

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ
БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
МЕХАНИКО-МАТЕМАТИЧЕСКИЙ ФАКУЛЬТЕТ
Кафедра дифференциальных уравнений и системного анализа

КОЛЯДА

Мария Игоревна

Агентный подход к моделированию цепочки поставок

Дипломная работа

Научный руководитель:
кандидат физ.-мат. наук,
доцент О. А. Лаврова

Допущена к защите

«___» _____ 2017 г.

Зав. кафедрой дифференциальных уравнений и системного анализа
доктор физ.-мат. наук, профессор В. И. Громак

Минск, 2017

ОГЛАВЛЕНИЕ

РЕФЕРАТ	3
РЭФЕРАТ	4
ABSTRACT	5
ВВЕДЕНИЕ	6
ГЛАВА 1 ИМИТАЦИОННОЕ МОДЕЛИРОВАНИЕ	8
1.1 Понятия и определения	8
1.2 Виды имитационного моделирования	10
1.2.1 Дискретно-событийное моделирование	11
1.2.2 Системная динамика	12
1.2.3 Агентное моделирование	13
1.3 ИМИТАЦИОННОЕ МОДЕЛИРОВАНИЕ В СРЕДЕ ANYLOGIC.....	14
1.3.1 Базовые инструменты для разработки модели	17
1.4 Выводы	20
ГЛАВА 2 РЕАЛИЗАЦИЯ МОДЕЛИ ЦЕПОЧКИ ПОСТАВОК.....	21
2.1 СБОР ГЕОДАННЫХ ОБ АГЕНТАХ	22
2.2 РЕАЛИЗАЦИЯ АГЕНТНОЙ МОДЕЛИ	24
2.2.1 Агент-Магазин и Агент-Заказ: создание запроса на поставку продуктов.....	25
2.2.2 Агент-Склад: пополнение складов товарами	25
2.2.3 Агент-Склад и Агент-Заказ: обработка заказов на складах.....	26
2.2.4 Агент-Грузовик: движение.....	28
2.2.5 Добавление складов и магазинов на Гис-карту	29
2.3 ВЫПОЛНЕНИЕ МОДЕЛИ.....	31
2.3.1 Модель 1. Торговая сеть «Евроопт» в Беларуси	32
2.3.2 Модель 2. Торговая сеть «ASDA» в Великобритании	33
2.4 Выводы	34
ЗАКЛЮЧЕНИЕ.....	35
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ.....	37

РЕФЕРАТ

В дипломной работе 36 страниц, 18 рисунков, 16 источников, 6 приложений.

ИМИТАЦИОННОЕ МОДЕЛИРОВАНИЕ, АГЕНТНОЕ МОДЕЛИРОВАНИЕ, СИСТЕМНАЯ ДИНАМИКА, ДИСКРЕТНО-СОБЫТИЙНОЕ МОДЕЛИРОВАНИЕ, ЦЕПОЧКА ПОСТАВОК, ГИС-КАРТА

Целью дипломной работы является разработка и тестирование имитационной модели цепочки поставок на основании агентного подхода, где среда моделирования реализована с помощью карты геоинформационной системы.

Объектом исследования является нелинейная цепочка поставок продуктов грузовиками из складов в магазины на основании заказов магазина.

В дипломной работе получены следующие результаты:

- Разработана анимированная имитационная модель доставки продуктов грузовиками из складов в магазины, движение которых происходит на карте геоинформационной системы средствами AnyLogic. Модель позволяет динамически отслеживать статистические данные о поведении системы.
- Протестирована имитационная модель на реальных геоданных торговых сетей «Евроопт» (Беларусь) и «ASDA» (Великобритания).

Дипломная работа носит практический характер. Разработанную модель цепочки поставок можно использовать для анализа и оптимизации функционирования торговых сетей.

Дипломная работа выполнена автором самостоятельно.

РЭФЕРАТ

У дыпломнай працы 36 старонак, 18 малюнкаў, 16 крыніц, 6 прыкладанняў.

ІМІТАЦЫЙНАЕ МАДЭЛЯВАННЕ, СІСТЭМНАЯ ДЫНАМІКА,
ДЫСКРЭТНА-ПАДЗЕЙНАЕ МАДЭЛЯВАННЕ, АГЕНТНАЕ
МАДЭЛЯВАННЕ, ЛАНЦУЖОК ПАСТАВАК, ГІС-КАРТЫ

Мэтай дыпломнай працы з'яўляецца распрацоўка і тэставанне імітацыйнай мадэлі ланцужка паставак на падставе агентнага падыхода, дзе асяроддзе мадэлявання рэалізаванае з дапамогай карты геаінфармацыйнай сістэмы ў асяроддзі мадэлявання распрацоўкі мадэляў AnyLogic.

Аб'ектам даследавання з'яўляецца нелінейная ланцужок паставак прадуктаў грузавікамі з складоў у крамы на падставе заказаў крамы.

У дыпломнай працы атрыманы наступныя вынікі:

- Распрацавана аніміраваная імітацыйная мадэль дастаўкі прадуктаў грузавікамі з складоў у крамы, рух якіх адбываецца на карце геаінфармацыйнай сістэмы сродкамі AnyLogic. Мадэль дазваляе дынамічна адсочваць статыстычныя дадзеныя аб паводзінах сістэмы.

- Пратэставана імітацыйная мадэль на рэальных геоданых гандлёвых сетак «Еўраопт» (Беларусь) і «ASDA» (Вялікабрытанія). Рэалізаваную мадэль ланцужка паставак можна выкарыстоўваць на практыцы на любых рэальных звестках гандлёвых сетак.

Дыпломная праца носіць практычны характар. Распрацаваную мадэль ланцужкі паставак можна выкарыстоўваць для аналізу і аптымізацыі функцыянавання гандлёвых сетак.

Дыпломная праца выканана аўтарам самастойна.

ABSTRACT

Graduation project consists of 36 pages, 18 illustrations, 16 sources, 6 appendices.

SIMULATION MODELING, SYSTEM DYNAMICS, DISCRETE EVENT MODELING, AGENT BASED MODELING, SUPPLY CHAIN, GIS MAPS

The aim of graduation project is the development and testing of the supply chain simulation model based on agent based perspective, where simulation area is implemented using geoinformation system map in the AnyLogic model development environment.

The object of the research is a non-linear supply chain of products by trucks from distribution centers to retailers on the basis of retailers' orders.

In the Graduation project the following results were obtained:

- An animated simulation model of delivering products by trucks from distribution centers to retailer has been developed, the movement of which proceed on the map of the geoinformation system using AnyLogic. The model allows to dynamically monitor statistical data about the behavior of the system;
- The simulation model for real geodata of retail chains "Euroopt" (Belarus) and "ASDA" (Great Britain) was tested. The implemented supply chain model can be used in practice with any real data of store retailers.

The Graduation project has a practical nature. The developed supply chain model can be used to analyze and optimize the functioning of retail chains.

Graduation project was made by the author herself.

ВВЕДЕНИЕ

Благодаря интенсивному развитию компьютерных технологий стало намного проще решать сложные задачи, требующие больших временных и финансовых затрат. Намного упростить их решение возможно с использованием моделирования.

Одним из наиболее распространенных и удобных способов моделирования сложных систем является имитационное компьютерное моделирование объектов и процессов реального мира.

Имитационное моделирование логистических процессов позволяет повысить точность и надежность принимаемых управленческих решений, снизить финансовые и бизнес-риски, обнаружить экстренные ситуации, требующие пристального внимания менеджеров.

Только выбор оптимального расположения распределительных центров (складов) с помощью имитационного моделирования может быть ключом к повышению конкурентоспособности торговой или производственной сети.

Для создания имитационного моделирования имеется очень много программных пакетов, но большинство из них имеют как несомненные достоинства, так и довольно весомые недостатки, среди которых можно привести: узкую направленность, нелокализованный интерфейс, привязку моделей к среде разработки (не автономность) и дороговизну.

Но среди всех инструмент имитационного моделирования явно выделяется такой пакет как AnyLogic, поскольку в отличие от остальных инструментов в нем можно реализовать сразу все направления имитационного моделирования по отдельности или даже в рамках одной модели.

Имитационное моделирование AnyLogic позволяет оценить варианты размещения распределительных центров (складов) в разных точках страны и их экономическую эффективность в рамках цепи поставок, а также выработать

общие рекомендации по структуре логистической сети. В системе AnyLogic пользователь может также еще точнее понять структуру сети поставок благодаря визуализации поставки на ГИС-карте и в дальнейшем расширять модель до необходимого уровня с помощью возможностей AnyLogic. На основании всей полученной информации можно понять, каким образом получится снизить расходы на перевозку товара.

Целью дипломной работы является разработка и тестирование имитационной модели цепочки поставок с использованием таких подходов как: системная динамика, агентное и дискретно-событийное моделирование, где среда моделирования реализована с помощью карты геоинформационной системы в средствах AnyLogic.

Выполнение данной модели проверяется на реальных географических данных распределительных центров (складов) и магазинов двух крупных торговых сетей из разных стран:

- ASDA (Великобритания);
- Евроопт (Беларусь).

Обе эти сети являются крупнейшими сетями розничных магазинов в своих регионах.

Дипломная работа разделена на две части:

- Теоретическая – где приводятся основные, необходимые для понимания, определения; описания использованных в работе подходов имитационного моделирования и небольшое описание пакета AnyLogic.

- Практическая – где приводятся основные шаги, необходимые для создания имитационной модели, и результаты выполнения имитационной модели на геоданных торговых сетей ASDA и Евроопт.

ГЛАВА 1

ИМИТАЦИОННОЕ МОДЕЛИРОВАНИЕ

1.1 Понятия и определения

Система – это упорядоченное множество структурно взаимосвязанных и функционально взаимодействующих однотипных элементов любой природы, объединенных в целостный объект, состав и границы которого определяются целями системного исследования. [1]

Модель – это система (объект), исследование которой служит средством для получения информации о другой системе. [2]

Моделирование – процесс отражения свойств одного объекта (оригинала) в другом объекте (модели). [3]

Логистическая сеть – совокупность взаимосвязанных и взаимозависимых центров логистических действий, которые, если они существуют, образуют некоторую сетевую структуру. Составной частью логистической сети является цепочка поставок.

Логистическая система – это относительно устойчивая совокупность звеньев (структурных/функциональных подразделений компании, а также поставщиков, потребителей и логистических посредников), взаимосвязанных и объединенных единым управлением корпоративной стратегии организации бизнеса.[4]

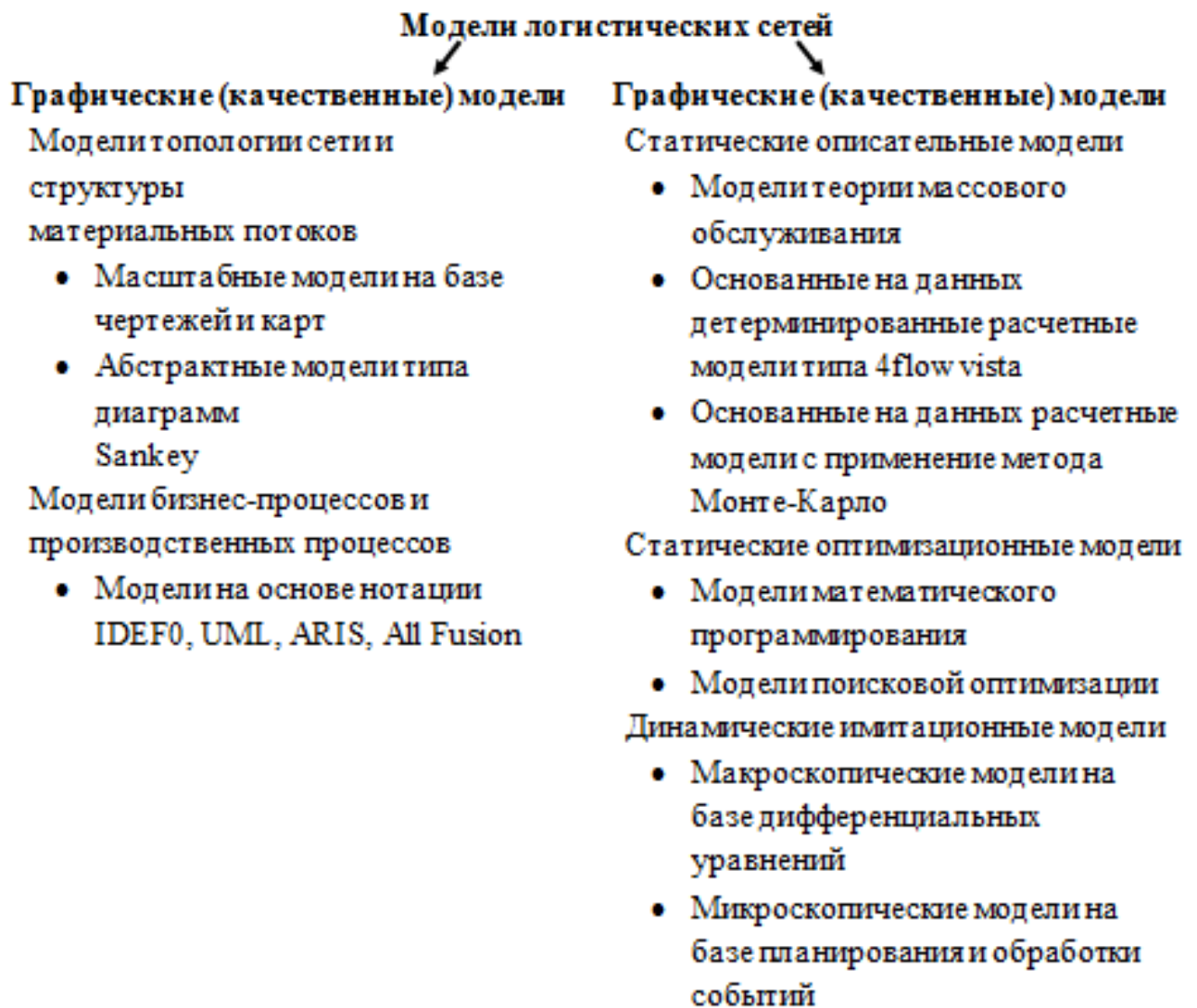


Рисунок 1.1. Методы моделирования логистических сетей

Имитационное моделирование – это разработка модели системы в виде программы для компьютера и проведение экспериментов с программой, вместо проведения экспериментов с реальной системой или объектом. [1]

Имитационное моделирование применяют, когда параметров много, зависимости не линейны, система имеет качественно различные состояния (непрерывные процессы прерываются дискретными переходами), траекторию во времени (объект эволюционирует), обладает вероятностным поведением и обратными связями. Имитационное моделирование незаменимо, когда нужно сопровождать модель анимационной презентацией (симуляцией). [3]

В любом исследовании, которое связано с использованием имитационного моделирования в логистике, можно выделить такие этапы:

1. Разработка концептуальной модели.
2. Разработка имитационной модели с использованием соответствующего программного пакета.
3. Планирование и проведение экспериментов с работающей моделью. [5]

1.2 Виды имитационного моделирования



Рисунок 1.2. Виды имитационного моделирования

В работе были использованы такие виды имитационного моделирования, как:

- Дискретно-событийное моделирование
- Системная динамика
- Агентное моделирование

1.2.1 Дискретно-событийное моделирование

В *дискретно-событийном моделировании* функционирование системы представляется как хронологическая последовательность событий.

В дискретно-событийном моделировании выделяются состояния системы и описываются действия, которые переводят ее из одного состояния в другое. Состояние системы может меняться только в моменты свершения событий.

Данное направление моделирования используют при описания работы системы или процесса из одного состояния в другое дискретным образом в виде события.

В более узком смысле "дискретно-событийное моделирование" используют, обозначая "процессное" моделирования, здесь динамика системы представлена в виде последовательности операций над некими сущностями. Данные сущности пассивны, сами они свою динамику не контролируют, но в то же время они могут иметь определённые атрибуты, которые влияют на процесс их обработки или накапливают статистические данные. Дискретно-событийное моделирование используется на среднем или низком уровне абстракции, где: моделирование каждого объекта происходит индивидуально, каждый объект как отдельная сущность, но множество деталей "физического уровня" опускается. [6]

Этот вид моделирования чаще всего используется для производственных процессов, где динамика системы может быть представлена как последовательность операций. [1] Используется так же для решения задач связанных с логистикой и системой массового обслуживания.

Примеры программных пакетов для реализации такого направления имитационного моделирования как дискретно-событийное моделирование: GPSS, SYMULA, Arena, AutoMod, Enterprise Dynamics, FlexSim, AnyLogic и др.

1.2.2 Системная динамика

Системная динамика – это такой подход моделирования, в котором для изучаемой системы строятся графические диаграммы причинных связей и глобальных влияний одних параметров на другие во времени, а затем созданная на основе этих диаграмм модель имитируется на компьютере.

Можно сказать, что такой вид моделирования лучше всех других подходов помогает разобраться в сути происходящего выявления причинно-следственных связей между объектами и явлениями. С использованием системной динамики можно реализовать модели производства, общества, мировой динамики, бизнес-процессов, динамики популяции, экологии, развития эпидемии и прочие.

Системная динамика – это подход, который своими методами и инструментами позволяет понять структуру и динамику сложных систем. Также системная динамика – это метод моделирования, который используется для создания точных компьютерных моделей сложных систем для проектирования более эффективной организации и политики взаимоотношений с данной системой для дальнейшего использования.

Системная динамика в основном используется в долгосрочных, стратегических моделях и принимает высокий уровень абстракции. Люди, продукты, события и другие дискретные элементы в моделях системной динамики рассматриваются не как отдельные элементы, а как система в целом.

[1]

Примеры программных пакетов для реализации такого направления имитационного моделирования, как системная динамика: Easy5, Lab View, VisSim, MATLAB, AnyLogic и др.

1.2.3 Агентное моделирование

Агентное моделирование – направление имитационного моделирования, один из современных методов, позволяющих изучить работу рассредоточенных агентов и то, как поведение отдельных агентов определяет поведение системы целиком.

По сравнению с системной динамикой в агентном моделировании можно определить поведение каждого агента на индивидуальном уровне, а поведение всей системы получить в результате деятельности множества агентов, т.е. моделирование происходит «снизу вверх». [1]

Составляющие агентного моделирования:

Агент – это некая сущность, обладающая активностью, автономным поведением, может принимать решения в соответствии с некоторым набором правил, взаимодействовать с окружением, а также самостоятельно изменяться. Агентами могут быть: люди (потребители, жители, работники пациенты, доктора, клиенты, солдаты и др.); транспорт, оборудование (автомобили, краны, самолёты, вагоны, станки, ...); нематериальные вещи (проекты, продукты, инновации, идеи, инвестиции...); организации (компании, политические партии, страны, ...).

Среда – некоторое пространство, в котором находятся агенты, характеризуемая своими состояниями и факторами, агенты находятся в определенном месте этого пространства, с возможностью ориентирования и передвижения в данном пространстве;

Правила взаимодействия – законы взаимодействия агентов в окружающей среде, с процедурами принятия решения и выбора стратегии при очередном шаге взаимодействия.

Задача имитационного моделирования при агентном подходе заключается в определении характеристик состояния агентов и среды, изучения поведения

агентов при различных ситуациях взаимодействия и изменяющихся состояниях среды.

Через изучение поведения множества агентов в некотором пространстве согласно некоторым правилам взаимодействия, прогнозирование поведения системы в целом.

Целью создания агентных моделей является получение представления об этих глобальных правилах, общем поведении системы, исходя из предположений об индивидуальном, частном поведении ее отдельных активных объектов и взаимодействии этих объектов в системе.

Агентный подход позволяет исследовать задачи коллективного взаимодействия, эффективно решать задачи прогнозирования. Агентные системы позволяют исследовать процессы самоорганизации, дают возможность естественного описания сложных систем, обладают высокой гибкостью. [1]

Примеры программных пакетов для реализации такого направления имитационного моделирования, как агентное моделирование: Net Logo, Swarm, Repast, ASCAPE, AnyLogic и др.

1.3 Имитационное моделирование в среде AnyLogic

AnyLogic — это программное обеспечение для имитационного моделирования сложных систем и процессов, позволяющее поддерживать направление агентного моделирования, дискретно-событийного моделирования и разработки моделей системной динамики (именно поэтому данный программный пакет и получил такое название). [1]

Кроме того в AnyLogic возможно реализовать в пределах одной модели комбинацию всех трех перечисленных выше подходов имитационного моделирования (см. Рисунок 1.3).

В AnyLogic имеется графический язык моделирования, в нем также можно расширить созданные модели, используя язык Java. [8]



Рисунок 1.3. Три подхода имитационного моделирования в AnyLogic

В AnyLogic можно разработать модели в совершенно различных сферах, таких как:

- логистика и цепочки поставок;
- здравоохранение и фармацевтика;
- рынок и конкуренция;
- телекоммуникации и информационные системы;
- бизнес-процессы и сфера обслуживания;
- производство;
- оборона;
- управление активами и проектами;
- социальные и экологические системы;
- пешеходная динамика. [1]

Клиентами AnyLogic является большое количество крупных компаний по всему миру, примеры см. Рисунок 1.4



Рисунок 1.4. Клиенты AnyLogic

В редакторе AnyLogic возможно создать анимацию и интерактивный графический интерфейс модели. Анимация может быть иерархической и поддерживать несколько перспектив. Например, возможно отобразить глобальную перспективу на процесс производства, а также детальные анимации конкретных операций и переключение между ними. [1]

Модель и анимация в AnyLogic строятся в формате drag-and-drop, а также гибко параметризуется. Пользователь может довольно просто реализовать стандартные объекты, как угодно расширить их функциональность, вплоть до того, что создать собственные библиотеки. Пользователь может создавать масштабируемые модели, используя иерархию и регулярные структуры объектов.

1.3.1 Базовые инструменты для разработки модели

Окно редактора AnyLogic состоит из следующих элементов (см. Рисунок 1.5):

- Панели инструментов;
- Панель Проекты;
- Панель Палитра;
- Панель Ошибки;
- Окно свойств;
- Окно графического редактора.

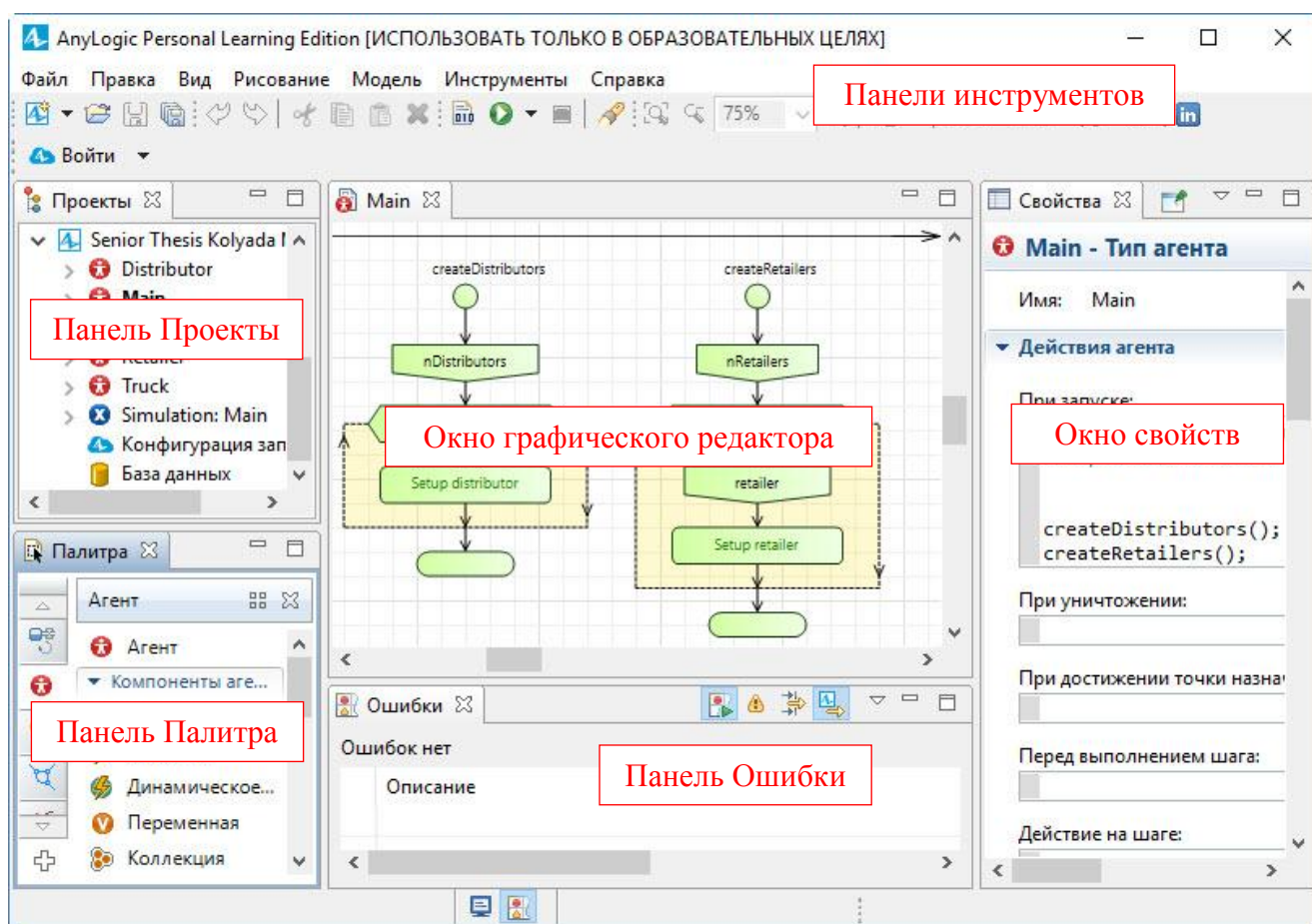


Рисунок 1.5. Интерфейс AnyLogic

1.3.1.1 Панели инструментов

Панели инструментов предоставляют пользователю быстрый доступ к наиболее часто используемым командам. При наведении мыши на кнопку

панели инструментов отображается подсказка с именем данной кнопки. Пример: Создать, Открыть модель, Повторить, Отменить и прочие.

1.3.1.2 Панель проекты

В панели Проекты реализована навигация по элементам модели, которые открыты в данный момент времени. Модель организована иерархически, поэтому изображается в виде дерева: верхний уровень дерева образован самой моделью; следующий уровень образован экспериментами, типами агентов и Java-классами; элементы и т.д.

1.3.1.3 Панель Палитра

В панели Палитра расположены все элементы, которые Пользователь может добавить на диаграмму агента или эксперимента. Каждый из элементов относится к определенной задаче:

- в основной содержатся элементы, при помощи которых Пользователь может реализовать структуру, данные и динамику модели;
- системная динамика состоит из: элементов диаграммы потоков и накопителей, параметров, соединителей и табличной функции;
- диаграмма состояний состоит из блоков диаграмм, которые позволяют задавать поведение объекта графически;
- диаграмма действий состоит из блоков структурированных блок-схем, которые позволяют визуально задавать алгоритмы;
- статистика состоит из элементов, которые используются для сбора, анализа и отображения результатов выполнения модели;
- презентация состоит из элементов для создания презентаций: примитивных фигур, элементов управления, и необходима для придания презентации интерактивности;
- внешние данные состоят из инструментов, позволяющих работать с базами данных и текстовыми файлами;

- картинки состоят из набора изображений объектов, которые наиболее часто моделируют, например: человек, грузовик, фура, погрузчик, склад, магазин и т. д. [1]

1.3.1.4 Панель Ошибки

Во время компиляции модели AnyLogic проверяет синтаксис диаграмм, типов и параметров. Если в процессе компиляции и построения модели были обнаружены ошибки, то они отобразятся в панели Ошибки. Для каждой ошибки приводится ее описание и местоположение в модели, а именно: имя элемента модели, во время задания которого и была допущена ошибка.

1.3.1.5 Окно свойств

В панели Свойств можно просмотреть и изменить свойства рассматриваемого в текущий момент элемента модели. При выборе какого-либо элемента, например, из панели Проекты или же в окне графического редактора, в панели Свойства отобразятся свойства выбранного элемента. Некоторые из свойств элементов можно задать с использованием функций, написанных с использованием синтаксиса Java.

1.3.1.6 Окно графического редактора

У каждого типа агентов и эксперимента имеется своя собственная диаграмма, которая можно отредактировать в окне графического редактора.

На диаграмме эксперимента можно:

- Создать презентацию для этого эксперимента. Она будет отображаться в окне презентации при запуске этого эксперимента. Презентация может содержать интерактивные элементы управления, с помощью которых вы можете, например, изменять параметры эксперимента перед запуском модели.
- Добавить на презентацию элементы сбора данных, полученных в результате моделирования и графики, визуализирующие эти данные.

На диаграмме типа агентов вы можете:

- Нарисовать презентацию типа с помощью фигур презентации и элементов управления.
- Задать интерфейс типа агентов.
- Нарисовать значок типа.
- Задать поведение агента с помощью событий и диаграмм действий.
- Задать структуру типа, добавив вложенные или библиотечные объекты (и возможно, установив между ними взаимосвязи).
- Добавить на презентацию элементы сбора данных, полученных в результате моделирования и графики, визуализирующие эти данные.

1.3.1.7 Выполняемая модель

При выполнении модели запускается компилятор, который построит исполняемый модельный код в языке Java, оттранслирует его и после запустит модель на выполнение. И при этом откроется окно презентации.

1.4 Выводы

В первой главе:

- Были приведены теоретические сведения, касающиеся понятия имитационного моделирования.
- Подробно рассказано о трех видах имитационного моделирования, использованных в работе.
- Приведена базовая информация, необходимая для понимания процесса работы в программном пакете AnyLogic.

ГЛАВА 2

РЕАЛИЗАЦИЯ МОДЕЛИ ЦЕПОЧКИ ПОСТАВОК

В практической части работы была реализована имитационная модель управления поставками продукции. Данная модель реализована в системе AnyLogic PLE с использованием геоинформационных систем на реальных данных. AnyLogic PLE – бесплатная версия пакета AnyLogic, которая может быть использованная только в учебных целях.

В качестве исходных данных использовались следующие модельные показатели:

- количество складов;
- количество магазинов;
- местоположение складов и магазинов;
- размер автопарка каждого склада;
- скорость восполнения запасов товаров на складах;
- вместимость складов;
- количество товаров, заказываемых магазином;
- частота появления заявки на доставку.

Работа модели проверялась на реальных геоданных таких торговых сетей как:

- Сеть супермаркетов «Евроопт» в Беларуси [12]:
 - количество складов - 11
 - количество магазинов - 416
- Сеть супермаркетов «ASDA» в Великобритании [13]:
 - количество складов - 16
 - количество магазинов - 499

Выполнение созданной модели представлено в виде анимации передвижения грузовиков между складами и магазинами на ГИС-карте (см.

ПРИЛОЖЕНИЕ В, ПРИЛОЖЕНИЕ Д). В процессе выполнения имитационного эксперимента собирается статистическая информация о работе модели, представленная в виде динамически изменяющихся диаграмм (см. ПРИЛОЖЕНИЕ Г, ПРИЛОЖЕНИЕ Е).

2.1 Сбор геоданных об агентах

Сбор данных о местоположении магазинов осуществлялся посредством подключения к API карт Open Street Maps [14] поскольку собрать такое количество данных вручную было бы очень трудоемко. Данные карты используются также и в AnyLogic.

Реализация была выполнена на языке Python в среде разработки Jupyter Notebook с помощью встроенной библиотеки requests, а именно:

- Серверу OPS [16] отправляется поисковой get-запрос с определенными параметрами:
 - что искать: 'евроопт' или 'asda';
 - где искать: по координатам Беларуси или Великобритании (эти данные необходимо было подавать по небольшим координатным квадратам, поскольку серверы карт почему-то предоставляют не более 50 результатов на запрос).
- В ответ сервер OPS присылает результаты поискового запроса в формате json.

Далее полученные данные необходимо было обработать, поскольку результаты поиска выдают много параметров объекта, которые не требуются для задания местоположения в AnyLogic, например: id, класс объекта, название и прочее. Необходимо было получить только данные долготы и широты и свести эти данные в удобный для использования AnyLogic формат, а именно Excel.

Код программы см. ПРИЛОЖЕНИЕ А.

Сбор данных о местоположении складов осуществлялся вручную, поскольку количество складов значительно меньше количества магазинов. Для сбора уже были использованы карты Google Maps [15], поскольку данные по объектам в Open Street Maps вносятся самими пользователями карты, и такие объекты, которые вряд ли могут пригодиться для поиска обычному пользователю (например, склады), как правило, никто не вносит. Из результатов поисковых запросов также выбирались долгота и широта и затем вносились в Excel.

В Excel значения широты хранятся в первом столбце, значения долготы - во втором. Геоданные для магазинов хранятся на странице Retailers, для складов на странице Distributors.

The screenshot shows a Microsoft Excel window titled 'EurooptLatLon.xlsx - Microsoft Excel (Сбой актив...)'. The 'Retailers' sheet is active. The data is as follows:

	A	B	C	D	E
406	53.8408954	27.7025756			
407	53.9086038	27.548221			
408	53.91579325	27.439343425			
409	53.8052826	30.9749945465388			
410	52.5273228	24.9835976			

Рисунок 2.1. Отображение собранных геоданных в Excel

2.2 Реализация агентной модели

При построении модели были использованы три описанные ранее методологии имитационного моделирования: системная динамика (см. 1.2.2), агентное (см. 1.2.3) и дискретно-событийное (см. 1.2.1) моделирование.

В модели участвуют 4 агента: Склад, Магазин, Грузовик и Заказ

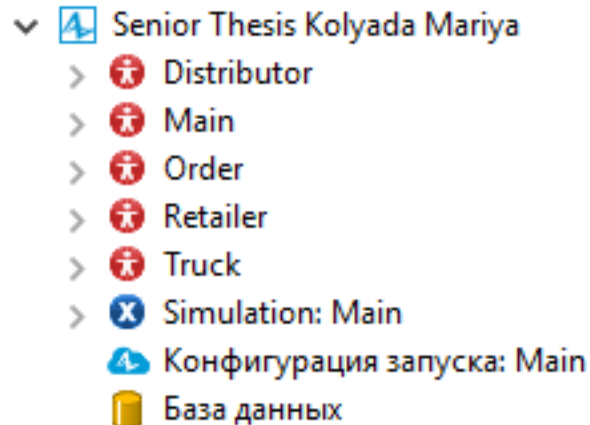


Рисунок 2.2. Дерево модели

Большая часть исходных данных в модели задается при помощи:

- равномерного непрерывного распределения (встроенная функция `uniform ()`).

$$f(x) = \frac{1}{\max(x) - \min(x)} \quad (2.1)$$

- равномерного дискретного распределения (встроенная функция `uniform_discr()`)

$$f(x) = \frac{1}{\max(x) - \min(x) + 1} \quad (2.2)$$

Равномерное распределение используется для задания случайной переменной, которая может принимать любое значение в интервале между `min` и `max` с равной вероятностью (вероятность максимального значения равна 0), например, для времени загрузки и разгрузки грузовика, для задания количества товаров на складе и прочее.

Дискретное равномерное распределение используется тогда, когда событие имеет ограниченное количество равновероятных исходов, например, количество продуктов, заказываемых магазином у склада, или скорость восполнения запасов на складе и прочее.

2.2.1 Агент-Магазин и Агент-Заказ: создание запроса на поставку продуктов

В модели процесс доставки начинается, когда один из магазинов отправляет запрос на поставляемый продукт на склад. Для создания такого запроса используется объект Событие *generateDemand*, задающийся в окне графического редактора агента *Retailer(Магазин)*. Т.е. создается объект *order* класса *Order*, данные которые подаются в конструктор это кто заказывает, в нашем случае это магазин (*retailer*) и сколько товара

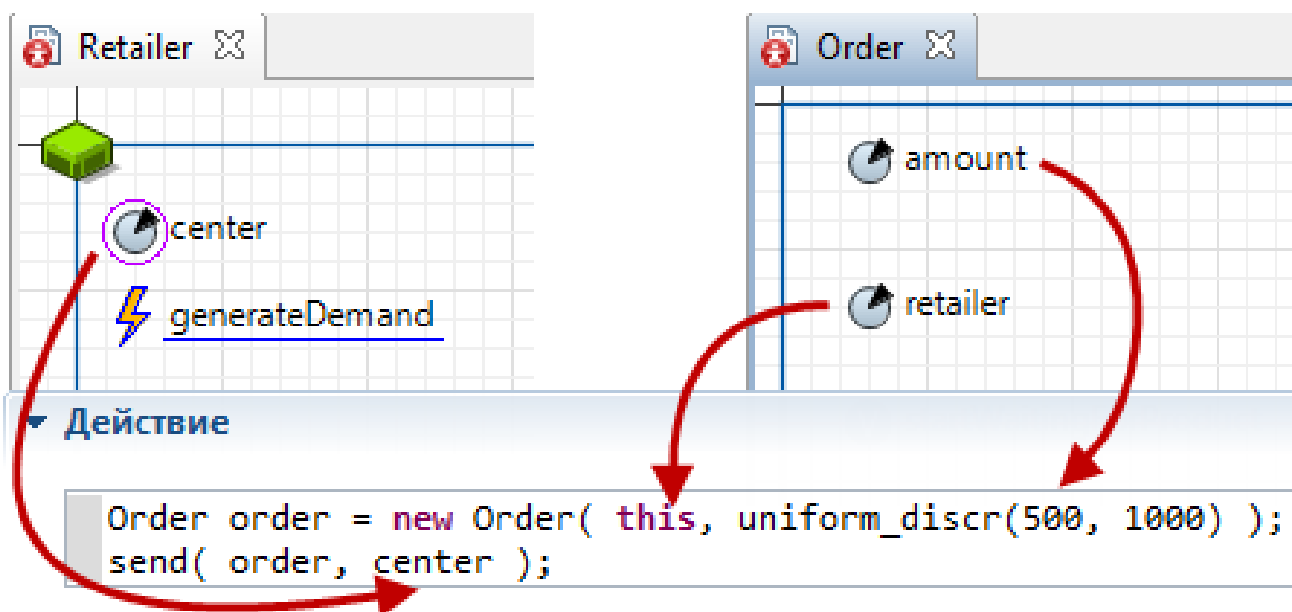


Рисунок 2.3. Действие события *generateDemand*

2.2.2 Агент-Склад: пополнение складов товарами

Системная динамика (см. 1.2.2) в модели представлена диаграммой пополнения складов товарами (см. Рисунок 2.4). Данная диаграмма задается в окне графического редактора агента *Distributor(Склад)*. Поток восполнения

запасов товаров на складах - *productionRate* формируется параметром *capacity*. Параметр *productsInStorage* определяет вместимость складов. Текущее количество товаров на складе показывает *products*, который после загрузки грузовика товарами уменьшается на величину заказа.

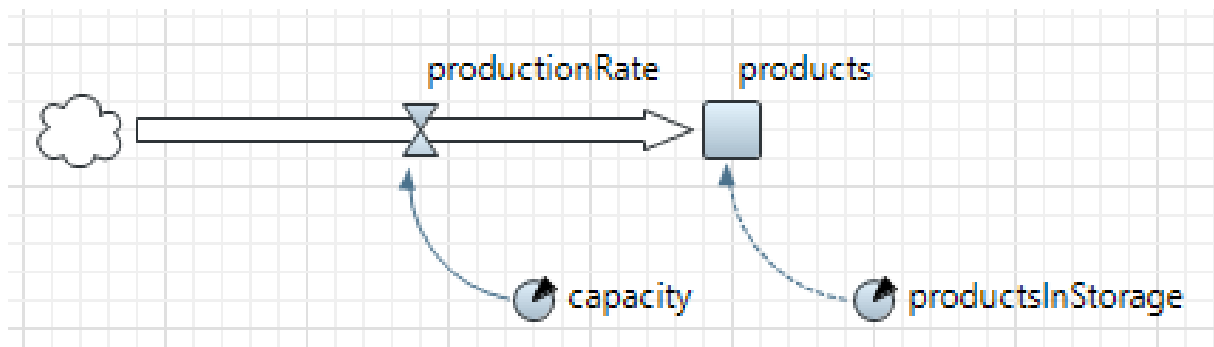


Рисунок 2.4. Диаграмма пополнения складов товарами

2.2.3 Агент-Склад и Агент-Заказ: обработка заказов на складах

Дискретно-событийное (см. 1.2.1) моделирование используется при обработке заказов на складах (см. Рисунок 2.5). Данная диаграмма также задается в окне графического редактора агента Distributor (Склад). У каждого склада имеется парк грузовых автомобилей— *fleet*. Когда на склад приходит заказ от торговой точки, происходит проверка наличия товаров на складе — *waitForProduct*. Если товар в требуемом количестве имеется на складе, то к торговой точке отправляется грузовик с товаром (см. Рисунок 2.7). Иначе, ожидается пополнение склада товарами.

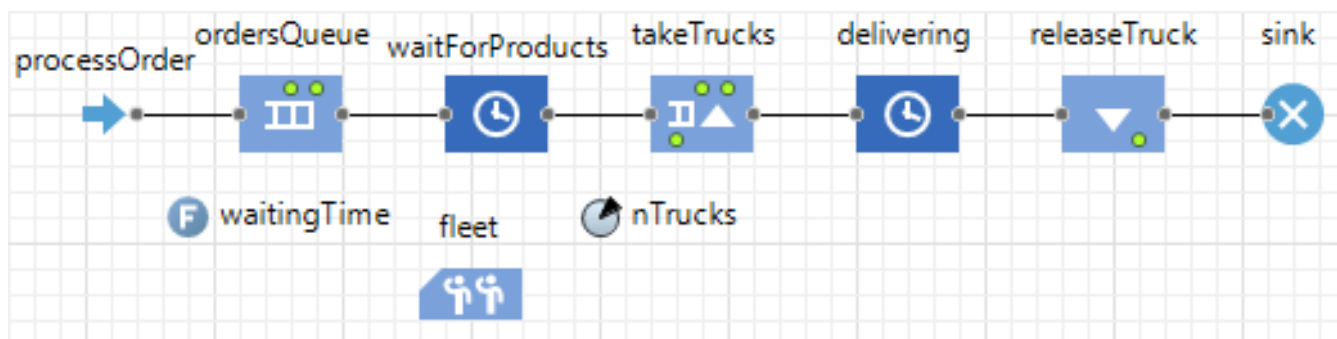


Рисунок 2.5. Процесс обработки заказа на складе

processOrder – объект, который вставляет существующих агентов в блок-схему процесса.

ordersQueue – объект, который формирует очередь между объектами для перехода на следующий шаг.

waitForProducts - представляет временной интервал, необходимый для проверки наличия товаров на складе, для этого используется функция *waitingTime* (см. Рисунок 2.6).

```
if( order.amount <= products )  
    return 0;  
else  
    return (order.amount - products )/productionRate;
```

Рисунок 2.6. Тело функции *waitingTime*

takeTrucks – объект, который захватывает ресурсы из объекта *fleet* (парк грузовых автомобилей, в котором имеется *nTrucks* грузовиков), чтобы использовать их.

delivering - представляет временной интервал, который требуется грузовику для доставки продукта. Поскольку продавцы расположены в разных местах, неизвестно точно, сколько времени грузовикам необходимо времени, чтобы добраться до них, поэтому задержка заканчивается, когда грузовик прибывает и сигнализирует об этом.

releaseTrucks - естественно освобождает ресурсы, которые были взяты *takeTrucks* , когда эти единицы больше не нужны. Прежде чем агенты будут размещены в конце блок-схемы, они должны освободить все свои ресурсы.

sink - заканчивает блок-схему.

2.2.4 Агент-Грузовик: движение

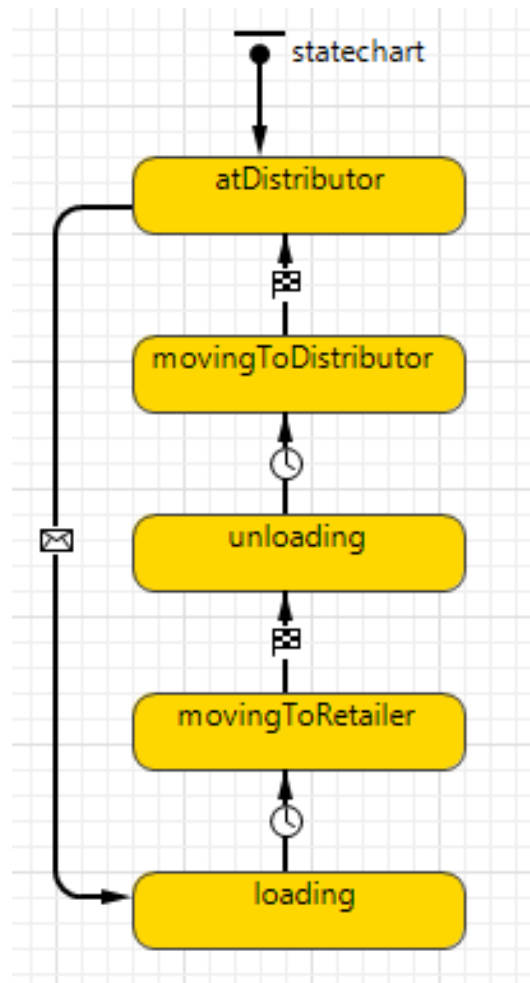


Рисунок 2.7. Схема движения грузовиков

Грузовик принимает на складе заявку на загрузку товара, загрузка *loading* занимает 1-2 ч, т.е. переход в следующее состояние осуществляется через 1-2 часа. После грузовик отправляется к магазину — источнику заявки (см. Рисунок 2.7, Рисунок 2.8).

Происходит:	По таймауту
Таймаут:	uniform(1, 2) часы
Действие:	moveTo(order.retailer);
Доп. условие:	

Рисунок 2.8. Свойства перехода из состояния *loading* в *movingToRetailer*

Грузовики перемещаются по существующим автомобильным дорогам; маршрут движения создается при отправлении грузовика к целевой точке. По прибытии, грузовика осуществляется разгрузка *unloading*, переход с данного состояние в другое, так же как и после загрузки осуществляется через 1-2 ч. После возвращается обратно в парк и заявка помечается как обслуженная. Схема движения грузовика задается в виде диаграммы состояний в окне графического редактора агента *Truck (Грузовик)*.

2.2.5 Добавление складов и магазинов на Гис-карту

Для добавления складов и магазинов на карту были созданы функции *createDistributors* и *createRetailers* соответственно, представляющие собой диаграммы действий в графическом редакторе эксперимента (см. Рисунок 2.9).

В данные диаграммы подаются геоданные (долгота и широта) из элемента Файла Excel, содержимое которого описывалось ранее (см. 2.1).

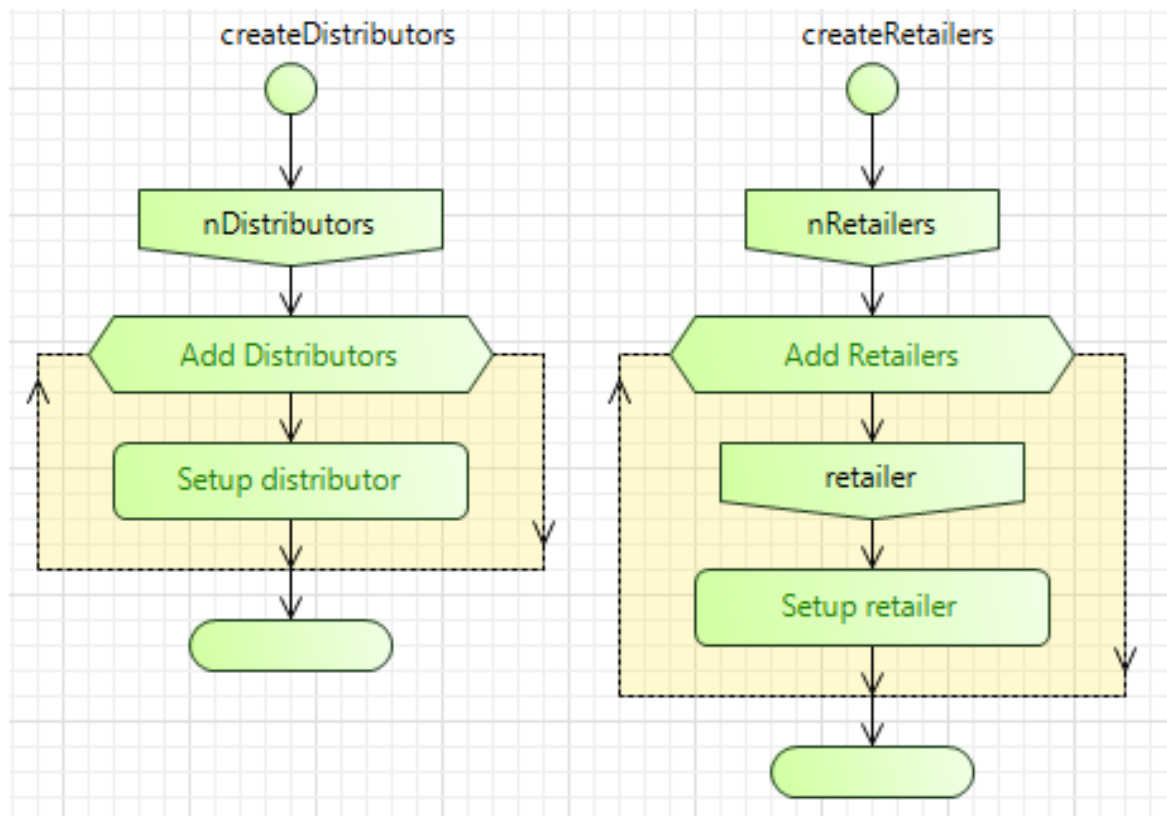


Рисунок 2.9. Диаграммы действий для создания магазинов и складов в Гис-пространстве

nDistributors и ***nRetailers*** – локальные переменная задающие количество складов и магазинов соответственно, которые соответствует номерам последних строк из Excel-файла с данными о долготе и широте (например, для супермаркетов «Евроопт»: *nDistributors*=11, *nRetailers*=416).

Имя: `nDistributors`

Тип: `int` ▼

Начальное значение: `coordinates.getLastRowNum( "Distributors" )`

Рисунок 2.10. Задание количества складов (*nDistributors*)

Add Distributors и ***Add Retailers*** – циклы *For*, проходящие по всем данным из Excel-файла с первой строки до *nDistributors* и *nRetailers* соответственно.

Тип: ☒ Цикл со счетчиком ☐ Итератор по коллекции

Инициализация: `int i = 1`

Условие: `i <= nDistributors`

Счетчик: `i++`

Рисунок 2.11. Цикл *For* по геоданным Складов

retailer – локальная переменная с типом агента *Retailer* (Магазин), т.е. объект, который необходимо расположить на карте.

Setup distributor и ***Setup retailer*** – в это части пишется код для задания локальных (глобальных) переменных долготы и широты в цикле и размещение каждого *i*-го объекта в Гис-пространстве (см. Рисунок 2.12, Рисунок 2.13).

```
latitude = coordinates.getCellStringValue("Distributors", i, 1);
longitude = coordinates.getCellStringValue("Distributors", i, 2);
add_distributors();
```

Рисунок 2.12. Код для популяции агентов *Distributor* (Склад)

```
double lat = Double.parseDouble(coordinates.getCellStringValue("Retailers", i, 1));
double lon = Double.parseDouble(coordinates.getCellStringValue("Retailers", i, 2));

retailer.setLatLon(lat, lon);
retailer.center = retailer.getNearestAgentByRoute( distributors );|
```

Рисунок 2.13. Код для популяции агентов *Retailer* (Магазин)

Для агента *Distributor* (Склад) переменные *latitude* и *longitude* задаются глобально в окне графического редактора эксперимента, и далее как видно по коду (см. Рисунок 2.12) каждой переменной *latitude* и *longitude* присваиваются значения из файла Excel *coordinates*, в котором берется текстовое значение каждой из ячеек первого и второго столбцов соответственно с листа «Distributors».

Аналогичным образом задаются переменные широты и долготы для популяции агентов *Retailer* (Магазин), только уже локально, и магазины размещаются в Гис-пространстве с использованием встроенной функции *setLatLon* (см. Рисунок 2.13).. Далее необходимо каждый магазин закрепить за определенным складом, чтобы при заказе магазином продуктов, продукты были привезены грузовиком из ближайшего склада, для этого используется встроенная функция *getNearestAgentByRoute*.

2.3 Выполнение модели

В процессе выполнения модели в презентации:

- На первой вкладке отображается анимация доставки товаров грузовиками из складов в закрепленные за этими складами магазины.
- На второй вкладке отображается динамически изменяющаяся статистика в виде диаграмм, отображающих:
 - данные о текущих запасах товаров на произвольно выбранных трех складах;

- информацию об использовании парка грузовиков произвольно выбранных трех складов;
- среднее время ожидания выполнения заявки на доставку.

2.3.1 Модель 1. Торговая сеть «Евроопт» в Беларуси

Модельные показатели:

- количество складов - **416**;
- количество магазинов - **11**;
- размер автопарка каждого склада – **uniform_discr (10, 20)**;
- скорость восполнения запасов товаров на складах - **uniform_discr (5000, 8000)**;
- вместимость складов - **uniform (100000, 200000)**;
- количество товаров, заказываемых магазином - **uniform (500000, 100000)**
- частота появления заявки на доставку - **uniform (1, 2)**.

Статистика:

- данные о текущих запасах товаров отображаются для следующих складов:
 - Склад под Минском;
 - Склад под Витебском;
 - Склад под Гомелем.
- информацию об использовании парка грузовиков отображаются для следующих складов:
 - Склад под Брестом;
 - Логистический центр в Барановичах;
 - Склад под Гродно.
- среднее время ожидания выполнения заявки на доставку.

Результаты:

- Фрагмент анимации модели см. ПРИЛОЖЕНИЕ В

- Фрагмент статистики см. ПРИЛОЖЕНИЕ Г

2.3.2 Модель 2. Торговая сеть «ASDA» в Великобритании

Модельные показатели:

- количество складов - **499**;
- количество магазинов - **16**;
- размер автопарка каждого склада – **uniform_discr (5, 15)**;
- скорость восполнения запасов товаров на складах - **uniform_discr (6000, 9000)**;
- вместимость складов - **uniform (100000, 200000)**;
- количество товаров, заказываемых магазином - **uniform (500000, 100000)**
- частота появления заявки на доставку - **uniform (1, 2)**.

Статистика:

- данные о текущих запасах товаров отображаются для следующих складов:
 - Склад под Лондоном;
 - Склад под Ливерпулем;
 - Склад под Белфастом.
- информацию об использовании парка грузовиков отображаются для следующих складов:
 - Склад под Эдинбургом;
 - Склад под Бристолем;
 - Склад под Латтервортом.
- среднее время ожидания выполнения заявки на доставку.

Результаты:

- Фрагмент анимации модели см. ПРИЛОЖЕНИЕ Д
- Фрагмент статистики см. ПРИЛОЖЕНИЕ Е

2.4 Выводы

Во второй главе:

- Были приведена постановка задачи.
- Описан процесс сбора геоданных для моделей.
- Приведены случаи, в которых использовались основные подходы имитационного моделирования.
- Описан процесс создания магазинов и складов в ГИС-пространстве.
- Приведены результаты созданной модели на геоданных торговых сетей «Евроопт» и «ASDA»

ЗАКЛЮЧЕНИЕ

После проведения теоретических и экспериментальных исследований в дипломной работе можно сделать следующие выводы:

- Без изучения основной теоретической информации, касающейся имитационного моделирования, понимания исследуемой предметной области получено быть не может.

- Пакет AnyLogic определенно можно назвать интересным и полезным для изучения.

- Среди достоинств AnyLogic можно выделить следующее:

- вполне доступный и понятный интерфейс;
- подробная справочная документация, представленная также и на русском языке;
- возможность изменения языка интерфейса, что будет очень актуально для пользователей, не знающих английский язык;
- большой выбор 2D и 3D анимации для визуализации агентов, анимацию также можно и загрузить из внешнего файла;
- возможность отображения различной интересной статистики при выполнении модели.
- большой выбор возможных способов задания объектов и их параметров, каждый пользователь может выбрать и использовать понравившийся ему способ, а не быть ограниченным одним.

и прочее.

- Среди недостатков AnyLogic можно выделить следующее:

- базовых знаний Java не всегда хватает, чтобы задать какие-нибудь свойства агентов, их элементов или параметров;
- учебная версия программы плохо работает на большом количестве данных, вычислительной мощности компьютера с процессором Intel

Core 3 для выполнения модели на данных Евроопт и ASDA периодически не хватало. Причины выяснить не удалось, но иногда модель запускалась уже через 10 минут, а иногда и 2 часа не могла посчитаться. При использовании же компьютера с процессором Intel Core 7 модель всегда считалась, время ожидания составляло до 20 минут.

- Независимо от всех приведенных выше недостатков учебной версии можно сказать, что AnyLogic отлично подходит для создания имитационных моделей и может быть использован организацией для проведения анализа текущего состояния процесса поставок и разработки улучшений.

- Также хочется еще раз отметить информацию о том, что доступные сервисы карт геоинформационных систем, например, Open Street Maps и Google Maps, использованные в работе, почему-то не дают возможность получить все данные поискового запроса, а только 50 или 60, что по мне довольно странно. Почему бы, если предоставляется возможность получать данные поисковых запросов, не предоставлять их все.

Единственная причина, которая пришла ко мне в голову – возможно потому что в данных поисковых сервисах встроена система ранжирования, и они предоставляют пользователю не все большое количество результатов поискового запроса, а только лучшие из результатов.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Маликов, Р. Ф. Практикум по имитационному моделированию сложных систем в среде AnyLogic 6 : учеб. пособие / Р. Ф. Маликов. Уфа: Изд-во БГПУ, 2013. – 296с.
2. Уёмов, А. И. Логические основы метода моделирования / А. И. Уёмов. – М.: Мысль, 1971. — 311 с.
3. Куприяшкин, А.Г. Основы моделирования систем: учеб. пособие / А.Г. Куприяшкин; Норильский индустр. ин-т. – Норильск: НИИ, 2015. – 135 с.
4. Савенкова, Т.И. Логистика : учеб. пособие / Т.И. Савенкова. — 5-е изд., стер. — М.: Издательство «Омега-Л», 2010. – 255 с.
5. Ю.И. Толуев, Имитационное моделирование логистических сетей / Ю.И. Толуев – Магдебург: Университет им. Отто фон Герике, Институт организации и автоматизации промышленного производства общества Фраунгофера. – 13 с.
6. Справочная система Anylogic. [Электронный ресурс] / Anylogic – Режим доступа: <http://www.anylogic.ru/anylogic/help/index.jsp?nav=/0>. – Дата доступа: 25.05.2017
7. Замятина, Е.Б. Архитектура агентно-ориентированной системы имитации с агентами, основанными на нейронных сетях / Е.Б. Замятина, Д.Ф. Каримов, А.А. Митраков – Пермь: ПГНИУ, 2014. – 13 с.
8. Java For AnyLogic Users [Электронный ресурс] – Режим доступа: http://www.xjtek.com/files/book/Java_for_AnyLogic_users.pdf
9. Видео-лекции об использовании AnyLogic [Электронный ресурс] / Anylogic – Режим доступа: <https://www.youtube.com/channel/UCdHe29FvfphfWmI2EMZPhg>. – Дата доступа: 14.05.2017

10. Топ-20 торговых сетей Беларуси [Электронный ресурс] / TUT.BY – Режим доступа: <https://news.tut.by/economics/484077.html>. – Дата доступа: 23.03.2017
11. ASDA [Электронный ресурс] / Wikipedia – Режим доступа: <https://en.wikipedia.org/wiki/Asda>. – Дата доступа: 23.03.2017
12. Сайт торговой сети «Евроопт» [Электронный ресурс] / ООО «Евроторг» – Режим доступа: <http://evroopt.by/>. – Дата доступа: 15.05.2017
13. Сайт торговой сети «ASDA» [Электронный ресурс] / ASDA – Режим доступа: <http://www.asda.com/>. – Дата доступа: 15.05.2017
14. Карты Open Street Map [Электронный ресурс] / OpenStreetMap – Режим доступа: <https://www.openstreetmap.org>. – Дата доступа: 13.04.2017
15. Карты Google Maps [Электронный ресурс] / Google – Режим доступа: <https://maps.google.com/>. – Дата доступа: 27.04.2017
16. Nominatim [Электронный ресурс] / OpenStreetMap Wiki – Режим доступа: <http://wiki.openstreetmap.org/wiki/Nominatim>. – Дата доступа: 13.04.2017

Код программы на Python

openstreetmap

```
In [1]: import requests
import json
import ast
import numpy as np
```

```
In [2]: belarus_squares = [[(57.0-i, 23.264771), (56.5-i, 32.427668)] for i in np.arange(0, 6, 0.5)]
uk_squares = [[(61.0-i, -10.541363), (60.0-i, 2.546294)] for i in range(13)]
```

```
In [3]: adress = 'http://nominatim.openstreetmap.org/search'
#q = 'asda'
q = 'евпоонт'
limit = 50
places = []

results = ast.literal_eval(request.text)
print(len(results), len(places))
except:
    print(request.text)
    break

places_ids = [result['place_id'] for result in results]
exclude_place_ids += places_ids
exclude_place_ids = list(np.unique(exclude_place_ids))

places += results
```

```
In [4]: shops = [place for place in places if place['class'] == 'shop']
unique_shops = list({shop['place_id']:shop for shop in shops}.values())
```

Магазины Евроопт

```
In [5]: unique_shops_asda = unique_shops
```

Магазины Асда

```
In [7]: unique_shops_euroopt = unique_shops
```

Сохранение результатов

```
In [8]: from openpyxl import Workbook
```

Широта и долгота

```
In [17]: wb = Workbook()

# grab the active worksheet
ws = wb.active
ws['A1'] = 'lat'
ws['B1'] = 'lon'

for i, shop in enumerate(unique_shops_asda):
    ws['A{}'.format(i+2)] = shop['lat']
    ws['B{}'.format(i+2)] = shop['lon']

# Save the file
wb.save("AsdaLatLon.xlsx")
```

```
In [11]: wb = Workbook()

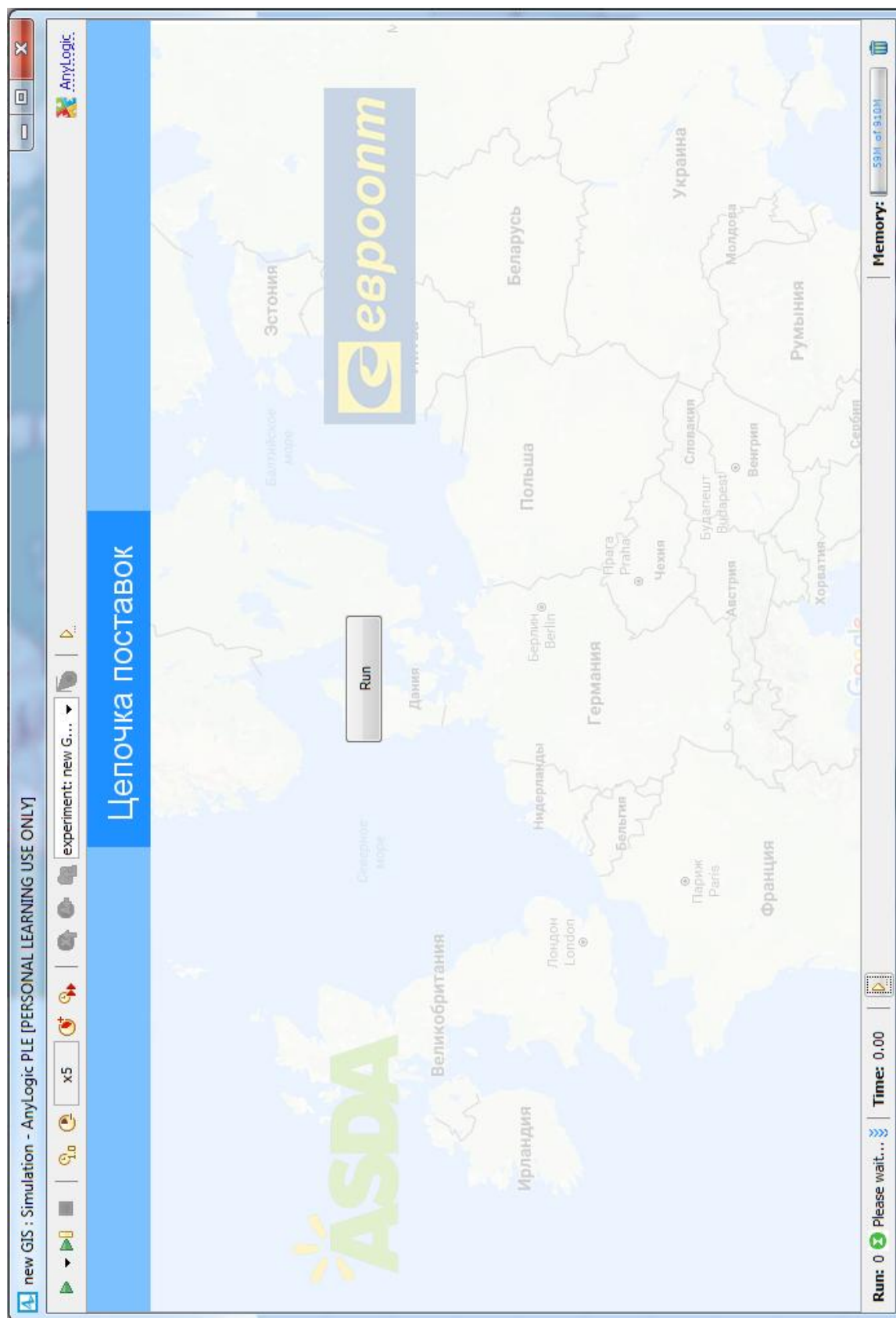
# grab the active worksheet
ws = wb.active
ws['A1'] = 'lat'
ws['B1'] = 'lon'

for i, shop in enumerate(unique_shops_euroopt):
    ws['A{}'.format(i+2)] = shop['lat']
    ws['B{}'.format(i+2)] = shop['lon']

# Save the file
wb.save("EurooptLatLon.xlsx")
```

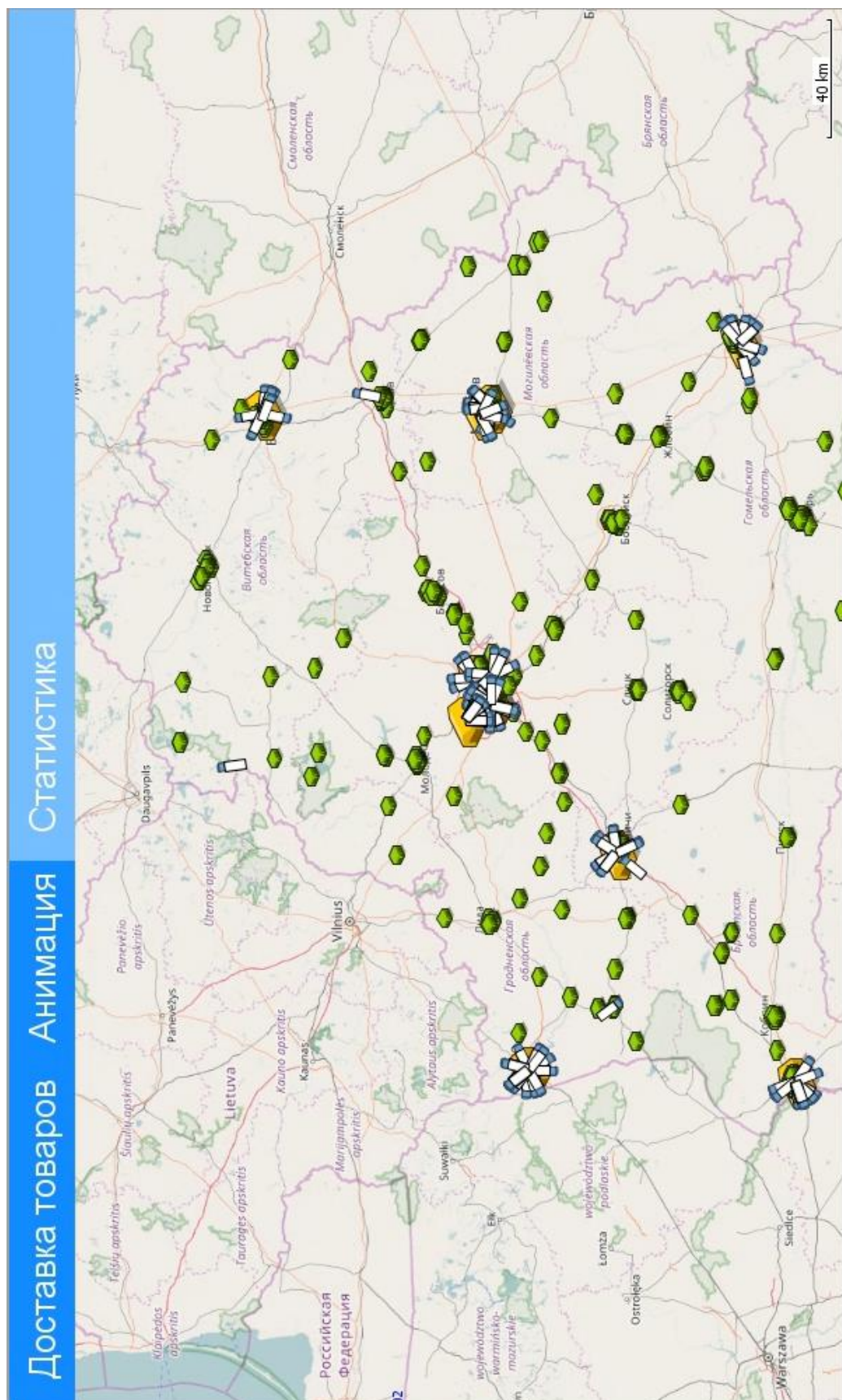
ПРИЛОЖЕНИЕ Б.

Страница запуска модели

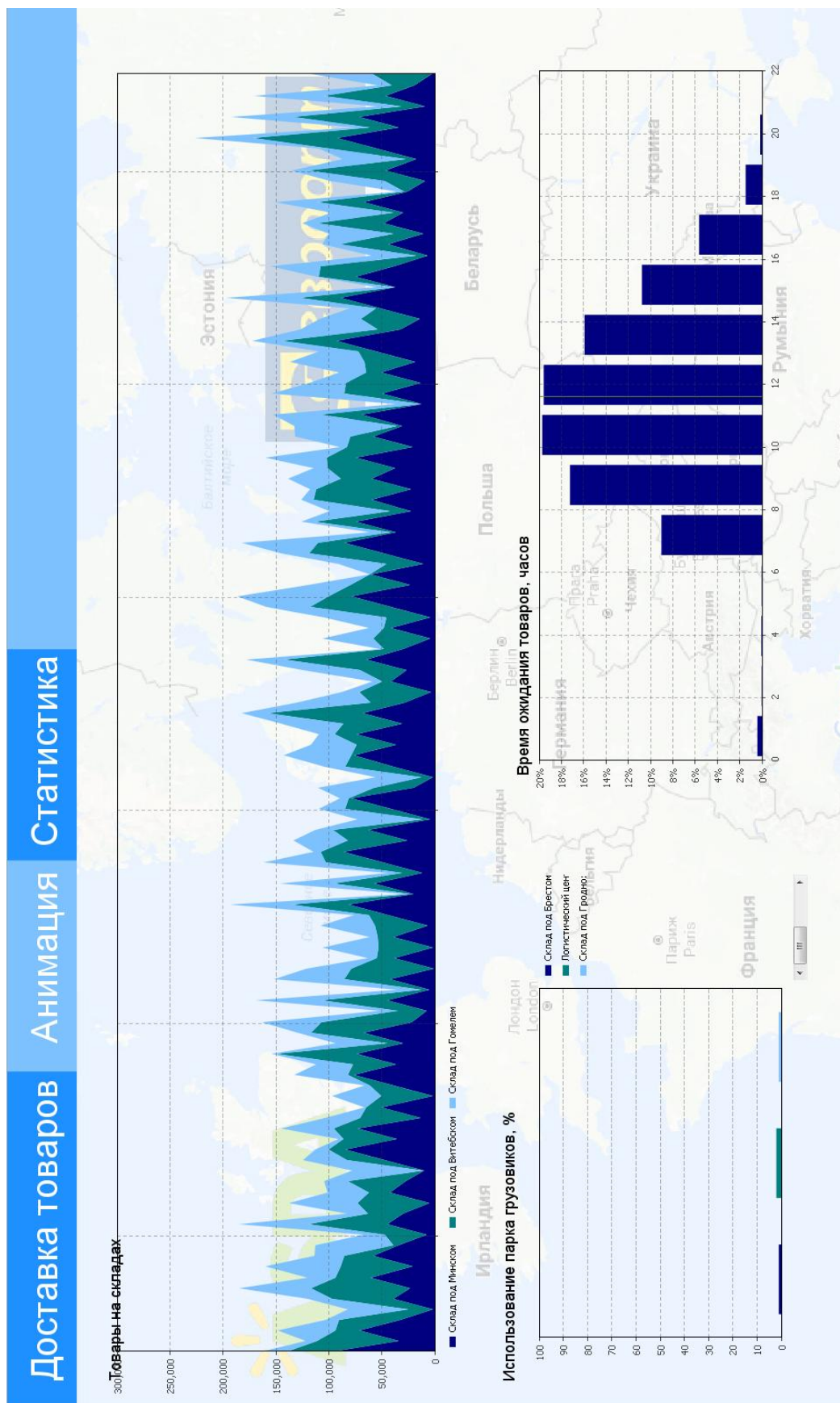


ПРИЛОЖЕНИЕ В.

Модель 1. Евроопт: фрагмент анимации

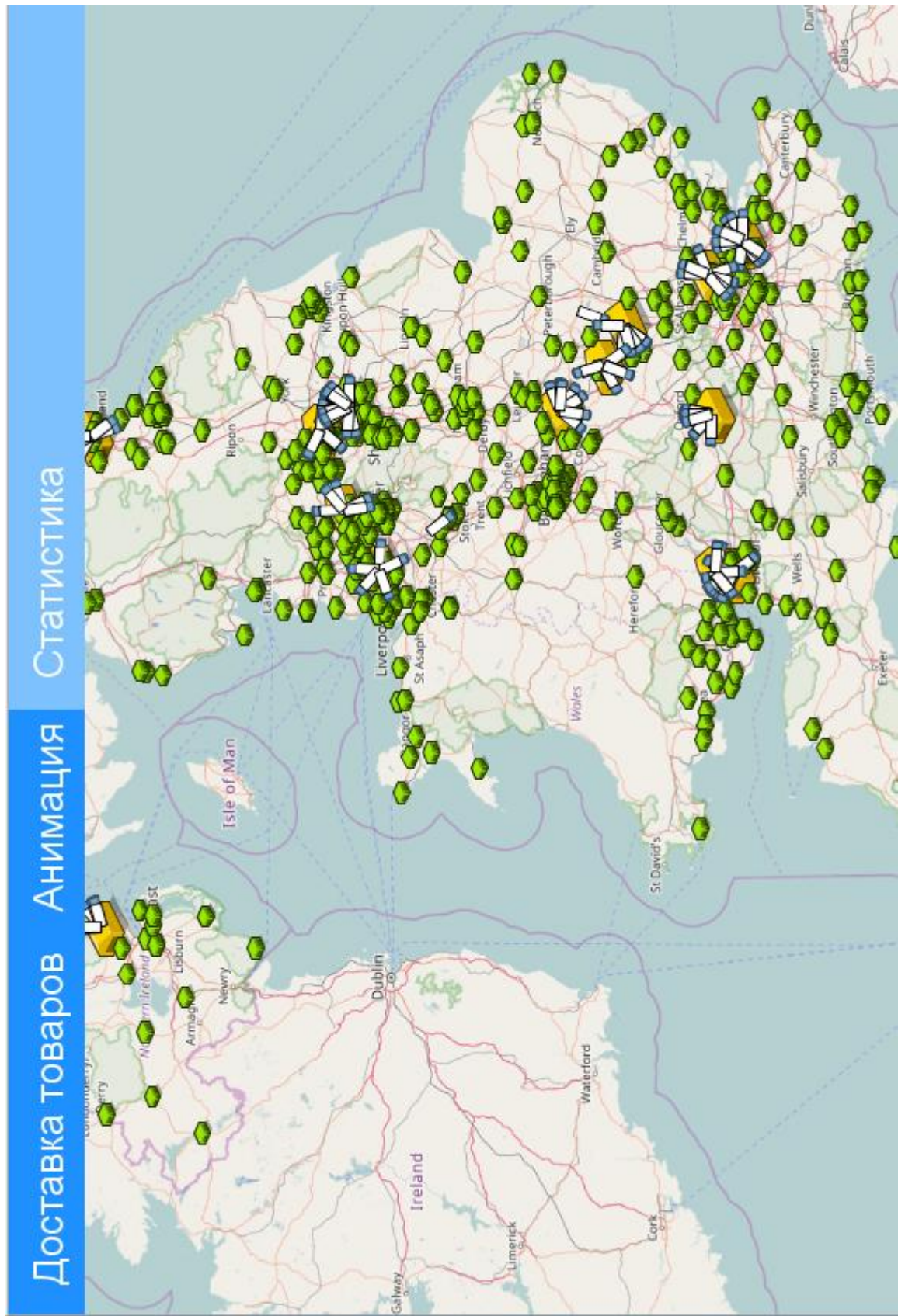


Модель 1. Евроопт: фрагмент статистики



ПРИЛОЖЕНИЕ Д.

Модель 2. ASDA: фрагмент анимации



Модель 2. ASDA: фрагмент статистики

