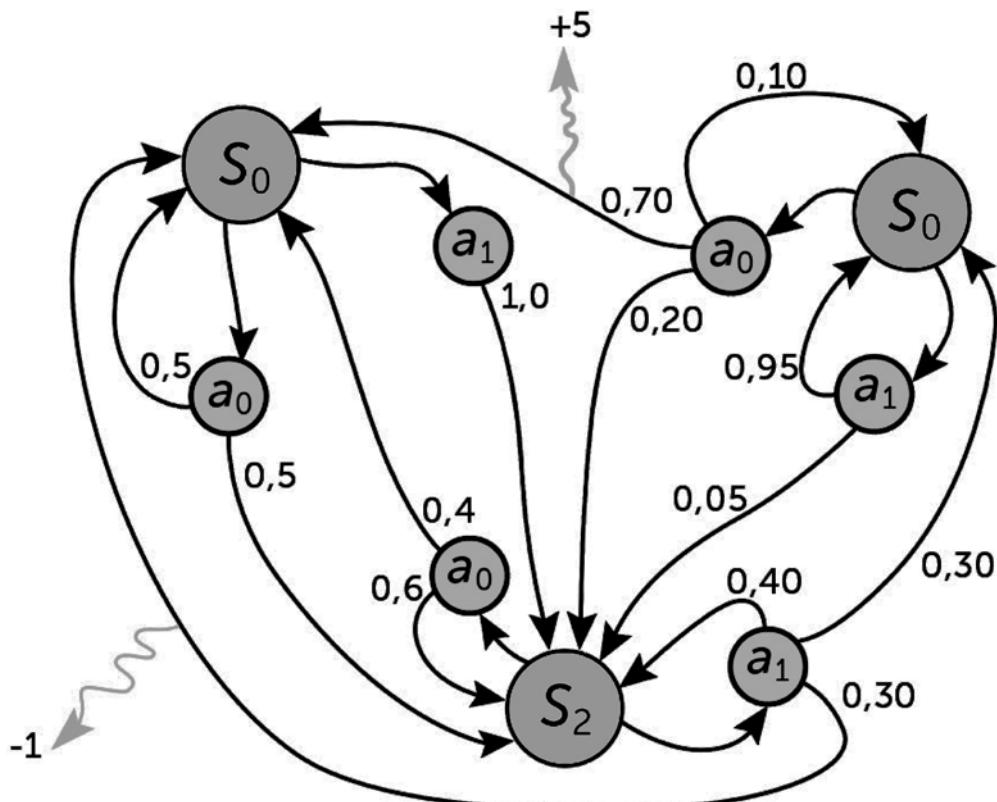
$$\begin{aligned} \text{Reward_norm} = \\ = (\text{Reward}_i - \text{Reward_mean}) / \text{Reward_std}, \end{aligned}$$

где Reward_i — текущая оценка у ученика;

Reward_mean — средняя оценка по классу и предмету ученика;

Reward_std — среднее квадратическое отклонение оценок по классу и предмету ученика.

Получаем множество вознаграждений R , рассчитанное по формуле выше.



Пример классического марковского процесса принятия решения:
 S — состояния; a — действия

Приведем пример процесса обучения ученика.

- Решил перейти к уроку № 194.
- Прошел урок № 194.
- Получил за этот урок оценку 5.
- Решил перейти к уроку № 279.
- Прошел урок № 279.
- Получил за этот урок оценку 4.
- Решил перейти к уроку № 1736.
- Прошел урок № 1736.
- Получил за этот урок оценку 5.
- Решил перейти к уроку № 8362.
- Прошел урок № 8362.
- Получил за этот урок оценку 2.

Формально на языке марковского процесса принятия решений это можно описать так:

$A_{194}, S_{194}, R = 5 \gg$

$\gg A_{279}, S_{279}, R = 4 \gg$

$\gg A_{1736}, S_{1736}, R = 5 \gg$

$\gg A_{8362}, S_{8362}, R = 2 \gg$

Технически на платформе ИнтернетУрок возможно пройти урок заново или пройти несколько уроков по кругу. Но накопленные данные с оценками содержат только дату последнего прохождения урока. Поэтому для упрощения считаем, что в модели повторы уроков не учитывается.

Имеется массив исторических данных о поведении всех учеников (прохождение уроков и получение оценок). Из этих данных формируем Q -таблицу $Q[s, a]$ [18].

В таблице $Q[s_i, a_j]$ хранится оптимальное вознаграждение, если в состоянии s_i выполнить действие a_j . То есть после урока номер i выполнить урок номер j .

При прохождении по массиву исторических данных с каждой строчкой Q -таблица меняется следующим образом:

$$Q_{\text{new}}[s', a'] = Q[s', a'] + \\ + LF(r + DF * \max(Q, s) - Q[s', a'])$$

где LF — фактор обучения; чем он выше, тем сильнее агент [18] доверяет новой информации;

DF — фактор дисконтирования; чем он меньше, тем меньше агент задумывается о выгоде от будущих своих действий;

r — нормированная оценка за урок;

$\max(Q, s)$ — максимальное уже существующее в Q -таблице вознаграждение, возможное после перехода к выбранному уроку.

После нескольких итераций прохождения циклом по массиву исторических данных получаем таблицу с оптимальными путями прохождения уроков. Интерпретировать ее нужно следующим образом.

Текущий шаг: $s_{current}$.

Выбираем в таблице $Q[s, a]$ с $s = s_{current}$, в полученной строке выбираем максимальное значение и принимаем соответствующее действие a .

Таким образом для каждого урока можно получить оптимальный (с точки зрения максимизации оценки) следующий шаг.

В следующем разделе для полученной Q -таблицы $Q[s, a]$ будем использовать другие индексы $Q[lesson_current, lesson_next]$. Сама таблица и ее содержание остаются неизменными. Эта таблица содержит информацию о том, какой урок ($lesson_next$) изучить дальше после прохождения текущего урока ($lesson_current$).

Расчет рекомендаций

В оригинальной модели Q-обучения выбор дальнейшего действия носит вероятностный характер. В нашей же адаптированной версии выбор действия носит рекомендательный характер. Алгоритм не генерирует образовательный маршрут полностью, но помогает ученикам справляться с локальными трудностями в обучении.

Оценки по всем пользователям также нормализуются по средней оценке по классу и предмету.

Рекомендации рассчитываются индивидуально по каждому пользователю. Для этого формируем два списка полученных оценок:

- хорошие оценки (нормализованная оценка больше нуля);
- плохие оценки (нормализованная оценка меньше нуля).

В каждом списке оставляем последние пять просмотренных уроков.

Рекомендации для дальнейшего изучения формируются по приведенному далее алгоритму.

Шаг 1. Формирование списка пяти последних хороших оценок.

Шаг 2. Из $Q[lesson_current, lesson_next]$ выбор пяти векторов с $lesson_current$ из предыдущего шага.

Шаг 3. Суммирование выбранных значений для каждого урока.

Шаг 4. Сортировка списка по убыванию полученных скоров.

Шаг 5. Запись первых трех уроков из этого списка в рекомендации.

Рекомендации для повторения формируются согласно приведенному ниже алгоритму.

Шаг 1. Формирование списка пяти последних плохих оценок.

Шаг 2. Из $Q[lesson_current, lesson_next]$ выбор пяти векторов с $lesson_next$ из предыдущего шага.

Шаг 3. Суммирование выбранных значений для каждого урока.

Шаг 4. Сортировка списка по убыванию полученных скоров.

Шаг 5. Запись первых трех уроков из этого списка в рекомендации.

Заключение

Разработан алгоритм, который дает возможность проводить персонализацию учебной программы любому ученику с накопленной историей оценок. Такой подход не требует ресурсов методистов для ручного составления планов, поэтому данная образовательная услуга становится доступной для широкого круга учеников.

Также алгоритм обладает большой гибкостью и скоростью пересчета рекомендаций. Индивидуальное расписание можно пересматривать после получения любой новой оценки.

В настоящее время проводятся работы по проверке качества данного алгоритма. Для этого на платформу ИнтернетУрок встраивается блок рекомендаций. После его запуска будет проводиться А/В-тестирование для определения эффективности алгоритма.

Начальные исследования индивидуальных цепочек уроков дали полезную информацию о составлении расписания, зависимости успеваемости от последовательности уроков.

Приведенный в статье алгоритм может найти применение как в библиотеке уроков платформы ИнтернетУрок, где сейчас нет фиксированного

учебного плана, так и в Домашней Школе [19] на этой же платформе, где обучение построено по расписанию.

Список литературы

1. Юрловская И. А. Индивидуализация обучения как одна из тенденций современного образования // Вектор науки ТГУ. Серия: Педагогика, психология. 2013. № 3. С. 292–294.
2. Российская педагогическая энциклопедия. М: Большая Российская Энциклопедия. В 2. томах. Т. 1: А-М. / Под ред. В. Г. Панова. 1993.
3. Приказ Министерства просвещения Российской Федерации № 286 от 31.05.2021 Об утверждении федерального государственного образовательного стандарта начального общего образования. URL: <http://publication.pravo.gov.ru/Document/View/0001202107050028> (дата обращения: 01.04.2022).
4. Приказ Министерства просвещения Российской Федерации № 287 от 31.05.2021 Об утверждении федерального государственного образовательного стандарта начального общего образования. URL: <http://publication.pravo.gov.ru/Document/View/0001202107050027> (дата обращения: 01.04.2022).
5. Приказ Министерства образования и науки Российской Федерации № 413 от 17.05.2012 Об утверждении федерального государственного образовательного стандарта среднего общего образования. URL: <https://docs.edu.gov.ru/document/bf0ceabdc94110049a583890956abbfa/> (дата обращения: 01.04.2022).
6. Яковлева М. С., Вайтекунене Е. Л. Автоматизация процесса составления расписания учебных занятий // Решетневские чтения. 2016. Том 2. С. 176–178.
7. Хасухаджиев А. С.-А., Сибикина И. В. Обобщенный алгоритм составления расписания в вузе с учетом новых требований федеральных государственных образовательных стандартов // Вестник АГТУ. Серия: Управление, вычислительная техника и информатика. 2016. № 3. С. 78–86.
8. Галаванова Ю. И. Обзор современных методов в автоматизации составления расписания в организациях общего образования // Достижения науки и образования. 2018. № 3 (25). С. 15–17.
9. Aytug H., Bhattacharyya S., Koehler G., Snowdon J. A Review of Machine Learning in Scheduling // Engineering Management, IEEE Transactions on. 1994. Vol. 41, Is. 2. P. 165–171. DOI: 10.1109/17.293383.
10. Nanda A., Pai M., Gole A. An Algorithm to Automatically Generate Schedule for School Lectures Using a Heuristic Approach // International Journal of Machine Learning and Computing. 2012. Vol. 2, No. 4. P. 492–495. DOI: 10.7763/IJMLC.2012.V2.I4.492.
11. Lovelace A. L. On the complexity of scheduling university courses. Thesis Presented to the Faculty of California Polytechnic State University, San Luis Obispo In Partial Fulfillment of the Requirements for the Degree Master of Science in Computer Science. URL: <https://digitalcommons.calpoly.edu/cgi/viewcontent.cgi?referer=&httpsredir=1&article=1255&context=theses>
12. Верёвкин В. И., Исмагилова О. М., Атавин Т. А. Автоматизированное составление расписания учебных занятий вуза с учетом трудности дисциплин и утомляемости студентов // Доклады ТУСУР. 2009. № 1 (19). Ч. 1. С. 221–225.
13. Маstryев Ф. А. Автоматизация формирования учебных программ и контроля их исполнения в системе высшего профессионального образования // Прикладная информатика. 2006. № 6. URL: http://www.appliedinformatics.ru/r/articles/article/index.php?article_id_4=766
14. Personalized Learning. URL: <https://www.valamis.com/hub/personalized-learning>
15. Shawky D., Badawi A. A Reinforcement Learning-Based Adaptive Learning System // The International Conference on Advanced Machine Learning Technologies and Applications (AMLTA2018). AMLTA 2018. Advances in Intelligent Systems and Computing, Springer, Cham. 2018. Vol. 723. P. 221–231. DOI: 10.1007/978-3-319-74690-6_22.
16. Образовательная платформа Библиотека видеоуроков по школьной программе ИнтернетУрок. URL: <https://interneturok.ru/>
17. Bellman R. A Markovian Decision Process// Journal of Mathematics and Mechanics. 1957. Vol. 6, Is. 5. P. 679–684.
18. Watkins C., Dayan P. Q-learning// Machine Learning. 1992. Vol. 8, Is. 3–4. P. 279–292. DOI: 10.1007/BF00992698.
19. Образовательная платформа Домашняя школа ИнтернетУрок. URL: <https://home-school.interneturok.ru/>

Curriculum Personalization through the Application of Reinforcement Learning Algorithms

E. E. Ermilov, Lead Data Scientist, egor.e.ermilov@gmail.com, **A. V. Postnyi**, Deputy Head of Methodological Department, alekh.po@gmail.com, Limited liability company "Interda", Moscow, 117639, Russian Federation

Corresponding author:

Egor E. Ermilov, Lead Data Scientist,
Limited liability company "Interda", Moscow, 117639, Russian Federation
E-mail: egor.e.ermilov@gmail.com

Received on April 11, 2022
Accepted on May 11, 2022

Individual curriculum Personalization contributes to improving the education quality. It takes into account the individual issues of each student and changes the program accordingly. This article discusses the application of recommender algorithms and Markov decision-making process to the problem of making an individual student's schedule.

The algorithm is based on the collected historical data of assessments both to identify the most effective lesson chains among all students, and to combine individual lesson chains.

Keywords: education, personalization, artificial intelligence, machine learning, curriculum, schedule, reinforcement learning

For citation:

Ermilov E. E., Postnyi A. V. Curriculum Personalization through the Application of Reinforcement Learning Algorithms, *Programmnaya Ingeneriya*, 2022, vol. 13, no. 7, pp. 354–359.

DOI: 10.17587/prin.13.354-359

References

1. **Yurlovskaya I. A.** Individualization of learning as one of the trends in modern education, *Vector of science TSU. Series: Pedagogy, psychology*, 2013, no. 3, pp. 292–294 (in Russian).
2. **Russian Pedagogical Encyclopedia.** Moscow, Great Russian Encyclopedia / Ed. V. G. Panova, 1993, vol. 1 (in Russian).
3. **Order of the Ministry of Education of the Russian Federation No. 286 dated May 31, 2021:** On the approval of the federal state educational standard of primary general education, available at: <http://publication.pravo.gov.ru/Document/View/0001202107050028> (in Russian).
4. **Order of the Ministry of Education of the Russian Federation N 287 dated May 31, 2021:** On the approval of the federal state educational standard of primary general education, available at: <http://publication.pravo.gov.ru/Document/View/0001202107050027> (in Russian).
5. **Order of the Ministry of Education and Science of the Russian Federation N 413 dated May 17, 2012:** On the approval of the federal state educational standard of secondary general education, available at: <https://docs.edu.gov.ru/document/bf0ceabdc94110049a583890956abbfa> (in Russian).
6. **Yakovleva M. S., Vaitekunene E. L.** Automation of the process of scheduling training sessions, *Reshetnev readings*, 2016, vol. 2, pp. 176–178 (in Russian).
7. **Khasukhadzhiev A. S.-A., Sibikina I. V.** A generalized algorithm for scheduling at a university taking into account the new requirements of federal state educational standards, *Bulletin of ASTU. Series: Management, Computer Engineering and Informatics*, 2016, no. 3, pp. 78–86 (in Russian).
8. **Galavanova Y. I.** Review of modern methods in automation of scheduling in organizations of general education, *Achievements of science and education*. 2018, no. 3 (25), pp. 15–17 (in Russian).
9. **Aytug H., Bhattacharyya S., Koehler G., Snowden J.** A Review of Machine Learning in Scheduling, *Engineering Management, IEEE Transactions on*, 1994, vol. 41, is. 2, pp. 165–171, DOI: 10.1109/17.293383.
10. **Nanda A., Pai M., Gole A.** An Algorithm to Automatically Generate Schedule for School Lectures Using a Heuristic Approach, *International Journal of Machine Learning and Computing*, 2012, vol. 2, no 4, pp. 492–495, DOI: 10.7763/IJMLC.2012.V2.174.
11. **Lovelace A. L.** On the complexity of scheduling university courses. Thesis Presented to the Faculty of California Polytechnic State University, San Luis Obispo In Partial Fulfillment of the Requirements for the Degree Master of Science in Computer Science, available at: <https://digitalcommons.calpoly.edu/cgi/viewcontent.cgi?referer=&httpsredir=1&article=1255&context=theses>
12. **Veryovkin V. I., Ismagilova O. M., Atavin T. A.** Automated scheduling of university classes, taking into account the difficulties of disciplines and fatigue of students, *Doklady TUSUR*, 2009, no. 1 (19), part 1, pp. 221–225 (in Russian).
13. **Mastyaev F. A.** Automation of the formation of training programs and control of their execution in the system of higher professional education, *Applied Informatics*, 2006, no. 6, available at: http://www.appliedinformatics.ru/r/articles/article/index.php?article_id_4=766 (in Russian).
14. **Personalized Learning**, available at: <https://www.valamis.com/hub/personalized-learning>
15. **Shawky D., Badawi A.** A Reinforcement Learning-Based Adaptive Learning System, *The International Conference on Advanced Machine Learning Technologies and Applications (AMLTA2018). AMLTA 2018. Advances in Intelligent Systems and Computing*, Springer, Cham., 2018, vol. 723, pp. 221–231, DOI: 10.1007/978-3-319-74690-6_22.
16. **InternetUrok** educational platform Library, available at: <https://interneturok.ru/>
17. **Bellman R.** A Markovian Decision Process, *Journal of Mathematics and Mechanics*, 1957, vol. 6, is. 5, pp. 679–684.
18. **Watkins C., Dayan P.** Q-learning, *Machine Learning*, 1992, vol. 8, is. 3–4, pp. 279–292, DOI: 10.1007/BF00992698.
19. **InternetUrok** educational platform Home school, available at: <https://home-school.interneturok.ru/>

XVI Международная конференция инженерии программного обеспечения "Software Engineering Conference" (SECR) 17—18 марта 2023 г., Москва, отель «Holiday Inn Сокольники»

SECR — одно из старейших и авторитетных ИТ-событий, посвященных индустрии разработки ПО.

Уникальность конференции — в широком охвате тем. SECR представляет разрез всей ИТ-индустрии и преподносит его участникам в виде 100+ различных докладов, мастер-классов, дискуссий за два дня. Современные подходы, тренды, прогнозы, экспертные мнения — все это SECR. Конференция интересна без исключения всем, кто вовлечен в разработку ПО.

Тематики докладов

Принимаются доклады на темы, относящиеся к разработке ПО. Вот **некоторые** из них.

- Programming Technologies and Tools
- DevOps
- AI; Intelligent Apps and Analytics; Intelligent Things
- IoT; Digital Twins
- Data Science; Big Data
- Blockchain; Event-Driven
- Cloud; Cloud to the Edge
- High-Loaded Systems
- Embedded Systems; Hardware and Low-level Programming
- Open Source Technologies & Community
- Voice Interfaces; Natural Languages Text Processing; Conversational Platforms
- Security; Continuous Adaptive Risk and Trust
- VR, AR, Immersive Experience
- Software Quality Assurance
- System and Business Analysis and Requirements Engineering
- Project and Product Management (Classical & Agile)
- Knowledge Management
- Software Architecture
- Continuous Professional Development: From Classical to Corporate University
- Business and Entrepreneurship
- Human-Computer Interaction; User Experience, Usability

Если в списке нет вашей темы, но она относится к разработке ПО и может быть интересна аудитории, предлагайте доклад на рассмотрение программным комитетом.

Типы докладов

Секционные доклады

Они составляют большинство докладов на конференции. Секционные доклады отбираются экспертами Программного комитета на конкурсной основе. Для каждого устного доклада в программе конференции будет отведены 40 мин (или 1,5 ч — два слота 40 мин + 40 мин). Это время включает как непосредственно выступление автора(ов) доклада, так и ответы на вопросы аудитории.

Блиц-доклады

Короткие (15—20 минут) выступления на тему, связанную с тематикой конференции или близкую ей.

Мастер-классы

Практические обучающие выступления продолжительностью 1,53 ч.

Круглые столы

Обсуждение экспертным составом выбранной тематики с модерацией (длительность до 1,5 ч). Свои позиции высказывают сторонники и противники.

Подробности: <https://secrus.ru>

ООО "Издательство "Новые технологии". 107076, Москва, ул. Матросская Тишина, д. 23, стр. 2
Технический редактор Е. М. Патрушева. Корректор А. В. Чугунова.

Сдано в набор 16.06.2022 г. Подписано в печать 21.07.2022 г. Формат 60×88 1/8. Заказ Р1721
Цена свободная.

Оригинал-макет ООО "Адвансед солюшнз". Отпечатано в ООО "Адвансед солюшнз".
119071, г. Москва, Ленинский пр-т, д. 19, стр. 1. Сайт: www.aov.ru

Organizers

FRC CSC of RAS
ISP RAS
MSU
ETU "LETI"
BMSTU

Supported by

Russian Academy
of Sciences

General chair

Diveev A.I., Russia

Vice chairs

Avetisyan A.I., Russia
Fomichev V.V., Russia
Ilin A.V., Russia
Pereira F.L., Portugal
Sofronova E.A., Russia

**Call for Papers**

Federal Research Center "Computer Science and Control" of the RAS, Ivannikov Institute for System Programming of the RAS, Lomonosov Moscow State University, St. Petersburg Electrotechnical University "LETI", and Bauman Moscow State Technical University invite you to participate in the 15th International Conference «Intelligent Systems – 2022» (INTELS'22), that will take place in **Moscow, Russia, December 14–16, 2022**.

INTELS'22 will focus on areas of intelligent systems and artificial intelligence and their application to sustainable development. INTELS'22 provides a leading international forum that brings together science and industry from diverse fields with the purpose of exploring the fundamentals, technologies, achievements and impact of Artificial Intelligence.

Venue:

Russian Academy of Sciences
32A, Leninskiy prospect, Moscow, Russia

Format: Hybrid (follow the current information on conference site www.intels-conf.ru)

Topics of interest include but not limited to

- | | |
|---|---|
| Theoretical foundation of intelligent systems | Adaptive Security Architecture for intelligent systems |
| Logic inference and many-valued logic | Autonomous Agents and Things |
| Intelligent and AI-based control | Big Data Analysis |
| Optimal control | Data mining: WEB mining, image mining |
| Optimization | Deep Learning |
| Artificial neural networks | Design and implementation of interdisciplinary systems for computer mathematics |
| Fuzzy systems | Computer algebra and symbolic computation in theorem proving systems |
| Machine Learning | Knowledge representation and automated reasoning |
| Symbolic regression and genetic programming | Decision support systems |
| Patterns recognition | Multi-agent systems |
| Evolutionary computations | Swarm intelligence in robotics |
| Bionic systems | Intelligent systems in robotics |
| Cognitive systems | |
| Brain-machine interface systems | |
| Identification of dynamical plants | |
| Intelligent aerospace systems | |
| Intelligent systems in bioinformatics | |

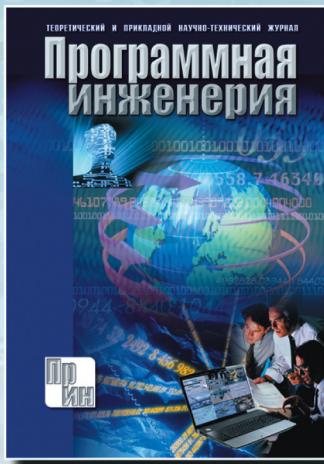
Proceedings of INTELS'22 will be published in Procedia Computer Science (Elsevier, ISSN: 1877-0509), indexed in SCOPUS, Web of Science, INSPEC, Conference Proceedings Citation Index

Important dates:

- Submission of Full Papers:
Notification of Acceptance:
Submission of Final Manuscripts:
Registration Date:
Conference:

- June 30, 2022**
October 1, 2022
October 15, 2022
October 15, 2022
December 14-16, 2022

Издательство «НОВЫЕ ТЕХНОЛОГИИ» выпускает научно-технические журналы



Теоретический и прикладной научно-технический журнал **ПРОГРАММНАЯ ИНЖЕНЕРИЯ**

В журнале освещаются состояние и тенденции развития основных направлений индустрии программного обеспечения, связанных с проектированием, конструированием, архитектурой, обеспечением качества и сопровождением жизненного цикла программного обеспечения, а также рассматриваются достижения в области создания и эксплуатации прикладных программно-информационных систем во всех областях человеческой деятельности.

Подписной индекс по Объединенному каталогу
«Пресса России» – 22765



Подписной индекс по
Объединенному каталогу
«Пресса России» – 72656

Ежемесячный теоретический и прикладной научно- технический журнал **ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ**

В журнале освещаются современное состояние, тенденции и перспективы развития основных направлений в области разработки, производства и применения информационных технологий.

Междисциплинарный
теоретический и прикладной
научно-технический журнал

НАНО- И МИКРОСИСТЕМНАЯ ТЕХНИКА

В журнале освещаются современное состояние, тенденции и перспективы развития нано- и микросистемной техники, рассматриваются вопрос разработки и внедрения нано микросистем в различные области науки, технологии и производства.



Подписной индекс по
Объединенному каталогу
«Пресса России» – 79493



Подписной индекс по
Объединенному каталогу
«Пресса России» – 79492

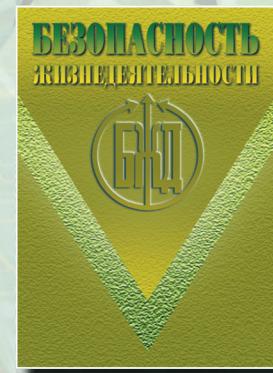
Ежемесячный теоретический и прикладной научно-технический журнал **МЕХАТРОНИКА, АВТОМАТИЗАЦИЯ, УПРАВЛЕНИЕ**

В журнале освещаются достижения в области мехатроники, интегрирующей механику, электронику, автоматику и информатику в целях совершенствования технологий производства и создания техники новых поколений. Рассматриваются актуальные проблемы теории и практики автоматического и автоматизированного управления техническими объектами и технологическими процессами в промышленности, энергетике и на транспорте.

Научно-практический
и учебно-методический журнал

БЕЗОПАСНОСТЬ ЖИЗНEDЕЯТЕЛЬНОСТИ

В журнале освещаются достижения и перспективы в области исследований, обеспечения и совершенствования защиты человека от всех видов опасностей производственной и природной среды, их контроля, мониторинга, предотвращения, ликвидации последствий аварий и катастроф, образования в сфере безопасности жизнедеятельности.



Подписной индекс по
Объединенному каталогу
«Пресса России» – 79963

Адрес редакции журналов для авторов и подписчиков:

107076, Москва, ул. Матросская Тишина, д. 23, стр. 2, оф. 45. Издательство "НОВЫЕ ТЕХНОЛОГИИ".

Тел.: (499) 270-16-52. E-mail: antonov@novtex.ru