

## ЗАНЯТИЕ 5

**Тема: Работа с ГИС картами в AnyLogic. Моделирование процесса поставки запчастей в аэропорты. Оптимизация процесса поставки**

### 1. Постановка задачи

Имеется семь аэропортов (Пулково, Шереметьево, Домодедово, Внуково, аэропорт г. Ярославля, аэродром Смоленск-Северный, аэропорт г. Вологда), которым два раза в неделю требуются запасные части для технического обслуживания и ремонта самолетов. Предприятие в Нижнем Новгороде осуществляет производство и доставку необходимых компонентов.

Заявки от аэропортов приходят на предприятие в форме заказа. После чего на погрузку фуры требуется от двух до трех часов. Столько же времени требуется и на разгрузку фуры в аэропорту. После получения запчастей аэропорт оповещает об этом предприятие сообщением “Доставлено!”, далее фура отправляется обратно на предприятие.

**Задача:** смоделировать процесс доставки запасных частей для оценки оптимального количества транспортных средств, учитывая, что всего на предприятии имеется пять грузовиков.

### 2. Этапы для построения модели в AnyLogic

Для решения необходимо выполнить следующие действия:

1. Задать местоположение и маршруты до заданных пунктов.
2. Описать процесс оформления заказа запчастей, полагая, что каждый аэропорт отправляет запрос одинаковой формы.
3. Описать логику обработки заявки предприятием, где необходимо учесть: получение заявки, время на погрузку фуры, отправку до клиента, разгрузку фуры, оповещение о доставке и возврат грузовика на предприятие.
4. Провести оптимизацию, с целью установления необходимого количества грузовиков для предприятия, чтобы загруженность при доставке запчастей составляла не более 85%.

### 3. Создание модели:

1. Выполните команду **Файл/Создать/Модель** на панели инструментов.
2. В поле **Имя модели** диалогового окна **Новая модель** введите Dostavka.

**Выберите каталог**, в котором будут сохранены файлы модели.

Щёлкните кнопку **Готово**.

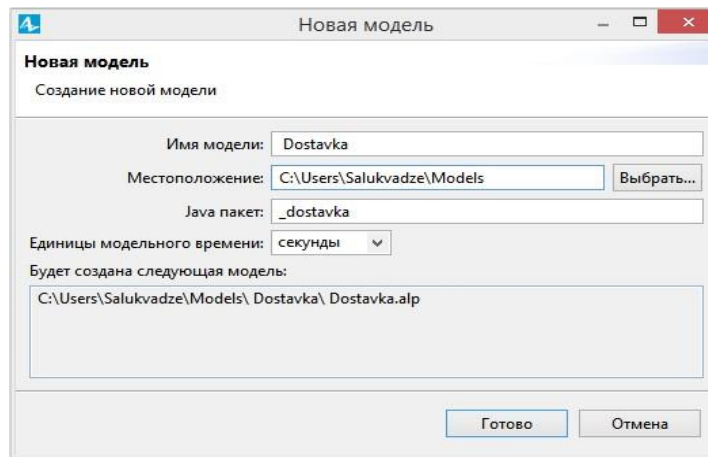


Рис. 1. Создание новой модели.

После нажатия на кнопку **Готово**, открывается окно редактирования агента **Main**, в котором и начинается моделирование заданного процесса.

### 3.1. Определение исходных данных

Исходными данными являются координаты аэропортов, координаты предприятия, производящего и доставляющего запасные части для аэропортов, и пути сообщения между ними, по которым осуществляется доставка. Так как речь в задаче идет о реально существующих объектах, удобно использовать компонент **ГИС Карта** из палитры **Разметка пространства**. По умолчанию карта и масштаб будут выбраны, как на рисунке 2:



Рис. 2. Отображение компонента ГИС Карта по умолчанию.

Рассмотрите свойства компонента **ГИС Карта**. На вкладке **Тайлы** можно выбрать поставщика карты, что изменит внешний вид объекта; на вкладке **Маршруты** задаются параметры маршрутов (по аналогии с автомобильными

навигаторами): можно выбрать источник маршрутов (с сервера OSM или из файла PBF), сервер с которого маршруты загружаются, критерий выбора маршрута (быстрейший, кратчайший), сеть дорог (автомобильная, железнодорожная, велосипедная, пешеходная). Укажите свойства объекта **ГИС Карта** согласно рисунку 3.

map - ГИС Карта

Имя:   Отображается на верхнем уровне

Блокировать

Видимость:  да

Тайлы

Отображать тайлы

Поставщик тайлов:

Маршруты

Маршруты:  Загружаются с сервера OSM  
 Считываются из файла PBF  
 Строятся по прямой

Сервер маршрутов:

Выбирается маршрут:

Сеть дорог:

Если маршрут не найден:  Строить маршрут по прямой  
 Отображать сообщение об ошибке

Рис. 3. Свойства компонента ГИС Карта.

Теперь необходимо перейти в режим редактирования карты для выбора конкретной области отображения (в контексте данной задачи – европейской части Российской Федерации). Для перехода в режим редактирования выполните двойной щелчок левой клавишей мыши. Изменение масштаба карты осуществляется колесиком мыши, перемещение по карте – с зажатой левой клавишей мыши.

Выбрав область отображения карты, перейдите к ее разметке, для этого в режиме редактирования воспользуйтесь встроенной функцией поиска: в строку поиска введите названия искомого объекта и нажмите Enter. Ниже отобразится область с результатами поиска, выделите нужный вариант правой клавишей мыши и выберите **Преобразовать в ГИС точки\регионы**, на карте появится метка выбранного объекта. На рисунке 4 показан процесс поиска, а на рисунке 5 показана карта с метками всех объектов из текста задачи.

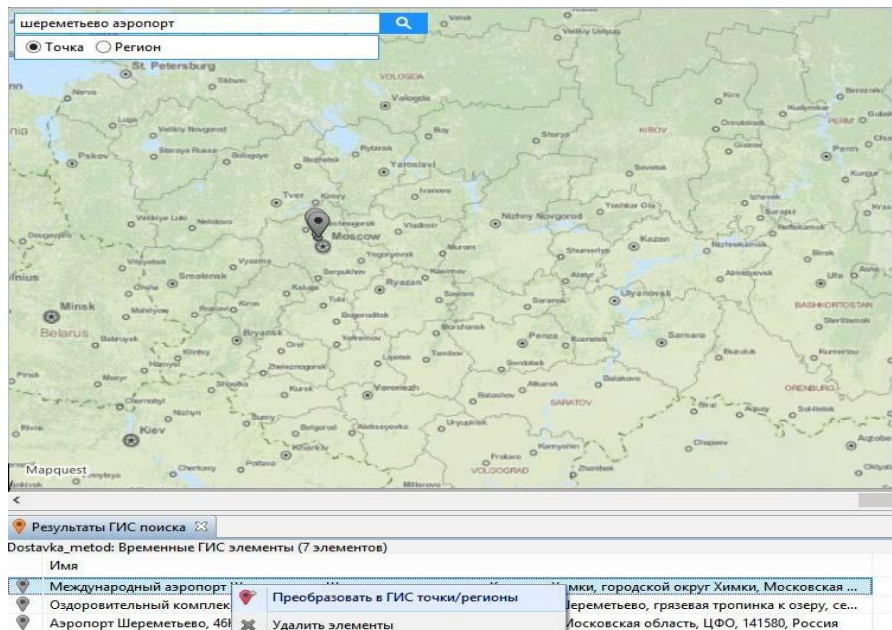


Рис. 4. Процесс поиска по названию объекта.

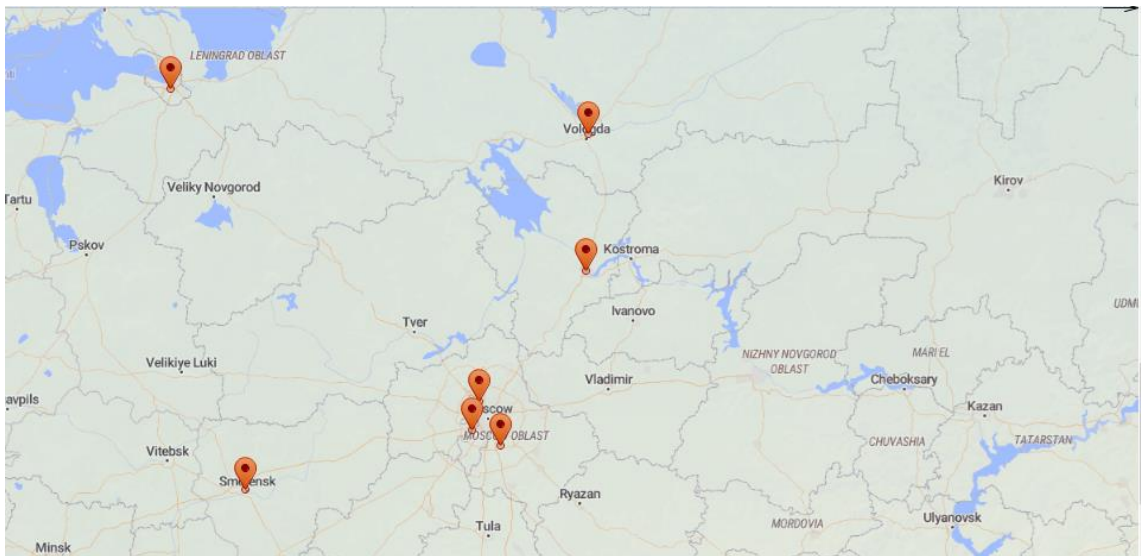


Рис. 5. Карта с метками всех объектов из текста задачи.

Для удобства размещения **Агентов** в отмеченных точках, объедините их в коллекцию. Для этого выделите все метки аэропортов щелкните правой клавишей мыши на любой из меток и выберите пункт **Создать коллекцию**. В свойствах коллекции задайте имя **airportLocation**. Как понятно из названия в этой коллекции будут храниться координаты аэропортов.

В эти координаты необходимо поместить аэропорты. Для этого создайте популяцию агентов, которые и будут моделями аэропортов. Из палитры **Агент** перетащите в область редактирования компонент **Агент**. Открывается окно создания агентов. На рисунках 6-10 показан процесс создания новой популяции агентов:

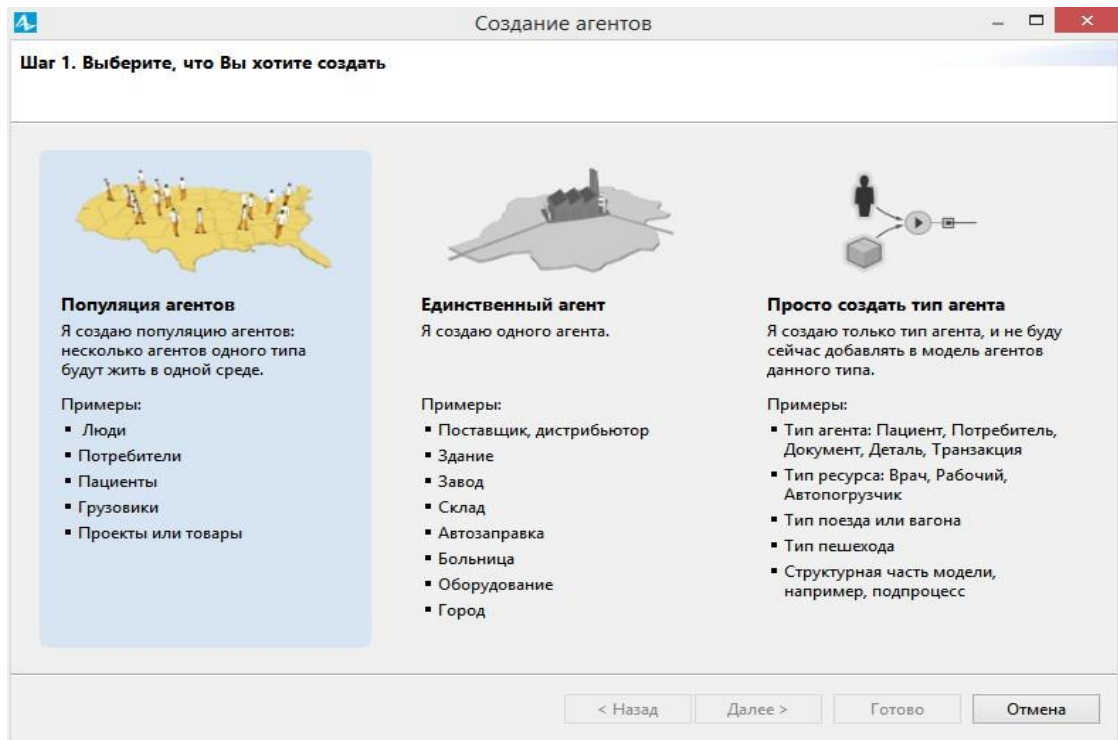


Рис. 6. Создание популяции агентов.

После каждого этапа выбираем пункт **Далее**.

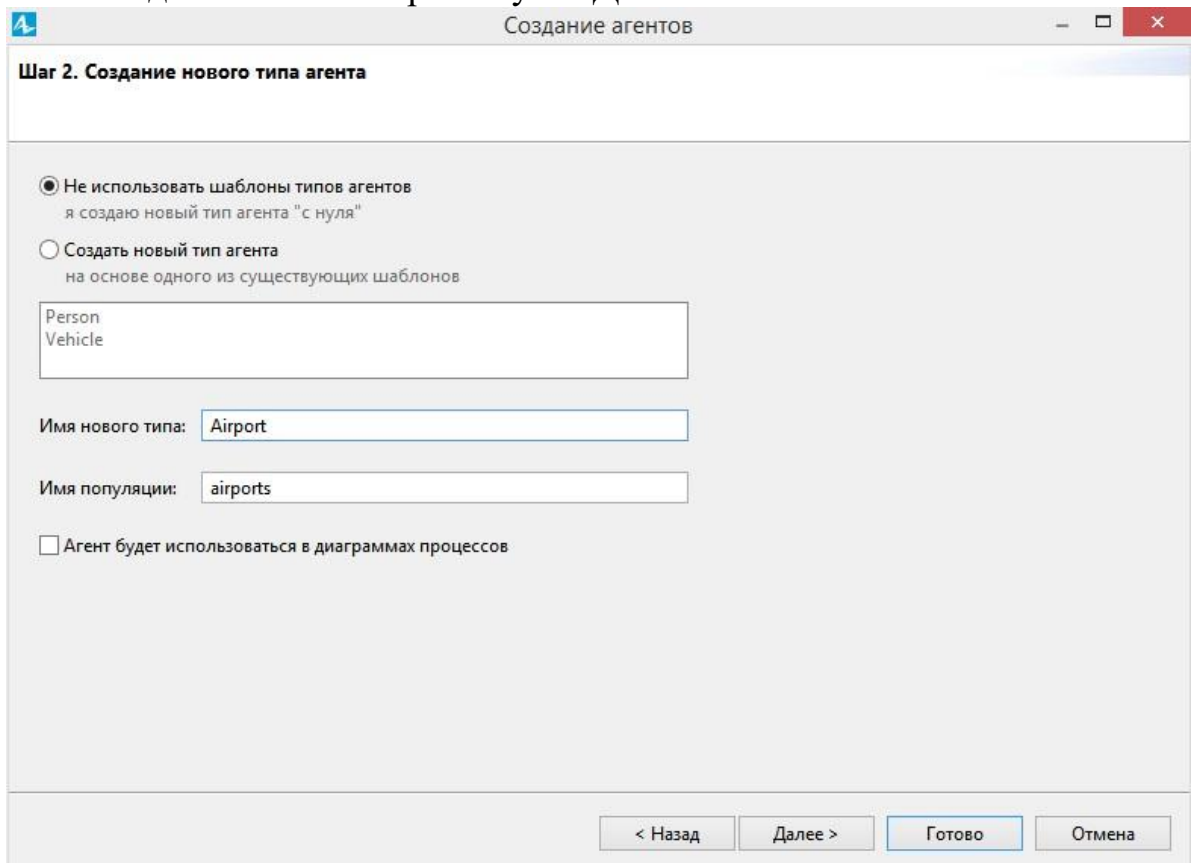


Рис. 7. Задание имени новой популяции агентов.

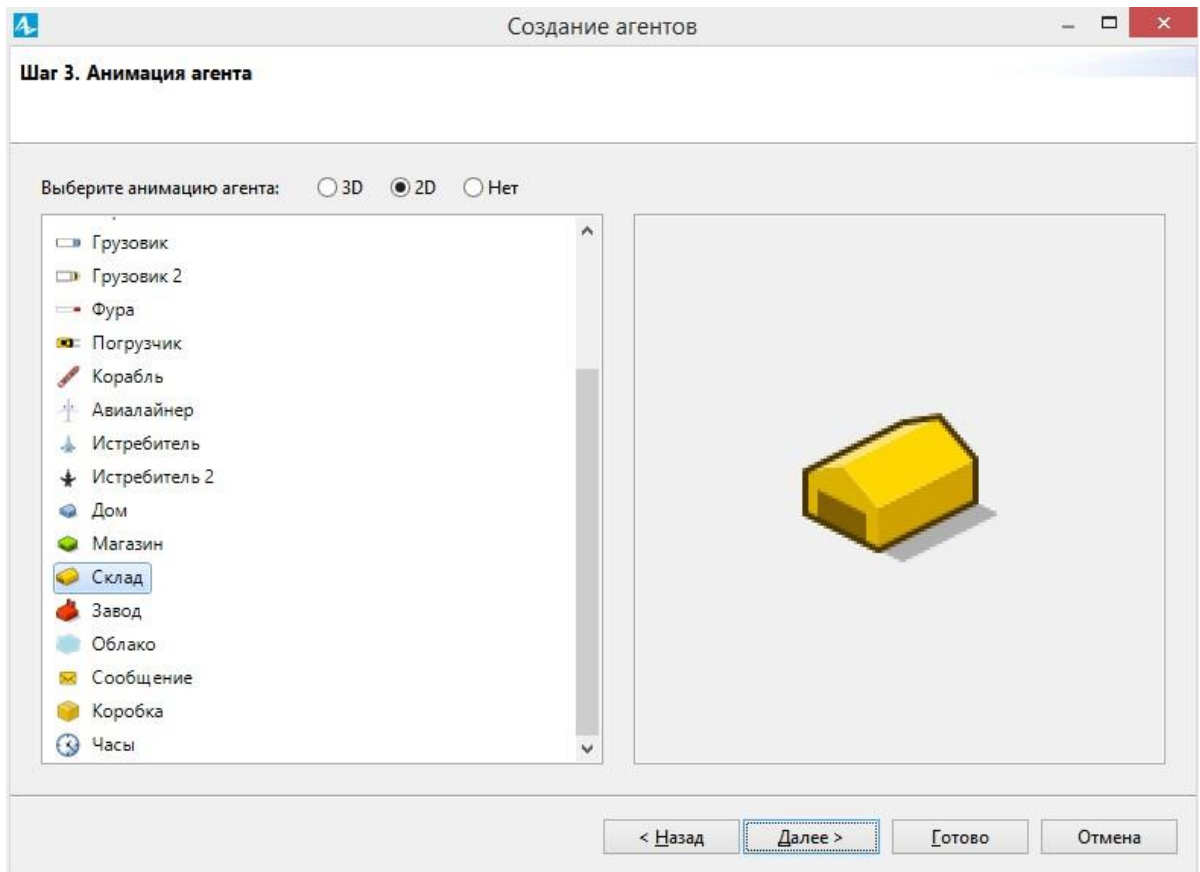


Рис. 8. Выбор анимации для агента.

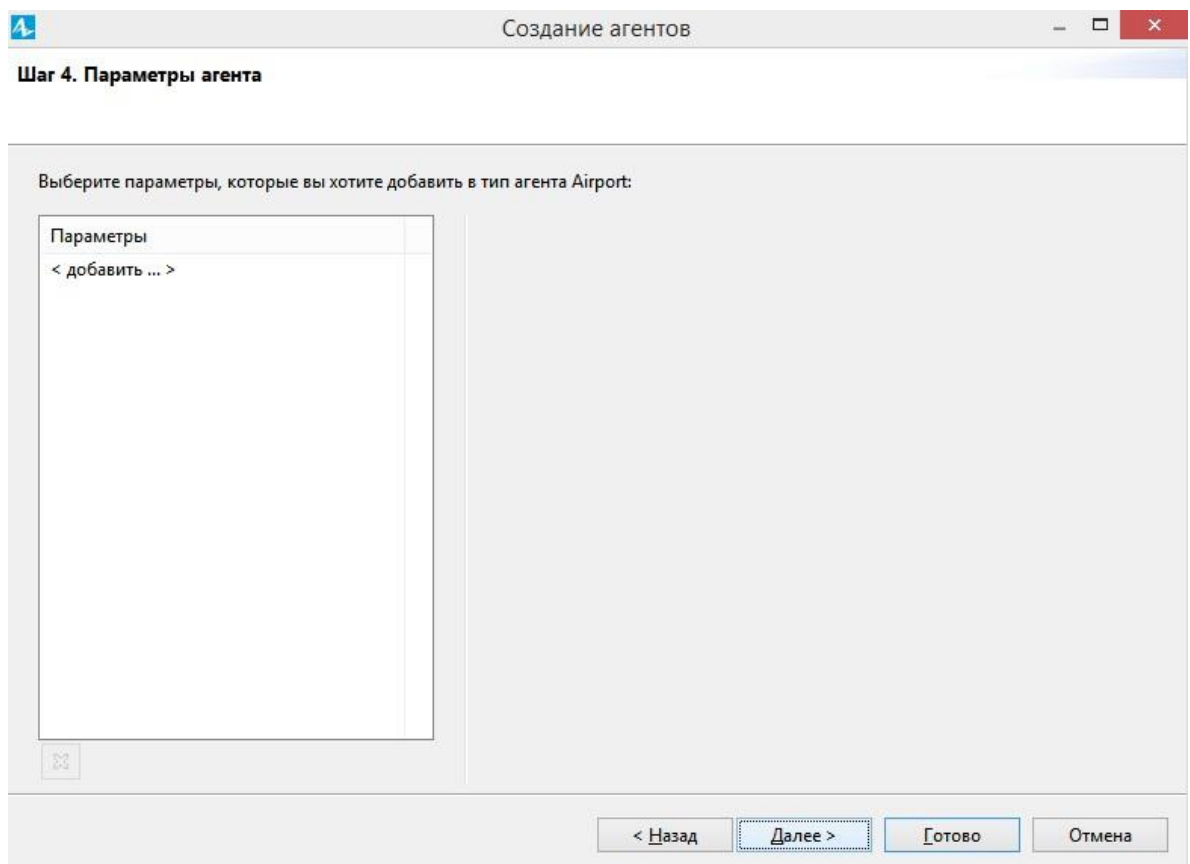


Рис. 9. Добавление параметров для агента (в данном случае остается пустым)

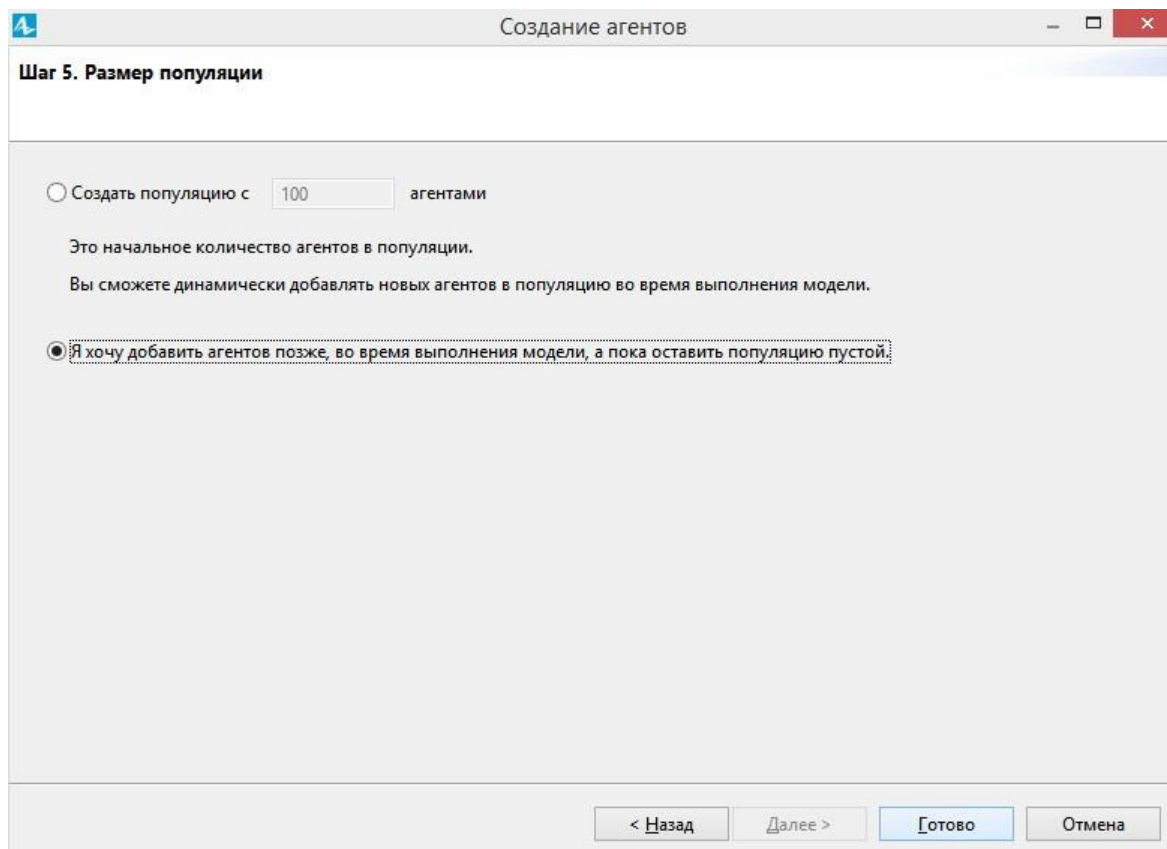


Рис. 10. Выбор количества популяции агентов.

После нажатие кнопки **Готово** на объекте **main** появится иконка созданного агента (выбранная на шаге 3, рис. 8).

Теперь необходимо связать созданную популяцию агента **Airport** с координатами аэропортов и с количеством аэропортов, то есть связать с коллекцией **airportLocation**. Чтоб это сделать перейдем в свойства популяции, щелкнув левой клавишей мыши популяцию **airports** на объекте **main**. В графу **Начальное количество агентов** введите следующее выражение:

`airportLocation.size()`

Данная функция возвращает количество элементов в коллекции **airportLocation** (в данном случае 7). Для связи аэропортов с их координатами, в пункте **Начальное местоположение** выберите

**Расположить агентов:** в узле и в графу **Узел** введите выражение:

`airportLocation.get(index)`

Данная функция возвращает адреса (индексы) элементов в коллекции **airportLocation** (тех самых меток на карте). Свойства популяции **airports** представлены на рисунке 11:

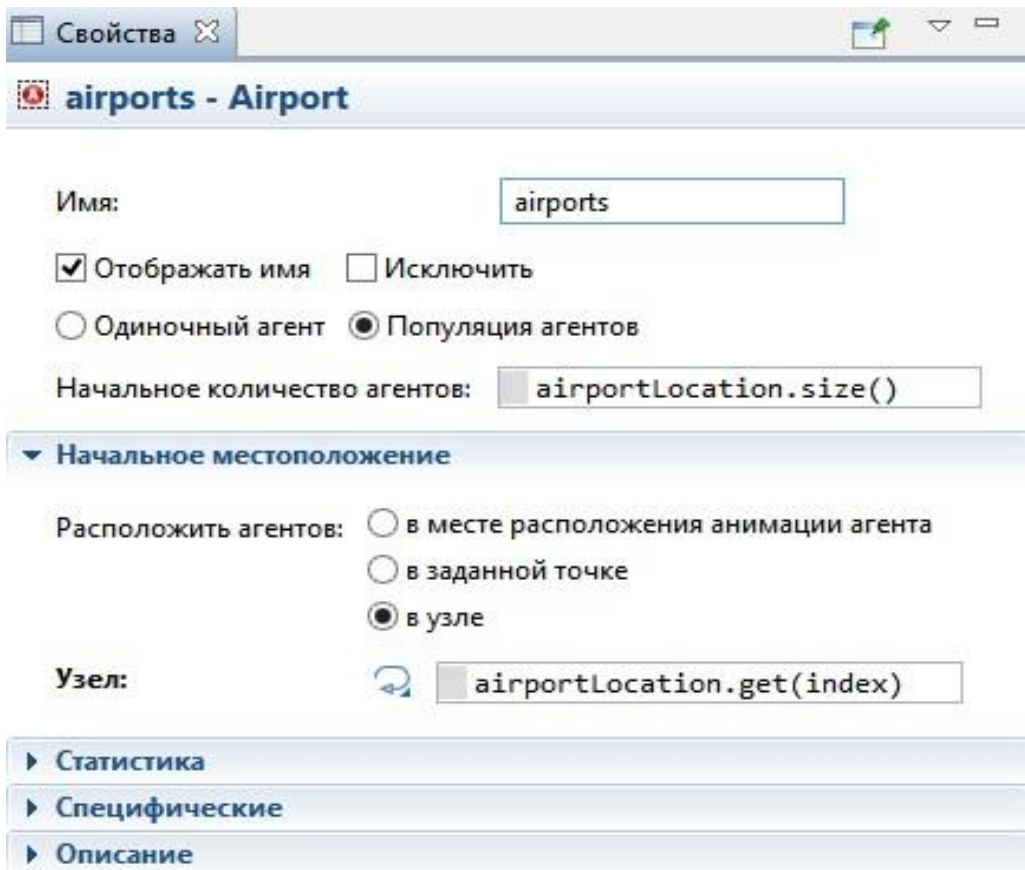


Рис. 11. Свойства популяции агента **Airport**

Запустите модель, нажав F5, чтобы проверить внесенные нами данные. Если все было сделано правильно, то на экране вы увидите следующее:

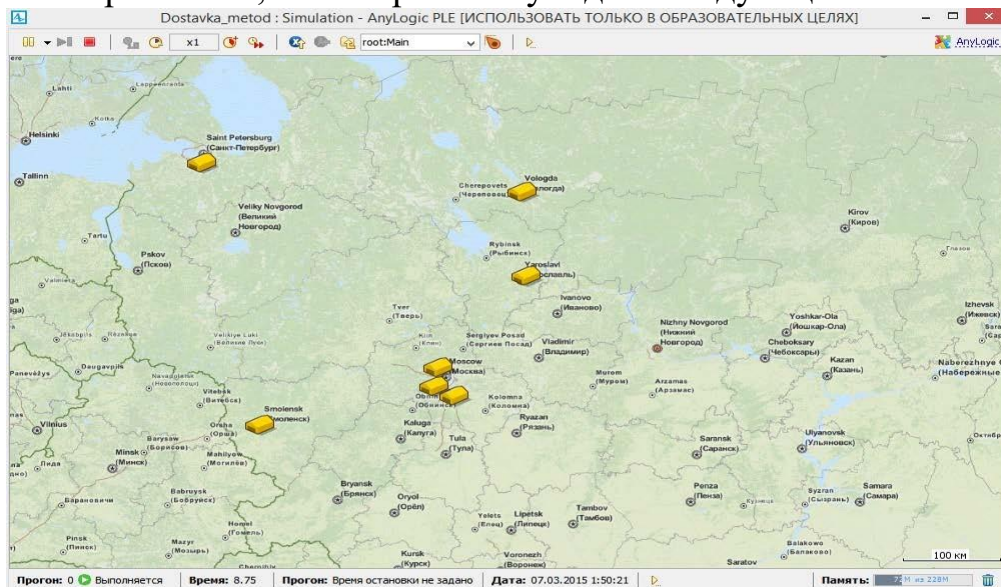


Рис. 12. Выполнение модели с отмеченными аэропортами.

Как видно из рисунка 12 агенты типа **Airport** находятся в координатах аэропортов и их ровно семь штук.

Теперь создайте тип агента с именем **Manufacturing** для размещения предприятия в Нижнем Новгороде. Создается агент аналогичным образом, как показано на рисунках 6-10, с небольшими отличиями:

1. Вместо **Популяции агентов** выбрать **Единственный агент**;
2. Создать новый тип агента;
3. Имя нового типа: **Manufacturing**;
4. Анимацию выбирать: 2D/Завод; 5. Поле параметров оставить пустым.

На объекте **main** появляется иконка нового агента (завод). Теперь необходимо связать агент **manufacturing** с меткой на карте. Откройте свойства агента **manufacturing** и в пункте **Начальное местоположение** выберите **Расположить агентов: в узле** и в графе **Узел** вызовите выпадающий список, в котором выберите метку Нижнего Новгорода, как на рисунке ниже:

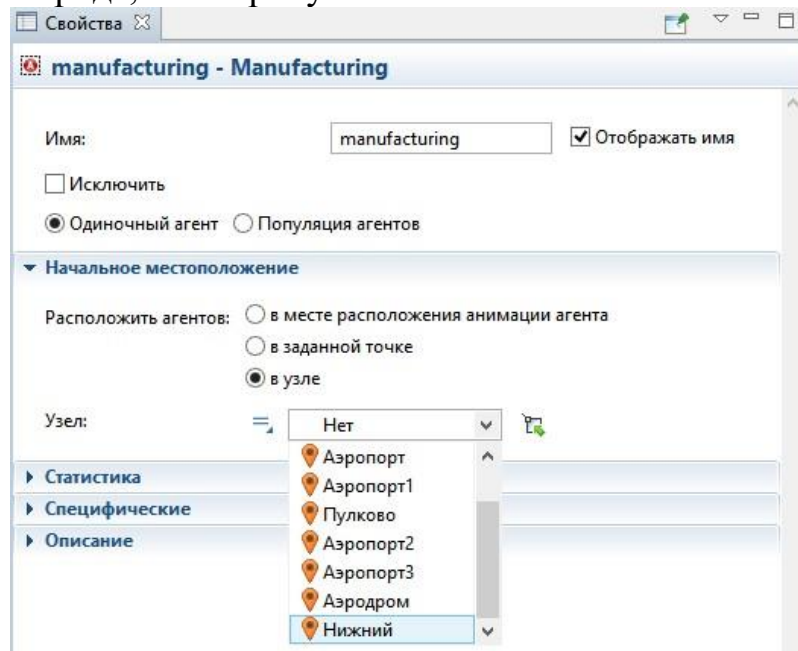


Рис. 13. Свойства агента manufacturing

Далее проверьте, что получается при исполнении модели:

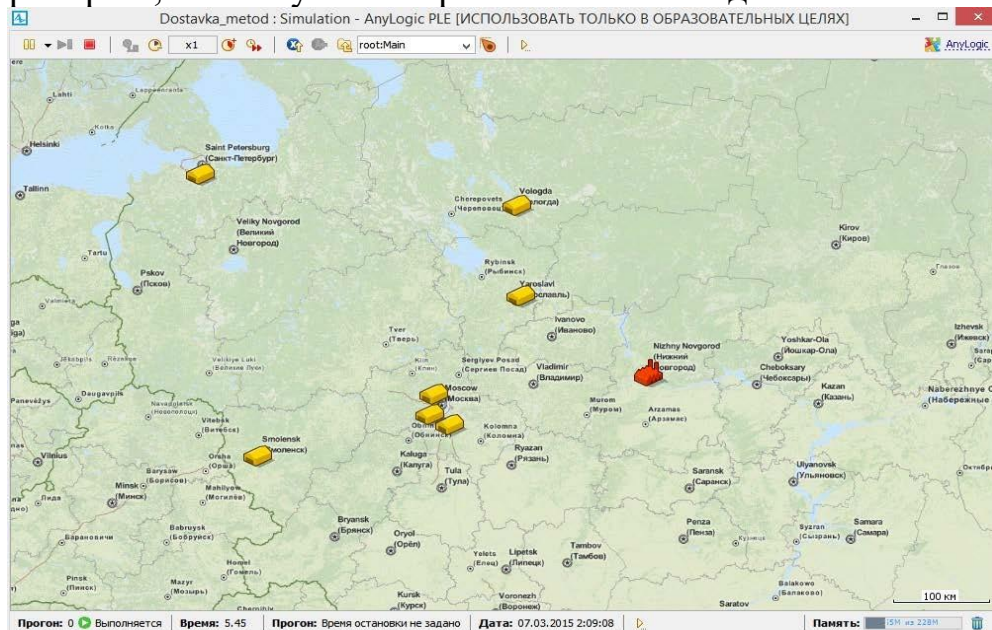


Рис. 14. Выполнение модели с отмеченным производством.

Из текста задачи известно, что доставка осуществляется грузовиками, чтобы внести их в модель, необходимо создать новую популяцию агентов. Делается это аналогично, как на рисунках 6-10, с некоторыми изменениями:

1. Создать новый тип агента;
2. Имя нового типа: **Truck**;
3. Выбрать анимацию: 2D/Грузовик;
4. Указать параметры согласно рисунку 15;

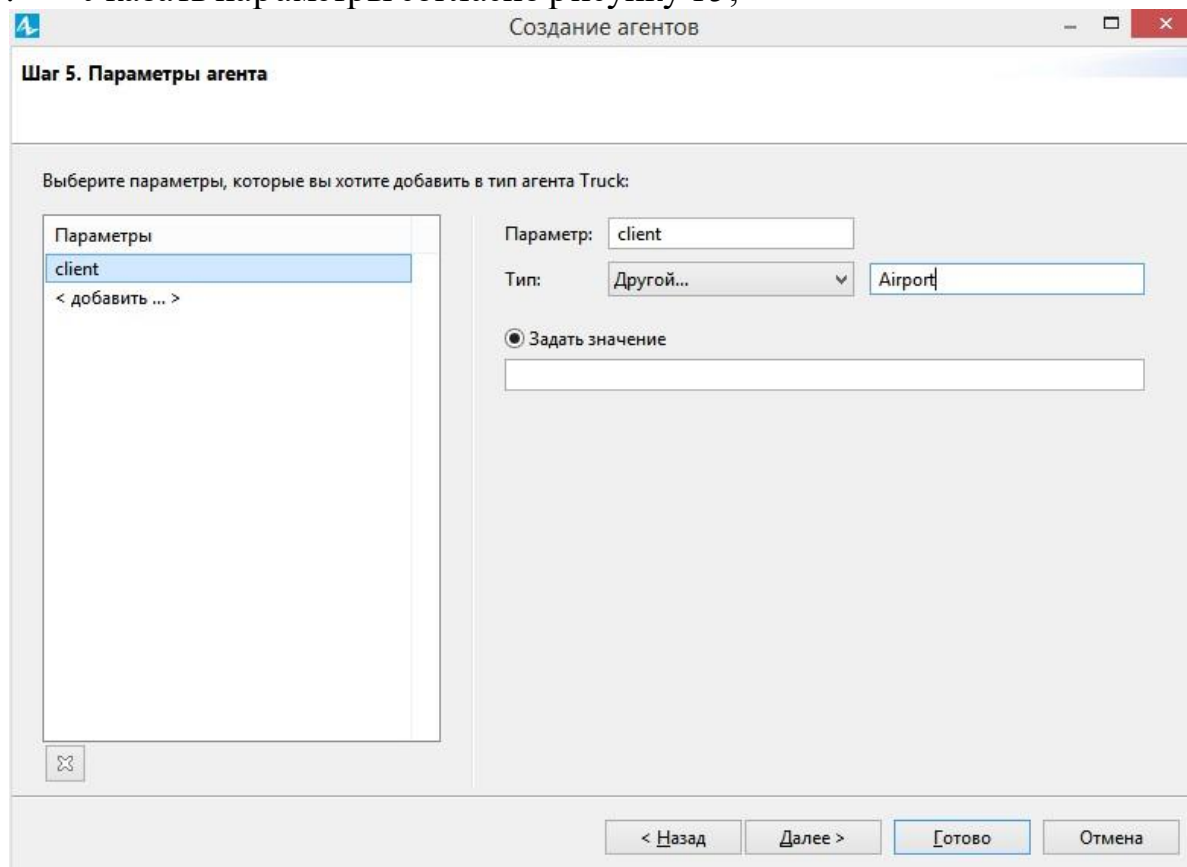


Рис. 15. Параметры типа агента Truck.

Для дальнейшей оптимизации, вынесите количество грузовиков как отдельный параметр. Для этого нужно вытащить из палитры **Агент** компонент **Параметр**. Установите следующие свойства:

- Имя: `numberTrucks`
- Тип: `int`
- Значение по умолчанию: 5

Понятно, что все грузовики принадлежат производству (агент **manufacturing**), необходимо в свойствах популяции агента **trucks** указать начальное местоположение, как на рисунке 16:

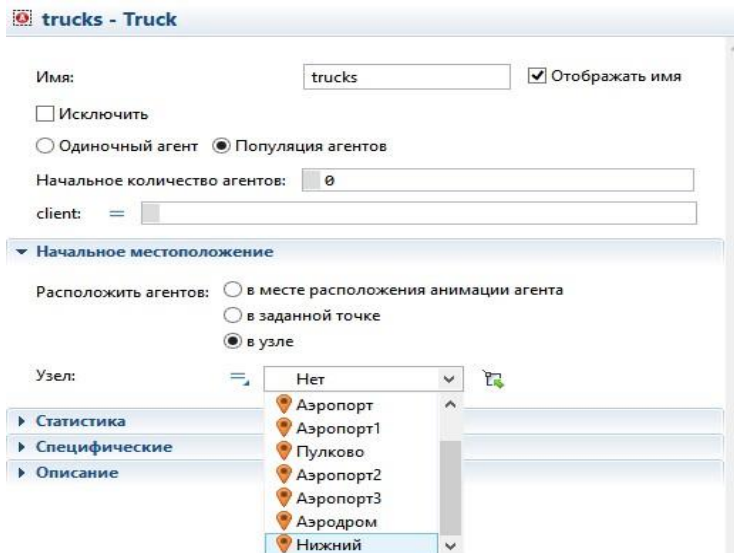


Рис. 16. Свойства популяции агента trucks.

Из условия задачи известно, что для получения запасных частей аэропорты должны сформировать заказ. Чтобы отразить это в модели создайте новый агент, как на рисунках 6-10, со следующими изменениями:

1. Вместо **Популяции агентов** выбрать **Просто создать тип агента**;
2. Имя нового типа: **Order**;
3. В пункте **Анимация** выбрать **Нет**;
4. Указать параметры согласно рисунку 17;

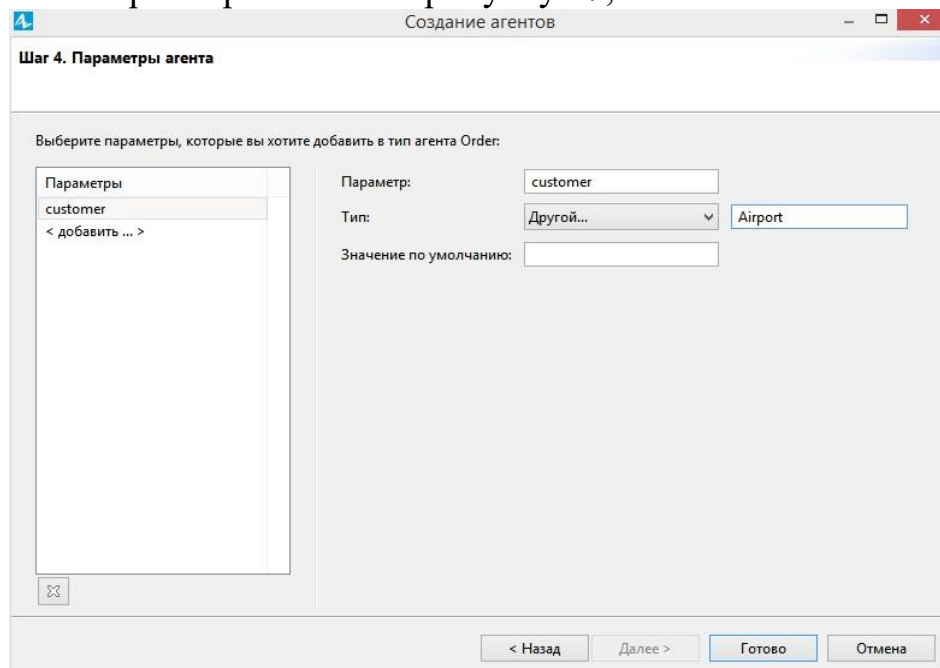


Рис. 17. Параметры типа агента Order.

Теперь необходимо показать, что грузовики являются ресурсом производства. Для этого из вкладки **Проекты** (или из объекта **main**) откройте тип агента **Manufacturing** и перенесите на него из палитры **Библиотека моделирования процессов** блок **Resource Pool**, который задает набор доступных ресурсов, в данном случае – грузовиков. Далее необходимо задать параметры блока **Resource Pool**, согласно рисунку 18:

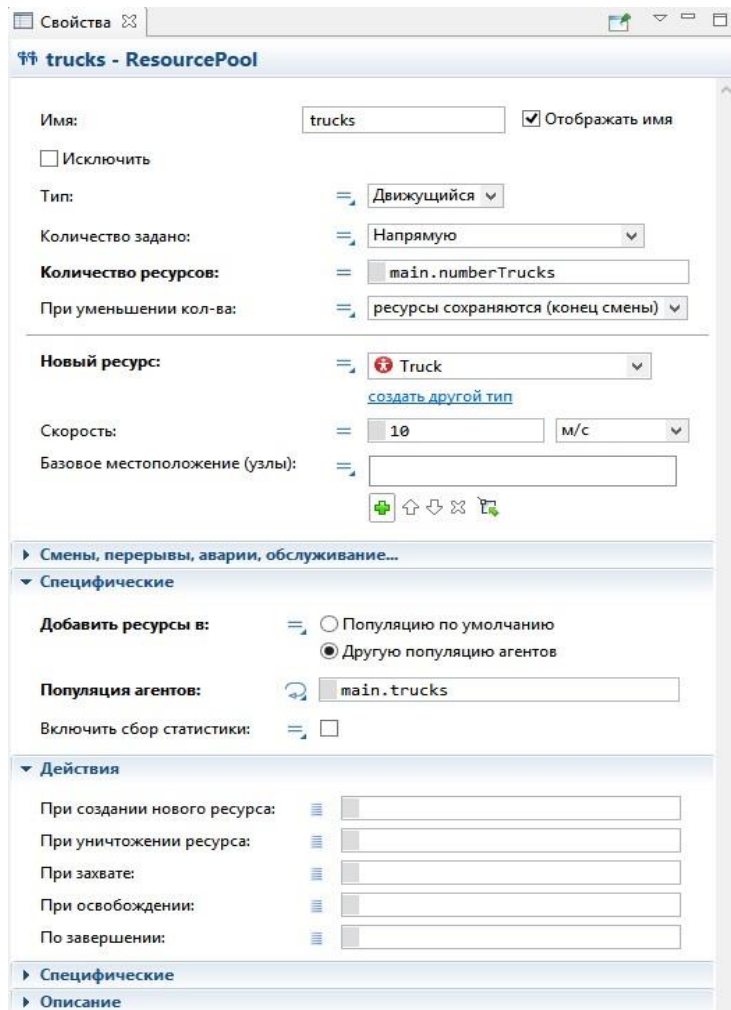


Рис. 18. Свойства блока Resource Pool

На рисунке 18 видно, что значение **Количество ресурсов** соответствует значению созданного ранее параметра **numberTrucks**, а так же необходимо отметить, что этот блок добавляет к новым ресурсам популяцию **trucks** (Популяция агентов: **main.trucks**).

Запустите модель для проверки внесенных изменений. Если все верно модель будет соответствовать приведенной на рисунке 19:

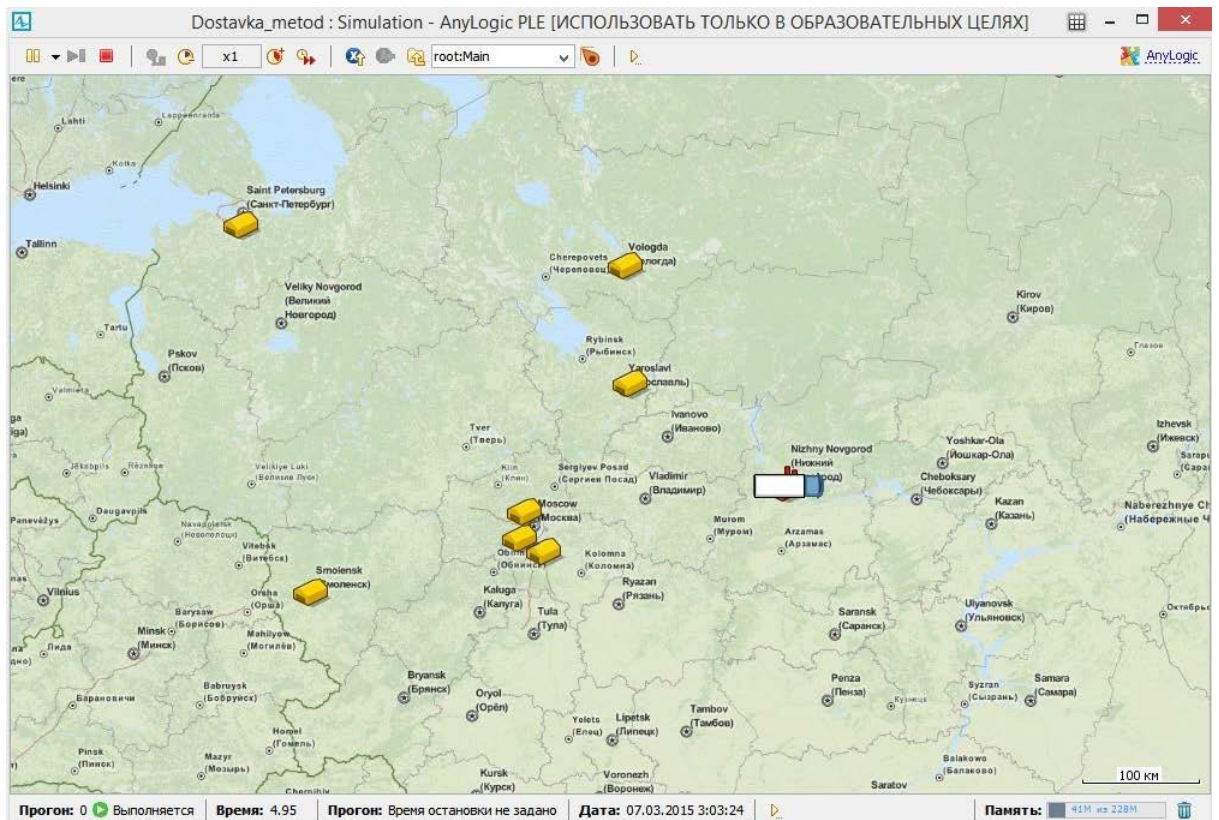


Рис. 19. Выполнение модели с грузовиками на предприятии

На этом ввод исходных данных можно считать оконченным. В качестве исходных данных описаны: аэропорты (их координаты), производство (координаты), грузовики (как ресурс производства), создана заявка для заказа запасных частей (как тип агента Order).

Теперь перейдем к логике работы аэропортов и производства.

### 3.2. Формирование заказа. Поведение агента Airport

Логика работы аэропортов, в данной задаче, можно рассматривать как последовательные переходы из состояния нормальной работы в состояние ожидания запасных частей и обратно. Чтобы смоделировать данный процесс необходимо воспользоваться палитрой **Диаграмма состояний**.

Перейдите в режим редактирования агента **Airport**. Далее нужно открыть палитру **Диаграмма состояний**. Любая диаграмма состояний должна начинаться с элемента **Начало диаграммы состояний** – он является точкой входа в диаграмму состояний (особенно это важно для многоуровневых диаграмм). Его нужно соединить с элементом **Состояние**, который будет отвечать за режим нормальной работы аэропортов (тот период времени, когда запасные части не нужны). Укажите имя **normalWork**, в свойства изменения вносить не нужно. Как было сказано выше работа аэропортов еще включает в себя режим ожидания запчастей, поэтому необходимо вынести еще один элемент **Состояние** в рабочую область. Назовите это состояние **waitingDetails**. К свойствам данного элемента вернемся позже.

Как сказано в тексте задачи, детали требуются аэропортам 2 раза в неделю – это значит, что переход из состояния нормальной работы в состояние ожидания запчастей происходит с заданной интенсивностью, чтобы отметить это в диаграмме

состояний перетащите из палитры **Диаграмма состояний** объект **Переход**, в свойствах которого укажите следующие параметры:

- Происходит: С заданной интенсивностью;
- Интенсивность: 2 раза в неделю;

Как только аэропорт переходит в состояние ожидания запчастей, должен быть сформирован и отправлен на предприятие заказ. Чтобы отразить это в диаграмме состояний вернемся к свойствам элемента **waitingDetails**. В графе **Действие при входе** напишите следующее выражение:

```
Order order = new Order ( this );  
send ( order , main.manufacturing );
```

В первой строчке формируется новый заказ с параметром **this**, который указывает на аэропорт, с которого заказ поступает. Вторая строчка кода нужна для отправки – функция **send**, которая имеет два аргумента: что отправить (созданный выше заказ) и кому отправить (на производство – агенту **manufacturing**).

Чтобы диаграмма состояний полностью отражала логику работы аэропорта, в контексте данной задачи, необходим еще один компонент **Переход** из состояния **waitingDetails** в состояние **normalWork**. Как известно из текста задачи, переход в нормальный режим работы осуществляется при получении сообщения “Доставлено!”. Для этого свойства компонента **Переход** должны соответствовать свойствам, указанным на рисунке 20:

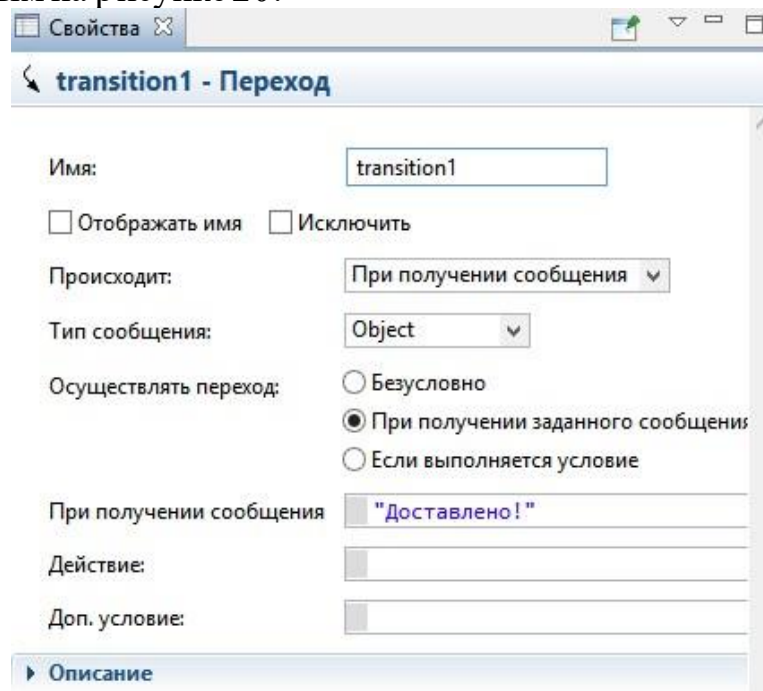


Рис. 20. Свойства элемента Переход

На рисунке 21 показана описанная выше диаграмма состояний.

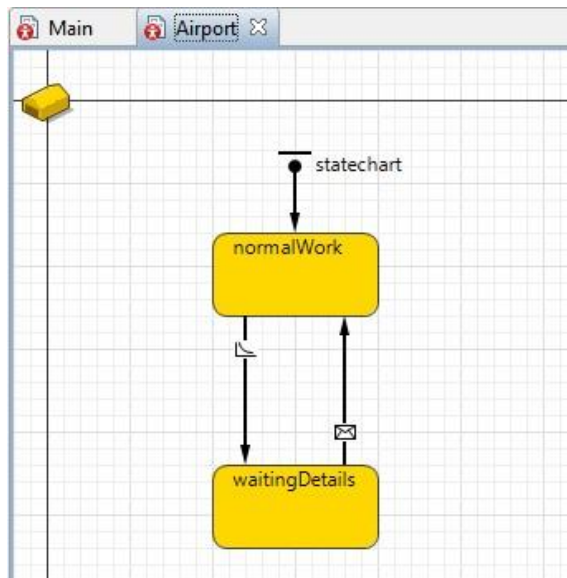


Рис. 21. Логика работы агента Airport

Теперь аэропорт может отправлять заказы на запасные части. Переходим к обработке и исполнению данного заказа.

### 3.3. Обработка заказа. Поведение агента Manufacturing

После получения заказа, на производстве выделяют ресурс (грузовик) для исполнения данного заказа. Его загружают заказанными запчастями, на что нужно от двух до трех часов, и отправляют в аэропорт. Там машину разгружают (в течение двух-трех часов), после чего посылается оповещение о доставке, и грузовик возвращается на производство, становясь свободным ресурсом.

Ниже, на рисунке 22, приведена модель данного процесса, далее каждый блок будет рассмотрен в отдельности.

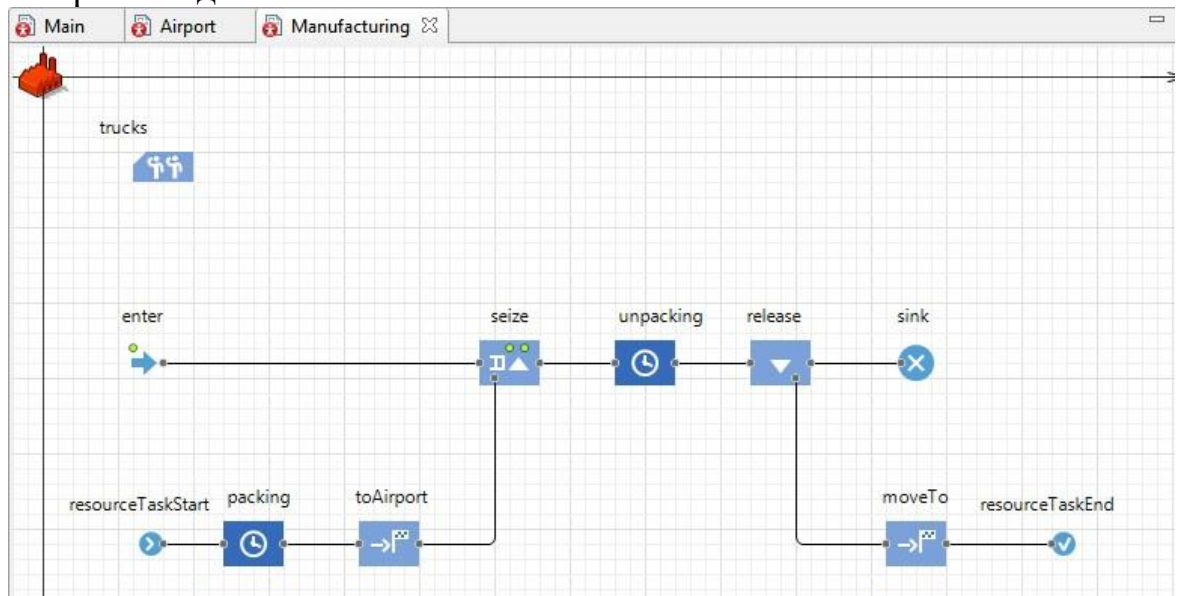


Рис. 22. Обработка и исполнение заказа на предприятии.

Чтобы реализовать показанный выше процесс, откройте редактирование агента **Manufacturing** и палитру **Библиотека моделирования процессов**. Вход в процесс осуществляется через блок **Enter**, в которые поступают заказы. Перетащите этот блок на рабочую область в свойствах задайте:

- Тип агента: **Order**;

Далее полученный заказ поступает в очередь на ожидание ресурсов, для этого перетащите блок **Seize**, который и отвечает за захват ресурсов и соедините его с блоком **Enter**. Откройте свойства блока **Seize** и в пункте

**Набор ресурсов** кликните на изображение “+” – из всплывающего списка выберите **trucks** (грузовики).

Перед тем, как ресурс будет захвачен, он должен быть подготовлен, что в нашем случае означает загрузку машины. Подготовка ресурсов и их отправка происходит в специальном подпроцессе для ресурсов, который начинается с блока **Resource Task Start**. Его необходимо перетащить из палитры **Библиотека моделирования процессов** и расположить ниже, как показано на рисунке 22. В свойствах данного блока необходимо указать:

- Тип ресурса: **Truck**;
- Начинать задачу: всем ресурсам;

На загрузку ресурса требуется время. Перетащите блок **Delay**, назовите его **packing** и соедините с блоком **Resource Task Start**. Свойства блока **Delay**, приведены на рисунке 23:

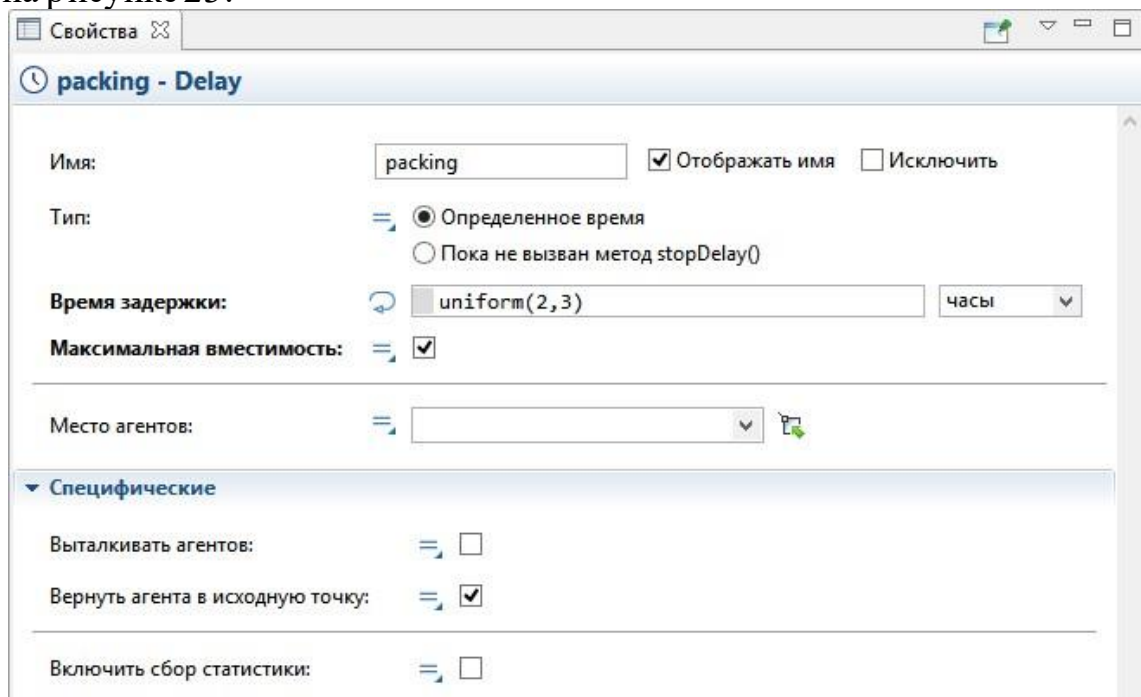


Рис. 23. Свойства блока Delay (packing)

Для времени задержки указан равномерный закон распределения между двумя и тремя часами:  $uniform(2, 3)$ .

После того, как ресурс загружен, его отправляют в аэропорт. Перетащите элемент **Move To**, переименуйте его в **toAirport** и соедините, как показано на рисунке 22. В свойствах необходимо указать пункт назначения, как показано на рисунке 24:

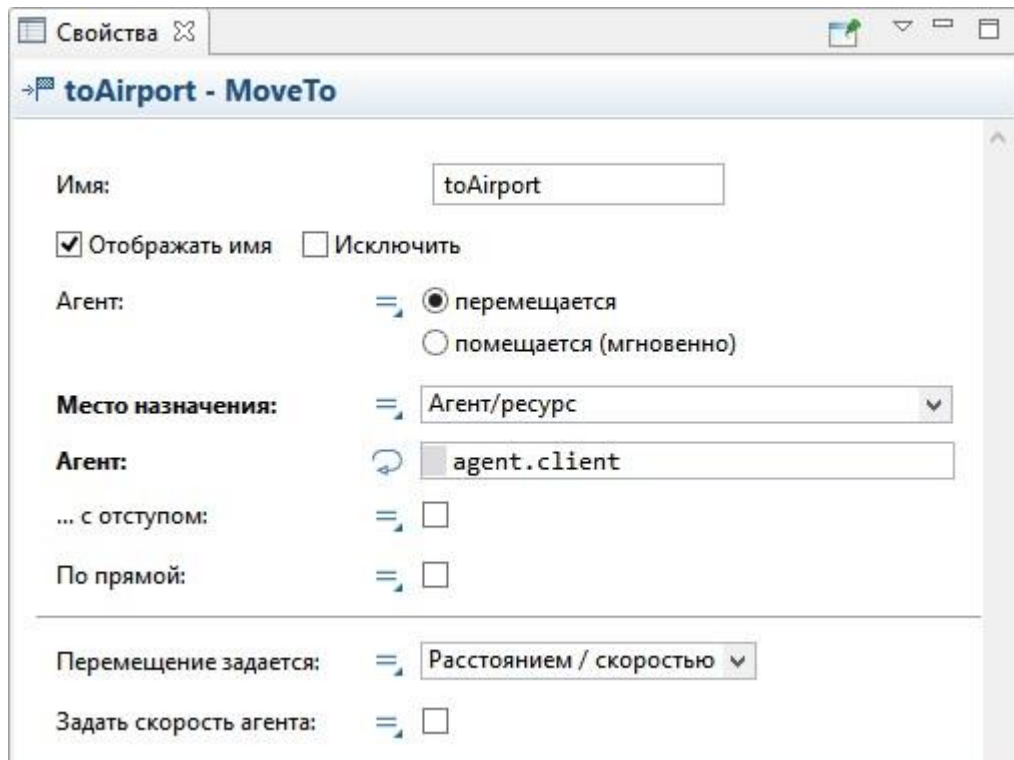


Рис. 24. Свойства блока Move To (toAirport)

Ресурс (грузовик) отправлен в блок **Seize**, теперь этому ресурсу должна быть передана информация внутри заказа, чтобы он знал, куда именно должен быть доставлен груз. Для этого перейдите в свойства объекта **Seize** и

в пункте **Действия** в графу **При захвате ресурса** напишите следующее выражение:

$$((Truck) unit). client = agent. customer ;$$

В данном выражении значению параметра **client**, который хранится внутри ресурса **Truck**, присвоено значение параметра **customer** агента **Order**.

Как только грузовик приехал в аэропорт, необходимо осуществить его разгрузку. Этот процесс моделируется аналогично процессу загрузки машины, а именно блоком **Delay**, который в данном случае будет называться **unpacking**. Свойства данного блока приведены на рисунке 25:

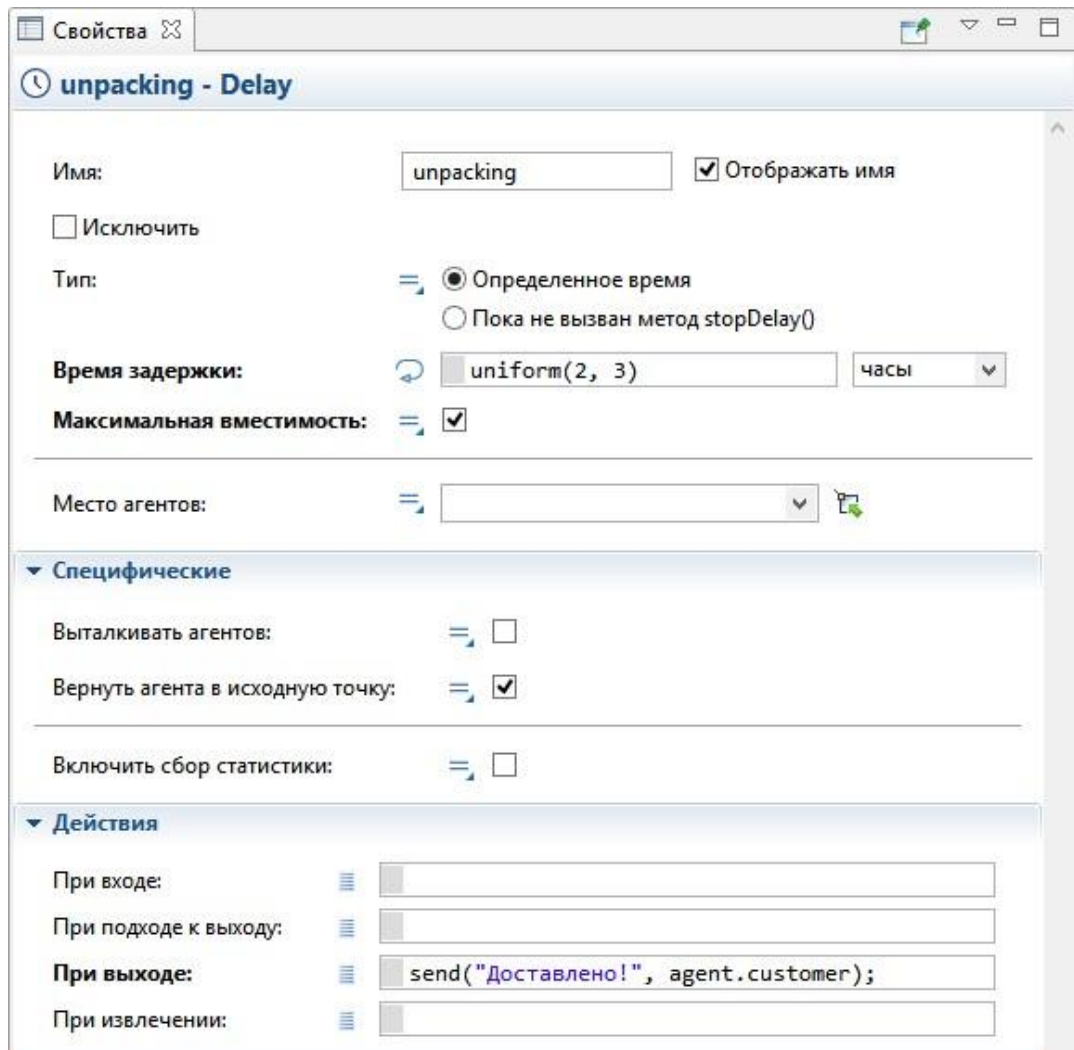


Рис. 25. Свойства блока Delay (unpacking)

При моделировании логики работы аэропорта было указано, что переход из состояния ожидания запчастей в состояние нормальной работы происходит при получении сообщения “Доставлено!”. Данное сообщение отправляется, как только закончена разгрузка, то есть происходит выход из блока **unpacking**. Для отправки использована встроенная функция **Send**, которая имеет два аргумента (что отправить, кому отправить).

После выполнения заказа ресурс становится свободным (блок **Release**, рис. 22), а агент (**Order**) отправляется в блок **Sink**, где будет удален. Так как ресурс освобожден, он должен быть отправлен на производство, для этого воспользуйтесь еще раз блоком **Move To**, в котором укажите место назначения агент **manufacturing**, который расположен на агенте **main** (см рис. 24).

Подпроцесс для ресурса должен быть закончен блоком **Resource Task End**, для того, чтобы ресурс вернулся в общий пул с ресурсами и был доступен для нового захвата.

На этом процесс обработки заказов готов, осталось только сделать так, чтобы все входящие заказы поступали на блок **Enter** обработки.

Все входящие сообщения обрабатываются в стандартном блоке **connections**, который по умолчанию существует внутри каждого агента. Откройте свойства блока **connections** агента **manufacturing** (рисунок 26):

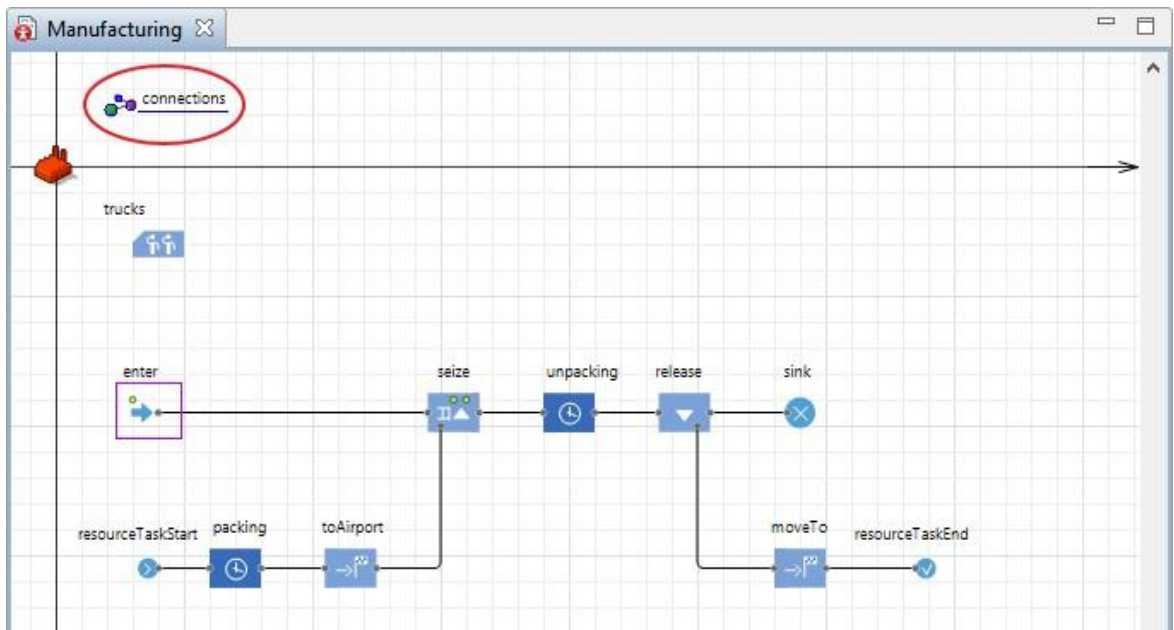


Рис. 26. Блок connections

В свойствах данного блока откройте пункт **Взаимодействие** и измените его поля согласно рисунку 27:

**▼ Взаимодействие**

Эти действия выполняются для сообщений, поступающих во все связи агента

Тип сообщения:

Действие при получении сообщения:

`enter.take( msg );`

Перенаправлять сообщение в:

Диаграммы состояний	

Рис. 27. Пункт Взаимодействие свойств connections

В поле **Действие при получении сообщения** введена команда, которая указывает, что блок **Enter** должен взять на обработку (функцией **take**) пришедшее сообщение (используя локальную переменную **msg**). Тип сообщения указывается согласно созданному ранее в агенте **manufacturing**.

Модель процесса доставки запасных частей на аэропорты готова и теперь можно перейти к запуску и оптимизации созданной модели.

### 3.4. Запуск и оптимизация модели

Перед тем, как запустить модель, необходимо изменить единицы модельного времени. Для этого откройте вкладку **Проекты**, выделите проект **Dostavka** и в пункте свойств **Единицы модельного времени** укажите **дни**. Запустить модель на исполнение, нажав **F5**. Результат исполнения модели указан на рисунке 28:

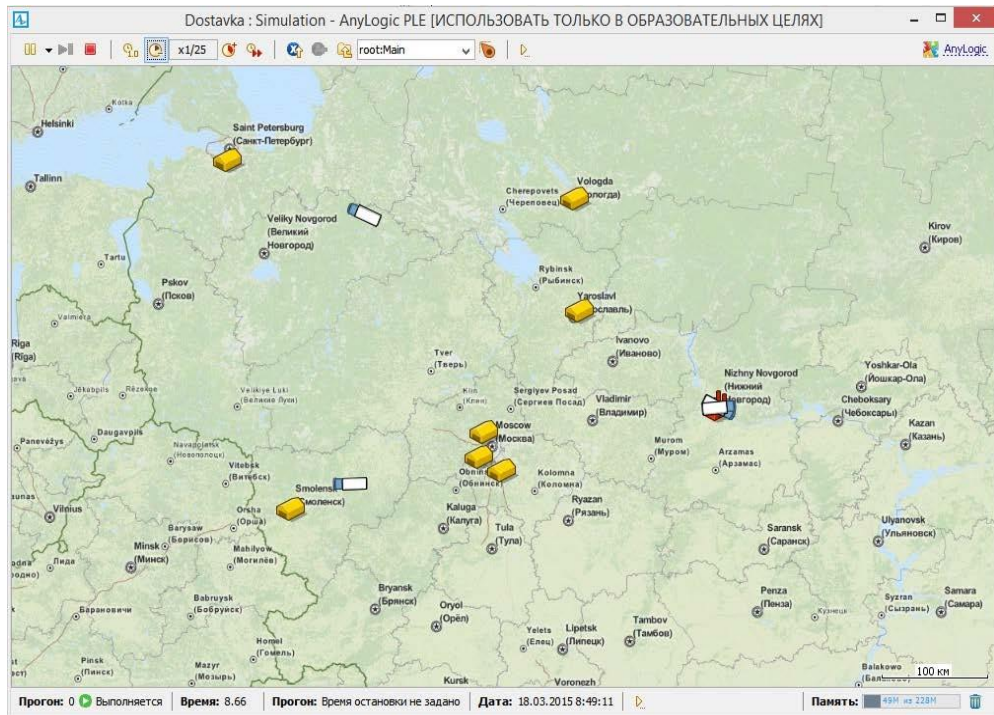


Рис. 28: Модель доставки запчастей в аэропорты.

Если все было сделано правильно, то можно увидеть, как грузовики выезжают с предприятия, доезжают до аэропортов и возвращаются обратно. Изменяя масштаб карты, вы увидите, что грузовики движутся по реальным маршрутам, которые, как и сама карта, подгружаются из сети во время исполнения модели.

Чтобы оценить загруженность транспортного парка предприятия откроем агент **manufacturing** при запущенной модели. Для этого переместитесь по области отображения модели, зажав ЛКМ, как на рисунке 29.

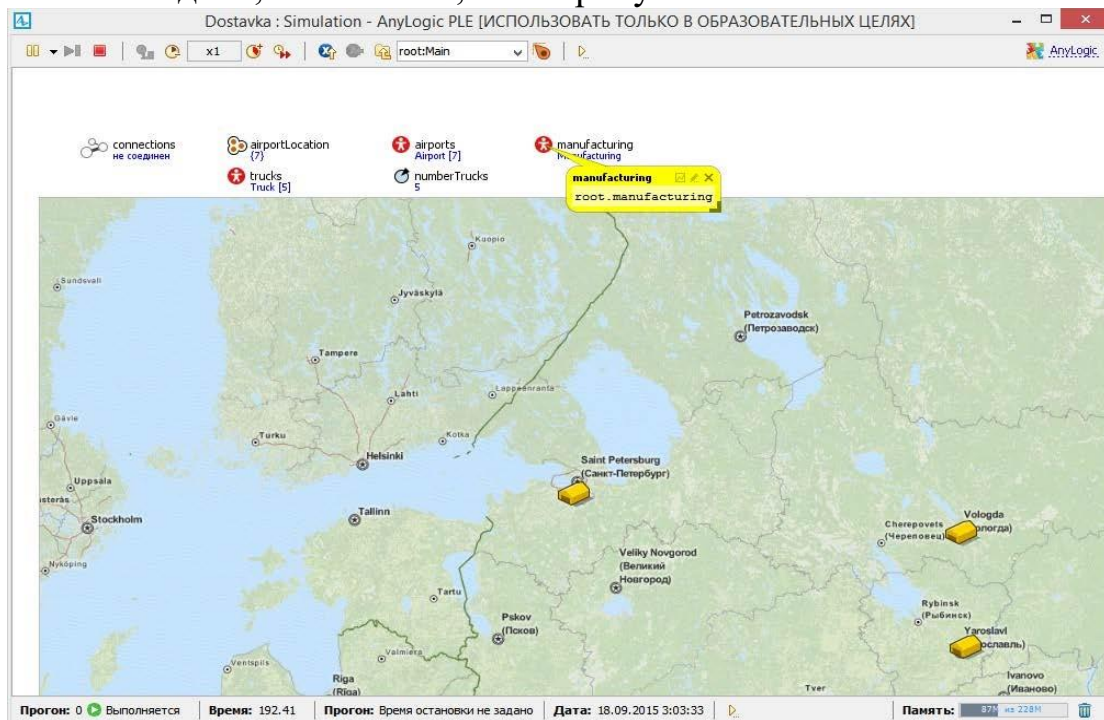


Рис. 29: Обращение к агенту manufacturing из запущенной модели.

Открыв агент **manufacturing**, обратите внимание на блок **Resource Pool**.

По ходу обработки заказа фиксируется количество захваченных ресурсов, по которому можно оценить рациональность использования ресурсов. При использовании пяти грузовиков, средняя загруженность ресурсов составляет примерно 50 процентов, то есть только три из пяти грузовиков задействованы (рисунок 30).

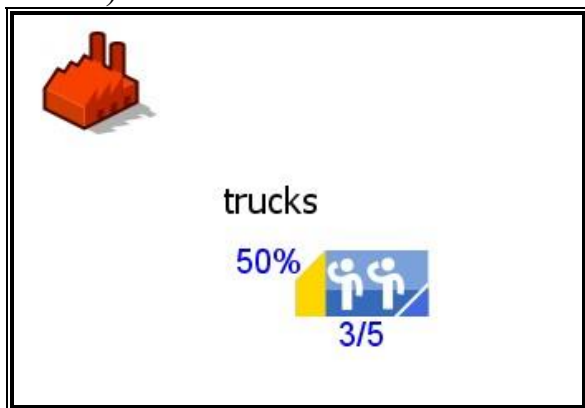


Рисунок 30. Использование ресурсов при доставке запчастей

По условию задачи, загруженность ресурсов при доставке должна быть не более 85%, следовательно, часть ресурсов простаивает и приносит убытки предприятию. Проведем оптимизацию.

Чтобы провести оптимизацию необходимо вызвать новый эксперимент.

Для этого на вкладке **Проекты**, вызовите контекстное меню ПКМ на **Simulation: Main**, далее Создать/Эксперимент:

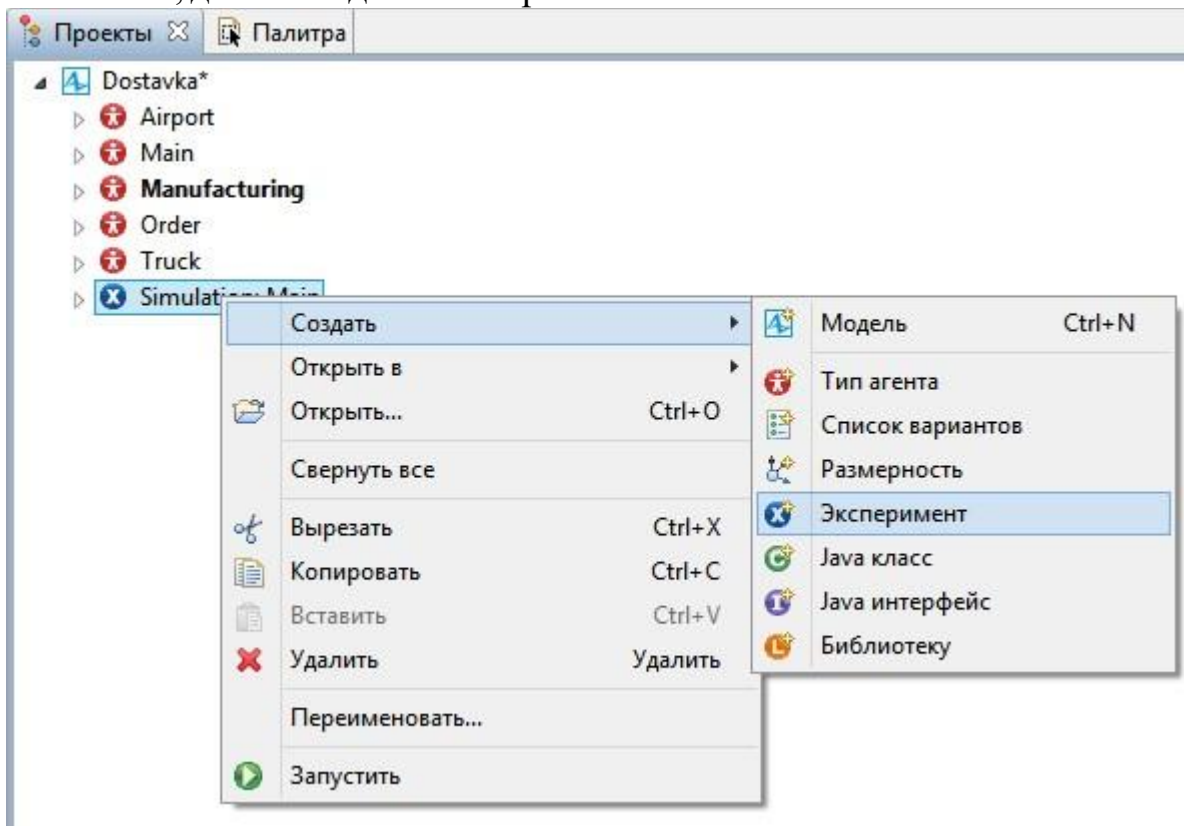


Рис. 31. Вызов эксперимента для оптимизации

Далее откроется окно с параметрами эксперимента. Здесь выберите **Тип эксперимента: Оптимизация** и нажмите **Готово**.

32. Перейдите в свойства эксперимента и измените их соответственно рисунку

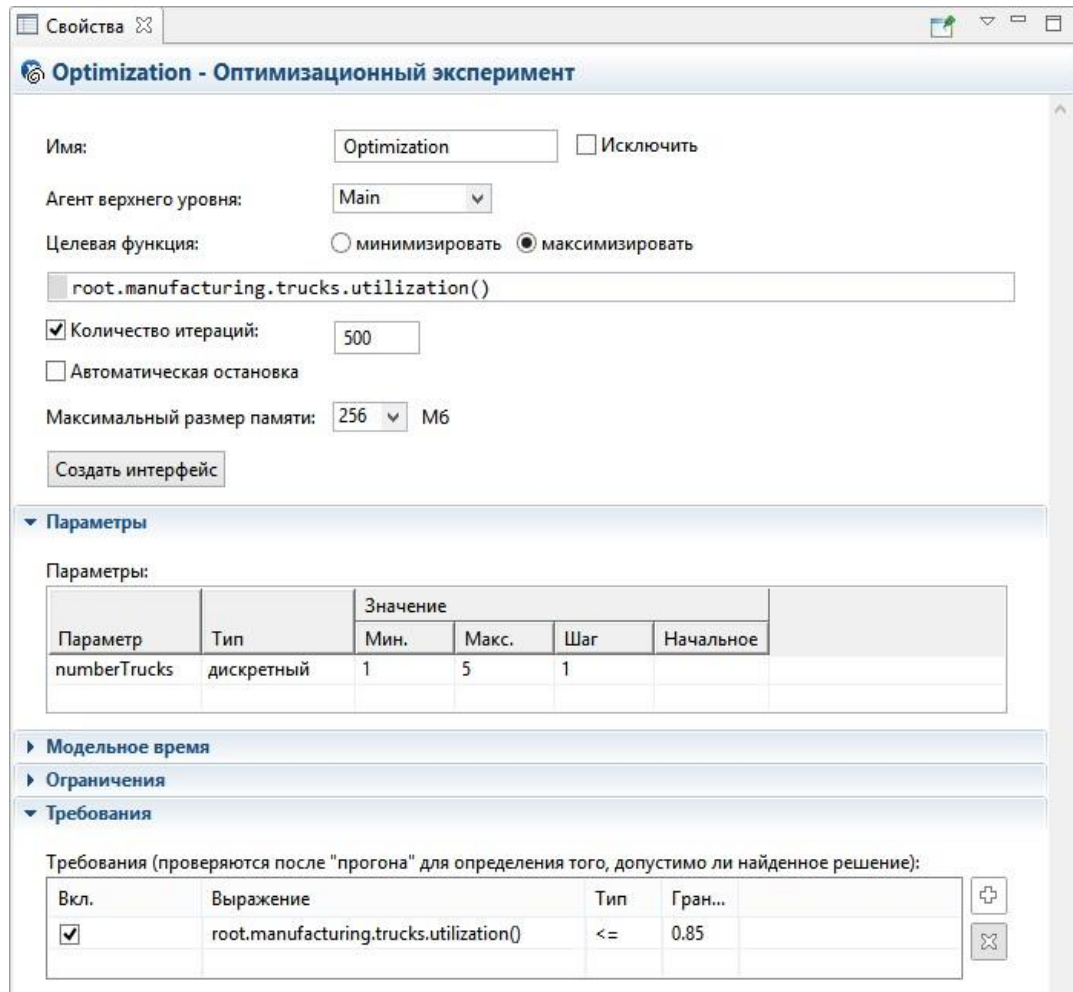


Рис. 32. Свойства оптимизационного эксперимента

В качестве изменяемого параметра выбран **numberTrucks** (количество грузовиков), которое меняется дискретно от 1 до 5. За целевую функцию принята средняя загрузка всех грузовиков, которая максимизирована с ограничением (не больше 85%) в пункте **Требования**. Строчка кода *root.manufacturing.trucks.utilization()* позволяет обратиться к параметру средней загрузки **utilization()** ресурса **trucks** агента **manufacturing**, находящегося на корневом агенте **root** (в данном случае **main**).

Запустите эксперимент по оптимизации, выбрав его из вкладки **Палитра**. На рисунке 33 приведен пример исполнения эксперимента.

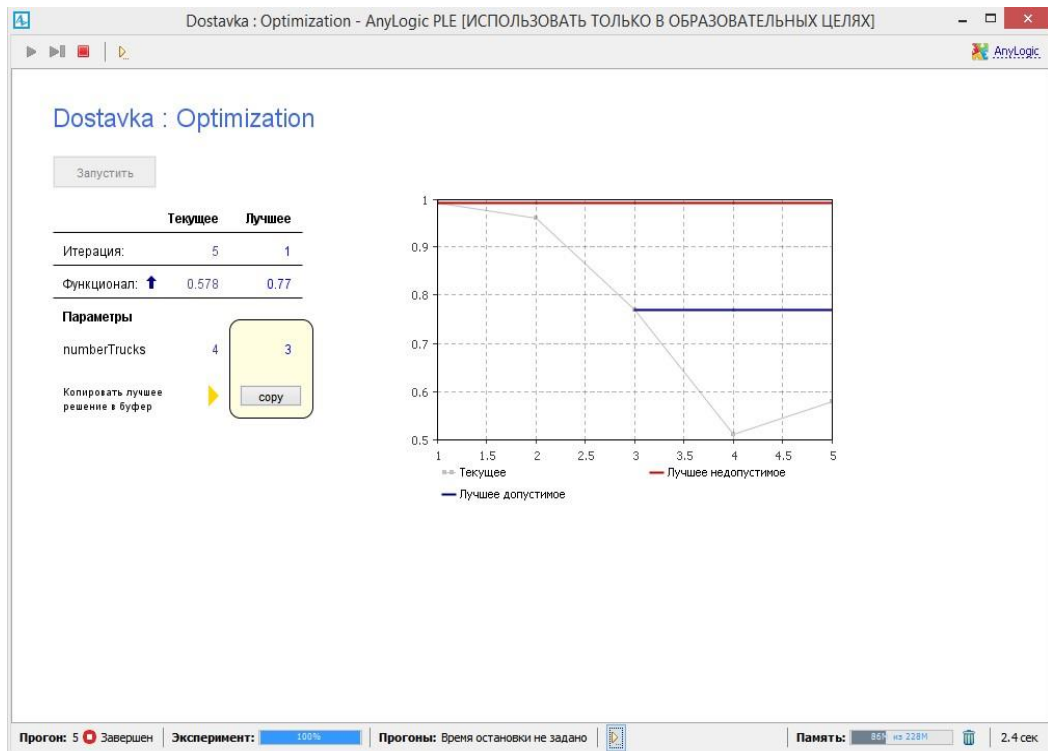


Рис. 33. Оптимизация процесса доставки запчастей

Как видно из рисунка 33, лучшее допустимое значение загруженности ресурсов находится на отметке 77% при использовании 3-х грузовиков. При использовании 4-х грузовиков целевая функция превышает заданное в свойствах эксперимента требование.

Теперь откройте свойства параметра **numberTrucks** и установите:

- Значение по умолчанию: 3.

Запустите симуляцию модели и оцените среднюю загруженность ресурсов при данном количестве грузовиков, аналогично, как на рисунке 30.

Если все сделано согласно методическим указаниям, среднее значение загруженности ресурсов на предприятии будет колебаться между 75-77%.