



ИПМ им.М.В.Келдыша РАН • Электронная библиотека

Препринты ИПМ • Препринт № 74 за 2022 г.



ISSN 2071-2898 (Print)
ISSN 2071-2901 (Online)

В.А. Судаков, Т.В. Сивакова

Унифицированная программа
прогнозирования
распространения вирусных
инфекций

Статья доступна по лицензии
Creative Commons Attribution 4.0 International



Рекомендуемая форма библиографической ссылки: Судаков В.А., Сивакова Т.В. Унифицированная программа прогнозирования распространения вирусных инфекций // Препринты ИПМ им. М.В.Келдыша. 2022. № 74. 12 с. <https://doi.org/10.20948/prepr-2022-74>
<https://library.keldysh.ru/preprint.asp?id=2022-74>

**Ордена Ленина
ИНСТИТУТ ПРИКЛАДНОЙ МАТЕМАТИКИ
имени М.В.Келдыша
Российской академии наук**

В.А. Судаков, Т.В. Сивакова

**Унифицированная программа
прогнозирования распространения
вирусных инфекций**

Москва — 2022

Судаков В.А., Сивакова Т.В.

Унифицированная программа прогнозирования распространения вирусных инфекций

За последние несколько лет в условиях распространения коронавирусной инфекции существенно возросла роль имитационного моделирования. Разработка и внедрение в практику мультиагентных моделей помогает не только спрогнозировать количество заболевших и предсказать возможные пики заболеваемости, но также служит оценкой для проведения необходимых мероприятий по сдерживанию распространения вируса, использованию средств индивидуальной защиты, введению ограничительных мер, связанных с работой организаций и транспорта. Разработанная программа распространения вирусных инфекций построена на основе мета-классов объектов и позволяет учитывать «большое» число контактов. Программа была реализована на примере распространения коронавирусной инфекции. В качестве геосервисов для населенных пунктов использовались открытые данные OpenStreetMap (OSM).

Ключевые слова: унифицированная программа, мета-класс, имитационное моделирование, симуляция, коронавирусная инфекция, Overpass Turbo

Vladimir Anatolievich Sudakov, Tatiana Vladimirovna Sivakova

Unified program for predicting the spread of viral infections

Over the past few years, in the context of the spread of coronavirus infection, the role of simulation has increased significantly. The development and implementation of multi-agent models in practice helps not only to predict the number of cases and predict possible peaks in incidence, but also serves as an assessment for taking the necessary measures to contain the spread of the virus, use personal protective equipment, and introduce restrictive measures related to the work of organizations and transport. The developed program for the spread of viral infections is built on the basis of meta-classes of objects and allows you to take into account a "large" number of contacts. The program was implemented on the example of the spread of coronavirus infection. OpenStreetMap (OSM) open data were used as geoservices for settlements.

Key words: unified program, metaclass, simulation, simulation, coronavirus infection, overpass Turbo

Исследование выполнено при финансовой поддержке РФФИ и CNPq (Бразилия), Фонда содействия инновациям (Россия), DBT, DST (Индия), MOST, NSFC(Китай), SAMRC(ЮАР) в рамках научного проекта № 20-51-80002.

Введение

Актуальность создания моделей распространения инфекций в современных условиях не вызывает сомнения. Можно выделить два основных класса моделей: макромоделей и микромоделей. В макромоделях человек обычно не представлен в качестве самостоятельного объекта, а рассчитывается лишь число людей в тех или иных состояниях. Наиболее типичные макромоделей – это компартментальные модели [1]. Подобные модели предполагают описание пандемий через систему дифференциальных уравнений, которые хорошо описывают ситуацию в типовых условиях. Значимым преимуществом моделей, базирующихся на аппарате дифференциальных уравнений, является возможность их аналитического исследования. Тем не менее для всех таких моделей характерно еще одно допущение: характеристики и поведение всех индивидов, отнесенных к одной подгруппе, считаются одинаковыми. Кроме того, с помощью compartmental models проблематично оценить последствия тех или иных управленческих решений, например, по изменению режима работы ресторанов. Compartmental models содержат ряд коэффициентов; если бы было известно, как они изменятся при наших мягких управляющих воздействиях, то макромоделей были бы отличным инструментом. Но обычно мы не можем этого знать. Микромоделей – это прежде всего дискретные имитационные модели, которые могут быть созданы в парадигмах императивного или мультиагентного программирования и легко позволяют учесть наши управленческие решения на уровне правил обработки событий и поведения агентов. Специализированные языки и инструментальные средства имитационного моделирования преимущественно создавались для решения задач бизнеса: производственных задач, логистических и транспортных задач. Хороший, но достаточно старый инструментарий – это система GPSS [2]. Она позволяет декларативно описать многие бизнес-процессы в терминах транзактов, которые обрабатываются определенными блоками. Аналогичные решения есть и среди современных систем, например, симуляция в нотации BPMN [3]. Данная работа преследует цель разработки аналогичного преимущественно декларативного инструментария для другой предметной области – моделирования контактов больших масс людей и распространения инфекции среди них.

Обзор моделей

Первыми имитационными моделями контактов были клеточные автоматы [4], которые представляют собой совокупность ячеек, каждая из которых принимает состояние из конечного множества. Обычно используются квадратные ячейки, объединенные в прямоугольную решетку. Тогда узлы решетки моделируют индивидов, каждый из которых имеет фиксированное положение в пространстве. Время для клеточного автомата движется дискретно, на каждом шаге инфекция может с определенной вероятностью предаться от больного индивида к его восприимчивому соседу. Клеточный

автомат для большого числа людей будет достаточно неэффективно функционировать как по скорости, так и по памяти на обычных процессорах. Исправить ситуацию с производительностью может реализация клеточных автоматов на графических процессорах. Основным ограничением клеточного автомата является моделирование окружающего индивидов пространства как решетки. В действительности для большинства заболеваний можно считать, что инфекция распространяется посредством контактов между людьми.

Контакты между индивидами не ограничиваются непосредственными соседями – они носят нерегулярный во времени и пространстве характер. Учесть их позволяют сетевые модели [5], которые представляют схему возможных контактов в виде графа. Вершины графа здесь — объекты с набором индивидуальных свойств, сколь угодно детализированно описывающие состояние отдельных индивидов. Подобные модели также достаточно ресурсоемки, но эффективно реализуются на графических процессорах [6,7]. Сетевые модели предлагают общую технологию моделирования окружающего пространства, но не специфицируют способ организации графа контактов для населения конкретной территории.

Наиболее детальные модели на сегодня — это agent-based models [8]. Они не используют централизованный расчет вероятностей передачи инфекции, а описывают поведение каждого человека (агента) индивидуально. Большинство агентных моделей дискретно-событийные, то есть процесс их моделирования сводится к последовательной обработке списка запланированных событий. Каждое событие влечет за собой некоторый набор атомарных действий, меняющих состояние системы. Агенты самостоятельно формируют события, связанные с их деятельностью, что позволяет избежать фиксированного шага дискретизации и сложных формул вероятности инфицирования. Агентные модели — максимально реалистичные, но требуют существенных вычислительных мощностей. Их калибровка и анализ, необходимые для использования модели в качестве инструмента прогнозирования, базируются на проведении множества численных экспериментов. Агентные модели позволяют сформировать прогноз на любой срок, но их исследование и практическое применение значительно сложнее других техник. Агентные модели разрабатываются и настраиваются для решения конкретных задач, особенности которых отражают в самой модели и значениях ее параметров.

Наиболее естественным является формирование связей между индивидами и возможных путей распространения инфекции на основе их социальной принадлежности. Такой способ применяется в population-based моделях [9].

Окружающее пространство популяционных моделей разбито на локации — места, в которых индивиды могут контактировать, например, школы, офисы, жилые помещения и проч. В зависимости от возрастной и социальной группы индивида он связан с теми или иными локациями — ежедневно «посещает» их. Время в популяционных моделях движется дискретно, обычно с шагом в 12 или

24 часа. На каждом шаге рассчитывается вероятность заражения каждого индивида по формулам, учитывающим число инфицированных в посещаемых индивидом локациях. Популяционные модели позволяют представить структуру общества любой сложности, но требуют тщательной настройки параметров на основе эмпирических данных.

Достаточно мощным средством моделирования, в том числе и для пандемий, является AnyLogic [10]. Но оно не содержит готовых специальных классов для создания локаций и требует существенных затрат ресурсов на разработку и отладку модели даже небольшого города. Для реализации моделей контактов предпочтительно применять специализированные программные средства. Одна из специализированных систем «COVID-19 Outbreak Симулятор вспышки COVID-19» — это популяционный симулятор, который имитирует распространение вируса SARS-CoV-2 в динамичной и неоднородной популяции с учетом профилактических мер и планов действий после вспышки, таких как карантин, тестирование и устранение случаев с симптомами. [11]. Однако и другие рассмотренные программы не претендуют на универсальность. Данные системы максимум дают возможность варьировать параметры, при этом для изменения структуры учитываемых классов объектов приходится модифицировать программный код. Поэтому актуальной задачей является разработка унифицированного программного продукта, в котором пользователь, определив характеристики классов в декларативном стиле, может без использования императивного программирования запускать различные сценарии поведения людей.

Постановка задачи

Метод решения задачи моделирования контактов заключается в построении вариативных моделей путем автоматической генерации необходимого числа готовых блоков из описания разных типов пространственных, эпидемических, транспортных взаимодействий на уровне атрибутов мета-классов.

Дадим описание некоторых основных мета-классов.

Все местонахождения людей привязаны к локациям определенного типа. Для описания типов локаций служит мета-класс `location_types`. Его экземплярами являются конкретные классы локаций: кинотеатры, магазины, квартиры, офисы. В свою очередь при генерации конкретной модели создаются конкретные объекты заданных классов: Кинотеатр 1, Кинотеатр 2, ..., Кинотеатр N. Разработчик модели свободен в выборе степени необходимой детализации. Он может детализировать предприятие с точностью до конкретных помещений или считать все предприятие одной локацией. Степень такой детализации должна учитываться при расчете вероятности заражения. Важно, что унифицированное программное обеспечение не нужно модифицировать при изменении конкретных классов локаций. Унифицированное программное обеспечение содержит только определение

мета-классов, а конкретные классы создаются из настроек. Для создания этих настроек не требуется навыков программирования, а требуется глубокое понимание структуры современных городских агломераций и целей моделирования. Поэтому описание классов – это уже задача аналитика предметной области. Так, если цель моделирования – поиск решений по минимизации числа контактов в процессе перемещений по предприятию, то необходимо детализировать конкретные помещения. Но если решается задача борьбы с пандемией на уровне муниципалитета – то такая степень детализации может быть избыточной. Таким образом, система масштабируется на объекты моделирования разного размера: конкретные здания, города, области, страны, континенты или вся планета.

Создание конкретных локаций из заданных типов может быть организовано двумя способами:

- путем автоматической случайной генерации на заданной системе координат с заданным законом распределения;
- путем импорта реальных данных из геоинформационной системы с привязкой к типам локаций.

Следующим важным мета-классом являются типы людей - `people_types`. Разделение людей на типы позволяет задать оценки вероятностных характеристик для поведения людей данного типа: как часто они посещают локации определённых типов, как много времени там проводят, в какие дни недели и часы они там обычно бывают. Под словом "обычно" понимается определение параметров и вида закона распределения соответствующей характеристики при выполнении моделирования. Эти законы распределения задаются в другом мета-классе – связь типов людей и типов локаций (`link_loc_people_types`).

Сам по себе человек – это экземпляр класса определенного типа людей. Где типы людей порождаются мета-классом `people_types`. Человек обладает следующими атрибутами: пол, возраст, состояние человека (в соответствии с SEIR моделью), текущая локация, детализации местоположения в рамках локации с текущей реализации отсутствует, хотя возможна в дальнейшем. Атрибуты человека могут влиять на функции распределения характеристик поведения человека, описанных в классах, принадлежащих метаклассу `link_loc_people_types`, делая их многомерными.

Для отражения финансовой составляющей взаимодействий в `link_loc_people_types` указываются возможные движения денежных средств, к которым приводит нахождение в соответствующей локации. Для этого в системе есть мета-классы статей доходов и расходов, которые приводят к счетам расходов и доходов, которые можно вести по типам людей и конкретным людям. Движение по счету возможно как в плюс, так и в минус. Периодичность начислений задается в атрибуте `income.rate` экземпляра `link_loc_people_types`. Возможные типы начислений: `per fact` – за перемещение в

локацию, *per day* – раз в день если локация посещалась, *per minute* – поминутно за нахождение в локации, *per hour* – за каждый час нахождения в локации.

С точки зрения унификации транспортные средства – это особый тип локации, который имеет не постоянное географическое положение, а способен перемещаться, и вместе с ним перемещаются все люди, если они пребывают в данной локации. Тип транспорта (*transport_types*) – это мета-класс, унаследованный от мета-класса (*location_types*). Он может приводить к уменьшению статей дохода людей и увеличению статей дохода муниципалитета, если таковые заданы. Публичный транспорт привязан к определенным остановкам. Для этого используется мета-класс *trans_station_link*. Для создания остановок используется тот же мета-класс *location_types*, что и для других локаций. Остановки могут быть разбиты на отдельные типы локаций по типам транспорта, а могут не разделяться и пользоваться одним классом. Определённая последовательность остановок задает маршрут транспорта. Генерация маршрутов, как и генерация локаций, может быть организована двумя способами: по датчикам псевдослучайных чисел или загрузкой из реальной транспортной сети.

Моделирование выполняется с фиксированным шагом по времени, так как предполагается проведение прогонов модели на существенном количестве людей, и переменный шаг по времени не даст существенного эффекта.

Программное обеспечение каркаса было реализовано на языке C#. Оно может функционировать на всех современных операционных системах Linux Ubuntu 20.04 и аналогах, Windows 7/8/10, MacOS 10.15 и выше.

Программа принимает в качестве входных параметров json-файл, в котором должны быть описаны все требуемые мета-классы.

Работа программы представляет собой повторяющийся цикл, в ходе которого происходит перемещение людей между локациями и передача инфекции с некоторой вероятностью, если они находятся в одной локации.

Перед запуском симуляции создаётся таблица маршрутов, в которой содержится информация о кратчайшем пути между любыми двумя локациями. При составлении таблицы маршрутов приоритет будет отдаваться перемещению между станциями (т.е. с помощью транспорта). Программа поддерживает работу в нескольких потоках, что существенно сокращает время вычислений.

При планировании нахождения в локациях с учетом случайных законов распределения времени посещения могут возникать конфликты, когда человек должен покинуть локацию, где не истекло его время нахождения. Такие ситуации могут создаваться осознанно, например, если человеку нужно посетить столовую в обеденный перерыв. В стандартном агентном моделировании такие ситуации решаются заранее путем описания алгоритма активностей человека и человек перемещается между локациями следуя заданному алгоритму. В предлагаемой системе алгоритм реализован неявно и переходы возможны не всегда. Как и происходит в реальной жизни, нам не

всегда удастся попасть на обед. Для реализации такого поведения предусмотрен механизм стека. Он состоит из локаций, посещение которых подразумевается в текущий момент времени, но фактически человек находится только в одной локации, расположенной в вершине стека. Вычисления включают следующие этапы:

1. Обработка класса «люди»:

- удаление из стека локаций, которые человек должен посетить за день, если время симуляции больше, чем определённые временные границы нахождения в конкретной локации;

- если был выполнен переход к новому дню, то генерируется новый список локаций, которые человек должен посетить за день;

- если время симуляции попадает во временные границы запланированного посещения локации, то локация добавляется в стек;

- если человек в данный момент не находится в локации, находящейся на вершине стека, то начинается перемещение в эту локацию;

- если человек заражён и пришло время перехода в следующую стадию болезни, то этот переход осуществляется.

2. Обработка класса «локации»:

- если в локации появился новый человек, выполняется попытка передачи вируса между прибывшими людьми и теми, кто уже был в локации.

3. Сохранение результатов:

- запись изменений состояний;

- запись критериев оценки результатов моделирования;

- запись среднего за определённый промежуток времени кол-ва людей в локациях.

Перемещение в пространстве можно рассматривать как перемещение между локациями, причём перемещение между любыми локациями осуществляется либо напрямую, либо через станции.

Для организации параллельной работы множества людей и локаций делятся на группы по количеству заданных в настройках потоков, вычисления с группами выполняются параллельно.

Результат моделирования

Для моделирования потока людей в населенных пунктах будем использовать данные OpenStreetMap (OSM). Данные в OSM заносятся через специальные программы-редакторы, как правило, это Overpass API и Overpass Turbo.

На рисунке 1 показан фрагмент работы с приложением Overpass Turbo. Выбирается населенный пункт, с помощью запроса определяются объекты на карте, которые будут использоваться при моделировании: это могут быть жилые дома, учреждения, остановки общественного транспорта и т.д. Далее данные экспортируются в формат osm, рисунок 2.

Доступ к картам реализован через веб-сервис <https://overpass-turbo.eu/>, который находится в открытом доступе [12]. С помощью этого веб-сервиса можем к нашей программе подключать различные населенные пункты.

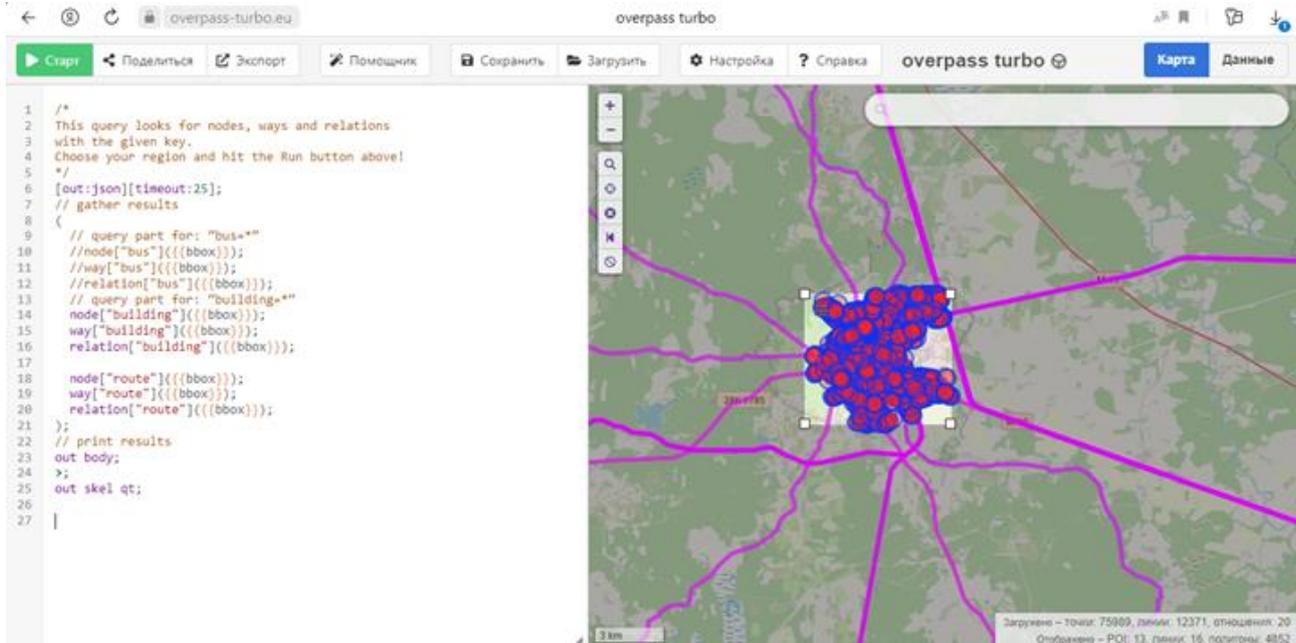


Рис. 1. Приложение Overpass Turbo

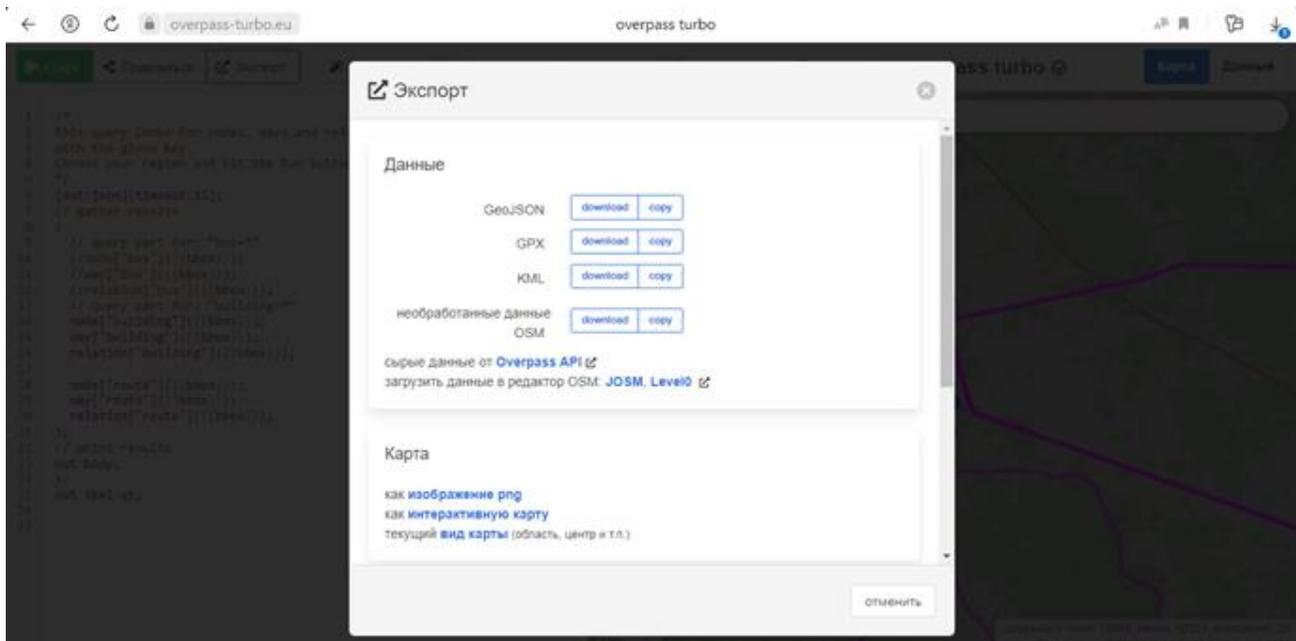


Рис.2. Экспорт данных в формат osm

На рисунке 3 показано подключение карты в формате osm к программе через файл UPDESUA.json.

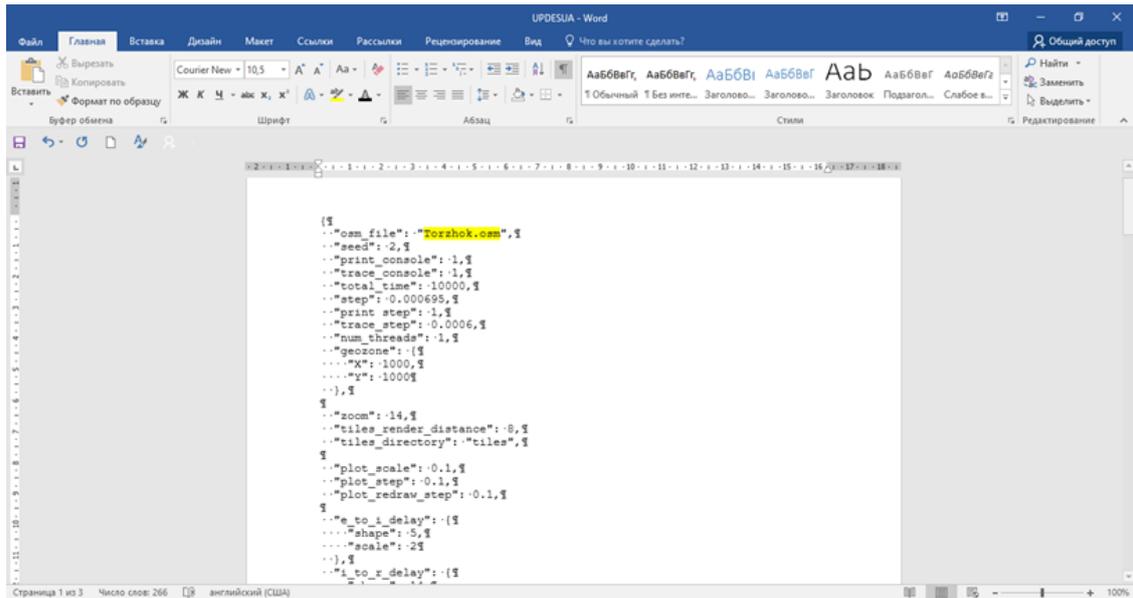


Рис. 3. Фрагмент использования в программе данных карты в формате osm

Программа размещена в открытом доступе на сайте <https://github.com/dark1103/CitySimulation>.

Результаты работы сохраняются в форме таблицы и выводятся в виде графиков. Примеры графиков показаны на рисунках 4, 5.

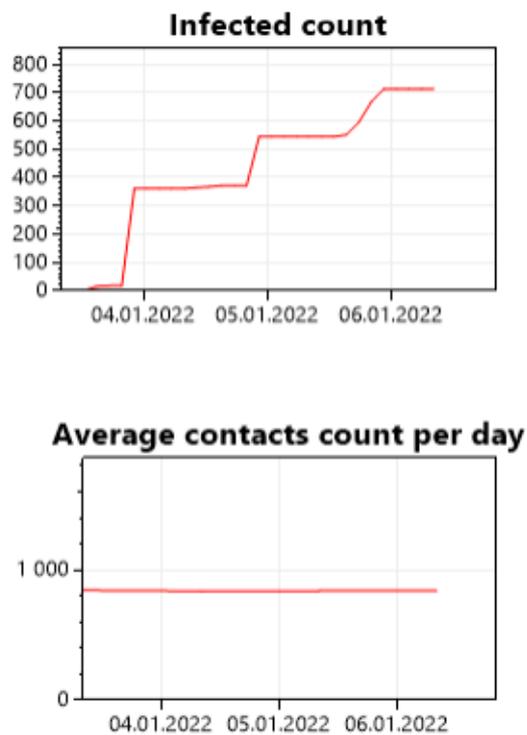


Рис. 4. Графическое представление результатов моделирования

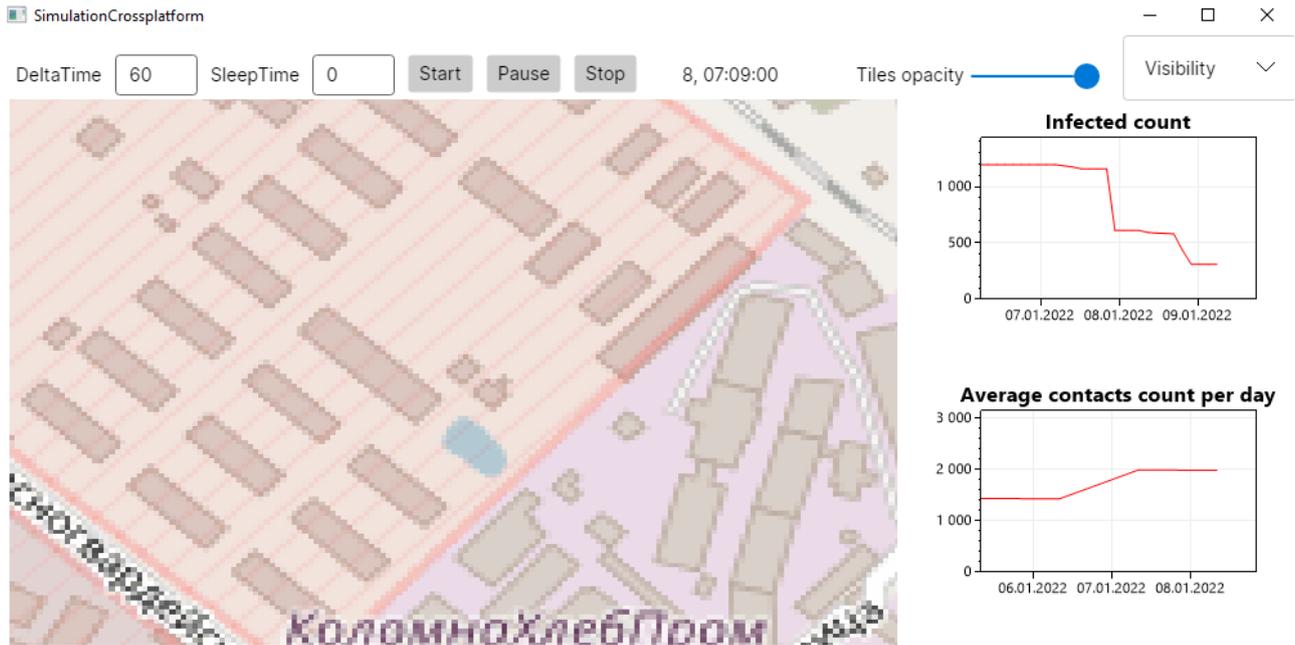


Рис. 5. Фрагмент результатов программы

Заключение

Унификация программного обеспечения моделирования позволяет существенно повысить скорость создания модели.

Предложенная унифицированная модель позволяет создавать сложные модели территориальных образований произвольного масштаба на основе реальных данных из геоинформационных систем, таких как OpenStreetMap (OSM).

Данный программный комплекс позволяет проводить моделирование несколько потоков, к тому же использование мета-классов дает возможность использовать большое количество входных данных. Это позволяет приблизить результат моделирования к реальной ситуации и использовать полученные данные для оперативной поддержки принятия решений по борьбе с пандемиями. Данный комплекс был апробирован на ряде населенных пунктов.

Библиографический список

1. A modified SEIR model to predict the behavior of the early stage in coronavirus and coronavirus-like outbreaks / Angulo W. [et al.] // Sci. Rep. 2021. 11. 16331. <https://doi.org/10.1038/s41598-021-95785-y>
2. GPSS 50 years old, but still young / Stahl I. [et al.] // Proceedings of the 2011 Winter Simulation Conference (WSC). 2011. 3947-3957.
3. Pufahl, L., Wong, T.Y, Weske M. Design of an Extensible BPMN Process Simulator. // Lecture Notes in in Business Information Processing. 2018. vol. 308. 782-795. https://doi.org/10.1007/978-3-319-74030-0_62

4. Schimit P.H.T A model based on cellular automata to estimate the social isolation impact on COVID-19 spreading in Brazil // Computer Methods and Programs in Biomedicine. 2021. 200. 105832. <https://doi.org/10.1016/j.cmpb.2020.105832>
5. Mosler K. Ernesto Estrada and Philip A. Knight (2015) // A First Course in Network Theory. Oxford University Press. 2017. Stat papers 58. 1283–1284. <https://doi.org/10.1007/s00362-017-0961-1>
6. Мельничук А.В., Сивакова Т.В., Судаков В.А. Решение задач оптимизации с использованием мультиагентных моделей // Препринты ИПМ им. М.В.Келдыша. 2019. № 100. 16 с. <https://doi.org/10.20948/prepr-2019-100>
7. Sudakov V., Yashin N. Implementing the Graph Model of the Spread of a Pandemic on GPUS // Korenkov V., Nechaevskiy A., Zaikina T. (eds) 9th International Conference Distributed Computing and Grid Technologies in Science and Education, CEUR Workshop Proceedings. 2021. 3041. 488 – 493. <https://doi.org/10.54546/mlit.2021.53.71.001>
8. Covasim: An agent-based model of COVID-19 dynamics and interventions. / Kerr C.C.[et al.] // PLOS Comput. Biol. 2021. 17(7), e1009149. <https://doi.org/10.1371/journal.pcbi.1009149>
9. Peng B., Pettit R., Amos C. Population simulations of COVID-19 outbreaks provide tools for risk assessment and continuity planning. 2021. JAMIA Open 4(3). oaaa074. <https://doi.org/10.1093/jamiaopen/oa074>
10. COVID-19 and Simulation – AnyLogic. URL: <https://www.anylogic.com/blog/covid-19-and-simulation>. (дата обращения: 01.11.2022).
11. Population-based Forward-time Simulator for the Outbreak of COVID-19. URL: <https://github.com/ictr/covid19-outbreak-simulator> (дата обращения: 01.11.2022).
12. Overpass-turbo. URL: <https://overpass-turbo.eu/> (дата обращения: 01.11.2022).

Оглавление

Введение	3
Обзор моделей	3
Постановка задачи.....	5
Результат моделирования	8
Заключение.....	11
Библиографический список.....	11