

А.А. Антонов, Г.В. Дук, С.А. Чернышев

ФРЕЙМВОРК ДЛЯ СОЗДАНИЯ МУЛЬТИАГЕНТНЫХ СИСТЕМ

Аннотация. Разработана платформа для создания имитационных моделей, в основе которых лежит мультиагентный подход. Рассмотрены существующие аналоги, а также выявлены их достоинства и недостатки. Фреймворк позволяет реализовать любую имитационную модель независимо от предметной области. Представлена архитектура ядра, стек технологий разработки. Для демонстрации работы создана имитационная модель предметной области «грузоперевозки».

Ключевые слова: мультиагентная система, фреймворк, имитационная модель, грузоперевозки, C++, AnyLogic, Jade, GAMA, NetLogo.

A.A. Antonov, G.V. Duk, S.A. Chernyshev

FRAMEWORK FOR CREATION MULTI-AGENT SYSTEMS

Abstract. The platform has been developed for creating simulation models based on a multi-agent approach. The existing analogues were considered in the work, as well as their advantages and disadvantages were identified. The framework allows you to implement any simulation model regardless of the subject area. The article presents: core architecture, development technology stack. In conclusion, a simulation model of the subject area «Cargo transportation» was created to demonstrate the work.

Keywords: multi-agent system, framework, simulation model, cargo transportation, C++, AnyLogic, Jade, GAMA, NetLogo.

Введение

На фоне бурного роста технологий выделяются те сферы деятельности, в которых все вычислительные процессы всё еще выполняет человек. В качестве примера можно привести профессию диспетчера. Выполнять задачу в полном объеме не способен ни один человек, а в случае диспетчеров невозможно учесть все влияющие факторы, рассмотреть все возможные варианты и принять наиболее правильное решение за доли секунды. Именно поэтому еще с прошлого века начали разрабатываться алгоритмы по автоматизации бизнес-процессов: алгоритмы обнаружения проблем, прогнозирования событий, оценка результатов и др. Существует множество методов, способных объединить в себе эти алгоритмы и реализовать в виде кода. Самый распространенный – метод имитационного моделирования.

Имитационное моделирование является отличным инструментом для решения трудноформализуемых задач, где присутствует большое количество различных факторов. Такие задачи часто встречаются в следующих областях: ИТ-инфраструктура, поставка товаров, демографический контроль, космическая отрасль и др.

При создании моделей разных областей можно выделить общие характеристики и функции. В связи с этим возникает потребность в создании инструмента, который будет инкапсулировать общие понятия и предоставлять удобный интерфейс для расширения под конкретную область.

На сегодняшний день многие компании, в том числе и крупные, сосредоточили большие силы вокруг имитационного моделирования. Каждая из них стремится привести

Антонов Александр Александрович

магистрант. Санкт-Петербургский государственный университет аэрокосмического приборостроения, Санкт-Петербург. Сфера научных интересов: имитационное моделирование; мультиагентные системы; разработка распределенных систем. Автор 6 опубликованных научных работ.

Электронный адрес: grand.antonov@yandex.ru

Дук Герман Викторович

бакалавр. Санкт-Петербургский государственный университет аэрокосмического приборостроения, Санкт-Петербург. Сфера научных интересов: системы поддержки принятия решений; разработка распределенных систем. Автор 6 опубликованных научных работ.

Электронный адрес: ger-duk@mail.ru

Чернышев Станислав Андреевич

кандидат технических наук, доцент кафедры интеллектуальных систем и защиты информации. Санкт-Петербургский государственный университет промышленных технологий и дизайна; доцент кафедры информатики. Санкт-Петербургский государственный экономический университет, Санкт-Петербург. Сфера научных интересов: имитационное моделирование; мультиагентные системы; системы поддержки принятия решений; разработка распределенных систем. Автор более 40 опубликованных научных работ.

Электронный адрес: chernyshev.s.a@bk.ru

что-то свое. Однако, проанализировав текущую ситуацию, можно отметить, что проблема нехватки программного обеспечения с применением имитационного моделирования очень актуальна. Основная причина связана с тем, что компании не хотят вкладывать большие средства в разработку системы с нуля, не имея инструмента. В интернете можно найти много фреймворков по разработке имитационных моделей, но многие из них совершенно не востребованы [4; 5]. Рассмотрим самые актуальные.

Обзор и анализ аналогов

Одной из самых ярких сред для разработки мультиагентных систем является JADE. JADE (Java Agent Development Framework) – программная платформа, полностью реализованная на языке Java, что упрощает разработку мультиагентных систем с помощью промежуточного программного обеспечения, соответствующего спецификациям FIPA, и набора графических инструментов, поддерживающих этапы отладки, а также развертывания [8]. Основные преимущества фреймворка JADE:

многопоточность – каждый агент имеет свой поток управления, а их взаимодействие осуществляется в многопоточном режиме;

агентная платформа – платформа, основанная на стандарте FIPA, при запуске которой все добавленные агенты начинают выполнять действия согласно заложенной онтологии.

распределенная платформа подразумевает запуск всей системы в одноранговой сети. Это означает, что каждый узел равноправный и выполняет функции как сервера, так и клиента.

графический интерфейс позволяет отлаживать, создавать, уничтожать просматривать и контролировать действия агентов.

Из-за быстрых темпов роста технологий отсутствие обновлений сильно сказалось на актуальности JADE.

AnyLogic – инструмент имитационного моделирования, поддерживающий все основные подходы к разработке имитационных моделей [6]. При разработке пользователю предоставляется возможность написания собственных классов и интерфейсов на языке Java. Вместе с этим разработчику предоставляются и отраслевые библиотеки:

- моделирования процессов;
- моделирования потоков;
- железнодорожная;
- пешеходная;
- дорожного движения;
- производственных и транспортных систем.

Отдельно стоит выделить железнодорожную библиотеку. Она является одним из главных достоинств AnyLogic и часто демонстрируется на вебинарах. Но на деле доступа к железнодорожным путям практически нет, так как большинство станций на карте показаны в виде одной линии без полной схемы путевого хозяйства. Также данные, находящиеся в шейп-файлах, часто бывают не актуализированы, поэтому в большинстве случаев железная дорога не соответствует реальности.

AnyLogic позволяет разрабатывать модели с использованием различных тайл-серверов и шейп-файлов. И это кажется решением проблемы с неактуальностью данных, то есть нужно сформировать свой шейп-файл и импортировать его. Однако, не имея в наличии профессиональной версии программного обеспечения, это сделать невозможно. Механизм, заложенный в импорт шейп-файлов, не до конца проработан, поэтому пользователь может потратить несколько часов, чтобы привести импортируемые данные к нужному формату. Явным недостатком является также отсутствие возможности взаимодействовать со сторонними тайл-сервисами, что значительно снижает вариативность создаваемых моделей.

Что касается использования данного продукта при обучении – тоже всё не так однозначно. Бесплатная версия имеет ряд критических ограничений [3], которые хотя и можно обойти без нарушения лицензионных соглашений, но это требует от обучающихся хорошего знания языка программирования Java и принципов объектно ориентированного программирования.

Другим аналогом является RepastSimphony. Это бесплатный набор инструментов для разработки мультиагентных программ с открытым исходным кодом. Основным целевым языком является Java [10]. Он предлагает пользователям широкий спектр функций, включая следующее:

- графический интерфейс с моделью «укажи и щелкни»;
- средства логирования и построения графиков результатов;
- возможность моделировать в 2D- и 3D-среде;
- поддержка геоинформационных систем (далее – ГИС);
- возможность заморозки, хранения и воспроизведения моделей;
- наличие многопоточного планировщика дискретных событий.

Практика показывает, что Repast – это обычная синхронная система, в которой все действия выполняются последовательно. Это значит, что какая бы мультиагентная система ни была создана, она будет иметь существенно другую семантику выполнения, чем в распределенной системе. В Repast отсутствует обмен сообщениями между агентами. И это весомый минус, который не позволяет разработать множество систем, связанных постоянным обменом информации.

GAMA – плагин с открытым исходным кодом для интегрированной среды разработки Eclipse [7]. Разработка этой платформы началась еще в 2007 году и продолжается сейчас. С тех пор в нее было заложено множество функциональных возможностей: поддержка ГИС, многоуровневое моделирование моделей, встроенное управление базами данных, а также возможность подключения физического 3D-движка. GAMA предоставляет пользователю написать свой пользовательский графический интерфейс, однако из-за отсутствия интуитивно понятного механизма его создания пользователь тратит большое количество времени на разработку. Данная платформа имеет собственный язык программирования, что также плохо сказывается на ее популярности. Поэтому, несмотря на все плюсы, в крупных коммерческих проектах GAMA не используется. По своим параметрам он отлично подходит для обучения или создания небольших прототипов.

NetLogo имеет собственный синтаксис, что уже является большим недостатком при интегрировании в современные проекты [9]. Присутствуют примитивы для псевдоасинхронного выполнения, но они недостаточно мощны для моделирования действительно асинхронной системы. Как и в Repast, отсутствует система обмена сообщениями между агентами. Но главным достоинством NetLogo является возможность создания быстрого прототипа системы, после чего можно принимать решения о дальнейшем развитии. Также данная платформа отлично подходит для обучения и изучения основ мультиагентного моделирования.

Требования к разрабатываемому фреймворку

Проанализировав наиболее успешные платформы, можно сформулировать основные требования к современным средам разработки распределенных систем мультиагентного моделирования:

- использование современных языков программирования;
- поддержка современных стандартов языка;
- удобный пользовательский графический интерфейс;
- непривязанность к предметной области;
- наличие многопоточности;
- возможность запускать модели на нескольких устройствах;
- постоянная поддержка.

Стек технологий

В качестве основного языка программирования был выбран C++, так как он является объектно ориентированным языком, что позволяет проще реализовать большой функционал с минимальным повторением, а также ускоряет процесс написания программы и делает код легко читаемым. Эти качества важны как отдельному пользователю, так и целым компаниям. При этом программы, написанные на C++, выполняются в несколько раз быстрее, чем на других языках.

Первое, на что обращает внимание пользователь, – это графический интерфейс. Зачастую пользователи, работающие с имитационными моделями, далеки от программирования, поэтому помимо того, что они могут самостоятельно настроить интерфейс, первоначально должен предоставляться весь инструментарий, позволяющий взаимодействовать с моделью, не влезая в код. Наиболее подходящей для этого является библиотека Qt, позволяющая вести кроссплатформенную разработку GUI на C++, а также предоставляющая собственный сигнал/слотовый механизм для более простой организации межпоточного взаимодействия между объектами.

Структура фреймворка

Структурная схема архитектуры разработанного фреймворка представлена на Рисунке 1.

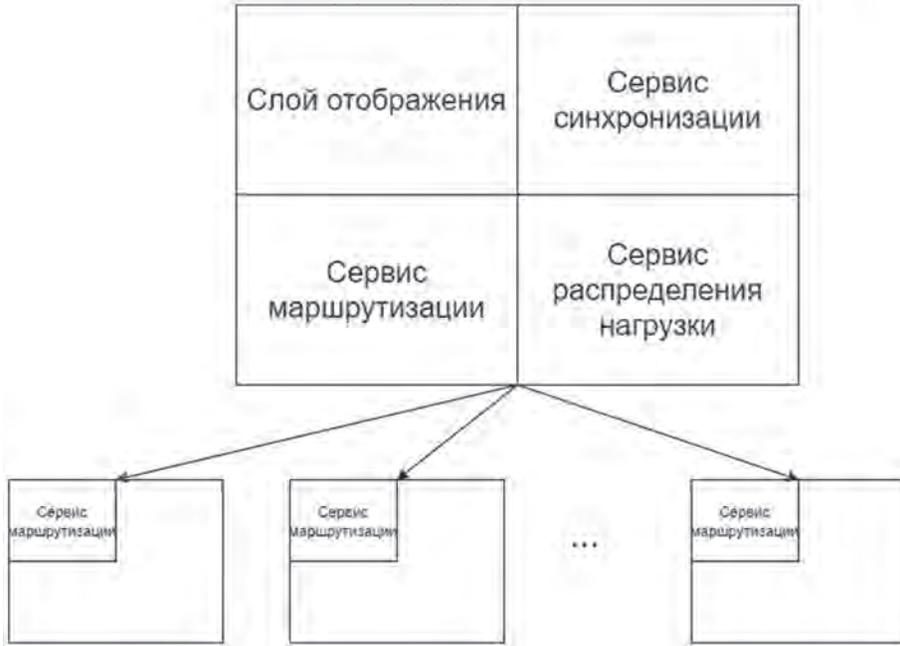


Рисунок 1. Структурная схема архитектуры фреймворка

В верхней части рисунка изображен основной блок, включающий четыре составляющих:

1. Слой отображения. Данный компонент отвечает за UI разрабатываемой модели. Пользователь имеет возможность подгрузить свой графический компонент – карту, макет, графики и др.

2. Сервис синхронизации. Перед тем как перейти к описанию сервиса, необходимо сказать, что в основе имитационной платформы заложено комбинирование дискретно-событийного и мультиагентного подхода. Основная задача дискретно-событийного метода – замена реального времени на дискретное, из-за чего система представлена в виде последовательности событий, переводящих систему из одного состояния в другое [1]. В связи с этим сервис синхронизации является реализацией данного подхода и позволяет синхронизировать действия всех агентов по времени. Кроме того, он отвечает за контроль времени, позволяя его ускорять, замедлять или полностью останавливать.

3. Сервис маршрутизации. Как описывалось ранее, одно из главных требований к разработке фреймворка – возможность разворачивать платформу на нескольких устройствах, следовательно, разрабатываемая система должна быть распределенной. В зависимости от предметной области и сложности количество агентов каждой популяции может насчитываться от нескольких единиц до десятков миллионов. Поэтому в разрабатываемой системе агенты могут находиться как на одном устройстве, так и на разных. Однако их расположение никак не должно влиять на процесс межагентного взаимодействия. Для этого был создан сервис маршрутизации. Он осуществляет обмен сообщениями между агентами как локально, так и между устройствами. Общение между устройствами осу-

ществляется по TCP/IP с помощью сокетов. На рисунке под основным блоком можно увидеть несколько небольших кластеров. Это отдельные устройства, на которых агенты выполняют свои действия. Сервис маршрутизации находится также на каждом кластере, что позволяет не только принимать информацию от главного устройства, но и осуществлять полноценный обмен информацией с остальными участниками сети.

4. Сервис распределения нагрузки. Участниками сети могут стать почти любые устройства: мощные и слабые компьютеры, ноутбуки, планшеты и даже телефоны. В связи с этим должен осуществляться постоянный контроль за загруженностью того или иного кластера. При наличии большого количества свободных ресурсов кластеру будет передано несколько агентов, чтобы разгрузить другое устройство. При распределении нагрузки учитывается множество факторов, среди которых техническая составляющая устройства.

Демонстрация работы фреймворка

Для демонстрации работы средствами фреймворка была разработана имитационная модель системы поставки продуктов по магазинам. Данный выбор связан с тем, что предметная область товароснабжения очень обширна и содержит множество динамически изменяющихся факторов, которые в комплексе сильно влияют на конечный результат. Разрабатываемая модель максимально приближена к реальному миру, следовательно, все участники системы конфликтуют за главный ресурс – деньги. Получение прибыли при наименьших затраченных ресурсах – главная цель каждого участника. Поэтому основной целью имитационной модели является воссоздание реальной рыночной ситуации, в которой система может проанализировать правильность принимаемых решений и найти наиболее благоприятные из возможных. Это позволяет разрабатывать структуру логистических цепей поставок товаров по магазинам, оперативно принимать решения, а также получать подробный анализ при изменениях внутренних и внешних факторов.

Модель включает 3 популяции агентов: магазины, склады и транспортные средства. Начинается симуляция с расстановки агентов по карте и ввода начальных значений (см. Рисунки 2, 3). Все агенты имеют пересекающиеся параметры (например, время работы), располагаемые деньги, тип, количество сотрудников и др. Помимо общих, агенты имеют характеристики, которые принадлежат только их популяции. Например, для транспорта – это скорость и грузоподъемность, для магазина – время заказа и время анализа продаж, для склада – количество станций погрузки. Использование в системе имитационного времени позволяет скоординировать все действия агентов по времени, следовательно, процесс анализа и внесения изменений во время симуляции становится более удобным и наглядным.

После старта каждый агент поэтапно начинает выполнять свои действия со скоростью шага имитационного времени. Глобально логика системы заключается в том, что при нехватке продуктов в магазине запускается система поиска поставщика из возможных складов, после чего идет поиск транспорта, который доставит недостающий товар. Однако локально логика каждого агента гораздо сложнее. Примером может служить логика продажи товаров в магазине. Запрограммировать продажу товаров в магазине практически невозможно, так как существует неопределенность, при которой трудно найти прямую зависимость между внешними факторами и продажами в магазине. Поэтому был создан модуль, в основе которого лежит нейронная сеть, способная в зависимости от текущей даты и заселенности района выдать предполагаемое количество товара, проданного за день в магазине.

Фреймворк для создания мультиагентных систем

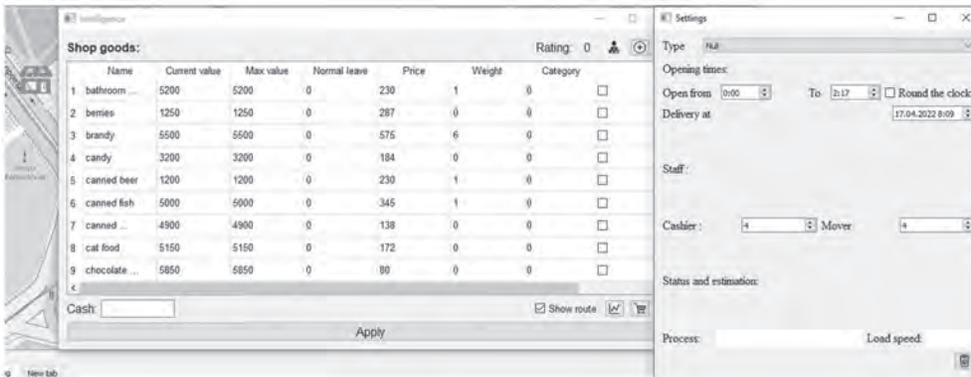


Рисунок 2. Ввод параметров магазина

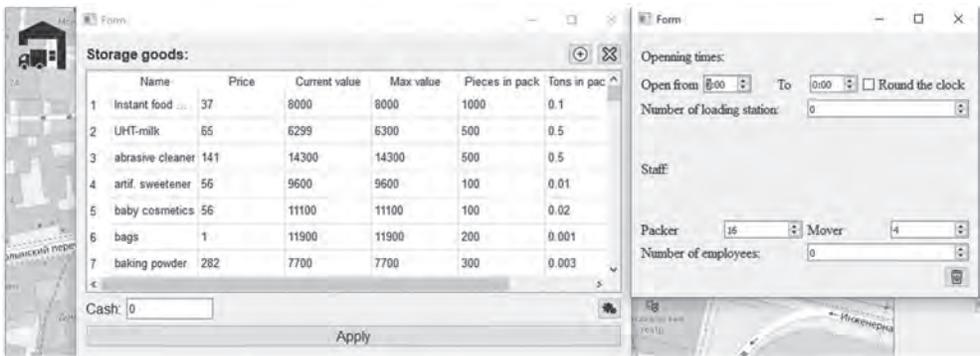


Рисунок 3. Ввод параметров склада

На Рисунке 4 приведен пример транспортировки товаров со склада в магазин.

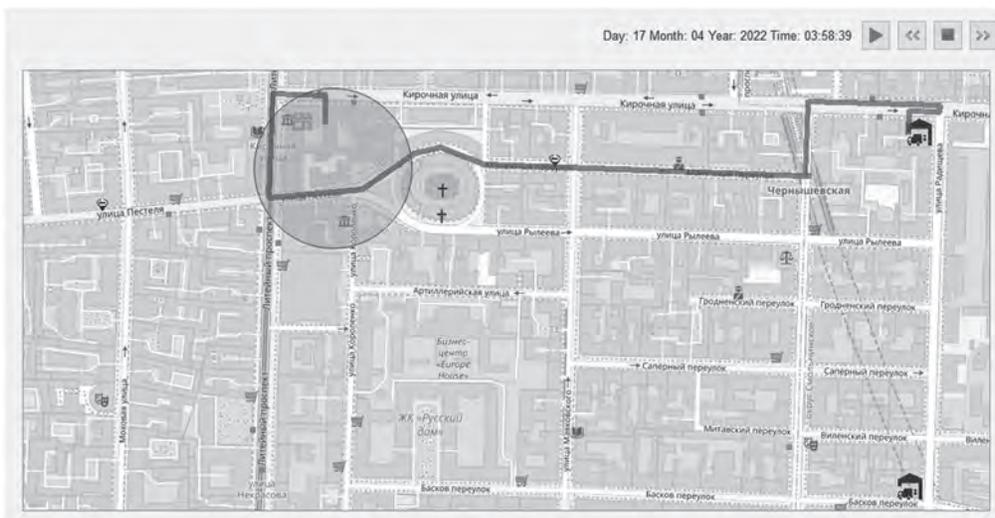


Рисунок 4. Доставка товаров со склада в магазин

Результат работы имитационной модели можно наблюдать по нескольким критериям. У каждого агента имеется стохастический осциллятор, который показывает, как менялось соотношение прибыли к затратам в течение нескольких дней (см. Рисунок 5) [2]. Помимо этого у магазина имеется рейтинг, который меняется в конце каждого дня.

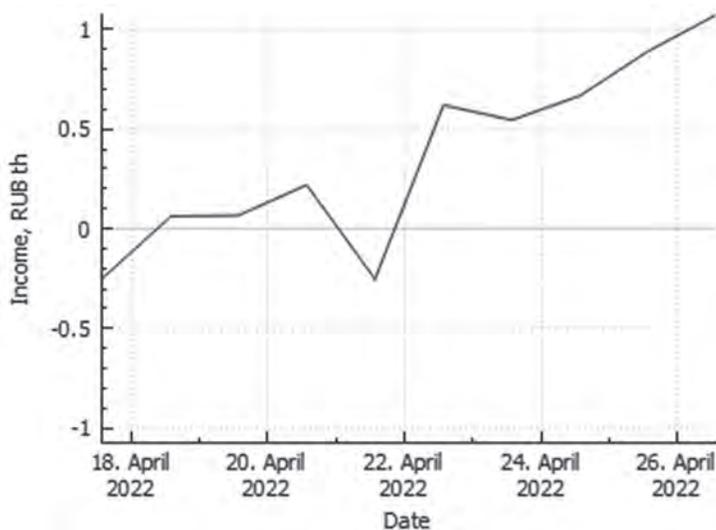


Рисунок 5. График изменения дохода магазина

Заключение

Популярность имитационного моделирования за последние годы сильно возросла. Главным ее преимуществом является решение трудноформулируемых задач, в которых невозможно найти единственно правильное решение. Однако, проанализировав существующие платформы для реализации имитационных моделей, можно выделить ключевые ошибки – устаревшие технологии и ограничение по применению. Важность этих критериев при проектировании системы крайне велика, а игнорирование может вылиться в серьезные последствия в будущем. Именно по этой причине множество платформ уже не так актуальны, хотя были выпущены не более 3 лет назад.

Предложенный фреймворк нацелен на исправление перечисленных проблем. Представлена его архитектура, благодаря которой платформа имеет высокую производительность. Данная система является распределенной и может быть развернута практически на любой платформе. Одно из главных достоинств – фреймворк написан на языке C++. Кроме того, пользователю также предоставляется полный контроль над изменением графического интерфейса.

В настоящее время осуществляется получение свидетельства государственной регистрации программы для ЭВМ, после чего исходный код разработанного фреймворка будет выложен в открытый доступ для некоммерческого использования.

Литература

1. Даденков С.А., Кон Е.А. Анализ моделей и методов агентного и дискретно-событийного имитационного моделирования // Известия СПбГЭТУ «ЛЭТИ». 2015. № 5. С. 35–41.

2. Жмурко Д.Ю. Визуально-графический анализ показателей деятельности интегрированных производственных систем сахарного подкомплекса АПК. Часть 1. Постановка задачи и моделирование // Научный журнал КубГАУ. 2016. № 122 (08). С. 1–27.
3. Чернышев С.А. Разработка имитационных моделей в условиях ограничений AnyLogic PLE // Вестник Санкт-Петербургского государственного университета технологии и дизайна. Серия 1: Естественные и технические науки. 2021. № 1. С. 53–58. DOI: 10.46418/20798199_2021_1_8.
4. Чернышев С.А., Кияев В.И., Газуль С.М. Использование платформы Python для разработки мультиагентных систем // Вестник Санкт-Петербургского государственного университета технологии и дизайна. Серия 1: Естественные и технические науки. 2021. № 4. С. 82–88. DOI: 10.46418/2079-8199_2021_4_12.
5. Чернышев С.А., Филяев М.П., Николаев П.А. Сравнительный анализ инструментальных программных средств для разработки имитационных моделей логистических процессов // Научные проблемы материально-технического обеспечения Вооруженных сил Российской Федерации. 2020. № 2 (16). С. 92–104.
6. AnyLogic. Официальный сайт [Электронный ресурс]. URL: <https://www.anylogic.ru/> (дата обращения: 21 марта 2022).
7. GAMA. Официальный сайт [Электронный ресурс]. URL: <https://gama-platform.org/> (дата обращения: 22 марта 2022).
8. JADE. Официальный сайт [Электронный ресурс]. URL: <https://jade.tilab.com/> (дата обращения: 20 марта 2022).
9. NetLogo. Официальный сайт [Электронный ресурс]. URL: <https://ccl.northwestern.edu/netlogo/> (дата обращения: 23 марта 2022).
10. Repast. Официальный сайт [Электронный ресурс]. URL: <https://repast.github.io/> (дата обращения: 21 марта 2022).

References

1. Dadenkov С.А., Kon E.L. (2015) *Analiz modelej i metodov agentnogo i diskretno-sobyitjnogo imitacionnogo modelirovanija* [Analysis of models and methods of agent-based and discrete-event simulation modeling]. *Izvestiya SPbSETU «LETI»*, No. 5, pp. 35–41 (in Russian).
2. Zhmurko D.Yu. (2016) *Postanovka zadachi i modelirovanie* [Problem statement and modeling]. *Nauchnyj zhurna IKubGAU*, No. 122 (08), pp. 1–27 (in Russian).
3. Chernyshev S.A. (2021) *Razrabotka imitacionnyh modelej v uslovijah ogranichenij AnyLogic PLE* [Development of simulation models under AnyLogic PLE constraints]. *Vestnik Sankt-Peterburgskogo gosudarstvennogo universiteta tehnologii i dizajna. Serija 1: Estestvennye i tehnicheckie nauki*, No. 1, pp. 53–58 (in Russian). DOI: 10.46418/20798199_2021_1_8.
4. Chernyshev S.A., Kiyayev V.I., Gazul S.M. (2021) *Ispol'zovanie platformy Python dlja razrabotki mul'tiagentnyh sistem* [Using the python platform for the development of multi-agent systems]. *Vestnik Sankt-Peterburgskogo gosudarstvennogo universiteta tehnologii i dizajna. Serija 1: Estestvennye i tehnicheckie nauki*, No. 4, pp. 82–88 (in Russian). DOI: 10.46418/2079-8199_2021_4_12.
5. Chernyshev S.A., Filyayev M.P., Nikolaev P.A. (2020) *Sravnitel'nyj analiz instrumental'nyh programmyh sredstv dlja razrabotki imitacionnyh modelej logisticheski hprocessov* [Comparative analysis of instrumental software tools for the development of simulation models of logistics processes]. *Nauchnye problem material'no-tehnicheskogo obespechenija Vooruzhennyh sil Rossijsko jFederacii*, No. 2 (16), pp. 92–104 (in Russian).

6. AnyLogic. Official site. Available at: <https://www.anylogic.ru/> (date of the application: 21 March 2022).
7. GAMA. Official site. Available at: <https://gama-platform.org/> (date of the application: 22 March 2022).
8. JADE. Official site. Available at: <https://jade.tilab.com/> (date of the application: 20 March 2022).
9. NetLogo. Official site. Available at: <https://ccl.northwestern.edu/netlogo/> (date of the application: 22 March 2022).
10. Repast. Official site. Available at: <https://repast.github.io/> (date of the application: 21 March 2022).