

Санкт–Петербургский государственный университет
Математическое обеспечение и администрирование информационных систем
Информационно-аналитические системы

Ким Юния Александровна

Веб-редактор для платформы REAL.NET

Отчёт по учебной практике

Научный руководитель:
доцент кафедры системного программирования,
к.т.н. Литвинов Ю. В.

Санкт-Петербург
2021

Содержание

Введение	3
Постановка задачи	4
1. Обзор	5
1.1. Существующие решения	5
1.1.1 TRIK Studio	5
1.1.2 Diagrams.net	6
1.1.3 Noodle	7
1.1.4 GenMyModel	8
1.1.5 Visual Paradigm	8
1.1.6 YEd Live	9
1.2. Существующие библиотеки	10
1.2.1 STORM React Diagrams	10
1.2.2 React Flow	11
1.2.3 React d3 graph	11
1.2.4 JointJS	12
1.2.5 GoJS	12
2. Описание решения	13
3. Апробация	19
Заключение	21
Список литературы	22

Введение

В наше время наука не стоит на месте и каждый день появляются новые технологии и развиваются уже существующие. Вместе с тем растёт и потребность в использовании таких разработок людьми, которые далеки от мира высоких технологий, а также в упрощении сложных задач. Визуальные предметно-ориентированные языки программирования хорошо себя зарекомендовали в решении подобных задач, например, см. работы [1] [2]¹. Такие языки направлены на создание программ для конкретной предметной области, что увеличивает эффективность решаемых ими проблем. Помимо этого, они позволяют разрабатывать программы, манипулируя графическими объектами, а не печатая текстовые команды, что делает их очень удобными для пользователей любого контингента.

Поскольку создание отдельного редактора для каждого конкретного языка трудо- и времязатратно, существуют единые универсальные платформы, с помощью которых можно быстро и удобно создавать и использовать такие языки программирования. Несмотря на большое количество подобных платформ, которые представлены и в виде десктопных инструментов, и в виде веб-сервисов, создание новых языков с их помощью достаточно сложно. С целью значительно облегчить этот процесс на кафедре системного программирования математико-механического факультета СПбГУ ведётся разработка среды для предметно-ориентированного моделирования REAL.NET [3].

Однако в настоящий момент у платформы нет полноценного веб-редактора, который позволил бы разработчикам и пользователям работать с языками программирования, созданными с помощью REAL.NET. Веб-приложения, в отличие от десктопных программ, не требуют для работы скачивания и установки каких-либо файлов или программного обеспечения. Кроме того, они не требовательны к ресурсам устройства, а также более удобны для обновления.

Поэтому необходимо разработать веб-редактор для REAL.NET.

¹Официальная страница проекта Scratch.
URL: <https://scratch.mit.edu> (дата обращения: 06.06.2021)

Постановка задачи

Целью данной работы является реализация веб-редактора для платформы REAL.NET, согласно требованиям «универсального редактора» для произвольных графовых языков, создаваемых с помощью системы.

Для достижения данной цели были поставлены следующие задачи.

- Обзор существующих редакторов.
- Формирование требований к веб-редактору.
- Обзор существующих библиотек и выбор библиотеки для реализации.
- Реализация веб-редактора.
- Апробация веб-редактора.

1. Обзор

1.1 Существующие решения

В данном разделе приводится обзор некоторых существующих редакторов, включая некоторые десктопные приложения. Большинство обзриваемых инструментов было выбрано как одни из самых популярных в соответствующей сфере. Остальные редакторы были взяты в рассмотрение из-за того, что они созданы на основе платформ, которые были разработаны на математико-механическом факультете СПбГУ.

В процессе обзора анализировались доступные пользователю функции, возможность совершения и способ реализации различных действий с элементами модели, внешнее устройство редактора, а также удобство использования. На основании полученных результатов были сформулированы требования к реализуемому веб-редактору.

1.1.1 TRIK Studio

TRIK Studio² – бесплатная среда программирования для роботов, созданная на базе платформы предметно-ориентированного моделирования QReal [4] [5], которая была разработана на математико-механическом факультете СПбГУ в процессе работы исследовательской группы, в рамках которой ведётся работа над этой практикой. Данная среда позволяет решать различные задачи используя последовательности различных картинок, а также с помощью текстового языка. Кроме того, стоит отметить, что TRIK Studio является десктопной программой.

Редактор среды, как и большинство всех существующих редакторов, состоит из сцены, палитры и окна свойств. Сцена занимает большую часть редактора и представляет собой область для создания моделей, в которую можно перетащить элементы из палитры, той части редактора, где отображаются элементы. В окне свойств пользователь может редактировать значения различных свойств элементов модели.

²Официальная страница проекта TRIK Studio.
URL: <https://trikset.com/products/trik-studio> (дата обращения: 06.06.2021)

Особенностями редактора TRIK Studio являются специфичность возможных операций с элементами, так как данная среда предназначена для работы с определённым языком программирования в области робототехники. Внешний вид редактора можно увидеть на рис. 1.

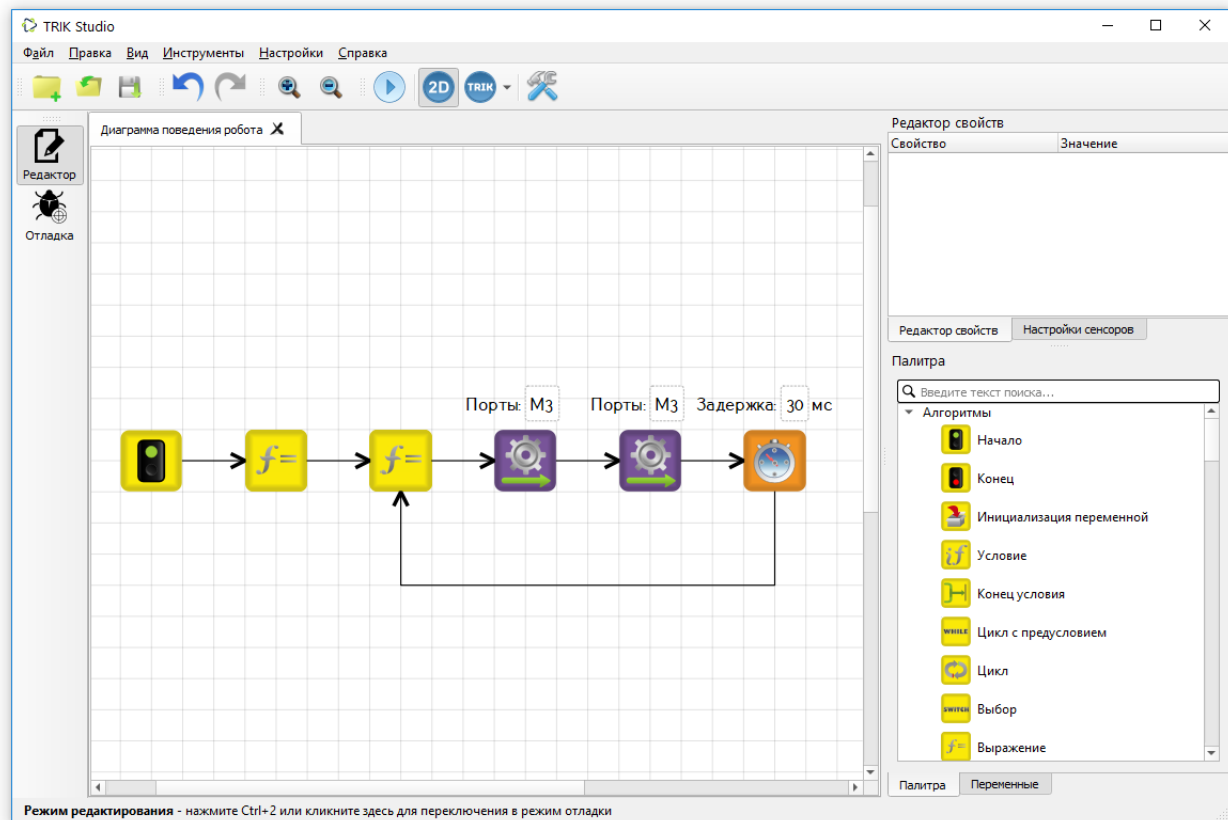


Рис. 1: Редактор TRIK Studio (источник³)

1.1.2 Diagrams.net

Diagrams.net⁴ – это бесплатный веб-сервис, ранее именуемый как «draw.io». Он предназначен для создания и редактирования различных диаграмм и схем. Инструмент был разработан с использованием библиотеки mxgraph⁵, на основе которой был создан первый прототип веб-редактора

³Упражнения. Справочный центр ТРИК.

URL: <https://help.trikset.com/studio/exercises> (дата обращения: 06.06.2021)

⁴Официальная страница проекта diagrams.net.

URL: <https://www.diagrams.net> (дата обращения: 06.06.2021)

⁵GitHub репозиторий mxgraph.

URL: <https://github.com/jgraph/mxgraph> (дата обращения: 06.06.2021)

для платформы REAL.NET. Редактор веб-сервиса, который представлен на рис. 2, отличается широким разнообразием возможностей и свободой действий, что обосновано тем, что diagrams.net является средой для рисования, а не программирования.

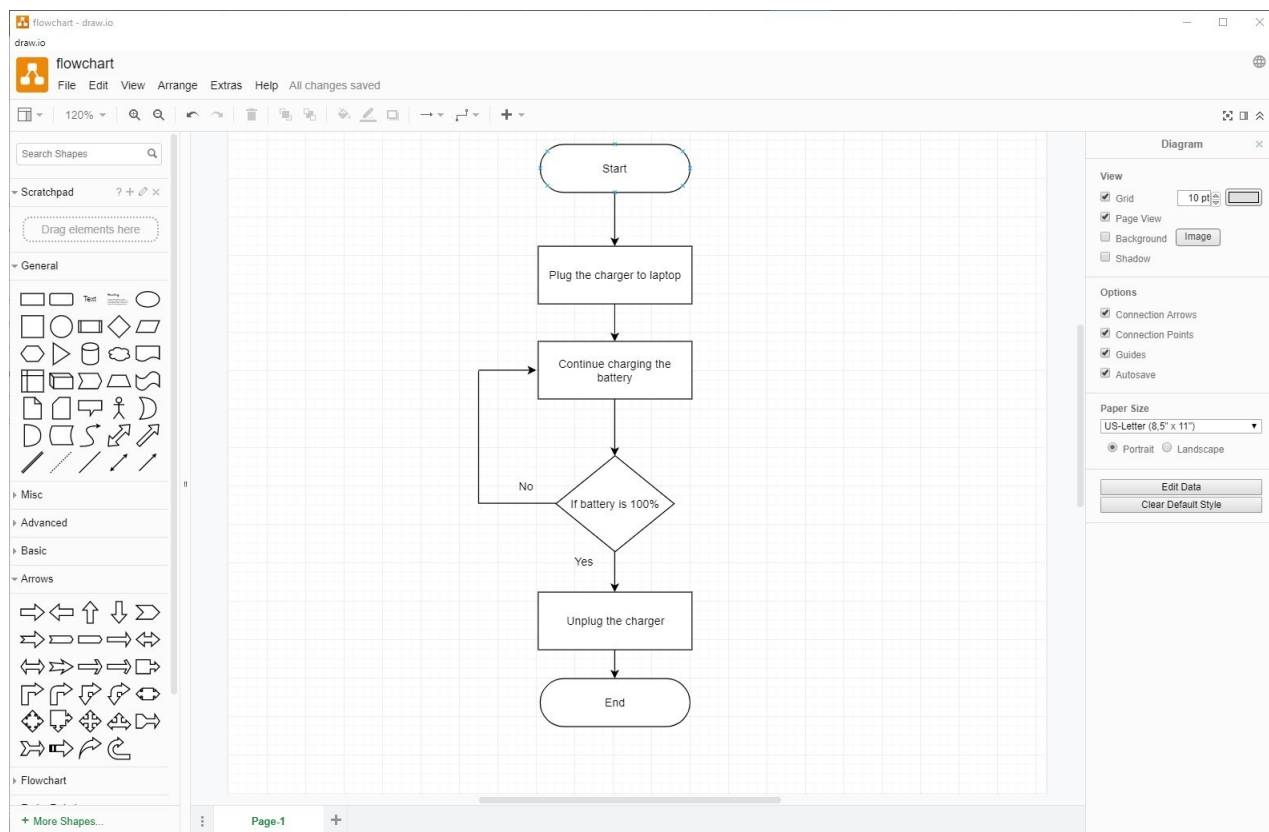


Рис. 2: Редактор diagrams.net (источник⁶)

1.1.3 Noodle

Noodle⁷ – пользовательский интерфейс для визуального программирования, а также фреймворк, написанный на языке PureScript⁸. Он является наследником библиотеки RPD⁹ (Reactive Patch Development, реактивная раз-

⁶Ashwin. Draw.io is a free Flowchart and diagram creation software for Windows, Linux, macOS and your browser.

URL: <https://www.ghacks.net/2019/09/07/draw-io-free-flowchart-creation-software-windows-linux-macos> (дата обращения: 06.06.2021)

⁷GitHub репозиторий Noodle.

URL: <https://github.com/shamansir/noodle> (дата обращения: 06.06.2021)

⁸Официальная страница языка программирования PureScript.

URL: <https://www.purescript.org> (дата обращения: 06.06.2021)

работка патчей), которая реализует пользовательские интерфейсы, построенные на узлах.

Noodle имеет достаточно большое количество недостатков, а также очень ограниченный набор возможностей, так как его разработка ещё не завершена. Тем не менее проект может иметь большое будущее.

1.1.4 GenMyModel

GenMyModel¹⁰ – это онлайн-инструмент для моделирования различных типов диаграм, блок-схем и прочего. Данный сервис был реализован с использованием фреймворка Web Modeling Framework¹¹, который входит в состав набора инструментов и фреймворков Eclipse Modeling Project¹².

Редактор GenMyModel является достаточно функциональным, но имеет некоторые весомые недостатки, например, невозможность создания связей между узлами сразу на сцене и отсутствие точек перегиба у связей. Редактор можно увидеть на рис. 3.

1.1.5 Visual Paradigm

Visual Paradigm¹⁴ – набор инструментов, с помощью которых можно осуществлять построение диаграм и схем, проектирование баз данных, моделирование процессов и так далее. Редактор данного сервиса обладает высокой функциональностью, а также имеет большое количество различных возможностей и функций для пользователя. Однако программа является платным, закрытым и по большей части десктопным продуктом. Внешний вид редактора можно увидеть на рис. 4.

⁹GitHub репозиторий RPD.

URL: <https://github.com/shamansir/rpd> (дата обращения: 06.06.2021)

¹⁰Официальная страница проекта GenMyModel.

URL: <https://www.genmymodel.com> (дата обращения: 06.06.2021)

¹¹Официальная страница фреймворка Web Modeling Framework.

URL: <https://projects.eclipse.org/proposals/web-modeling-framework> (дата обращения: 06.06.2021)

¹²Официальная страница проекта Eclipse Modeling Project.

URL: <https://www.eclipse.org/modeling> (дата обращения: 06.06.2021)

¹³ИТ блог Олега Солозобова. GenMyModel.

URL: <https://8d9.ru/program/genmymodel> (дата обращения: 06.06.2021)

¹⁴Официальная страница проекта Visual Paradigm.

URL: <https://www.visual-paradigm.com> (дата обращения: 06.06.2021)

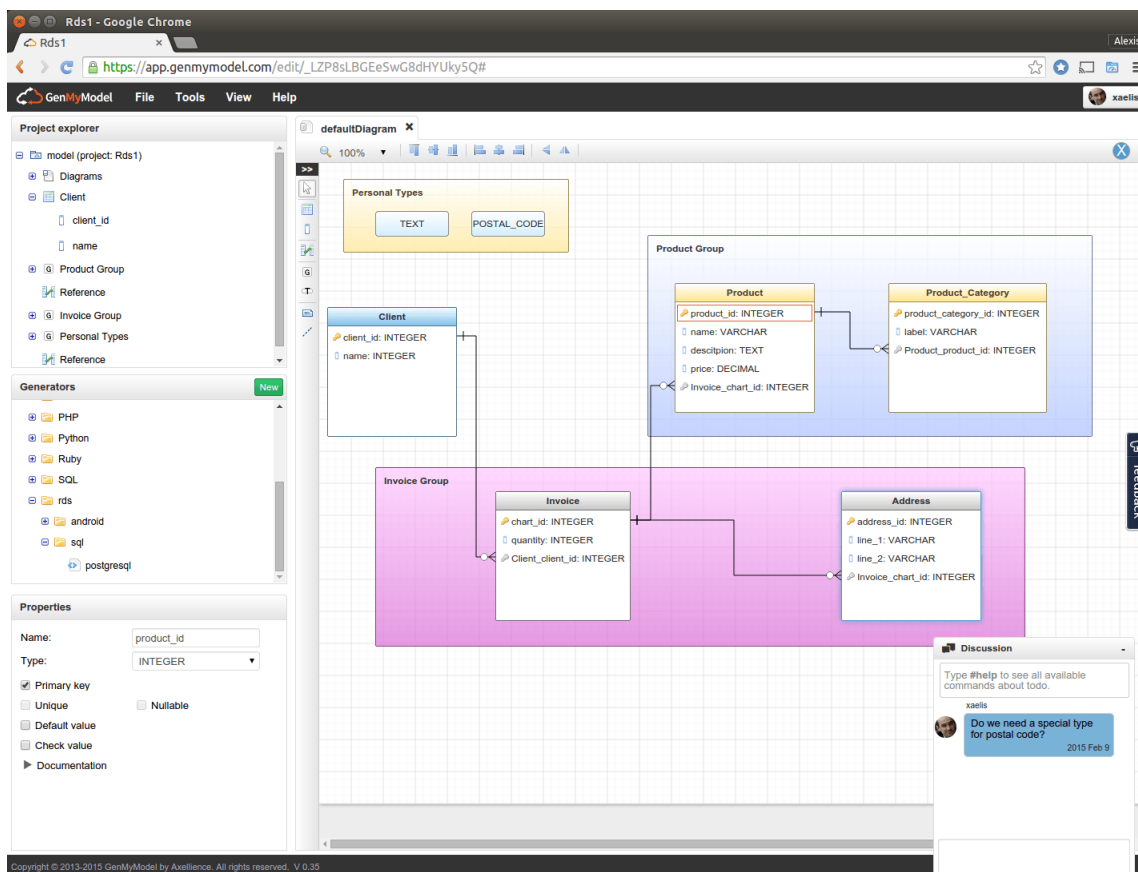


Рис. 3: Редактор GenMyModel (источник¹³)

1.1.6 YEd Live

YEd Live¹⁶ – это бесплатное веб-приложение, аналог десктопной программы yEd¹⁷, предназначенное для эффективного построения диаграмм и схем. Оба продукта были разработаны с использованием библиотеки yFiles¹⁸. Инструмент обладает всеми стандартными функциями и, по мнению автора отчёта, является достаточно функциональной средой для рисования. Редактор yEd Live представлен на рис. 5.

¹⁵Visual Paradigm. DBMS Tools.

URL: <https://dbmstools.com/tools/visual-paradigm> (дата обращения: 06.06.2021)

¹⁶Официальная страница проекта yEdLive.

URL: <https://www.yworks.com/yed-live> (дата обращения: 06.06.2021)

¹⁷Официальная страница проекта yEd.

URL: <https://www.yworks.com/products/yed> (дата обращения: 06.06.2021)

¹⁸Официальная страница библиотеки yFiles.

URL: <https://www.yworks.com/products/yfiles> (дата обращения: 06.06.2021)

¹⁹Graphity for Confluence. Yworks.

URL: <https://www.yworks.com/products/graphity-for-confluence> (дата обращения: 06.06.2021)

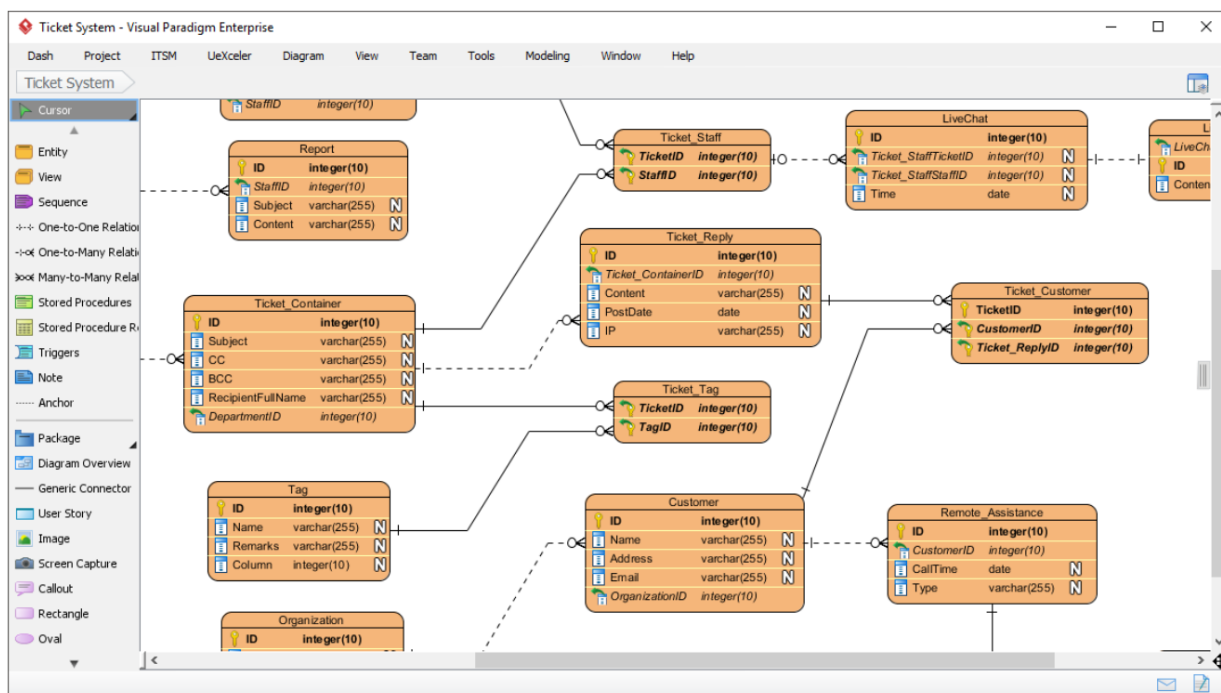


Рис. 4: Редактор Visual Paradigm (источник¹⁵)

1.2 Существующие библиотеки

В этом разделе приводится обзор некоторых существующих библиотек для реализации диаграмм и блок-схем. Для обозрения были взяты одни из самых популярных и используемых библиотек в данной области.

В ходе обзора были рассмотрены встроенные в библиотеки функции, настройка элементов под нужды системы, а также способ организации хранения существующих в системе моделей. Проанализировав все достоинства и недостатки обзореваемых библиотек, по итогу была выбрана наиболее подходящая для дальнейшей реализации библиотека.

1.2.1 STORM React Diagrams

STORM React Diagrams²⁰ – библиотека, написанная на языке TypeScript с использованием библиотеки React. Она обладает широким спектром возможностей для построения различных диаграм, а также является очень гибкой и хорошо настраиваемой. Единственный недостаток данной биб-

²⁰GitHub репозиторий библиотеки STORM React Diagrams.

URL: <https://github.com/projectstorm/react-diagrams> (дата обращения: 06.06.2021)

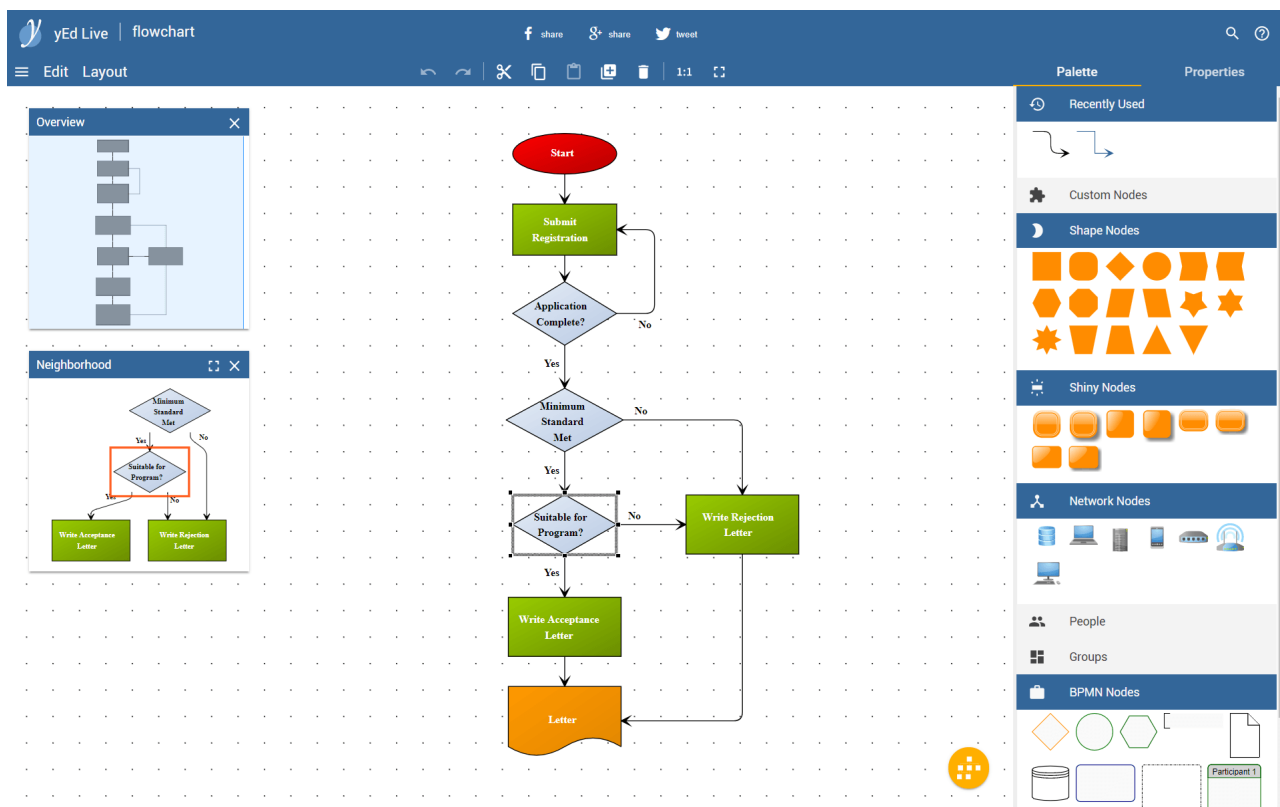


Рис. 5: Редактор yEd Live (источник¹⁹)

лиотеки – сложность с откреплением и удалением связей.

1.2.2 React Flow

React Flow²¹ – TypeScript и React библиотека, которая используется для построения диаграмм на узлах. Библиотека не только обладает современным пользовательским интерфейсом и имеет большой набор полезных компонентов, но и удобно настраивается под любые запросы разработчика, что делает её довольно подходящей для реализуемого веб-редактора.

1.2.3 React d3 graph

React d3 graph²² – JavaScript библиотека для интерактивных и настраиваемых графов, разработанная с помощью React на базе другой библиотеки

²¹Официальная страница библиотеки React Flow.

URL: <https://reactflow.dev> (дата обращения: 06.06.2021)

²²GitHub репозиторий библиотеки React d3 graph.

URL: <https://github.com/danielcaldas/react-d3-graph> (дата обращения: 06.06.2021)

D3²³. Несмотря на большое разнообразие встроенных свойств, в библиотеке отсутствует палитра элементов. Кроме того, React d3 graph имеет достаточно ограниченную функциональность, что является большим минусом.

1.2.4 JointJS

JointJS²⁴ – достаточно популярная бесплатная библиотека, созданная для визуализации и манипулирования различными диаграммами и графами и написанная на языке JavaScript. С помощью данной библиотеки можно без затруднений создавать модели самых разных видов, размеров и сложностей, однако в её реализации использовались технологии, которые начинают устаревать.

1.2.5 GoJS

GoJS²⁵ – хорошо интегрируемая с распространёнными сейчас фреймворками библиотека, которая предназначена для построения диаграмм и графов. Будучи разработанной на TypeScript, она может также использоваться в качестве JavaScript библиотеки. GoJS можно настроить под свои нужды, но нельзя свободно использовать в коммерческих целях.

²³Официальная страница библиотеки D3.

URL: <https://d3js.org> (дата обращения: 06.06.2021)

²⁴Официальная страница библиотеки JointJS.

URL: <https://www.jointjs.com/opensource> (дата обращения: 06.06.2021)

²⁵Официальная страница библиотеки GoJS.

URL: <https://gojs.net> (дата обращения: 06.06.2021)

2. Описание решения

На основании обзора существующих редакторов и библиотек были сформулированы следующие требования к реализуемому веб-редактору.

- Добавление узлов на сцену из палитры перетаскиванием.
- Перемещение узлов по сцене.
- Добавление связей (отведение от портов на узле, обязательное приведение к другому узлу).
- Удаление узлов и связей (за удалением узла следует удаление всех его связей).
- Поддержка связей узла в самого себя.
- Открепление и прикрепление связей к другому узлу.
- Поддержка операций отмены и возврата (undo & redo).
- Сохранение созданной модели и открытие существующей из файла.
- Возможность изменения размера узлов (ограничение по размеру) и поддержка изменения размера через изменение свойств узлов.
- Возможность изменения свойств узлов и связей непосредственно на сцене.
- Добавление связей из палитры перетаскиванием.
- Создание точек перегиба у связей.
- Поддержка операций копирования, вставки, вырезания, дублирования элементов.
- Возможность масштабирования.
- Отображение сетки и возможность расположения и изменения размера элементов по сетке.

- Изменение направления связи.
- Возможность изменения типа связи в свойствах.
- Существование ограничений по доступности добавления связей.
- Возможность выделения нескольких элементов и совершения действий над ними.
- «Подсветка» узла при установке связи.
- Поддержка узлов с ограниченным количеством портов.

Для разработки веб-редактора была выбрана широко распространённая библиотека React в сочетании с языком программирования TypeScript, который, в отличие от популярного JavaScript, строго типизирован и объектно-ориентирован. Библиотека React была выбрана исходя из того факта, что на данный момент она является самой популярной на IT-рынке, поэтому большинство других библиотек адаптированы для совместного с ней использования.

Для реализации всех предъявленных требований была выбрана библиотека React Flow, которая не только обладает множеством различных встроенных функций, но и написана на TypeScript с использованием React, что обеспечивает прекрасную совместимость.

Стоит отдельно отметить, что разработка редактора выполнялась командой, в которой было двое второкурсников. Им были поставлены следующие задачи.

- Реализации окна палитры и редактора свойств.
- Развёртывание демо-версии веб-редактора.
- Поддержка отображения картинок в качестве узлов.
- Поддержка связей в себя (ещё в разработке).

В ходе их работы автором отчёта оказывалась необходимая консультация, помощь, а также проводился код-ревью пулл-реквестов.

Редактор состоит из нескольких частей: палитры, сцены и панели работы со свойствами. Каждая из этих частей представляет собой функциональный компонент с пропсами.

Функциональный React-компонент – это, в нашем случае, TypeScript-функция, возвращающая часть кода, которая представляет собой фрагмент веб-страницы. Компоненты написаны с помощью JSX-синтаксиса, который позволяет сочетать HTML-разметку и TypeScript-логику. Пропсами называются данные, представленные в виде одного объекта, которые передаются и используются как родительским компонентом, так и компонентами-потомками. Дерево React-компонентов веб-редактора можно увидеть на рис. 6.

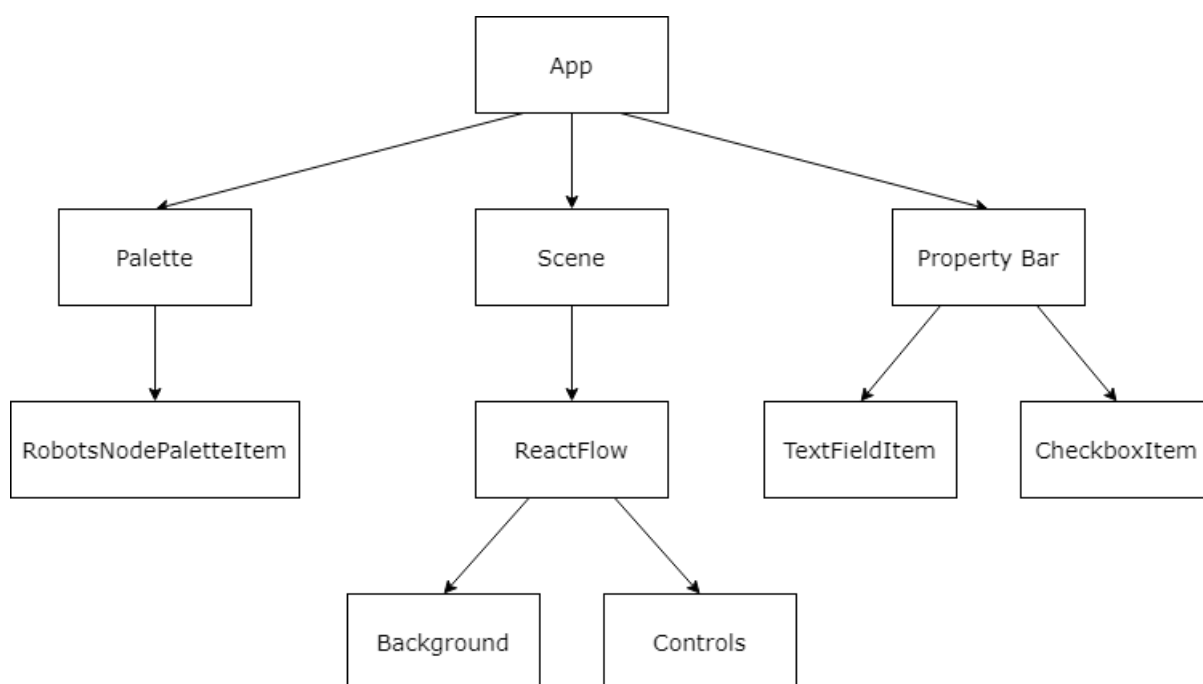


Рис. 6: Дерево React-компонентов веб-редактора

Компоненты взаимодействуют с репозиторием бэкенда через серверную часть фронтенда, спроектированную по паттерну Gateway (рис. шлюз) и реализованную с помощью фреймворка Express, обладающего простотой, гибкостью и масштабируемостью.

Паттерн Gateway применяется, чтобы обеспечить более удобный интерфейс для доступа к API. Он предполагает наличие специального объекта, называемого шлюзом, который инкапсулирует доступ к внешней системе или источнику данных. Остальные объекты обращаются к API через этот шлюз,

который транслирует эти простые вызовы в полноценные запросы к API. Графическая интерпретация работы данного паттерна представлена на рис. 7.

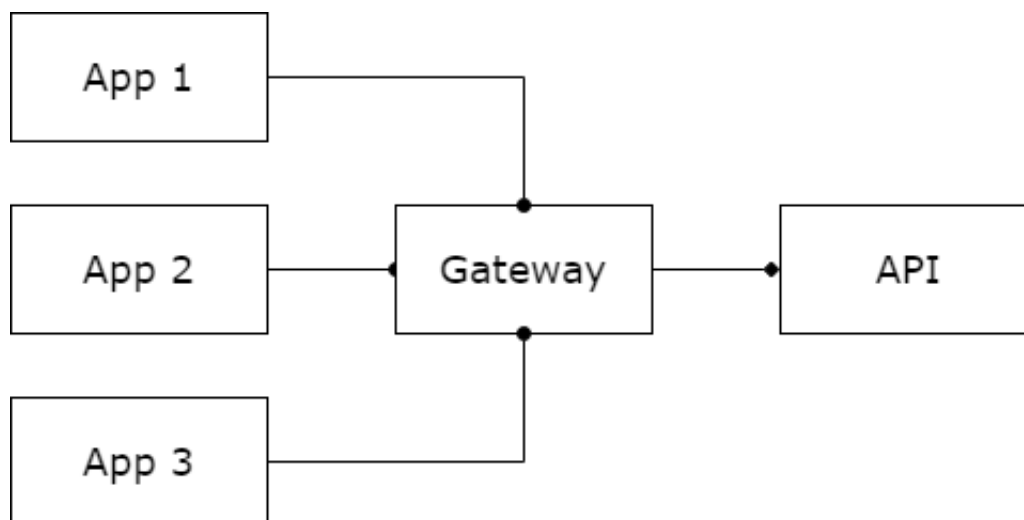


Рис. 7: Паттерн Gateway

Все запросы на фронтенде выполняются с помощью библиотеки `axios`²⁶, которая обладает широкой функциональностью и удобством по сравнению с существующими аналогами.

При начальном запуске веб-редактора компоненты палитры и сцены принимают в качестве пропсов имена метамодели и модели соответственно, после чего отправляют запрос на получение элементов через шлюз и отрисовывают их, используя метод массива `map`. Компонент панели работы со свойствами делает запрос на получение свойств и их значений при выборе какого-либо элемента на сцене пользователем и также отрисовывает их с помощью метода массива `map`. Помимо этого, панель работы со свойствами и сцена делают запросы на изменение значений свойств элементов, например, имени или координат, сразу же после их изменения.

В данный момент в палитре имеются элементы языка моделирования для роботов, который представлен в TRIK Studio. Этот язык включает в себя узел с таймером, финальный узел, связи между узлами и несколько узлов для управления моторами: `MotorsForward`, `MotorsBackward` и `MotorsStop`.

²⁶GitHub репозиторий библиотеки `axios`.

URL: <https://github.com/axios/axios> (дата обращения: 06.06.2021)

Итого на данный момент веб-редактор обладает следующей функциональностью.

- Взаимодействие с репозиторием бэкенда.
- Работа со свойствами элементов, а именно загрузка свойств элементов из репозитория и изменение их значений.
- Автоматическое сохранение изменений в модели, например, добавление новых узлов, связей, перемещение элементов и изменение значений их свойств.
- Работа с языком моделирования для роботов, представленным в TRIK Studio.

Для развёртывания веб-редактора была выбрана платформа Heroku²⁷, позволяющая бесплатно запустить приложение, и платформа Docker²⁸ для автоматизации и управления. Бэкенд и фронтенд веб-редактора разворачивались в двух разных приложениях. Для этого в первом случае использовался уже имеющийся docker-образ, а во втором — написанный специально для разворачивания файл Dockerfile, автоматизирующий сборку docker-образа, и файл heroku.yml, определяющий сборку.

Нынешний внешний вид веб-редактора можно увидеть на рис. 8. В левой части экрана располагается панель работы со свойствами, справа представлена палитра элементов, между ними — сцена.

²⁷Официальная страница платформы Heroku.

URL: <https://www.heroku.com> (дата обращения: 06.06.2021)

²⁸Официальная страница платформы Docker.

URL: <https://www.docker.com> (дата обращения: 06.06.2021)

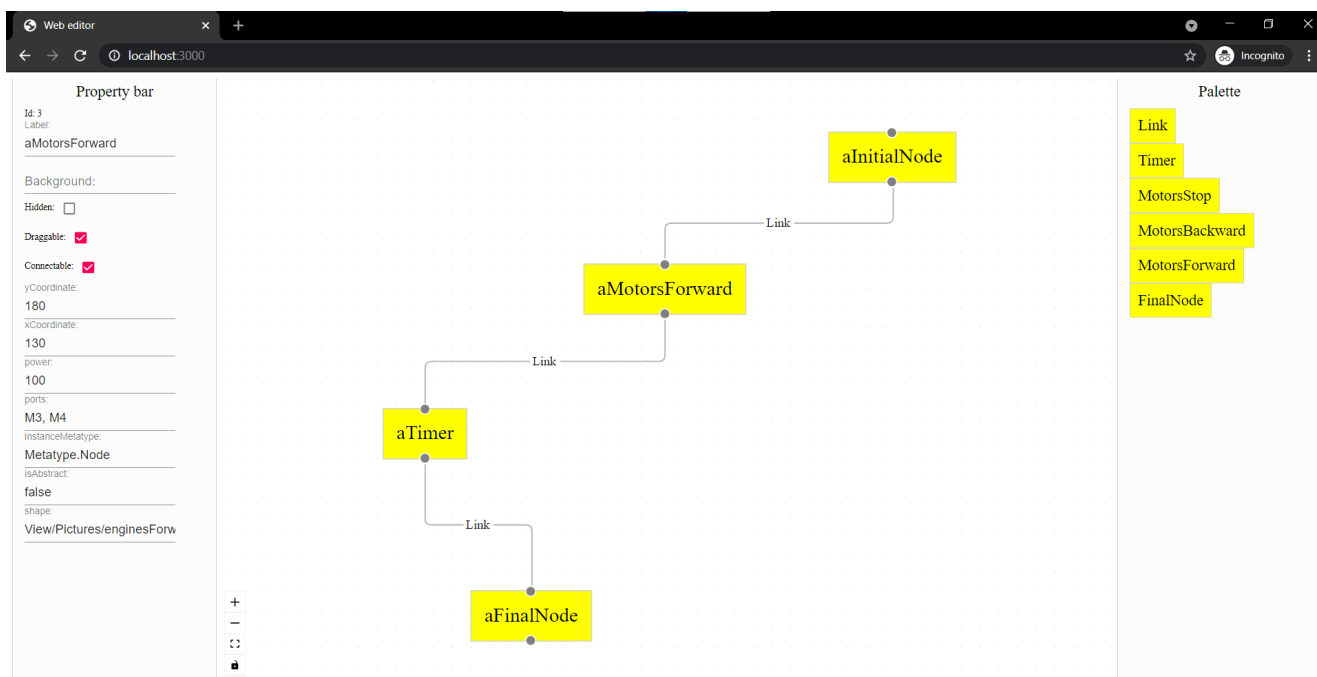


Рис. 8: Внешний вид веб-редактора

3. Апробация

Апробация проводилась с использованием метода Single Ease Question (SEQ) и шкалы System Usability Scale (SUS). В ней участвовало 8 человек.

Участникам опроса предлагалось выполнить несколько простых действий по взаимодействию с интерфейсом веб-редактора, такими как «Добавить новый узел из палитры», «Создать связь между двумя узлами», «Изменить значение свойства» и так далее. Перед и после каждого из заданий нужно было оценить его сложность по шкале от 1 (проще простого) до 5 (невозможно) по методу SEQ. После выполнения всех заданий участникам предлагалось ответить на вопросы по шкале SUS.

Результаты опроса по методу SEQ можно увидеть на рис. 9. По шкале SUS было получено среднее количество баллов 72.8, что соответствует оценке "В-" («Хорошо»).

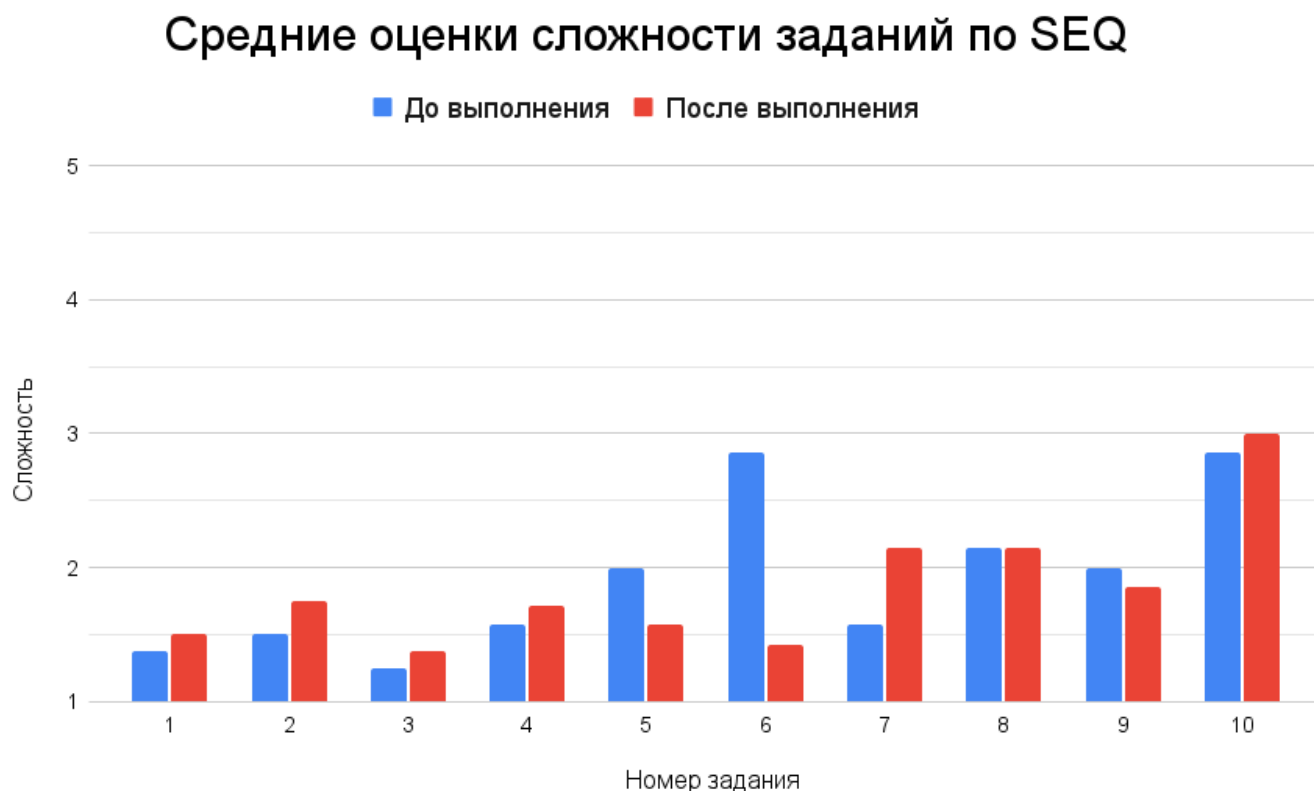


Рис. 9: Результаты опроса по методу SEQ

Кроме того, благодаря апробации были выявлены некоторые недостатки системы.

- Использование клавиши "Delete" для удаления элементов со сцены не совсем очевидно. Решением данной проблемы является добавление дополнительной кнопки для удаления элемента в панели для работы со свойствами, а также добавление всплывающего меню с действиями над элементом по нажатию правой кнопки мыши.
- Долгая загрузка элементов в палитре и на сцене, а также свойств.
- Изменение значений некоторых свойств, например, имени элемента, удобнее делать прямо на сцене, а не только в панели для работы со свойствами.

Заключение

В рамках данной учебной практики были достигнуты следующие результаты.

- Выполнен обзор существующих редакторов.
- Сформированы требования к веб-редактору.
- Выполнен обзор существующих библиотек и произведён выбор библиотеки для реализации.
- Реализована рабочая версия веб-редактора со следующими функциями.
 - Взаимодействие с репозиторием бэкенда.
 - Поддержка работы со свойствами.
 - Автоматическое сохранение изменений.
 - Поддержка языка моделирования для роботов.
- Выполнен код-ревью пулл-реквестов второкурсников.
- Развёрнут веб-редактор.
- Проведена апробация.

Текущие результаты представлены в репозитории на GitHub²⁹, а также на платформе Heroku³⁰, где развёрнут веб-редактор.

²⁹GitHub репозиторий проекта.

URL: <https://github.com/REAL-NET/web-editor-frontend> (дата обращения: 06.06.2021)

³⁰Развёрнутый веб-редактор REAL.NET на платформе Heroku.

URL: <https://web-editor.herokuapp.com> (дата обращения: 06.06.2021)

Список литературы

- [1] Литвинов Ю.В., Кириленко Я.А. TRIK Studio: среда обучения программированию с применением роботов. 2015.
URL: https://robofinist.ru/files/47994/filename/Thesis_2015.pdf (дата обращения: 06.06.2021).
- [2] Erwin B., Cyr M., Rogers. C. Lego engineer and Robolab: Teaching engineering with Labview from kindergarten to graduate school. 2000.
URL: https://www.ni.com/pdf/academic/us/journals/ijee_05.pdf (дата обращения: 06.06.2021).
- [3] Литвинов Ю.В., Кузьмина Е.В, Небогатиков И.Ю, Алымова Д.А. Среда предметно-ориентированного программирования REAL.NET. 2017.
URL: <https://oops.math.spbu.ru/SE/seminar/materials/realnet.pdf> (дата обращения: 06.06.2021).
- [4] Терехов А.Н., Брыксин Т.А., Литвинов Ю.В., Смирнов К.К., Никандров Г.А., Иванов В.Ю. Таkun Е.И. Архитектура среды визуального моделирования QReal. 2009.
URL: <https://github.com/qreal/qreal/wiki/QReal-architecture-2009.pdf> (дата обращения: 06.06.2021).
- [5] Терехов А.Н., Брыксин Т.А., Литвинов Ю.В. QReal: платформа визуального предметно-ориентированного моделирования. 2013.
URL: http://novtex.ru/prin/full/06_2013.pdf (дата обращения: 06.06.2021).