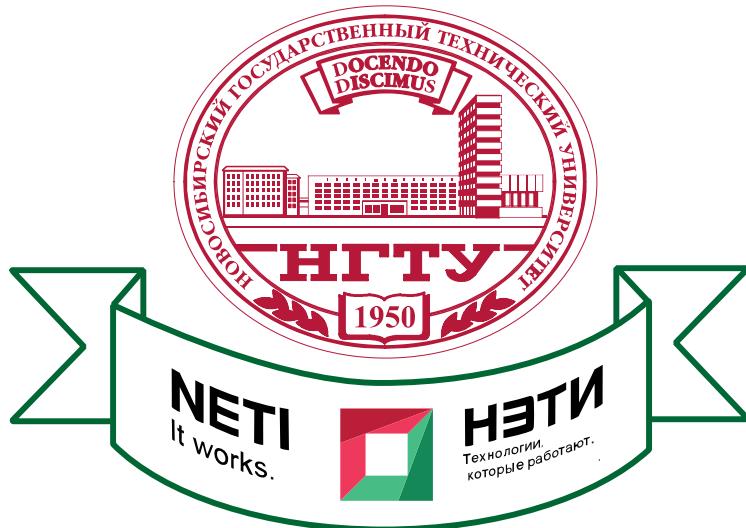


Министерство науки и высшего образования
Российской Федерации

Федеральное государственное бюджетное
образовательное учреждение высшего образования

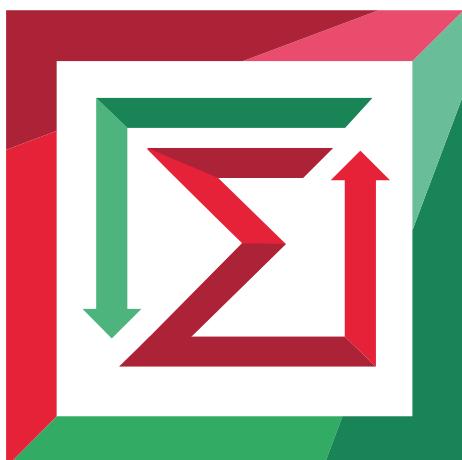
«НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»



Теоретической и прикладной информатики

Лабораторная работа № 1
по дисциплине «Компьютерное моделирование»

МОДЕЛИРОВАНИЕ СИСТЕМЫ ОБСЛУЖИВАНИЯ КЛИЕНТОВ



Факультет: ПМИ
Группа: ПМИ-02
Студент: Сидоров Даниил,
Дюков Богдан
Преподаватель: Карманов Виталий Сергеевич

Новосибирск

2026

1. Формулировка задания

Смоделировать одноканальную и двухканальную систему массового обслуживания. Протестировать модель двухканальной системы массового обслуживания, определив оптимальное количество консультантов. Определить максимальное значение интенсивности входного потока заявок, при котором система сохраняет работоспособность.

В банковский офис обращаются клиенты. Офис представляет собой пункт обслуживания, в котором установлен банкомат и консультируют $n = 3$ работников банка. Банкомат единовременно может обслуживать только одного клиента. Клиенты прибывают с интенсивностью $\lambda = 0.95$. Одновременно в офисе может находиться не более $N = 35$ клиентов. Распределение времени работы банкомата описывается равномерным законом распределения.

2. Цели работы

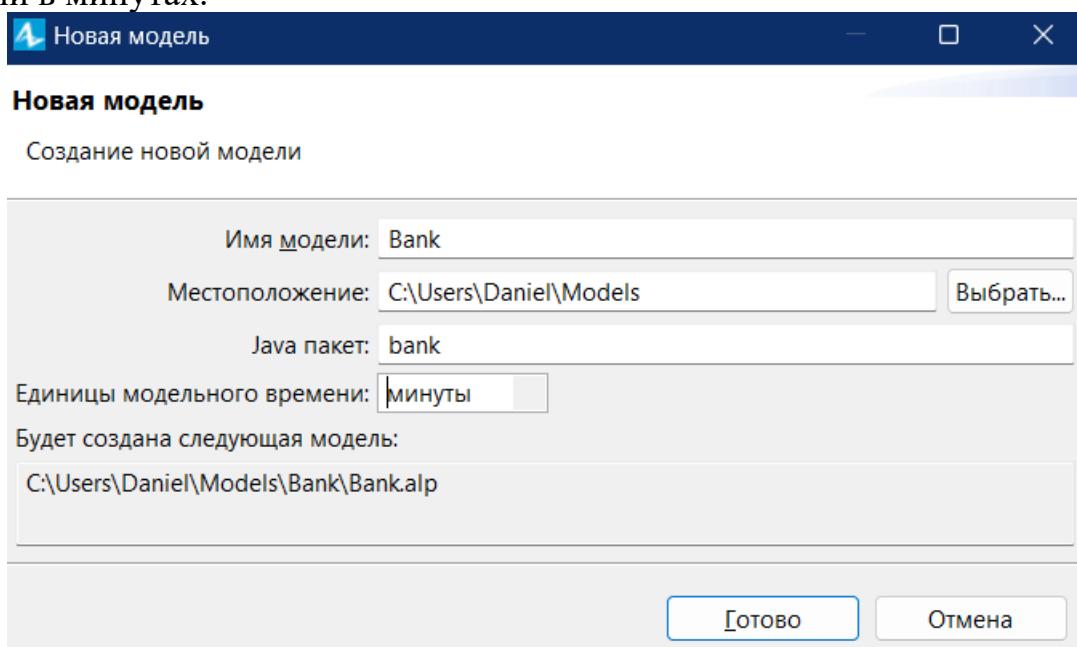
Построить и проанализировать работу одноканальной и двухканальной модели системы обслуживания клиентов.

3. Описание выполненных действий

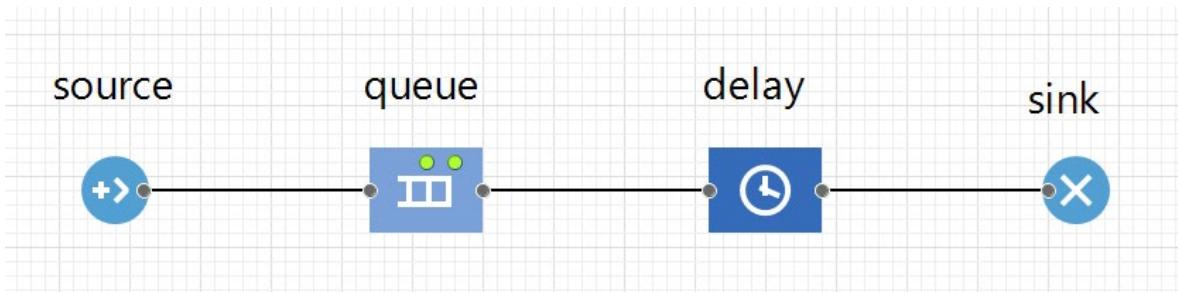
Моделирование одноканальной системы массового обслуживания

Создание модели

Создадим новую модель с именем Bank и зададим единицы модельного времени в минутах.



Создадим диаграмму процесса потока клиентов банка и выполним настройку модели с помощью мастера создания модели. Добавим блоки библиотеки моделирования процессов на диаграмму и соединим их между собой.



Source – источник заявок. Объект генерирует агентов определенного типа. В нашем случае такими агентами являются клиенты банка.

Queue – очередь заявок, ожидающих обслуживания. Объект моделирует очередь клиентов банка, ожидающих освобождения банкомата.

Delay – узел обслуживания. Объект задерживает агента на определенный промежуток времени. В лабораторной работе узлом обслуживания является банкомат, у которого клиент банка проводит некоторое время для выполнения необходимой операции.

Sink – элемент, принимающий отработанные заявки. Объект уничтожает поток клиентов.

Далее перейдем к заданию вышеуказанных объектов Source, Queue и Delay.

Настройка объекта Source

Согласно постановке задачи:

- заявки прибывают с интенсивностью $\lambda = 0.95$;
- количество заявок, прибывающих за один раз, равно единице.

Имя:	<input type="text" value="source"/>	<input checked="" type="checkbox"/> Отображать имя
<input type="checkbox"/> Исключить		
Прибывают согласно:	<input type="button" value="Интенсивности"/>	
Интенсивность прибытия:	<input type="text" value="0.95"/>	<input type="button" value="в минуту"/>
Считать параметры агентов из БД:	<input type="checkbox"/>	
За 1 раз создается несколько агентов:	<input type="checkbox"/>	
Ограниченнное кол-во прибытий:	<input type="checkbox"/>	
Местоположение прибытия:	<input type="button" value="Не задано"/>	

Настройка объекта Queue

Согласно заданию вместимость очереди равна 35, т. е. в очереди могут находиться не более 35 человек. Будем считать, что в очереди действует порядок обслуживания FIFO (First In – First Out), означающий, что клиент, первым подошедший к банкомату, первым и будет обслужен.

Имя:	<input type="text" value="queue"/>	<input checked="" type="checkbox"/> Отображать имя
<input type="checkbox"/> Исключить		
Вместимость:	<input type="text" value="35"/>	
Максимальная вместимость:	<input type="text"/>	<input type="checkbox"/>
Место агентов:	<input type="text"/>	
▼ Специфические		
Очередь:	<input type="text" value="FIFO"/>	
Разрешить уход по таймауту:	<input type="checkbox"/>	
Разрешить вытеснение:	<input type="checkbox"/>	
Вернуть агента в исходную точку:	<input checked="" type="checkbox"/>	
Включить сбор статистики:	<input checked="" type="checkbox"/>	

Настройка объекта Delay

Интервал времени работы банкомата подчиняется равномерному закону распределения с параметрами $x_{min} = 1$, $x_{max} = 1,8$. Воспользуемся функцией uniform, которая является стандартной функцией генератора случайных чисел AnyLogic.

Имя:	<input type="text" value="delay"/>	<input checked="" type="checkbox"/> Отображать имя
<input type="checkbox"/> Исключить		
Тип задержки:	<input checked="" type="radio"/> Определенное время <input type="radio"/> До вызова функции stopDelay()	
Время задержки:	<input type="text" value="uniform(1, 1.8)"/>	минуты
Вместимость:	<input type="text" value="1"/>	
Максимальная вместимость:	<input type="text"/>	
Место агентов:	<input type="text"/>	
▼ Специфические		
Выталкивать агентов:	<input type="checkbox"/>	
Вернуть агента в исходную точку:	<input checked="" type="checkbox"/>	
Включить сбор статистики:	<input checked="" type="checkbox"/>	

После настройки элементов диаграммы нужно выполнить настройку модельного времени.

▼ Модельное время

Режим выполнения: Виртуальное время (максимальная скорость)

Реальное время со скоростью

Остановить:

Начальное время:

Конечное время:

Начальная дата:

Конечная дата:

▼ Специфические

Расширяет тип агента: Нет

Вести журнал в базе данных

[Вести журнал выполнения модели](#)

Создавать наборы данных для динамических переменных

Использовать модельное время Использовать календарные даты

Время первого обновления: минуты

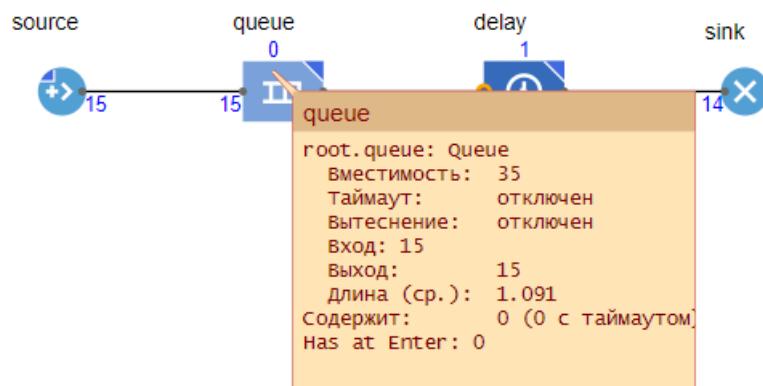
Дата обновления: 8:00:00

Период: минуты

Ограничить количество хранимых значений

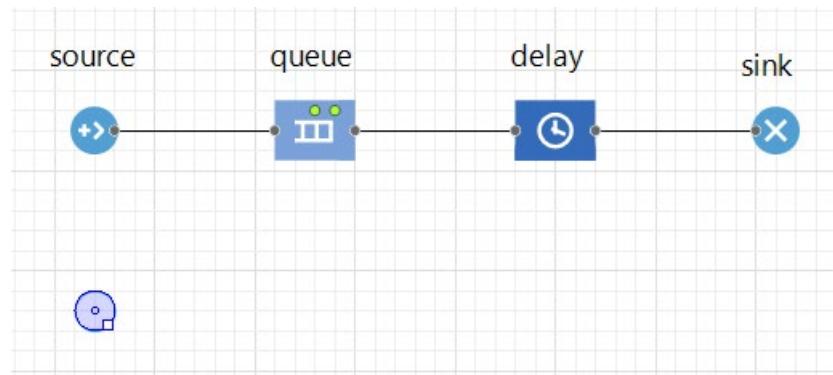
[Создать функцию `toString\(\)` с параметрами](#)

Ниже представлен вариант рабочей модели.



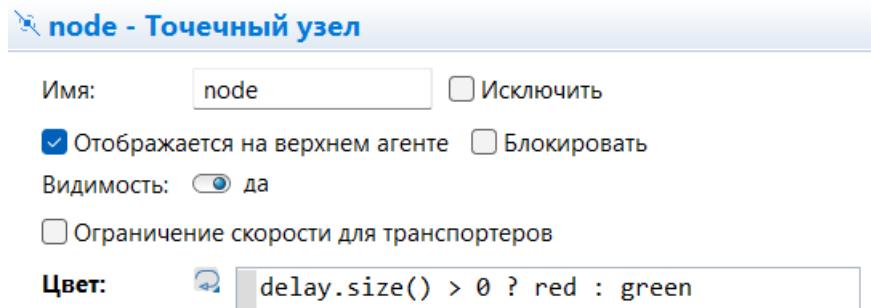
Создание анимации модели

Для визуализации банкомата выберем на панели «Палитра» точечный узел и переместим элемент в графический редактор под блок - схемой процесса.

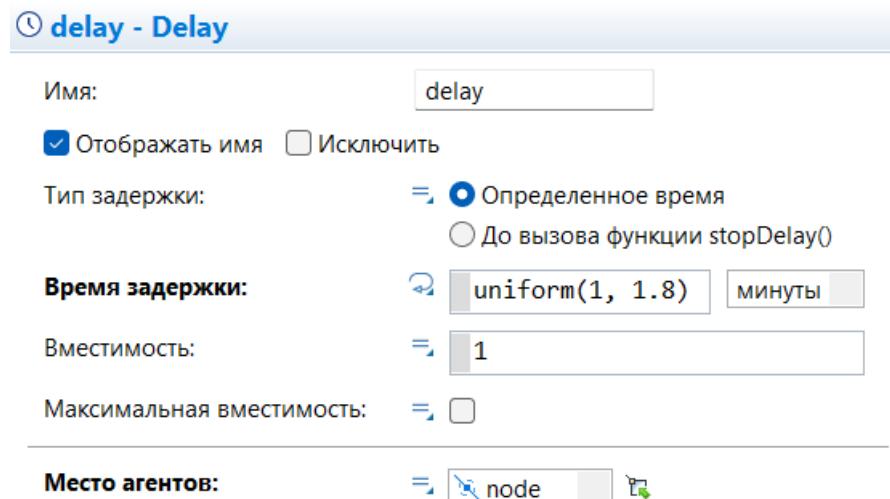


В свойствах точечного узла установим красный цвет для случая, если банкомат занят, и зеленый цвет, если банкомат свободен. Для изменения цвета фигуры необходимо в соответствующей строке ввести следующую команду:

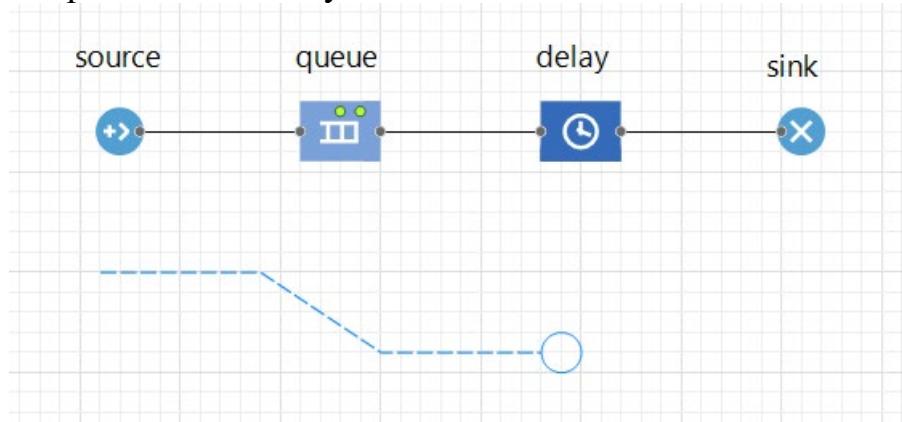
```
delay.size() > 0 ? red : green
```



Вернемся к свойствам объекта Delay и определим в качестве места размещения агентов точечный узел.



Далее с помощью инструментов панели «Палитра» создадим маршрут, обозначающий очередь к банкомату.



Скорректируем свойства объекта Queue, установив путь.

III queue - Queue

Имя:

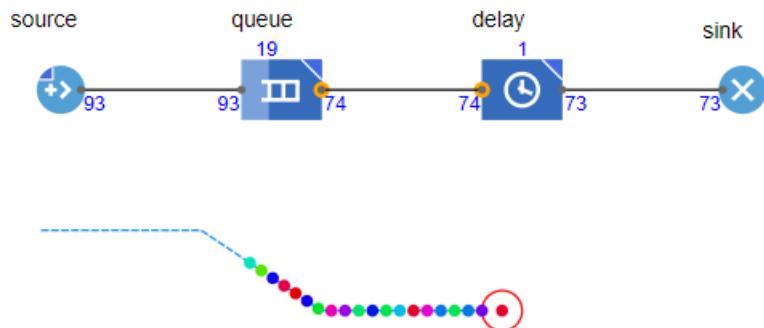
Отображать имя Исключить

Вместимость:

Максимальная вместимость:

Место агентов:

При запуске модели будет отображаться простейшая анимация:



Для настройки 3D-анимации установим 3D-окно из палитры «Презентация». Добавим 3D-объекты клиентов и банкомата.

Шаг 2. Создание нового типа агента

Имя нового типа: Клиент

- Создать новый тип агента "с нуля"
 Использовать таблицу базы данных
Я хочу создать тип агента на базе данных из таблицы БД
 Агент будет использоваться в диаграммах процессов

< Назад

Далее >

Готово

Отмена

Шаг 3. Анимация агента

Выберите анимацию агента: 3D 2D Нет

▼ Люди

Человек

Служащий

Рабочий

Врач

Медсестра

Лаборант

Полицейский

Солдат

Офицер

Женщина 1

Женщина 2



< Назад

Далее >

Готово

Отмена

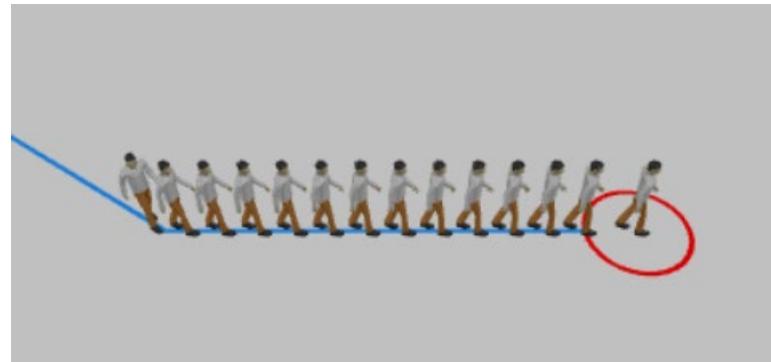
В свойствах объекта Source укажем созданного агента.

▼ Агент

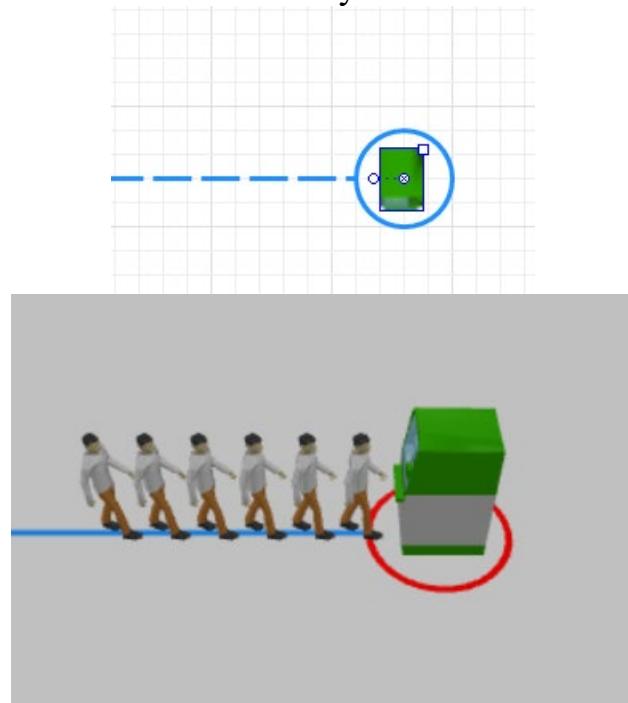
Новый агент: = Клиент

Изменить размеры: =

Запустив модель, видим, что клиенты банка стали 3D-объектами.



Для иллюстрации банкомата выберем из секции палитры «Супер - маркет» 3D-банкомат и переместим его на точечный узел.



Для визуализации результатов изобразим на графиках среднее значение клиентов в очереди и среднее значение числа обслуженных клиентов в приборе обслуживания. Для размещения графиков используем палитру «Статистика» и элемент «Столбиковая диаграмма».

▼ Данные

Заголовок:	Длина очереди
Цвет:	deepPink
Значение:	queue.statsSize.mean()

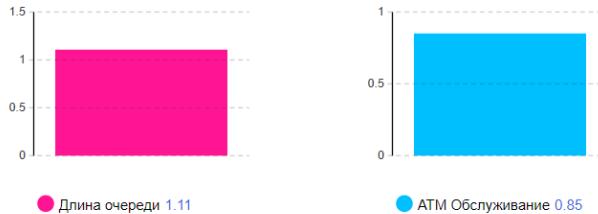
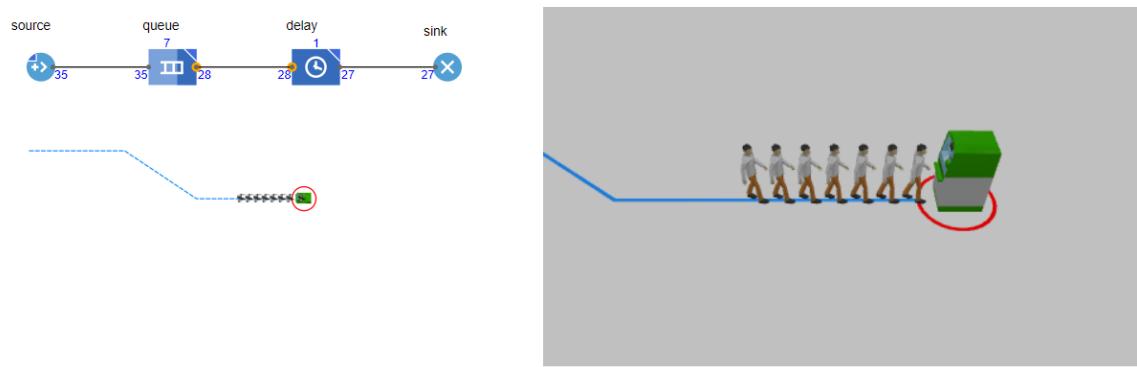
+ × ↑ ↓

▼ Данные

Заголовок:	ATM Обслуживание
Цвет:	deepSkyBlue
Значение:	delay.statsUtilization.mean()

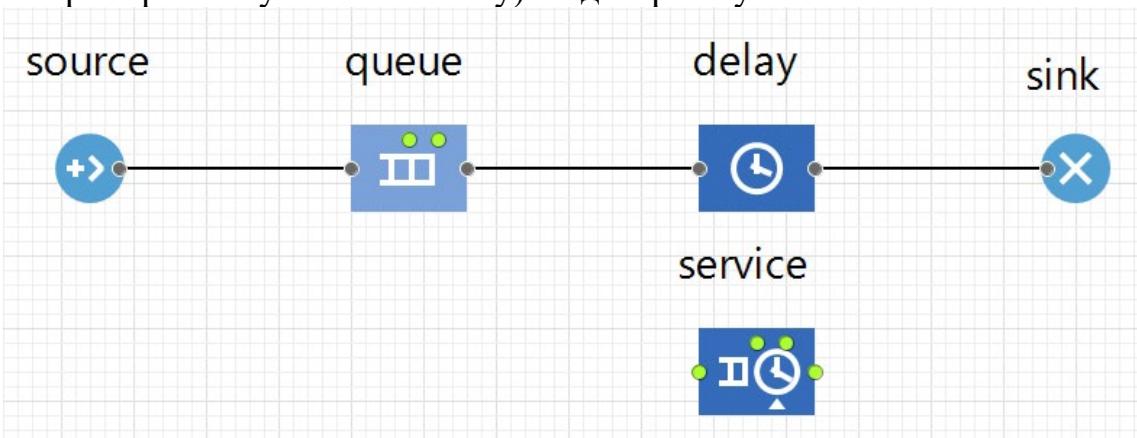
+ × ↑ ↓

Запустим окончательный вариант модели:



Моделирование двухканальной системы массового обслуживания Создание модели

Приходя в банковский офис, клиент может снять деньги в банкомате либо получить консультацию у работников банка. Рассмотрим в качестве первого канала очередь клиентов к банкомату, а в качестве второго канала – очередь к трем консультантам. Из библиотеки моделирования процессов в панели «Палитра» переместим блок Service (узел обслуживания клиентов, состоящий из очереди и прибора обслуживания Delay) на диаграмму Main.

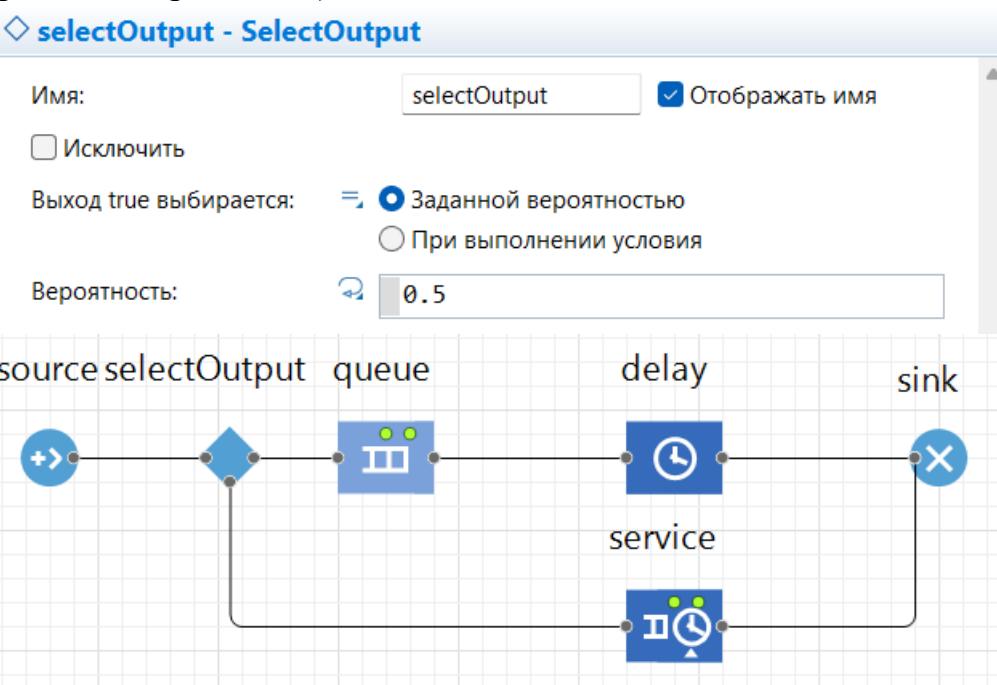


Установим в свойствах блока Service время задержки, соответствующее равномерному закону распределения с параметрами $x_{min} = 1$, $x_{max} = 1,8$. Определим вместимость очереди из 20 клиентов и включим сбор статистики.

service - Service

Имя:	service	<input checked="" type="checkbox"/> Отображать имя
<input type="checkbox"/> Исключить		
Захватить:	<input type="radio"/> (альтернативный) набор ресурсов <input checked="" type="radio"/> ресурсы одного типа	
Тип ресурсов:	<input type="radio"/> Работники	
Количество ресурсов:	<input type="radio"/> 1	
Вместимость очереди:	<input type="radio"/> 20	
Максимальная вместимость:	<input type="radio"/>	
Время задержки:	<input type="radio"/> uniform(1, 1.8) минуты	

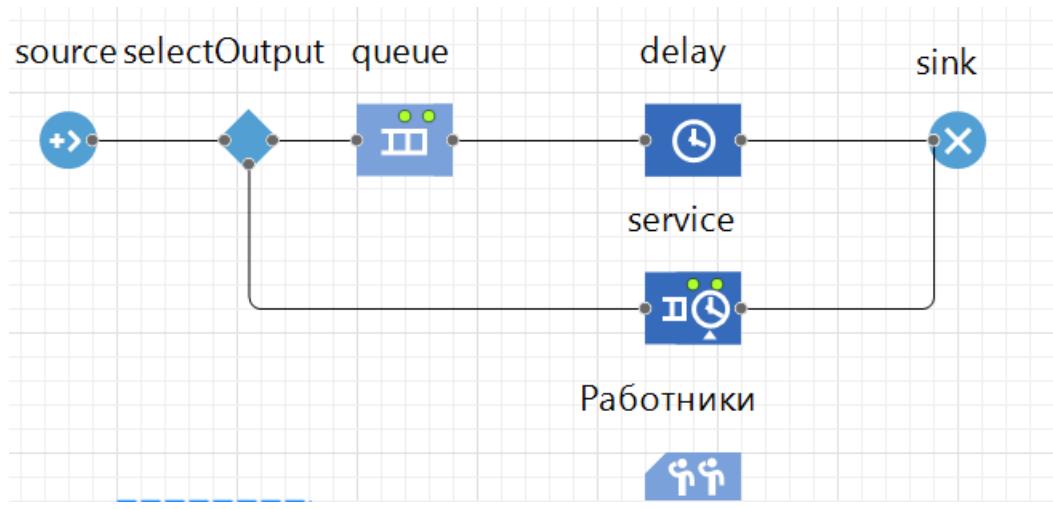
Смоделируем равновероятный выбор клиентом варианта обслуживания (значение вероятности равно 0,5).



Для отображения банковских работников в системе воспользуемся ResourcePool. Выберем количество ресурсов, равное трем, и присвоим элементу имя “Работники”.

Работники - ResourcePool

Имя:	Работники	<input checked="" type="checkbox"/> Отображать имя
<input type="checkbox"/> Исключить		
Тип:	<input type="radio"/> Движущийся	
Количество задано:	<input type="radio"/> Напрямую	
Количество ресурсов:	<input type="radio"/> 3	
При уменьшении кол-ва:	<input type="radio"/> ресурсы сохраняются (конец смены)	



Далее выделим блок Service и отредактируем его свойства.

service - Service

Имя:	service	<input checked="" type="checkbox"/> Отображать имя
<input type="checkbox"/> Исключить		
Захватить:	<input type="radio"/> (альтернативный) набор ресурсов <input checked="" type="radio"/> ресурсы одного типа	
Тип ресурсов:	<input type="radio"/> Работники	

Разместим элемент-ползунок для текущего выбора количества консультантов, считая минимальное количество консультантов равным нулю, а максимальное количество равным трем. Необходимо также установить связь со свойством capacity (число ресурсов) элемента “Работники”.

slider - Бегунок

Имя:	slider	<input type="checkbox"/> Исключить
<input checked="" type="checkbox"/> Отображается на верхнем агенте	<input type="checkbox"/> Блокировать	
Ориентация:	<input type="radio"/> Вертикальная <input checked="" type="radio"/> Горизонтальная	
<input checked="" type="checkbox"/> Связать с:	<input type="radio"/> Работники	
Параметр:		
<input checked="" type="radio"/> capacity		
Минимальное значение:	0	
Максимальное значение:	3	
Шаг:		
Доступность:	<input checked="" type="radio"/> да	

Работники



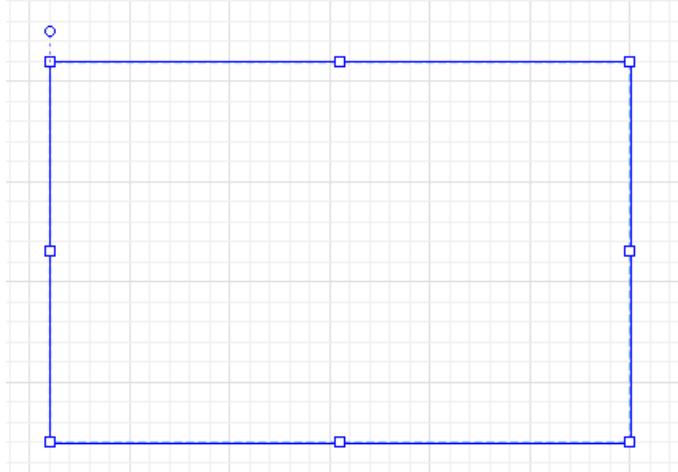
Создание анимации модели

С помощью прямоугольного узла с именем waitingArea отобразим место ожидания клиентов банковскими работниками.

waitingArea - Прямоугольный узел

Имя: waitingArea Исключить
 Отображается на верхнем агенте Блокировать
Видимость: нет
Расположение внутри: Случайное

АтTRACTоры...



Отредактируем свойства блока Service.

Место агентов (queue):

=  waitingArea 

Место агентов (delay):

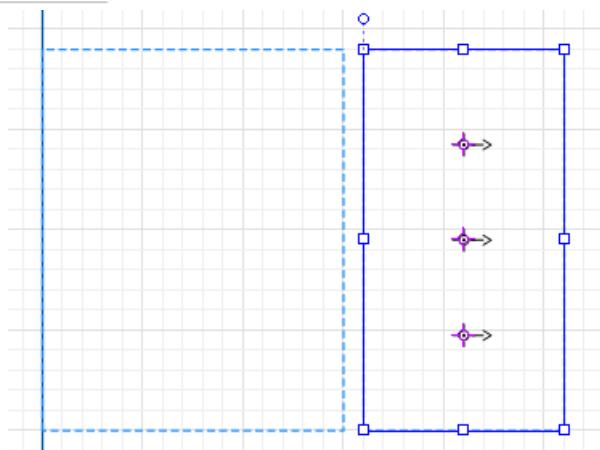
=  

Аналогичным образом отобразим место обслуживания клиентов с помощью прямоугольного узла. Чтобы задать местоположение клиентов, которые будут обслуживаться, используем три атTRACTора.

customerPlaces - Прямоугольный узел

Имя: customerPlaces Исключить
 Отображается на верхнем агенте Блокировать
Видимость: нет

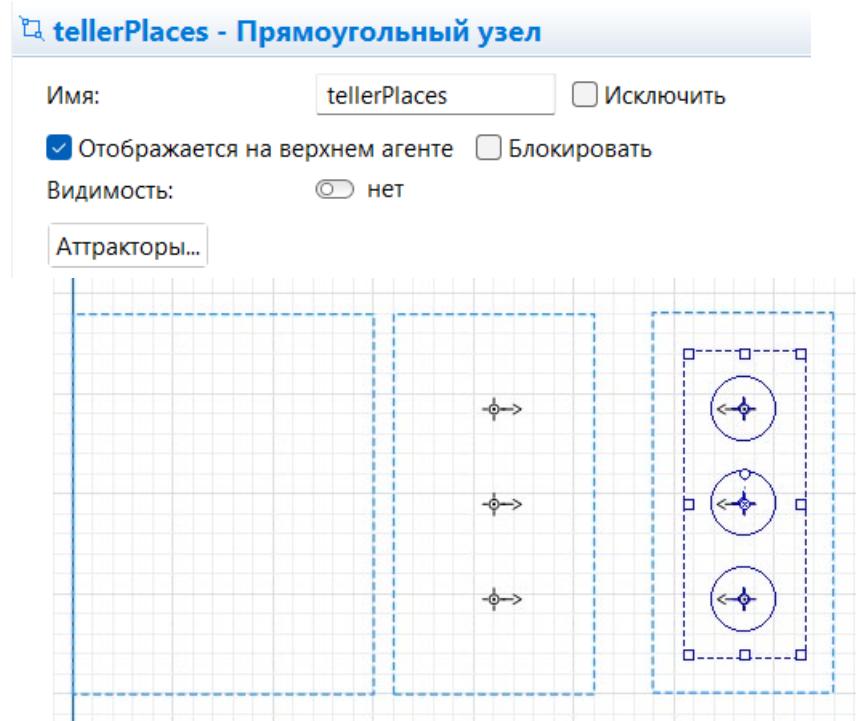
АтTRACTоры...



Отредактируем блок Service.

Место агентов (queue):	<input type="button" value="waitingArea"/>  
Место агентов (delay):	<input type="button" value="customerPlaces"/> 

Создадим место для работников банка аналогичным образом, используя атTRACTоры, и разместим их лицом к клиентам.



Свойства элемента tellers:

Базовое местоположение (узлы):	<input type="button" value="tellerPlaces"/>     
---------------------------------------	---

Создадим новый тип ресурсов для анимации второго потока клиентов.

Шаг 1. Создание нового типа агента

Имя нового типа: Работник

Создать новый тип агента "с нуля"

Использовать таблицу базы данных

Я хочу создать тип агента на базе данных из таблицы БД

< Назад

Далее >

Готово

Отмена

Шаг 2. Анимация агента

Выберите анимацию агента: 3D 2D Нет

▼ Люди

Человек

Служащий

Рабочий

Врач

Медсестра

Лаборант

Полицейский

Солдат

Офицер

Женщина 1

Женщина 2



< Назад

Далее >

Готово

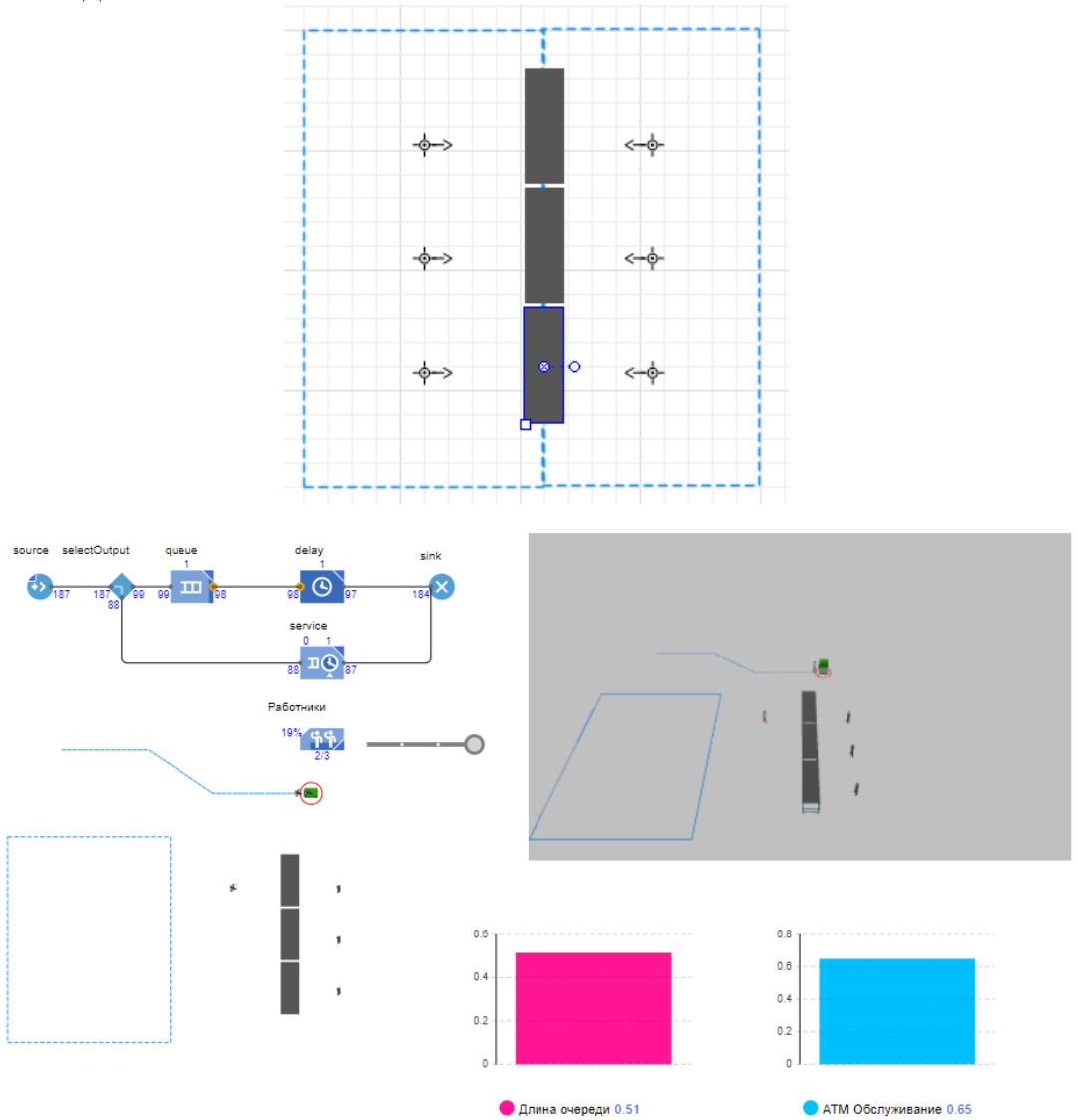
Отмена

Свойства элемента tellers:

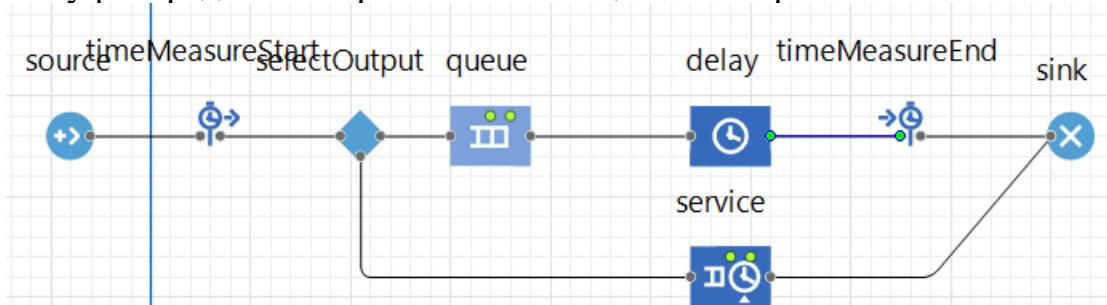
Новый ресурс:

= Работник

Отобразим банковских работников на построенной модели. Расположение элементов в модели:



Для получения информации о времени пребывания клиента в банковском отделении поместим в диаграмму два новых блока из библиотеки моделирования процессов: timeMeasureStart и timeMeasureEnd. Отобразим статистику распределения времени с помощью гистограммы.



Для расчета распределения времени в свойствах блока timeMeasureEnd должен быть указан как минимум один блок timeMeasureStart.

→⌚ timeTotal - TimeMeasureEnd

Имя: Отображать имя

Исключить

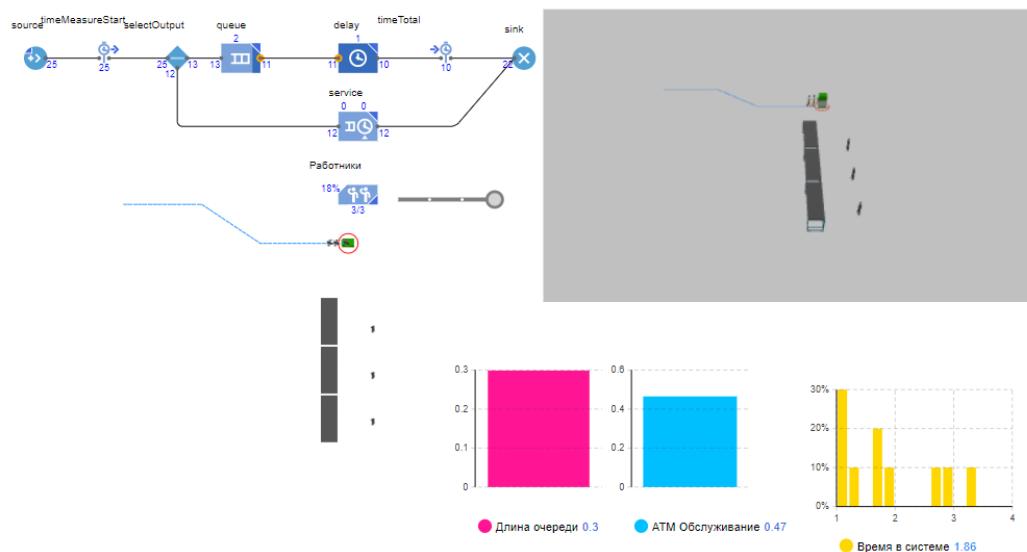
Объекты TimeMeasureStart:

Вместимость набора данных:

Добавим гистограмму на диаграмму агента из палитры «Статистика».



Текущий внешний вид модели банка выглядит так:



Создание Java - класса

Зададим имя Customer_class, указав имя базового класса com.xj.anylogic.libraries.enterprise.Entity. Глобальные атрибуты класса: enteredSystem – время появления клиента в системе, startWaiting – время начала ожидания обслуживания.

Java класс

Создание нового Java класса

Имя:

Базовый класс: ▼

Добавить возможность сохранения состояния

< Назад
Далее >
Готово
Отмена

Поля класса

Добавьте поля Java класса

Имя	Тип	Доступ	Начальное зн...
enteredS...	double	default	
startWait...	double	default	

Создать конструктор
 Создать функцию `toString()`

< Назад
Далее >
Готово
Отмена

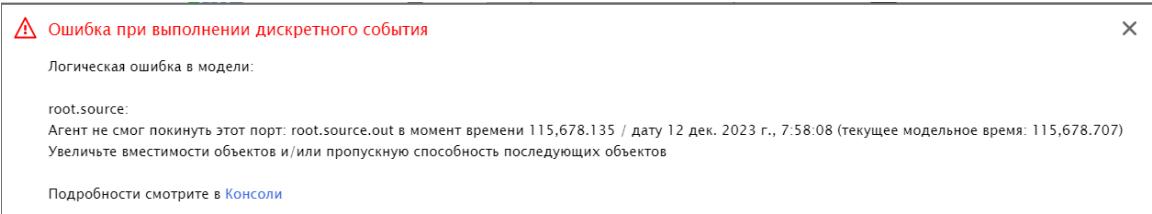
4. Полученные результаты и их анализ

Проведем исследование зависимости времени ожидания от количества консультантов. Получим данные при 2, 3 и 4 работниках, 1 и 2 банкоматах и с разной интенсивностью:

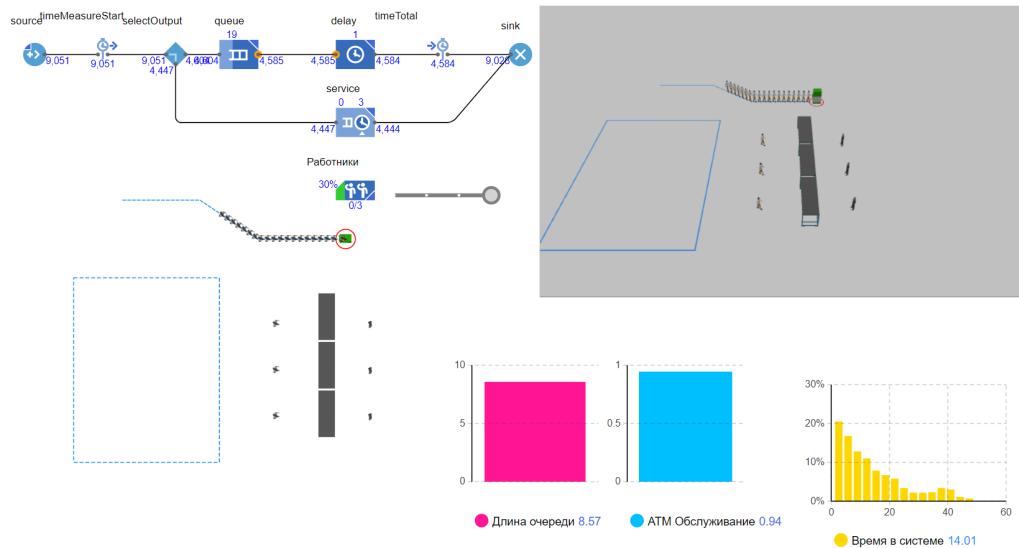
Кол-во банкоматов , шт	Кол-во работников , чел	Интенсивность , чел/мин	Средняя длина очереди , чел	Интенсивность обслуженных клиентов, чел/мин	Время в системе , мин
1	2	0.95	0.69	0.66	2.86
	3	0.95	0.65	0.66	2.83
	4	0.95	0.65	0.65	2.82
	2	0.5	0.1	0.35	1.79
	3	0.5	0.1	0.35	1.78
	4	0.5	0.1	0.35	1.79
	2	1.2	2.32	0.84	5.28
	3	1.2	2.31	0.84	5.26
	4	1.2	2.31	0.84	5.25
2	2	0.95	0.05	0.33	1.5
	3	0.95	0.05	0.33	1.5
	4	0.95	0.04	0.33	1.5
	2	0.5	0.01	0.17	1.43

	3	0.5	0.01	0.17	1.43
	4	0.5	0.01	0.17	1.43
	2	1.2	0.01	0.42	1.56
	3	1.2	0.01	0.42	1.56
	4	1.2	0.01	0.42	1.56

Анализ модели показывает, что оптимальное количество работников в банке – 3 консультанта, дополнительный сотрудник не приведет к существенному изменению ситуации. Увеличивая интенсивность прибытия клиентов, видим, что при $\lambda = 1,25$, трех сотрудниках и одном банкомате возникает ошибка системы.



Однако при значении $\lambda = 1,24$ система сохраняет работоспособность:



5. Вывод

Таким образом, в работе построена одноканальная и двухканальная модель обслуживания клиентов банковского офиса. С использованием методов компьютерного моделирования установлено, что оптимальное количество консультантов для безотказной работы офиса равно 3, а максимальное значение интенсивности входного потока заявок, при котором система сохраняет работоспособность, равно 1,24.

Было выяснено, что увеличение числа банкоматов сильно влияет на очередь, и установка второго банкомата будет лучшим решением для уменьшения времени ожидания. Кол – во работников слабо влияет на результаты моделирования и увеличение числа работников не будет хорошим решением. При разных значениях интенсивности получили закономерные результаты: с увеличением интенсивности растет очередь.