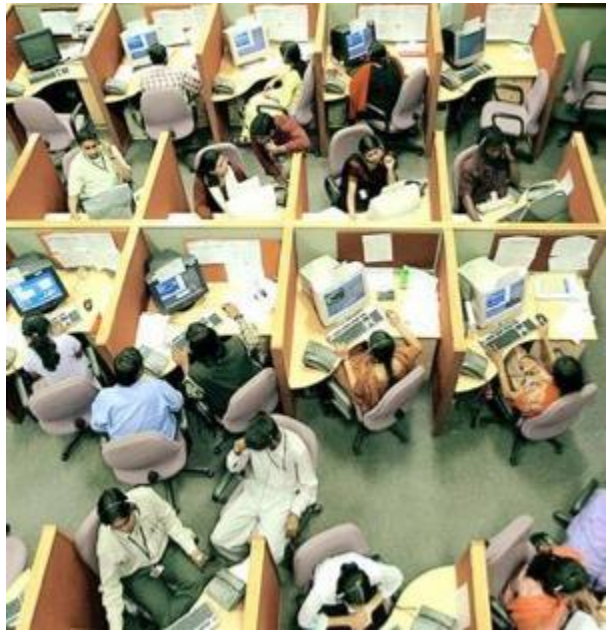
Упражнение. Колл-центр

Дополнительное задание по дискретно-событийному моделированию

anylogic 7



Колл-центр (1/3)

- Поступают два типа звонков:
  - 1 типа с интенсивностью ArrivalRate1 = 1.5 в секунду
  - 2 типа с интенсивностью ArrivalRate2 = 1 в секунду
- Если клиенты ждут слишком долго, они прекращают ожидание
  - Максимальное время ожидания распределено экспоненциально
  - Для 1 типа среднее AbandonmentTimeMean1 = 100 секунд
  - Для 2 типа среднее AbandonmentTimeMean2 = 100 секунд
- На звонки отвечает две группы операторов
  - Их число NOperators1 = 100 и NOperators2 = 100
  - Время обслуживания распределено по треугольному закону triangular(ServiceTimeXX/2, ServiceTimeXX, 2*ServiceTimeXX), где среднее ServiceTimeXX задается так:
  - Для операторов 1 при ответе на звонки 1 типа ServiceTime11 = 100
  - Для операторов 2 при ответе на звонки 1 типа ServiceTime12 = 200
  - Для операторов 2 при ответе на звонки 2 типа ServiceTime22 = 100 секунд

Колл-центр (2/3)

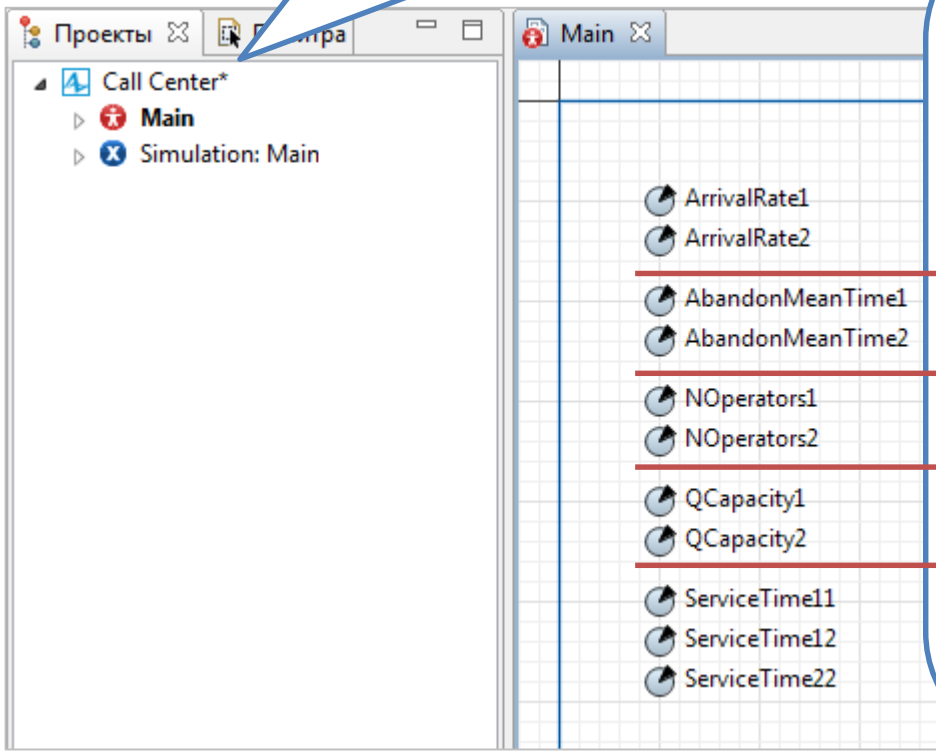
- У каждой группы операторов есть очередь ожидающих звонков
  - Максимальная длина очереди к группе 1 QCapacity1 = 50
  - Максимальная длина очереди к группе 2 QCapacity2 = 50
- При поступлении нового звонка, он обрабатывается следующим образом:
  - Если поступает звонок 1 типа, и очередь к группе 1 не полна, то звонок ставится в эту очередь, иначе, если не полна очередь к группе 2, то он помещается туда; если же обе очереди полны, звонок теряется
  - Если прибывает звонок 2 типа, и очередь к группе 2 не полна, то звонок помещается туда, иначе он теряется

Колл-центр (3/3)

- Создайте имитационную модель центра обработки звонков
- Измерьте процентную долю следующих групп звонков:
  - Упущенные
  - Прекратившие ожидание
  - Обслуженные
- Вычислите распределение времени ожидания обслуженных звонков (отдельно для каждого типа звонка)
- Добавьте возможность изменения интенсивностей поступления звонков и максимальных длин очередей по ходу моделирования и пронаблюдайте, как Ваши изменения будут влиять на качество обслуживания

Колл-центр. Фаза 1. Шаг 1

❶ Создайте новую модель: *Call Center*



❷ Создайте на диаграмме типа агента *Main* следующие параметры с указанными значениями по умолчанию:

ArrivalRate1	*1.5 в секунду*	**Тип**: *Интенсивность*
ArrivalRate2	*1.0 в секунду*	
AbandonMeanTime1	*100 секунд*	**Тип**: *Время*
AbandonMeanTime2	*100 секунд*	
NOperators1	*100*	**Тип**: *int*
NOperators2	*100*	
QCapacity1	*50*	
QCapacity2	*50*	
ServiceTime11	*100 секунд*	**Тип**: *Время*
ServiceTime12	*200 секунд*	
ServiceTime22	*100 секунд*	

❸ По умолчанию, элемент **Параметр** имеет **Тип**: *double*, то есть *вещественное число*.

Нам же необходимо задать определенное количество операторов линии и звонков, то есть, *целое число*, которые задается параметрами типа *int*.

Чтобы задать количество поступающих звонков в единицу времени, мы выбираем **Тип** параметра: *Интенсивность*. Затем нам будут предложены на выбор единицы, в которых можно задать интенсивность. В нашей модели мы везде используем секунды.

Так же мы задаем параметры, задающие промежутки времени: выбираем *Время* как **Тип** параметра, чтобы затем выбрать нужные единицы времени.

Колл-центр. Фаза 1. Шаг 2

❸ *Service*

Захватить: *ресурсы одного типа*

Тип ресурсов: *operators1*

Вместимость очереди: *QCapacity1*

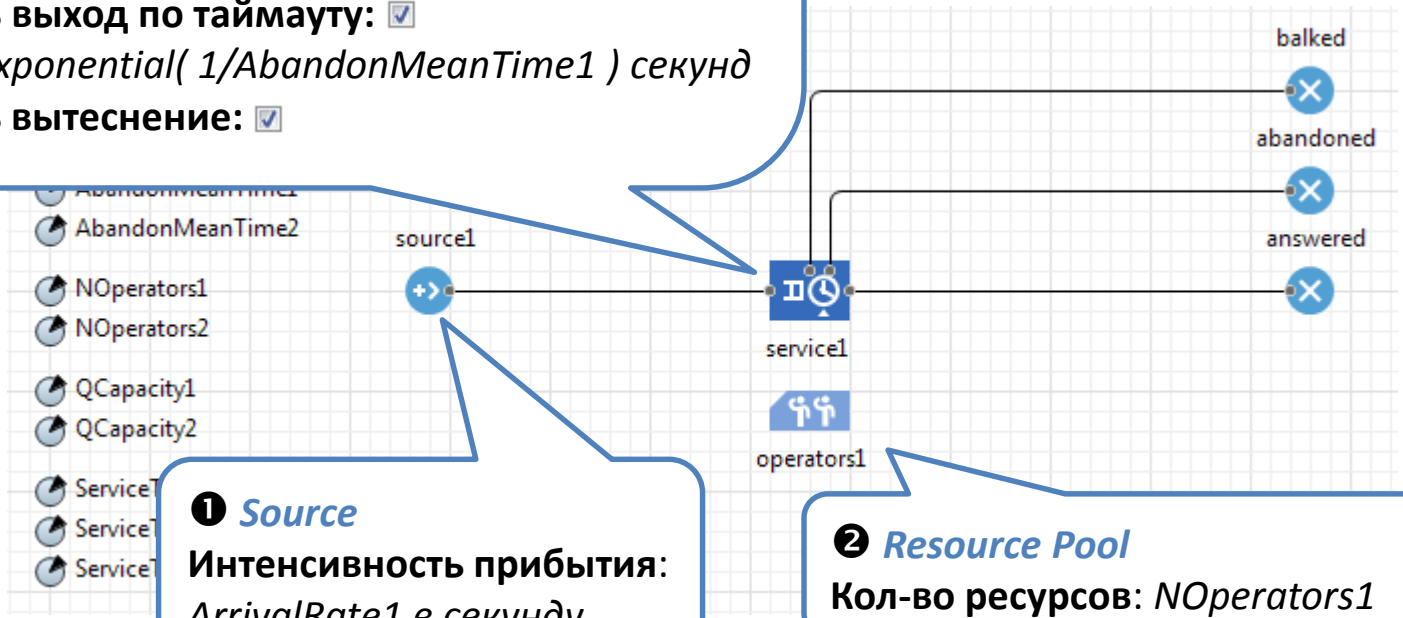
Время задержки: *triangular(ServiceTime11/2, ServiceTime11, 2*ServiceTime11) секунд*

Специфические

Разрешить выход по таймауту: ☑

Таймаут: *exponential(1/AbandonMeanTime1) секунд*

Разрешить вытеснение: ☑



❶ *Source*

Интенсивность прибытия: *ArrivalRate1 в секунду*

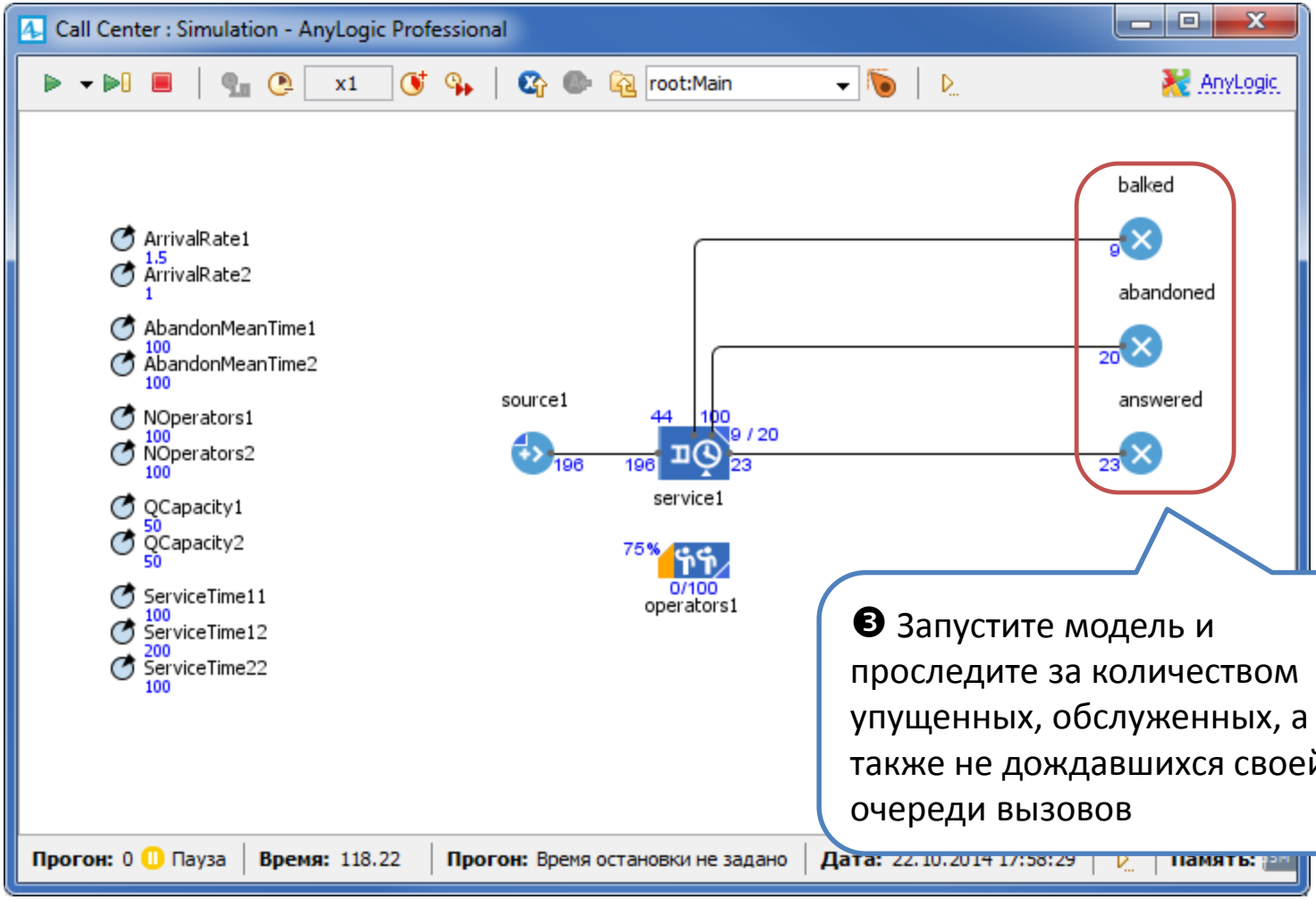
❷ *Resource Pool*

Кол-во ресурсов: *NOperators1*

Таймаут и вытеснение в объекте Queue

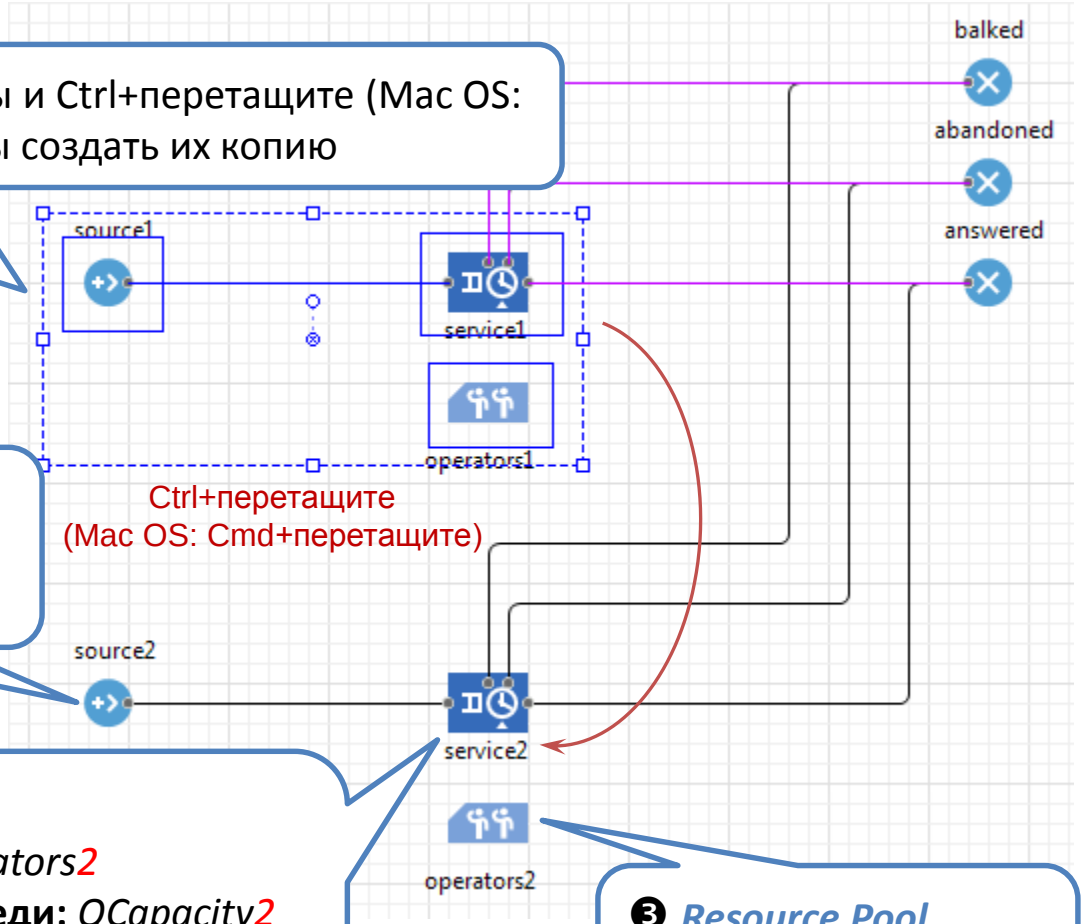
- Вы можете настроить объект ***Queue*** так, чтобы он позволял агентам покидать объект в том случае, если они ждали в очереди в течение заданного таймаута. Для этого нужно выбрать опцию **Разрешить уход по таймауту** и задать таймаут, который динамически вычисляется для каждого агента, так что он может зависеть от самого агента и может быть стохастическим. Агенты, покидающие объект по таймауту, направляются в верхний правый порт объекта.
- Если Вы выберете опцию **Разрешить вытеснение**, то любые поступающие агенты смогут быть помещены в очередь, но если очередь будет полна, то агенты с более низкими приоритетами будут вытесняться из очереди, покидая объект через верхний левый порт. Приоритет вычисляется отдельно для каждого агента. Если Вы оставите поле **Приоритет агента** пустым, то все агенты будут рассматриваться как имеющие одинаковые приоритеты, и поэтому если очередь будет переполнена, то новые агенты будут отвергаться и покидать объект через порт вытеснения.

Колл-центр. Фаза 1. Шаг 3



Колл-центр. Фаза 2. Шаг 1

❶ Выделите эти объекты и Ctrl+перетащите (Mac OS: Cmd+перетащите), чтобы создать их копию



❷ ***Source***

Интенсивность прибытия: *ArrivalRate2*

❸ ***Resource Pool***

Количество ресурсов: *NOperators2*

❹ ***Service***

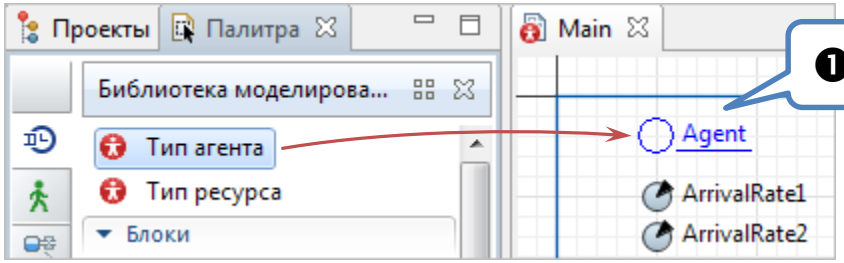
Тип ресурсов: *operators2*

Вместимость очереди: *QCapacity2*

Пока не меняйте другие параметры

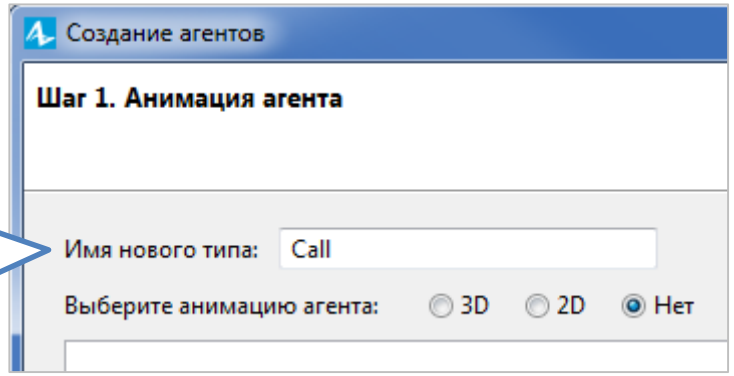
❶ Здесь мы создаем новый процесс, пока что абсолютно независимый от заданного ранее. Два источника агентов, два типа ресурсов и т.д. будут работать параллельно.

Колл-центр. Фаза 2. Шаг 2

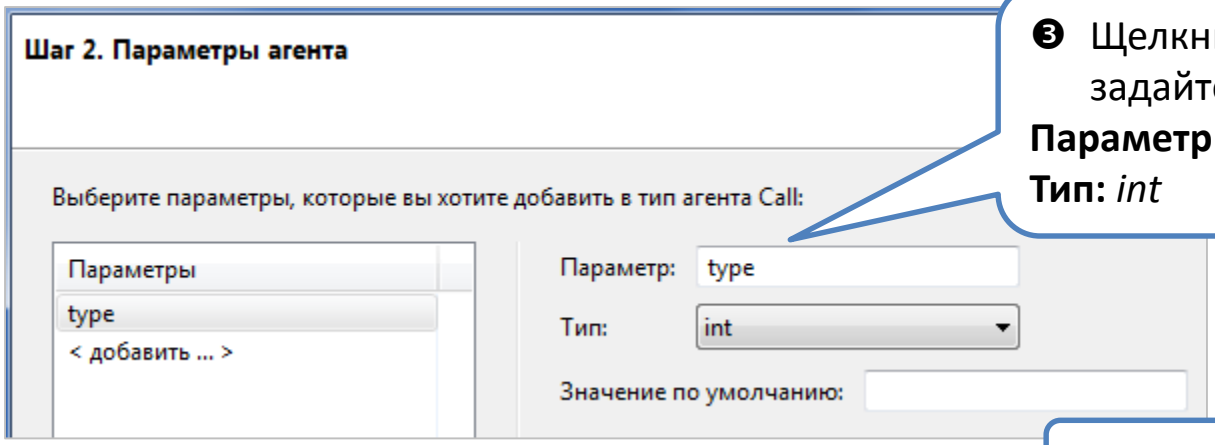


Перетащите в графический редактор

❶ Создайте новый **Тип агента**



❷ Задайте **Имя нового типа** *Call*, выберите *Нет* как тип анимации и щелкните по кнопке **Далее**



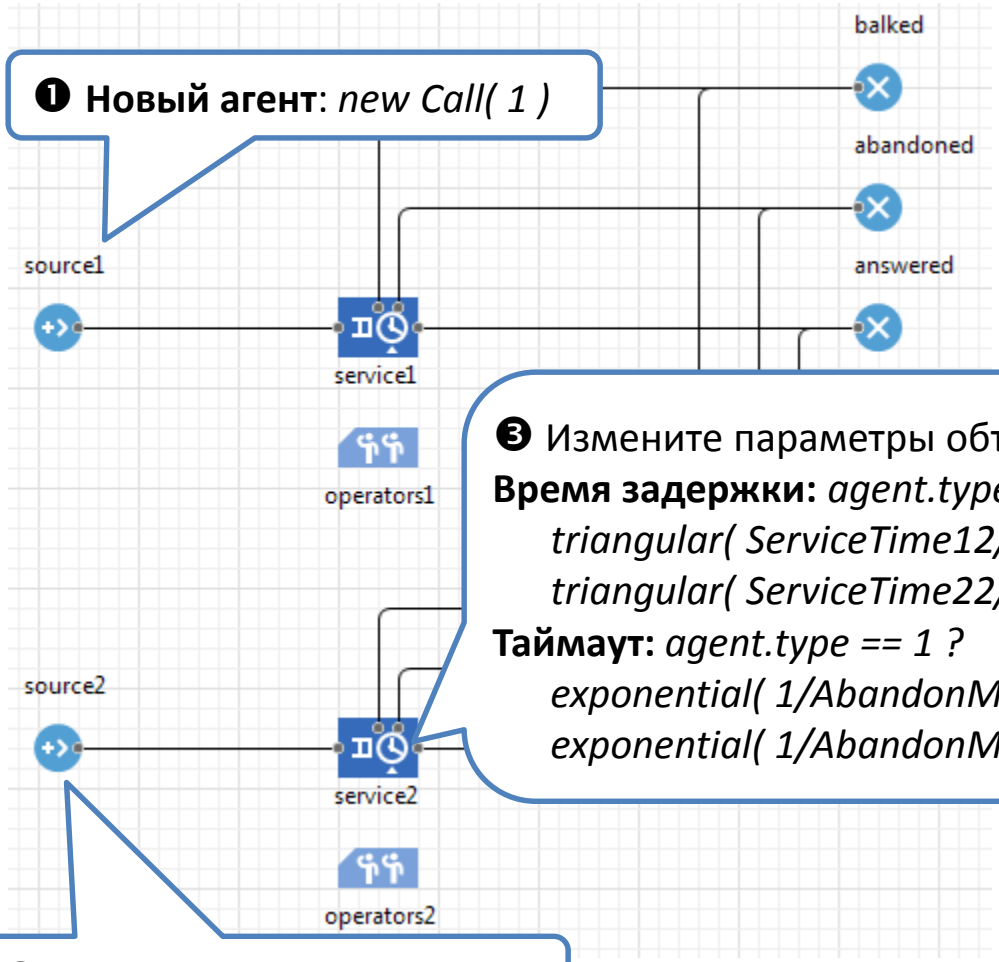
❸ Щелкните **< добавить ... >** и задайте:
Параметр: *type*
Тип: *int*

❹ Щелкните по кнопке **Готово**

Агенты с расширенной функциональностью

- По умолчанию в процессных моделях передаются агенты типа *Agent*. Такие агенты дают не так уж и много возможностей для настройки (например, Вы можете изменить цвет заданной фигуры анимации агента или вообще поменять эту фигуру). Если же Вам нужно добавить для агента свои собственные свойства, функции, собрать статистику и т.д., то Вам будет нужно создать свой собственный тип агента.
- Мастер создания нового типа агента позволяет Вам добавлять поля различных типов и задавать для них значения по умолчанию. Также создаются два конструктора типа – один создает агента с заданными по умолчанию значениями параметров, а другой позволяет задать их пользователю с помощью аргументов.

Колл-центр. Фаза 2. Шаг 3



❶ **Новый агент**: *new Call(1)*

❷ **Новый агент**: *new Call(2)*

❸ Измените параметры объекта *service2*:

Время задержки: *agent.type == 1 ?*
*triangular(ServiceTime12/2, ServiceTime12, 2*ServiceTime12) :*
*triangular(ServiceTime22/2, ServiceTime22, 2*ServiceTime22)*

Таймаут: *agent.type == 1 ?*
exponential(1/AbandonMeanTime1) :
exponential(1/AbandonMeanTime2)

❷ *new Call(1)* является вызовом конструктора созданного нами ранее типа **Call**. Он используется объектом *Source* для создания новых агентов именно этого типа.
Чтобы ввести *new Call(1)* в параметре **Новый агент**, щелкните значок параметра рядом с выпадающим списком:

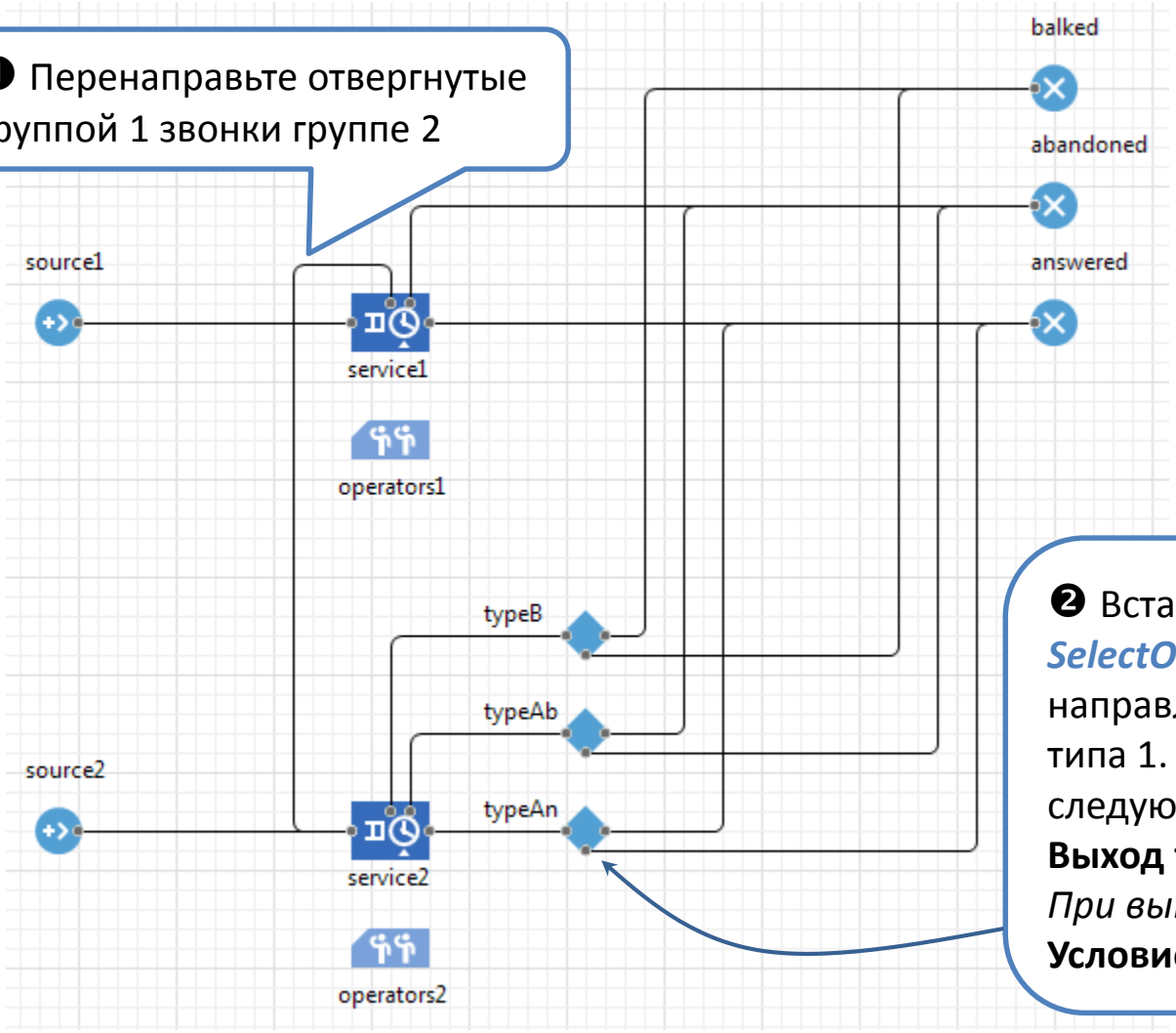
❸ Время задержки и таймаут объекта *service2* теперь вычисляются по-разному, в зависимости от типа поступившего звонка (1 или 2).
Единицами времени здесь все так же являются *секунды*.

Тип агента и доступ к параметрам агента

- Вы создали свой собственный тип агента и указали объектам *Source*, что они должны создавать агентов именно этого типа. В объектах потоковой диаграммы нам нужно работать с созданными нами параметрами агентов, например, в этой модели нам нужно узнать тип агента (параметр *type*, значение 1 или 2) в параметрах **Время задержки** и **Таймаут** объекта *Service*.
- Как только Вы выберите тип агента *Call* в параметре **Новый агент** блока *Source*, переменная *agent*, доступная в параметрах объекта *Service* **Таймаут** и т.д., будет экземпляром типа агента *Call*, и Вы сможете просто писать *agent.type*.

Колл-центр. Фаза 3. Шаг 1

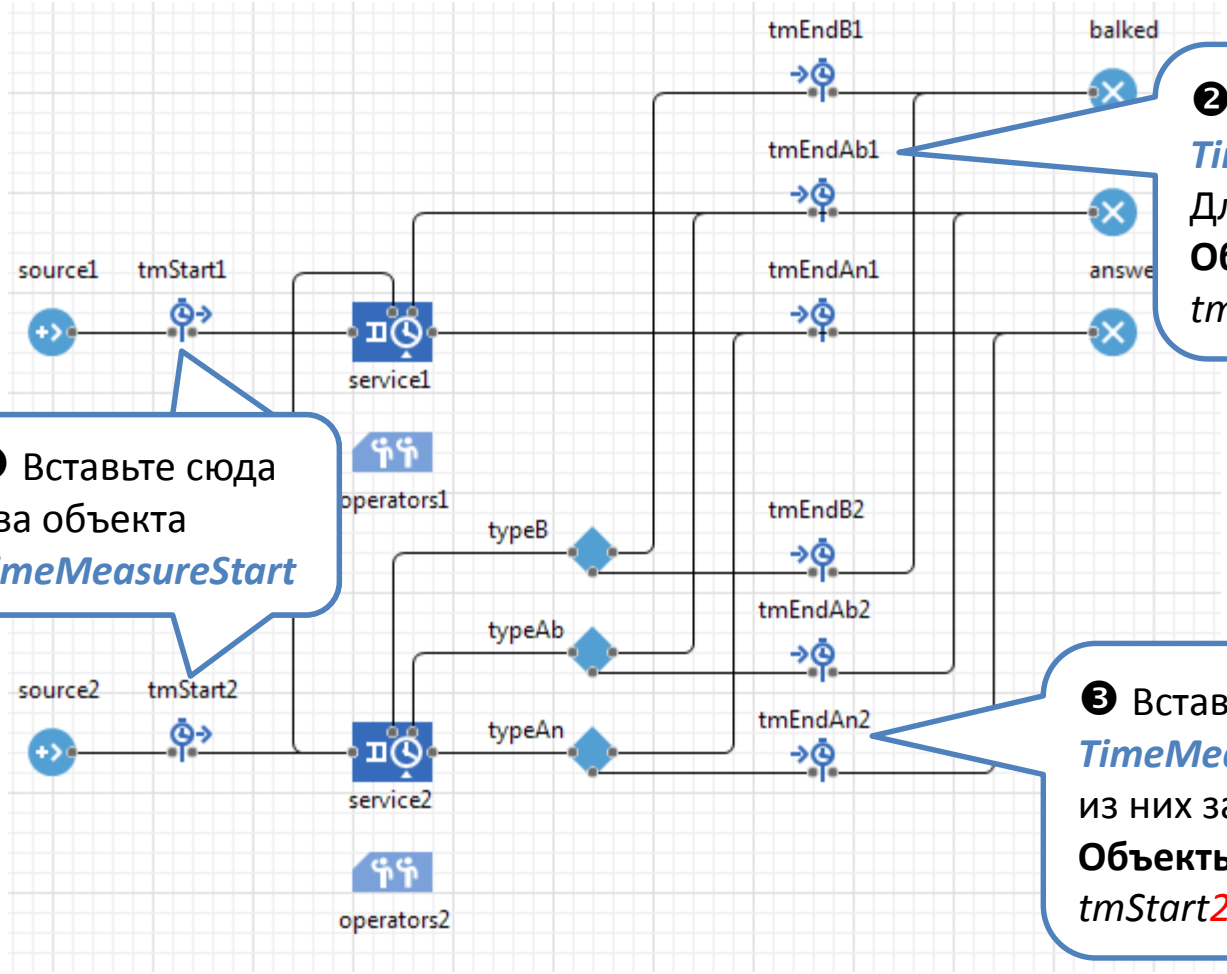
❶ Перенаправьте отвергнутые группой 1 звонки группе 2



❷ Вставьте три объекта ***SelectOutput*** для того, чтобы направлять обратно звонки типа 1. Задайте у всех объектов следующие параметры:
Выход true выбирается:
При выполнении условия
Условие: *agent.type == 1*

❷ Поскольку мы направили часть звонков типа 1 второй группе операторов, то во втором процессе у нас теперь будут звонки обоих типов. И так как мы будем собирать статистику отдельно по каждому типу звонка, то нам нужно перенаправить звонки типа 1 обратно в первый процесс, что мы и сделаем с помощью трех объектов *SelectOutput*. Через объекты *SelectOutput* проходят агенты типа *Call,* и мы имеем простой доступ к параметру агента *type*.

Колл-центр. Фаза 3. Шаг 2



❶ Вставьте сюда два объекта *TimeMeasureStart*

❷ Вставьте три объекта *TimeMeasureEnd*.
Для всех из них задайте:
Объекты TimeMeasureStart:
tmStart1

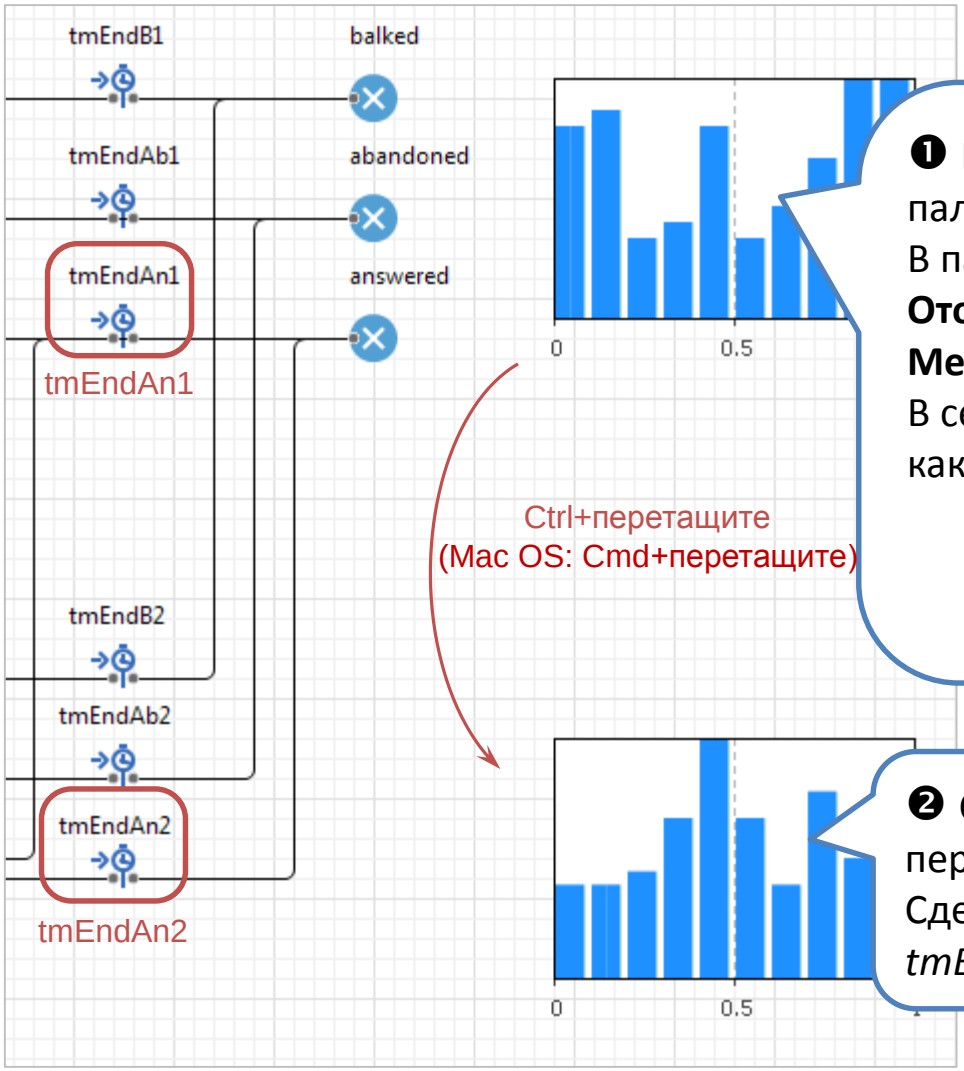
❸ Вставьте еще три объекта *TimeMeasureEnd*. Для всех из них задайте:
Объекты TimeMeasureStart:
tmStart2

❷, ❸ Вы можете указать объект *TimeMeasureStart* в свойстве объекта *TimeMeasureEnd*, щелкнув кнопку «плюс» под параметром и выбрав его из списка доступных объектов или выбрав его в графическом редакторе с помощью кнопки выбора объекта.

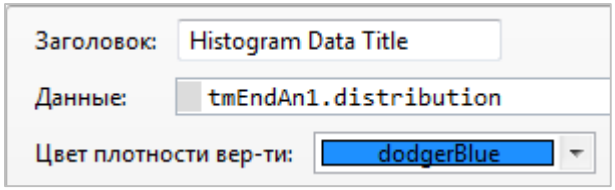
Измерение времени в процессных моделях

- Часто в процессных моделях бывает нужно узнать, какое время агенты провели на определенном участке процесса. Проще всего это сделать с помощью объектов *TimeMeasureStart* и *TimeMeasureEnd*. Объекты *TimeMeasureStart* должны быть помещены перед каждым входом в интересующую нас часть потоковой диаграммы, а *TimeMeasureEnd* – перед каждым выходом из нее. Каждому объекту *TimeMeasureEnd* нужно знать обо всех соответствующих объектах *TimeMeasureStart*.
- Объект *TimeMeasureEnd* собирает статистику и распределение времени, проведенного на заданном участке процесса.
- Важно не выпустить из модели агентов, прошедших через *TimeMeasureStart* без прохождения через *TimeMeasureEnd*: последний объект сохраняет агентов в списке, в то время как первый их удаляет. В нашей модели мы хотим только лишь собрать статистику по обслуженным звонкам, но нам приходится помещать *TimeMeasureEnd* на каждый из трех выходов.

Колл-центр. Фаза 3. Шаг 3



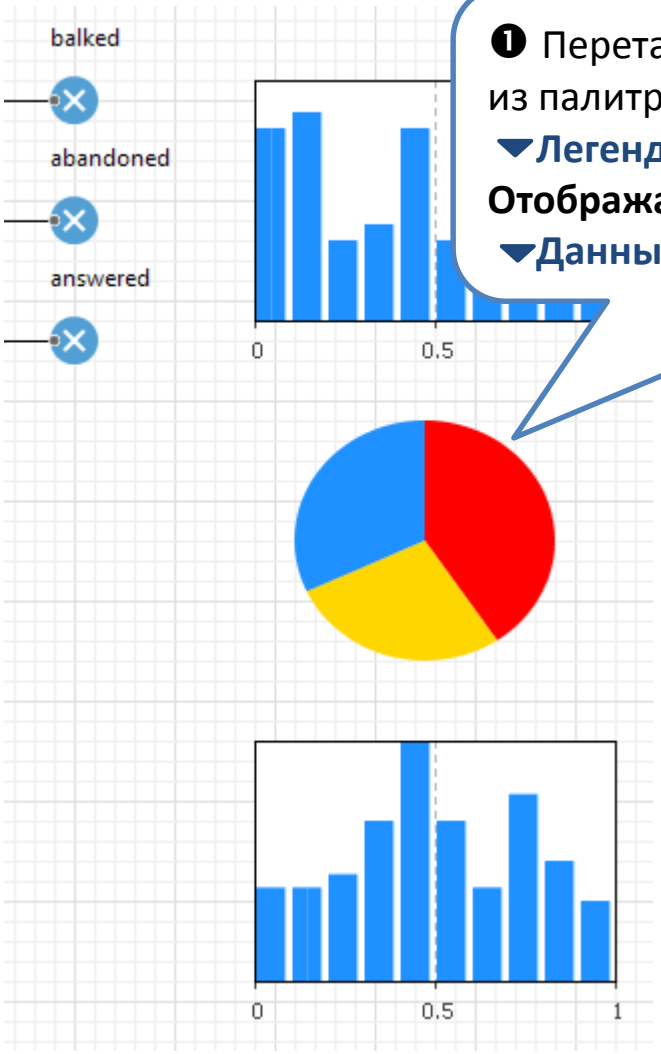
❶ Перетащите сюда элемент ***Гистограмма*** из палитры **Статистика**.
В панели **Свойства** задайте:
Отображать легенду: ☐ (секция ▾ **Легенда**)
Метки по оси Y: *Нет* (секция ▾ **Внешний вид**)
В секции ▾ **Данные** добавьте элемент данных, как показано ниже:



❷ Ctrl+перетащите (Mac OS: Cmd+перетащите) первую диаграмму, чтобы создать ее копию. Сделайте изменение в свойстве **Данные:** *tmEndAn2.distribution*

❶, ❷ Помните, что объект *TimeMeasureEnd* собирает информацию о распределении времени, проведенного агентами на заданном участке процесса. Распределение (объект *Данные гистограммы*) доступно как *имяОбъекта.distribution* и может отображаться на **Гистограмме**.

Колл-центр. Фаза 3. Шаг 4

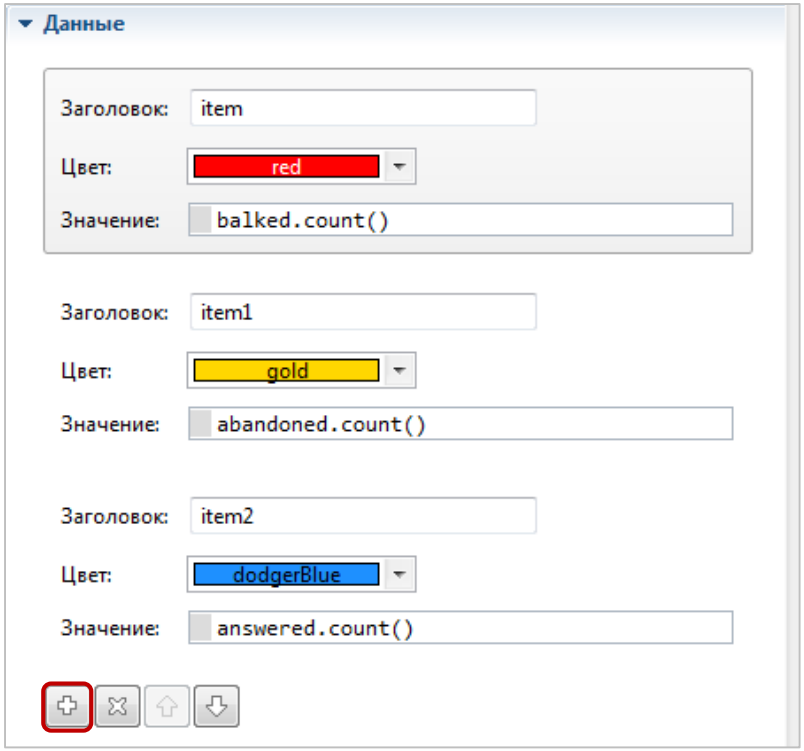


❶ Перетащите элемент ***Круговая диаграмма*** из палитры **Статистика**.

▼**Легенда**:

Отображать легенду: ☐

▼**Данные**: Добавьте три элемента данных

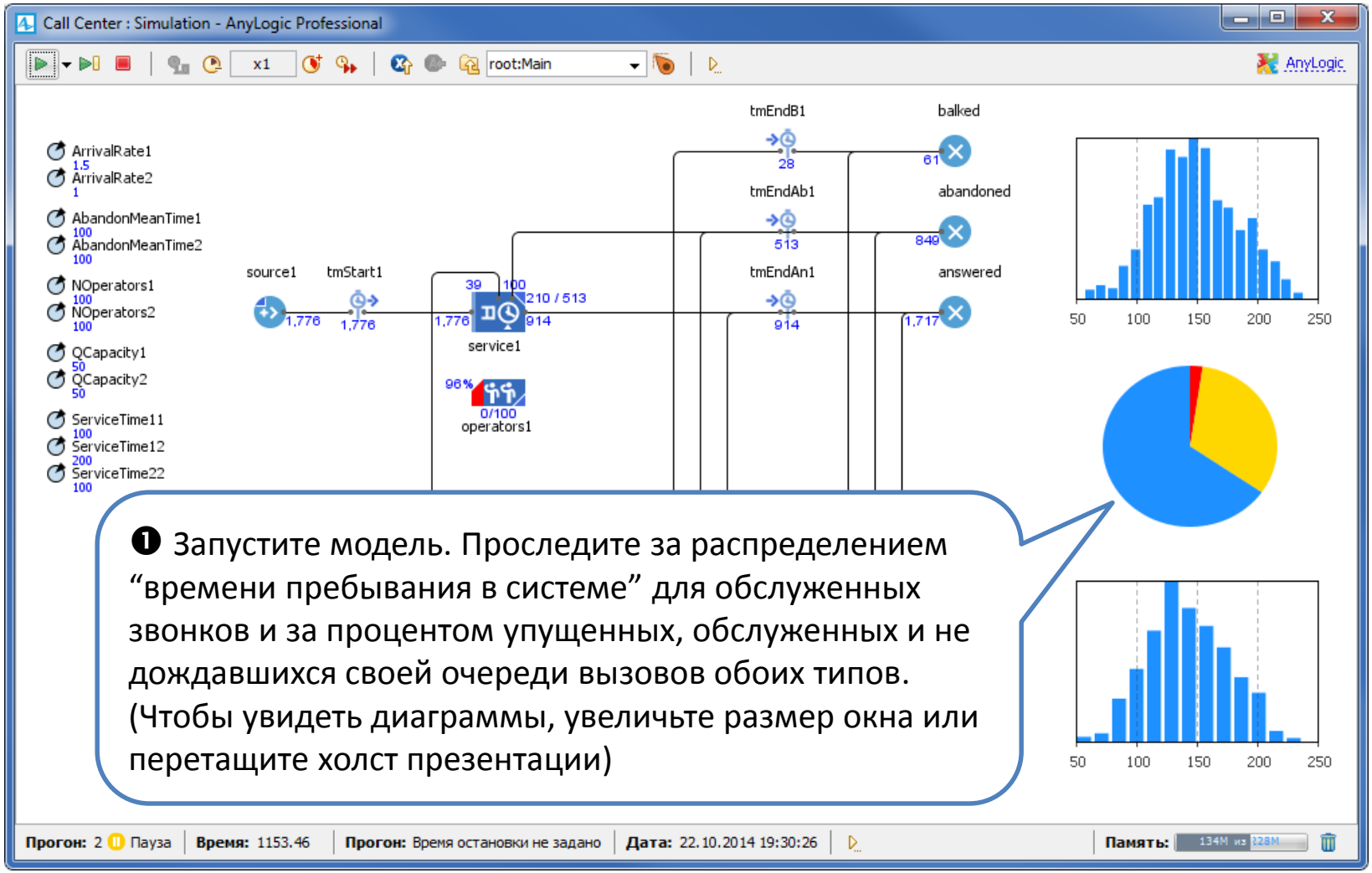


❶ У объекта *Sink* есть функция *count()*, которая возвращает количество агентов, покинувших этот объект за все время работы модели.

Получение количества агентов, прошедших через порт

- Вы можете получить количество агентов, прошедших через любой порт любого объекта Библиотеки моделирования процессов, с помощью метода *имяОбъекта.имяПорта.count().* Например, Вы можете вызвать *typeB.outT.count().*
- Но для объектов, составленных из других объектов (например, **Service**), это может не работать. Например, чтобы получить количество агентов, покинувших объект *service2* через порт *outTimeout*, Вам нужно будет вызвать *service2.seize.queue.outTimeout.count().*

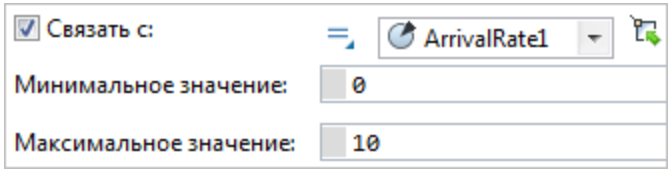
Колл-центр. Фаза 3. Шаг 5

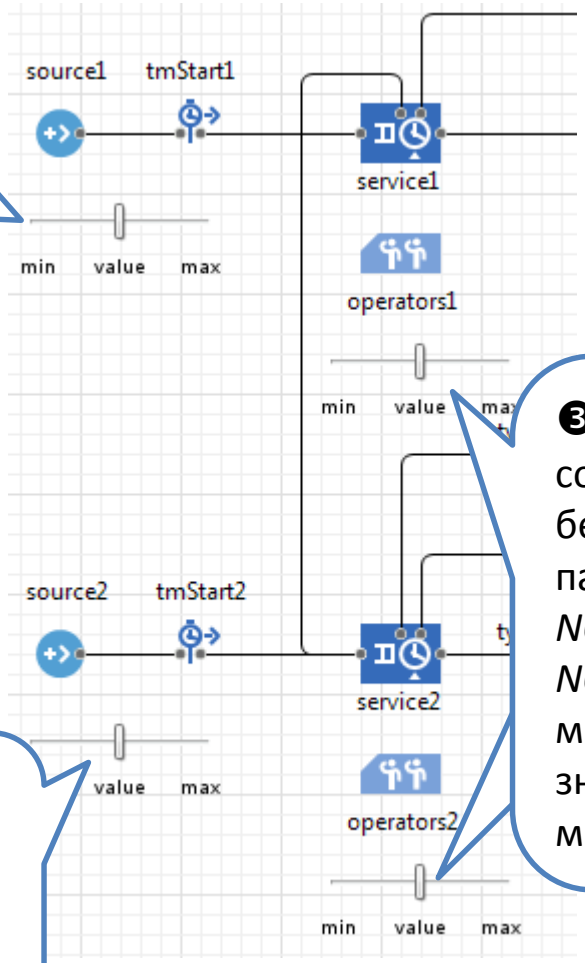


❶ В чем Вы видите причину разницы распределений “времен пребывания в системе”? Почему качество обслуживания для звонков типа 2 хуже?

Колл-центр. Фаза 4. Шаг 1

❶ Перетащите сюда *Бегунок* из палитры **Элементы управления**. В панели **Свойства**, нажмите на кнопку **Добавить метки**. Свяжите бегунок с параметром *ArrivalRate1*





❷ Ctrl+перетащите (Mac OS: Cmd+перетащите) бегунок, чтобы создать его копию. Добавьте метки и свяжите его с параметром *ArrivalRate2*, оставив те же значения минимума и максимума.

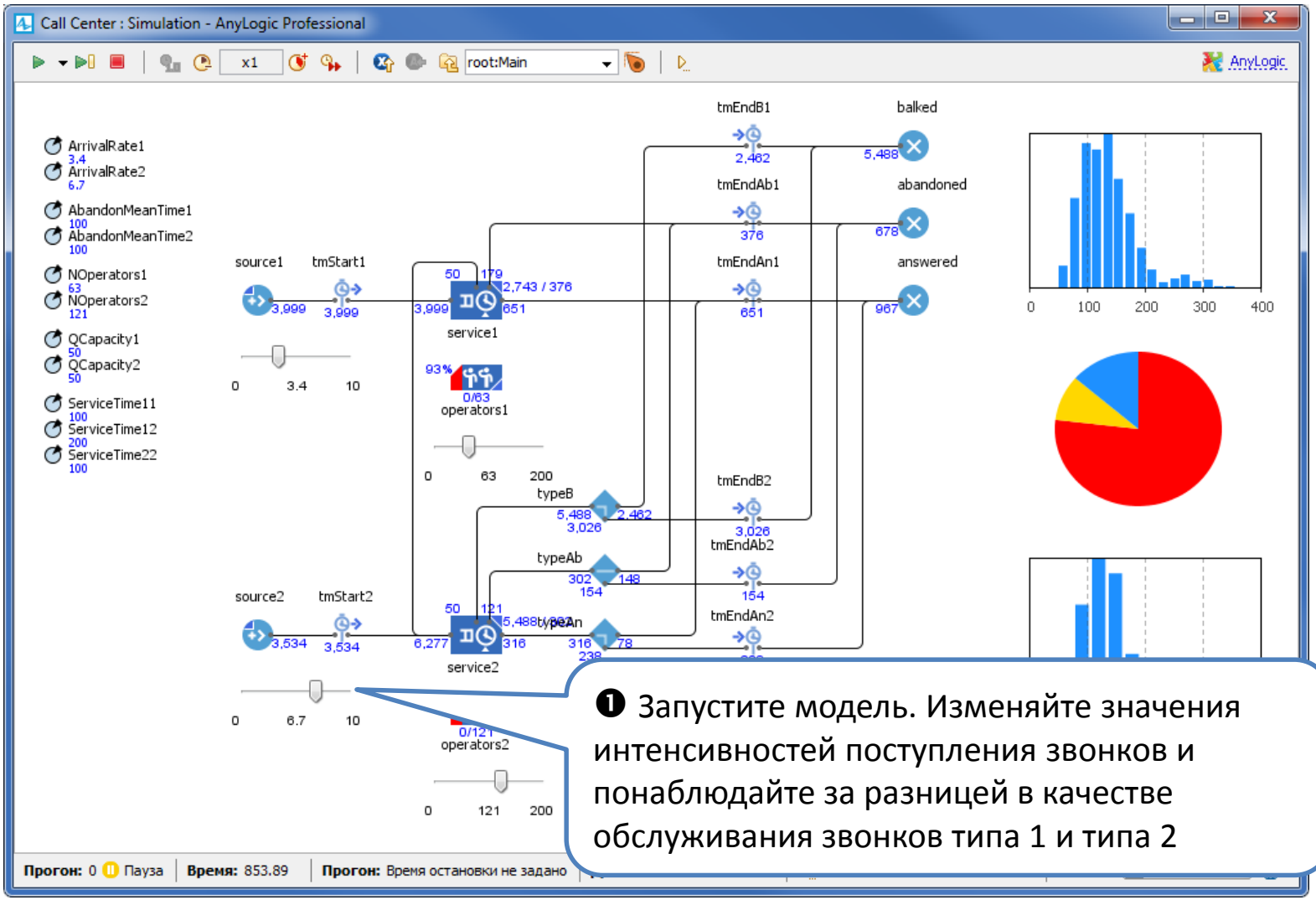
❸ Аналогично создайте еще два бегунка, связанных с параметрами *NOperators1* и *NOperators2*, с минимальным значением 1 и максимальным 200.

Чтобы связать элемент управления, в нашем случае, **Бегунок**, с каким-либо параметром, сначала поставьте флажок для опции **Связать с**, а затем выберите параметр из выпадающего списка, расположенного рядом.

Затем задайте минимальное и максимальное значения параметра, который регулируется этим бегунком.

Щелкните кнопку **Добавить метки**, чтобы видеть эти значения, а также то значение, которое выбрано для параметра с помощью бегунка (*value*). AnyLogic добавит три элемента **Текст** под бегунок.

Колл-центр. Фаза 4. Шаг 2



❶ Запустите модель. Изменяйте значения интенсивностей поступления звонков и понаблюдайте за разницей в качестве обслуживания звонков типа 1 и типа 2

Колл-центр. Вопросы

1. Предложите другой способ перенаправления звонков типа 1 операторам типа 2, который бы не использовал функцию вытеснения объекта Service.
2. Измеренное нами “время пребывания в системе” включает в себя как время ожидания, так и время обслуживания. Как бы Вы измерили только время ожидания?
3. При изменении параметров модели с помощью бегунков статистика всегда содержит данные, запомненные как до, так и после изменения. Как бы Вы удалили устаревшую статистику при изменении значения с помощью бегунка?

1. **Перенаправление звонков:** Выключите вытеснение у обоих объектов **Service**. С помощью объекта **SelectOutput** направляйте входящие звонки типа 1 в *service1*, только если *service1.queueSize() < service1.queueCapacity*, а иначе направляйте их в другой аналогичный объект **SelectOutput** с условием *service2.queueSize() < service2.queueCapacity*. При выполнении этого условия агенты перейдут в *service2*, при невыполнении – покинут систему через *balked1*.

2. **Время ожидания.** Использовать для данной задачи объекты *TimeMeasureStart/End* не получится, поскольку их нельзя поместить внутрь объекта *Service* между внутренней очередью этого объекта и его объектом задержки. Однако Вы можете добавить дополнительный параметр *arrivalTime* в тип агента *Call* и запоминать в этом параметре текущее время в коде параметра *Действие при выходе* объекта **Source**. Тогда в поле *Действие при начале задержки* объекта **Service** Вы сможете вычислить время ожидания как *time() - agent.arrivalTime* и добавить это время в объект сбора статистики и/или распределение.

3. **Сброс статистики.** Если сбросить собираемое объектом **TimeMeasureEnd** распределение времени можно легко (например, вызвав *tmEndA2.distribution.reset()*), то сбросить счетчик объекта **Sink** Вы не можете. Поэтому Вам нужно будет создать свои собственные счетчики (переменные типа *int*), увеличивать их значения в объектах **Sink** и сбрасывать их при изменении значения соответствующего бегунка. Имеет смысл написать функцию *resetStats()* и вызывать ее в поле **Действие** каждого бегунка.

Колл-центр. Вопросы. Подсказки