

DOI 10.21672/2074-1707.2021.53.1.018-027
УДК 004.4:004.62

РАЗРАБОТКА МОДУЛЯ УПРАВЛЕНИЯ ДАННЫМИ ОБ ОБЪЕКТАХ НА ОНЛАЙН-КАРТЕ ГОРОДА*

Статья поступила в редакцию 11.01.2021, в окончательном варианте – 29.01.2021.

Бурова Алена Алексеевна, Волгоградский государственный технический университет, 400005, Российская Федерация, г. Волгоград, пр. Ленина, 28, магистрант, e-mail: ata343@mail.ru

Буров Сергей Сергеевич, Волгоградский государственный технический университет, 400005, Российская Федерация, г. Волгоград, пр. Ленина, 28, магистрант, e-mail: sergey.burovic@gmail.com

Парыгин Данила Сергеевич, Волгоградский государственный технический университет, 400005, Российская Федерация, г. Волгоград, пр. Ленина, 28, кандидат технических наук, доцент, ORCID 0000-0001-8834-5748, e-mail: dparygin@gmail.com

Финогеев Антон Алексеевич, Пензенский государственный университет, 440026, Российская Федерация, г. Пенза, ул. Красная, 40,

кандидат технических наук, доцент, e-mail: fanton3@ya.ru

Рент Вячеслав Эдуардович, Волгоградский государственный технический университет, 400005, Российская Федерация, г. Волгоград, пр. Ленина, 28, магистрант, e-mail: vyacheslav61g@gmail.com

Решение широкого спектра задач, направленных на обеспечение устойчивого функционирования городской инфраструктуры, требует всестороннего прогнозирования развития ситуации. Работа распределенных технических систем в условиях урбанизированной территории непосредственно связана с множественными субъектно-объектными взаимодействиями, так как компоненты систем чаще всего расположены вблизи от практически непрерывной активности людей, имеющих отношение к их обслуживанию или занятых своей повседневной деятельностью. Поэтому построение точных моделей городских процессов зависит от обширной информационной базы об объектах инфраструктуры территории. В первую очередь это касается агрегирования и унификации данных разрозненных источников, а также последующей верификации, дополнения и обновления исходных данных. В статье описываются этапы разработки программного модуля управления данными, производящего парсинг данных из файла с форматом OSM XML, преобразующего их к необходимой структуре для работы в рамках единой платформы пространственного моделирования Live.UrbanBasis.com и производящего процедурную генерацию недостающих данных. Подход к генерации дополнительных данных по объектам инфраструктуры продемонстрирован на примере подъездов, этажей и квартир. В статье раскрываются подходы и технологии, примененные для работы с данными веб-картографического проекта OpenStreetMap.

Ключевые слова: OpenStreetMap, NetTopologySuite, объект инфраструктуры, пространственные данные, процедурная генерация, преобразование данных, онлайн-карта, город

DEVELOPMENT OF THE MODULE FOR DATA MANAGEMENT ON OBJECTS FOR ONLINE MAP OF A CITY

The article was received by the editorial board on 11.01.2021, in the final version – 29.01.2021.

Burova Alena A., Volgograd State Technical University, 28 Lenin Ave., Volgograd, 400005, Russian Federation,

post-graduate student, e-mail: ata343@mail.ru

Burov Sergey S., Volgograd State Technical University, 28 Lenin Ave., Volgograd, 400005, Russian Federation,

post-graduate student, e-mail: sergey.burovic@gmail.com

Parygin Danila S., Volgograd State Technical University, 28 Lenin Ave., Volgograd, 400005, Russian Federation,

Cand. Sci. (Engineering), Associate Professor, ORCID 0000-0001-8834-5748, e-mail: dparygin@gmail.com

Finogeev Anton A., Penza State University, 40 Krasnaya St., Penza, 440026, Russian Federation,

Cand. Sci. (Engineering), Associate Professor, e-mail: fanton3@ya.ru

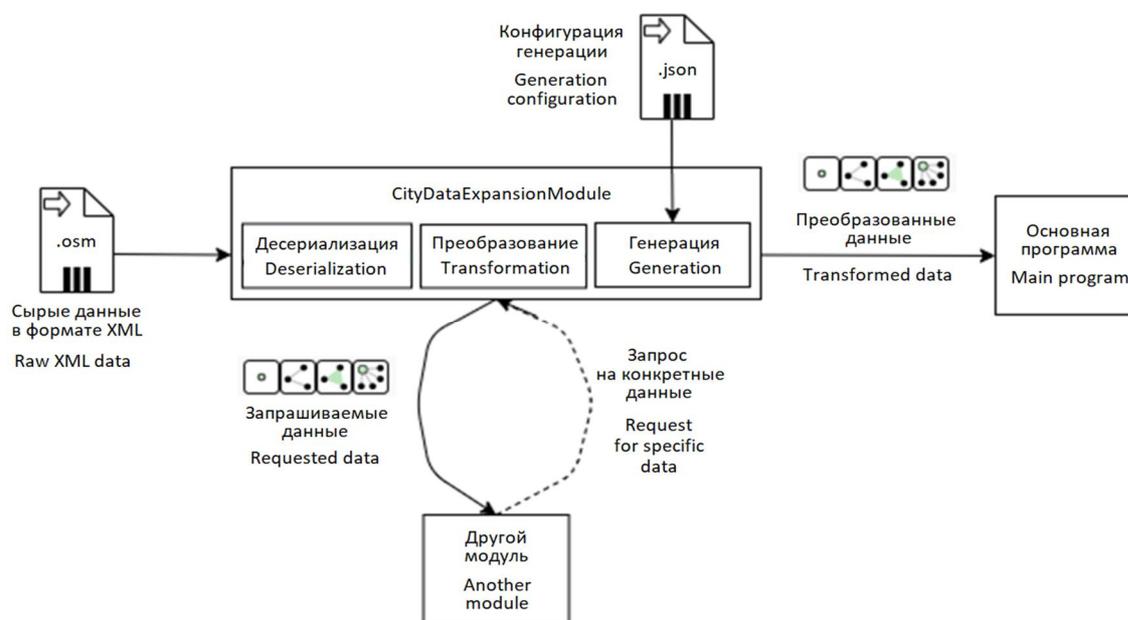
* Исследование выполнено в рамках гранта Российского научного фонда (РНФ, проект № 20-71-10087). Авторы выражают благодарность коллегам по лаборатории UCLab, особенно Русиновой Алле Викторовне за участие в разработке проекта Live.UrbanBasis.com.

Rent Vyacheslav E., Volgograd State Technical University, 28 Lenin Ave., Volgograd, 400005, Russian Federation,
post-graduate student, e-mail: vyacheslav61g@gmail.com

The solution of a wide range of tasks aimed at ensuring the sustainable functioning of urban infrastructure requires a comprehensive forecast of the development of the situation. The operation of distributed technical systems in an urbanized area is directly related to multiple subject-object interactions, since system components are most often located close to the almost continuous activity of people related to their maintenance, or just engaged in their daily routine. Therefore, the construction of urban processes models requires an extensive information base about the area infrastructure objects. First of all, this is associated with the aggregation and unification of data from disparate sources, and further verification, addition and updating of the original data is required. The paper describes the stages of development of a data management software module that parses data from a file with OSM XML format, transforms them into the necessary structure for working within a single spatial modeling platform Live.UrbanBasis.com and makes procedural generation of missing data. The approach to additional data generation of infrastructure objects is demonstrated using the example of entrances, floors, and apartments. The paper reveals the approaches and technologies used to work with the data of the OpenStreetMap web mapping project.

Keywords: OpenStreetMap, NetTopologySuite, infrastructure object, spatial data, procedural generation, data conversion, simulation, online map, city

Graphical annotation (Графическая аннотация)



Введение. Решение производственных задач различными учреждениями и организациями, обеспечивающими функционирование городской инфраструктуры, связано с учётом состояния и характеристик большого количества объектов территории [1]. Еще большую значимость имеет возможность оценивать их количественные и качественные изменения в результате их эксплуатации городским населением или обслуживания коммунальными службами. Для реализации такой предиктивной аналитики в лаборатории городских вычислений UCLab была создана клиент-серверная платформа OsmLifeSimulation (OSMLS) для моделирования перемещений и взаимодействий в рамках участка карты города [8]. Изначально программная система разрабатывалась с учётом потребностей служб управления транспортом, экологического мониторинга и управления отходами, однако в дальнейшем была расширена для учёта широкого спектра запросов потенциальных пользователей. Платформа получила возможность расширения функционала моделирования за счет подключаемых к ней модулей.

Связано это решение было также и с тем, что одной из проблем взаимодействия городских служб является множественность систем учёта информации, а также геоинформационных подоснов, используемых для работы с пространственными данными. Для нивелирования этих противоречий и снижения входного порога при создании собственных модулей было решено использовать в качестве картографической основы открытые данные проекта OpenStreetMap. Актуальные гео-данные этого проекта поставляются в файле формата OSM XML.

В итоговой модели городских процессов стили отображаемых на карте объектов могут быть изменены отдельными модулями платформы [4]. Так как каждый модуль должен иметь возможность производить дополнительные действия с данными об объектах, то на этом этапе возникала необходимость структурирования данных для всех модулей системы. Исходный файл нуждался в парсинге и приведении полученных данных к новой структуре, совместимой с основной программой. При наличии в общей системе всего лишь нескольких отраслевых модулей, каждый из них мог решать эту задачу самостоятельно. Но при масштабировании платформы такая организация работы с данными становилась менее эффективной. В связи с этим целью представленного в данной статье исследования была выработка подхода к управлению данными об объектах городской инфраструктуры от их получения в источнике до генерации недостающей информации.

1. Анализ существующих подходов к работе с открытой базой картографических данных OpenStreetMap. OpenStreetMap (OSM) является некоммерческим веб-картографическим сервисом, поддерживающим множество данных об объектах инфраструктуры по всему миру. Для создания карт используются данные с персональных GPS-трекеров, аэрофотографии, видеозаписи, спутниковые снимки и панорамы улиц, предоставленные некоторыми компаниями, а также знания человека, рисующего карту [10].

Популярные сервисы с миллионами пользователей по всему миру используют данные OpenStreetMap в своих разработках. «Tesla Smart Summon», получившая широкое распространение в США, использует данные OSM для автономной навигации транспортных средств на парковках без водителя [11]. «Geotab» использует данные OpenStreetMap в своей платформе программного обеспечения для отслеживания транспортных средств MyGeotab [9]. Множество аналитических проектов и задач в сфере управления инфраструктурой реализуется основываясь на данных OSM [3].

Базовым элементом структуры данных OSM является точка (node) с географическими координатами – широтой (latitude) и долготой (longitude). Точка может быть отдельным объектом или входить в состав других объектов (линий, полигонов, мультиполигонов).

Линия (way) – это последовательности точек. Менять последовательность нельзя. Несколько линий логически могут представлять один объект. Например, длинная дорога состоит из нескольких линий.

Полигон – это замкнутая линия, у которой совпадают первая и последняя точки. Полигон не является самостоятельным элементом OSM.

Отношение – это логическое объединение точек, линий и других отношений в единый объект.

Каждый объект имеет набор тегов, описывающих его. Тег (tag) определен как $k = \langle \text{ключ} \rangle v = \langle \text{значение} \rangle$. Если элемент не имеет тегов, то он не является объектом, а входит в состав других объектов, как и некоторые элементы с тегами.

В целом, всю структуру данных OSM можно представить схематично (рис. 1).

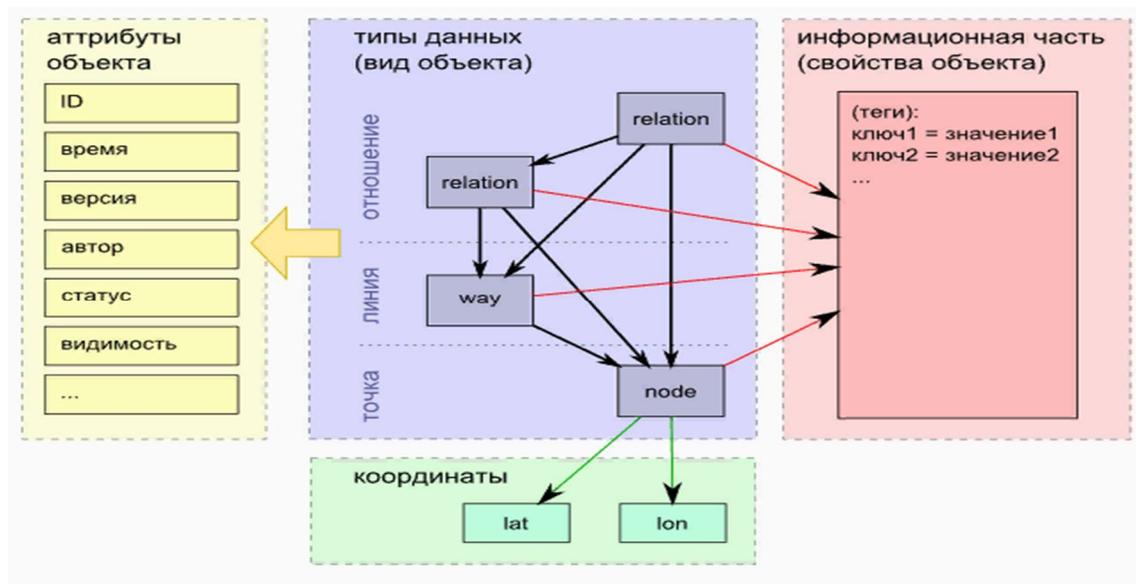


Рисунок 1 – Структура данных OpenStreetMap [5]

Все данные можно разбить на три группы:

1) типы данных (вид объекта) – основная группа, описывающая геометрические типы объектов на карте и их иерархию по отношению друг к другу, где основная единица точка (node) обязательно включает в себя координаты (lat, lon) и может входить во все остальные типы объектов;

2) атрибуты объектов – описывает формальные данные об авторах, времени добавления/правки, статусе, видимости, организует структуру типов объектов за счет присваивания к ним ID;

3) информационная часть (свойства объектов) – описательная группа, формирующая представление об объекте и его свойствах путем добавления определенных тегов к объекту.

Существует ряд решений для парсинга данных из osm-файла. Например, библиотека «BasicOSMParser» (доступна на GitHub в репозитории аккаунта PanierAvide/BasicOSMParser) – это набор классов Java, позволяющий анализировать необработанные XML-файлы OSM. Также библиотека позволяет экспортировать данные в виде файлов CSV. Он использует Java по умолчанию SAX parser (org.xml.sax). Тесты приложения используют платформу JUnit 4 framework.

Библиотека «osm4routing» (доступна на GitHub в репозитории аккаунта Tristramg/osm4routing) – инструмент, предоставляющий парсер данных из OpenStreetMap, чтобы превратить их в узлы-ребра, приспособлено для применений трассы. Входными данными является XML-файл OpenStreetMap. Выходными данными библиотеки могут быть CSV-файл или база данных (postgres, mysql, sqlite, postgres).

Библиотека «Libosmium» (доступна на GitHub в репозитории аккаунта osmcode/libosmium) – быстрая и гибкая библиотека C++ для работы с данными OpenStreetMap. Библиотека может прочитать OSM XML данные или в двоичном формате (PBF) и может вызывать различные обработчики для каждого OSM-объекта. Libosmium доступен под лицензией на программное обеспечение Boost.

Тем не менее существующие решения оказались не применимы к разрабатываемой платформе, так как соответствующие ее модули после парсинга из файла «.osm» принимают данные определенной структуры точек, линий и отношений, наследующихся от классов библиотеки NetTopologySuite (доступна на GitHub в репозитории аккаунта NetTopologySuite/NetTopologySuite).

2. Подход к подготовке данных для формирования среды моделирования городских процессов. Информационным обеспечением основной программы и других модулей является файл формата OSM XML (экспортированные данные участка карты OpenStreetMap). В основном файле содержится список экземпляров примитивных данных, таких как узлы, пути и отношения («nodes», «ways» и «relations»), а также их описание и связи.

Для корректной работы с коллекцией основной программы необходимо все полученные данные из файла приводить к структуре (точек, линий и отношений) при помощи библиотеки NetTopologySuite.

Коллекция основной программы MapObjects передается с сервера на клиент в формате GeoJSON. Для преобразования данных из объектов типа Geometry в GeoJSON используется GeoJsonWriter – средство библиотеки NetTopologySuite.

Соотношение структур полученных объектов из файла OSM XML и приведенных к структуре Geometry при помощи библиотеки NetTopologySuite представлено в таблице.

Таблица 1 – Соотношение структур объектов

Объект OSM XML	Наследуемый объект NetTopologySuite.Geometries
Точка (node)	Point
Линия (way)	LineString
Замкнутая линия (closed way)	LinearRing
Отношение (relation)	GeometryCollection

Для каждого из типов данных необходимо хранить его теги, описывающие объект карты и его свойства. Для типа «relation» помимо тегов необходимо хранить ещё и роль каждого из объектов.

Структура данных, которая должна быть получена после преобразования и использоваться в дальнейшем другими модулями и базовыми компонентами платформы, представлена на рисунке 2.

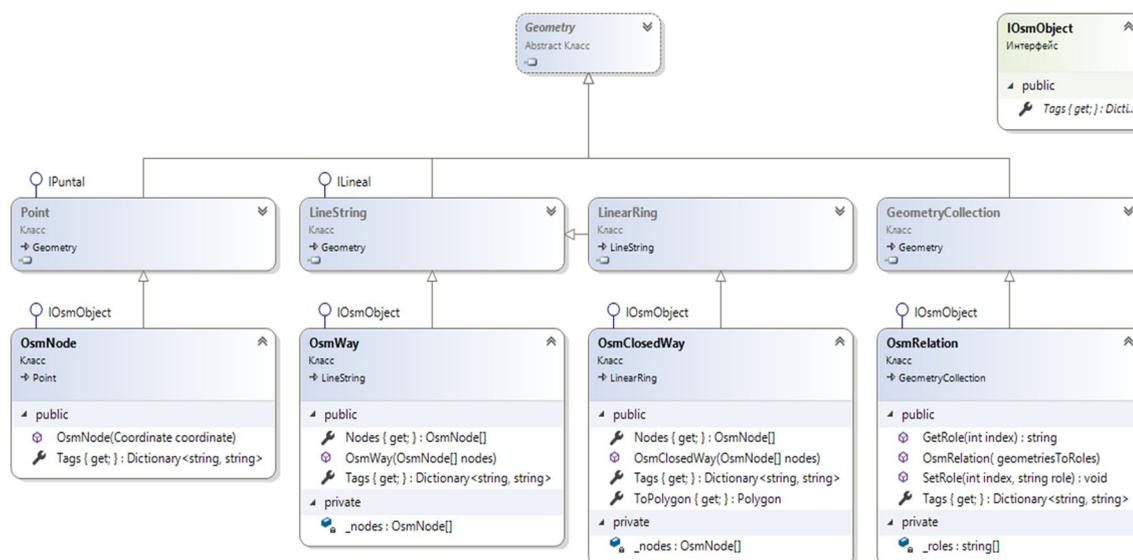


Рисунок 2 – Диаграмма классов OSM

В связи с этим программный модуль управления данными об объектах на онлайн-карте города должен соответствовать следующим требованиям: наследоваться от класса базового модуля; как и остальные модули, должен наследоваться от класса OSMLSModule и инициализироваться основной программой; производить парсинг данных из файла с форматом OSM XML, используя стандартные инструменты десериализации платформы DOT NET; преобразовывать полученные данные к новой структуре точек, линий и отношений при помощи библиотеки NetTopologySuite.

Кроме того, необходимо осуществлять процедурную генерацию недостающих данных (например, подъездов, этажей, квартир). Этих данных об объектах в исходном файле может быть недостаточно для дальнейшего моделирования процессов взаимодействия с объектами на карте города [2]. Генерация данных должна быть основана на данных конфигурации генерации, находящихся в файле «.json».

Помещать преобразованные и дополненные данные необходимо в коллекцию основной программы OSMLS. Основная программа хранит эти данные у себя в оперативной памяти и предоставляет их другим модулям. Выборка преобразованных данных (все замкнутые линии зданий, границы карты, все незамкнутые линии рельсовых путей и велодорожек) должна быть предоставлена по запросам другим модулям.

В общем виде всю работу данного модуля можно описать схемой, представленной в графической аннотации.

3. Реализация модуля управления данными об объектах инфраструктуры. Для реализации программного модуля в качестве интегрированной среды разработки программы была использована многофункциональная интегрированная среда разработки Microsoft Visual Studio 2017 community edition, позволяющая разрабатывать проекты для научных исследований или проекты с открытым исходным кодом.

Исходные коды программного модуля были реализованы на языке C# 7.3 с использованием модульной платформы .NET Core 2.1 [12]. Версия .NET Core 2.1. была обязательным условием для разработки модулей, в связи с требованием платформы.

Все объекты в программе наследуются от геометрических примитивов библиотеки NetTopologySuite версии 2.0.0.

3.1. Преобразование данных исходной базы. Класс OsmObjectsConverter реализует преобразование данных типа OsmXml, считанных из файла к коллекции osmObjects. Класс включает в себя функцию ConvertObjectsOSM(), которая, в первую очередь, находит и преобразует координаты границы используемой карты из классических Lat/Lon-координат в сферическую проекцию меркатора. Создает объект типа LinearRing с этими координатами.

Дальше идет основная часть работы класса, где поэтапно создаются словари OsmNode, OsmWays, OsmClosedWay и OsmRelations, где ключ – Id объекта, значение – преобразованный объект.

Словарь со значением типа OsmNode включает в себя преобразованные координаты (границы карты преобразуются, как и координаты) и теги для определенной точки.

Словари со значением типа `OsmWays` и `OsmClosedWay` отличаются лишь тем, что первая и последняя точка в `OsmClosedWay` должны быть одинаковы, когда в `OsmWays` они разные и не образуют замкнутую линию. В остальном `OsmWays` и `OsmClosedWay` образуются по одному принципу: из имеющегося словаря точек `OsmNode` берутся те, чьи `Id` (ключи словаря) совпадают со списком `Id` точек в `OsmXml.Way` и добавляются в словарь.

Словари со значением типа `OsmRelations` могут содержать в себе любые объекты: точки, линии и другие отношения. Следовательно, в отношениях указывается не только `Id` объекта, но и его тип (`member.Type`). `Id` текущего `OsmRelations` записывается в ключ словаря. По типу и `Id` объекта находится сам объект и добавляется в значение словаря.

После завершения преобразования все словари сливаются в один `mapNodes`, который возвращает данная функция. Отдельно возвращаются преобразованные данные о границах карты в виде полигона (`Polygon`).

3.2. Генерация дополнительных данных об инфраструктуре города. Также было выявлено, что для решения некоторых задач моделирования, связанных с объектами инфраструктуры, данных исходного файла может быть недостаточно. Для устранения этой проблемы необходима дополнительная генерация различных типов объектов, таких как подъезды, квартиры, строения и т.п. на основе существующих данных карты.

Для генерации недостающих данных квартир, этажей и подъездов используется класс `OsmObjectsGenerator`. В него подаются преобразованные данные `osmObjects` и данные конфигурации.

Данные конфигурации содержатся в объекте класса `GeneratorConfiguration` в следующих полях:

- 1) `BuildingsTypesToLevelsCount` – словарь, в котором ключ = количество этажей здания, значение = лист типов зданий, которые имеют данное количество этажей;
- 2) `DefaultLevelsCount` – дефолтное значение количества этажей, присваивается в случае, если тип здания не совпадает со значением из листа;
- 3) `IsLevelsGenerationEnabled` – отключает/включает генерацию этажей в зданиях;
- 4) `BuildingsTypesToEntrancesCount` – словарь, в котором ключ = количество подъездов в здании, значение = лист типов зданий, которые имеют данное количество подъездов;
- 5) `DefaultEntrancesCount` – дефолтное значение количества подъездов, присваивается в случае, если тип здания не совпадает со значением из листа;
- 6) `IsEntrancesGenerationEnabled` – отключает/включает генерацию подъездов в зданиях;
- 7) `BuildingsTypesToFlatsCount` – словарь, в котором ключ = количество квартир в здании, значение = лист типов зданий, которые имеют данное количество квартир;
- 8) `DefaultFlatsCountForEveryEntranceLevels` – дефолтное значение количества квартир в здании, присваивается в случае, если тип здания не совпадает со значением из листа;
- 9) `IsFlatsGenerationEnabled` – отключает/включает генерацию квартир в зданиях.

Данные конфигурации берутся из файла `generatorConfig.json` [6], если он существует. Если программа не может найти файл, она создает новый файл с дефолтными значениями для генерации. Конфигурация может задаваться или обновляться самим пользователем через файл.

При генерации этажей учитывается тип зданий. Если тип здания или данные конфигурации для такого типа здания не указаны, то берется значение этажности соседнего здания. В случае если в соседних зданиях этажность не указана или здания не были найдены, то берется дефолтное значение количества этажей.

Генерация подъездов исходит не из площади, а из длины фасада здания. Берется самый длинный из фасадов для дальнейшей генерации. Подъезды генерируются на одинаковом друг от друга расстоянии. При генерации подъездов зданий учитывается тип зданий. Если тип здания или данные конфигурации для такого типа здания не указаны, то берется дефолтное значение количества подъездов.

Для вычисления расположения точек-подъездов на линии фасада здания использовалась формула деления отрезка в данном отношении на плоскости $xM = (xA + \lambda \cdot xB) / (1 + \lambda)$ и $yM = (yA + \lambda \cdot yB) / (1 + \lambda)$, где $A(xA; yA)$ и $B(xB; yB)$ являются точками плоскости, а точка $M(xM; yM)$ делит отрезок AB в отношении $\lambda = AM / MB$.

Генерация количества квартир в здании происходит в случае, если для здания указано количество подъездов и этажей. Генерация квартир в доме происходит по формуле «количество квартир на этаже * количество подъездов * количество этажей», где значение количества квартир на этаже берется по типу здания из конфигурации. Если тип здания или данные конфигурации для такого типа здания не указаны, то берется дефолтное значение количества квартир на этаже.

Наглядно результат работы генерации подъездов можно увидеть на рисунке 3, где точками обозначаются подъезды, а здания выделяются рамками. На рисунке 3 слева видно, что в исходном файле до генерации подъездов очень мало и они охватывают не все дома на карте. После генерации (рис. 3, справа) многие здания уже имеют подъезды, расположенные вдоль фасадов.



Рисунок 3 – Отображение подъездов на карте до генерации (слева) и после генерации (справа)

3.3. Тестирование работы модуля с данным для онлайн-карты. Для интеграционного тестирования [7] работы программного модуля был создан отдельный модуль `CityDataExpansionTestModule`, который должен со стороны другого модуля показать работу с предоставляемыми данными модуля `CityDataExpansionModule`. Данный модуль обращается к коллекции основной программы и глобальным переменным (`AllBuildings`, `AllRailways`, `AllCycleways`, `MapBorders`) модуля `CityDataExpansionModule` для дальнейшего отображения этих данных на карте с помощью инструментов основной программы.

Для отображения магазинов на онлайн-карте города были взяты объекты из коллекции основной программы `OSMLS` и из них выбраны объекты с тегом «shop». Для изменения отображения точек магазинов был создан класс «Shop», в котором был прописан стиль точки (красная непрозрачная точка). Также в стиле можно изменить цвет точки, ее границы, ширину и даже добавить ссылку на изображение, которое будет отображаться вместо точки.

Для отображения границы участка карты, с объектами которой сейчас работает модуль, достаточно обратиться к глобальной переменной `MapBorders` в модуле `CityDataExpansionModule` и просто вывести полигон с помощью инструмента основной программы.

Для вывода зданий тоже достаточно просто обратиться к глобальной переменной `AllBuildings` в модуле `CityDataExpansionModule`. Также эти данные можно взять из коллекции основной программы. Объекты представляют собой замкнутые линии с тегом «building» и могут включать в себя точки-подъезды с тегом «entrance».

На рисунке 4 отображены объекты зданий, магазинов и границы моделируемой области на онлайн-карте города, вывод которых был описан выше.

Кроме того, в процессе исследования качества открытых геоданных была выявлена проблема их неполноты. В целях решения данной проблемы был предложен подход к генерации различных типов объектов инфраструктуры. В этом же направлении предполагается в дальнейшем проводить исследования и совершенствовать информационную базу платформы. Для этого планируется использовать нейросетевые модели обработки данных дистанционного зондирования Земли. Они позволят с необходимой регулярностью обновлять и дополнять информацию по территориям мониторинга и моделирования, а также станут основой для более точной генерации данных в местах с отсутствующими исходными источниками.

Библиографический список

1. Алешкевич А. А. Оценка доступности территории города / А. А. Алешкевич // XXII Региональная конф. молод. ученых Волгоградской области. Волгоград, 21–24 нояб. 2017 г. : тезисы докл. – Волгоград, 2017. – С. 210–211.
2. Архипова А. С. Визуализация поведения акторов в модели города / А. С. Архипова // Конкурс научно-исследовательских работ студентов Волгоградского государственного технического университета. Волгоград, 19–22 мая 2020 г. : тезисы докл. – Волгоград, 2020. – С. 175–176.
3. Зеленский И. С. Интеллектуальная поддержка решений по использованию объектов недвижимости для управления урбанизированными территориями / И. С. Зеленский, Д. С. Парыгин, О. В. Савина, А. А. Финогеев, А. А. Шуклин, А. Ю. Антюфеев // *International Journal of Open Information Technologies*. – 2020. – Т. 8, № 11. – С. 13–29.
4. Анохин А. О. Моделирование поведения агентов для реализации игрового искусственного интеллекта / А. О. Анохин, Н. П. Садовникова, А. В. Катаев, Д. С. Парыгин // *Прикаспийский журнал: управление и высокие технологии*. – 2020. – № 2 (50). – С. 85–99. – DOI: 10.21672/2074-1707.2020.50.2.096-110.
5. Структура данных проекта OpenStreetMap. – 2020. – Режим доступа: <https://habr.com/ru/post/146503/>, свободный. – Заглавие с экрана. – Яз. рус. (дата обращения: 01.05.20).
6. How to serialize and deserialize (marshal and unmarshal) JSON in .NET. – 2020. – Режим доступа: <https://docs.microsoft.com/en-us/dotnet/standard/serialization/system-text-json-how-to#additional-resources>, свободный. – Заглавие с экрана. – Яз. англ. (дата обращения: 21.04.20).
7. Integration tests in ASP.NET Core. – 2020. – Режим доступа: <https://docs.microsoft.com/en-us/aspnet/core/test/integration-tests?view=aspnetcore-3.1>, свободный. – Заглавие с экрана. – Яз. англ. (дата обращения: 23.04.20).
8. Parygin D. Multi-agent Approach to Modeling the Dynamics of Urban Processes (on the Example of Urban Movements) / D. Parygin, A. Usov, S. Burov, N. Sadovnikova, P. Ostroukhov, A. Pyannikova // *Communications in Computer and Information Science*. – 2020. – Vol. 1135. – P. 243–257. – DOI : 10.1007/978-3-030-39296-3_18.
9. MyGeotab. – 2020. – Режим доступа: <https://www.geotab.com/blog/smart-fleet-management-road-speed-information/>, свободный. – Заглавие с экрана. – Яз. англ. (дата обращения: 10.12.20).
10. OpenStreetMap. – 2020. – Режим доступа: <https://www.openstreetmap.org/>, свободный. – Заглавие с экрана. – Яз. англ. (дата обращения: 28.11.20).
11. Tesla Smart Summon. – 2020. – Режим доступа: <https://teslamotorsclub.com/tmc/threads/openstreetmaps-and-smart-summon.170675/>, свободный. – Заглавие с экрана. – Яз. англ. (дата обращения: 15.12.20).
12. What's new in .NET Core. – 2020. – Режим доступа: <https://docs.microsoft.com/en-us/dotnet/core/whats-new/dotnet-core-2-1>, свободный. – Заглавие с экрана. – Яз. англ. (дата обращения: 28.04.20).

References

1. Aleshkevich A. A. Otsenka dostupnosti territorii goroda [Urban area accessibility assessment]. *Tezisy dokladov XXII Regionalnoy konferentsii molodykh uchenykh Volgogradskoy oblasti* [Abstracts of the XXII Regional Conference of Young Scientists of the Volgograd Region], Volgograd, 21–24 November 2017. Volgograd, 2017, pp. 210–211.
2. Arkhipova A. S. Vizualizatsiya povedeniya aktorov v modeli goroda [Visualization of actor's behavior in a city model]. *Tezisy dokladov konkursa nauchno-issledovatel'skikh rabot studentov Volgogradskogo gosudarstvennogo tekhnicheskogo universiteta* [Abstracts of the competition of research works of students of Volgograd State Technical University], Volgograd, 19–22 May 2020. Volgograd, 2020, pp. 175–176.
3. Zelenskiy I. S., Parygin D. S., Savina O. V., Finogeev A. A., Shuklin A. A., Antyufeyev A. Yu. Intellectuálnaya podderzhka resheniy po ispolzovaniyu obektov nedvizhimosti dlya upravleniya urbanizirovannymi territoriyami [Intelligent Support to Real Estate Use Decisions for Urbanized Areas Management]. *International Journal of Open Information Technologies*, 2020, vol. 8, no. 11, pp. 13–29.
4. Anokhin A. O., Sadovnikova N. P., Kataev A. V., Parygin D. S. Modelirovaniye povedeniya agentov dlya realizatsii igrovogo iskusstvennogo intellekta [Modeling of agent's behavior to implement gaming artificial intelligence]. *Prikaspiyskiy zhurnal: upravlenie i vysokie tekhnologii* [Caspian Journal: Control and High Technologies], 2020, no. 2 (50), pp. 85–99. DOI: 10.21672/2074-1707.2020.50.2.096-110.
5. *Struktura dannykh proyekta OpenStreetMap* [Data structure of the OpenStreetMap project]. Available at: <https://habr.com/ru/post/146503/> (accessed 01.05.20).
6. *How to serialize and deserialize (marshal and unmarshal) JSON in .NET*. Available at: <https://docs.microsoft.com/en-us/dotnet/standard/serialization/system-text-json-how-to#additional-resources> (accessed 21.04.20).

7. *Integration tests in ASP.NET Core*. Available at: <https://docs.microsoft.com/en-us/aspnet/core/test/integration-tests?view=aspnetcore-3.1> (accessed 23.04.20).
8. 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*, 2020, vol. 1135, pp. 243–257. DOI: 10.1007/978-3-030-39296-3_18.
9. *MyGeotab*. Available at: <https://www.geotab.com/blog/smart-fleet-management-road-speed-information/> (accessed 10.12.20).
10. *OpenStreetMap*. Available at: <https://www.openstreetmap.org/> (accessed 28.11.20).
11. *Tesla Smart Summon*. Available at: <https://teslamotorsclub.com/tmc/threads/openstreetmaps-and-smart-summon.170675/> (accessed 15.12.20).
12. *What's new in .NET Core*. Available at: <https://docs.microsoft.com/en-us/dotnet/core/whats-new/dotnet-core-2-1> (accessed 28.04.20).

УДК 004.832.2

К ВОПРОСУ О ПОДДЕРЖКЕ ПРОЦЕССА ПРИНЯТИЯ РЕШЕНИЯ ОБ УЛУЧШЕНИИ ДЕЯТЕЛЬНОСТИ В ИСПЫТАТЕЛЬНОЙ ЛАБОРАТОРИИ

Статья поступила в редакцию 29.10.2020, в окончательном варианте – 12.11.2020.

Аль-Бусаиди Саид Султан Саид, Тамбовский государственный технический университет, 392000, Российская Федерация, г. Тамбов, ул. Советская, 106, аспирант, ORCID: www.orcid.org/0000-0003-3990-8123, e-mail: al-busaidi2020@hotmail.com

Воякина Юлия Николаевна, Тамбовский государственный технический университет, 392000, Российская Федерация, г. Тамбов, ул. Советская, 106, магистрант, ORCID: www.orcid.org/0000-0003-1654-4232, e-mail: miss.voyakina2011@yandex.ru

Пономарев Сергей Васильевич, Тамбовский государственный технический университет, 392000, Российская Федерация, г. Тамбов, ул. Советская, 106, доктор технических наук, профессор, ORCID: www.orcid.org/0000-0003-0228-912X, e-mail: svponom@yahoo.com

Рассмотрены вопросы поддержки процесса принятия решения с применением функций принадлежности на основе нормального закона распределения Гаусса. Обсуждаются рекомендации по осуществлению процедурной модели для определения параметров функций принадлежности путем статистической обработки результатов, полученных в процессе работы экспертной группы. Приведен пример применения разработанной процедурной модели при оценке показателя «Приоритетное число возможности улучшения» (по результатам работы экспертной группы) при подготовке принятия решения по улучшению деятельности в испытательной лаборатории.

Ключевые слова: экспертные оценки, статистическая обработка, доверительный интервал, функции принадлежности: трапециевидные, треугольные, гауссова типа, определение параметров, поддержка принятия решений

TO THE QUESTION OF SUPPORTING THE DECISION-MAKING PROCESS ON PERFORMANCE IMPROVEMENT IN THE TESTING LABORATORY

The article was received by the editorial board on 29.10.2020, in the final version – 12.11.2020.

Al-Busaidi Said Sultan Said, Tambov State Technical University, 106 Sovetskaya St., Tambov, 392000, Russian Federation, post-graduate student, ORCID: www.orcid.org/0000-0003-3990-8123, e-mail: al-busaidi2020@hotmail.com

Voyakina Yuliya N., Tambov State Technical University, 106 Sovetskaya St., Tambov, 392000, Russian Federation, undergraduate student, ORCID: www.orcid.org/0000-0003-1654-4232, e-mail: miss.voyakina2011@yandex.ru

Ponomarev Sergey V., Tambov State Technical University, 106 Sovetskaya St., Tambov, 392000, Russian Federation, Doct. Sci. (Engineering), Professor, ORCID: www.orcid.org/0000-0003-0228-912X, e-mail: svponom@yahoo.com