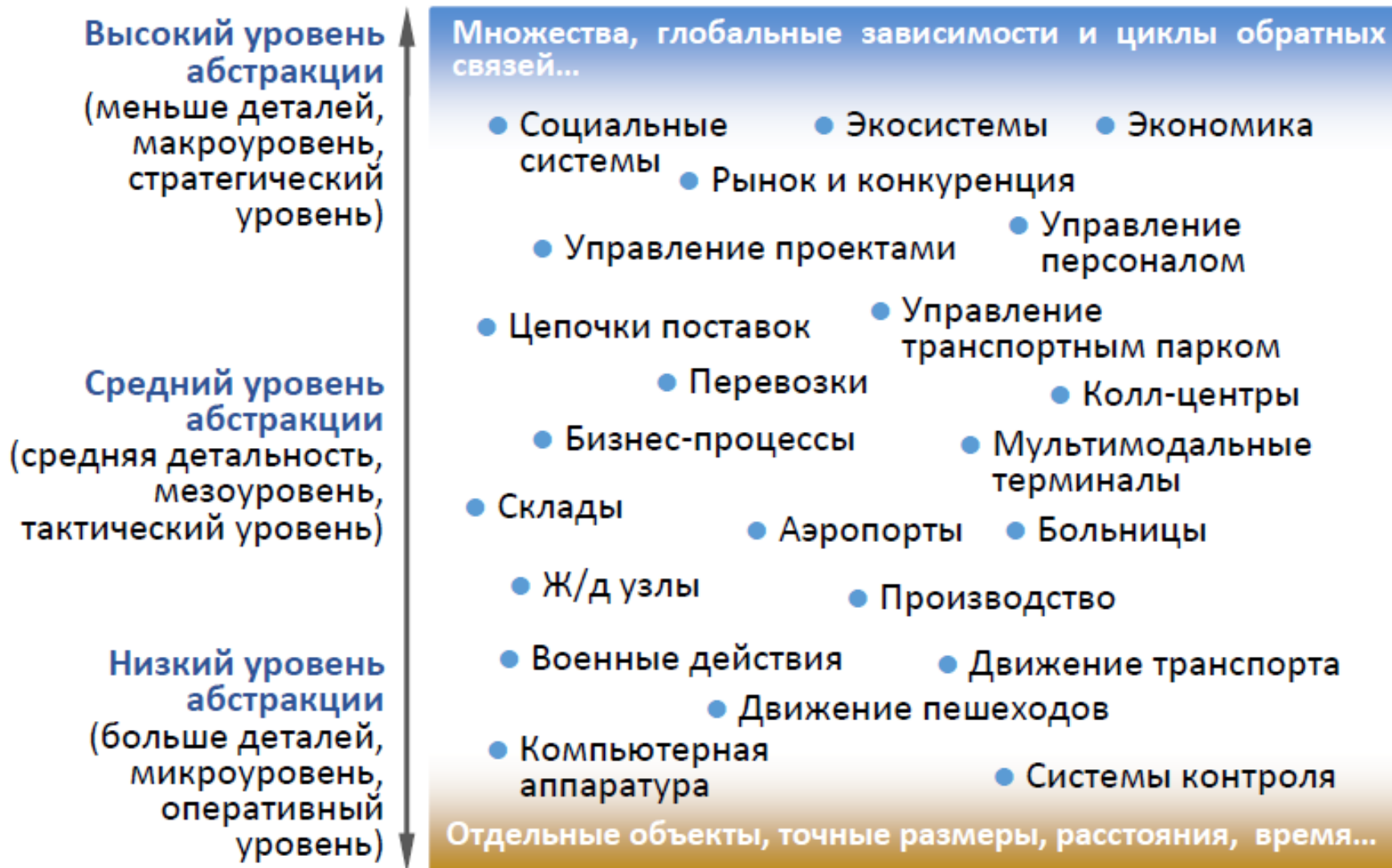


Преимущества имитационного моделирования

1. Анализ систем и поиск решений, когда аналитические методы и линейное программирование не применимы.
2. Разработка имитационной модели проще, чем создание аналитической.
3. Структура имитационной модели естественным образом отображает структуру моделируемой системы.
4. Имитационная модель позволяет отслеживать требуемые объекты системы и проводить статистический анализ.
5. Возможность проигрывать модель во времени и анимировать ее поведение.
6. Более убедительная презентация модели для неспециалистов.

Области применения имитационного моделирования



Основные подходы ИМ в Anylogic

1. Системная динамика

- высокий уровень абстракции – на уровне функций
- задачи стратегического уровня в экономических, социальных, технических, физических и др. системах
- анализ взаимосвязи разных процессов, прогноз реакции системы

2. Дискретно-событийное (процессное)

- средний и низкий уровень абстракции – на уровне структуры
- задачи оперативного и тактического уровня в производственных, транспортных, торговых, информационных процессах
- анализ загруженности отдельных узлов, оценка инвестиций

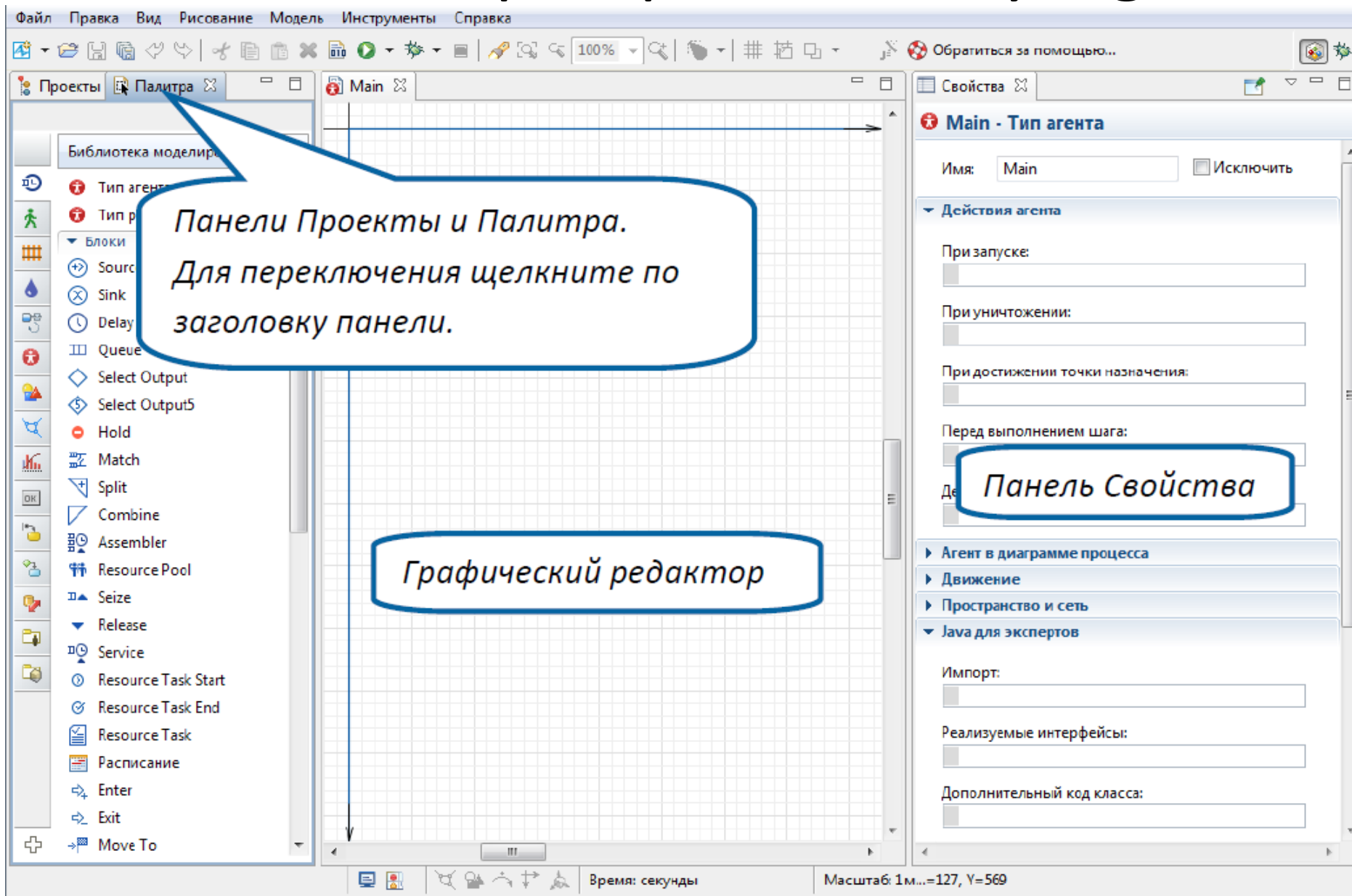
3. Агентное

- любой уровень абстракции – агент может олицетворять собой любой объект в действии
- задачи, когда неизвестны поведение системы в целом, зависимости между состояниями системы, нет четкой схемы процессов, но при этом известно, как ведут себя отдельные элементы системы

4. Многоподходное

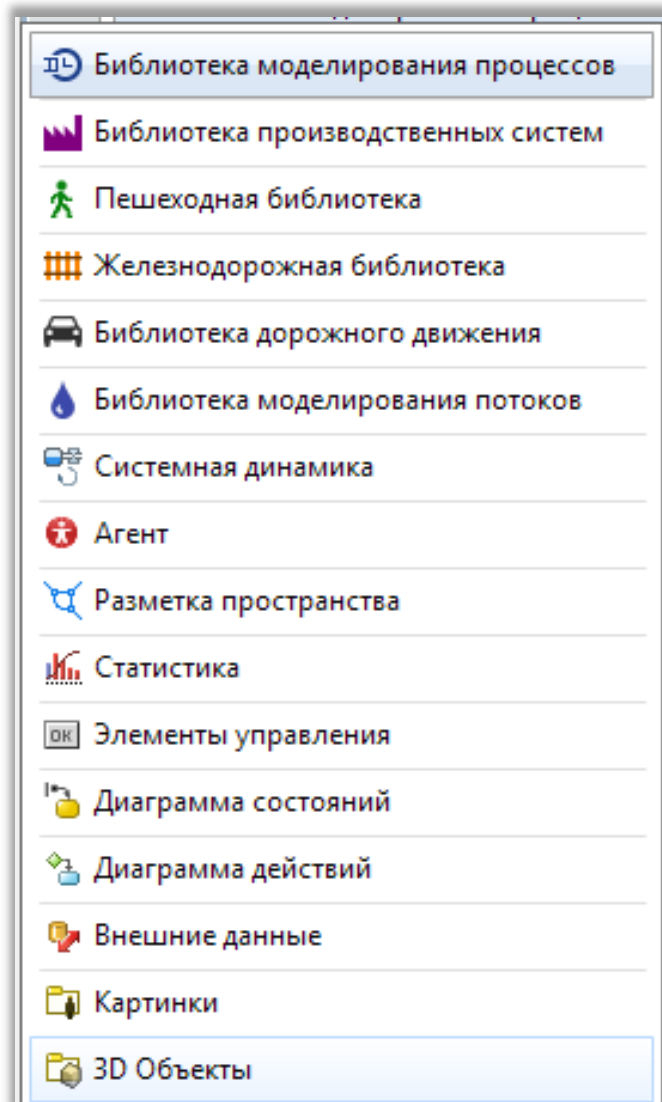
- комбинирование подходов к ИМ так, чтобы достоинства одних компенсировали недостатки других

Рабочее пространство AnyLogic

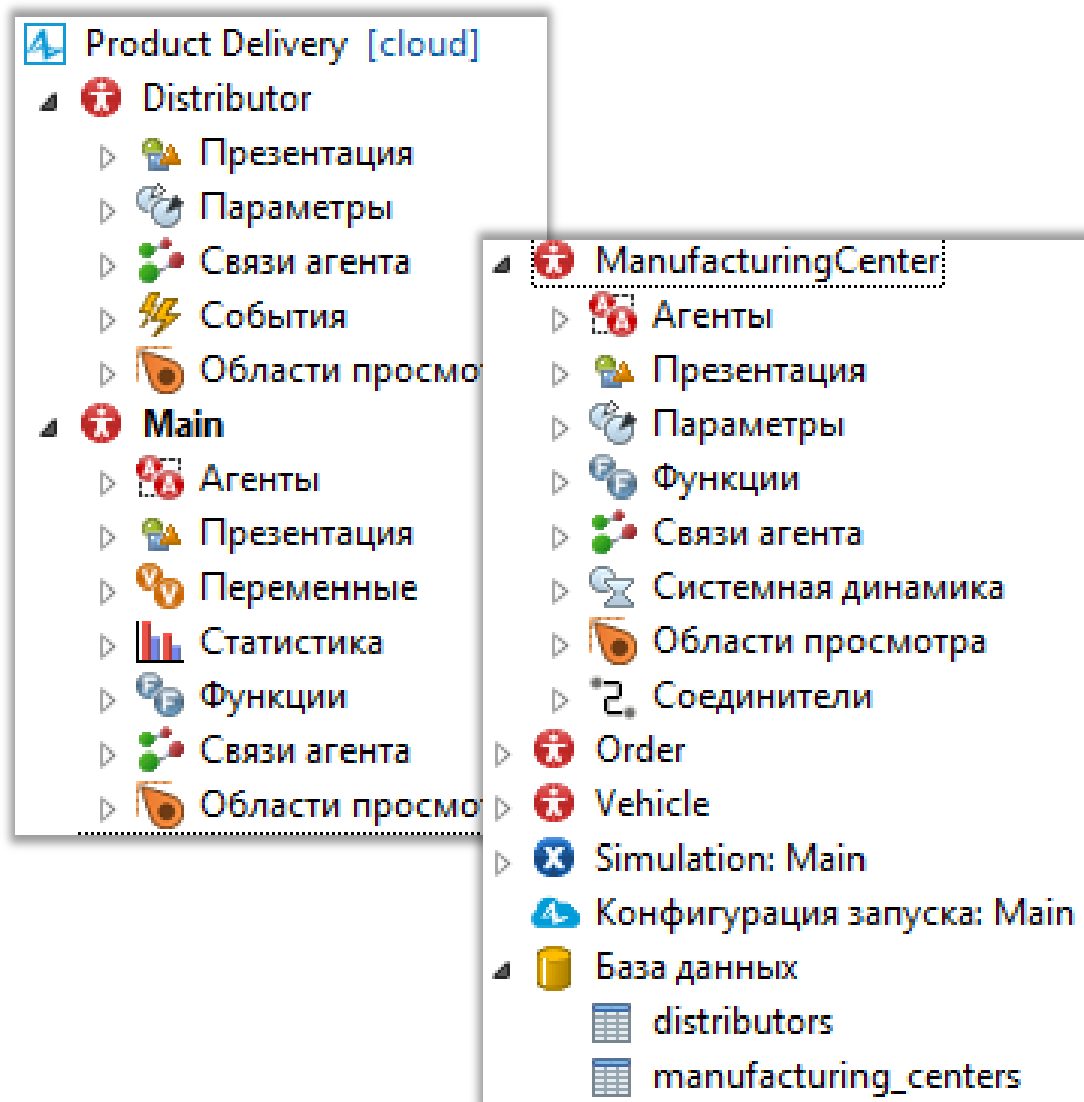


Рабочее пространство AnyLogic

Библиотеки:

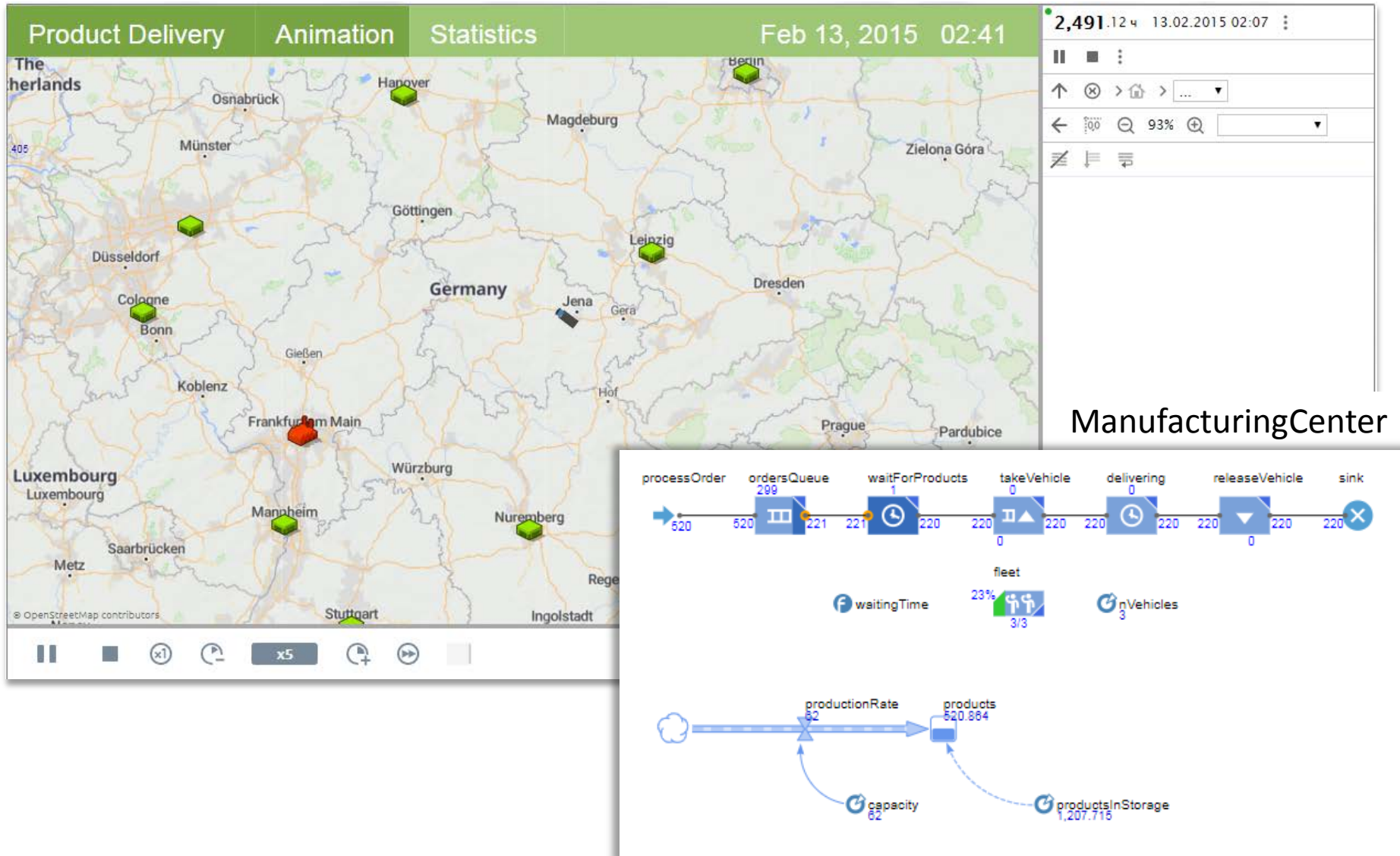


Структура проекта:



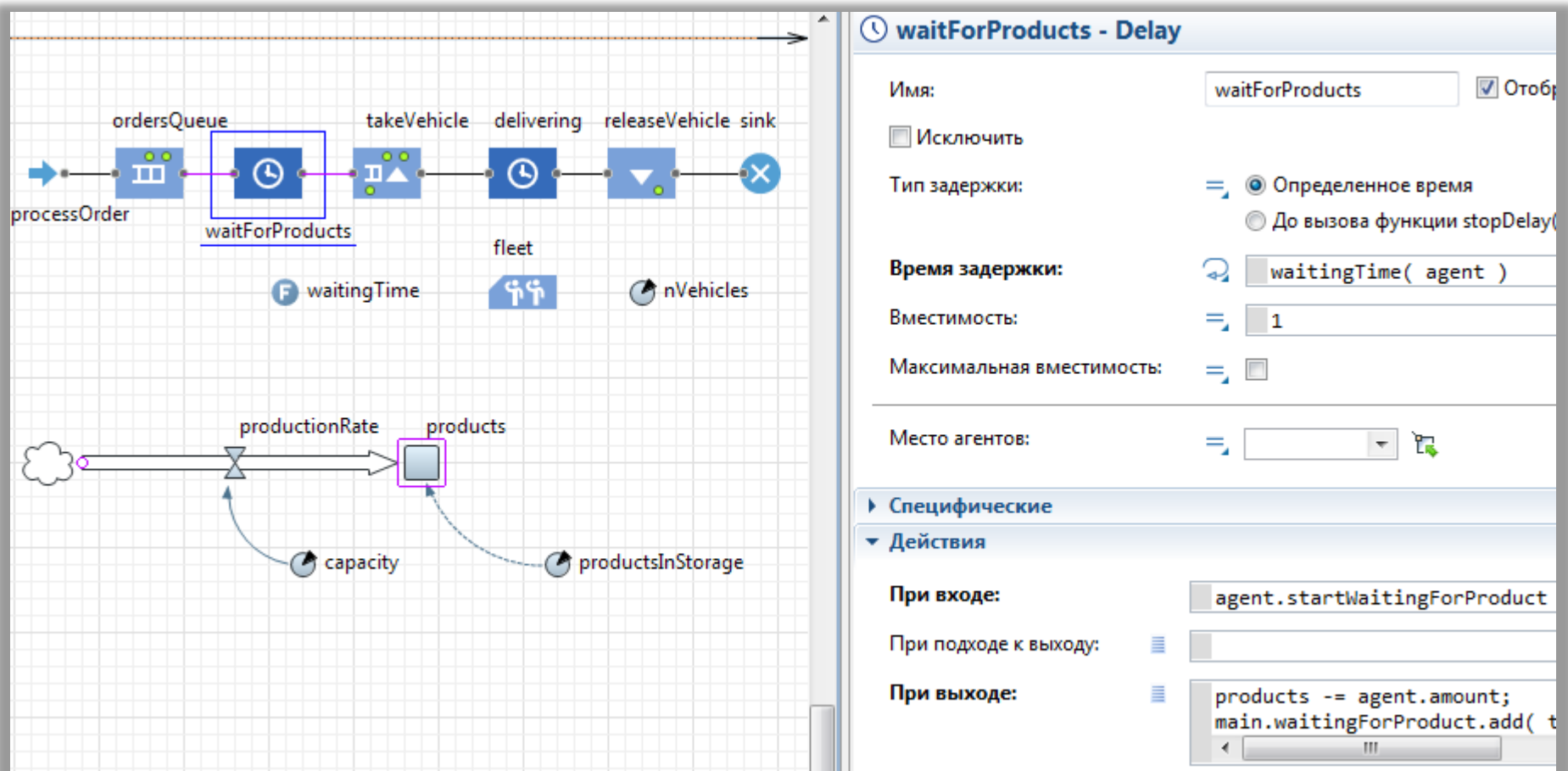
Пример многоподходной модели

Модель поставки товаров (Main)



Пример многоподходной модели

Агент ManufacturingCenter:



Вариант использования Anylogic как среды визуального программирования

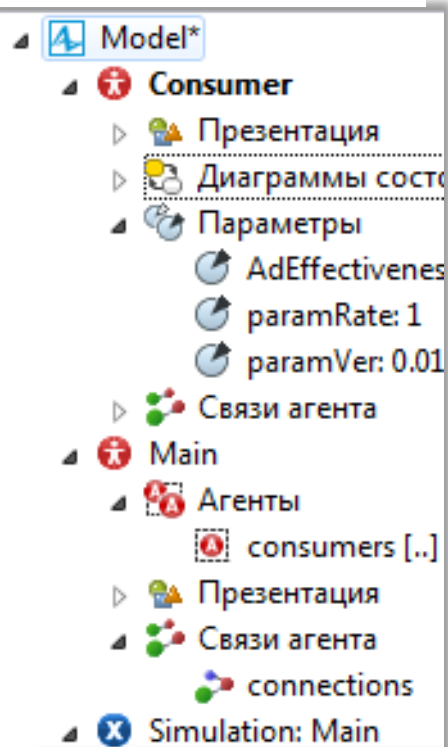
The image displays the Anylogic visual programming environment. On the left, a calculator UI is shown on a grid. It features two input fields at the top labeled `n_str` and `n`. Below them is a numeric keypad with buttons for digits 0-9, a clear button (C), and a factorial button (n!). The factorial button is currently selected, indicated by a blue dashed border. On the right, a properties panel for the `fact_btn - Кнопка` is visible. It includes fields for the button's name (`fact_btn`), a checkbox for "Отображается на верхнем уровне" (checked), a label field containing `n!`, and a visibility toggle set to "да". The "Действие" (Action) section contains the following Java code:

```
n=Integer.parseInt(n_str);
double fact=1;
for (int i=1;i<n+1;i++)
{
    fact=fact*i;
}
n_txt.setText(n_str+"!");
fact_txt.setText(fact);
```

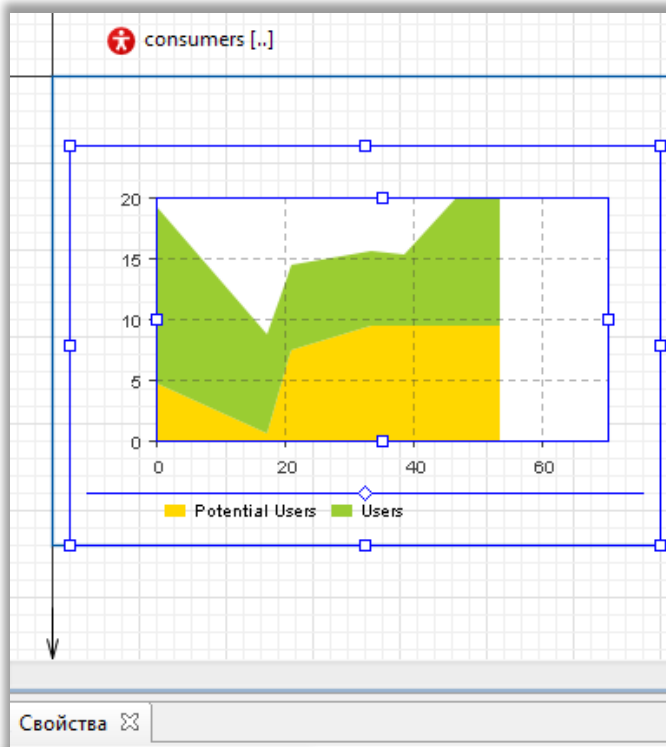
Below the code, there are expandable sections for "Внешний вид" (Appearance), "Местоположение и размер" (Location and Size), "Специфические" (Specific), and "Описание" (Description).

Использование агентов и диаграммы состояний

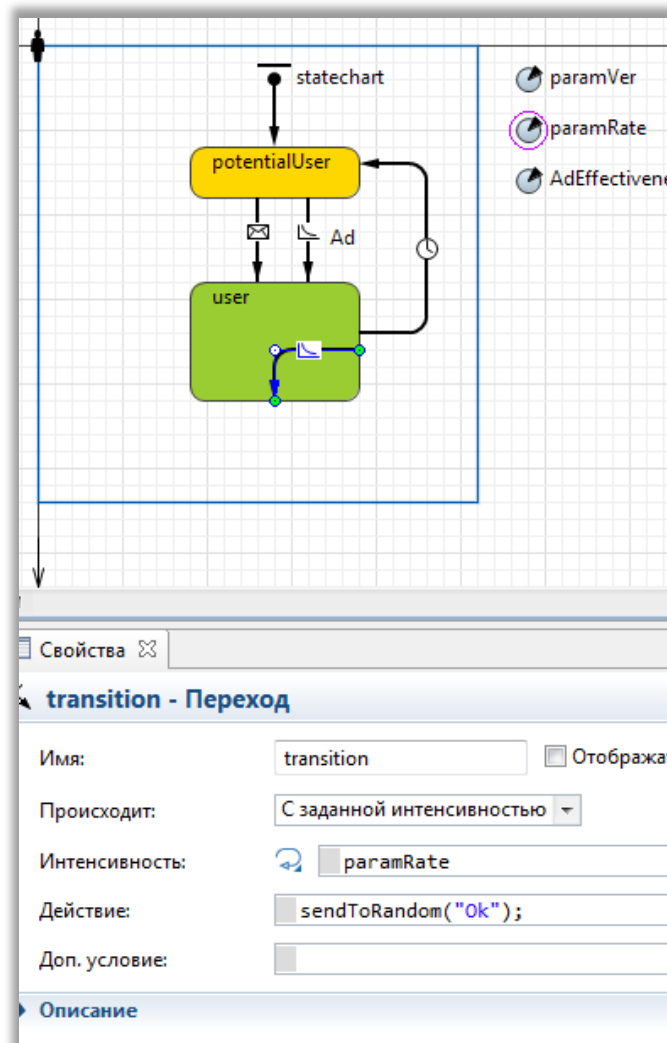
Структура проекта:



Main:



Consumer:



consumers - Consumer

Статистика

Имя: consumersStat

Тип: ☒ Кол-во ☐ Сумма ☐ Среднее ☐ Мин. ☐ Макс.

Условие: item.inState(Consumer.potentialUser)

Имя: consumersStat2

Тип: ☒ Кол-во ☐ Сумма ☐ Среднее ☐ Мин. ☐ Макс.

Условие: item.inState(Consumer.user)

Использование агентов, процессов, диаграммы состояний, диаграммы действий и средств ГИС

The screenshot displays a simulation environment with a map of Crimea. The interface includes a project tree on the left, a main map area, and several configuration windows.

Project Tree (Left):

- m04_GIS*
 - Main
 - Агенты
 - производство
 - склады [...]
 - транспорты [...]
 - Презентация
 - Связи агента
 - Производство
 - Агенты
 - Презентация
 - Диаграммы действий
 - Связи агента
 - Соединители
 - Склад
 - Презентация
 - Связи агента
 - Транспорт
 - Презентация
 - Диаграммы состояний
 - Связи агента
 - Simulation: Main
 - Конфигурация запуска: Main
 - База данных
 - ModelMarket
 - Product Delivery [cloud]

Map Area:

The map shows three agent locations marked with red pins: "производство" (production) near Simferopol, "склады [...]" (warehouses) near Kerch, and "транспорты [...]" (transport) near Sevastopol. The city names "Симферополь" and "Керчь" are visible on the map.

Configuration Windows:

- Расположить агентов:** Options for placing agents: ☐ в месте расположения анимации, ☐ в точке (широта, долгота), ☒ в узле, ☐ в 1-м результате ГИС поиска. Below, "Узел:" is set to "Симферополь".
- Начальное количество агентов:** Set to 2.
- Популяция:** Options: ☐ Изначально, ☒ Содержит, ☐ Загружает.
- Начальное количество агентов:** Set to 5.

Properties Panel (Bottom Right):

Main - Тип агента

Имя: Main ☐ Исключить

Действия агента

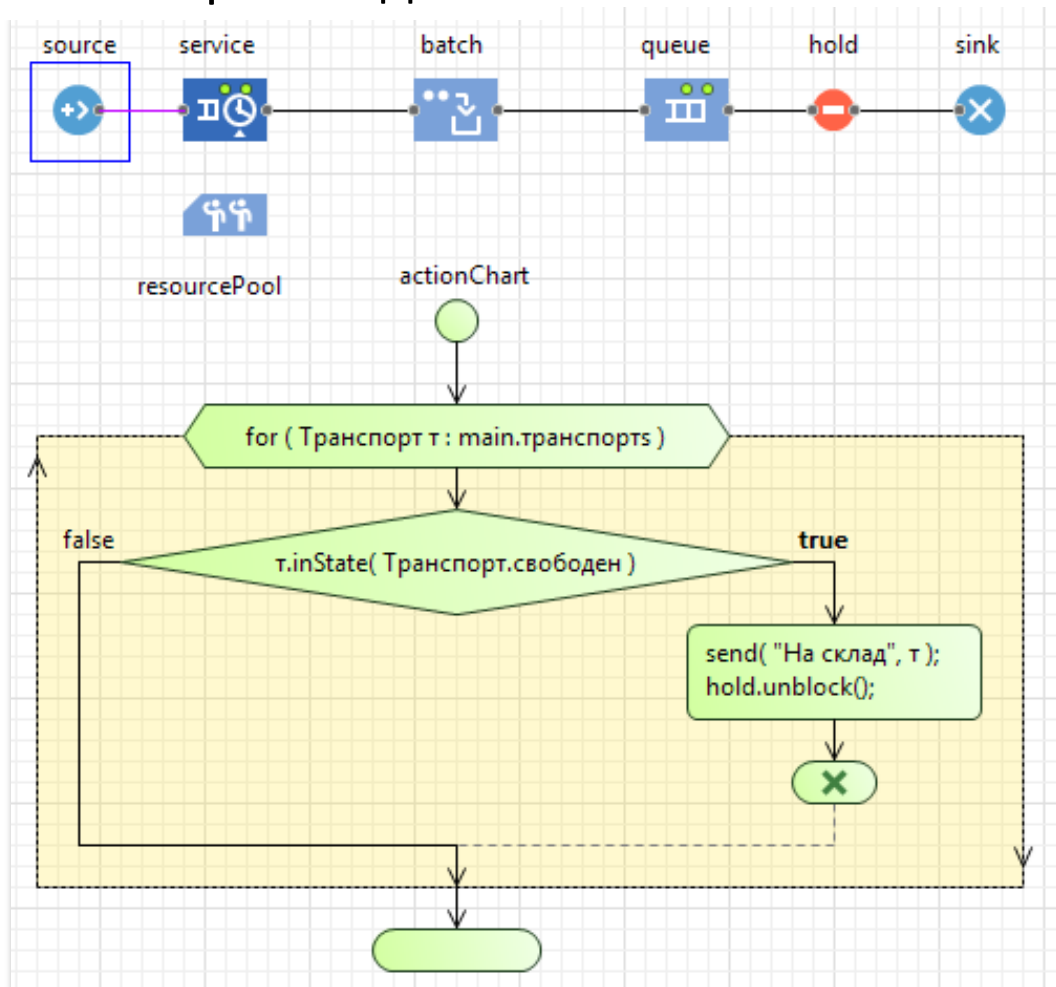
При запуске:

```
склады.get(0).setLocation( Севастополь );  
склады.get(1).setLocation( Керчь );
```

При уничтожении:

Использование агентов, процессов, диаграммы состояний, диаграммы действий и средств ГИС

Агент Производство:



source:

Прибывают согласно:

Интенсивность прибытия:

service:

Захватить: ☒ (альтернативный) набор ресурсов
☐ ресурсы одного типа

Набор(ы) ресурсов:

 Добавить список

Максимальная вместимость: ☒

Время задержки:

resourcePool:

Количество ресурсов:

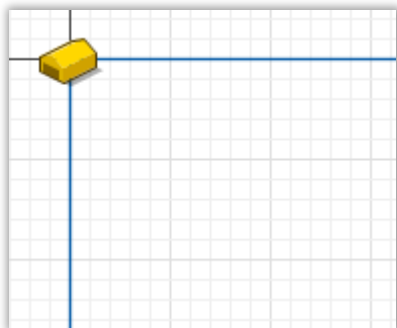
batch:

Размер партии:

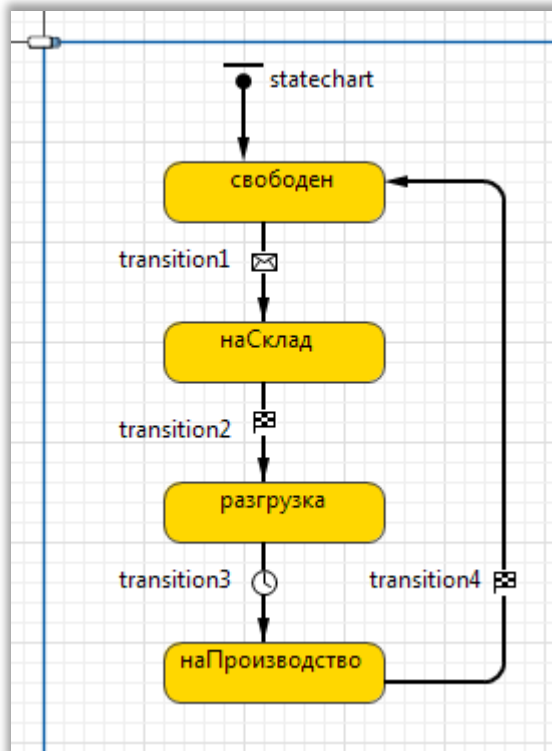
При выходе:

Использование агентов, процессов, диаграммы состояний, диаграммы действий и средств ГИС

Агент Склад:



Агент Транспорт:



transition1:

Осуществлять переход: ☐ Безусловно ☒ При получении заданного сообщения: ☐ Если выполняется условие

Сообщение:

наСклад:

Действие при входе:

transition2, transition4 :

Происходит:

transition3:

Происходит:

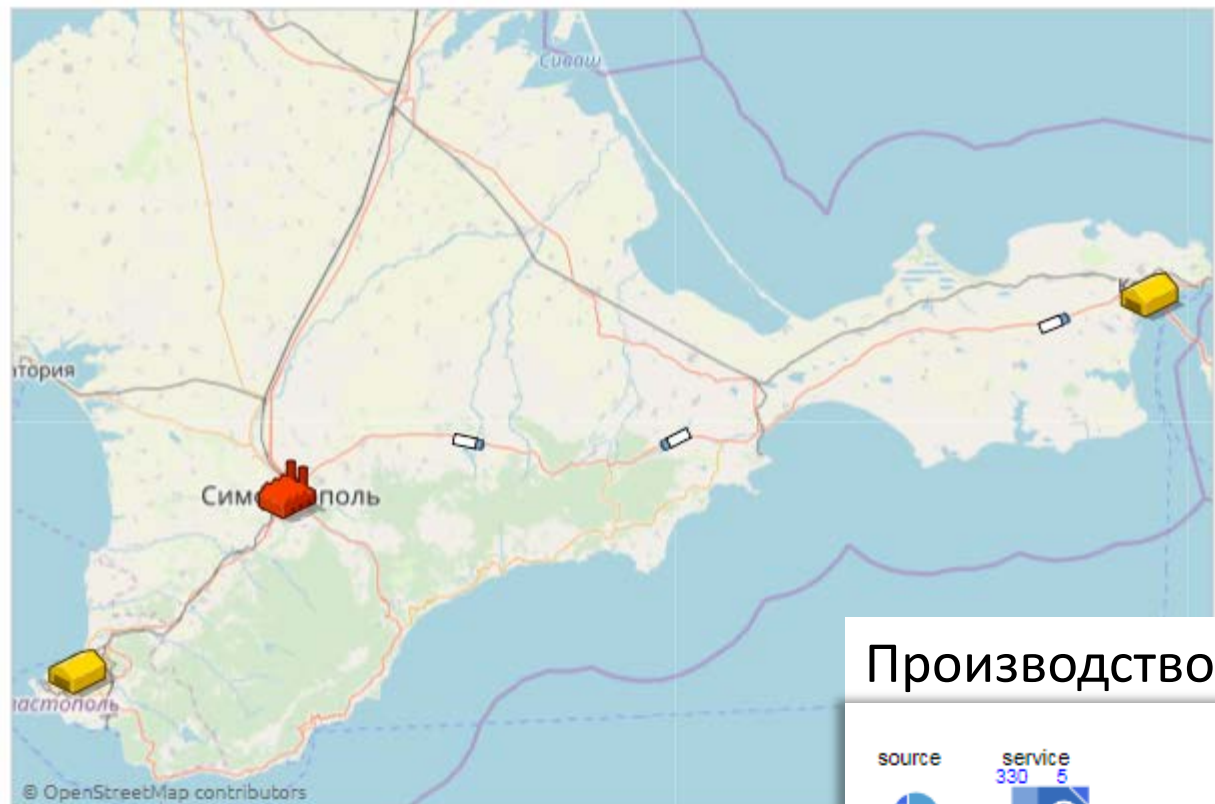
Таймаут:

наПроизводство:

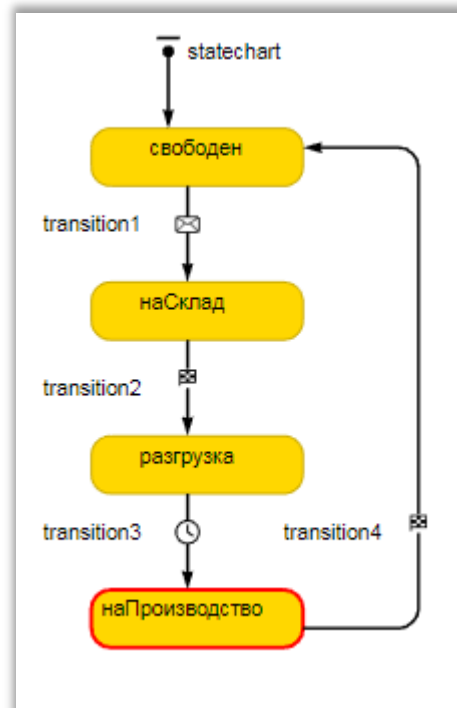
Действие при входе:

Использование агентов, процессов, диаграммы состояний, диаграммы действий и средств ГИС

Процесс выполнения Main:



Транспорт:



Производство:

