

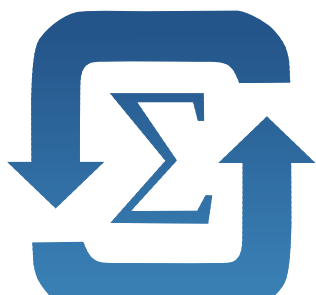
# Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное  
учреждение высшего образования  
«НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»



Кафедра теоретической и прикладной информатики

Лабораторная работа № 2  
по дисциплине «Компьютерное моделирование»



ФАКУЛЬТЕТ:	ПМИ
ГРУППА:	ПМИ-61
СТУДЕНТЫ:	Ершов П.К., Мамонова Е.В., Цыденов З.Б.
ПРЕПОДАВАТЕЛЬ:	Черникова О. С. Карманов В. С.

Новосибирск

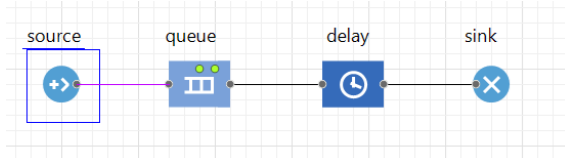
2020

## 1. Цель работы

Научиться строить модели систем массового обслуживания.

## 2. Ход работы

### 1. Строим первичную модель банковского офиса



**source - Source**

Имя:  ☒ Отображать имя ☐ Исключить

Прибывают согласно:

Интенсивность прибытия:  в минуту

Считать параметры агентов из БД: ☐

За 1 раз создается несколько агентов: ☐

Ограниченное кол-во прибытий: ☐

Местоположение прибытия:

---

**queue - Queue**

Имя:  ☒ Отображать имя ☐ Исключить

Вместимость:

Максимальная вместимость: ☐

Место агентов:

**Специфические**

Очереди:

Разрешить уход по таймауту: ☐

Разрешить вытеснение: ☐

Вернуть агента в исходную точку: ☒

Включить сбор статистики: ☒

---

**delay - Delay**

Имя:  ☒ Отображать имя ☐ Исключить

Тип задержки: ☒ Определенное время ☐ До вызова функции stopDelay()

Время задержки:  минуты

Вместимость:

Максимальная вместимость: ☐

Место агентов:

**Специфические**

Выталкивать агентов: ☐

Вернуть агента в исходную точку: ☒

Включить сбор статистики: ☒

### 2. Задаём параметры эксперимента (запускаем в 10 режиме)

**Simulation - Простой эксперимент**

Имя:  ☐ Исключить

Агент верхнего уровня:

Максимальный размер памяти:  M6

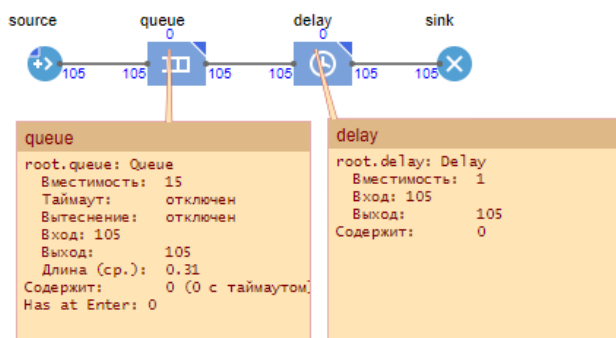
**Модельное время**

Режим выполнения: ☐ Виртуальное время (максимальная скорость) ☒ Реальное время со скоростью

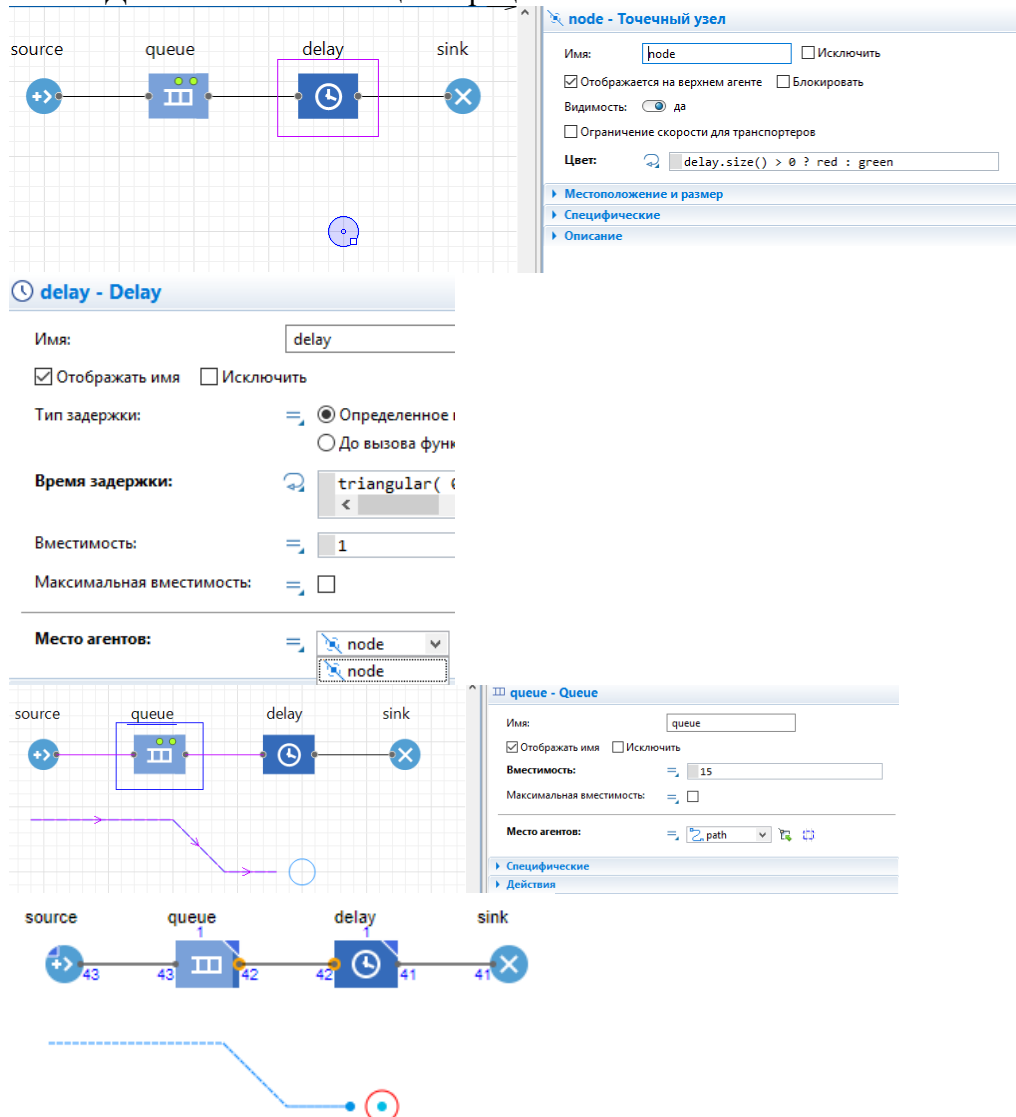
Остановить:

### 3. Проверка работоспособности модели

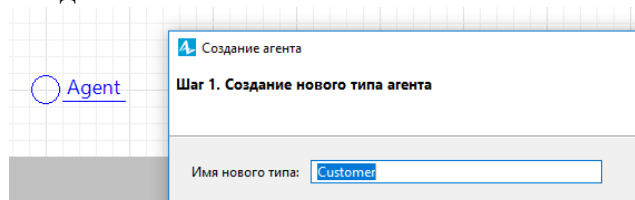
Bank\_office : Simulation - AnyLogic Professional [ТОЛЬКО В ЦЕЛЯХ ОЗНАКОМЛЕНИЯ]

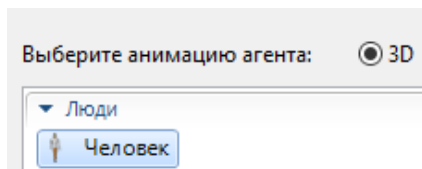


### 4. Добавляем 3d анимацию процесса

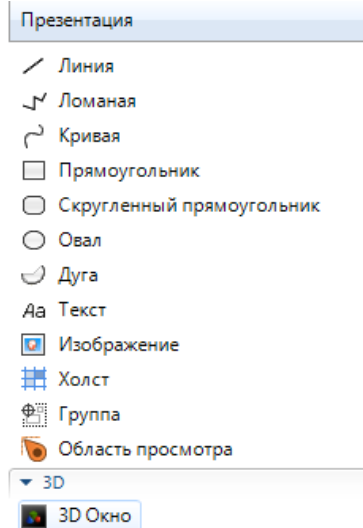


Создаём нового агента:

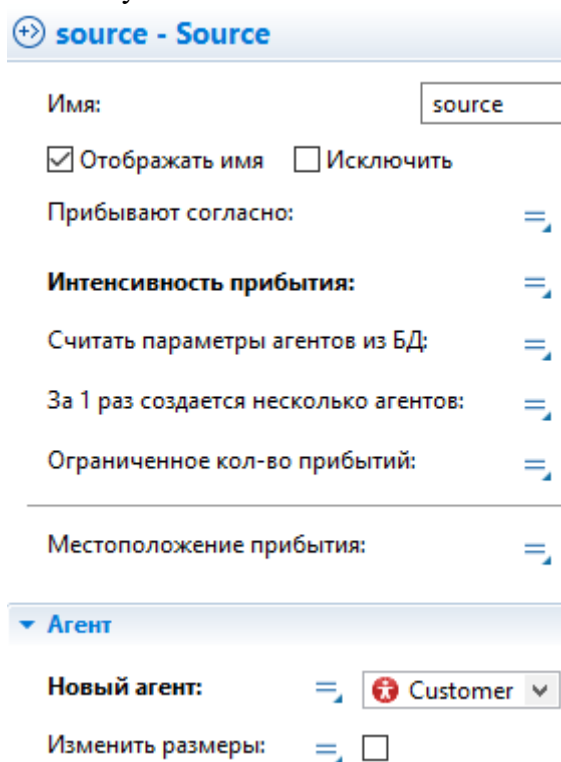




Добавляем 3D окно из палитры Презентация:



Используем новый тип агента в source:



Поместим 3D объект банкомат из палитры 3D объекты (вкладка супермаркет) в точечный узел:



**atm1 - 3D Объект**

Имя:  ☐ Исключить

☒ Отображается на верхнем агенте ☐ Блокировать

Видимость: ☒ да

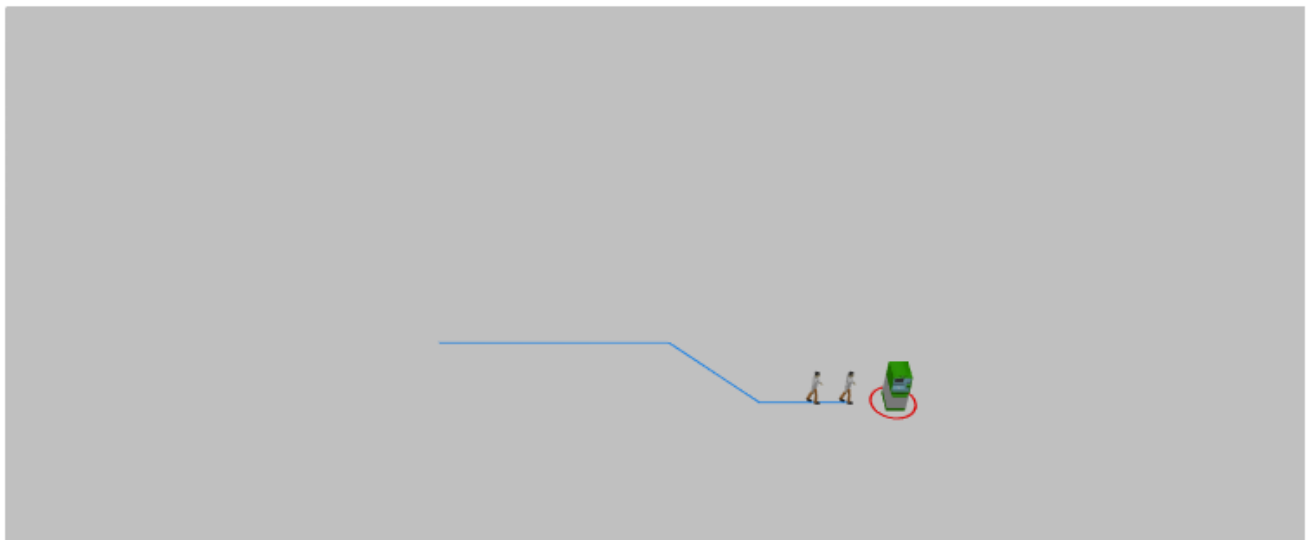
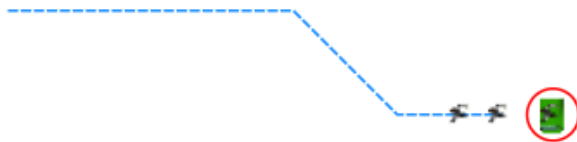
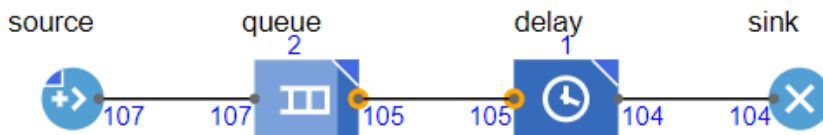
Файл:  ...

☐ Автоматически изменять размер для соответствия масштабу агента

Доп. масштабирование:

Ориентация: ☒ ☐ ☐ ☐

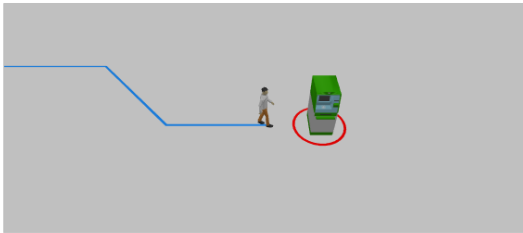
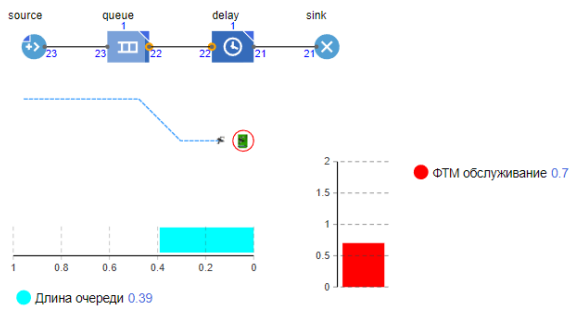
Проверим работоспособность:



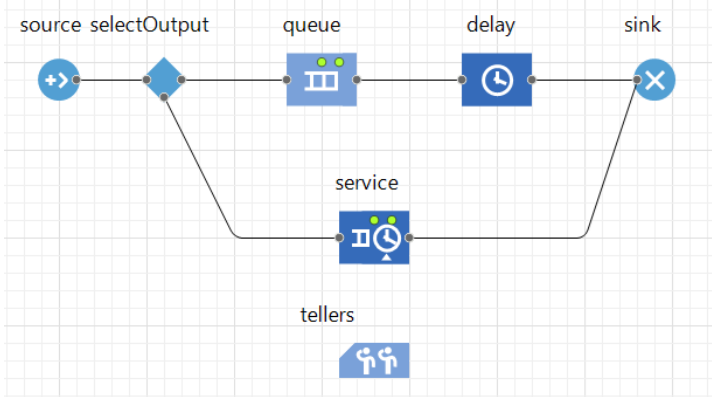
## 5. Добавляем графики



Проверяем работоспособность:



## 6. Добавляем область с консультантами



**tellers - ResourcePool**

Имя:  ☒ Отображать имя

☐ Исключить

Тип:

Количество задано:

Количество ресурсов:

При уменьшении кол-ва:

---

Новый ресурс:

[создать другой тип](#)

Скорость:  м/с

Базовое местоположение (узлы):

---

Обслуживание, аварии, смены, перерывы

Специфические

Добавить ресурсы в: ☒ Популяцию по умолчанию ☐ Другую популяцию агентов

Включить сбор статистики: ☒

**service - Service**

Имя:  ☒ Отображать имя

☐ Исключить

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

Набор(ы) ресурсов:

Вместимость очереди:

Максимальная вместимости: ☐

Время задержки:

Пересылать захваченные ресурсы: ☐

Место агентов (queue):

Место агентов (delay):

► Приоритеты / вытеснение

▼ Специфические

Задать выбор ресурса: ☐

Политика выбора ресурса:

Разрешить выход по таймауту: ☐

Разрешить вытеснение: ☐

Вернуть агента в исходную точку: ☒

Включить сбор статистики: ☒

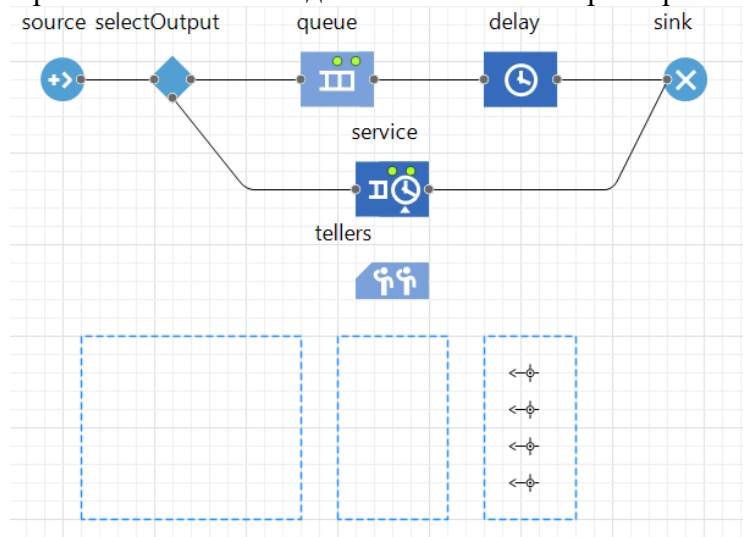
7. Добавляем анимацию второго канала

Создадим три Прямоугольных узла:

Левый назовём waitingArea;

Средний customerPlaces;

Правый tellerPlaces и добавим в него 4 аттрактора:

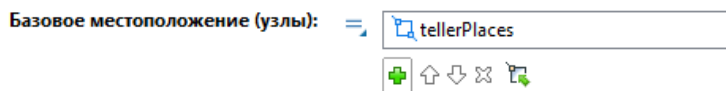


Укажем в service места агентов:

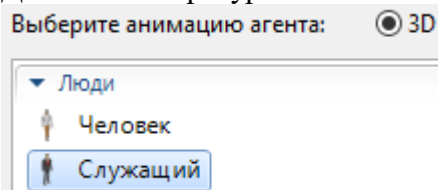
Место агентов (queue):

Место агентов (delay):

Укажем в ResourcePool место агентов:



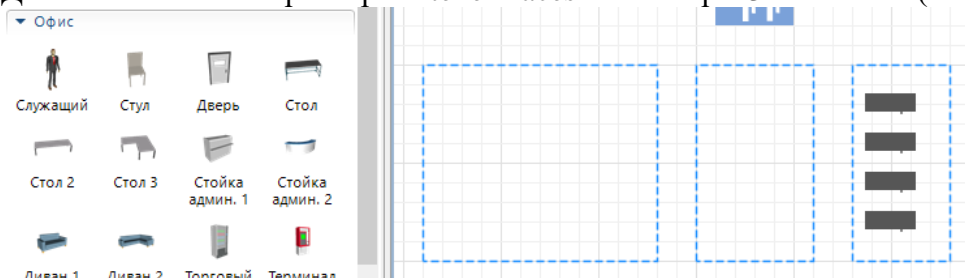
Добавим тип ресурса Teller и выберем ему модель Служащий:



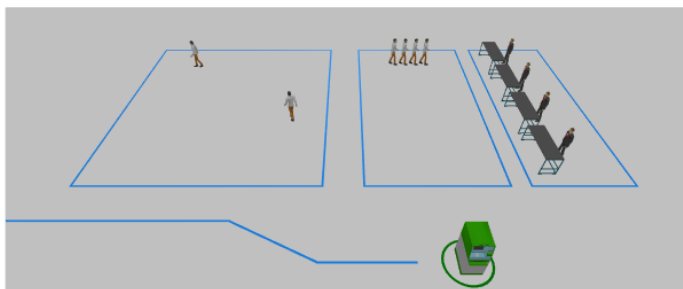
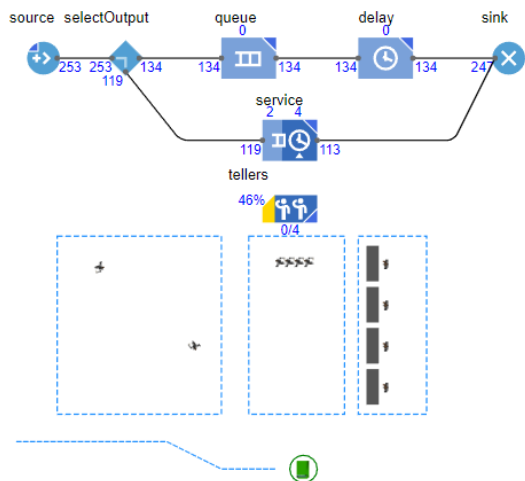
Выберем в ResourcePool Teller как ресурс:



Добавим столы в аттракторы в tellerPlaces из палитры 3D объекты (вкладка Офис):



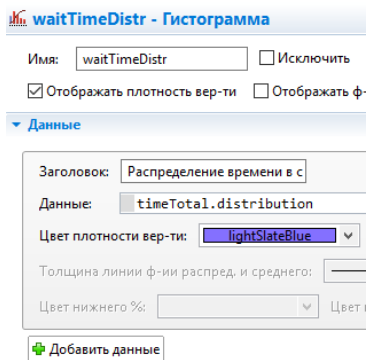
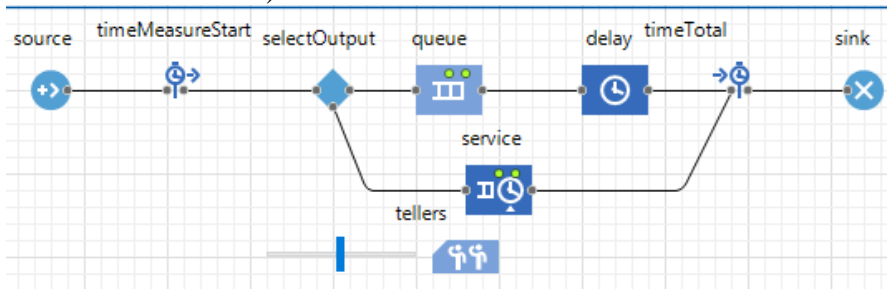
Проверяем работоспособность:



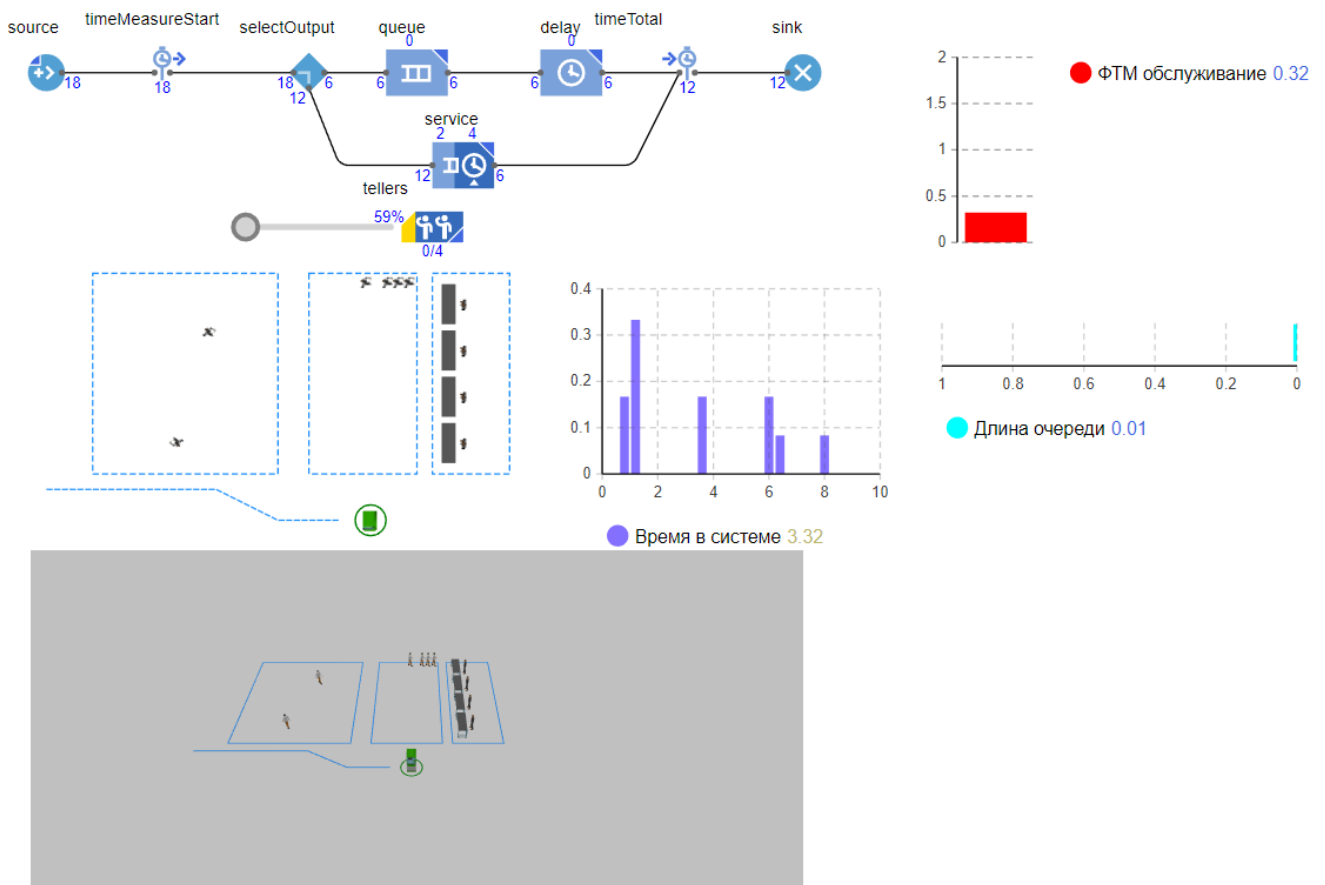


## 8. Добавляем сбор параметров системы

Добавляем блоки измерения времени (так же добавим бегунок, для удобного управления количеством tellers):

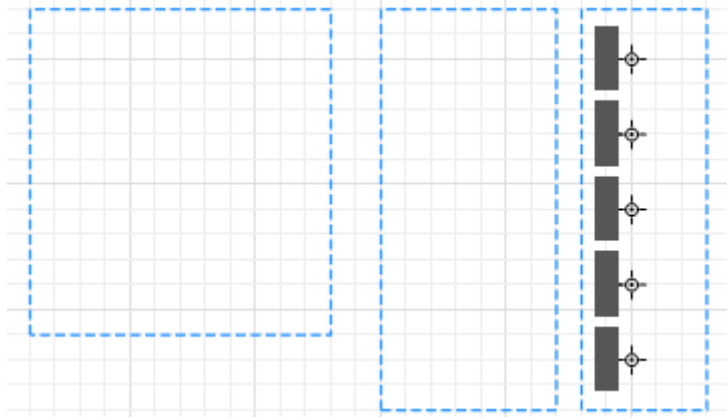


Проверка работоспособности:



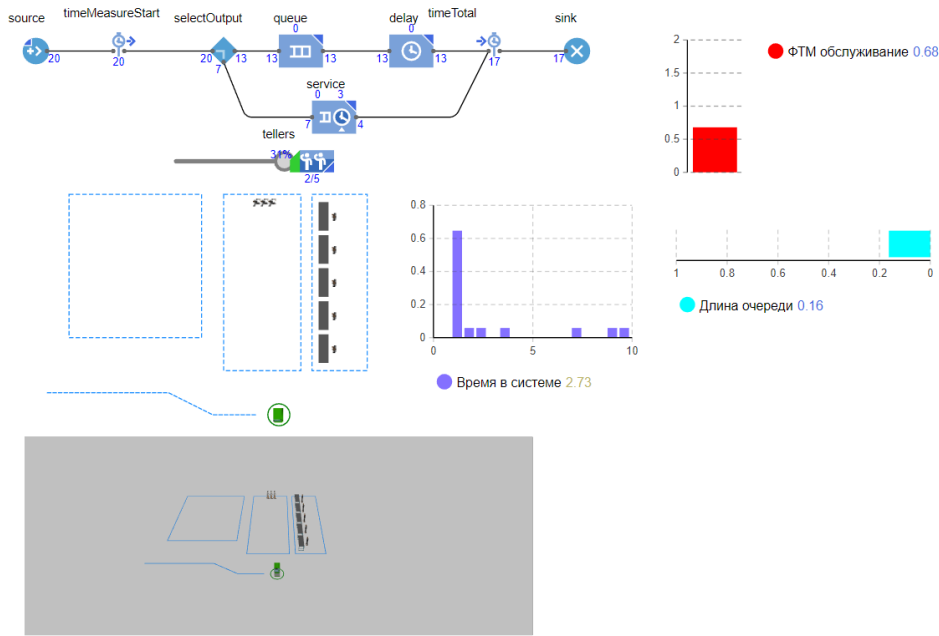
9. Тестирование системы

9.1. Результат работы системы с пятью (5) консультантами



При работе 4 консультантов среднее время в системе составляет около 4 минут.

При добавлении пятого консультанта время в системе сократилось:




9.2. Тестирование системы на поиск предельной интенсивности потока заявок без потери работоспособности

Для этого изменим параметр интенсивности в source:  
Тестировать будем для четырёх консультантов.

№ теста	Интенсивность	Результат
1	1	Система потеряла работоспособность 22 мая 2020 г.
2	0.9	Система потеряла работоспособность 23 июня 2020 г.
3	0.8	Система сохраняет работоспособность.
4	0.89	Система сохраняет работоспособность.
5	0.899	Система сохраняет работоспособность.
6	0.8999	Система потеряла работоспособность 23 июня 2020 г

Таким образом, можно считать предельной интенсивностью потока заявок 0.899.

## Пример потери работоспособности:

 Ошибка при выполнении дискретного события

×

Логическая ошибка в модели:

root.source:

Агент не смог покинуть этот порт: root.source.out в момент времени 191,987.148 / дату 23 июн. 2020 г., 7:47:08  
(текущее модельное время: 191,987.184)

Увеличьте вместимости объектов и/или пропускную способность последующих объектов

Подробности смотрите в [Консоли](#)

## 10. Поиск оптимального количества консультантов для определённой интенсивности

Интенсивность	Оптимальное количество консультантов (количество, при котором система не сломается и консультанты будут загружены не более чем на 60 процентов)
1	6
0.9	5
0.8999	5

## 3. Выводы

В ходе проведённой работы была построена модель банковского офиса с одним банкоматом и четырьмя консультантами. Так же был поставлен эксперимент с целью определения предельной интенсивности потока заявок в минуту, при которой система сохранит свою работоспособность.