



大连理工大学

DALIAN UNIVERSITY OF TECHNOLOGY

智能体仿真 供应链配送仿真实验 实验报告

学 院（系）： 经济管理学院

专 业： 物流管理专业

姓 名： 严梓锴

学 号： 201903020

指 导 教 师： 白朝阳

完 成 日 期： 2022 年 5 月 15 日

大连理工大学

Dalian University of Technology

目录

供应链配送仿真实验.....	1
1 实验内容.....	1
2 实验方法.....	1
2.1 智能体建模基本概念.....	1
3 实验模型.....	1
3.1 智能体模型.....	1
3.2 供应链配模型.....	2
4 实验仿真程序.....	2
4.1 添加 GIS 地图.....	2
4.2 创建分销商智能体类和智能体群.....	3
4.3 添加 Excel 文件.....	3
4.4 添加行动图.....	3
4.5 创建零售商智能体类和智能体群.....	4
4.6 创建订单智能体（单智能体）.....	4
4.7 创建卡车智能体类及智能体群.....	4
4.8 为分销商类添加内容.....	4
4.9 为卡车定义状态逻辑.....	5
4.10 分销商找到卡车行为.....	5
4.11 修改行动图以在模型中添加卡车.....	6
4.12 模拟产品请求的发送.....	6
5 实验结果.....	7
6 实验思考.....	10
6.1 关键问题的解答.....	10
6.2 思考题解答.....	10
6.3 完成思考题后的仿真.....	21
7 参考文献.....	22

图目录

图 1 行动图 1	3
图 2 卡车状态逻辑.....	5
图 3 分销商找到卡车行为.....	5
图 4 修改后的行动图.....	6
图 5 模拟产品请求的发送.....	6
图 6 修改代码错误后正确的代码块.....	7
图 7 程序主界面建模.....	7
图 8 分销商建模.....	8
图 9 订单建模.....	8
图 10 零售商建模.....	9
图 11 卡车建模.....	9
图 12 仿真运行效果.....	10
图 13 零售商只给最近的分销商派单情况下的运行示意图.....	12
图 14 添加订单时间参数.....	13
图 15 分销商添加积压订单数量和仿真运行时显示的文本.....	14
图 16 订单列表属性.....	14
图 17 分销商智能体类用 connections 接收传递过来的订单数据.....	15
图 18 卡车行动图修改 1	16
图 19 卡车行动图修改 2	16
图 20 卡车智能体类添加最大承载量.....	17
图 21 调整零售商随机产生订单的数量分布.....	18
图 22 卡车智能体类添加当前承载量变量.....	18
图 23 修改卡车行动图第一阶段的触发条件的行动代码.....	19
图 24 分销商处理订单的算法.....	20
图 25 完成思考题后的仿真示意图.....	21

表目录

表 1 零售商只给最近的分销商派单情况下零售商、分销商对应关系表.....	12
表 2 仿真模型参数调整表.....	21

供应链配送仿真实验

1 实验内容

构建一个简单的供应链模型仿真，包含以下要点：

- (1) 分销商和零售商将位于不同的地方。
- (2) 有时零售商从分销商订购一定数量的产品。
- (3) 产品用卡车运输。
- (4) 运输后，卡车返回在分销商处的原始位置。
- (5) 模拟零售商如何请求产品以及产品如何从分销商运往零售商。
- (6) 在模型中添加卡车，生成产品请求并将它们分配到卡车上，让卡车移动到零售商
- (7) 后返回到他们分销商处的原始位置。

2 实验方法

2.1 智能体建模基本概念

基于智能体建模可以被简单地定义为分散的、以个体为中心的一种模型设计方法。在构建基于智能体的模型时，建模者需要判断智能体的类型（如人、车辆、订单、产品、公司、项目等）、智能体的活动，定义其行为（如主要驱动因素、反应等），并将它们放置在特定环境中，连接智能体使其相互间能进行互动，运行模型。

3 实验模型

3.1 智能体模型

在创建一个基于智能体模型的过程中，需考虑以下几点：

- i. 确定实际系统中比较重要的对象，这些对象将被确定为模型的智能体。
 - ii. 确定实际对象间的持续性关系，并建立智能体之间相应的关系链接。
 - iii. 根据需要选择空间模型并且设定智能体在空间中的位置，如果智能体是移动的，则需设置智能体移动的速度、路径等。
 - iv. 判断智能体生命周期中的重要事件，这些事件可能由外部触发，也可能是由智能体自身动态性引起的内部事件。
 - v. 明确智能体的行为。
- ①智能体对外部事件的反应，可以使用消息处理和函数调用两种方法。

- ②智能体的状态及变化，可以使用状态图定义智能体的状态及变化。
- ③智能体内部事件处理，可以使用事件或到时变迁来定义。
- ④智能体的内部流程，可以在智能体中创建流程图。
- ⑤智能体动态的连续时间，可以在智能体中创建存量、流量图。
- vi. 确定智能体间的通信模式及其时间规律。
- vii. 确定智能体需要记录的各类信息，并确定存储方式和信息内容。
- viii. 存在于所有智能体外部并且被所有智能体所共享的信息、动态，被定义为模型的全局变量。
- ix. 定义各类数据统计结构，保存模型的输出结果。

3.2 供应链配模型

在供应链系统中，配送中心对产品的配送是物流作业流程的重要环节之一，选择合理的配送路径，以达到运输总路程、运输总成本、运输总时间最优，这是物流供应链系统中产品配送需要考虑的重要问题。

产品配送问题的研究以配送中心为核心，配送中心根据零售商的订单，将产品以合理的配送路线由配送车辆运输至各零售商处，将运达信息由车辆带回配送中心，对产品配送问题的研究旨在降低运输成本并缩短配送时间，提高配送效率。

产品配送的一般流程：

- ①接收订单。配送中心接收到订单后，系统自动显示订单客户信息，包括货物类型、数量等，并根据订单需求以及配送地点、运输能力等，对订单进行拆分、整理、组合。
- ②确定产品的配送路径。确定车辆在各配送节点的先后顺序。
- ③下发运输计划。将确定好的运输路径下发至车辆部门，车辆部门按照运输需要安排车辆，等待配送，并对配送车辆安排情况及时反馈至配送中心部门。
- ④货物配送。车辆在接收到配送任务后，按照计划规定的配送路径进行货物的发运工作。
- ⑤车辆将货物运输至指定的客户地点后，返回配送中心后将客户签收信息及时反馈。

4 实验仿真程序

4.1 添加 GIS 地图

目标：为仿真模型创建 GIS 环境。

具体步骤：

- ① 新建模型，添加 GIS 地图。
- ② 添加图形文件 world_borders.shp

- ③ 设定参数：纬度:47.5；经度:2；比例 1:10000000；填充颜色:skyBlue；宽度:590；高度: 490

4.2 创建分销商智能体类和智能体群

目标：引入分销商

具体步骤：

- ① 添加智能体
- ② 选择智能体群体
- ③ 定义属性名
- ④ 选择二维仓库图像
- ⑤ 设定群的大小为 5
- ⑥ 移动智能体群体和动画到合适的位置

4.3 添加 Excel 文件

- ① 将 Excel 文件拖入 Main 界面
- ② 在“文件”属性中索引需要打开的包含分销商和零售商位置数据的 Excel 文件。

4.4 添加行动图

目标：读取 Excel 中的数据，对分销商的位置数据进行初始化操作。

具体步骤：

- ① 行动图如图 1 所示。

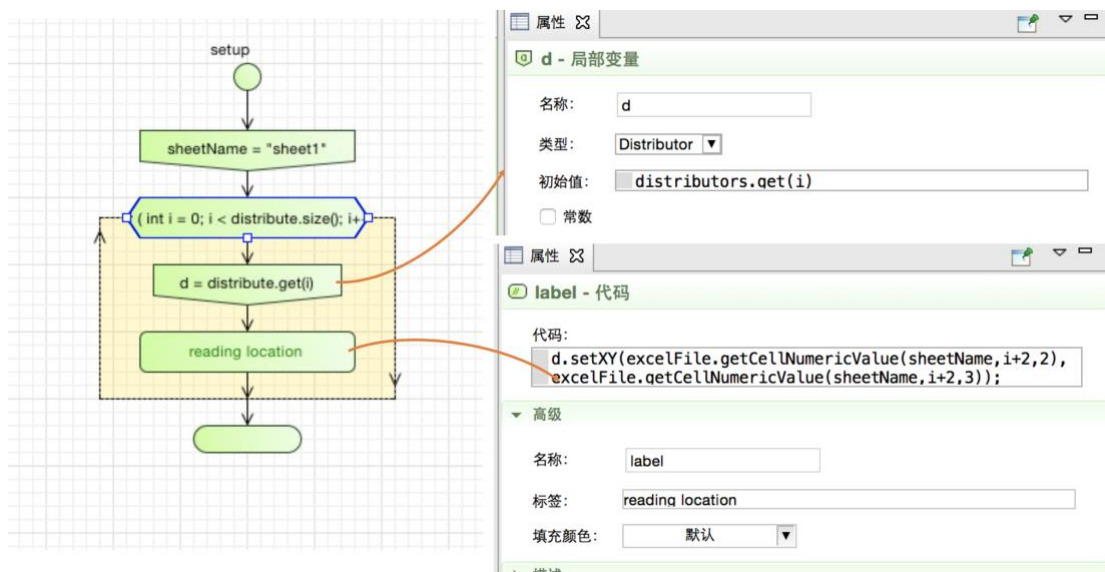


图 1 行动图 1

- ② 放置行为图，并在启动代码中调用（Main-智能体行动-启动时-setup();）

4.5 创建零售商智能体类和智能体群

按照与分销商相同的方式构建零售商智能体类和智能体群，并从 Excel 表格中读取位置数据。

4.6 创建订单智能体（单智能体）

- ① 添加 int 型参数 amount
- ② 设置无动画
- ③ 手动 添加 Retailer 类的第二个参数 retailer

4.7 创建卡车智能体类及智能体群

- ① 选择二维卡车形象
- ② 设置初始群体大小为 0
- ③ 添加参数：经销商类型
- ④ 添加变量：订单类型
- ⑤ 设置空间类类型为 GIS

4.8 为分销商类添加内容

- ① 添加一个代表分销商所拥有的卡车数量的变量，设为 1-5 随机（初始值：`uniform_discr(1,5)`）
- ② 添加集合，集合类设定为 `ArrayList`，集合元素设定为卡车类。

4.9 为卡车定义状态逻辑

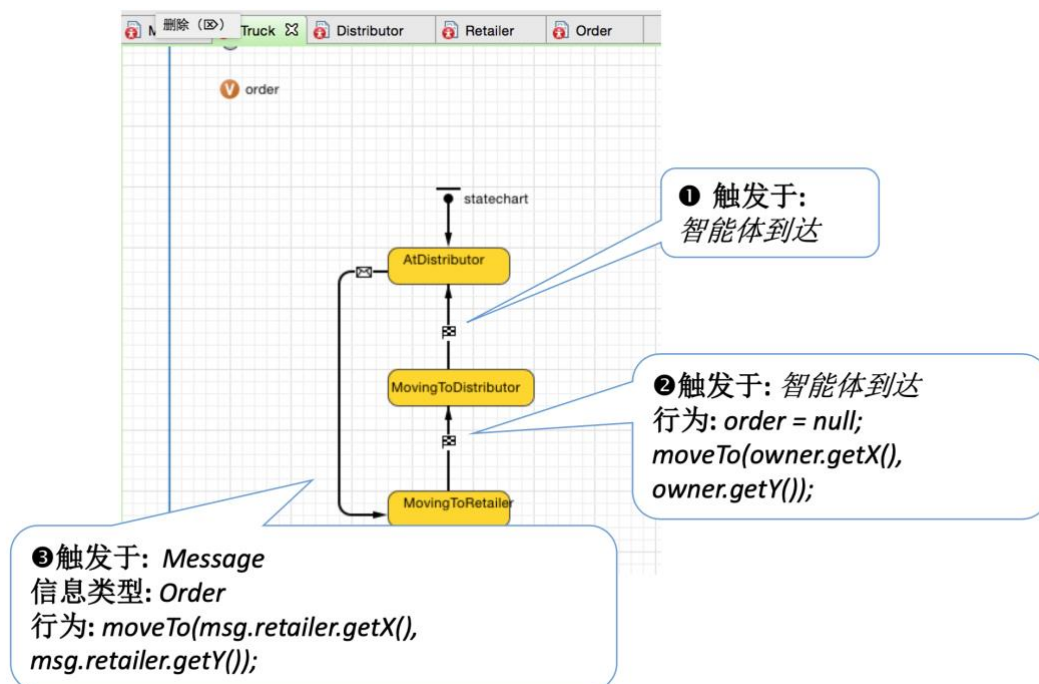


图 2 卡车状态逻辑

4.10 分销商找到卡车行为

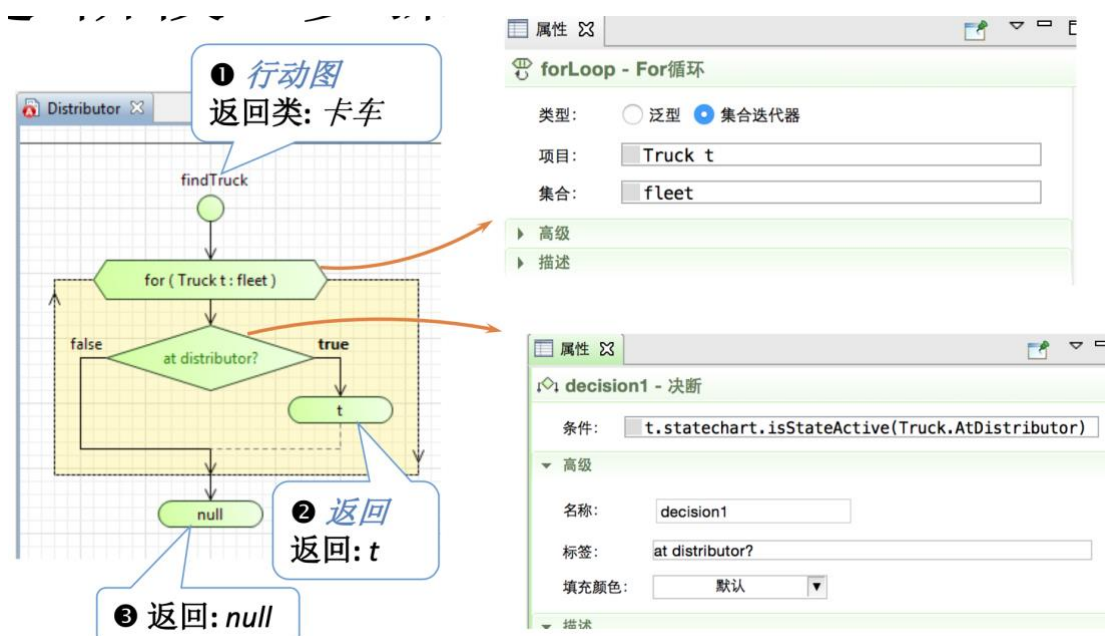


图 3 分销商找到卡车行为

4.11 修改行动图以在模型中添加卡车

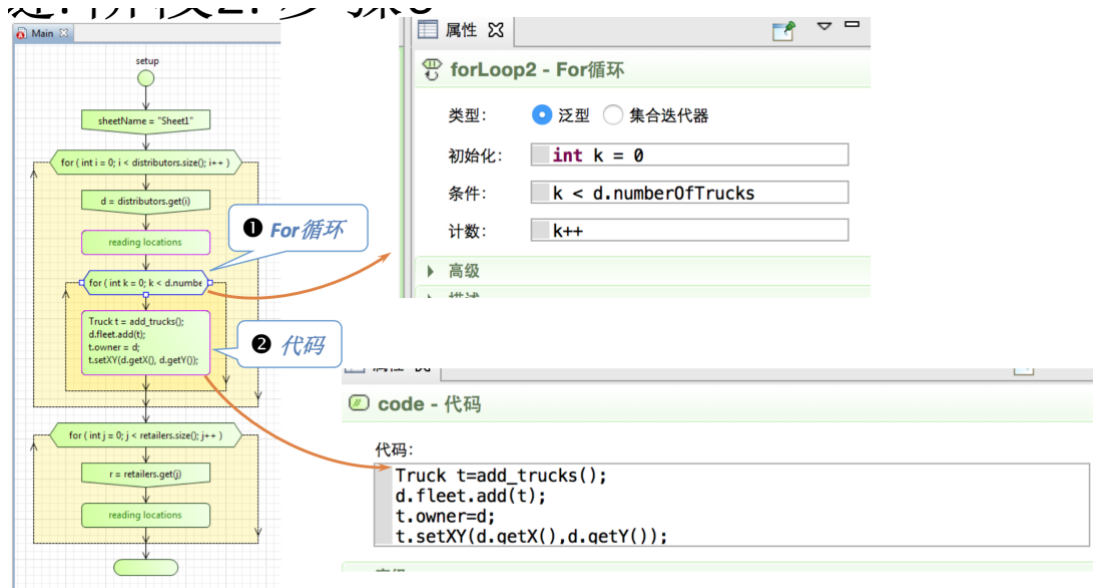


图 4 修改后的行动图

4.12 模拟产品请求的发送

供应链. 阶段2. 步骤7

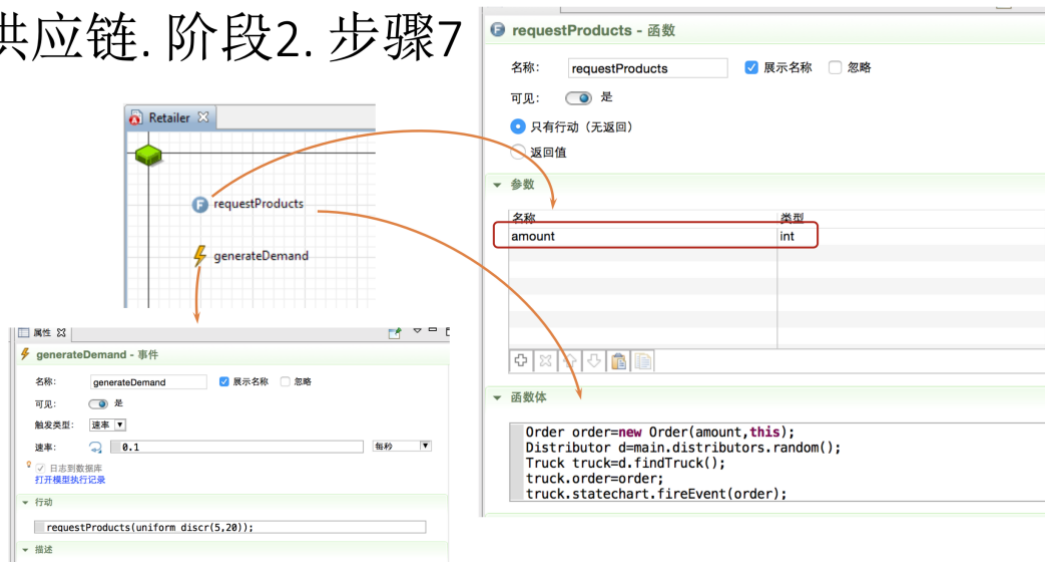


图 5 模拟产品请求的发送

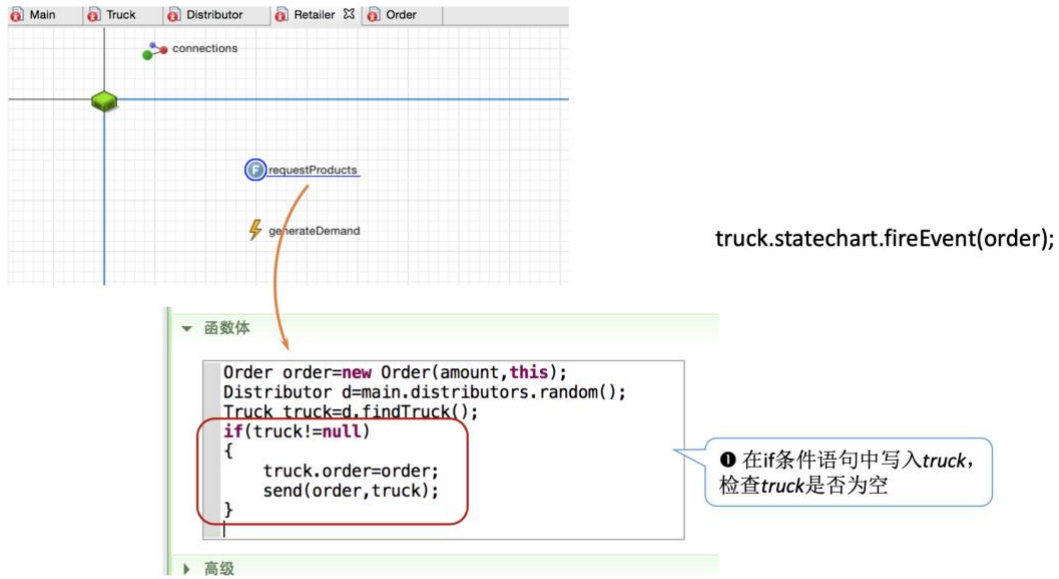


图 6 修改代码错误后正确的代码块

5 实验结果

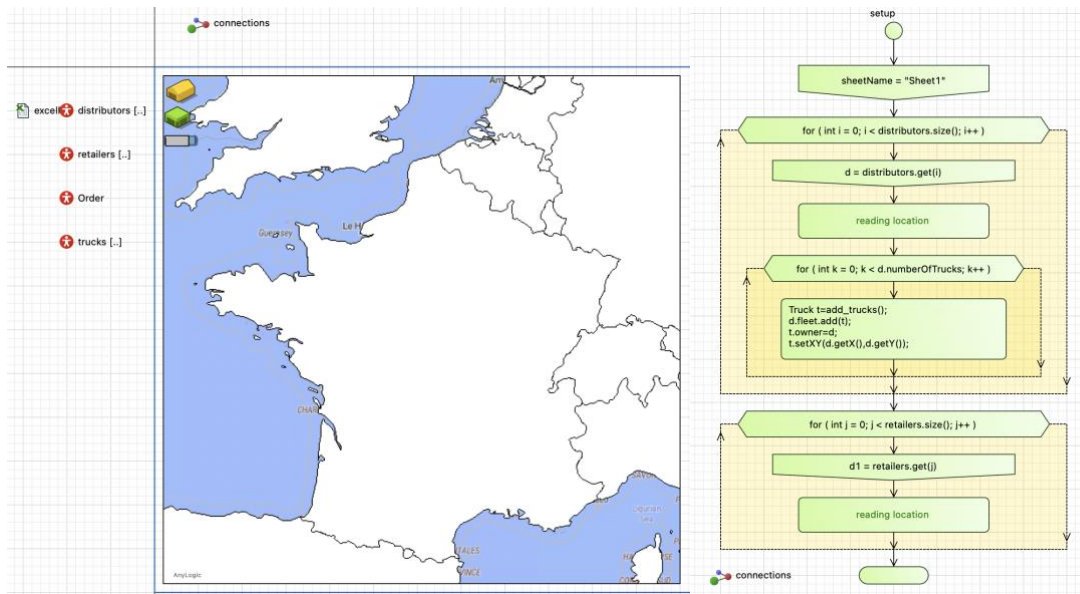


图 7 程序主界面建模

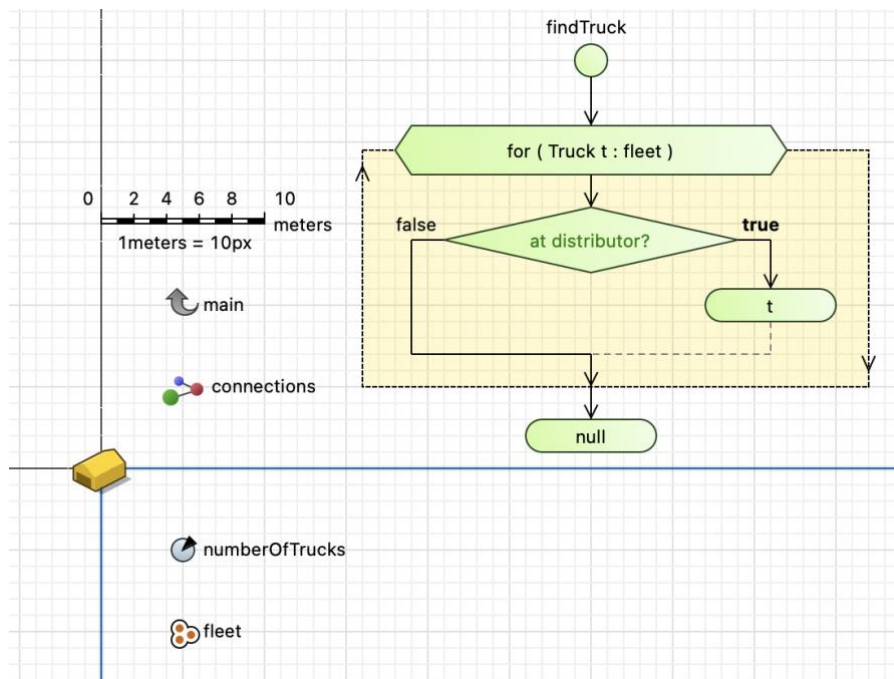


图 8 分销商建模

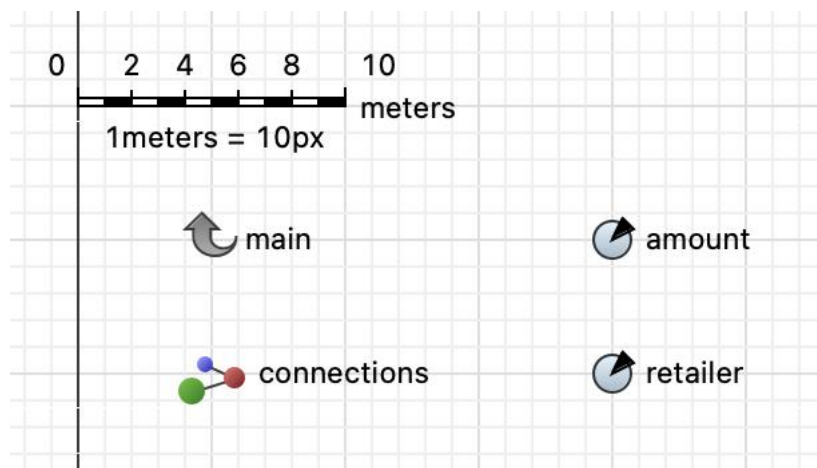


图 9 订单建模

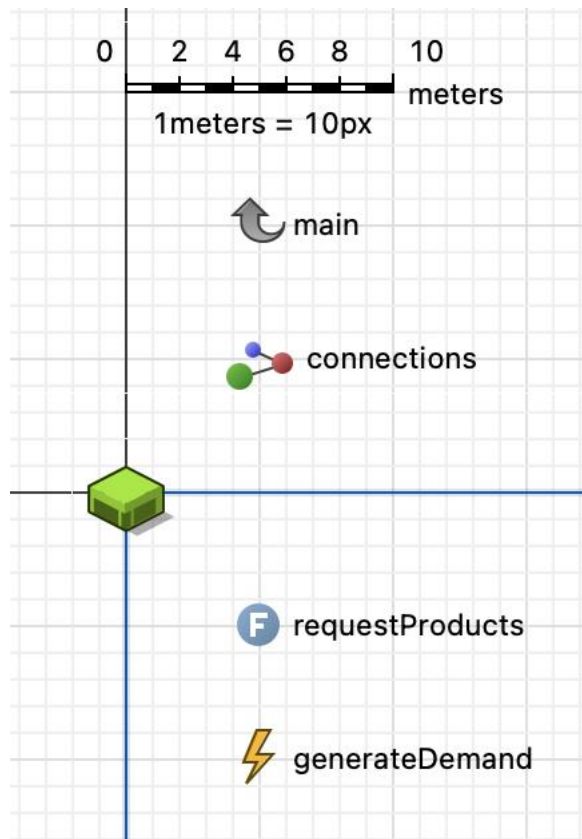


图 10 零售商建模

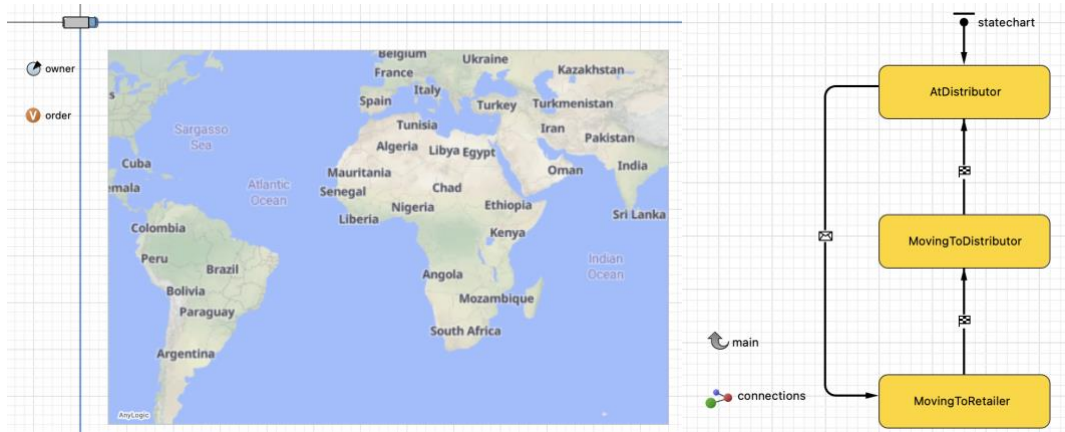


图 11 卡车建模

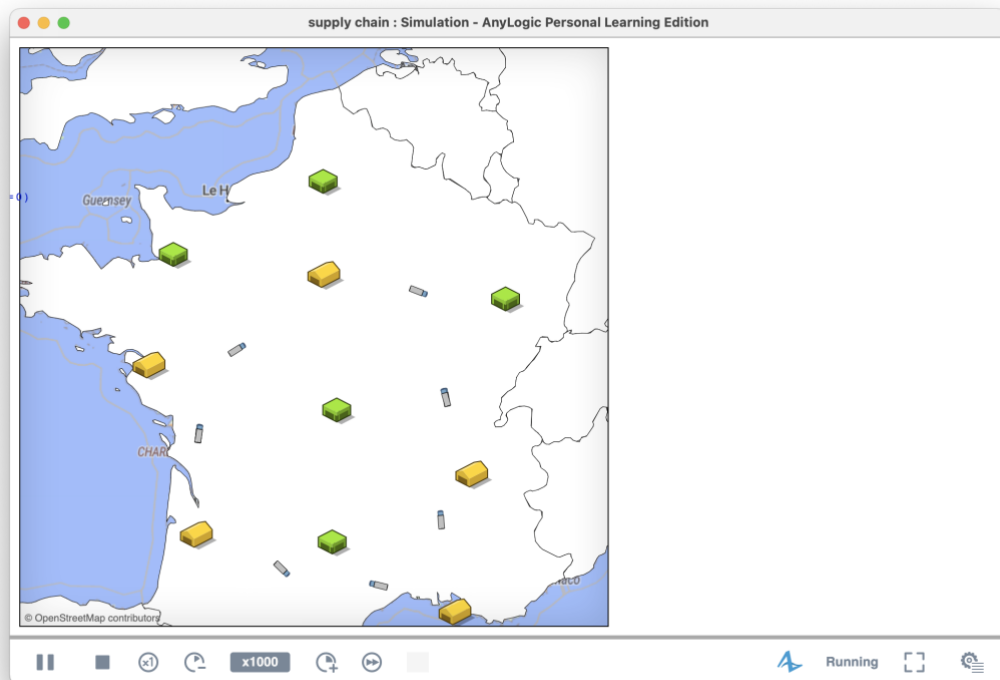


图 12 仿真运行效果

6 实验思考

6.1 关键问题的解答

(1) 请解释幻灯片步骤 6 解释中的“for”循环。

用于从 Excel 表格中循环读取每一个分销商的经纬度位置数据，并赋值给分销商。

(2) 为什么我们将工厂和卡车定义为智能体，而将订单简单地定义为 Java 类的实例？

因为订单用于在不同的智能体之间传递信息，属于虚拟的实体；且订单无动画，不需要在仿真界面中进行展示。

(3) 进入卡车从 **MovingToRetailer** 到 **MovingToDistributor** 状态的变迁的行为中。

这里的所属者是谁？我们可以用 `getOwner()` 替代它吗？为什么

所述者依然是 **Retailer**，因为卡车的实体对象是所属于分销商的卡车集合属性的，而零售商并没有与拥有卡车相关的参数和属性。

卡车的所属者一直没有变。因此，不可以用 `getOwner()` 代替。

6.2 思考题解答

模拟卡车容量有限情况下的订单配送。

(1) 零售商向离其最近的分销商发起订单

使用方法 `getNearestAgentByRoute(population)` 可以找到距离当前实体最近的实体。

对零售商 `Distributors` 的订单需求函数 `requestProducts` 的函数体进行修改。

原函数体如下：

```
1. Order order = new Order(amount, this);
2. /*随机找一个分销商*/
3. Distributor d = main.distributors.random();
4. Truck truck = d.findTruck();
5. if(truck != null)
6. {
7.     truck.order = order;
8.     send(order, truck);
9. }
```

修改后的函数体如下：

```
1. Order order = new Order(amount, this);
2. /*找最近的分销商*/
3. Distributor d = getNearestAgentByRoute(main.distributors);
4. Truck truck = d.findTruck();
5. if(truck != null)
6. {
7.     truck.order = order;
8.     send(order, truck);
9. }
```

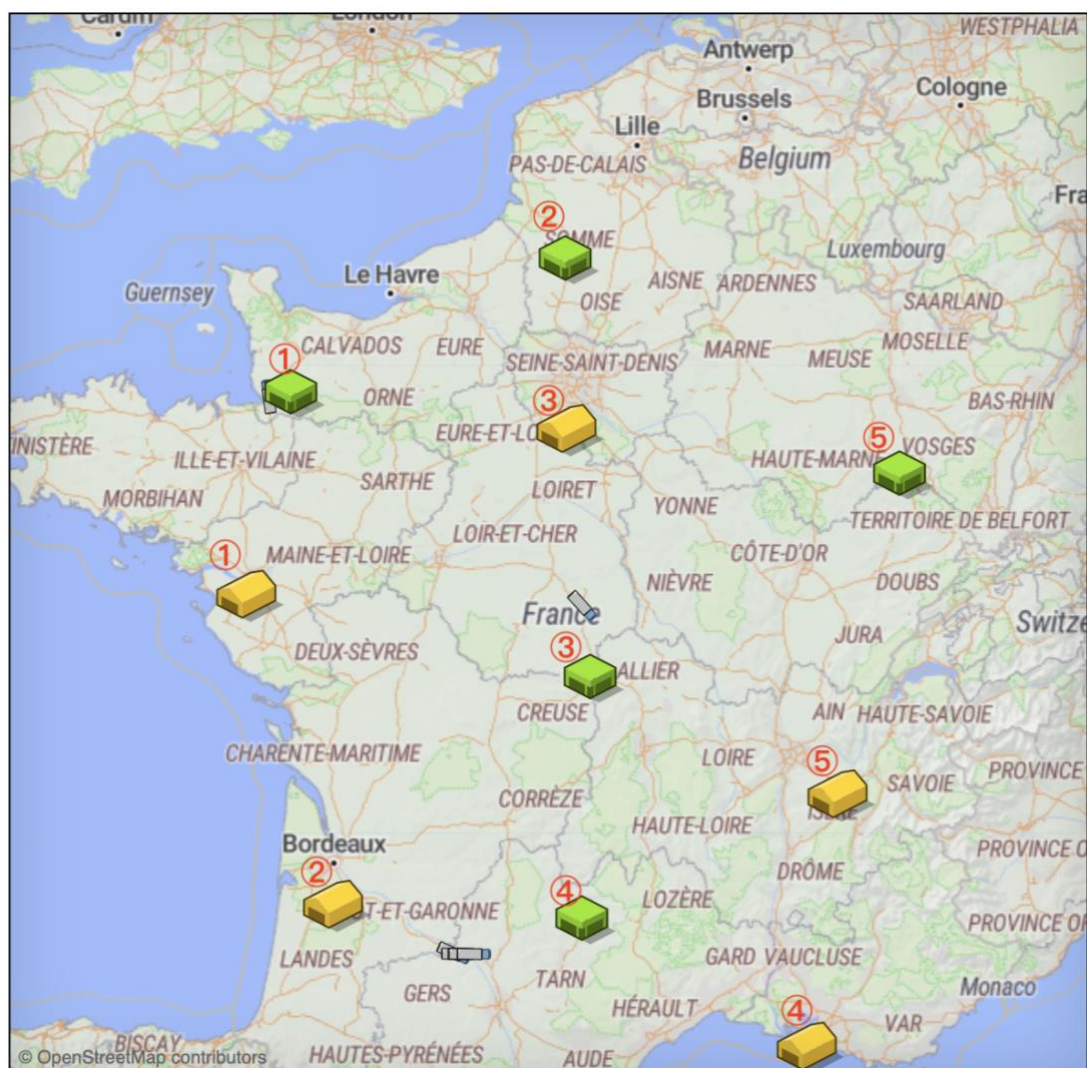



图 13 零售商只给最近的分销商派单情况下的运行示意图

如图所示，在零售商仅给最近的分销商派单的情况下，零售商和分销商之间出现了如表 1 所示的对应情况。

表 1 零售商只给最近的分销商派单情况下零售商、分销商对应关系表

分销商	零售商
1	1
2	4
3	2、3、5
4	无
5	无

结果显示，虽然每个零售商都能由离其最近的分销商为其供货，但整体存在两个分销商无货供货，一个分销商同时需要为三个零售商供货的缺陷。

(2) 分销商接收订单存入其订单列表（包括分配时间、目的分销商、订单数量）

在订单能体类中添加参数 `ordertime`，设定类型为 `Date`，用于记录订单发起的时间。

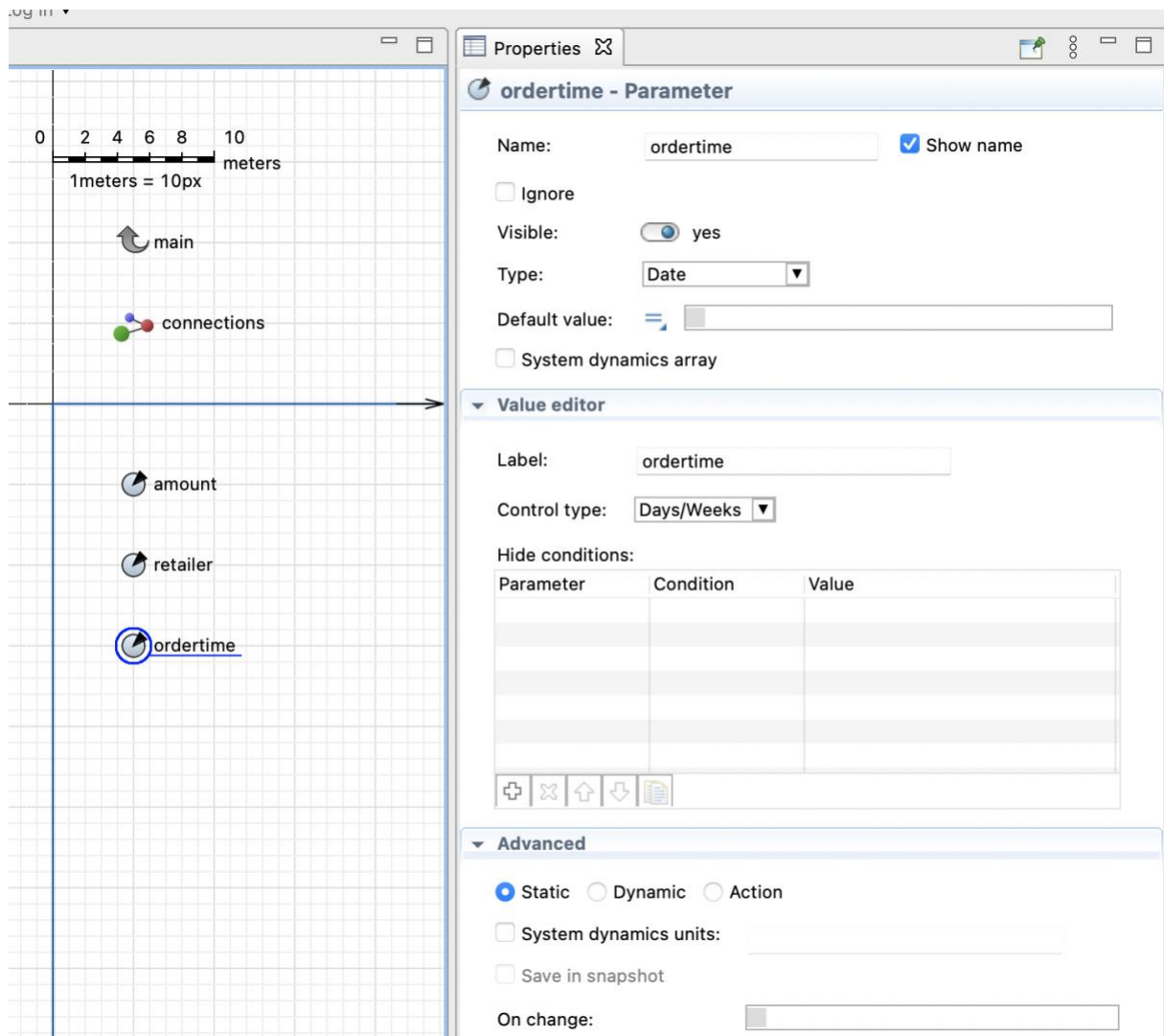


图 14 添加订单时间参数

添加订单数量变量 `numOforders`，设置类型为 `int`，用于分销商当前积压订单的数量。并添加文本，用于在仿真程序运行时显示供应商名称和积压订单数量。完成后的示意图如图 15 所示。

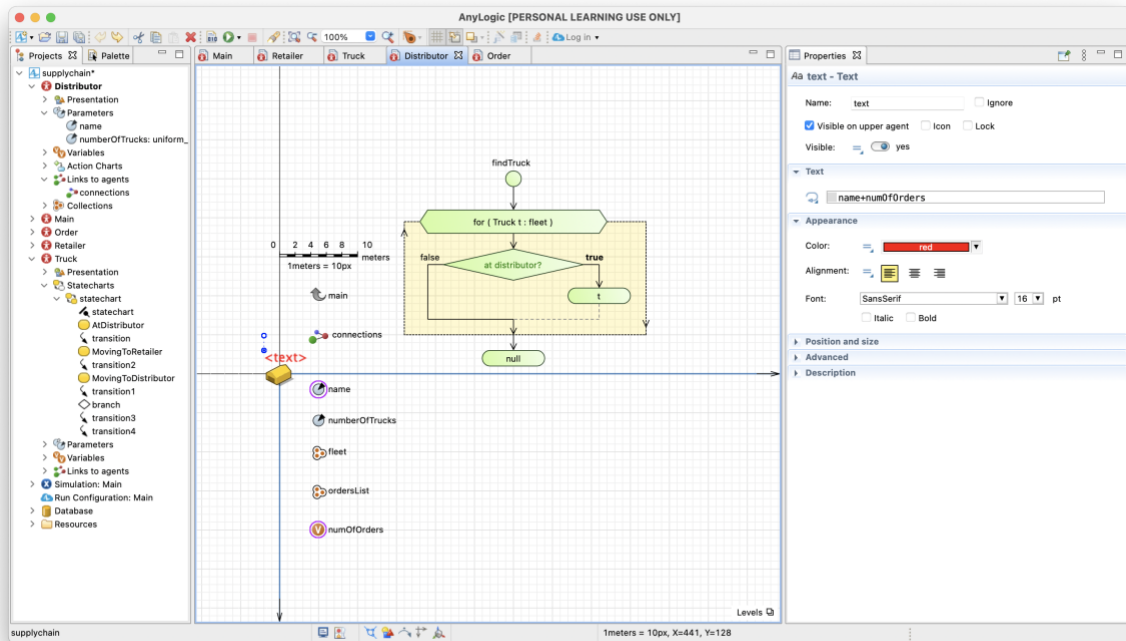


图 15 分销商添加积压订单数量和仿真运行时显示的文本

修改零售商智能体类的 `requestProduct` 函数，令其向分销商发送订单信息，而非直接向货车发送信息。修改后的函数体如下：

```
1. Order order=new Order(amount, this, date());
2. Distributor d=getNearestAgentByRoute(main.distributors);
3. send(order, d);
```

在分销商的智能体类中添加一个集合 `ordersList`，用于存放从零售商处传递过来的订单。设置其元素类型为订单类型（`Order`）。设置其集合类型为 `LinkedList`（首尾相连的线性链表）。

图 16 订单列表属性

- (3) 分销商按照先来先服务原则将订单随机分配给空闲卡车，卡车存在最大装载量 (100)

将分销商的智能体类的 connections 对象的 Message type 设置为订单类型 (Order)。在其 On message received 属性中添加如下代码块。

```
1. ordersList.addLast(msg);
2. Truck t = findTruck();
3. if(t!=null){
4.   send(ordersList.getFirst(), t);
5.   ordersList.removeFirst();
6. }
```

完成后如图 17 所示。

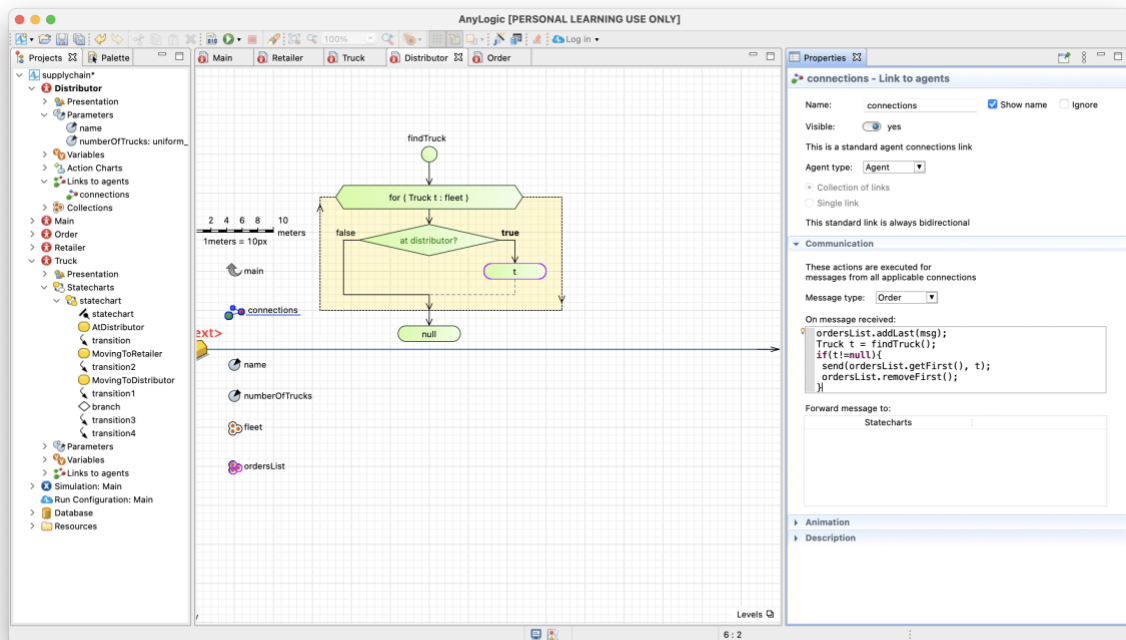


图 17 分销商智能体类用 connections 接收传递过来的订单数据

修改卡车的行动图，在 MovingToDistribution 状态后添加触发条件分支。相关属性和行动代码设置分别如图 18 图 19 所示。

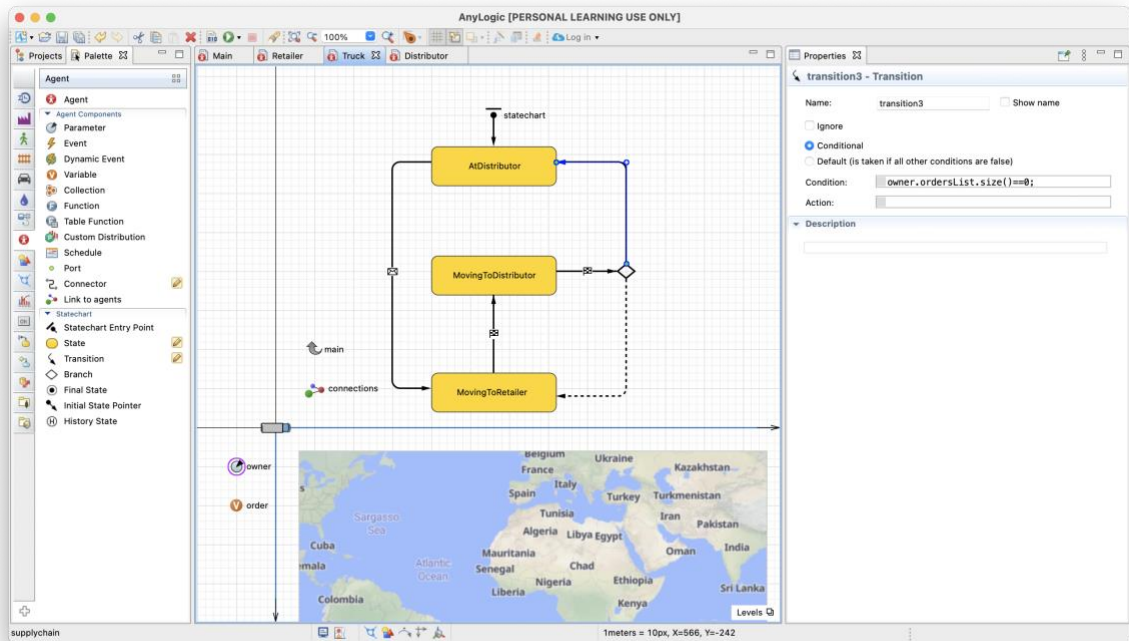


图 18 卡车行动图修改 1

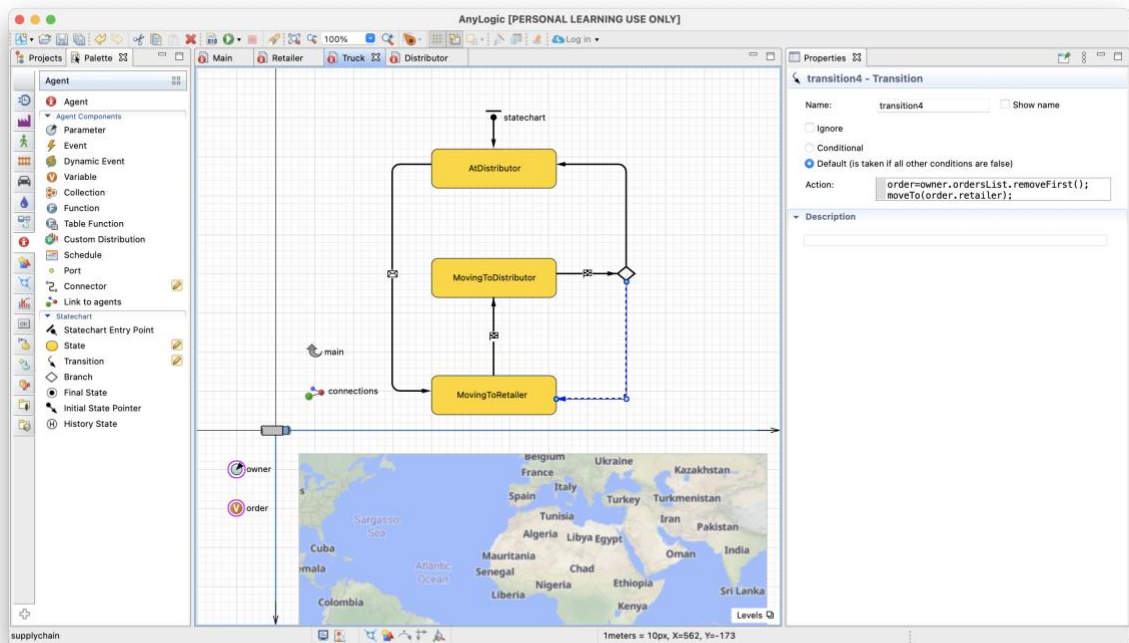


图 19 卡车行动图修改 2

为卡车智能体类添加参数“最大承载量”（maxCapacity），设置类型为 int，默认值为 100。设置完成后的示意图如图 20 所示。

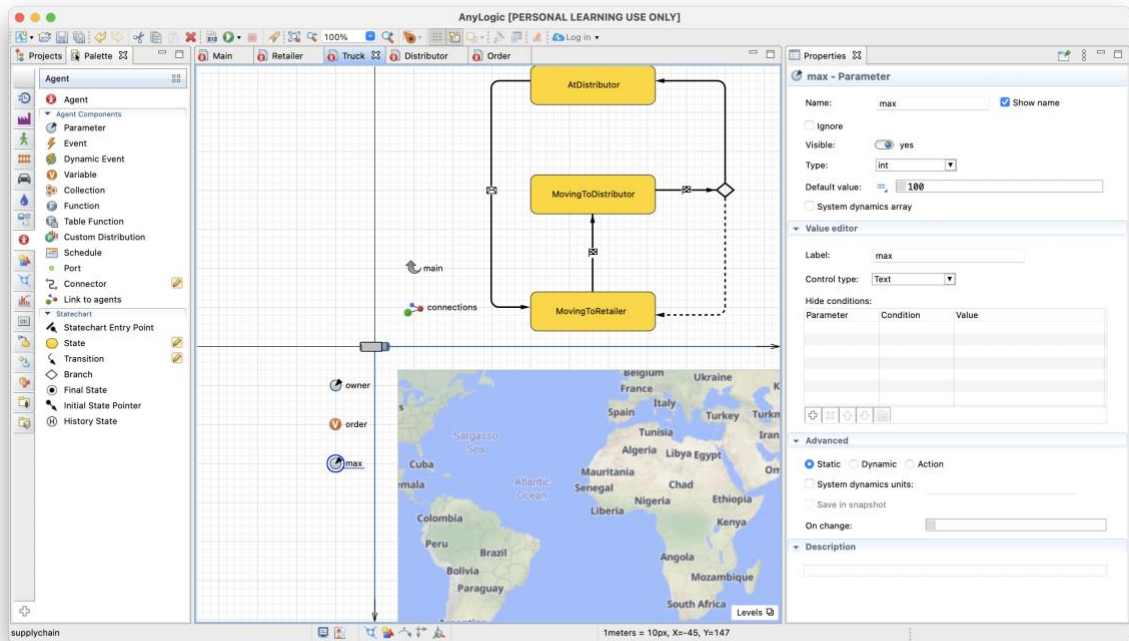


图 20 卡车智能体类添加最大承载量

- (4) 若存在空闲卡车，且装载量 $<$ 容量，则直接配送后返回；若装载量 $>$ 容量，则将未完成部分视作新订单随机分配至剩余空闲卡车
- (5) 若无空闲卡车，则继续等待

调整零售商随机产生订单的数量分布为 $[5, 200]$ （原先为 $[5, 20]$ ，均小于卡车的最大承载量，仿真时候体现不出程序逻辑）。如图 21 所示。

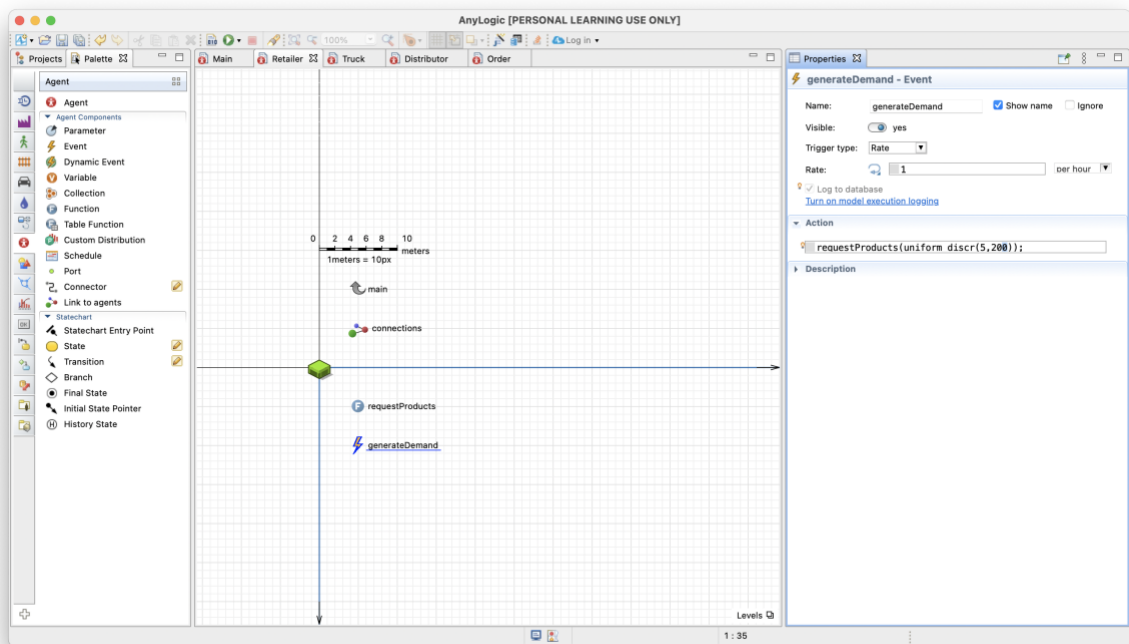


图 21 调整零售商随机产生订单的数量分布

为卡车智能体类添加当前承载量变量（loadAmount），设定类型为 int。如图 22 所示。

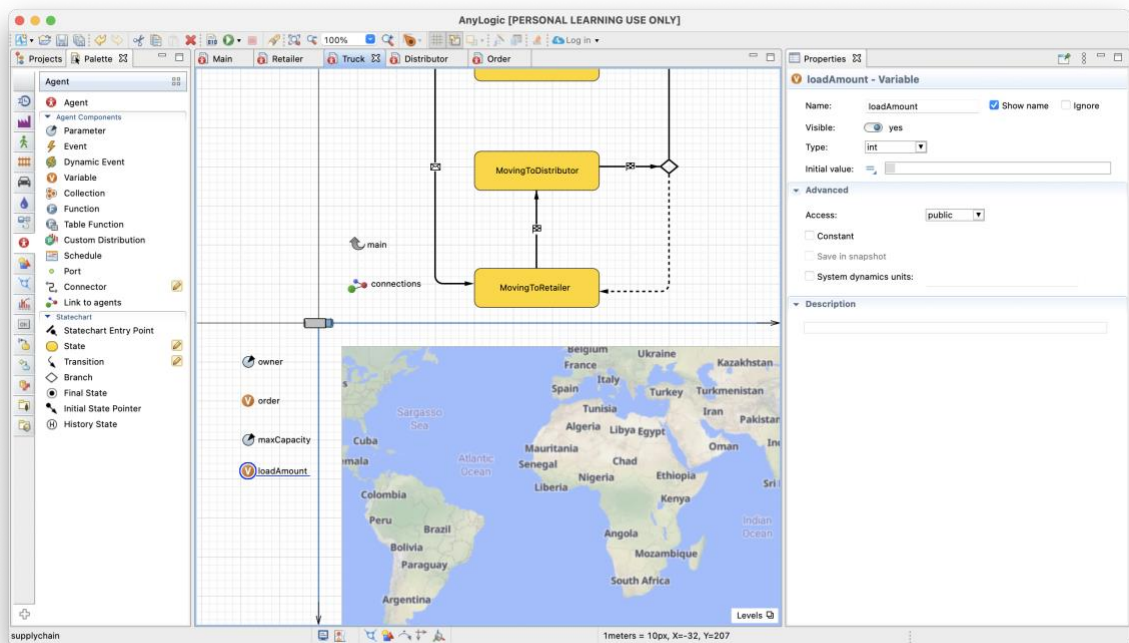


图 22 卡车智能体类添加当前承载量变量

修改卡车行动图第一阶段的触发条件的行动代码，相关代码块如下：

```
1. order=msg;
```


2. loadAmount=order.amount;
3. moveTo(order.retailer.getX(),order.retailer.getY());

示意图如图 23 所示。

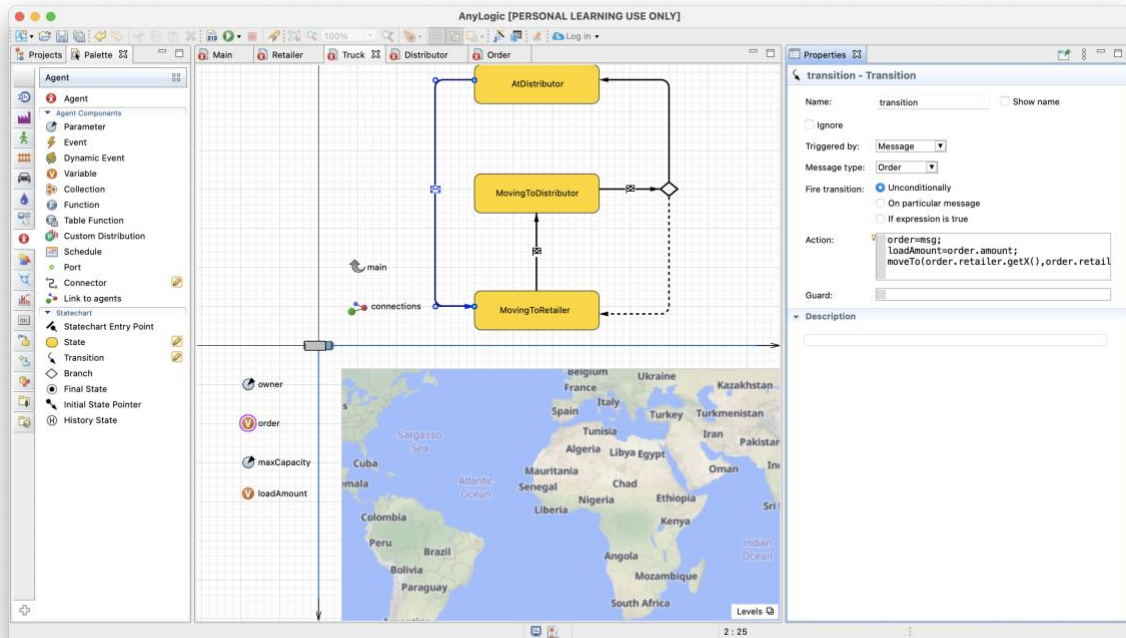


图 23 修改卡车行动图第一阶段的触发条件的行动代码

修改分销商的 connections 中的 On message received, 修改完后的函数体如下:

```

1. numOfOrders++;
2. ordersList.addLast(msg);
3. Truck t = findTruck();
4. if(t!=null){
5.   Order order = ordersList.getFirst();
6.   if (order.amount <= t.maxCapacity){
7.     send(order, t);
8.     ordersList.removeFirst();
9.     numOfOrders--;
10.  }
11.  else {
12.    int rest = order.amount - t.maxCapacity;
13.    order.amount = t.maxCapacity;
14.    send(order, t);
15.    ordersList.getFirst().amount = rest;
16.  }
17. }

```

示意图如图 24 所示。

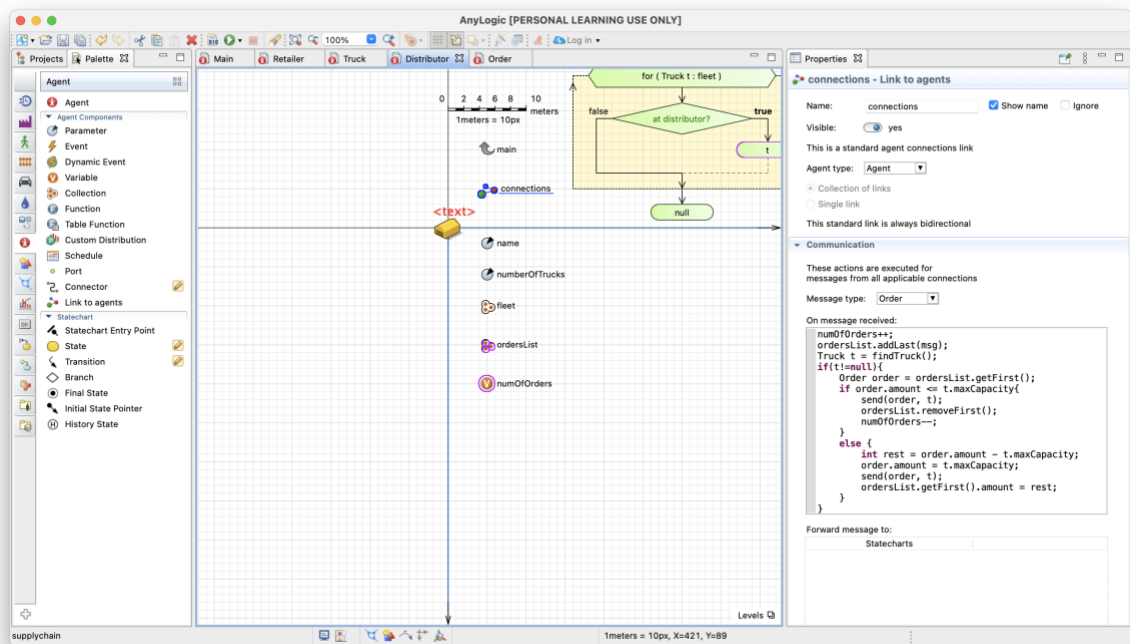


图 24 分销商处理订单的算法

6.3 完成思考题后的仿真

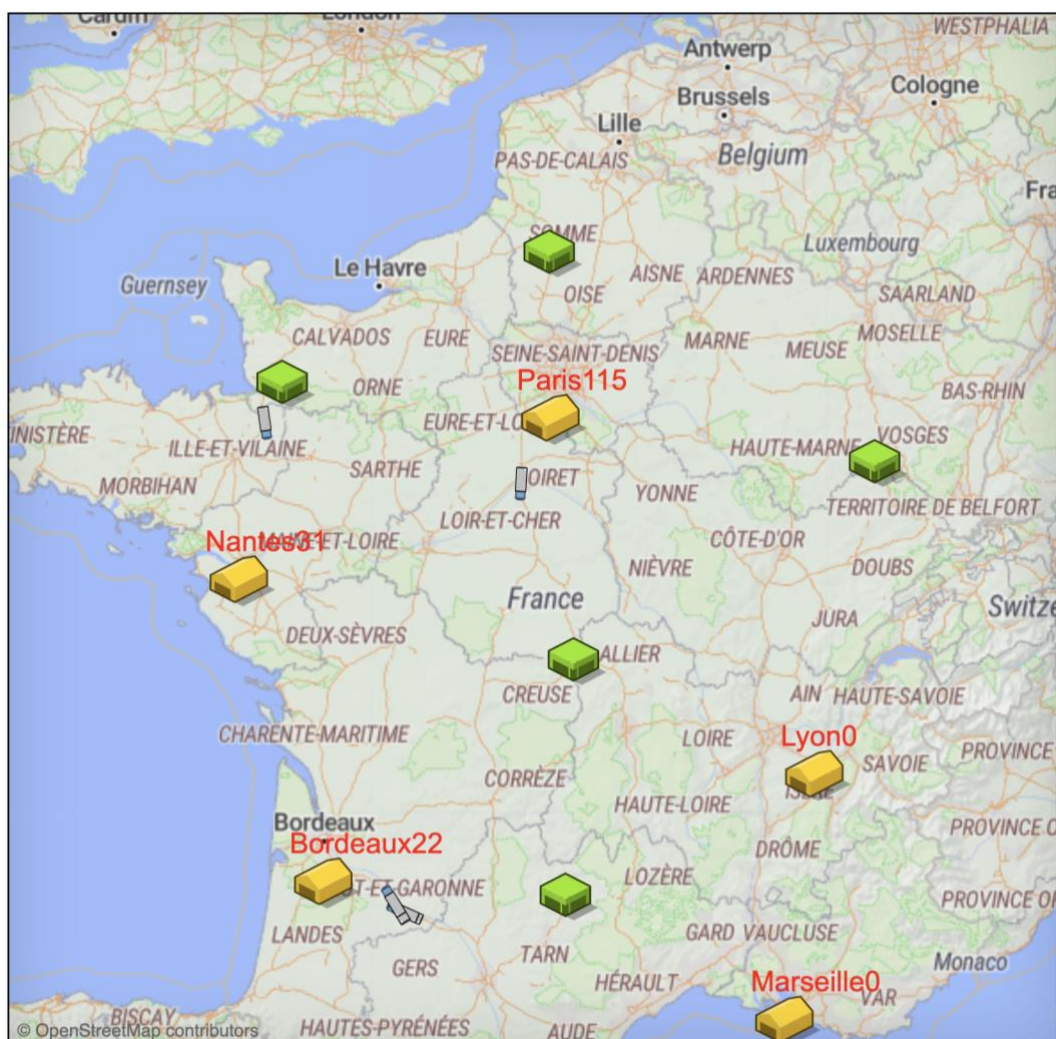


图 25 完成思考题后的仿真示意图

*注：在本次仿真中对部分仿真模型的参数做了调整，以更好地进行展示。具体调整的参数如表 2 所示。

表 2 仿真模型参数调整表

参数类型	原先值	修改后的值
模型时间单位	秒	小时
GIS Map Shapefiles	勾选 Visible	不勾选 Visible
零售商 generateDemand 的速率	1 秒 1 个	1 小时 1 个

7 参考文献

- [1] 赵瑞春等. 《基于 Anylogic 的系统动力学仿真与建模》
- [2] Bilibili. 北京格瑞纳官方账号. 20200306-基于 GIS 的网络配送建模
- [3] Anylogic Help. Supply Chain GIS (Agents).
<https://anylogic.help/tutorials/gis/index.html>