

Панель администрирования платформы многоагентного моделирования с возможностью построения графических отчетов

А.А. Булова, С.С. Буров, Д.С. Парыгин, А. Г. Финогеев, Т.В. Смирнова

Аннотация— В данной статье рассмотрены проектирование и реализация web клиента, выполняющего роль системы администрирования для платформы многоагентного моделирования перемещений и взаимодействий акторов в рамках участка карты города. Вся логика моделирования в данной платформе реализуется непосредственно в модулях, тогда как программная платформа осуществляет лишь её вызов для конкретных, подключенных модулей. Разрабатываемая система администрирования отображает текущее местоположение и топологию акторов, имеет функционал управления состоянием моделирования и списком участвующих в моделировании модулей, а также возможность отображения и изменения свойств модели и каждого актора в отдельности. Состояние модели (свойства и местоположение акторов) обновляется в реальном времени, с использованием системы gRPC, без необходимости перезагрузки веб страницы. Пользовательский интерфейс системы включает в себя карту и элементы управления. Программная платформа реализована на фреймворке ASP.NET Core версии 5.0. Для реализации веб клиента был выбран фреймворк Angular версии 11 с использованием компонентов пользовательского интерфейса Ant Design. Кроме того, для системы администрирования разработаны методы построения графических отчетов на основе полученных данных о моделировании.

Ключевые слова— Моделирование, симуляция, город, С#, модульность, клиент-сервер, система администрирования.

I. ВВЕДЕНИЕ

Существует множество платформ, позволяющих проводить моделирование с графическим

пользовательским интерфейсом. Рассмотрим три платформы в таб. 1: Ant Road Planner [1], NetLogo [2] и AnyLogic [3].

Графический пользовательский интерфейс в таких платформах зачастую имеет ряд ограничений и работает неотрывно от самой платформы, становясь из-за этого ее непосредственной частью и накладывая ряд ограничений на эксплуатацию самой платформы.

Были рассмотрены некоторые исследования работы с распределенными системами моделирования [4, 5], а также способы корректировки поведения и свойств акторов в существующих исследованиях [6, 7, 8].

Таб. 1. Обзор платформ моделирования с панелью администрирования

НАЗВАНИЕ ПЛАТФОРМЫ	РЕДАКТОР БАЗОВОЙ КАРТЫ	ДИНАМИЧЕСКОЕ ИЗМЕНЕНИЕ СВОЙСТВ МОДЕЛИРОВАНИЯ	МОДУЛЬНОСТЬ	ОТОБРАЖЕНИЕ РЕЗУЛЬТАТОВ МОДЕЛИРОВАНИЯ	ДИНАМИЧЕСКОЕ ИЗМЕНЕНИЕ ДИАПАЗОНА МОДЕЛИРОВАНИЯ
Ant Road Planner	+	-	-	+	-
NetLogo	+	-	-	+	+
AnyLogic	+	-	-	+	-

Целевым компонентом данной работы является приложение, представляющее собой клиент-серверную платформу для моделирования перемещений и взаимодействий акторов в рамках участка карты города [9, 10]. Построение модели ведется на основе многоагентного подхода [11, 12]. Программа осуществляет расширение функционала моделирования за счет подключаемых к ней модулей (инициализирует их) и отображает функционал на графической карте города. Разработка различных модулей ведется по сей день, и система постоянно расширяется.

Модули содержат в себе всю логику и правила моделирования. Изначально модули не поставляются в составе данного программного комплекса, каждый из них является отдельной обособленной библиотекой классов. Однако модули принимают непосредственное участие в работе программного комплекса в случае подключения одного или нескольких из них. К примеру, модуль управления данными об объектах на онлайн-

Статья получена 27 октября 2021.

Булова Елена Алексеевна, Волгоградский государственный технический университет, магистрант, ORCID 0000-0002-6733-8386 (attapi343@gmail.com)

Буров Сергей Сергеевич, Волгоградский государственный технический университет, магистрант, ORCID 0000-0001-7328-0953 (sergey.burovic@gmail.com)

Парыгин Данила Сергеевич, кандидат технических наук, доцент, Волгоградский государственный технический университет, ORCID 0000-0001-8834-5748 (dparugin@gmail.com)

Финогеев Алексей Германович, доктор технических наук, профессор, Пензенский государственный университет, ORCID 0000-0002-4777-3364 (alexeyfinoogeev@gmail.com)

Смирнова Татьяна Владимировна, доцент, Волгоградский государственный социально-педагогический университет, ORCID 0000-0003-2861-3119 (smirnova.tatiana.2013@yandex.ru)

карте города [13] производит парсинг данных из файла с форматом OSM XML и преобразует их к необходимой структуре для хранения в списке объектов основной программы и дальнейшего использования их другими модулями.

Система администрирования требуется для исполнения различных сценариев моделирования в реальном времени с расширяемым свойством списком свойств самой модели и акторов и подключаемыми компонентами для получения ответа на множественный вопрос "что, если".

В связи с этим, целью работы является разработка системы распределенного администрирования поведения акторов и свойств модели города в реальном времени. При этом в качестве ключевого компонента концепции создания такой системы было решено сфокусироваться на предоставлении возможности пользователю влиять на ход моделирования путем изменения свойств моделей и акторов в реальном времени.

II. ПОСТАНОВКА ЗАДАЧИ

Существующее приложение моделирует перемещения акторов на основе модулей, однако имеет ряд недостатков и ограничений, часть которых должна быть устранена в созданной панели администрирования.

Объектом настоящего исследования является процесс администрирования моделирования и получения результатов.

Предмет исследования – методы распределенного администрирования поведения акторов и свойств модели города в реальном времени.

Диаграмма бизнес-процессов ("BPMN AS IS") в случае с использованием приложения отображена на рис. 1.

Одной из проблем текущих систем администрирования моделирования является отсутствие подходов к обработке и представлению состояния акторов в реальном времени. Системы отображают результат моделирования без возможности приостановить моделирование или перезапустить его во время текущего моделирования уже с другими модулями и/или свойствами.

В некоторых системах моделирования, таких как Apt Road Planner после начальной настройки свойств моделирования и запуска моделирования нет возможности динамически изменить какие-либо свойства модели в целом и акторов в частности в реальном времени. Только после отработки моделирования и получения результатов можно заново запустить моделирование на измененных свойствах.

Проблемой также можно назвать отсутствие динамического изменения диапазона обрабатываемых акторов на основе текущего пользовательского представления. Часто в начале моделирования выбирается конкретный фиксированный участок города, который уже в ходе моделирования невозможно масштабировать без остановки и перезапуска моделирования целиком.

В связи с этим была поставлена задача разработать систему распределенного администрирования поведения акторов и свойств модели города в реальном времени с предоставлением возможности пользователю менять ход моделирования путем изменения свойств моделей и акторов. Система должна динамически менять диапазон обрабатываемых акторов на основе текущего пользовательского представления.

Для реализации веб клиента был выбран фреймворк Angular [14] версии 11 с использованием компонентов пользовательского интерфейса Ant Design [15].

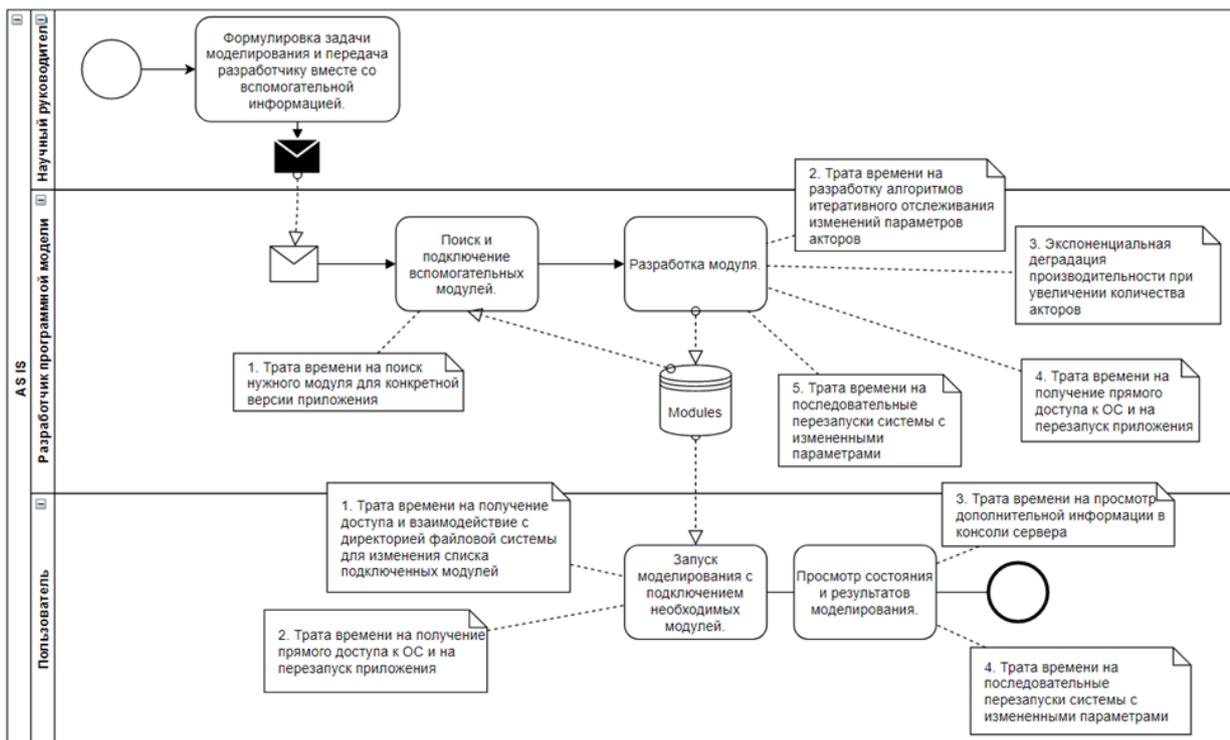


Рис. 1. BPMN AS IS

отображается на карте таким образом, что каждый объект типа `MapFeature` занимает отдельный слой со своим стилем, обозначенным в соответствующем `open_layers_style`. Стиль извлекается из строковой переменной с помощью функции `eval`. Итоговый вариант карты вместе с панелью администрирования отображен на рис. 6.

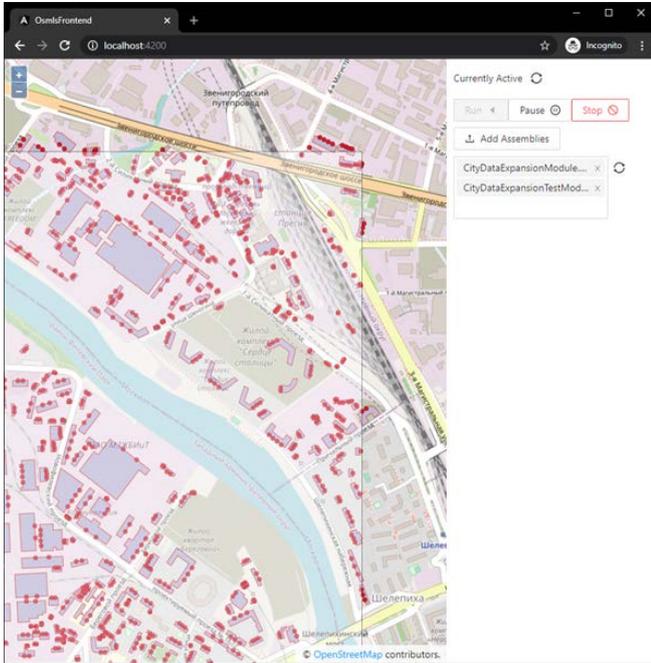


Рис. 6. Карта с панелью администрирования

V. ТЕСТИРОВАНИЕ ОБНОВЛЕННОЙ СИСТЕМЫ АДМИНИСТРИРОВАНИЯ

A. Общая информация о тестировании

Для тестирования результатов используется платформа моделирования с подключенным тестовым модулем.

Тестовым модулем на карту добавляются несколько акторов:

- Точка (`PointActor`).
- Ломаная линия (`LineStringActor`).
- Полигон (`PolygonActor`).
- Самолет 1 с начальной скоростью 1000 компании “Boeing” (`Airplane`, класс актора, созданный в модуле).
- Самолет 2 с начальной скоростью 0 компании “Airbus” (`Airplane`, класс актора, созданный в модуле).

Класс `Airplane` наследуется от класса `PointActor` и имеет несколько дополнительных обозреваемых свойств.

B. Отображение метаданных

Отображение метаданных происходит независимо от статуса процесса моделирования, они всегда отображаются в компоненте свойств карты для всех акторов, имеющих хотя бы одно обозреваемое свойство. На рис. 7 представлено отображение метаданных свойств акторов для акторов типа `Airplane`. Так как процесс моделирования остановлен и акторов в данный момент нет, таблица имеет только колонки, обозначающие обозреваемые свойства и не имеет строк.

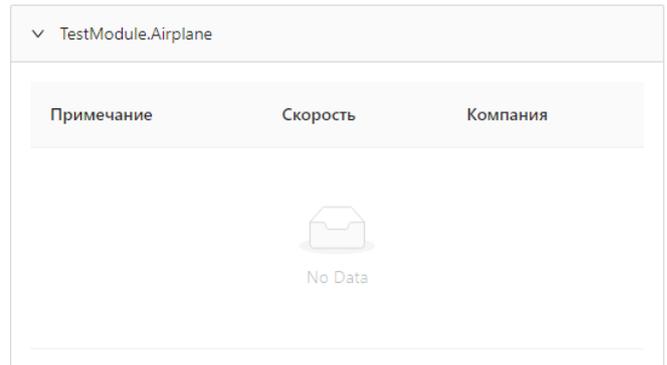


Рис. 7. Отображение метаданных свойств акторов типа `Airplane`

Метаданные, отвечающие за отображение стилей, можно увидеть при отображении акторов в компоненте обозревателя карты (рис. 8), отображение происходит только при запущенном или приостановленном процессе моделирования, так как только в этот момент на карте могут быть акторы.

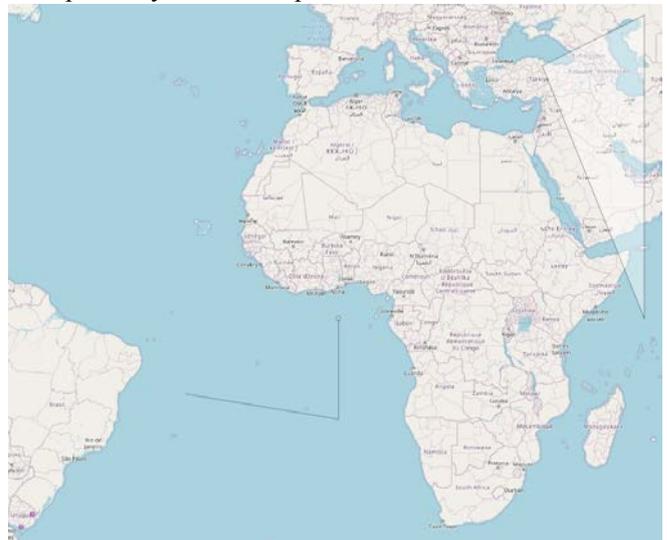


Рис. 8. Отображение акторов в компоненте обозревателя карты

На карте присутствуют два актора типа `Airplane` (левый нижний угол), отображающихся как фиолетовые небольшие точки, актор ломаной линии (в центре карты), актор точки (на одном из концов актора ломаной линии), актор полигона (левый верхний угол). Из всех перечисленных объектов измененный стиль имеют только объекты типа `Airplane`.

C. Отображение акторов

Отображение акторов происходит как при подключении системы администрирования к платформе моделирования с уже запущенным моделированием, так и при запуске моделирования со стороны системы администрирования с одинаковым результатом.

При этом обновления акторов корректно отображаются системой администрирования с течением времени (рис. 9).



Рис. 9. Отображение движущегося актора в момент времени 1 и 2

При остановке процесса моделирования компонент обозревателя карты очищается от акторов.

D. Отображение обозреваемых свойств

При запуске моделирования в компоненте обозревателя карты отображаются обозреваемые свойства, в соответствии с разметкой таблицы и метаданными (рис. 10).

TestModule.Airplane		
Примечание	Скорость	Компания
	1000	Boeing
	0	Airbus

Рис. 10. Отображение обозреваемых свойств акторов типа Airplane (с одним изменившимся свойством)

E. Редактирование обозреваемых свойств

При наведении на строку, в которой присутствуют редактируемые обозреваемые свойства, ячейки редактируемых свойств будут выделены (рис. 11).

TestModule.Airplane		
Примечание	Скорость	Компания
	1000.1	Boeing
	<input type="text" value="0"/>	Airbus

Рис. 11. Выделение ячеек редактируемых свойств строки

При нажатии на одну из таких ячеек будет открыт элемент редактирования (рис. 12).

TestModule.Airplane		
Примечание	Скорость	Компания
	1000.1	Boeing
	<input type="text" value="0"/>	Airbus

Рис. 12. Открытие элемента редактирования ячейки

При изменении значения свойства и закрытии диалога редактирования, данное свойство будет изменено на платформе моделирования, а также в текущей и во всех остальных подключенных панелях администрирования (рис. 13).

TestModule.Airplane		
Примечание	Скорость	Компания
	1000.1	Boeing
Этот самолет не движется!	0	Airbus

Рис. 13. Отображение свойства с измененным значением

VI. РЕАЛИЗАЦИЯ ОТЧЕТА О ХОДЕ МОДЕЛИРОВАНИЯ

A. Постановка задачи

На данном этапе разработки пользователю, использующему разрабатываемую систему администрирования, предоставляется лишь информация о процессе моделирования. Отсутствие какой-либо информации о результатах (или промежуточных результатах) моделирования сильно ограничивает возможность использования разрабатываемой системы в рамках процесса принятия решений, т.к. лицо принимающее решение не получает никаких обработанных результирующих данных.

Для решения поставленной проблемы требуется разработать функционал построения и отображения графических отчетов на основе данных платформы моделирования, который должен отображать результирующие данные о типах объектов, свойствах объектов и об объектах, участвующих в процессе моделирования.

B. Отображение отчета в пользовательском интерфейсе

Для отображения отчета в пользовательском интерфейсе было выбрано то же расположение, что и для обозреваемых свойств, для этого были созданы две вкладки: “Properties” (обозреваемые свойства, рис. 14) и

“Report” (отчет, рис. 15). Переключение между вкладками позволяет пользователю выбирать, какая именно дополнительная информация об объектах карты должна быть представлена в данный момент.



Рис. 14. Обновленное расположение раздела “Properties” в пользовательском интерфейсе

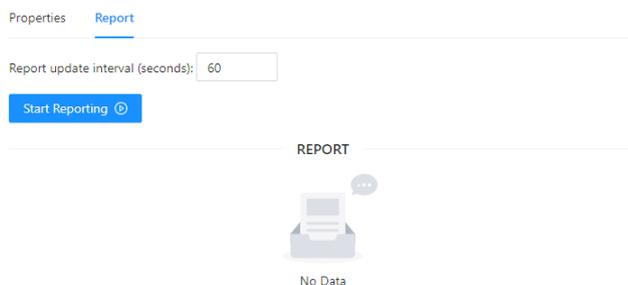


Рис. 15. Расположение раздела “Report” в пользовательском интерфейсе

C. Состояния отчета

Функция построения отчета по-умолчанию отключена, во многом в связи с тем, что при большом количестве объектов карты данный функционал может задействовать большие вычислительные ресурсы, чем может предоставить вычислительная машина пользователя.

Для того, чтобы пользователь имел возможность запускать построение отчета, была создана кнопка “Start Reporting”, кроме того, пользователь имеет возможность задать интервал, на основе которого отчет будет перестраиваться (рис. 16). Если построение отчета по интервалу в данный момент запущено, интервал не может быть изменен.

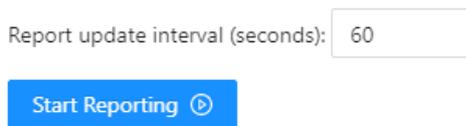


Рис. 16. Запуск построения отчета в пользовательском интерфейсе

Для приостановки периодического построения отчета пользователь может нажать кнопку “Stop Reporting” (рис. 17).



Рис. 17. Остановка построения отчета в пользовательском интерфейсе

D. Структура отчета

В верхней части отчета представлена общая информация об отчете: начальная и конечная дата и время отчетов.

Начальные дата и время отчета отражают момент начала сбора данных об объектах карты, в текущей реализации это происходит сразу же при запуске панели администрирования (таким образом, чтобы сбросить все заранее собранные данные, достаточно перезагрузить страницу).

Конечные дата и время отчета отражают момент окончания сбора данных об объектах карты, это происходит в момент построения данного экземпляра отчета.

Отображение общей информации об отчете представлено на рис. 18.

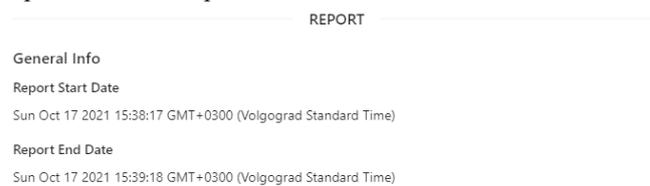


Рис. 18. Общая информация об отчете

Под общей информацией об отчете представлены отчеты для каждого из типов объектов, включающие в себя:

- название типа;
- график количества уникальных объектов с течением времени;
- список отчетов свойств для данного типа.

Структура интерфейса отчета для типа объектов на языке typescript:

```
export interface ITypeReport {
  name: string;
  featuresCountChartData:
  IFeaturesCountChartData;
  propertiesReports: IPropertyReport[];
}
```

Как было ранее отмечено, отчеты типов объектов включают в себя отчеты о свойствах объектов. Отчеты о свойствах объектов включают в себя:

- название свойства;
- числовые данные о свойстве;
- текстовые данные о свойстве.

Числовые данные о свойствах доступны только для свойств, имеющих числовой тип: DOUBLE, FLOAT, INT_32, INT_64, UINT_32, UINT_64. Это позволяет построить для таких свойств линейный график зависимости значения свойства от времени. Кроме того, это позволяет извлекать различные математические следствия из собранных данных, к примеру, позволяет строить график средней величины для каждого из объектов карты по выбранному свойству.

Текстовые данные о свойствах доступны для всех свойств, т.к. свойство с любым типом значения в конечном счете может быть интерпретировано как текстовое. На основе данных такого типа может быть построена, например, гистограмма, отображающая время, на протяжении которого объект имел конкретное значение данного свойства.

Структура интерфейса отчета для свойства на языке typescript:

```
export interface IPropertyReport {
  title: string;
  numericData?: {
```

```

    featuresChartData?:
INumericFeatureChartData[];
    featuresMeanChartData?:
INumericFeatureMeanChartData[];
};
    literalData?: {
        featuresChartData?:
ILiteralFeatureChartData[];
    };
}

```

Е. Инструменты построения отчета

Для преобразования данных из данных, приходящих от панели администрирования, в данные, подходящие для построения графиков, используется подход на основе функционального программирования с помощью lodash [16].

Преобразованные данные отображаются в виде графиков в рамках отчета с помощью фреймворка ngx-charts [17]. Для построения графиков был выбран данный инструмент в связи с его высокой производительностью (т.к. заявлено, что он напрямую использует возможности Angular, не выступая “оберткой” над каким-либо сторонним инструментом), настраиваемыми стилями диаграмм, возможностью реализации пользовательских диаграмм на основе модулей. Ожидается, что особенности выбранного инструмента, в сравнении с аналогами, должны позволить строить отчеты для сравнительно большого объема данных, а также, должны открыть возможность для увеличения количества графически представляемой информации, на основе диаграмм, которые могут быть созданы специально для поставленных целей.

Ф. Тестовые данные

Для тестирования построения отчета используется тестовый модуль, имеющий свойство “Количество самолетов”. При изменении данного свойства в большую сторону на карту добавляется один объект типа “Самолет” со случайными координатами и случайным значением скорости. При изменении данного свойства в меньшую сторону с карты удаляется ранее добавленный самолет.

Данное свойство напрямую взаимодействует с коллекцией объектов MapObjects и описывается на языке программирования C# следующим образом:

```

/// <summary>
/// Количество самолетов.
/// </summary>
[ObservableProperty("Количество самолетов",
true)]
public int AirplanesCount =>
{
    get
    MapObjects.Get<Airplane>().Count;
    set
    {
        var count = AirplanesCount;
        while (count < value)
        {
            MapObjects.Add(new Airplane(new
Coordinate(
    _Random.Next(0, 100000),
    _Random.Next(0, 100000)),

```

```

    _Random.Next(0, 1000),
    "Unknown" )
    );
        count++;
    }
    while (count > value)
    {
        MapObjects.Remove(MapObjects.Get<Airplane>()[
0]);
        count--;
    }
}

```

Отображение свойства “Количество самолетов” в пользовательском интерфейсе представлено на рис. 19.



Рис. 19. Отображение свойства “Количество самолетов” в пользовательском интерфейсе

Помимо двух стандартных объектов типа “Самолет”, на карту были добавлены ещё 3 объекта того же типа, с помощью изменения свойства модуля. У двух созданных объектов было изменено свойство “Примечание” через пользовательский интерфейс. Отображение свойств объектов типа “Самолет” представлено на рис. 20.

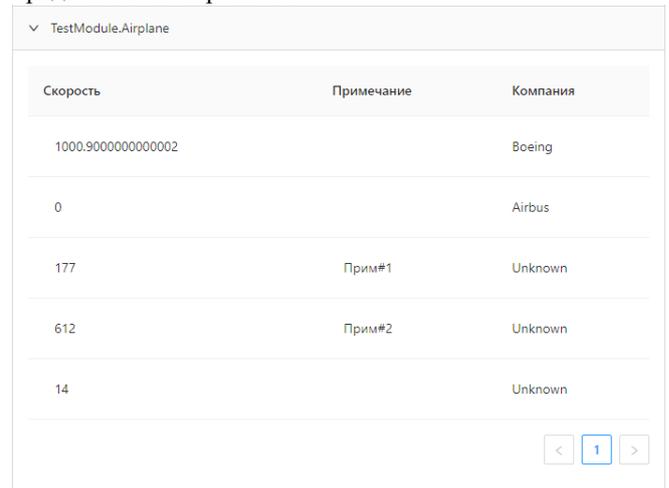


Рис. 20. Отображение свойств объектов типа “Самолет” в пользовательском интерфейсе

Г. Построение отчета на основе тестовых данных

Т.к. тип “TestModule” является типом модуля, график количества уникальных объектов для данного типа (рис. 21) всегда будет одинаковым: 1 элемент в начальный период времени. Это связано с текущими ограничениями платформы моделирования: всегда может быть один и только один модуль конкретного типа. Т.к. в будущем данное ограничение в теории

может быть снято, данный график отображается для всех объектов карты: как для акторов, так и для модулей.

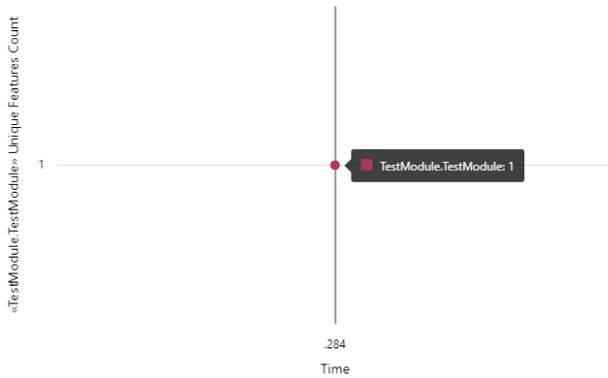


Рис. 21. Диаграмма зависимости количества уникальных объектов типа “TestModule” от времени

В рамках подготовки тестовых данных свойство “Количество самолетов” изначально имело значение 2, после чего было трижды итеративно увеличено на 1, в результате чего на момент построения отчета имеет значение 5. Процесс изменения значения данного свойства с течением времени представлен на графике на рис. 22.

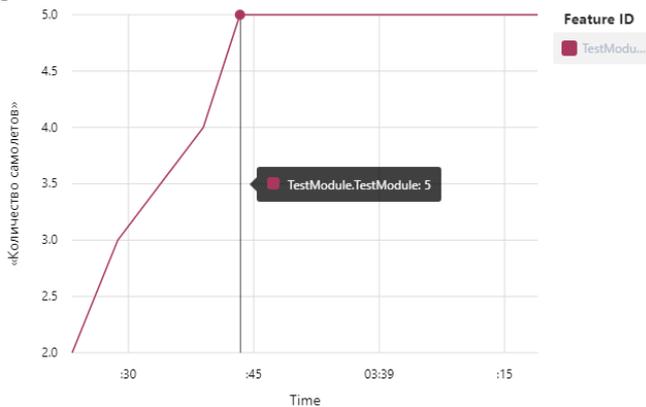


Рис. 22. Диаграмма зависимости значения свойства “Количество самолетов” от времени

На основе данных об изменении свойства, системой администрирования было рассчитано его среднее значение: $(2 + 3 + 4 + 5) / 4 = 3.5$. В связи с тем, что данному типу принадлежит лишь один объект, гистограмма имеет один столбец, отображающий среднее значение данного свойства (рис. 23).

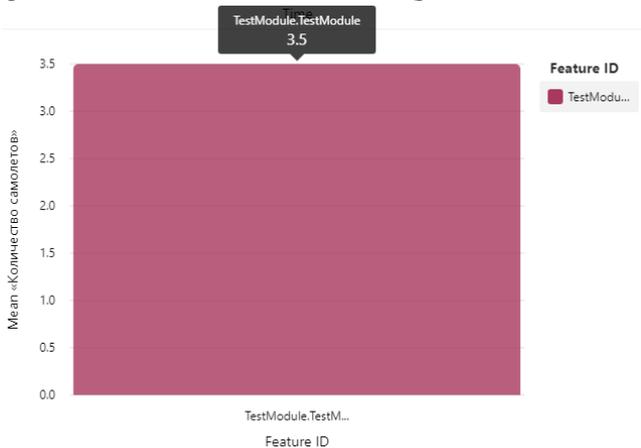


Рис. 23. Гистограмма среднего значения свойства “Количество самолетов”

Была также построена гистограмма, отображающая временное процентное соотношение каждого из значений (рис. 24). По данной гистограмме можно понять, как долго каждое из значений принадлежало данному свойству, относительно других.

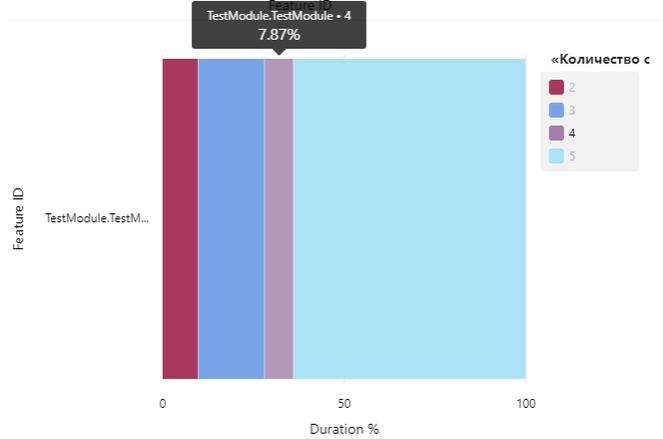


Рис. 24. Гистограмма зависимости значения свойства “Количество самолетов” от длительности соответствия конкретному объекту

Количество объектов типа “Airplane” в данном тестовом примере напрямую коррелирует со значением свойства “Количество самолетов” единственного объекта типа “TestModule” (рис. 25).

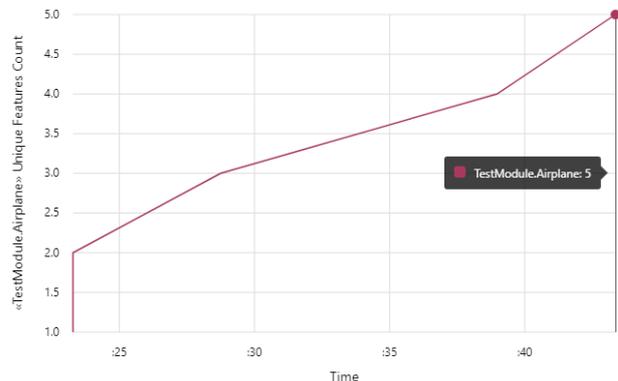


Рис. 25. Диаграмма зависимости количества уникальных объектов типа “Airplane” от времени

Свойство “Скорость” двух объектов типа “Airplane” заранее задано в значения 0 и 1000, при этом данное свойство у объекта со значением 1000 со временем инкрементируется. Значение данного свойства для трех оставшихся динамически созданных объектов (которые были добавлены на карту уже после инициализации модуля) задается случайно в диапазоне от 0 до 1000. Описанные значения свойства отображены на диаграмме линейной зависимости значения свойства от времени на рис. 26.

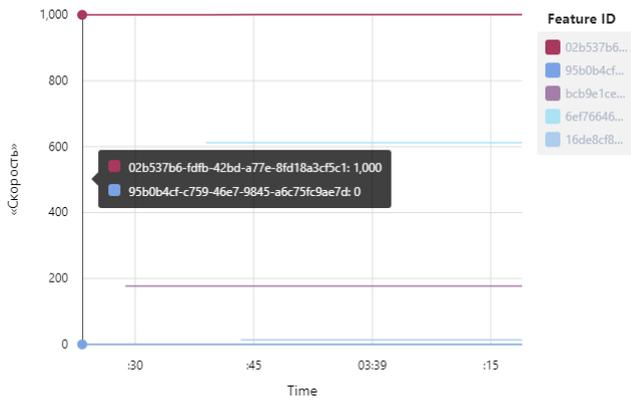


Рис. 26. Диаграмма зависимости значения свойства “Скорость” от времени

Кроме того, средние значения данного свойства для каждого из объектов соответствующего типа отображены на гистограмме на рис. 27.

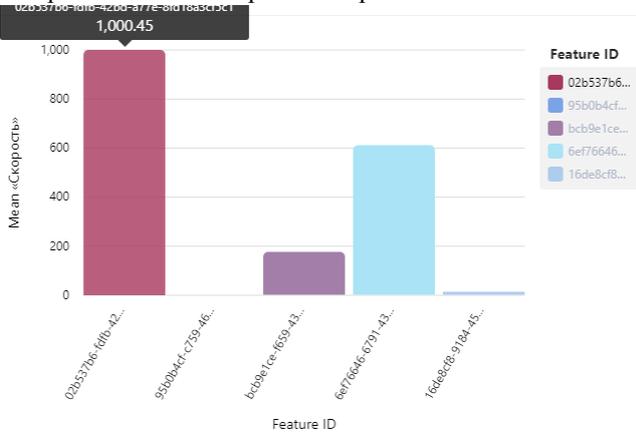


Рис. 27. Гистограмма среднего значения свойства “Скорость”

На гистограмме на рис. 28 отображены значения данного свойства, относительно продолжительности их соответствия каждому конкретному объекту.

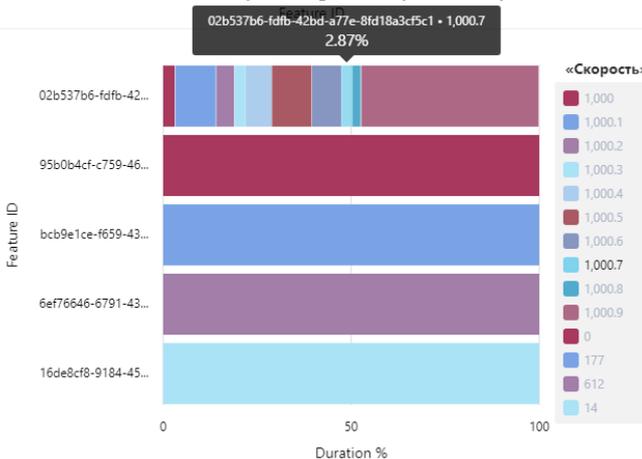


Рис. 28. Гистограмма зависимости значения свойства “Скорость” от длительности соответствия конкретному объекту

Свойство “Примечание” не является числовым, в связи с чем для него недоступны графические интерпретации в виде линейной зависимости величины от времени и вычисления среднего значения.

Для данного свойства в отчете представлена гистограмма соответствия значений данного свойства,

относительно продолжительности их соответствия каждому конкретному объекту (рис. 29). Изменения значений данного свойства выполнялись с использованием панели администрирования после запуска моделирования.

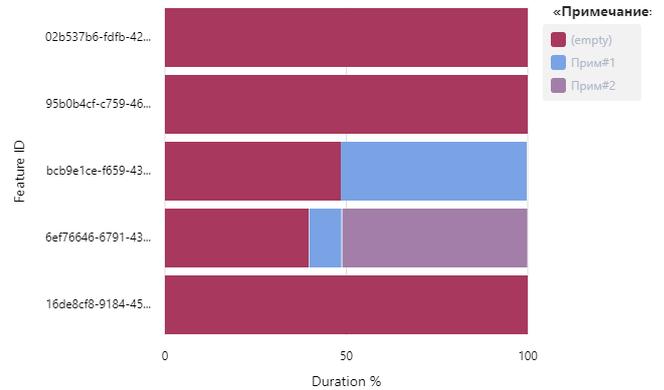


Рис. 29. Гистограмма зависимости значения свойства “Примечание” от длительности соответствия конкретному объекту

Для свойства “Компания” доступна только аналогичная гистограмма (рис. 30), что и для свойства “Примечание”, т.к. оно тоже не является числовым.

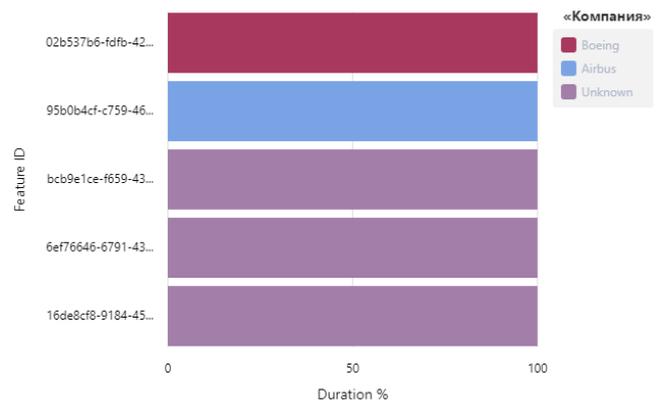


Рис. 30. Гистограмма зависимости значения свойства “Примечание” от длительности соответствия конкретному объекту

Н. Итоги раздела

В результате работы был создан функционал построения и отображения графических отчетов на основе данных платформы моделирования, отображающий результирующие данные о типах объектов, свойствах объектов и об объектах, участвующих в процессе моделирования.

Разработанный функционал призван упростить процесс принятия решений при использовании платформы моделирования.

Дальнейшими путями модернизации разработанного функционала могут быть:

- построение отчетов в реальном времени, а не по таймеру;
- увеличение количества отображаемых статистических данных (например, добавление гистограммы медианных значений свойств);
- вынесение части функционала на стороны панели администрирования, в целях снижения

вычислительной нагрузки на устройства конечных пользователей.

VII. ЗАКЛЮЧЕНИЕ

Для взаимодействия с платформой моделирования перемещений и взаимодействий акторов в рамках участка карты города был разработан web клиент, выполняющий роль панели администрирования и предоставляющий:

- пользовательское отображение акторов в реальном времени с конкретными стилями на основе типов акторов;
- возможность пользователю управлять списком модулей модели;
- возможность пользователю управлять жизненным циклом моделирования;
- возможность пользователю влиять на ход моделирования, путем изменения свойств модели и акторов;
- информацию для платформы моделирования о координатах и разрешении текущего пользовательского представления для дополнительных возможностей масштабирования и оптимизации.

В сравнении с существующими решениями данная панель администрирования отличается возможностью работы с конкретной платформой моделирования, которая в свою очередь предлагает ряд преимуществ, относительно других решений в сфере моделирования, таких как:

- кроссплатформенность;
- открытый исходный код;
- расширяемость функционала с помощью модулей.

Таким образом, разработанная панель администрирования делает платформу моделирования более доступной для взаимодействия с конечным пользователем, предоставляя графический пользовательский интерфейс, вместо программного (REST и gRPC API).

БЛАГОДАРНОСТИ

Исследование выполнено при финансовой поддержке гранта Российского научного фонда (РНФ, проект № 20-71-10087).

Авторы выражают благодарность коллегам по лаборатории UCLab за участие в разработке проекта Live.UrbanBasis.com.

БИБЛИОГРАФИЯ

- [1] Ant Road Planner [Электронный ресурс]. URL: <https://antroadplanner.ru/> (дата обращения: 21.10.2021).
- [2] NetLogo [Электронный ресурс]. URL: <https://www.netlogoweb.org/> (дата обращения: 21.10.2021).
- [3] AnyLogic [Электронный ресурс]. URL: <https://www.anylogic.ru/> (дата обращения: 21.10.2021).
- [4] Stehle S., Kitchin R. Real-time and archival data visualisation techniques in city dashboards // *International Journal of Geographical Information Science*. 2020. Vol. 34. P. 344-366.
- [5] Farmanbar M., Rong C. Triangulum City Dashboard: An Interactive Data Analytic Platform for Visualizing Smart City Performance // *Processes*. 2020. Vol. 8(2). P. 250.
- [6] Thompson E., Greenhalgh P., Muldoon-Smith K. et al. Planners in the Future City: Using City Information Modelling to Support Planners as Market Actors // *Urban Planning*. 2016. Vol. 1. P. 79-94.
- [7] Malinowski A., Czarnul P., Czurylo K. et al. Multi-agent large-scale parallel crowd simulation // *Procedia Computer Science*. 2017. Vol. 1. P. 917-926.
- [8] Simonov A., Lebin A., Shcherbak B. et al. Multi-agent crowd simulation on large areas with utility-based behavior models: Sochi Olympic Park Station use case // *Procedia Computer Science*. 2018. Vol. 136. P. 453-462.
- [9] Parygin D., Usov A., Burov S., Sadovnikova N., Ostroukhov P., Pyannikova A. Multi-agent Approach to Modeling the Dynamics of Urban Processes (on the Example of Urban Movements) // *Communications in Computer and Information Science: Proceedings of the 6th International Conference on Electronic Governance and Open Society: Challenges in Eurasia (EGOSE 2019)*, St. Petersburg, Russia, 13–14 November 2019. Springer, 2020. Vol. 1135. P. 243–257. DOI: 10.1007/978-3-030-39296-3_18
- [10] Парыгин Д.С., Буров С.С., Анохин А.О., Финогеев А.Г., Голубев А.В. Платформа для моделирования массовых перемещений объектов и субъектов в условиях городской среды // *Программные продукты и системы*. 2021. Т. 34. № 2. С. 354–364. DOI: 10.15827/0236-235X.134.354-364
- [11] Anokhin A., Burov S., Parygin D., Rent V., Sadovnikova N., Finogeev A. Development of Scenarios for Modeling the Behavior of People in an Urban Environment // *Studies in Systems, Decision and Control. Society 5.0: Cyberspace for Advanced Human-Centered Society*. Springer, 2021. Vol. 333. P. 103–114. DOI: 10.1007/978-3-030-63563-3_9
- [12] Парыгин Д.С. Управляемое данными развитие урбанизированных территорий: моногр. ВолгГТУ. Волгоград, 2021. 124 с.
- [13] Бурова А.А., Буров С.С., Парыгин Д.С., Финогеев А.А., Рент В.Э. Разработка модуля управления данными об объектах на онлайн-карте города // *Прикаспийский журнал: управление и высокие технологии*. 2021. № 1 (53). С. 18-27. URL: [https://hi-tech.asu.edu.ru/files/1\(53\)/18-27.pdf](https://hi-tech.asu.edu.ru/files/1(53)/18-27.pdf) (дата обращения: 21.10.2021).
- [14] Introduction to the Angular Docs [Электронный ресурс]. URL: <https://angular.io/docs> (дата обращения: 21.10.2021).
- [15] Ant Design of Angular [Электронный ресурс]. URL: <https://ng.ant.design/docs/introduce/en> (дата обращения: 21.10.2021).
- [16] lodash [Электронный ресурс]. URL: <https://github.com/lodash/lodash> (дата обращения: 21.10.2021).
- [17] swimlane /ngx-charts [Электронный ресурс]. URL: <https://github.com/lodash/lodash> (дата обращения: 21.10.2021).

Бурова Елена Алексеевна, Волгоградский государственный технический университет, магистрант, ORCID 0000-0002-6733-8386 (attari343@gmail.com)

Буров Сергей Сергеевич, Волгоградский государственный технический университет, магистрант, ORCID 0000-0001-7328-0953 (sergey.burovic@gmail.com)

Парыгин Данила Сергеевич, кандидат технических наук, доцент, Волгоградский государственный технический университет, ORCID 0000-0001-8834-5748 (dparygin@gmail.com)

Финогеев Алексей Германович, доктор технических наук, профессор, Пензенский государственный университет, ORCID 0000-0002-4777-3364 (alexeyfinogeev@gmail.com)

Смирнова Татьяна Владимировна, доцент, Волгоградский государственный социально-педагогический университет, ORCID 0000-0003-2861-3119 (smirnova.tatiana.2013@yandex.ru)

Administration panel of the multi-agent modeling platform with the ability to generate graphical reports

A. A. Burova, S. S. Burov, D. S. Parygin, A. G. Finogeev, T.V. Smirnova

Abstract— This article discusses the design and implementation of the web client that plays the role of an administration system for the platform for multi-agent modeling of movements and interactions of actors within a city map section. All modeling logic in this platform is implemented directly in modules, while the software platform only calls it for specific, connected modules. The administration system under development displays the current location and topology of the actors, has functionality to manage the state of the modeling and the list of modules involved in the modeling, as well as the ability to display and change the properties of the model and each actor separately. The state of the model (properties and location of actors) is updated in real time, using the gRPC system, without the need to reload the web page. The system's user interface includes a map and controls. The software platform is implemented on the ASP.NET Core 5.0 framework. For the implementation of the web client, the Angular 11 framework was chosen using the Ant Design UI components. In addition, methods for constructing graphical reports based on the received data on modeling have been developed for the administration system.

Keywords— modeling, simulation, city, C#, modularity, client-server, administration system.

REFERENCES

- [1] Ant Road Planner. URL: <https://antroadplanner.ru/> (access date: 21.10.2021).
- [2] NetLogo. URL: <https://www.netlogoweb.org/> (access date: 21.10.2021).
- [3] AnyLogic. URL: <https://www.anylogic.ru/> (access date: 21.10.2021).
- [4] Stehle S., Kitchin R. Real-time and archival data visualisation techniques in city dashboards // International Journal of Geographical Information Science. 2020. Vol. 34. P. 344-366.
- [5] Farmanbar M., Rong C. Triangulum City Dashboard: An Interactive Data Analytic Platform for Visualizing Smart City Performance // Processes. 2020. Vol. 8(2). P. 250.
- [6] Thompson E., Greenhalgh P., Muldoon-Smith K. et al. Planners in the Future City: Using City Information Modelling to Support Planners as Market Actors // Urban Planning. 2016. Vol. 1. P. 79-94.
- [7] Malinowski A., Czarnul P., Czurylo K. et al. Multi-agent large-scale parallel crowd simulation // Procedia Computer Science. 2017. Vol. 1. P. 917-926.
- [8] Simonov A., Lebin A., Shcherbak B. et al. Multi-agent crowd simulation on large areas with utility-based behavior models: Sochi Olympic Park Station use case // Procedia Computer Science. 2018. Vol. 136. P. 453-462.
- [9] Parygin D., Usov A., Burov S., Sadovnikova N., Ostroukhov P., Pyannikova A. Multi-agent Approach to Modeling the Dynamics of Urban Processes (on the Example of Urban Movements) // Communications in Computer and Information Science: Proceedings of the 6th International Conference on Electronic Governance and Open Society: Challenges in Eurasia (EGOSE 2019), St. Petersburg, Russia, 13-14 November 2019. Springer, 2020. Vol. 1135. P. 243-257. DOI: 10.1007/978-3-030-39296-3_18
- [10] Parygin D.S., Burov S.S., Anokhin A.O., Finogeev A.G., Golubev A.V. A platform for modeling mass movements of objects and subjects in an urban environment // Software products and systems. 2021. T. 34. № 2. P. 354-364. DOI: 10.15827/0236-235X.134.354-364 (In Russian)
- [11] Anokhin A., Burov S., Parygin D., Rent V., Sadovnikova N., Finogeev A. Development of Scenarios for Modeling the Behavior of People in an Urban Environment // Studies in Systems, Decision and Control. Society 5.0: Cyberspace for Advanced Human-Centered Society. Springer, 2021. Vol. 333. P. 103-114. DOI: 10.1007/978-3-030-63563-3_9
- [12] Parygin D.S. Data-driven development of urbanized territories: monograph. VSTU. Volgograd, 2021. 124 p. (In Russian)
- [13] Burova A.A., Burov S.S., Parygin D.S., Finogeev A.A., Rent V.E. Development of an object data management module on an online city map // Caspian Journal: Management and High Technologies. 2021. – № 1 (53). P. 18-27. URL: [https://hi-tech.asu.edu.ru/files/1\(53\)/18-27.pdf](https://hi-tech.asu.edu.ru/files/1(53)/18-27.pdf) (access date: 21.10.2021). (In Russian)
- [14] Introduction to the Angular Docs. URL: <https://angular.io/docs> (access date: 21.10.2021).
- [15] Ant Design of Angular. URL: <https://ng.ant.design/docs/introduce/en> (access date: 21.10.2021).
- [16] lodash. URL: <https://github.com/lodash/lodash> (access date: 21.10.2021).
- [17] swimlane /ngx-charts. URL: <https://github.com/lodash/lodash> (access date: 21.10.2021).

Burova Alena, Volgograd State Technical University, graduate student, ORCID 0000-0002-6733-8386 (attapi343@gmail.com)

Burov Sergey, Volgograd State Technical University, graduate student, ORCID 0000-0001-7328-0953 (sergey.burovic@gmail.com)

Parygin Danila, Candidate of Technical Sciences, Docent, Volgograd State Technical University, ORCID 0000-0001-8834-5748 (dparygin@gmail.com)

Finogeev Aleksey, Doctor of Technical Sciences, Professor, Penza State University, ORCID 0000-0002-4777-3364 (alexeyfinogeev@gmail.com)

Smirnova Tatiana, Docent, Volgograd State Pedagogical University, ORCID 0000-0003-2861-3119 (smirnova.tatiana.2013@yandex.ru)