

О.Н. Ярыгин

# **ИМИТАЦИОННОЕ МОДЕЛИРОВАНИЕ ДЕЯТЕЛЬНОСТИ ПРЕДПРИЯТИЯ В EXTEND LT**

---



Министерство образования и науки Российской Федерации  
Тольяттинский государственный университет  
Институт финансов, экономики и управления  
Кафедра «Менеджмент организации»

О.Н. Ярыгин

# **ИМИТАЦИОННОЕ МОДЕЛИРОВАНИЕ ДЕЯТЕЛЬНОСТИ ПРЕДПРИЯТИЯ В EXTEND LT**

Учебно-методическое пособие

Тольятти  
Издательство ТГУ  
2012

УДК (658.5: 004.4).001.572

ББК 65.005

Я799

Рецензенты:

д.э.н., профессор Волжского университета им. В.Н. Татищева

*А.И. Афоничкин;*

д.э.н., профессор Тольяттинского государственного университета

*А.А. Руденко.*

**Я799** Ярыгин, О.Н. Имитационное моделирование деятельности предприятия в Extend LT : учеб.-методическое пособие / О.Н. Ярыгин. — Тольятти : Изд-во ТГУ, 2012. — 123 с. : обл.

В учебно-методическом пособии представлены основные приемы имитационного моделирования производственной деятельности предприятий средствами Extend LT. Пособие предназначено для преподавания и изучения дисциплин «Моделирование систем управления», «Оптимизация управленческих решений» (по направлению 221400.62 «Управление качеством»), «Моделирование производственных процессов», «Имитационное моделирование производственно-хозяйственной деятельности предприятия» (по направлению 080502.65 «Экономика и управление на предприятиях машиностроения»).

Работа выполнена в рамках задания по теме № 461201 «Методология аналитической деятельности управления образованием».

УДК (658.5: 004.4).001.572

ББК 65.005

Рекомендовано к изданию научно-методическим советом Тольяттинского государственного университета.

© ФГБОУ ВПО «Тольяттинский  
государственный университет», 2012

## ВВЕДЕНИЕ

Необходимость изучения имитационного моделирования связана с тем, что в современной экономике и менеджменте будущим специалистам потребуется работать в системах, которые характеризуются высокой динамикой и недетерминированностью. При этом оказывается практически невозможным применение традиционных математических моделей для описания и исследования деятельности организаций или предприятий. Мощным инструментом, адекватным по сложности изучаемым объектам, становится имитационное моделирование, выполняемое с помощью специализированных компьютерных систем. В предлагаемом учебно-методическом пособии вводятся основные понятия имитационного моделирования, необходимые для формирования компетентности в области проектирования, создания и применения математических моделей в экономике и менеджменте. Применение основных блоков Extend LT иллюстрируется тщательно отработанными примерами с подробным описанием особенностей их использования. Система визуального моделирования Extend LT представляется как средство моделирования и анализа динамических недетерминированных систем. Описываются средства интерпретации результатов моделирования и принятия управленческих решений на основе этих результатов. Изложение доступно для студентов, владеющих основами статистики и высшей математики, при этом не требуется владения навыками программирования.



## 1. ИМИТАЦИОННОЕ МОДЕЛИРОВАНИЕ КАК ИНСТРУМЕНТ АНАЛИЗА СЛОЖНЫХ СИСТЕМ

Создание имитационных моделей и проведение имитационных экспериментов позволяют рассмотреть разнообразные возможные последствия принимаемых управленческих решений, произвести сравнение альтернативных вариантов и выбрать наилучший из них.

Имитационные модели, как и аналитические, создаются не только с целью оптимизации, но и с целью изучения поведения моделируемой системы и её подсистем и элементов.

Основными этапами имитационного моделирования можно считать:

- логическое моделирование, в результате которого определяются логика и цель взаимодействия системы и внешней среды, компонентов системы;
- структурное моделирование, позволяющее определить состав и взаимосвязи элементов системы, функции отдельных элементов и состав подсистем;
- функциональное моделирование, которое позволяет создать модель каждого функционального блока с учетом установленной структуры и логики взаимодействия;
- создание имитационной модели, позволяющее с помощью некоторого программного средства имитировать работу каждого элемента, подсистемы и системы в целом;
- собственно имитационное моделирование, позволяющее проводить анализ «что-если» для всей системы и отдельных элементов, анализ чувствительности различных подсистем; причем это моделирование показывает поведение моделируемой системы в моделируемом времени, в условиях неопределенности, когда математические модели либо слишком сложны, либо недостаточно адекватны;
- усовершенствование имитационной модели на основе результатов вычислительных экспериментов, проводимых с моделью, а также на основе новых данных, получаемых в ходе моделирования и из других источников.

Имитационное моделирование особенно эффективно при исследовании динамических систем в условиях неопределенности,

в частности, таких стохастических систем, как системы массового обслуживания (СМО), описываемых дискретными состояниями, наступающими в дискретные моменты времени, с заданными вероятностными распределениями. Вероятностные распределения, используемые в имитационных моделях, получаются либо из математической модели системы, либо строятся на основании статистических данных.

Обобщенная классификация моделируемых систем и классов применяемых моделей представлена в табл. 1.

Таблица 1

Классификация систем и методов моделирования

Динамизм	Статические	Динамические
Определенность		
Детерминированные	Алгебраические математические модели, математическое (линейное, нелинейное, целочисленное) программирование	Сбалансированные динамические системы, системы дифференциальных уравнений, рекуррентные формулы
Недетерминированные (вероятностные, нечеткологические)	Вероятностные распределения, нечеткологический вывод	Имитационное моделирование: метод Монте-Карло, системная динамика, компьютерное моделирование

Из табл. 1 видно, что имитационное моделирование оказывается необходимым при исследовании сложных динамических, возможно недетерминированных систем, каковыми являются большинство социально-экономических и производственных систем.

## 2. EXTEND LT. МОДЕЛИРОВАНИЕ КАК КОНСТРУИРОВАНИЕ

Как можно было увидеть при построении моделей производственной деятельности средствами Excel, электронные таблицы имеют существенные недостатки для моделирования дискретных событий. Во-первых, для больших моделей рабочий лист Excel быстро заполняется ячейками, занятыми формулами. Во-вторых, имитационные модели дискретных событий в Excel плохо масштабируются. Электронные таблицы плохо приспособлены для моделирования стохастических явлений и динамических процессов. Последние два недостатка с успехом преодолеваются специализированными системами имитационного моделирования, такими как GPSS World, однако они требуют навыков программирования и, по сути, сводятся к программированию на специализированном языке моделирования. Таким образом, опыт моделирования в Excel и GPSS подсказывает следующие пожелания для новой современной системы моделирования.

1. Представление модели должно быть компактным, для того чтобы избежать формирования гигантских таблиц (Excel) или длинных текстов программ (GPSS World), неудобных для манипулирования и отладки.
2. Модели должны быть просты для проведения имитации и получения ее результатов без обременительного программирования (формулы в ячейках Excel, текстовые модели GPSS World).
3. Для удобства построения моделей и документирования среда разработки моделей должна иметь графический интерфейс пользователя.

Все перечисленные требования выполняются системой имитационного моделирования Extend. Мы рассматриваем упрощенную, «студенческую» версию, которая, впрочем, позволяет освоить практически все возможности системы и создавать весьма сложные модели.

Система имитационного моделирования Extend LT позволяет строить модели и выполнять моделирование дискретных событий в визуальном режиме.

Имитационные модели строятся из блоков (представляющих аналогию блоков системы GPSS), но блоки имеют вид пиктограмм,

которые размещаются в окне модели и соединяются друг с другом линиями связи. Таким образом, наглядно и одновременно реализуются этапы структурного и функционального моделирования.

Модель строится подобно собиранию «puzzle» или детского конструктора из разнообразных элементов, моделирующих отдельные элементарные процессы, математические операторы и функции, технические устройства и прочее.

Дискретные модели в Extend являются объектно ориентированными и строятся с помощью предопределенных стандартных блоков.

*Блок* — это объект Extend, предназначенный для определения действия, процесса, накопления данных, статистического обобщения. Другими словами, это функциональный элемент, предназначенный для моделирования.

Связанные между собой блоки Extend формируют сеть взаимодействующих активностей. Элементы, моделирующие динамические объекты реальной системы (например, клиенты или задания), поведение которых представляет интерес, называются *транзактами*. Они перемещаются через связанную сеть блоков, а Extend собирает статистические данные об этом потоке. Extend отслеживает любое ожидание или время задержки при перемещении транзактов.

В Extend также имеются блоки специального назначения, используемые для сведения в таблицы статистических данных (таких как длина очереди и время ожидания) и построения на их основе диаграмм. Предопределенные блоки приложения Extend хранятся в файлах-библиотеках. Поэтому при построении новой модели следует открыть необходимые библиотеки, из которых будут копироваться блоки в создаваемую модель.

Существуют четыре типа *связей* между блоками: два типа для входа и выхода транзактов, и два — для ввода и вывода значений. Связи для входа и выхода транзактов образуют пути, по которым транзакты перемещаются через блоки модели. Эти связи изображаются в виде *двойных* линий, соединяющих маленькие коннекторы (разъемы), расположенные на одной из сторон каждого блока. Связи для ввода и вывода значений передают данные и параметры

транзактов между блоками и изображаются в виде *одинарных* линий, соединяющих коннекторы блоков.

Каждый блок имеет диалоговое окно для задания параметров и просмотра результатов после испытаний модели. Доступ к диалоговому окну блока осуществляется двойным щелчком на пиктограмме блока. В качестве примера на рис. 2.1 показан блок *Activity, Delay* (Активность, Задержка), используемый обычно для задержки транзакта в течение некоторого времени, что моделирует какое-либо обслуживание объекта в реальной системе. После задержки транзакт переходит к следующему блоку.

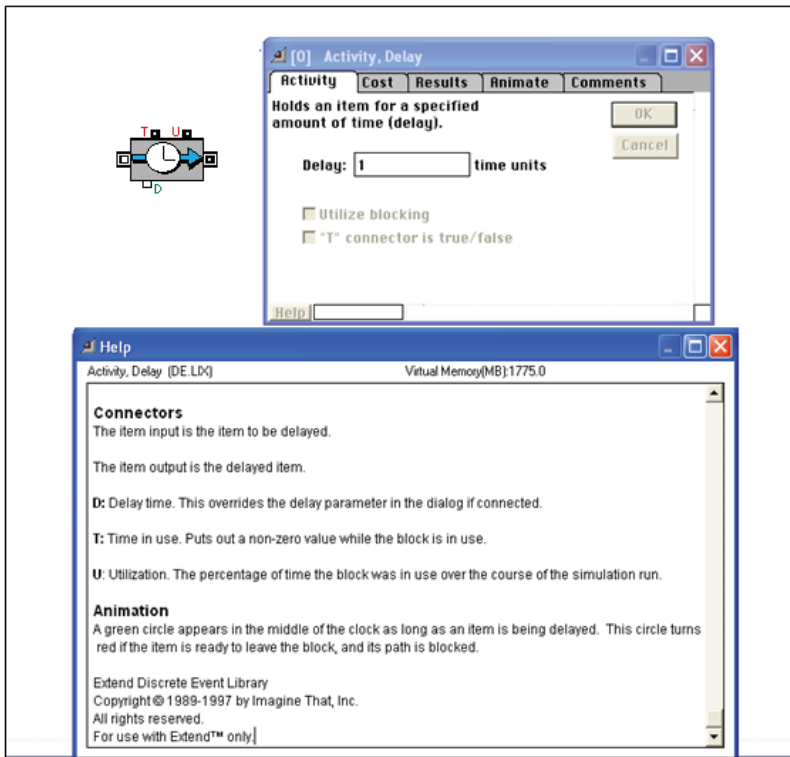


Рис. 2.1. Блок Activity, Delay и его диалоговое окно

Как только транзакт, моделирующий отдельного клиента, входит в блок, он задерживается в нем ровно на 1 единицу времени (минуту), как указано в поле Delay (Задержка) в диалоговом окне бло-

ка. Для получения более подробной информации о блоке *Activity, Delay* можно щелкнуть на кнопке **Help** (Справка), находящейся в этом диалоговом окне. В диалоговом окне Help можно увидеть, для каких целей предназначены три других коннектора, а также другую информацию о данном блоке.

После окончания процесса имитации во вкладке Results (Результаты) диалогового окна будет показана информация, записанная в блоке во время имитации, например, среднее значение использования блока (процент занятости сервиса – Utilization) и количество транзактов, вышедших из блока (т. е. количество обслуженных клиентов). Все блоки Extend обладают одинаковым базовым составом компонентов:

- a) входными и выходными коннекторами, используемыми для задания маршрутов транзактов в системе;
- b) диалоговым окном блока для установки параметров и записи собранных в процессе имитации статистических данных;
- c) коннекторами для входа или выхода данных во время испытаний модели.

Графический интерфейс пользователя системы Extend обеспечивает быстрое создание моделей и обучение пользователей, не владеющих программированием. Графический интерфейс понятен и доступен пользователям Windows-приложений без каких-либо дополнительных объяснений.

Extend позволяет устанавливать временные параметры моделирования (единицу измерения времени, период моделирования, масштаб модельного времени). Режим анимации позволяет видеть функционирование модели в динамике, что существенно упрощает процесс отладки и модификации модели.

### 3. ОСНОВЫ МОДЕЛИРОВАНИЯ В EXTEND

Цель имитационного моделирования обычно состоит в том, чтобы оптимизировать систему так, чтобы в реальном процессе использовалось меньше ресурсов и времени. Системы обслуживания с использованием очередей называют *системами массового обслуживания* (СМО).

- Очереди — последовательности, с которыми приходится сталкиваться в той или иной форме практически во всех аспектах деятельности. В очереди могут быть клиенты, пакеты данных, автомобили, задания и любые другие дискретные объекты, моделируемые транзактами.

- Транзактами называются динамические элементы имитационных моделей, которые моделируют обслуживаемые объекты, перемещающиеся в модели между функциональными блоками.

- Каналами обслуживания (Серверами) являются механизмы, с помощью которых обрабатываются объекты последовательности, а затем направляют их далее. Сервером может быть торговый персонал, машина, компьютер или любое другое устройство, которое задерживает объект на время обработки. Каналы обслуживания моделируются блоками, обеспечивающими заданные задержки транзактов.

В качестве примера рассматривается модель под названием «Bank Line» (в Extend модель представлена в файле C:\ExtendLT\Tutorial\BankLine.mox), которая моделирует поток банковских клиентов, приходящих в банк и оказывающихся в очереди к кассирам. На этой модели также демонстрируются результирующие изменения модели с целью снижения количества кассиров. Случайное прибытие клиентов моделируется с помощью настраиваемых генераторов случайных чисел.

#### **Файлы, создаваемые в Extend**

В процессе моделирования в Extend создаются три вида файлов:

- модели,
- библиотеки,
- текстовые файлы.

На рис. 3.1 показаны файлы Extend и их иконки.



Рис. 3.1. Типы файлов в Extend: .MOX – модели Extend; .LIX – библиотеки; .TXT – текстовые файлы

### **Обзор процесса моделирования**

Модель состоит из компонентов, называемых *блоками*, и связей между ними. Библиотека является хранилищем блоков. Для добавления функционального блока в модель блок копируется из библиотеки. Текстовые файлы хранят данные в форме, которая может быть прочитана из txt-файла.

Модели открываются и закрываются в меню «File». При открытии модели или создании новой окно модели появляется на экране как обычный документ. Три последних открытых файла, независимо от того, открыты они по-прежнему или нет, перечисляются в нижней части меню «File».

Библиотеки открываются и закрываются в меню Library. Библиотеки имеют свои собственные библиотечные окна. Открытые библиотеки также перечисляются в нижней части окна Library.

Создаваемые в Extend текстовые файлы открываются в меню «File». Эти файлы могут быть прочитаны и отредактированы через Extend или любым текстовым редактором и большинством других приложений.

### **Запуск Extend и открытие модели**

Приложение Extend запускается так же, как любое другое Windows-приложение, – с помощью двойного щелчка на соответствующей иконке (рис. 3.2).

При запуске Extend открывается пустое окно новой модели. Если не открывается уже существующая модель, то Extend не открывает какие-либо библиотеки при запуске.





Рис. 3.2. Иконка Extend

Далее этапы моделирования в Extend рассматриваются на примере учебной модели (модель «BankLine.mox» в папке Extend \Tutorial).

Открытие модели:

- Файл (File) – Открыть (Open);
- Найти и открыть папку Extend \ Tutorial;
- Найти модель «BankLine.mox» и открыть ее.

Модель появляется на экране в следующем виде:

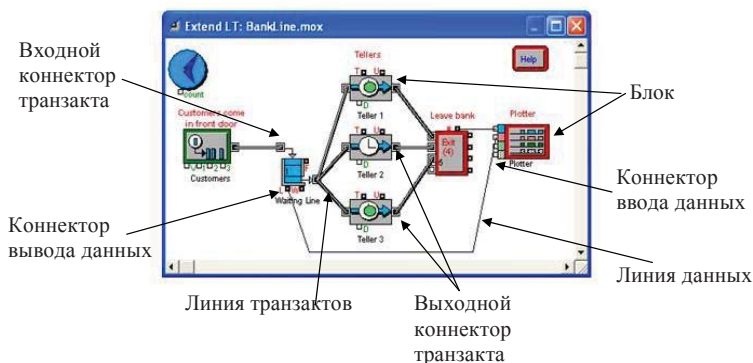


Рис. 3.3. Типичный вид модели Extend

Модель является лишь документом, как и другие окна. Её можно прокручивать горизонтально или вертикально с помощью полос прокрутки, можно уменьшить или увеличить окно, а также закрыть окно с помощью стандартных оконных функций.

### **Типовая модель**

На примере «Bank Line» видны основные компоненты Extend-модели. Наиболее важными компонентами модели являются:

- блоки;
- библиотеки, где хранятся блоки;
- диалоги, связанные с каждым блоком;
- коннекторы (разъемы) на каждом блоке;
- связи между блоками.

### **Блоки**

Большинство блоков в Extend имеют значок (рис. 3.3), диалоговое окно для ввода и просмотра данных, а также коннекторы. Блок определяет действие или процесс. Информация поступает в блок и обрабатывается программой в блоке. После этого блок передает информацию и транзакты на следующий блок в модели.

Блок в Extend похож на блок в блок-схеме и используется для моделирования некоторой функции модели, что позволяет одновременно с созданием имитационной модели выполнять и структурное моделирование.

Некоторые блоки представляют собой только источник информации, которая передается в другие блоки модели. Большинство блоков изменяют информацию, которая проходит через них. Блоки, называемые *иерархическими*, объединяют группы других блоков.

Блок Plotter (Плоттер) с правой стороны модели «Bank Line» является блоком вывода, который принимает информацию от модели и представляет её в графической форме.

Значок блока, как правило, изображает собой функцию блока. Например, в модели «Bank Line» блок под названием «Waiting line» является блоком «Queue, FIFO» из библиотеки Discrete Event (Дискретные события). Его значок изображает вход транзакта в очередь в верхней левой части и выход в том же порядке внизу справа.

В модели «Bank Line» имеется девять блоков, некоторые из которых являются дубликатами. В модели можно использовать более одной копии блока данного типа, например три блока «Teller» в модели «Bank Line».

### **Библиотеки**

Библиотеки являются хранилищами блоков. Все компоненты блока (его программы, иконки, диалог и т. д.) хранятся в библиотеке. Когда блок включается в модель, сам он не копируется в неё. Вместо этого создается ссылка на блок, которая информирует биб-

лиотеку о включении. Extend также хранит данные, которые вводятся в блок в диалоге в данной модели.

Когда модель сохраняется на диске, Extend сохраняет имена из блоков, а также места расположения библиотек, хранящих блоки. При открытии модели Extend автоматически открывает необходимые библиотеки. Например, когда открывается модель «Bank Line», Extend открывает три библиотеки: Discrete Event, Generic и Plotter (de.lix, generic.lix и plotter.lix).

Есть три способа открытия библиотеки:

- 1) автоматическое открытие библиотеки при запуске модели;
- 2) открытие библиотеки через пункт Open Library меню Library;
- 3) в команде Preferences в меню Edit можно указать до семи библиотек, которые будут автоматически открываться при запуске Extend.

После открытия библиотек меню Library содержит подменю, в котором все блоки библиотеки появляются рядом с её именем. В подменю блоки рассортированы по типам, они добавляются к модели с помощью щелчка на названии блока в меню библиотеки.

### ***Диалоги***

Большинство блоков имеют своё диалоговое окно (диалог). Диалоговые окна используются для ввода значений параметров перед запуском модели, а также для просмотра результатов моделирования.

Чтобы открыть диалог одного блока, достаточно дважды щелкнуть по значку блока.

Если дважды щелкнуть на одном из значков Teller (Кассир), то откроется диалог для этого блока (рис. 3.4).

Этот диалог позволяет настроить счетчик задержки блока, установив длительность моделируемой работы кассира. Время обслуживания задается в поле Delay (2 минуты). Индекс [3] в строке заголовка является номером блока и указывает, что это четвертый блок в таблице блоков модели (нумерация блоков начинается с 0).

Диалог имеет несколько страниц или вкладок, содержащих более подробную информацию и настройки.

Нажатие кнопки Cancel (Отмена) закрывает диалог и отменяет все сделанные изменения. (В блоках, которые имеют графические

участки, например блок ввода данных Input Data, изменения не могут быть отменены.) Нажатие кнопки ОК или закрытие окна кнопкой в левом верхнем углу указывает на то, что надо сохранить изменения.

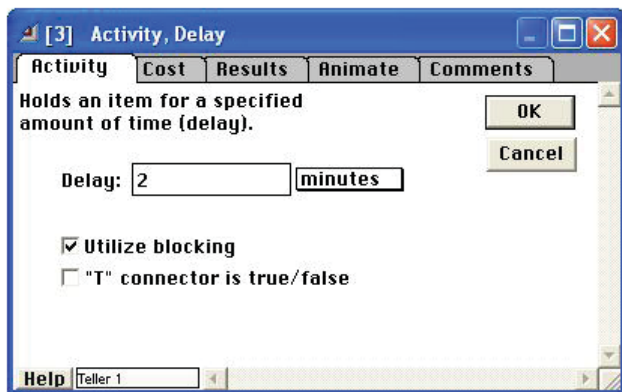


Рис. 3.4. Диалог (диалоговое окно) блока

В поля диалогов можно вводить текст или цифры. При вводе текста, например, во вкладке комментария Comments tab текст автоматически переносится по мере ввода. Можете перейти на следующую строку текста, нажав Enter. При вводе номера с десятичными точками Extend использует десятичный разделитель, заданный в настройке Windows. Можно оставлять диалоги открытыми при запуске модели, но это немного замедляет моделирование. Некоторые диалоги сообщают параметры модели, так что можно использовать диалоги для показа величин, которые наблюдаются в ходе моделирования. Можно даже изменять настройки в диалоге, когда модель запущена.

Когда нажимается кнопка «ОК» во время выполнения моделирования, блок получает информацию об изменении значений на следующем шаге. Однако, когда набирается текст или вводится число в поле, модель останавливает работу и возобновляет её тогда, когда закончится ввод и будет нажата кнопка Resume на панели управления моделью в нижней части экрана.

### ***Коннекторы и соединения***

Большинство блоков в Extend имеют входные и выходные коннекторы, (маленькие квадраты, прилегающие к значку блока). Через коннекторы блоки обмениваются транзактами и данными. Входные

и выходные коннекторы, как правило, заранее определены, и их конкретные функции известны. Блок может иметь множество входных и/или выходных коннекторов или не иметь их вовсе. Коннекторы каждого блока описаны в его Справке, вызываемой кнопкой Help.

Соединительные линии передают поток информации из блока в блок в составе данной модели и тем самым доопределяют структуру моделируемой системы. Имитационное моделирование представляет собой последовательность расчетов и действий, которые выполняются в соответствии с соединениями. Каждое повторяющееся действие называется *шаг* или *событие*. Количество шагов при моделировании не ограничено.

Перед запуском модели период моделирования (в модельном времени) задается в пункте **Simulation Setup** меню **Run**. Затем моделирование начинается нажатием **Run Simulation**.

Блок *Executive*, похожий на часы, в левом верхнем углу модели обрабатывается первым, блок, который назван «**Customers come in front door**» (Клиенты входят в дверь), обрабатывается вторым и так далее, вплоть до блока под названием *Plotter* (Графопостроитель). После этих шагов цикл моделирования повторяется.

### **Исследование модели «Bank Line»**

Перед построением модели следует определить, что моделируется и почему. Модель «Bank Line» представляет собой имитацию обращения клиентов в банк, ожидание в очереди, обслуживание кассирами, выход из банка. Цель моделирования состоит в том, чтобы установить:

- а) среднее количество клиентов, проходящих за 1 час при различных условиях;
- б) длину очереди в любой заданный момент времени.

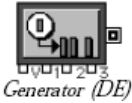
Во многих моделях интерес представляет то время, которое необходимо транзакту (в данном случае – клиенту), чтобы пройти от начала модели до её конца. Блоки могут задерживать транзакт. Например, в модели «Bank Line» есть задержка на расчеты, поскольку работа кассира занимает много времени.

Как в большинстве моделей Extend, действия в этой модели идут по пути, заданному соединениями.

## Описание блоков и библиотек

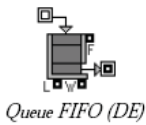


Блок *Executive* из библиотеки *Discrete Event* (Дискретных событий) должен быть включен во все модели дискретных событий. Он задает общее течение времени в модели, выполняя роль тактового генератора.



Блок, подписанный «Customers come in front door», представляет собой блок *Generator* из библиотеки *Discrete Event*.

Блок *Generator* используется для создания транзактов (в данном случае это клиенты, входящие в банк) с интервалом времени, задаваемым в настройках диалога. Блок изначально настроен так, будто новый клиент приходит каждые 39 секунд.



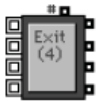
Когда клиенты входят в банк, они ждут своей очереди. В модели такого рода последовательность называется очередью. Если кассир свободен, клиент автоматически выходит из очереди и переходит к нему. Если кассир занят, клиенты ждут в очереди до тех пор, пока кассир не освободится.

Блок *Queue, FIFO* библиотеки *Discrete Event* моделирует очередь типа *First In First Out*. Транзакты входят в блок через входной коннектор и выходят в момент освобождения следующего блока. Через выводные коннекторы блок сообщает длину очереди (коннектор *L*) и время ожидания вышедшего транзакта (коннектор *W*). Если следующий блок свободен при входе транзакта в блок *Queue, FIFO*, то никакой задержки не будет.



Клиент выходит из очереди к первому доступному кассиру, моделируемому блоком *Activity, Delay* из библиотеки *Discrete Event*. Если свободны два и более кассира, то клиент может произвольно перейти к любому свободному кассиру.

В модели транзакт переходит из блока *Queue, FIFO* в блок *Activity, Delay* и задерживается в блоке на время, равное времени обслуживания клиента кассиром, заданное в диалоге блока *Activity, Delay* (2 минуты).



*Exit (4) (DE)*

После задержки клиент идет к блоку **Exit (4)** (4 выхода) из библиотеки **Discrete Event**, который выводит транзакты из моделирования. Верхний выводной коннектор выдает общее число клиентов, которые вошли в блок.



*Plotter DE (Plotter)*

Блок **Discrete Event Plotter** из библиотеки **Plotter** графически отображает результаты моделирования. Можно запустить модель без каких-либо плоттеров, но плоттер является наиболее распространенным методом сбора и представления результатов моделирования.



*Help (Generic)*

Блок **Help** (из библиотеки **Generic**) является одним из средств хранения и отображения пользовательской справочной информации об этой модели.

### **Запуск модели**

Модель запускается выбором пункта Run Simulation из меню Run.

Модель «Bank Line» работает в течение имитируемых (модельных) 60 минут. Пока модель Extend работает, выводятся результаты на плоттере. Линии передают длину очереди (красный коннектор) и число людей, которые покидают банк (синий коннектор). График, отображающий продолжительность ожидания, постепенно повышается. Завершив моделирование, Extend оставляет плоттер на экране. Это основной метод для представления результатов моделирования.

### **Моделирование состояния**

После запуска модели Extend отображает некоторую информацию о состоянии системы на экране монитора. В зависимости от скорости компьютера можно увидеть следующие сообщения: «Please Wait» («Подождите»), «Checking Data» («Проверка данных») или «Initializing Data» («Инициализация данных»).

Эти сообщения информируют о состоянии Extend.

После запуска модели Extend показывает строку состояния (рис. 3.5).



*Status Bar*

Рис. 3.5. Строка состояния модели во время моделирования

**Цифры и песочные часы** – оценка фактического времени, оставшегося до завершения моделирования (в формате «минуты : секунды»).

**Часы** с циферблатом показывают текущее время в модели в модельных единицах времени.

**Steps (Шаги)** – общее количество шагов или имитационных событий в модели.

**Run 0** – показывает число запусков моделирования, если используется многократный запуск модели (нумерация начинается с 0). Эти значения определяются через диалог *Simulation Setup* меню *Run*.

Следует отметить, что оставшееся время, отображающееся в строке состояния, является лишь оценкой. Для непрерывного моделирования, как правило, оно весьма точно. В случае же дискретного моделирования (например, «Bank Line») оно может быть весьма приблизительным, особенно на начальном этапе, поскольку невозможно узнать, когда произойдут события и сколько времени потребуется для завершения моделирования.

Для того чтобы остановить моделирование «на ходу», надо нажать кнопку *Stop* в строке состояния или использовать клавишу *Ctrl. Extend* запрашивает подтверждение. Если используется многократный запуск модели, то можно остановить именно текущий сеанс или все сразу.

Чтобы сделать паузу в моделировании, надо нажать кнопку *Pause*. Эта кнопка превратится в *Resume*-кнопку, которая запускает продолжение моделирования. Можно также остановить и приостановить моделирование из меню «*Run*».

Если выполняется моделирование с включенной анимацией, то она может происходить быстрее, чем требуется. Тогда можно использовать кнопку *Slower* для замедления отображения анимации. Эта кнопка превращается в кнопку *Faster*, которую можно использовать для возврата к нормальной скорости.

### **Плоттеры**

*Extend* располагает настраиваемыми плоттерами, которые можно использовать в моделях. Плоттеры показывают как графически представленные числовые функции, так и таблицы числовых значений. Плоттер в модели «Bank Line» – это блок *Plotter, Discrete*



Event из библиотеки Plotter. При запуске модели Extend показывает графики на экране. Плоттер для модели «Bank Line» выглядит следующим образом (рис. 3.6).

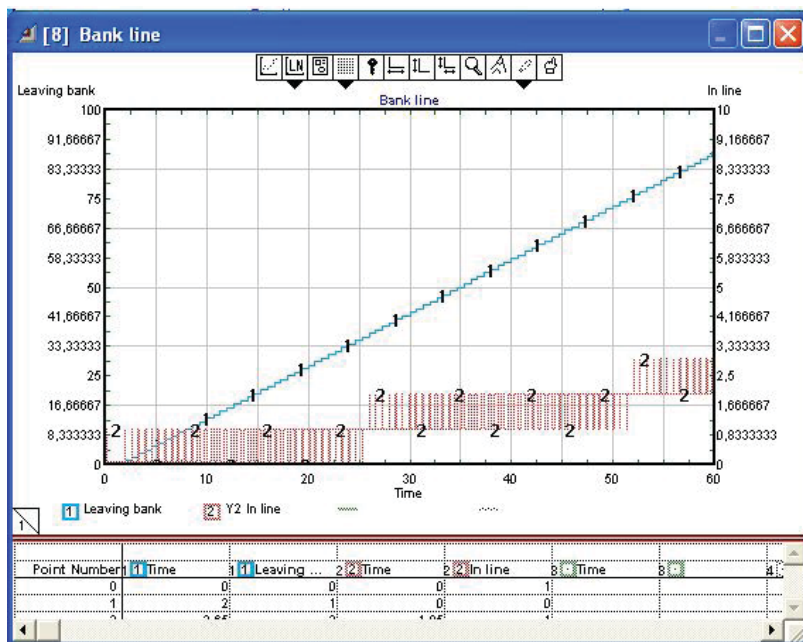


Рис. 3.6. Графики изменения длины очереди (2) и количества клиентов, покидающих банк (1), построенные блоком Plotter, DE

В этом примере два графика с двумя осями ординат. График 1 и левая ось показывают количество людей, которые покинули банк в течение имитируемого часа; график 2 и правая ось показывают длину общей очереди к кассирам. Заметим, что у каждой оси своя шкала измерения.

Можно наблюдать значения, перемещая курсор вдоль графиков на плоттере. Соответствующие значения отображаются в верхней строке таблицы данных плоттера.

В нижней части окна графопостроителя показываются данные, по которым построены графики. Чтобы увидеть числовые значения для двух графиков, можно прокручивать эту таблицу вниз.

Extend может запомнить графики последних четырех построений. Можно просмотреть другие построения, нажав на маленький символ загнутого угла страницы в нижнем левом углу графика.

### ***Изменение параметров***

Одной из сильнейших функций Extend является возможность изменять параметры «на лету» и просматривать результаты моментально. С помощью плоттера, который запоминает предыдущие четыре графопостроения, можно легко сравнить результаты после изменения параметров.

Можно изменять параметры модели как путем добавления или удаления блоков, так и путем изменения значений параметров блоков в диалоговом окне.

Для удаления блока надо выделить его и удалить, нажав на кнопку Delete. Вместе с блоком удаляются и его линии соединения.

Прямым методом передачи данных в модель является задание параметров через диалог блока. Если в диалоге вносятся изменения и модель сохраняется на диске, новые значения параметров сохраняются вместе с ней. Можно изменять значение диалогового параметра во время работы модели. Но для этого следует остановить моделирование. Чтобы его продолжить, следует щелкнуть кнопку Resume в строке состояния или выбрать Resume в меню Run.

### ***Пример изменения параметров блока *Generator****

Блок ***Generator*** (Генератор) порождает транзакты в процессе моделирования. Диалог блока позволяет установить средний интервал времени между возникновением транзактов (прибытиями клиентов), а также вероятностное распределение этих случайных интервалов (рис. 3.7).

Для рассматриваемой модели в меню запуска **Simulation Setup** глобальными единицами времени заданы «минуты». Тем не менее время между входами клиентов в банк может быть задано в любых единицах. Тогда время между порождением транзактов придется задать в тех же единицах. В диалоге блока ***Generator*** заданы «секунды» (поле *Time units*), и этот блок установлен генерировать новый транзакт каждые 39 секунд (поле *Constant*). Можно изменить величину *Constant* и проверить, как это влияет на длину очереди и количество обслуживаемых клиентов.

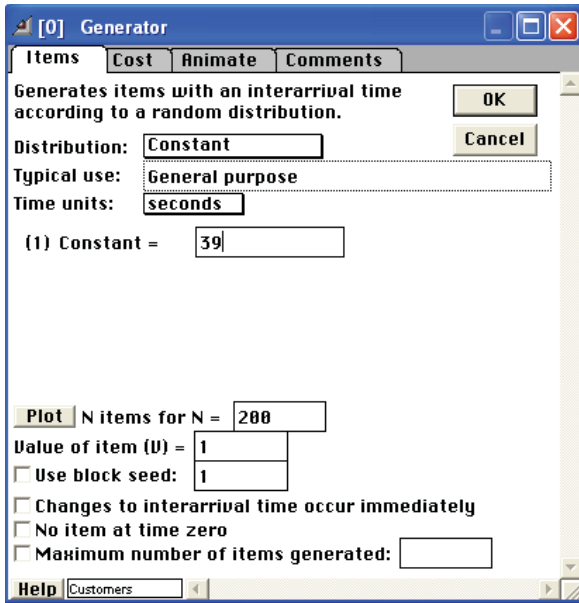


Рис. 3.7. Диалог блока *Generator*

Можно также выбрать другое распределение интервалов поступления транзактов. Для этого достаточно щелкнуть на поле *Distribution* и в выпадающем меню (popup menu) выбрать подходящее распределение. Например, если выбрать «real, uniform» («вещественные числа, равномерно распределенные»), то в диалоге появятся два поля, которые позволят вам задать диапазон для изменения вещественной случайной величины.

В поле «*Value of items (V)*» задается количество транзактов, порождаемых при каждом срабатывании генератора. В рассматриваемом случае это значение равно 1.

### ***Добавление случайных факторов***

Если бы клиенты банка всегда приходили через точные равные интервалы времени, то не составило бы труда рассчитать количество кассиров, обеспечивающее отсутствие очереди, с одной стороны, и постоянную занятость работников – с другой. В реальном мире, конечно, люди приходят в банк в произвольные моменты времени. Extend обеспечивает имитацию случайных интервалов

между транзактами, чтобы можно было моделировать реальные случайные процессы.

Чтобы задать случайное время между транзактами (время, через которое клиенты входят в банк), надо вместо Constant выбрать в блоке Generator функцию распределения Normal. Это позволит генерировать интервалы времени с нормальным распределением. Можно ввести в поле Mean (Среднее) значение 39, а в поле Std Dev (Среднеквадратическое отклонение) значение 15 (рис. 3.8).

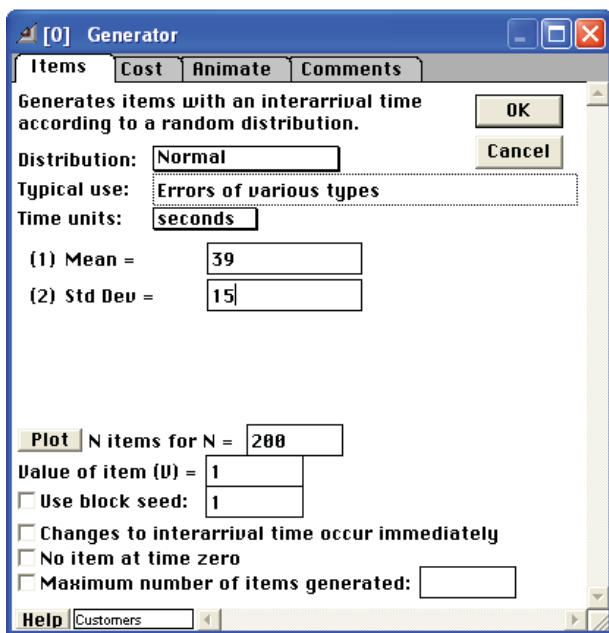


Рис. 3.8. Диалог блока *Generator*. Нормальное распределение

Среднее время 39 моделирует прибытие приблизительно каждые 39 секунд. Среднеквадратическое отклонение 15 означает, что разброс значений около 39 будет довольно широк.

Каждый прогон модели (Run Simulation) теперь будет отличаться от предыдущего. При этом форма линии (2), описывающая количество клиентов в очереди, значительно изменится (рис. 3.9).

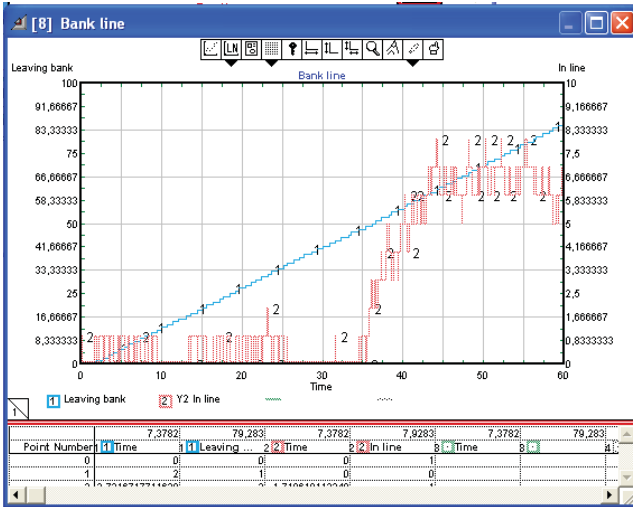


Рис. 3.9. Результат моделирования при нормальном распределении интервалов между транзактами

### Другие модификации модели

Модель «Bank Line» может совершенствоваться дальше во многих направлениях. Например:

- можно добавить новые черты моделируемой реальной системе, добавляя блоки модели. Так, если клиент приходит в банк, то перед ним оказывается очередь из 25 человек, то он уходит из банка. Можно добавить это свойство модели «Bank Line», введя блок, который анализирует величину, получаемую с коннектора L (длина очереди) блока *Queue, FIFO*. Если указанная величина больше 25, некоторое количество транзактов (клиентов) покинет модель, не входя в блок *Queue, FIFO* (уйдут, не занимая очереди);
- можно изменять параметры модели в зависимости от времени суток. Например, если имитируется очередь в течение целого рабочего дня, а не одного часа, следует принять во внимание, что кассиры останавливают движение очереди на время перерыва;
- можно изменить скорость работы кассиров в течение дня, чтобы имитировать изменение их производительности;
- можно связать количество кассиров, обслуживающих клиентов, с количеством клиентов в очереди.

Рассмотренные примеры и приёмы позволяют перейти к практическому построению моделей в Extend и проведению имитационного моделирования реальных процессов.

Применение Extend для дискретного моделирования очередей, систем массового обслуживания, производственных процессов будет показано на примерах при выполнении лабораторных работ.

### ***Основные шаги создания Extend-модели***

1. Открытие библиотек блоков.
2. Создание новой модели и включение блоков в модель (добавление и соединение их с другими блоками).
3. Установка параметров моделирования и запуск модели.
4. Получение результатов моделирования (в численном и графическом виде) и интерпретация результатов.

Каждый из перечисленных шагов будет описан в модели для предприятия, которое ранее моделировалось в MS Excel. Это позволит сделать очевидными преимущества современной визуальной системы имитационного моделирования и прийти к тому же выводу, что и студенты многих университетов мира, изучающие имитационное моделирование с помощью этой системы:

***Extend is a Cool Tool !!!***

## 4. ЛАБОРАТОРНЫЙ ПРАКТИКУМ

### Лабораторная работа 1 Создание имитационной модели средствами Extend LT, проведение анализа и принятие управленческого решения на его основе

#### Исходные условия

**Объект моделирования:** одноканальная система массового обслуживания – предприятие по прокату автомобилей в аэропорту.

Рассматривается работа предприятия, которое предлагает услуги по прокату автомобилей для пассажиров, прибывающих в аэропорт.

Режим работы – круглосуточный.

Пассажиры прибывают в здание аэропорта на микроавтобусах, приезжающих каждые 5 минут.

На основании статистических данных делается предположение о вероятности того, что из прибывшей группы некоторое количество пассажиров пожелают взять напрокат автомобиль в рассматриваемой фирме. Вероятности задаются следующими значениями (рис. 4.1.1).

Кол-во клиентов	Вероятность
0	0,7
1	0,12
2	0,16
3	0,02

Рис. 4.1.1. Вероятности для предполагаемого количества клиентов

На оформление каждого договора проката автомобиля отводится ровно 5 минут.

Перед менеджером стоит задача – определить необходимое число стоек для оформления заказов проката.

Для принятия управленческого решения менеджеру потребуются имитационная модель деятельности предприятия.

Модель должна быть динамической и отражать случайные процессы, характерные для работы.

Для принятия решения требуется определить временные показатели работы — среднее и максимальное время ожидания клиентов в очереди на оформление; среднюю и максимальную длину очереди; коэффициент занятости сотрудников, оформляющих договоры.

Для набора достаточной статистики требуется промоделировать работу предприятия в течение рабочей недели (7 суток).

Как отмечалось выше, этапы логического и структурного моделирования выполняются в Extend одновременно с построением имитационной модели.

## **Построение модели**

### ***Открытие библиотек***

Перед тем как приступить к построению модели, требуется обеспечить возможность копирования блоков из библиотек Extend. Чтобы скопировать определенный блок Extend в модель, библиотека, в которой он хранится, должна быть открытой. Обычно для создания модели требуется открыть библиотеки Generic (Общая), Discrete Event (Дискретные события), Plotter (Графопостроитель). Чтобы открыть библиотеку Discrete Event, следует выполнить команду Library – Open Library (Библиотека — Открыть библиотеку), найти папку Libs, выбрать библиотеку Discrete Event (файл DE.lib) и щелкнуть по кнопке Open. Открытие библиотек Generic и Plotter выполняется в такой же последовательности.

После открытия нужных библиотек в меню Library появится список этих библиотек, а при установке курсора на имени библиотеки — откроется список типов объектов, содержащихся в данной библиотеке. Выбор типа объектов, в свою очередь, открывает список названий блоков этого типа. Отсюда выбирается блок, добавляемый к модели.

### ***Добавление блоков в модель и соединение их с другими блоками***

#### ***1. Задание общих временных параметров моделирования***

Для этого выбираем закладки Run/Simulation Setup. В открывшемся окне в строке «End simulation at time» ставим значение 10080 (5 недель = 10080 минутных интервалов). В строке «Global time units» выбираем Minutes (рис. 4.1.2).



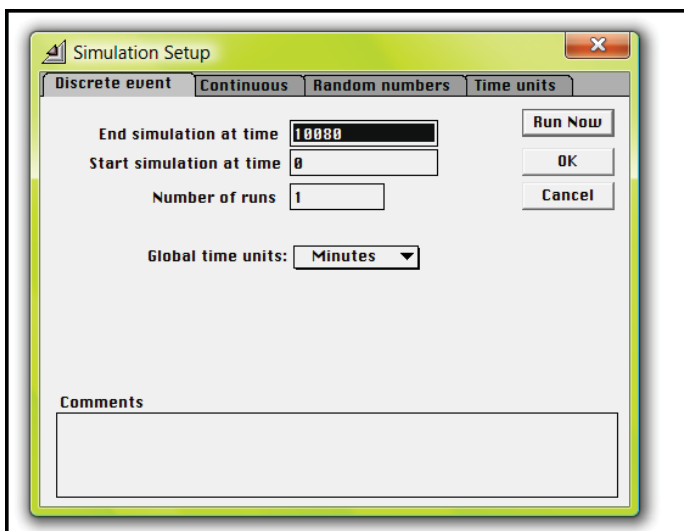


Рис. 4.1.2. Окно Simulation Setup для задания временных параметров моделирования

## 2. Создание часов модели

Для этого выбирается блок Executive из библиотеки DE.LIX :  
Library – DE.LIX – Executive.

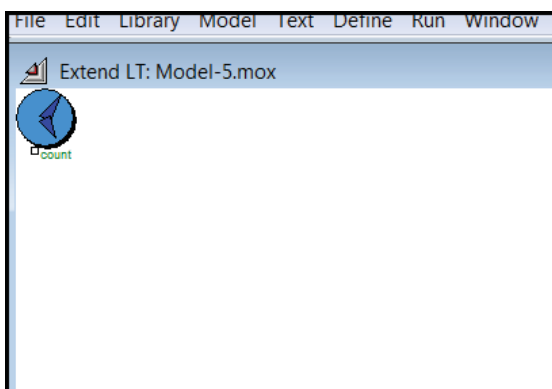


Рис. 4.1.3. Блок Executive – Диспетчер – тактовый датчик модели

Данный блок должен находиться левее всех других блоков, так как иначе невозможно будет запустить моделирование.

### 3. Создание генератора транзактов

Для этого выбирается блок *Generator* из библиотеки DE.LIX :  
Library – DE.LIX – Generators – *Generator*.

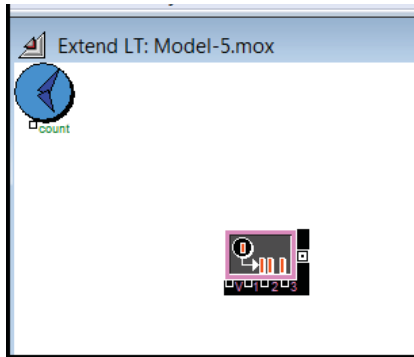


Рис. 4.1.4. Блок *Generator* – генератор транзактов модели

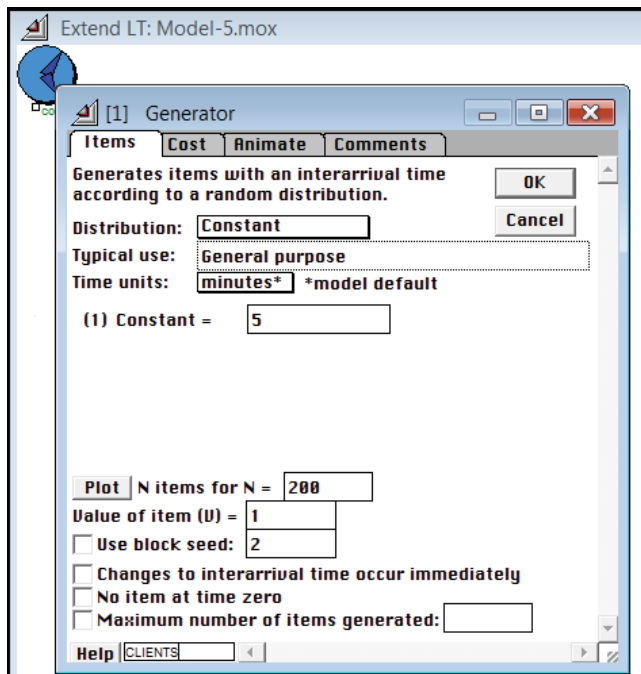


Рис. 4.1.5. Диалоговое окно блока *Generator*

Двойным щелчком на появившемся блоке открывается диалоговое окно. В разделе Items (Транзакты) в строке «Distribution» (Распределение) выбирается Constant.

В строке «Constant =>» ставится значение 5, чем обеспечивается генерация транзактов каждые 5 минут. В строке рядом с «Help» пишется имя блока CLIENTS.

#### 4. Задание случайного числа клиентов

Для этого выбирается блок Input Random Number из библиотеки GENERIC.LIX:

Library – GENERIC.LIX – Inputs/Outputs – Input Random Number. Появившийся блок – генератор случайных чисел.

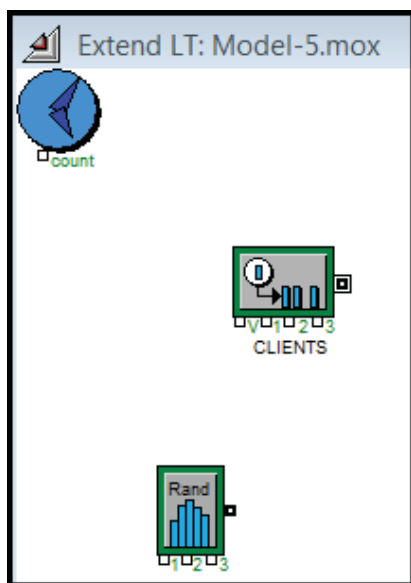


Рис. 4.1.6. Дополнение блока *Input Random Number*

Открывается его диалоговое окно. В разделе Distributions в строке «Distribution» выбирается Empirical table (Эмпирическая таблица). В открывшуюся таблицу заносятся заданные значения вероятностей появления указанных количеств клиентов. В строке рядом с «Help» пишется имя блока CLIENTS NUMBER.

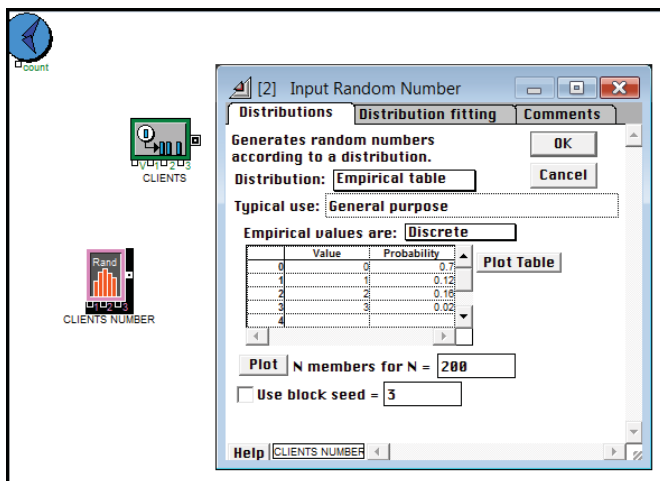


Рис. 4.1.7. Диалоговое окно блока *Input Random Number*

### 5. Соединение блоков

Блоки в Extend связываются с помощью коннекторов и линий соединения. Основные коннекторы блоков служат для входа и выхода транзактов. Информационные коннекторы служат для приема и передачи данных. Для соединения блоков линией связи достаточно установить курсор мыши на один коннектор, и, когда курсор примет вид «фломастера», нажав и не отпуская левую клавишу мыши, провести линию к выбранному коннектору другого блока. Линии могут быть прямыми, а могут быть ломаными (для этого клавиша мыши отпускается, а затем вновь нажимается и проводится следующий отрезок ломаной линии).

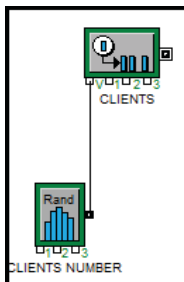


Рис. 4.1.8. Соединение блока *Input Random Number* и блока *Generator* линией данных

Соединим выход блока CLIENTS NUMBER и коннектор V блока CLIENTS. Коннектор V (value) получает данные, задающие количество транзактов, порождаемых блоком Generator.

Тем самым обеспечивается задание блоку-генератору количества выдаваемых транзактов при каждой генерации.

#### 6. Создание блока моделирования очереди

Для моделирования очереди выбирается блок Queue, FIFO из библиотеки DE.LIX:

Library – DE.LIX – Queues – Queue, FIFO.

Очередь типа FIFO предполагает, что клиент, приходящий первым, первым и обслуживается (first in – first out).

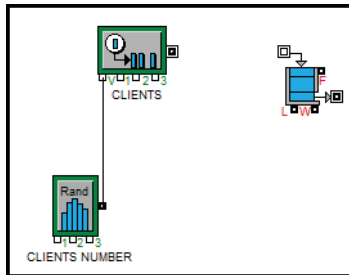


Рис. 4.1.9. Дополнение блока *Queue, FIFO*

Открывается диалоговое окно блока. В строке рядом с «Help» пишется имя блока QUEUE (Очередь).

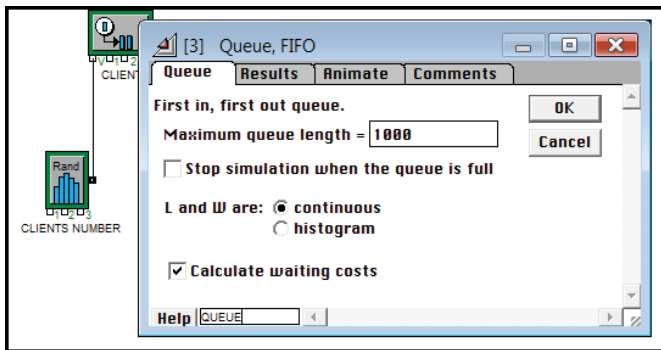


Рис. 4.1.10. Диалоговое окно блока *Queue, FIFO*

Параметр Maximum queue length (Максимальная длина очереди) оставляется неизменным, то есть очередь практически не ограничивается по длине.

### 7. Соединение блоков

Соединяются блок CLIENTS и блок QUEUE.

Соединение, по которому будут перемещаться транзакты, Extend изображает двойной линией.

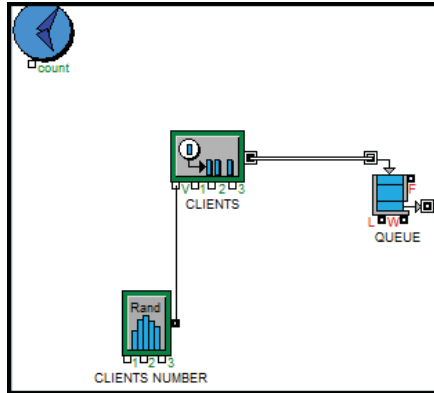


Рис. 4.1.11. Соединение блока *Generator* и блока *Queue, FIFO* линией транзактов

### 8. Создание блока обслуживания

Для моделирования места обслуживания выбирается блок *Activity, Multiple* из библиотеки DE.LIX:

Library – DE.LIX – Activities – Activity, Multiple.

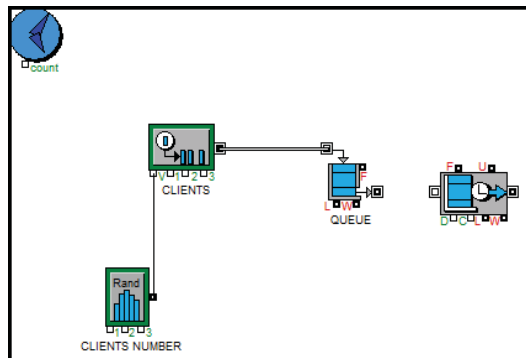


Рис. 4.1.12. Дополнение блока *Activity, Multiple*

Открывается диалоговое окно блока. В разделе Activity в строке «Delay» ставится значение 5 (время, затрачиваемое на оформление), в строке «Maximum number in activity» ставим значение 1 (количество мест обслуживания = 1). В строке рядом с «Help» пишется имя блока DESK. В строке рядом с «Help» пишется имя блока DESK.

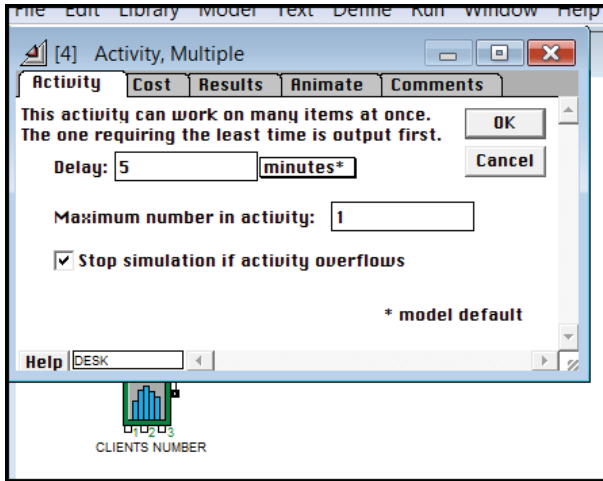


Рис. 4.1.13. Диалоговое окно блока *Activity, Multiple*

### 9. Соединение блоков

Соединяются блок QUEUE и блок DESK.

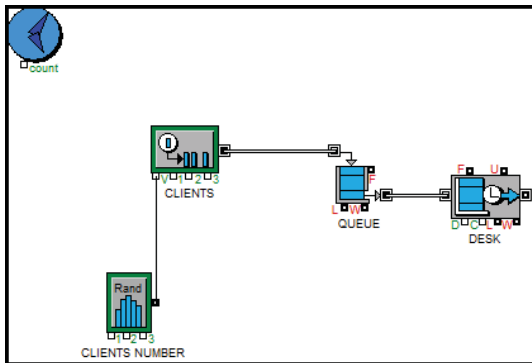


Рис. 4.1.14. Соединение блока *Queue, FIFO* и блока *Activity, Multiple* линией транзактов

## 10. Создание выхода для клиентов

Для моделирования выхода клиента на стоянку автомобилей выбирается блок Exit из библиотеки DE.LIX:

Library – DE.LIX – Routing – Exit.

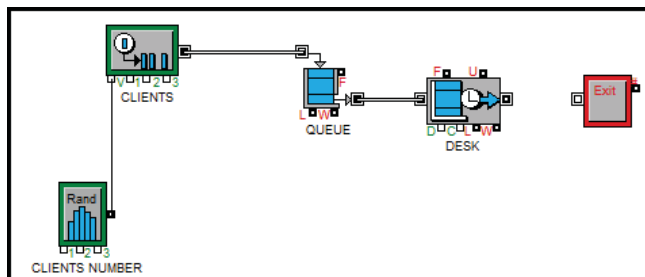


Рис. 4.1.15. Дополнение блока *Exit*

## 11. Соединение блоков

Соединяются блок DESK и блок EXIT.

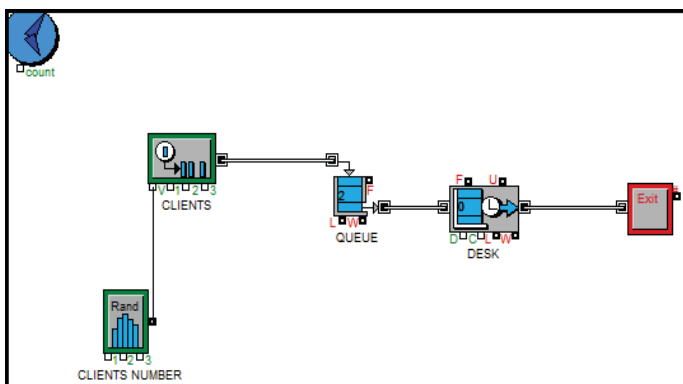


Рис. 4.1.16. Соединение блоков линией транзактов

## 12. Пробный запуск моделирования

Для запуска модели используется либо пиктограмма, либо закладка Run в главном меню Extend:

Run – Run Simulation.

Если необходимо отображение анимации, ставится  $\surd$  в строке Show Animation (Показать анимацию).



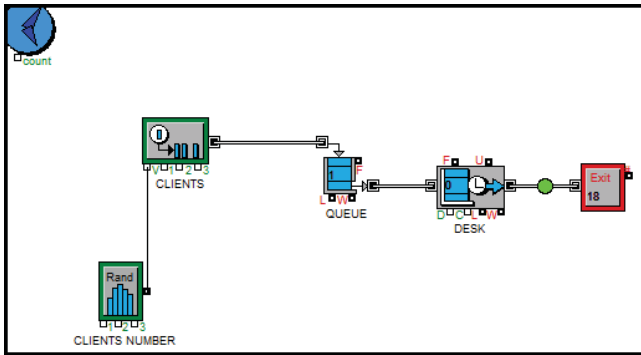


Рис. 4.1.17. Изображение транзакта как движущегося круга

Для изменения вида анимации транзактов (по умолчанию – Green circle (Зеленые кружки)) открывается диалоговое окно блока CLIENTS и в разделе *Animate* выбирается желаемый вид (например, вид Person).

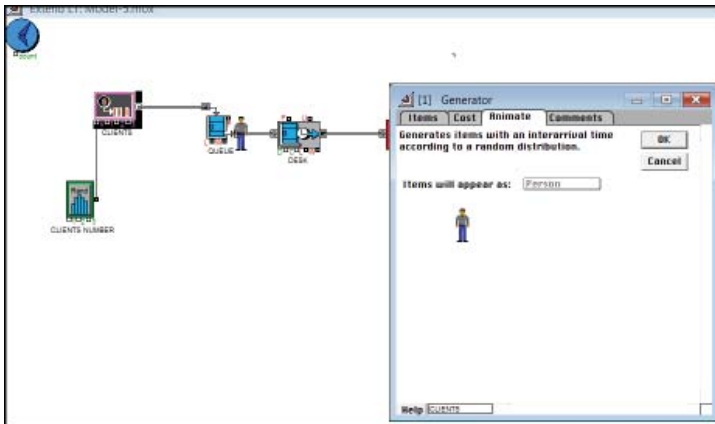


Рис. 4.1.18. Замена изображения транзакта на Person

### 13. Подключение блока графопостроителя *Plotter*

Для графического отображения результатов моделирования используется блок *Plotter, Discrete Event* из библиотеки **PLOTTER.LIX**.

**Library – PLOTTER.LIX – Plotter, Discrete Event.**

#### 14. Соединение блоков

Для отображения данных используются следующие коннекторы:

- коннектор W блока QUEUE (W означает waiting – ожидание) – текущее время ожидания транзакта в очереди;
- коннектор U блока DESK (U означает utilization – использование) – коэффициент использования места обслуживания.

Коннектор W блока QUEUE соединяется со вторым (красным) входом блока PLOTTER, а коннектор U блока DESK – с верхним (синим) входом блока PLOTTER.

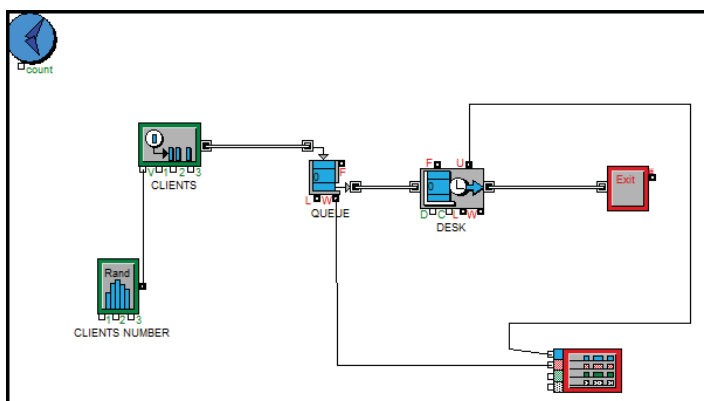


Рис. 4.1.19. Соединение блока Plotter и блоков Queue и Activity линиями данных

#### 15. Настройка блока графопостроителя Plotter

Открывается диалоговое окно, в котором имеется собственное меню настроек блока Plotter. Если щелкнуть по крайней левой пиктограмме меню настроек, откроется окно Tools. В окне Tools изменяются следующие параметры:

– в первой строке, соответствующей верхнему входу (синему цвету), задаются:

имя отображаемого параметра COEF;

тип кривой (толщина, штриховка, вид кривой);

ось ординат, по которой отображается данный элемент (справа,  $\uparrow$ );

– во второй строке задаются:

имя – WAITING;

тип кривой;

ось ординат, по которой отображается данный элемент (справа,  $\downarrow$ );

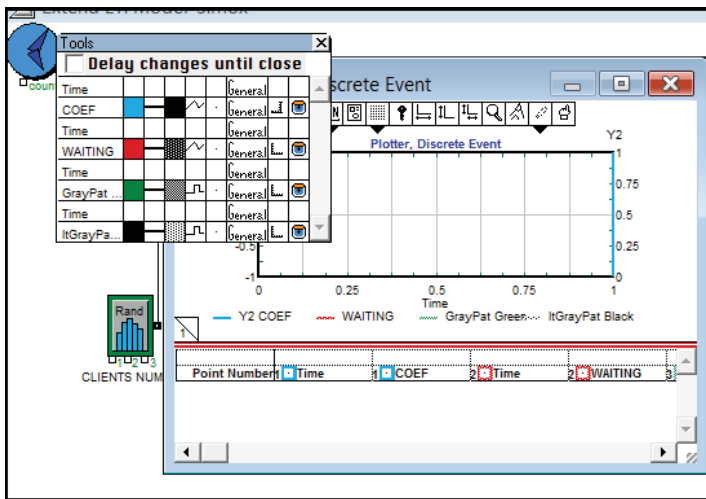


Рис. 4.1.20. Настройка параметров графопостроителя *Plotter*

## 16. Запуск моделирования

Для запуска модели используется либо пиктограмма, либо закладка Run в главном меню Extend:

Run – Run Simulation.

Так как анимация значительно замедляет процесс моделирования, то использовать её удобно лишь при отладке и демонстрации модели. Для получения итоговых результатов имитационного моделирования анимация отключается.

## 17. Построение графиков моделирования

В окне Plotter, Discrete Event получаем график, построенный после завершения моделирования процесса (рис. 4.1.21).

На полученном графике наглядно представлены результаты моделирования.

Синий (непрерывный) график отображает изменение коэффициента занятости оформителя в процессе работы (значения  $U$  в процессе моделирования). График показывает, что коэффициент занятости стабилизируется около значения 0.5 (правая ось  $Y_2$ )

Красный (столбчатый) график отображает изменение времени ожидания в очереди (значения  $L$  в процессе моделирования). График показывает, что максимальное время ожидания достигает 25 минут (левая ось  $Y_1$ ).

В таблице представлены результаты моделирования для всего периода моделирования поминутно (для всех 10080 минут).

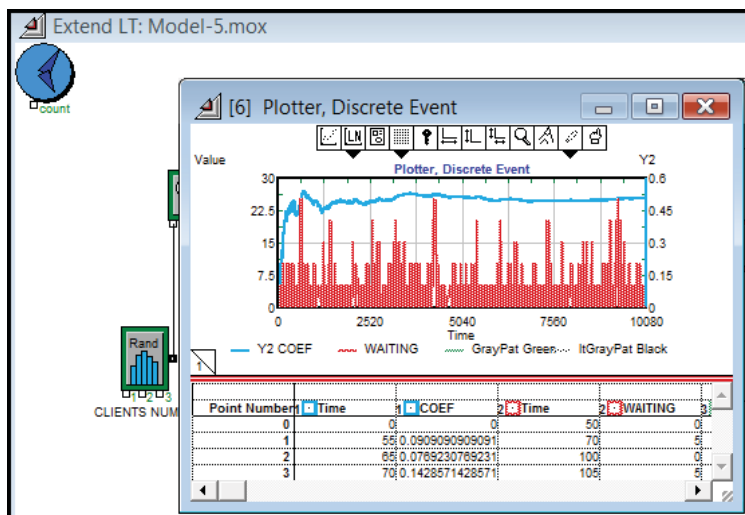


Рис. 4.1.21. Графическое представление результатов моделирования

### 18. Анализ полученных результатов

Для получения средних величин, требующихся при принятии решения, недостаточно графического представления результатов моделирования. Точно вычисленные средние значения временных параметров очереди и канала обслуживания могут быть найдены в диалоговых окнах блоков.

Для просмотра результатов моделирования очереди открывается диалоговое окно блока QUEUE, в котором выбирается раздел Results.

Из данного диалогового окна видно, что:

- средняя длина очереди равна 0,45;
- среднее время ожидания в очереди равно 4,5;
- максимальная длина очереди равна 6;
- максимальное время ожидания в очереди равно 25;
- общее количество людей, находившихся в очереди, равно 1015;
- процент загрузки равен 0,29.

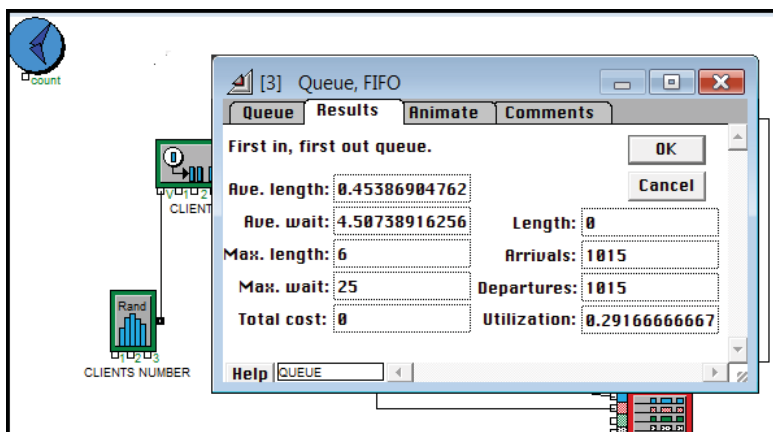


Рис. 4.1.22. Получение численных результатов моделирования очереди в диалоговом окне

Если открывается диалог блока DESK и выбирается раздел Results, то получаются следующие результаты (рис. 4.1.23):

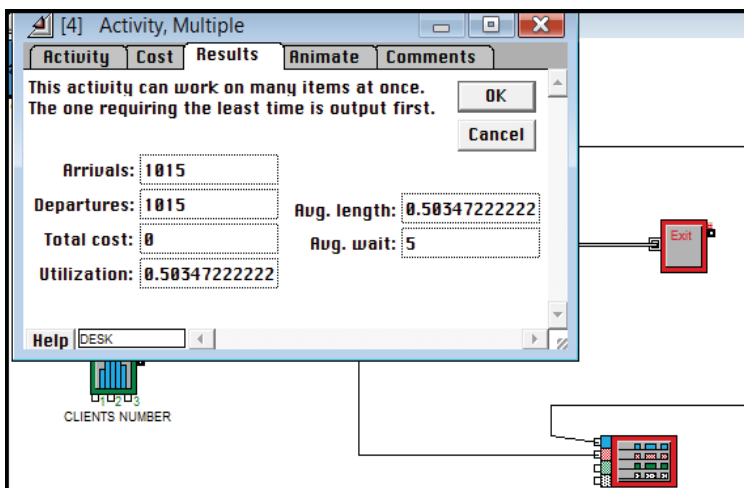


Рис. 4.1.23. Получение численных результатов моделирования канала обслуживания в диалоговом окне

Из данного диалогового окна видно, что:

- средняя длина очереди равна 0,5;
- среднее время обслуживания равно 5;

- процент загрузки стойки равен 0,5;
- количество обслуженных клиентов равно 1015.

Для того чтобы получить результат работы с двумя стойками обслуживания, достаточно в настройке блока *Activity, Multiple* задать параметр «Maximum number in activity» равным 2 (Максимальное количество мест обслуживания = 2).

Сравнение результатов моделирования при одной стойке обслуживания и при двух стойках показывает, что снижаются только параметры очереди и коэффициент занятости персонала. Это позволяет сделать вывод о нецелесообразности открытия второй стойки для оформления и принять соответствующее управленческое решение.

Результаты моделирования статистически совпадают с результатами, полученными с помощью модели в Excel. Но модель, построенная в Extend, более компактна и наглядна, в то же время более информативна. Этап структурного моделирования выполняется одновременно с функциональным моделированием. Модель значительно проще масштабируется и перенастраивается.

Данные получаются и в численном, и в графическом виде, что делает более простой их интерпретацию для принятия управленческого решения.

## **Лабораторная работа 2** **Создание имитационной модели средствами** **Extend LT, проведение статистического анализа,** **оптимизация параметров**

### **Исходные условия**

**Объект моделирования:** предприятие по прокату автомобилей в аэропорту, описанное в условиях лабораторной работы 1.

На основании статистических данных делается предположение о том, что микроавтобусы могут задерживаться и не всегда прибывают ровно через каждые 5 мин., а прибывают с интервалами от 3 до 7 мин. с «треугольным» распределением, с наиболее вероятным значением, равным 5 мин.

Кроме того, делается предположение, что служащему для оформления контракта на прокат автомобиля требуется не ровно

5 мин., а время, имеющее экспоненциальное распределение со средним значением 5 мин.

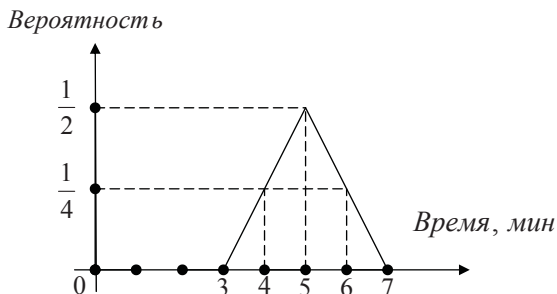


Рис. 4.2.1. «Треугольное» распределение случайной величины

Предполагается, что введение в модель дополнительных факторов изменчивости придаст ей большую реалистичность:

- приведет к большей вариативности итоговых значений времени ожидания и длины очереди;
- не скажется на средних значениях длины очереди и времени ожидания, *поскольку средние значения обоих распределений такие же, как в его первоначальной модели.*

Чтобы проверить эти предположения, требуется провести имитационное моделирование в новых условиях.

### Уточнение модели пункта проката автомобилей

1. Задание «треугольного» распределения выполняется в диалоговом окне блока **Generator**. Достаточно выбрать тип распределения *Triangle* из выпадающего меню, содержащего около 20 стандартных типов, предлагаемых Extend, и задать параметры распределения (рис. 4.2.2).

2. Моделирование случайного времени, затрачиваемого на оформление договора, выполняется новыми средствами, так как блок **Activity, Multiple** в отличие от блока **Generator** не имеет встроенного источника случайных величин. Но эту величину для длительности задержки (Delay) можно получить через коннектор D.

Для получения случайных значений времени добавляется блок **Input Random Number** (Ввод случайных чисел) (из подменю **Inputs/Outputs** библиотеки **Generic.lib**), генерирующий случайные чис-

ла с заданным распределением. Выводной коннектор этого блока можно соединить с входным коннектором D блока *Activity, Multiple*, который определяет время задержки транзакта в данном блоке. Таким образом, в этом блоке будет задано время обслуживания с экспоненциальным распределением (со средним временем задержки, как и прежде, 5 мин.).

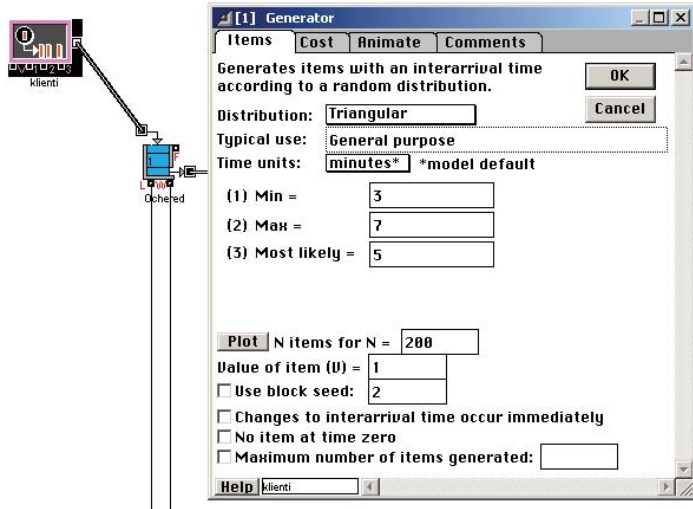


Рис. 4.2.2. Задание параметров «треугольного» распределения случайной величины в диалоге блока *Generator*

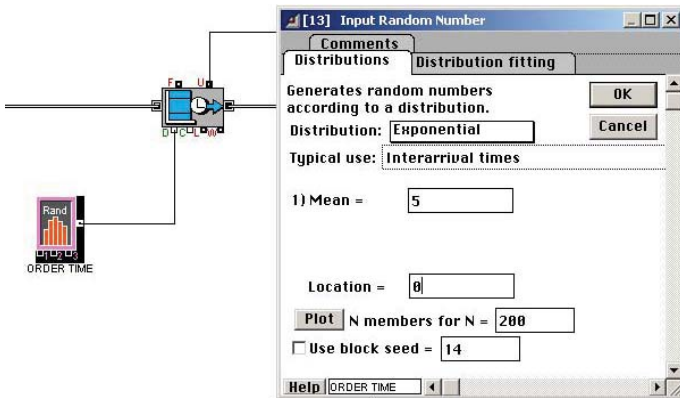


Рис. 4.2.3. Задание параметров экспоненциального распределения случайной величины в диалоге блока *Input Random Number*



3. Результаты имитационного моделирования показаны на рис. 4.2.4. Новый фактор изменчивости существенно увеличил максимальное время ожидания — до 54,4 мин. и немного — максимальную длину очереди — до 8 клиентов, увеличилось среднее значение времени ожидания (6,38 мин.) и среднее значение длины очереди (0,629 клиента) (рис. 4.2.5).

На основании полученных и многократно промоделированных результатов можно убедиться в **ошибочности первоначальных предположений**. Такое проникновение в суть исследуемых простых имитационных моделей важно для обобщений, особенно при анализе сложных систем.

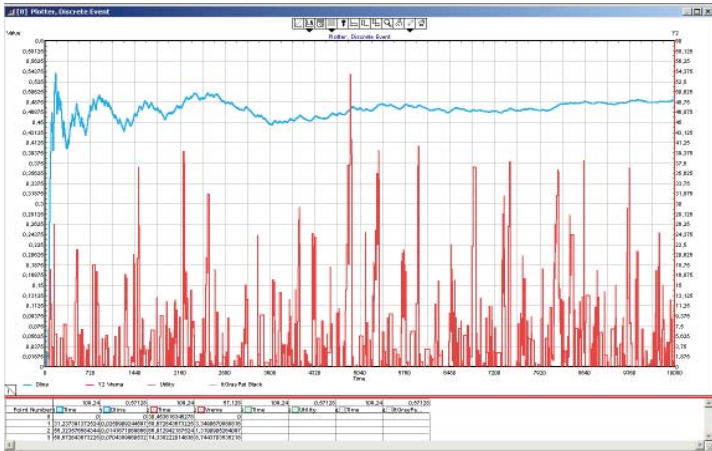


Рис. 4.2.4. Результаты моделирования с новыми случайными факторами

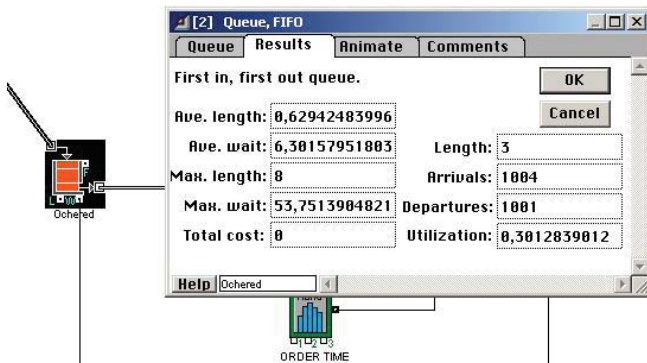


Рис. 4.2.5. Численные результаты моделирования очереди

**4. Сравнительный анализ.** В данной ситуации, как и ранее, возникает необходимость сравнения вариантов с одной и двумя стойками обслуживания.

Для этого двойным щелчком на блоке *Activity*, *Multiple* открывается диалог блока и изменяется значение параметра *Maximum Number in Activity* с 1 на 2, а затем запускается моделирование.



Рис. 4.2.6. Результаты моделирования с 2 стойками обслуживания

Результаты этой имитации показаны на рис. 4.2.6.

Новые значения:

- среднее время ожидания равно примерно 0,42 мин.;
- максимальное время ожидания – около 14 мин.;
- средняя длина очереди равна 0,04;
- максимальная длина очереди равна 4.

При этом коэффициент занятости также остается очень низким – 0,24.

**5. Внесение изменений в модель с целью повышения показателей работы моделируемого предприятия.** Предыдущие этапы моделирования показали, что изменчивость временных интервалов отрицательно воздействует на время ожидания в очереди, хотя их средние значения оставались такими же, как в предыдущей модели. Это подсказывает, что уменьшение изменчивости как в расписании

прибытия микроавтобусов, так и во времени обслуживания клиентов увеличит эффективность работы офиса.

Усматривая аналогию между различными системами массового обслуживания, можно предположить, что повышению эффективности будет способствовать использование части персонала по обслуживанию автомобилей для увеличения при необходимости количества стоек по оформлению проката. Это позволит сократить персонал до количества, которое необходимо для двух стоек в периоды низкого спроса, а в часы высокой нагрузки довести число оформителей до трех-четырех.

**Как преобразовать модель в Extend, чтобы она динамически изменяла свое поведение во время моделирования в соответствии с собственной статистикой производительности?**

Необходимо найти способ заставить Extend изменять значение в поле *Maximum Number in Activity* диалогового окна блока *Activity, Multiple* во время имитации на большее или меньшее число при увеличении и уменьшении длины очереди. Оказывается, в имитационной модели уже есть средства, необходимые для этого.

Двойным щелчком на блоке *Queue, FIFO* вызывается его диалог, в котором по кнопке *Help* открывается описание блока. В разделе, посвященном коннекторам, указано, что *коннектор L во время имитации передает значение длины очереди*.

Аналогичным образом устанавливается, что *коннектор C блока Activity, Multiple можно использовать для ввода значения, которое заменяет число в поле Maximum Number in Activity*.

Теперь требуется передать значения от выводного коннектора L блока *Queue, FIFO* к вводу коннектору C блока *Activity, Multiple* во время имитации модели.

В Extend есть блок для преобразования значений коннектора L в значения, приемлемые для коннектора C. Этот блок называется *Conversion Table* (Таблица преобразования), его можно найти в подменю *Math* (Математика) библиотеки *GENERIC.ltx*. На рис. 4.2.7 показана модель с добавленным блоком *Conversion Table*. На ввод этого блока подаются значения с коннектора L блока *Queue, FIFO*, а его выход передается на коннектор C блока *Activity, Multiple*. Таблица в диалоге блока *Conversion Table* задает правило, в соответствии с кото-

рым увеличивается количество стоек в пункте проката. Входные значения записываются в столбце «x in», а выходные— в столбце «y out». Предполагается, что очередь будет не более 7 человек (рис. 4.2.8).

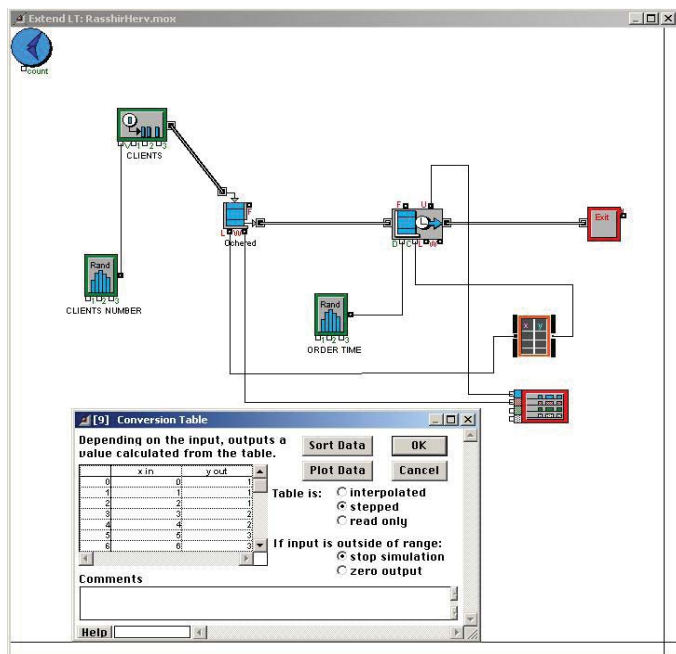


Рис. 4.2.7. Включение в модель блока *Conversion Table*, его соединения и диалог с таблицей преобразования «вход-выход»

	X in	Y out
0	0	1
1	1	1
2	2	1
3	3	2
4	4	2
5	5	3
6	6	3
7	7	4

Рис. 4.2.8. Таблица преобразования блока *Conversion Table*

**6. Имитационное моделирование новой версии модели** дает следующие результаты (рис. 4.2.9):

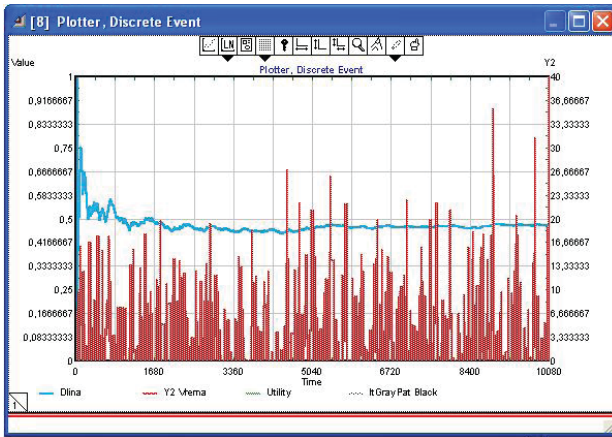
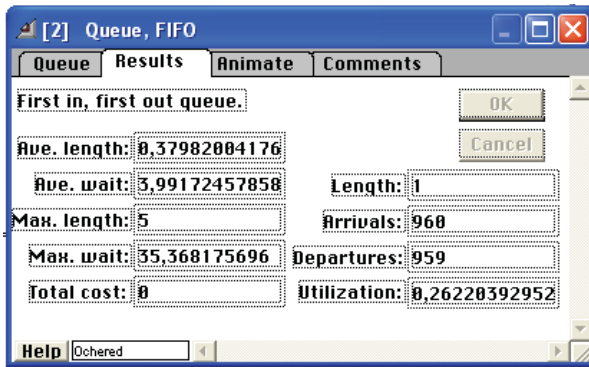


Рис. 4.2.9. Результаты моделирования с переменным числом стоек

a)



b)

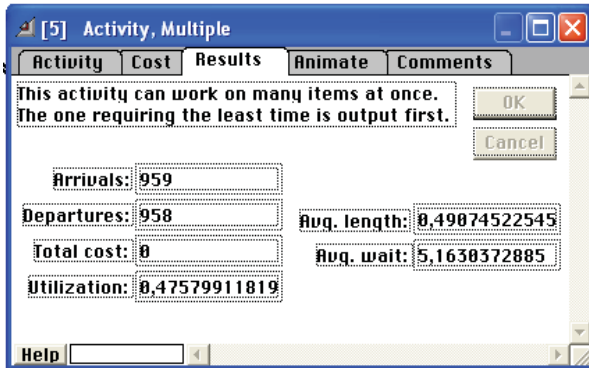


Рис. 4.2.10. Статистика блоков модели с переменным числом стоек

Результаты показывают, как плохо работает схема *динамического* использования персонала. Увеличение в два раза численности персонала во время прибытия максимального количества клиентов ухудшает статистику, отражающую время ожидания обслуживания.

Сравнение результатов, представленных на рис. 4.2.9 и 4.2.10, а, б, с результатами, представленными на рис. 4.2.6 (использование 2 стоек оформления), показывает, что среднее время ожидания для клиента увеличивается более чем в пять раз – приблизительно с 0,4 до 3,99 мин., а максимальное время ожидания в очереди возрастает почти до 35 мин. Это ухудшение связано с ростом средней длины очереди – с 0,04 до более 0,38.

Чем это вызвано?

Может быть, этот результат случаен и является следствием неблагоприятной последовательности случайных чисел при имитации модели? Проверить этот вывод можно, только многократно выполнив имитацию модели и сравнив полученные результаты, чтобы удостовериться в их статистической значимости.

**7. Статистическое исследование имитационной модели.** В библиотеке **STATS.lix** в **Extend** хранится блок *Queue Stats* (Статистики очереди), предназначенный для сведения в таблицу статистических данных об очереди, полученных в результате многократных запусков модели. Такое исследование позволяет систематично оценить влияние присущих модели случайностей на показатели очереди. На рис. 4.2.11 показан блок *Queue Stats*, который можно поместить в любом месте окна модели, а также диалоговое окно блока.

Для сведения в таблицу результатов многократного имитационного моделирования необходимо установить опции *Append new updates* (Добавить новые данные) и *Update at end of simulation* (Обновлять в конце имитации).

Затем выполняются команды:

*Run – Simulation Setup* (Установки имитации) – *Number of runs* (Количество испытаний) – установить 20 – ОК.

**Extend** выполнит 20 независимых имитаций модели. Результаты будут сведены в единую таблицу блока *Queue Stats*. В каждой строке таблицы будут представлены статистические данные об очереди, полученные при отдельной имитации.

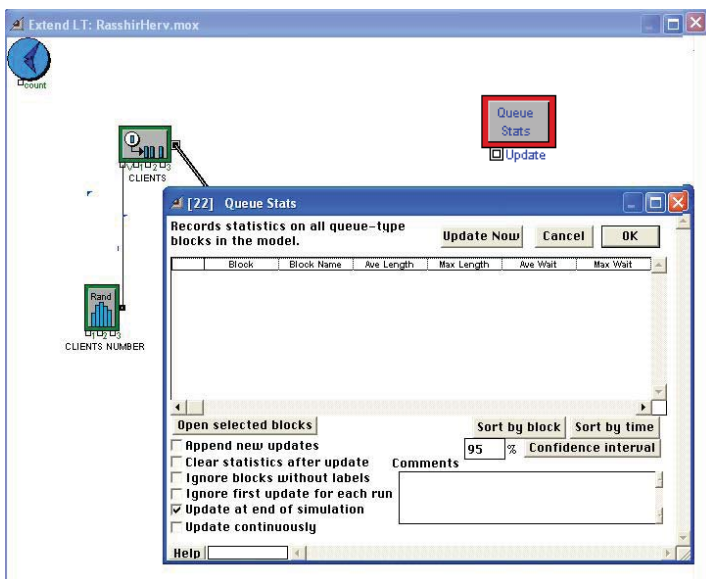


Рис. 4.2.11. Блок *Queue Stats* и его диалог

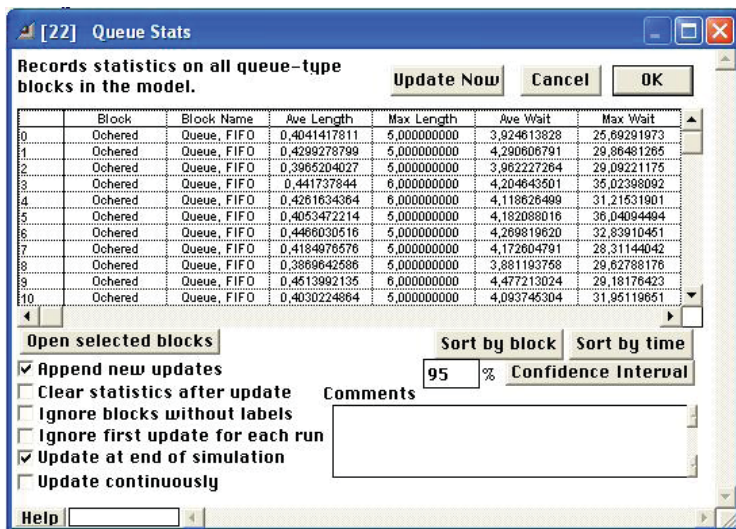


Рис. 4.2.12. Блок *Queue Stats* и его диалог

По данным таблицы видно, что они носят статистически устойчивый характер. Для статистической оценки значимости полученных значений блок *Queue Stats* позволяет вычислить до-



верительные интервалы для рассматриваемых оценок с уровнем значимости 95% (рис. 4.2.13).

Для этого достаточно нажать кнопку **Confidence Interval** (Доверительный интервал) в диалоге блока *Queue Stats* (при этом кнопка преобразуется в **Display Data**).

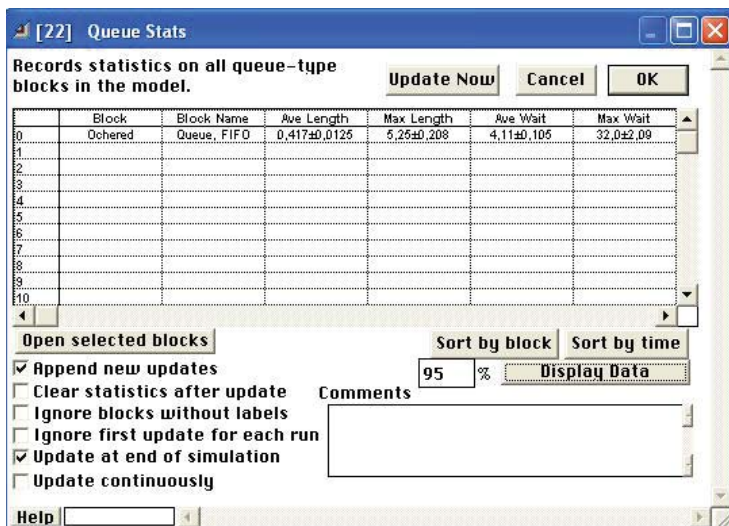


Рис. 4.2.13. 95-процентные доверительные интервалы

В результате 20-кратного моделирования получены 95-процентные доверительные интервалы для следующих величин:

Average Length (Средняя длина очереди)	- $0,417 \pm 0,0125$
Max Length (Максимальная длина очереди)	- $5,25 \pm 0,208$
Average Wait (Среднее время ожидания)	- $4,11 \pm 0,105$
Max Wait (Максимальное время ожидания)	- $32,0 \pm 2,09$

**8. Интерпретация результатов и принятие управленческого решения.** Моделирование является мощным инструментом, дополняющим и проверяющим интуицию, которая часто ошибается в предсказании поведения сложных систем в условиях перегрузки.

Результаты проведенного моделирования показывают, что смоделированный гибкий метод использования персонала не даёт положительных результатов, так как приводит к увеличению среднего и максимального времени ожидания обслуживания клиентами по сравнению с использованием двух постоянных служащих.



Таким образом, на основе статистических данных, полученных в результате имитационного моделирования деятельности предприятия, принимается решение отказаться от предложенных изменений.

### Лабораторная работа 3

#### Совершенствование имитационной модели предприятия в Extend LT, создание иерархического блока, проведение анализа и принятие управленческого решения на его основе

##### Исходные условия

**Объект моделирования:** предприятие по прокату автомобилей в аэропорту, описанное в условиях лабораторной работы 1 и уточненное в лабораторной работе 2.

Выбран вариант работы предприятия с 2 стойками оформления договоров, обеспечивающий наиболее быстрое обслуживание.

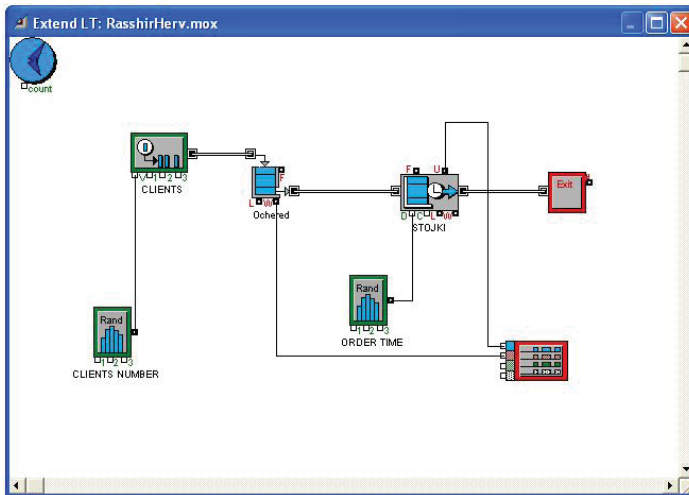


Рис. 4.3.1. Принятая модель предприятия по прокату автомобилей

Задача состоит в том, чтобы обеспечить достаточное количество автомобилей, считая модель работы офиса предприятия окончательно сформированной на данном этапе.

Предполагается только два типа заявок клиентов:

- 1) небольшие автомобили экономичного класса;
- 2) большие и мощные автомобили представительского класса.

При этом считается, что 70% клиентов предпочитают экономичные автомобили, а остальные выбирают большие автомобили. В случае отсутствия на стоянке экономичных автомобилей клиентам предоставляются большие автомобили по цене экономичных.

Статистика показывает, что клиенты, арендующие экономичные автомобили, берут их на 1-2 дня, а срок аренды больших автомобилей составляет  $2 \pm 0,5$  дня. Время проката экономичных автомобилей имеет равномерное распределение, а больших автомобилей — нормальное распределение со стандартным отклонением, равным 0,5 дня.

Прогноз еженедельного максимального спроса на большие автомобили — 100, на экономичные автомобили — 140.

Требуется определить количество автомобилей каждого типа.

### Построение модели

1. Для упрощения визуального представления модели используется иерархический блок Extend. Это позволяет создавать модели, составленные из подмоделей, каждая из которых является логически связанной группой блоков. Таким способом можно создавать *иерархические блоки* (суперблоки), каждый из которых является подмоделью, составленной из множества блоков Extend.

Для создания иерархического блока выбирается группа связанных блоков и выполняется команда:

**Model – Make Selection Hierarchical**

**(Модель → Создать иерархию).**

Extend «спрячет» выбранные блоки в суперблок, который можно обозначить и присоединить, в свою очередь, к другим блокам или суперблокам, что в результате приводит к очень компактной модели. Например, для моделирования управления парком автомобилей можно объединить блоки, относящиеся к модели офиса, в виде отдельного суперблока.

Выделяются указанные блоки либо щелчком на каждом из них левой кнопкой мыши при нажатой клавише <Shift>, либо охватом их «рамкой» на поле модели.

- Для суперблока выделяются все блоки, кроме блоков Executive и Exit.
- Model → Make Selection Hierarchical.
- В открывшемся диалоговом окне задается имя нового иерархического блока (рис. 4.3.2) «RECEPTION».
- Нажать Make H-Block (Создать иерархический блок).

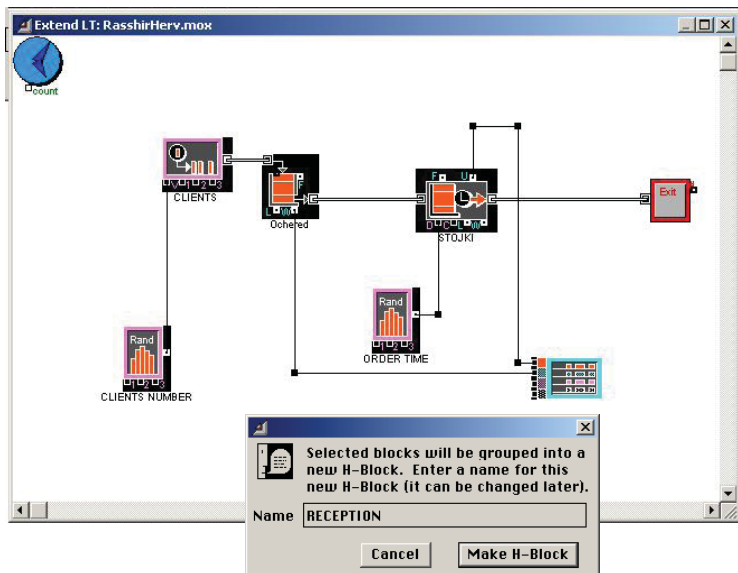


Рис. 4.3.2. Создание иерархического блока RECEPTION

В результате создания иерархического блока модель приобретет вид, представленный на рис. 4.3.3.

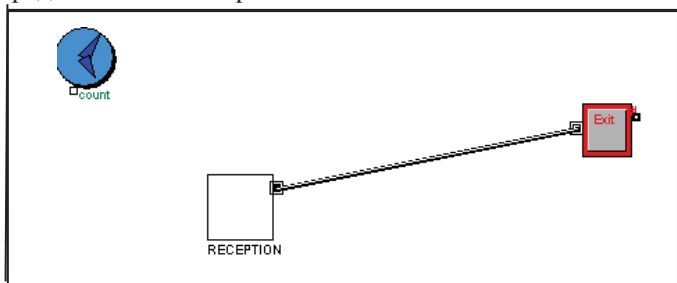


Рис. 4.3.3. Укрупненный вид модели

Структура суперблока может быть просмотрена с помощью команды **Model – Open Hierarchical Block Structure**.

- Открывается окно структуры блока (рис. 4.3.4), имеющее 4 поля:
- 1) поле графического обозначения иерархического блока;
  - 2) поле справочной информации об иерархическом блоке (on-line text help);
  - 3) поле описания коннекторов иерархического блока;
  - 4) поле подмодели.

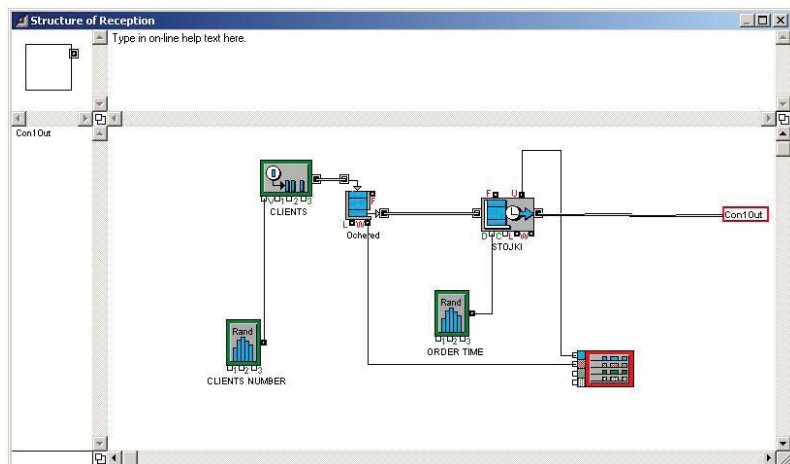


Рис. 4.3.4. Окно структуры иерархического блока

Двойным щелчком на значке суперблока также открывается окно со всеми входящими в него блоками.

2. К укрупненной модели на место блока Exit добавляются новые блоки так, чтобы транзакты, выходящие из подмодели RECEPTION, попадали в блок модели управления стоянкой автомобилей.

Чтобы смоделировать передачу клиентам автомобилей, требуется отличать клиентов, желающих арендовать экономичные автомобили, от желающих арендовать большие машины.

Удаляется блок Exit. Затем моделируется выход на автостоянку добавлением двух новых блоков, показанных на рис. 4.3.5.

Диалоговое окно блока Input Random Number (подменю Input/Output библиотеки GENERIC.lib) представлено на рис. 4.3.6. Для задания распределения случайной величины, принимающей значение 0 (моделирует требование экономичного автомобиля) с вероятностью 0,7 и значение 1 (моделирует требование большого авто-

мобиля) с вероятностью 0,3, используется распределение Empirical Table (Эмпирическая таблица).

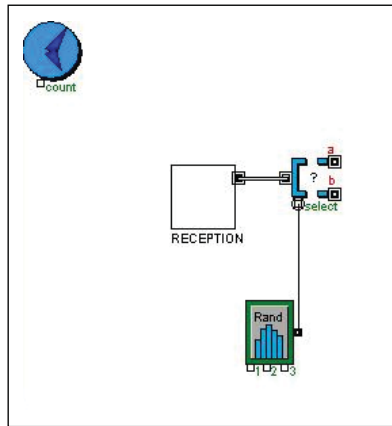


Рис. 4.3.5. Добавление блоков *Input Random Number* и *Select DE Output*

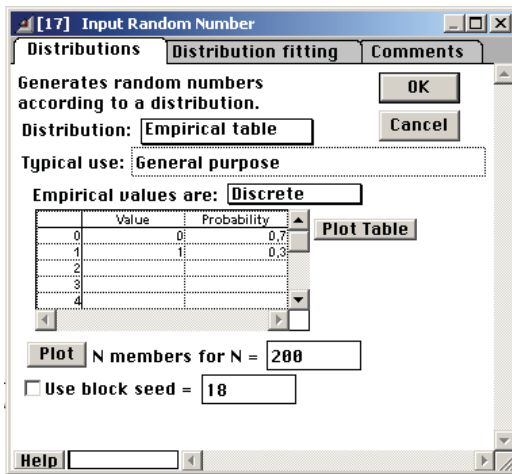


Рис. 4.3.6. Настройка распределения Empirical Table (Эмпирическая таблица) блока *Input Random Number*

Выход этого блока соединяется с вводным коннектором блока *Select DE Output* (Выбор выхода транзакта), находящегося в подменю Routing библиотеки DE.lnx. Этот блок разделяет поток транзактов (клиентов) и направляет их по одному из двух выходных коннекторов на основе значения, поступившего на его входной коннектор Select (Выбор).

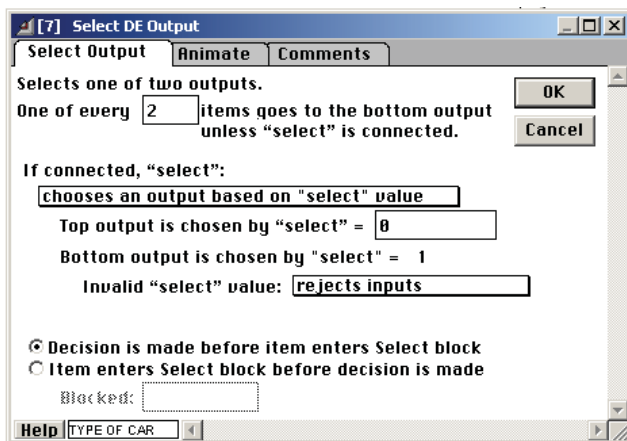


Рис. 4.3.7. Диалоговое окно блока *Select DE Output*

В диалоговом окне этого блока на рис. 4.3.7 видно, что входящий транзакт направляется к верхнему выходу (коннектор «а»), если число на коннекторе Select равно 0, а к нижнему выходу (коннектор «b»), если число на коннекторе Select равно 1. Таким образом, два новых блока направляют транзакт (клиента), выходящий из блока RECEPTION, к верхнему выходу блока *Select DE Output* (экономичный автомобиль) с вероятностью 0,7 и к нижнему (большой автомобиль) с вероятностью 0,3.

3. Передача автомобилей клиентам моделируется введением новых двух блоков.

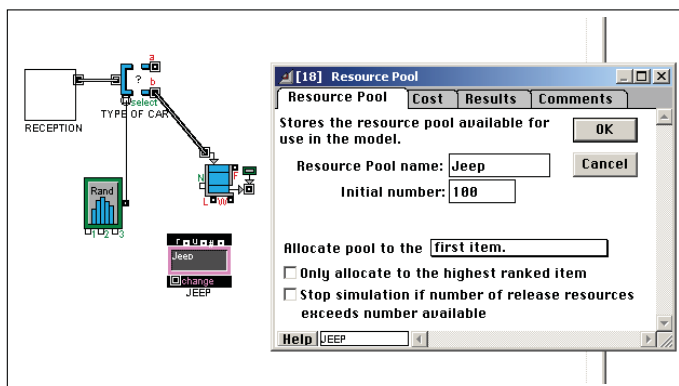


Рис. 4.3.8. Диалоговое окно блока *Resource Pool*

Блоки *Resource Pool* (Накопитель ресурсов) (из подменю **Resources** (Ресурсы) библиотеки **DE.lix**) и *Queue, Resource Pool* (Очередь, Накопитель ресурсов) (из подменю **Queues** (Очереди) библиотеки **DE.lix**) и их диалоговые окна показаны на рис. 4.3.8 и 4.3.9.

*Resource Pool* — это переменная, которая моделирует количество доступных элементов ограниченного ресурса (больших автомобилей).

В соответствии с прогнозом еженедельного максимального спроса на большие автомобили в диалоговом окне блока *Resource Pool* вводится число 100 (количество имеющихся в наличии больших автомобилей) в поле **Initial number** (Начальное значение).

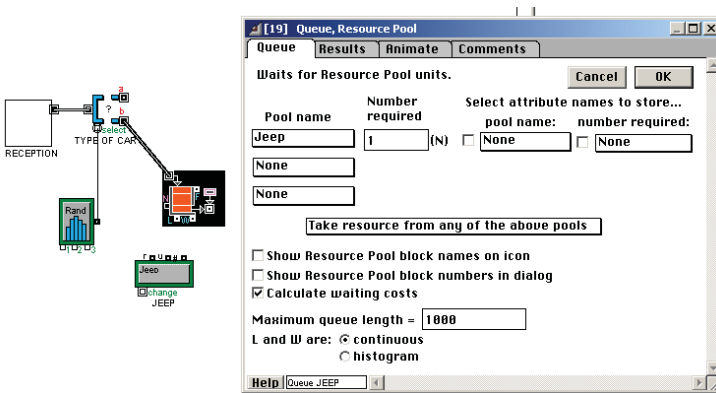


Рис. 4.3.9. Диалоговое окно блока *Queue, Resource Pool*

Блок *Queue, Resource Pool* задерживает транзакты (клиентов, арендующих большие автомобили) до тех пор, пока не будет доступно определенное количество элементов (в данном случае 1) в блоке *Resource Pool* по имени JEEP, а затем выводит элемент и уменьшает на 1 число элементов в блоке, моделируя передачу автомобиля клиенту. При заполнении поля **Pool name** (Имя хранилища) имя блока *Resource Pool*, данное ему в поле **Resource Pool name** (а не в поле рядом с **Help**), выбирается из списка, открывающегося при щелчке по полю **Pool name**. В поле **Number required** задается количество единиц хранимого ресурса, выделяемых каждому транзакту (1 автомобиль каждому клиенту).

**Внимание! Attention!** Информация блока *Resource Pool* используется как глобально доступная переменная, поэтому он не имеет соединений с другими блоками.

4. Моделирование интервала времени проката автомобиля производится блоком *Activity, Multiple* (подменю Activities библиотеки DE.lix).

Блок осуществляет задержку, время которой определяет новый блок *Input Random Number*. В диалоге этого блока задается нормальное распределение случайной задержки, среднее значение которой равно 2 дням (2880 мин.), а стандартное отклонение составляет 0,5 дня (720 мин.), как показано на рис. 4.3.10.

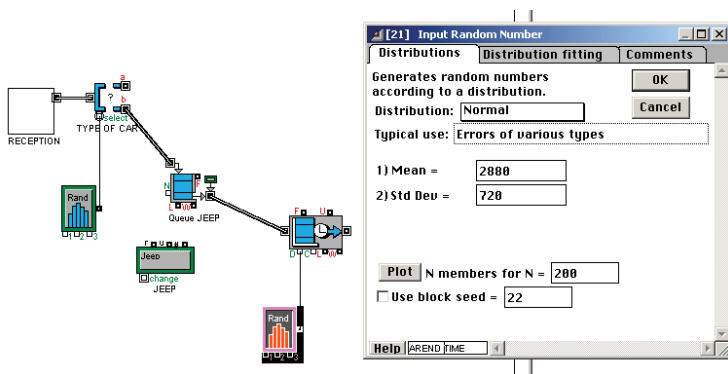


Рис. 4.3.10. Диалоговое окно блока *Input Random Number*

Количество одновременно используемых каналов обслуживания (автомобилей) блока *Activity, Multiple* задается значением 1000. Это гарантирует, что ни один транзакт не будет задержан при входе в этот блок.

5. После возврата автомобиль вновь становится доступным для последующей аренды. Возвращение автомобиля на автостоянку по истечении времени проката (случайной задержки) моделируется с помощью блока *Release Resource Pool* (Освобождение элементов ресурса) из подменю Resources библиотеки DE.lix. Блок *Release Resource Pool* и его диалоговое окно показаны на рис. 4.3.11.



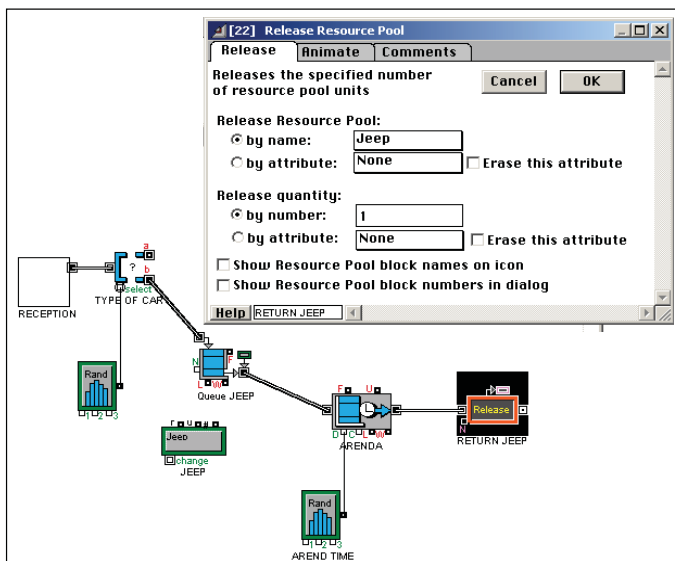


Рис. 4.3.11. Диалоговое окно блока *Release Resource Pool*

Освобождаемый ресурс определяется по имени (поле **by name**). Имя ресурса JEEP выбирается из открывающегося меню при щелчке по полю **by name**. Количество возвращаемых элементов задается в поле **by number**. В модели блок именуется RETURN JEEP.

6. Моделирование получения и возврата клиентами экономических автомобилей выполняется аналогично. Для экономических автомобилей добавляется другой блок *Resource Pool* под названием Micro, где устанавливается их начальное количество — 140. Также добавлен блок *Queue, Resource Pool*, моделирующий получение клиентом автомобиля (если экономические автомобили имеются в наличии). Далее создаются блоки *Activity, Multiple* и *Input Random Number*, моделирующие прокат в течение времени, распределенного равномерно от 1 до 2 дней (распределение Integer, Uniform, от 1440 до 2880 мин.). Наконец, добавляется блок *Release Resource Pool*, моделирующий возврат автомобилей для повторного проката (рис. 4.3.11).

Вывод транзактов обоих типов из модели выполняется блоком *Exit (4)* (подменю Routing библиотеки DE.lix) (рис. 4.3.12).

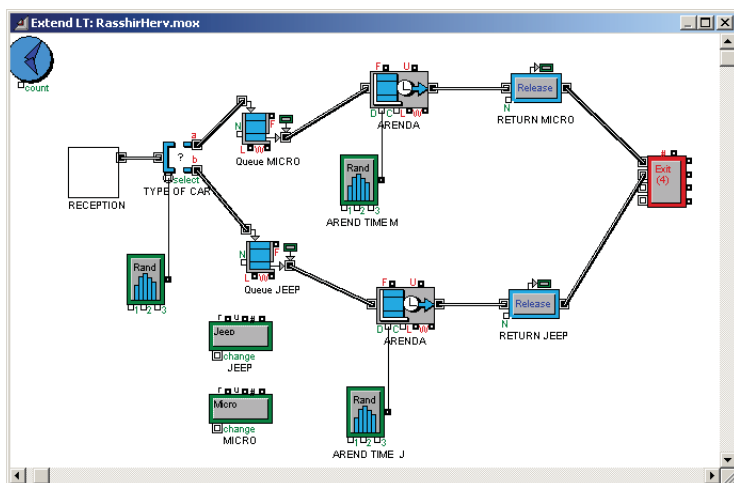


Рис. 4.3.12. Модель проката двух типов автомобилей

7. Моделируется предоставление клиентам больших автомобилей по цене экономичных в случае отсутствия экономичных автомобилей.

Такие действия моделируются в диалоговом окне блока *Queue, Resource Pool MICRO*, показанном на рис. 4.3.13.

Оба накопителя ресурсов поименованы (MICRO и JEEP), и Extend начинает поиск ресурсов в том накопителе ресурсов, имя которого стоит первым в списке Pool name (Имя накопителя) диалога блока *Queue, Resource Pool MICRO*. В данном случае Extend будет пытаться сначала взять автомобиль из ресурсов MICRO. Если этот накопитель ресурсов пуст, рассматривается накопитель ресурсов JEEP, и будет взят большой автомобиль в случае, если он доступен. В противном случае транзакт будет удерживаться до тех пор, пока один автомобиль любого типа не станет доступным.

**Внимание! Attention!** В результате такого подхода возникает «проблема возврата». Модели должно быть известно, какого типа автомобиль арендован, для того чтобы при возврате из проката вернуть его в соответствующий накопитель ресурсов.

Диалоговое окно на рис. 4.3.13 позволяет создавать атрибуты транзактов. Атрибут придаётся транзакту (клиенту) и следует с ним через всю модель.

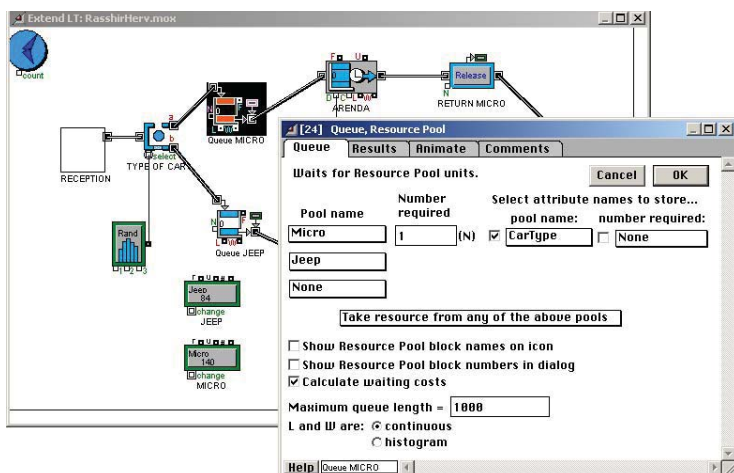


Рис. 4.3.13. Настройка блока Queue MICRO для использования ресурсов пула JEEP в случае пустого пула MICRO

В данном случае атрибут CarType (ТипАвто) был создан с помощью щелчка на кнопке **pool name** (имя накопителя), выбора пункта **New attribute** и ввода имени атрибута в открывшемся диалоговом окне (рис. 4.3.14).

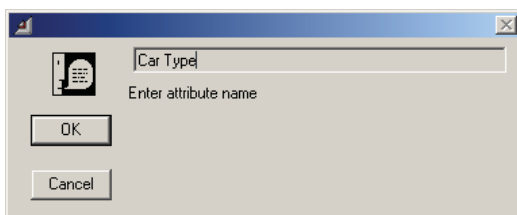


Рис. 4.3.14. Атрибут CarType

Во время имитации модели транзакт, проходящий через блок Queue MICRO (клиент, желающий арендовать экономичный автомобиль), использует ресурс одного из двух возможных накопителей. При этом в атрибут CarType будет записано имя того накопителя ресурсов, элемент которого действительно используется данным транзактом (из которого берется автомобиль).

Как показано на рис. 4.3.15 в диалоговом окне блока **Release Resource Pool**, освобождаемый ресурс (возвращенный из проката автомобиль) передается в один из двух накопителей в зависимости

от значения атрибута CarType, которое имеет пришедший транзакт. На это указывает «радиоточка» у поля **by attribute**.

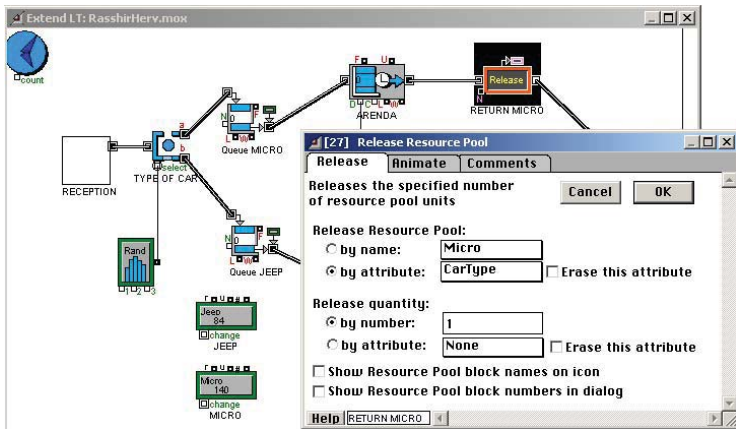


Рис. 4.3.15. Настройка места возврата ресурса в зависимости от значения атрибута CarType

Для задания атрибута, по которому определяется место возврата элемента ресурса, курсор устанавливается на поле **by attribute**. При щелчке в диалоге появляется список атрибутов транзакта, из которого выбирается атрибут CarType.

8. Для сбора статистики при многократном запуске имитационной модели в неё добавляются блоки *Queue Stats* и *Activity Stats* (оба из библиотеки *STATS.ltx*), а также *Plotter*, *Discrete Event* (из библиотеки *PLOTTER.ltx*) (рис. 4.3.16).

Блок *Queue Stats* собирает статистические данные для блоков *Queue*, а блок *Activity Stats* делает это для блоков *Activity*. Блоки *Activity Stats* и *Queue Stats* можно размещать в любом месте окна модели.

Для накопления результатов на протяжении многократных имитаций модели в диалогах блоков *Activity Stats* и *Queue Stats* задаются опции **Append new updates** и **Update at end of simulation** (как было сделано ранее в окне блока *Queue Stats* на рис. 4.2.12).

Для удобства интерпретации результатов, содержащихся в этих блоках, всем блокам *Activity*, *Multiple* присваиваются имена (в поле диалога рядом с кнопкой Help). Блоки *Activity*, *Multiple*, моделирующие прокат экономичных и больших автомобилей, названы соответственно AREnda MICRO и AREnda JEEP.

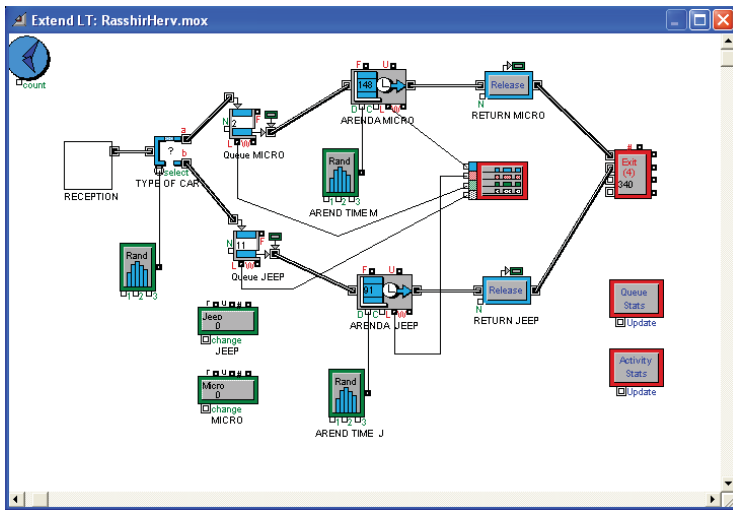


Рис. 4.3.16. Полная модель предприятия по прокату автомобилей

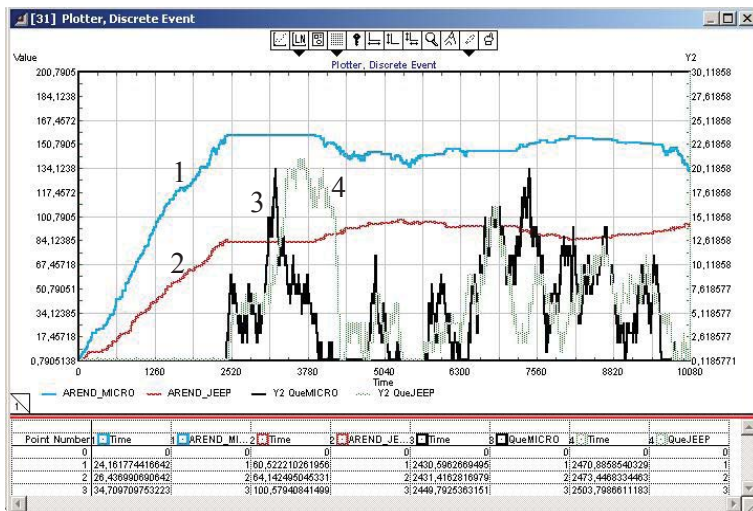
На 1-й и 2-й коннекторы блока *Plotter, Discrete Event* заводятся данные с коннекторов L блоков *Activity, Multiple*, передающие количество транзактов, находящихся в обслуживании (нажав HELP, в разделе Connectors можно прочесть «L: Number of items in the block» – Количество транзактов в блоке). Эти числа моделируют количество автомобилей каждого вида, находящихся в прокате.

На 3-й и 4-й коннекторы блока *Plotter, Discrete Event* заводятся данные с коннекторов L блоков *Queue, Resource Pool*, передающие количество транзактов, находящихся в очереди (нажав HELP, в разделе Connectors можно прочесть «L: Number of items in the queue at the current time» – Количество транзактов в очереди в текущий момент). Эти числа моделируют количество клиентов, ожидающих автомобиль каждого вида.

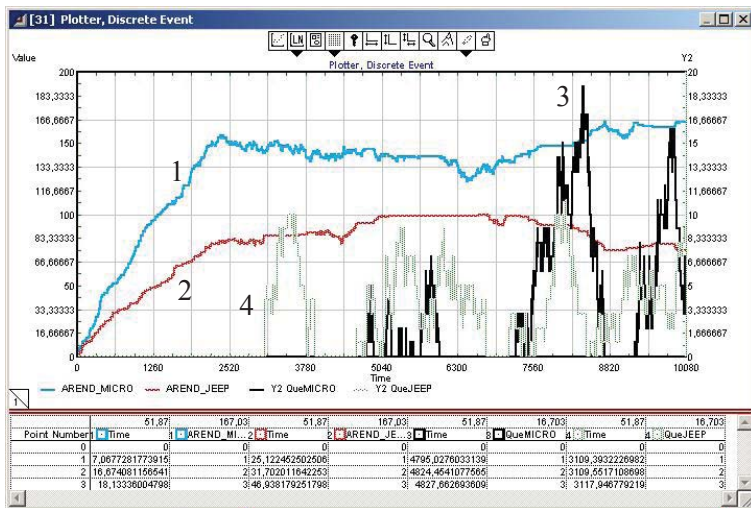
На рис. 4.3.17 представлены графики изменения наблюдаемых переменных, полученные в результате одной имитации модели, при заданных начальных количествах ресурсов, равных 140 экономичным и 100 большим автомобилям.

Графики 1 и 2 используют левую вертикальную шкалу значений и показывают количество автомобилей каждого типа, арендуемых в течение недели (верхняя линия — количество экономичных автомобилей, в число которых иногда включаются большие автомобили).

Графики 3 и 4 используют правую вертикальную шкалу значений и показывают количество клиентов, которые не могут сразу получить в аренду автомобиль из-за их отсутствия.



a)



b)

Рис. 4.3.17. Два варианта результатов однократного запуска полной модели предприятия

Результаты моделирования показывают, что заданного начального количества автомобилей недостаточно для удовлетворения всех требований клиентов.

Из графиков видно, что очередь за экономичными автомобилями вырастает до 20–30 человек (график 3), а очередь за джипами – до 10–20 человек (график 4). Горизонтальные участки на графиках 1 и 2 соответствуют тем периодам, когда все автомобили находятся в прокате. Как видно на рис. 4.3.17,а, б, в эти периоды происходит резкий рост графиков 3 и 4.

Точные значения параметров для очереди за экономичными автомобилями также показывают неудовлетворительную работу предприятия (рис. 4.3.18).

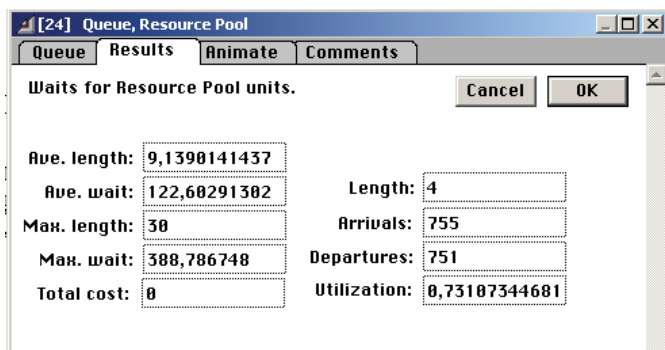


Рис. 4.3.18. Результаты моделирования очереди Queue MICRO

Средняя длина очереди – 9,1398

Максимальная длина очереди – 30

Среднее время ожидания – 122,68

Максимальное время ожидания – 388,79.

Аналогичная картина наблюдается и для показателей очереди за джипами.

9. Прежде чем принимать решение по результатам моделирования, следует убедиться, что выявленные взаимосвязи не случайны и имеют статистически устойчивый характер. Для статистической оценки результатов используются блоки *Queue Stats* и *Activity Stats*.

Для многократного моделирования, в процессе которого будут собираться статистические данные о наблюдаемых параметрах, задается количество запусков модели, равное 20:

Run – Simulation Setup – Number of runs – 20 .

После завершения всех 20 имитаций, каждая из которых сопровождается построением своих графиков на экране, результаты моделирования и их статистическая оценка сохраняются в блоках *Queue Stats* и *Activity Stats*.

Для просмотра этих результатов открываются диалоги блоков. Для вычисления границ доверительных интервалов достаточно задать их уровень надежности (обычно 95%) и нажать кнопку «Confidence interval».

На рис. 4.3.19 представлены диалоги блоков *Queue Stats* и *Activity Stats* с 95-процентными доверительными интервалами, построенные после 20 имитаций модели.

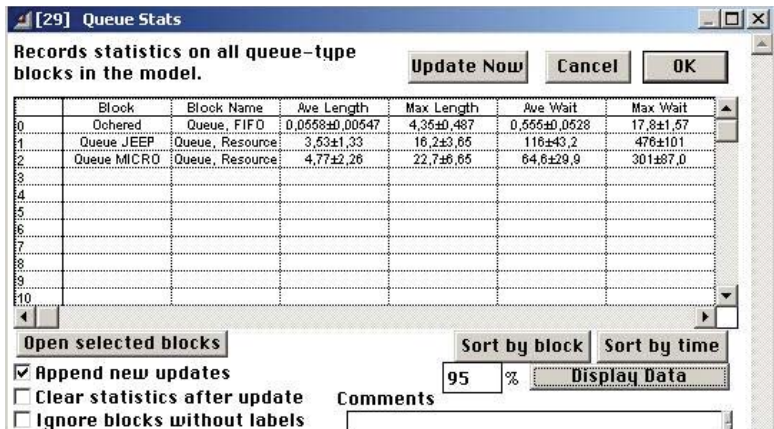


Рис. 4.3.19. Доверительные интервалы и результаты моделирования очередей

Средняя длина очереди за джипами с вероятностью 0,95 попадет в интервал (3,55–1,33; 3,55+1,33), а за экономичными автомобилями – в интервал (4,77–2,26; 4,77 + 2,26).

Максимальное время ожидания джипа с вероятностью 0,95 попадет в интервал (476 – 101; 476 + 101), то есть составит не менее 375 мин., а за экономичными автомобилями – в интервал (301 – 87; 301 + 87), то есть составит не менее 214 мин.

Очевидно, что клиенты не будут ожидать автомобиль столько времени – от одного до двух часов (в среднем).



Следовательно, необходимо увеличить число автомобилей. Но на сколько? Ведь увеличение их количества приведет к увеличению затрат на автомобили. Возникает вопрос о необходимом количестве, удовлетворяющем потребности клиентов и требующем наименьших затрат.

10. Для поиска оптимального варианта производится сравнение результатов экспериментов. План экспериментов состоит в том, чтобы, зная результат моделирования при 100 больших и 140 экономичных автомобилях, увеличить начальное количество автомобилей каждого типа примерно на 15%, то есть до 115 больших и 161 экономичного, а затем до 132 больших и 185 экономичных, и произвести новую 20-кратную имитацию (рис. 4.3.20).

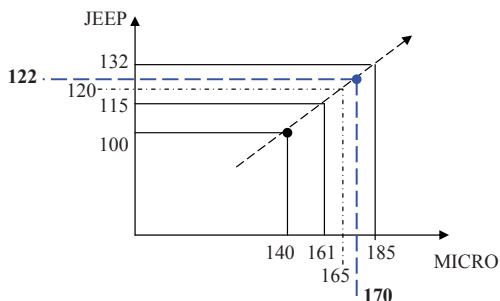


Рис. 4.3.20. Стратегия перебора альтернативных вариантов начального количества автомобилей от исходного варианта (100, 140) до (122, 170)

На рис. 4.3.21 показаны 95-процентные доверительные интервалы для наблюдаемых величин после 20 имитаций модели. Имитация с начальными значениями 115 и 161 показывает значительное уменьшение времени ожидания клиентов.

Среднее время ожидания для клиентов больших автомобилей равно  $7,02 \pm 5,93$  мин., а для клиентов экономичных автомобилей —  $1,06 \pm 1,23$  мин.

Следующий шаг приводит к имитации варианта (132, 185). Как видно из результатов моделирования, при этом варианте очереди отсутствуют. Но этот вариант значительно дороже по содержанию автопарка.

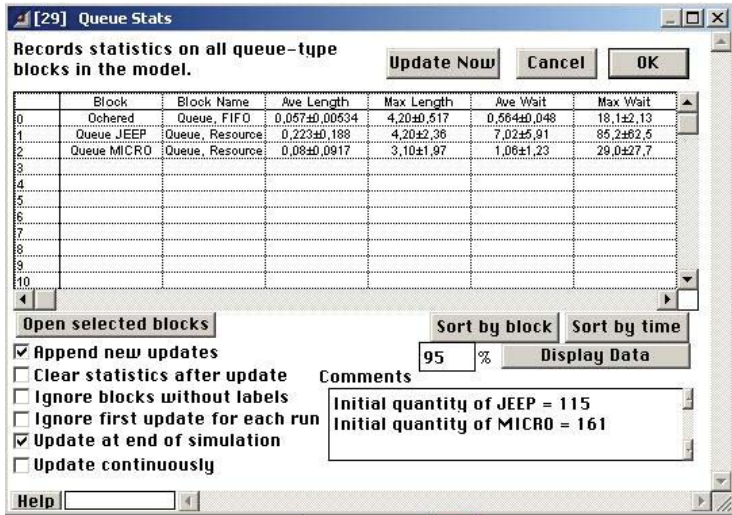


Рис. 4.3.21. Доверительный интервал времени ожидания варианта (115, 161)

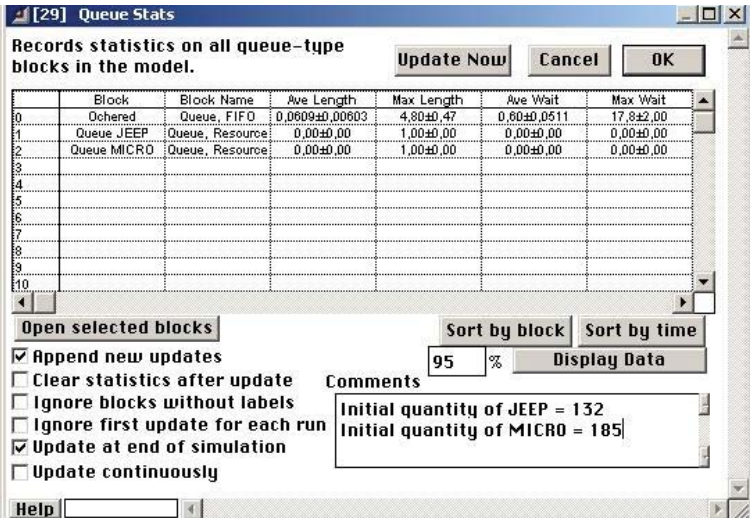


Рис. 4.3.22. Доверительный интервал времени ожидания варианта (132, 185)

Следующий шаг приводит к имитации варианта (120,165).

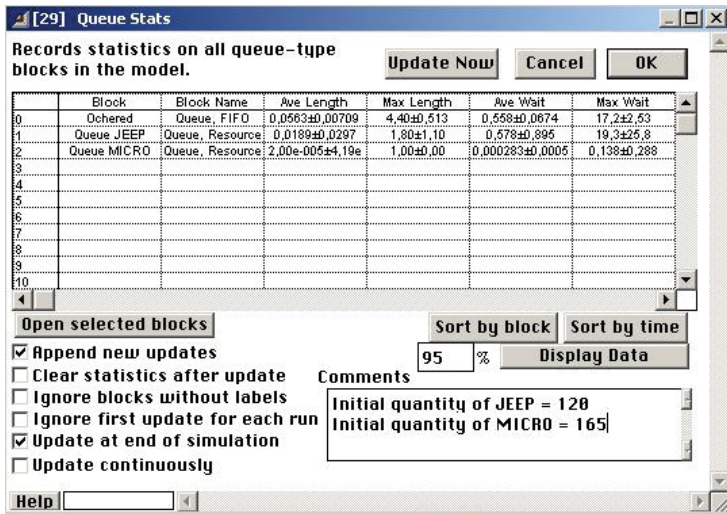


Рис. 4.3.23. Доверительный интервал времени ожидания варианта (120, 165)

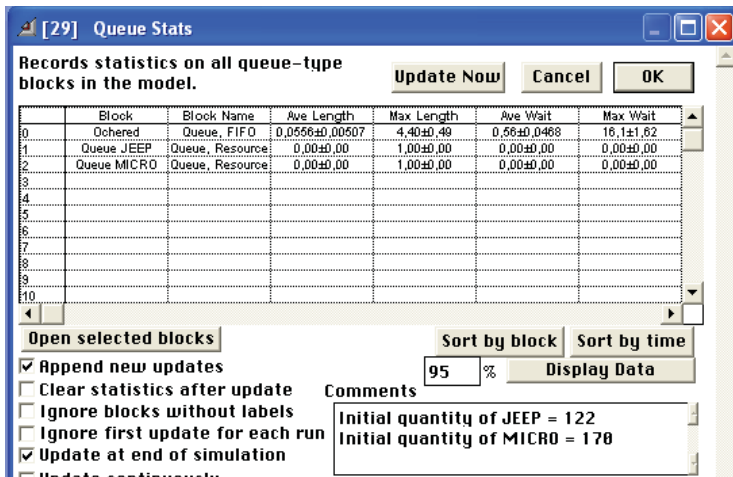


Рис. 4.3.24. Доверительный интервал времени ожидания варианта (122, 170)

В полученных результатах настраивает только один показатель: максимальное время ожидания большого автомобиля может оказаться в интервале от 0 до 45 мин.

Наконец, имитация варианта (122, 170) устойчиво даёт следующий результат (рис. 4.3.24).

Таким образом, нулевые интервалы ожидаемых средней и максимальной задержек в предоставлении автомобилей достигаются уже при варианте (122, 170), который значительно дешевле варианта (132, 185).

11. Интерпретация результатов моделирования в Extend, как и структурное моделирование, оказывается неотделимой от процесса моделирования и имитации. В результате модель претерпевает изменения как по структуре, так и по задаваемым параметрам, и позволяет не только описать поведение моделируемой системы, но и провести поиск оптимального соотношения рассматриваемых параметров (количества стоек обслуживания, начального количества автомобилей каждого типа, временных интервалов проката, вероятностных распределений случайных интервалов времени и т. д.).

#### **Лабораторная работа 4**

### **Разработка имитационной модели станции технического обслуживания в Extend LT, создание иерархического блока, проведение анализа и принятие управленческого решения на его основе**

#### **Исходные условия**

**Объект моделирования:** станции технического обслуживания автомобилей.

#### **Описание моделируемой системы**

Клиенты на автомобилях прибывают на СТО с интервалами времени распределенными экспоненциально со средним значением, равным 20 мин. Если в очереди на оформление обслуживания находится более 3 клиентов, клиент покидает СТО. Оформлением заявок занимается один работник. На оформление заявки одного клиента в зависимости от потребностей затрачивается от 5 до 9 минут (равномерное распределение). После оформления 60% машин направляются на мойку. Продолжительность мытья распределена равномерно в интервале от 10 до 20 мин.

В результате оформления автомобиль может получить несколько (от 1 до 4) заявок на обслуживание: техническое обслуживание (ТО); механический ремонт (МР); ремонт электрооборудования (ЕЛ); регулировка развала/схождения (РС).

По статистике обращений ТО требуется для 20% автомобилей клиентов, механический ремонт – для 40% автомобилей, ремонт электрооборудования – для 50% автомобилей, регулировка развала/схождения – для 30% автомобилей.

После мойки машина поступает на распределительную площадку, где дожидается своей очереди на один из заявленных видов обслуживания. На место обслуживания машина перегоняется водителем-диспетчером (количество водителей-диспетчеров может изменяться, предусмотреть возможные значения от 1 до 5).

### **Места обслуживания**

**Участок ТО и мехремонта.** ТО и механический ремонт производятся на общем участке, имеющем 5 подъемников. Если машине требуется и ТО, и мехремонт, то обслуживание производится без возвращения машины на распределительную площадку. Время, затрачиваемое на ТО, распределено равномерно от 30 до 80 минут. Время, затрачиваемое на мехремонт, распределено равномерно от 40 до 90 минут. После завершения обслуживания на этом участке машина возвращается водителем-диспетчером на распределительную площадку с отметкой о выполненных работах.

**Участок электроремонта.** Время, затрачиваемое на электроремонт, распределено равномерно от 40 до 120 минут. После завершения обслуживания на этом участке машина возвращается водителем-диспетчером на распределительную площадку с отметкой о выполненной работе.

**Участок регулировки развала/схождения.** Время, затрачиваемое на регулировку развала/схождения, распределено равномерно от 40 до 120 минут. После завершения обслуживания на этом участке машина возвращается водителем-диспетчером на распределительную площадку с отметкой о выполненной работе.

## Построение Extend-модели

Моделью автомобиля в имитационной Extend-модели служит транзакт, описываемый его атрибутами:

ServTO, ServMC, ServEL, ServRS.

Атрибут ServTO принимает значения 0 и 1, что означает отсутствие или наличие требования на техобслуживание автомобиля, моделируемого данным транзактом. Назначение атрибутов ServMC, ServEL, ServRS – аналогично.

Для создания модели требуется открыть библиотеки (Library – Open Library)

GENERIC.LIX

DE.LIX

MFG.LIX

STATS.LIX

PLOTTER.LIX

Модель составлена из 15 типов блоков (в скобках указаны библиотеки, из которых эти блоки берутся):

- 1) *Executive* (DE.LIX)
- 2) *Generator* (DE.LIX)
- 3) *Select DE Output* (DE.LIX)
- 4) *Queue (FIFO)* (DE.LIX)
- 5) *Activity, Multiple* (DE.LIX)
- 6) *Decision* (GENERIC.LIX)
- 7) *Input Random Number* (GENERIC.LIX)
- 8) *Combine (5)* (MFG.LIX)
- 9) *Buffer* (MFG.LIX)
- 10) *Set Attribute (5)* (DE.LIX)
- 11) *DE Equation* (DE.LIX)
- 12) *Get Attribute* (DE.LIX)
- 13) *Combine* (DE.LIX)
- 14) *Exit (4)* (DE.LIX)
- 15) *Plotter, Discrete Event* (PLOTTER.LIX)

Перечисленные блоки имеют следующие назначения.

Блок *Executive* (DE.LIX) – обеспечивает модель единым временем.

Блок **Generator** (DE.LIX) моделирует прибытие клиентов на СТО. Блок порождает транзакты с заданным в настройках экспоненциальным распределением со средним значением 20 мин.

Блок **Select DE Output** (DE.LIX) – обеспечивает выбор одного из двух выходов для транзакта в зависимости от заданного условия.

Блок **Queue, FIFO** (DE.LIX) – моделирует очередь типа FIFO.

Блок **Activity, Multiple** (DE.LIX) – моделирует задержку клиентов в заданном количестве параллельных каналов обслуживания.

Блок **Decision** (GENERIC.LIX) – моделирует принятие решения в зависимости от соотношения двух величин, поданных на входные коннекторы.

Блок **Input Random Number** (GENERIC.LIX) – генерирует случайную величину с заданным вероятностным распределением.

Блок **Combine (5)** (MFG.LIX) – объединяет 5 потоков транзактов в один.

Блок **Buffer** (MFG.LIX) – обеспечивает буферизацию входящих транзактов до заданного количества в буфере.

Блок **Set Attribute (5)** (DE.LIX) – создает атрибуты и устанавливает значения нескольких (до 5) атрибутов транзакта.

Блок **DE Equation** (DE.LIX) – вычисляет значение алгебраического выражения, включающего значения переменных, констант и атрибутов транзакта на текущий момент.

Блок **Get Attribute** (DE.LIX) – получение значения атрибута транзакта в численном виде.

Блок **Combine** (DE.LIX) – объединение двух потоков транзактов в один.

Блок **Exit (4)** (DE.LIX) – обеспечивает вывод из модели нескольких транзактов (до 4).

Блок **Plotter, Discrete Event** (PLOTTER.LIX) – обеспечивает графическое отображение результатов моделирования.

1. Блок **Generator** генерирует транзакты. Распределение интервалов времени между порождаемыми транзактами задается экспоненциальным законом со средним значением 20. Это значит, что в среднем каждые 20 минут на станцию техобслуживания прибывают автомобили, причем интенсивность потока примерно постоянна. В модели время исчисляется в целых минутах.

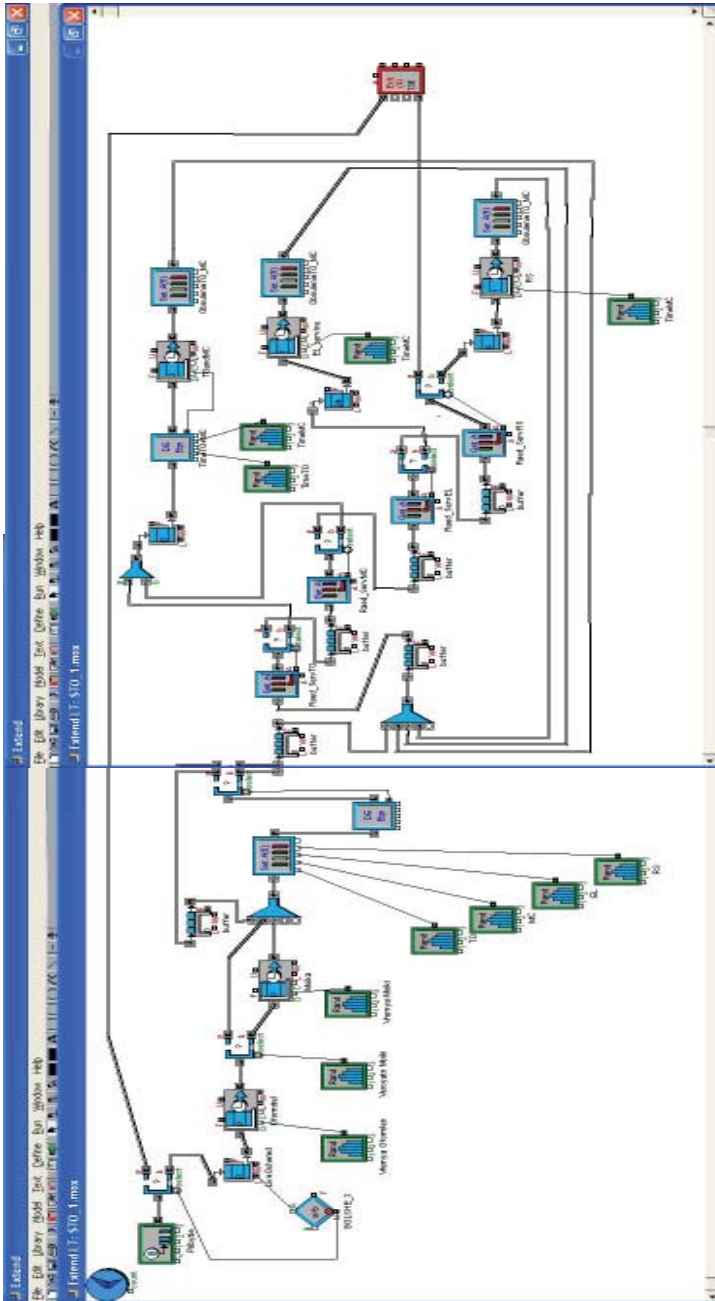


Рис. 4.4.1. Общий вид модели станции технического обслуживания



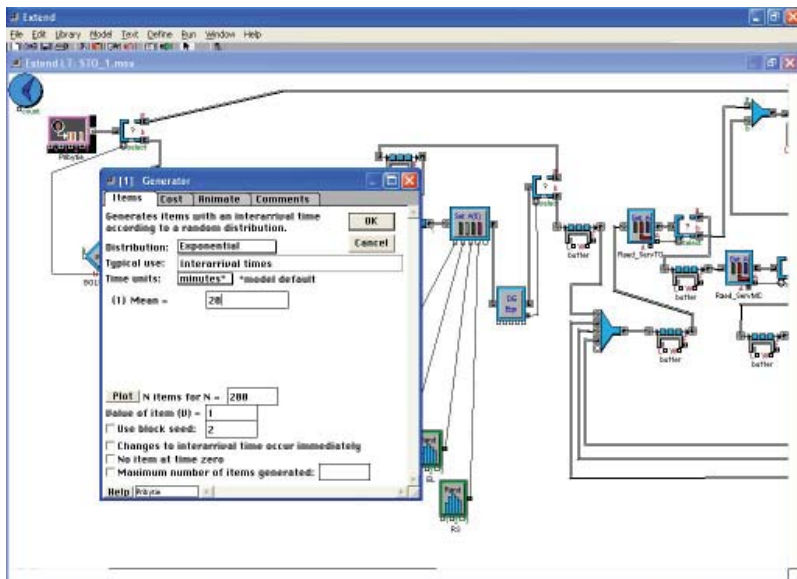


Рис. 4.4.2. Диалог блока *Generator*

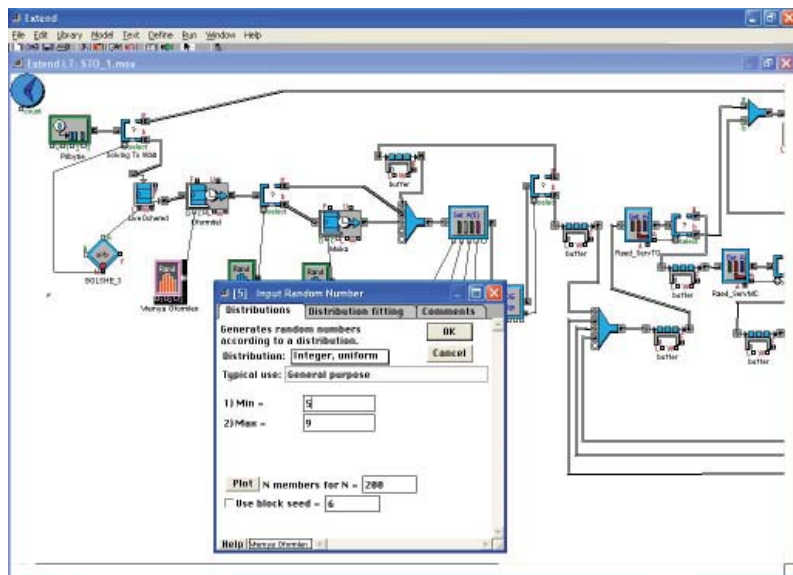


Рис. 4.4.3. Диалог блока *Select DE Output*

2. Блок *Select DE Output*, представленный на рис. 4.4.3, имеет два выходных коннектора, по которым выходят транзакты. Выбор выхода основывается на значении, передаваемом на коннектор *Select*. Если это значение равно 0, транзакт выходит через верхний выход, если 1 – через нижний. Устанавливается опция «Решение принимается до вхождения транзакта в блок выбора». В качестве переменной на вход *Select* передается результат сравнения блоком *Decision*.

3. Блок *Decision* (Решение). Сравнивает два значения: входной параметр А и вводимый параметр В. Значение В задается в диалоге и для данной модели (в данном случае=3). Формируется функция выбора. Если при сравнении двух значений  $A > B$ , то выходному значению присваивается значение «Истинности», а в противном случае – «Ложности». Следовательно, если в очереди находятся четыре транзакта, то порождаемые транзакты будут направляться по первой ветви (длинной) на выход.

4. Блок *Input Random Number* (рис. 4.4.5) случайным образом задает время обработки транзактов, моделируя тем самым задержку на оформление заказа на все виды обслуживания. Время задержки задается целым числом, случайно выбранным из интервала от 15 до 19 минут с равномерным распределением вероятности.

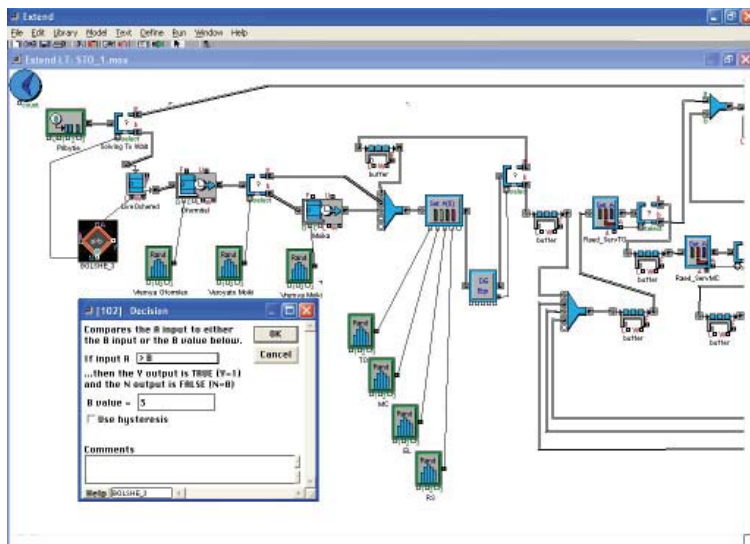


Рис. 4.4.4. Диалог блока *Decision*

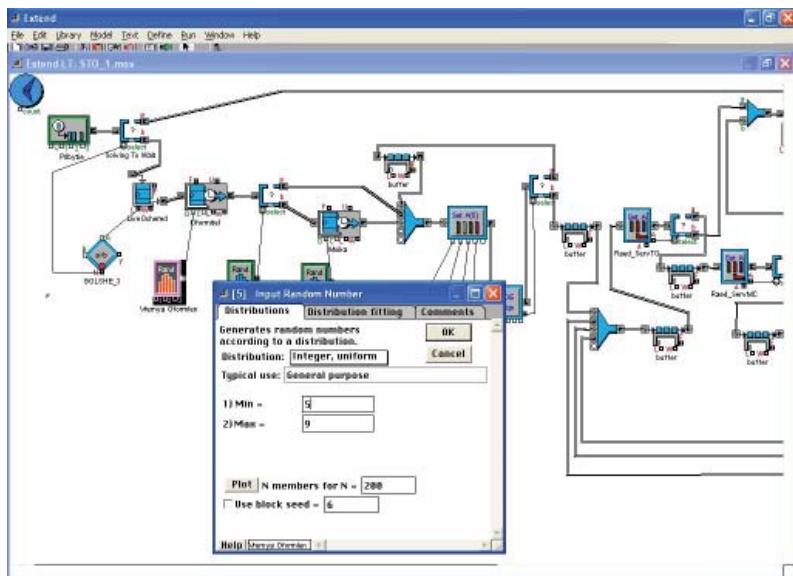


Рис. 4.4.5. Диалог блока *Input Random Number* (время оформления)

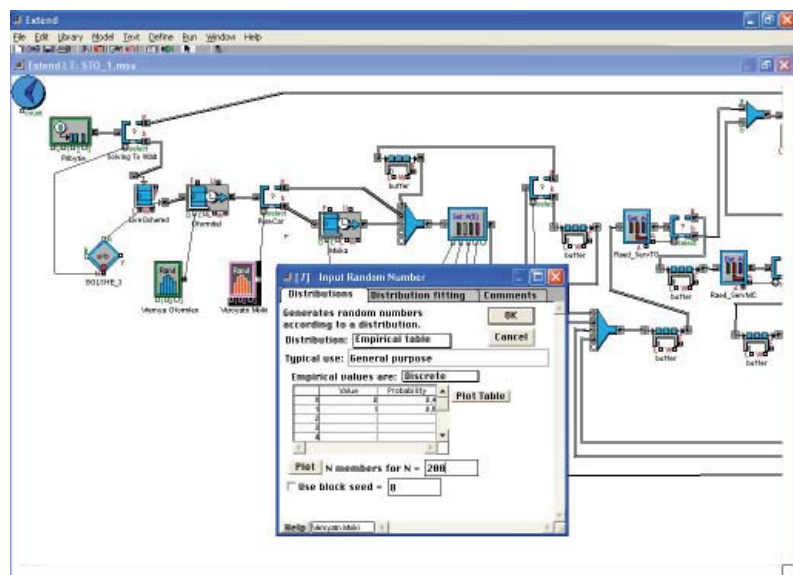


Рис. 4.4.6. Диалог блока *Input Random Number* (вероятность мытья машин)

5. Блок ***Input Random Number*** (рис. 4.4.6) задаёт вероятность прохождения машиной мойки. Вероятность задается с помощью эмпирической таблицы (Empirical Table). Значение 0, моделирующее отказ от мытья машины, присваивается с вероятностью 0,4, то есть оно сопровождается 40% транзактов. Значение 1, моделирующее необходимость мытья, сопровождается 60% транзактов, так как его вероятность равна 0,6. Присвоение любого другого значения невозможно.

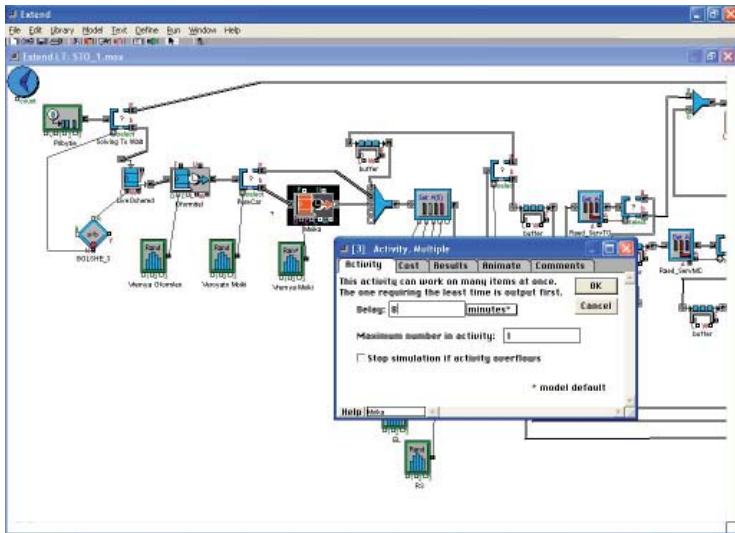


Рис. 4.4.7. Диалог блока ***Activity, Multiple*** (мойка)

6. Блок ***Activity, Multiple*** (рис. 4.4.7) моделирует задержку машины на время мытья. Одновременно происходит обработка только одного транзакта, что соответствует нахождению в мойке только одной машины. Время обработки задается случайным образом с помощью блока ***Input Random Number***.

7. Блок ***Input Random Number*** (рис. 4.4.8) задает случайное распределение интервала времени задержки транзактов, моделирующего время, занимаемое мытьем автомобилей. Время обработки задается целым числом, случайно выбранным из промежутка от 10 до 20 минут.

8. Поток транзактов, разделенный блоком ***Select DE Output*** на два потока, вновь соединяется в единый поток блоком ***Combine (5)***,

причем данный блок объединяет и транзакты, возвращаемые в случае неправильного задания всех атрибутов равными 0 (рис. 4.4.8).

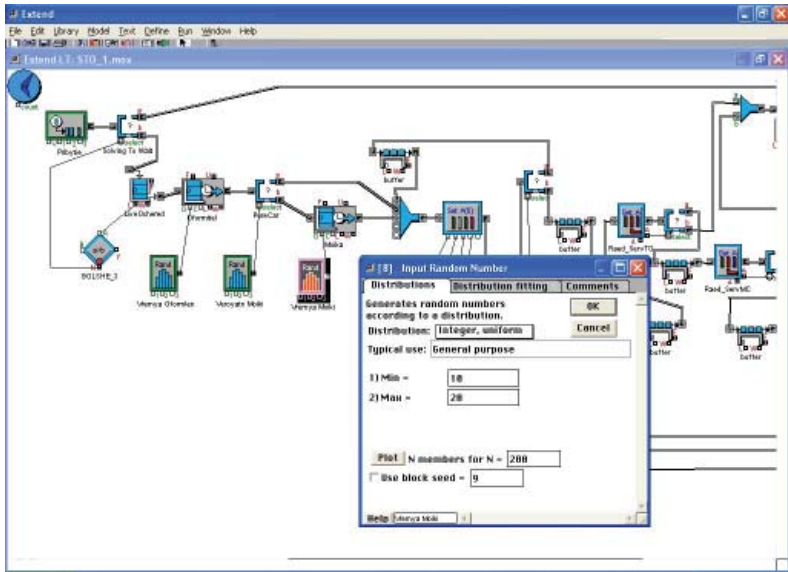


Рис. 4.4.8. Диалог блока *Input Random Number* (время мытья)

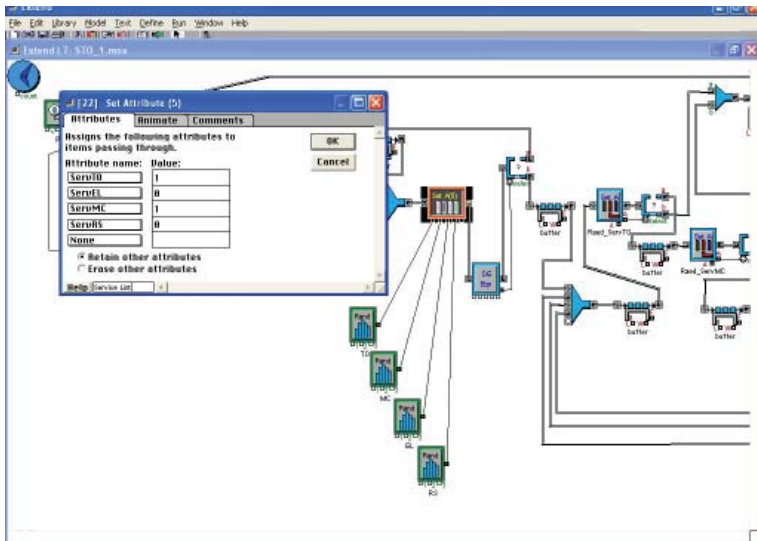


Рис. 4.4.9. Диалог блока *Set Attribute* (назначение видов обслуживания)

9. Блок *Set Attribute* задаёт 4 атрибута транзакта, вошедшего в блок. Атрибуты:

ServTO (техническое обслуживание),

ServMC (механический ремонт),

ServEL (ремонт электрооборудования),

ServRS (регулировка развала/схождения).

Для каждого из перечисленных видов обслуживания имеются статистические данные, показывающие, какая доля клиентов заказывает данный вид обслуживания. С этой статистически определенной вероятностью атрибут получает значение, равное 1, а нулевое значение атрибута означает, что данный вид обслуживания не заказан.

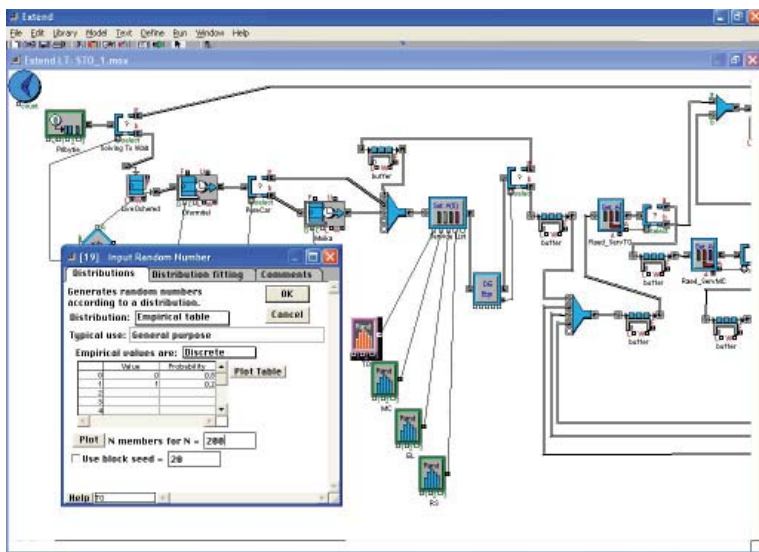


Рис. 4.4.10. Диалог блока *Input Random Number* (требование ТО)

10. Первый из 4 блоков *Input Random Number* задаёт вероятность присвоения значения 1 атрибуту *ServTO* транзакта, находящегося в блоке *Set Attribute*. Для атрибутов *ServMC*, *ServEL*, *ServRS* создаются аналогичные блоки, в которых в эмпирических таблицах задаются соответствующие вероятности присвоения атрибутам значения 0 или 1.

Например, блок *Input Random Number*, задающий значение атрибута *ServTO*, задаёт значение 0 с вероятностью 0,8, значение 1 – с вероятностью 0,2. Значение этой случайной величины с выход-

ного коннектора блока *Input Random Number* передается на входной коннектор блока *Set Attribute (5)*, соответствующий атрибуту *ServTO* (рис. 4.4.10). Задание остальных атрибутов выполняется аналогично.

11. Здесь следует предусмотреть ситуацию, в которой все генераторы случайных чисел одновременно установят нулевые значения для всех 4 атрибутов. Блок *DE Equation* рассчитывает алгебраическое выражение для проходящего транзакта. Равенство всех атрибутов нулю однозначно определяется равенством нулю суммы всех 4 атрибутов транзакта. В блоке вычисляется сумма атрибутов ( $ServTO + ServMC + ServEL + ServRS$ ), значения которых вводятся через коннекторы блока.

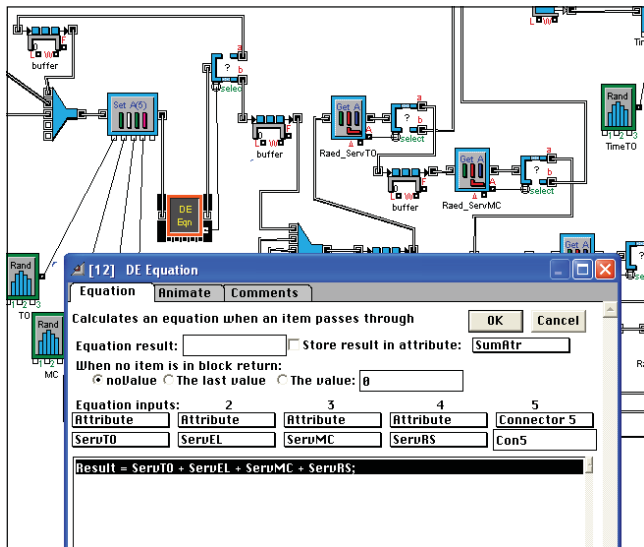


Рис. 4.4.11. Диалог блока *DE Equation* (проверка требований)

Выходной коннектор блока *DE Equation* пропускает транзакт на блок *Select DE Output*, который в зависимости от результата вычисления в блоке *DE Equation* направляет транзакт по одному из 2 выходов.

12. Выходной коннектор *A* блока *DE Equation* передает результат вычисления  $SumAttr$  на коннектор *Select* блока *Select DE Output*. При результате, равном 0, транзакт выйдет из блока *Select DE Output* через верхний выход, при результате, равном 1, — через нижний. При результатах, равных 2, 3 или 4, которые моделируют ситуации с 2, 3 или



4 видами требуемого обслуживания, транзакт также выйдет из блока через нижний выход, так как установлена опция *Invalid «select» value: chooses bottom output*. (При непредусмотренном значении выбирается нижний выход.) В диалоге устанавливается опция «Решение принимается до вхождения транзакта в блок выбора» (рис. 4.4.12).

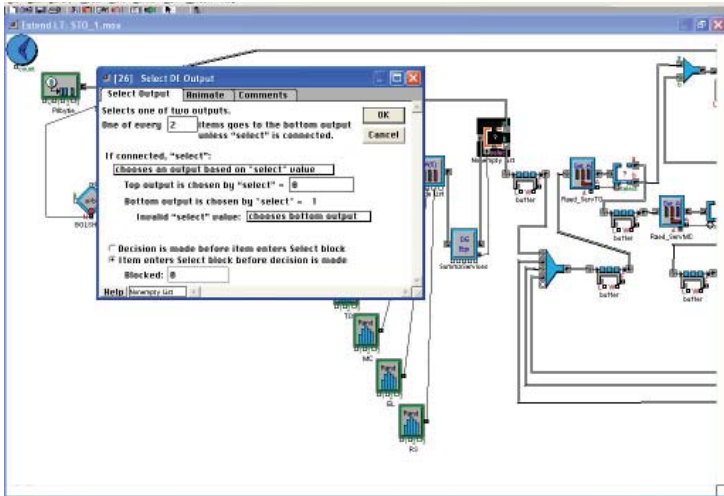


Рис. 4.4.12. Диалог блока *Select DE Output* (проверка требований)

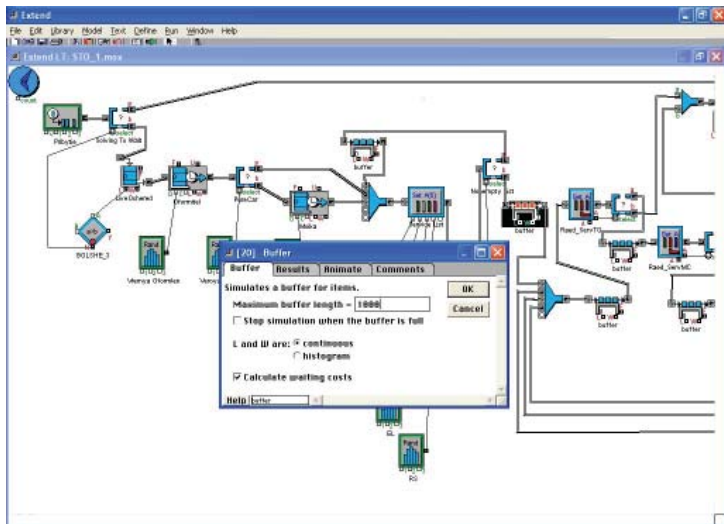


Рис. 4.4.13. Блоки *Buffer* (вспомогательная буферизация)



13. Блоки **Buffer**, следующие после обоих выходов блока **Select DE Output**, выполняют вспомогательную роль буферизации событий в процессе моделирования. Они обеспечивают возможность изменения состояний транзактов при пересчете значений их атрибутов. Настройки, установленные в блоке по умолчанию, обычно не изменяются.

14. Блок **Combine (5)** вместе со следующим блоком **Buffer** моделируют объединение потоков машин на «распределительной площадке», куда автомобили приходят как после первоначального оформления, так и после выполнения отдельных видов обслуживания.

15. Блок **Get Attribute** (получить значение атрибута) выбирает один из атрибутов транзакта (в данном случае *ServTO*) и передает его значение через выводной коннектор (рис. 4.4.14). Транзакт выходит неизменным через выходной коннектор. В диалоговом окне в выпадающем списке *Attribute name* выбирается имя атрибута. Задается опция *Read attribute* (читать атрибут). Опция *no value* означает, что если атрибут не найден, то никакое значение не передается.

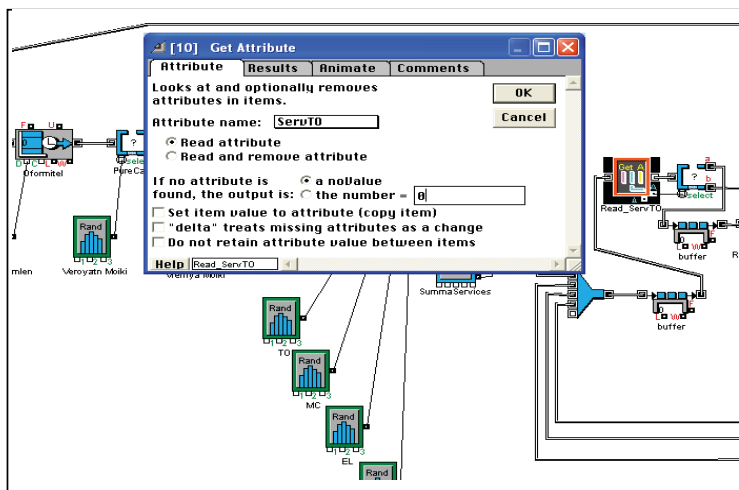


Рис. 4.4.14. Блок **Get Attribute**. Получение значения атрибута *ServTO*

16. Блок **Select DE Output** (рис. 4.4.15) действует аналогично описанному выше. В диалоге устанавливается опция «Решение принимается до вхождения транзакта в блок выбора».

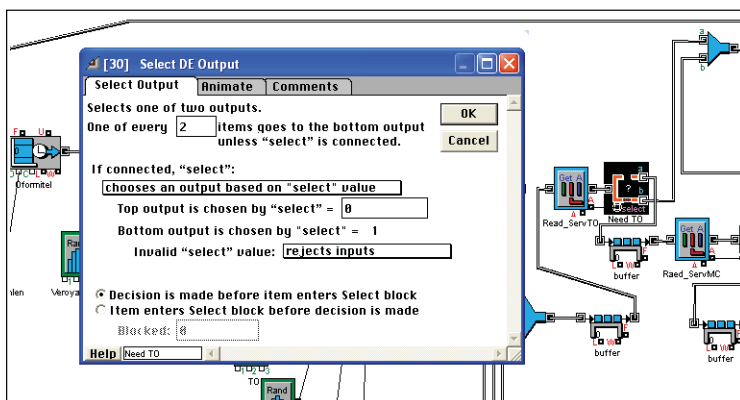


Рис. 4.4.15. Блок *Select DE Output*

17. Блок *Combine* моделирует объединение потоков машин, направляемых на ТО и механический ремонт с «распределительной площадки» на участок, имеющий 5 подъемников.

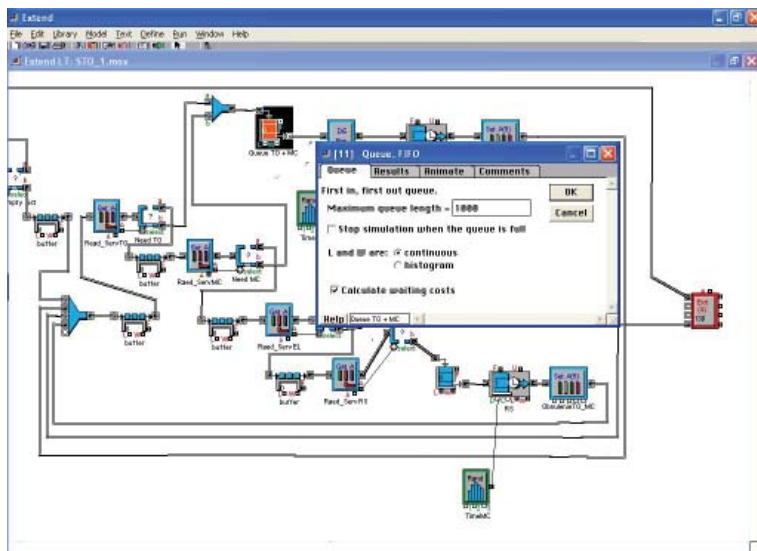


Рис. 4.4.16. Блок *Queue, FIFO*

18. Блок *Queue, FIFO* (рис. 4.4.16) моделирует ожидание направления на ТО и мехремонт в очереди на распределительной площадке. Данный блок хотя и не находится рядом с блоком *Combine* (5),

моделирующим некоторые функции распределительной площадки, но моделирует задержку, происходящую именно там.

19. Блок **DE Equation** в данном месте модели вычисляет время задержки транзакта в следующем блоке **Activity, Multiple** (моделирующем время, требующееся на обслуживание автомобиля на участке ТО и мехремонта). В диалоге блока задаются входные данные вычисляемого выражения, представляющие собой значения атрибутов, и данные, приходящие на коннекторы (*ServTO*, *ServMC*, *Connector 2*, *Connector 4*), и вид этого выражения:  $ServTO * Con2 + ServMC * Con 4$ . На указанные коннекторы подаются значения времени задержки на ТО и мехремонт соответственно, моделируемые блоками **Input Random Number**.

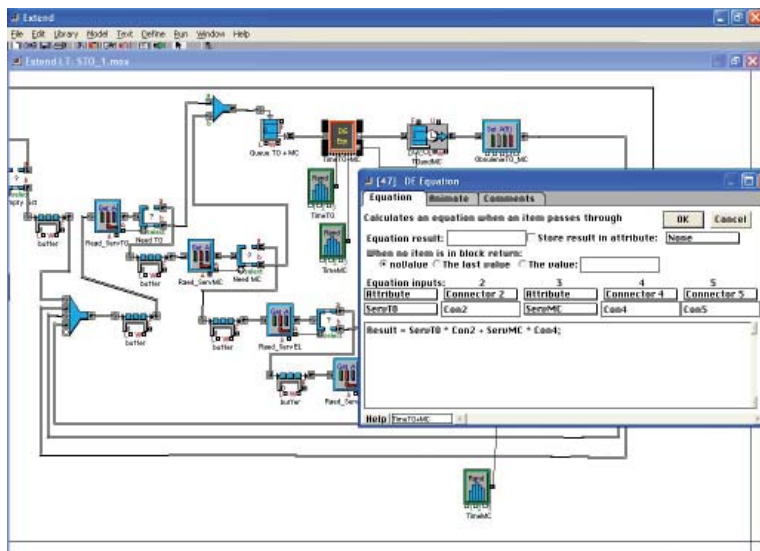


Рис. 4.4.17. Блок **DE Equation**

20. Блоки **Input Random Number** моделируют случайные распределения интервалов времени задержки на ТО и мехремонт. В первом блоке задается длительность технического обслуживания. Время обработки задается случайным целым числом, равномерно случайно распределенным от 30 до 80 мин. Вторым блоком **Input Random Number** аналогично задается целым числом, случайно распределенным от 40 до 90 мин., длительность задержки на мехре-

монт (рис. 4.4.18). Величины интервалов передаются на входные коннекторы блока *DE Equation*.

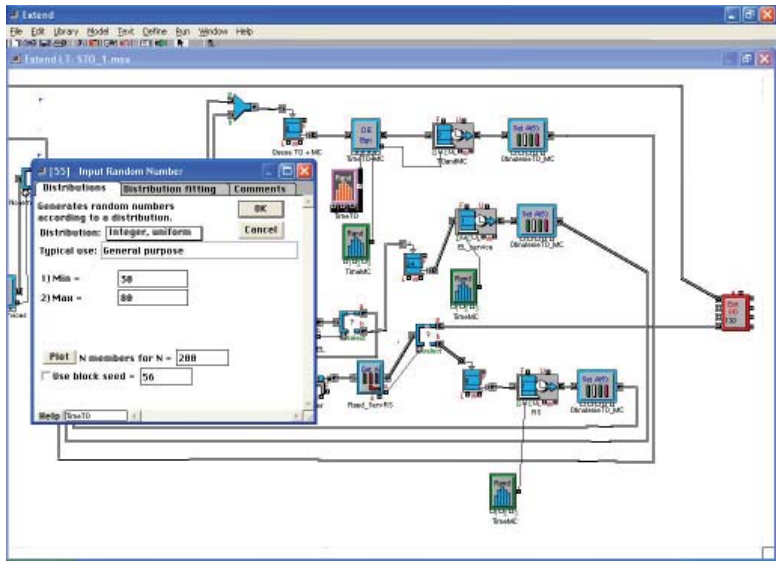


Рис. 4.4.18. Блоки *Input Random Number*

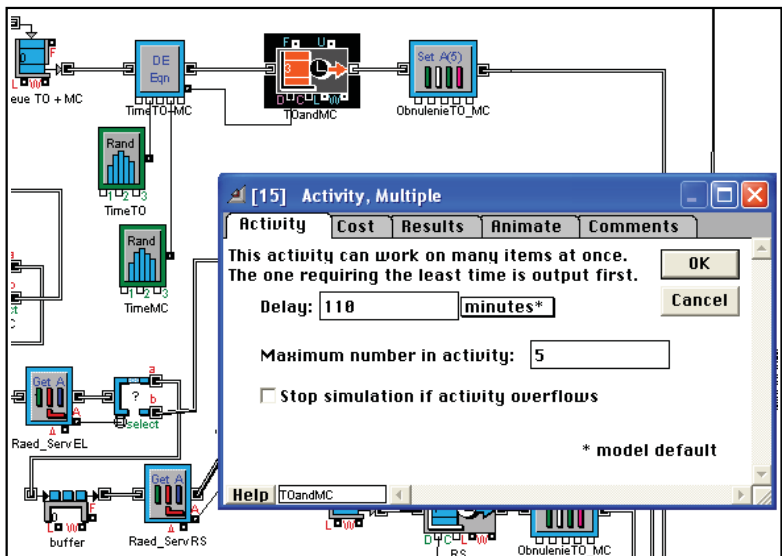


Рис. 4.4.19. Блок *Activity, Multiple* (участок ТО и мехремонта)

21. Блок *Activity, Multiple* моделирует техническое обслуживание машин и мехремонт. Одновременно может производиться обслуживание 5 машин, то есть обработка 5 транзактов. Время обработки пришедшего транзакта получается с выходного коннектора блока *DE Equation* (рис. 4.4.19).

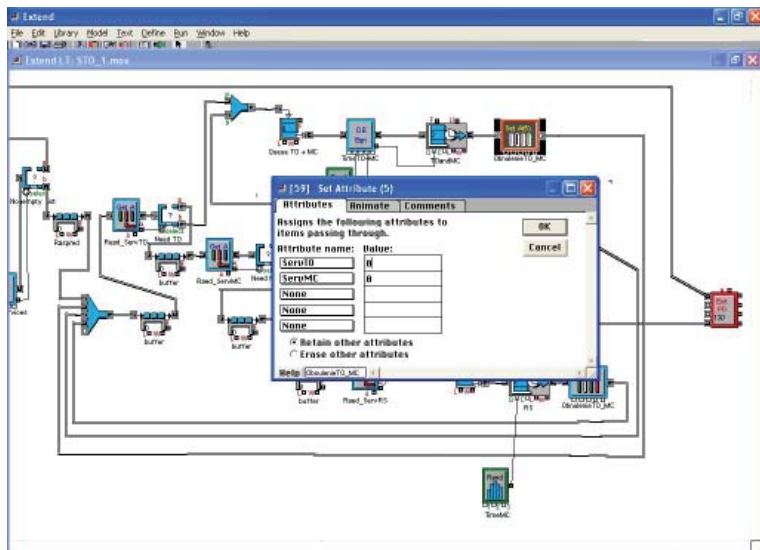


Рис. 4.4.20. Блок *Set Attribute* (отметка о прохождении участков ТО и мехремонта)

22. Блок *Set Attribute* обеспечивает снятие требования на ТО и мехремонт после выполнения данных видов обслуживания. В этом блоке атрибутам *ServTO* и *ServMC* транзакта присваиваются значения 0.

23. Участки электроремонта и регулировки развала/схождения моделируются аналогично участку ТО и мехремонта.

24. Возвращение автомобилей на распределительную площадку после выполнения каждого вида обслуживания моделируется возвратом транзактов на блок *Combine (5)*.

25. После выполнения всех видов обслуживания все атрибуты транзакта *ServTO*, *ServMC*, *ServEL*, *ServRS* получают нулевое значение и вследствие этого, пройдя все блоки *Select DE Output*,

направятся на блок *Exit (4)*, что моделирует выход автомобиля со станции техобслуживания (рис. 4.4.21).

26. Блок *Plotter* отображает коэффициенты загрузки трех участков обслуживания: ТО/мехремонта, электроремонта и регулировки развала/схождения.

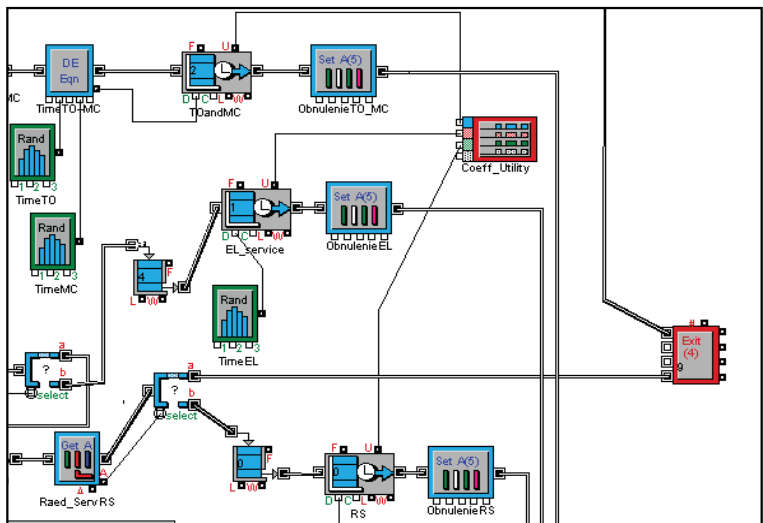


Рис. 4.4.21. Блок *Exit (4)* (выход автомобилей из СТО)

## Лабораторная работа 5

### Совершенствование имитационной модели станции технического обслуживания в Extend LT, создание иерархических блоков, моделирование работы водителя-диспетчера

#### Исходные условия

**Объект моделирования:** станция технического обслуживания автомобилей, описанная в лабораторной работе 4. Дополняется функция водителя-диспетчера, перегоняющего автомобили с распределительной площадки на участки обслуживания и на распределительную площадку после обслуживания.

## Описание фрагмента, моделирующего работу водителя-диспетчера

В построенной до данного момента модели не учитывается такой элемент реальной системы (ресурс модели), как водитель-диспетчер. Водитель-диспетчер перегоняет автомобили с распределительной площадки на участки обслуживания, а после выполнения обслуживания — обратно на распределительную площадку. Отсутствие в модели этого ресурса означает, что по ресурсу нет ограничения, то есть транзакты мгновенно переходят из блока в блок (автомобили мгновенно перегоняются из одного пункта в другой в момент освобождения канала обслуживания).

На самом деле перегон автомобилей занимает заметное время не столько из-за длительности переезда, сколько из-за занятости водителя-диспетчера. Вследствие этого в модели требуется отразить как временную задержку, так и ограниченный ресурс данного канала обслуживания.

Для отражения работы водителя-диспетчера, выполняющего функцию перегона, можно построить различные модели. Рассмотрим два способа моделирования:

1-й способ — с использованием блоков *Select Output, Get Attribute*;

2-й способ — с использованием блоков *Resource Pool, Queue Resource Pool, Release Resource Pool*.

### **1-й способ моделирования работы водителя-диспетчера**

Транзакты после каналов обслуживания через блок *Combine (5)* попадают в блок *Combine*, обеспечивающий два входа в группу блоков, моделирующих работу водителя-диспетчера (рис. 4.5.1).

Следующий блок *Activity, Multiple* моделирует задержку на перегон. (Многоканальный блок включен в модель для того, чтобы легко было моделировать изменение количества водителей-диспетчеров на СТО.)

Чтобы различать транзакты, приходящие до и после обслуживания, требуется ввести дополнительный атрибут транзакта. Этот атрибут будет определять, в каком направлении предстоит перегон транзакту — на распределительную площадку или на участок обслуживания.

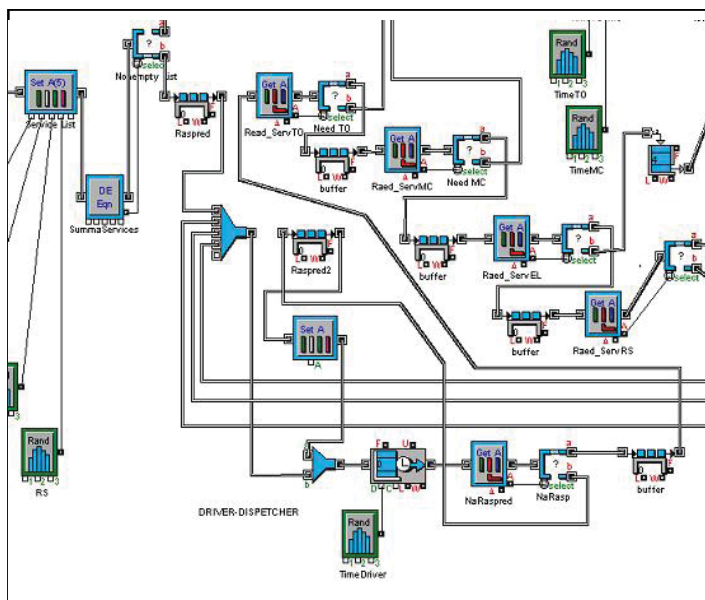


Рис. 4.5.1. Фрагмент, моделирующий работу водителя диспетчера.

Атрибут так и называется *NaRasp* (на распределительную площадку). Для его создания надо вернуться к блоку *SetAttribute*, моделирующему оформление заказа на обслуживание, и в диалоге этого блока дополнить атрибут *NaRasp* и присвоить ему значение 1. Действительно, в том месте транзакт движется к распределительной площадке.

Блок *Activity, Multiple* моделирует задержку времени на работу водителя-диспетчера, причем блок настраивается на задержку 1 транзакта (1 водитель-диспетчер).

Вероятностное распределение длительности задержки транзактов в блоке *Activity, Multiple* задается блоком *Input Random Number*, работа с которым описана выше (рис. 4.5.2).

- Блок *Get Attribute* (получение значения атрибута) определяет значение заданного атрибута у транзакта, вошедшего в блок.

В данном случае определяется значение атрибута *NaRasp*. В диалоговом окне выбирается пункт – *Read attribute* (прочитать атрибут). Если атрибут не найден, то значение атрибуту не присваивается (*no value*).



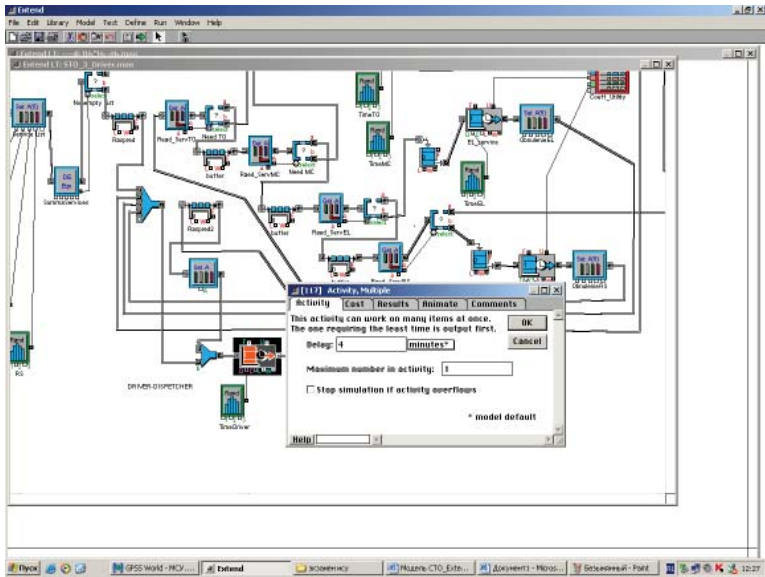


Рис. 4.5.2. Блоки *Combine*, *Activity*, *Multiple*, *Input Random Number* подсистемы, моделирующей работу водителя-диспетчера

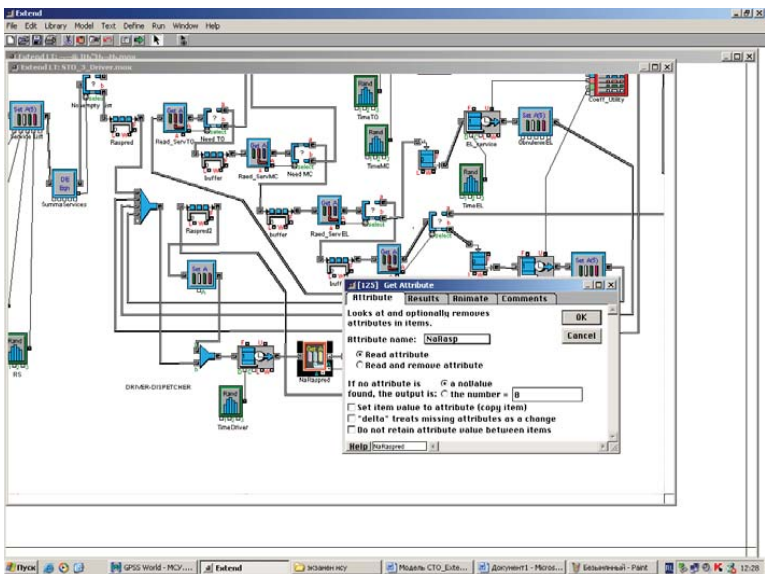


Рис. 4.5.3. Блок *Get Attribute*

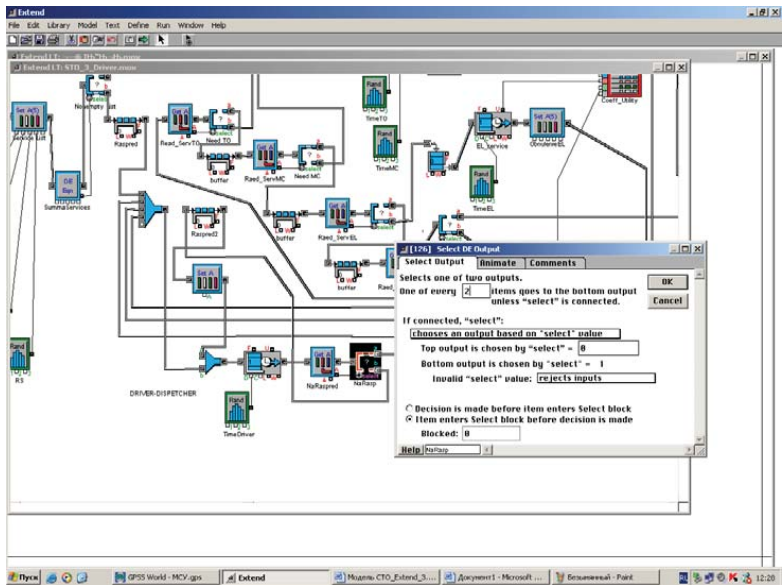


Рис. 4.5.4. Блок *Select DE Output*

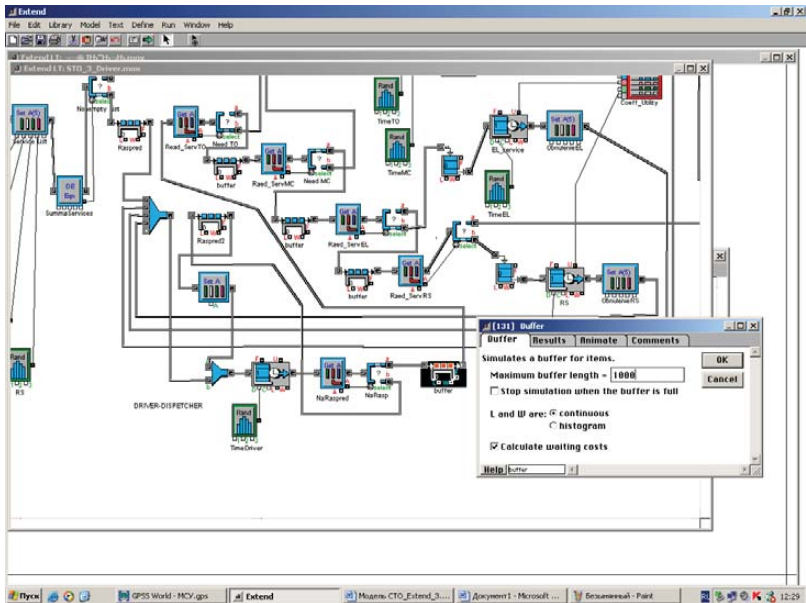


Рис. 4.5.5. Блоки *Select DE Output* и *Buffer*

Блок **Select DE Output** (выбор выхода) направляет транзакты по одному из двух выходов. Выбор выхода основывается на значении атрибута *NaRasp*, полученном коннектором *Select*. Если значение атрибута равно 0, транзакт направляется на верхний выход, если равно 1 – на нижний выход. В диалоге устанавливается опция «Решение принимается до вхождения транзакта в блок выбора».

Блоки **Buffer** после обоих выходов блока **Select DE Output** обеспечивают задержку на один такт для возможного изменения атрибутов транзакта. Настройки, установленные в блоке по умолчанию, удовлетворяют условиям нашей модели.

Блок **Set Attribute** (установка атрибутов) позволяет задать новое значение атрибута *NaRasp*.

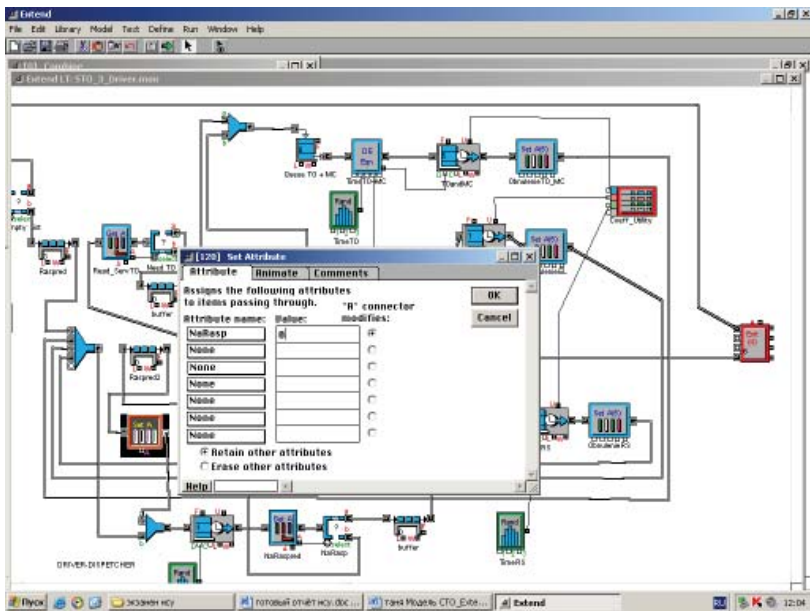


Рис. 4.5.6. Блок **Set Attribute** (обнуление значения атрибута *NaRasp*)

Транзакт, движущийся на распределительную площадку ( $NaRasp=1$ ) и вышедший через нижний выход блока **Select DE Output**, получает новое значение атрибута ( $NaRasp=0$ ). Это указывает, что задержка на перегон автомобиля на распределительную площадку смоделирована.

Вслед за этим транзакт с атрибутом  $NaRasp=0$  вновь через блок **Combine** направляется на моделирование перегона, теперь с распределительной площадки на участок обслуживания. Следует отметить, что задержка в общей очереди на распределительной площадке моделируется отдельными очередями на участках обслуживания. Здесь также возможны другие варианты.

Прежде чем переходить к описанию 2-го способа моделирования работы водителя-диспетчера, следует обратить внимание на слишком усложнившуюся структуру модели. Усложнение структуры модели затрудняет понимание происходящего процесса и приводит к потере основного преимущества Extend-модели – её наглядности (включая анимацию).

Чтобы структурировать модель на более крупные подсистемы, следует воспользоваться иерархическими блоками.

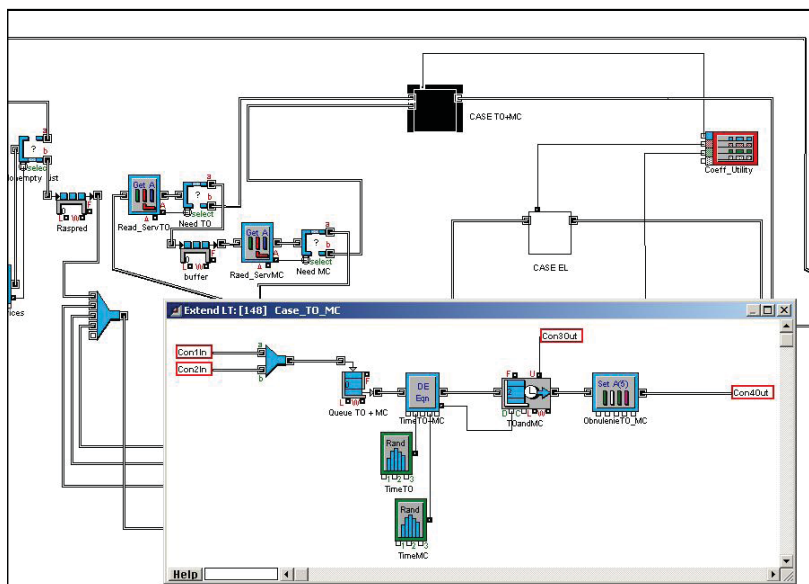


Рис. 4.5.7. Построение иерархического блока, моделирующего участок ТО и мехремонта

Порядок построения иерархических блоков описан в лабораторной работе 3, посвященной модели предприятия по прокату автомобилей.

(Выделить блоки, включаемые в иерархический блок, – *Model* – *Make Selection Hierarhical*. Задать имя иерархического блока в поле *Name* – *Make H-block*.)

Диалоговое окно иерархического блока **CASE TO + MC** на рис. 4.5.7 показывает, какие блоки объединены вместе.

Аналогичным образом в иерархические объединяются блоки, моделирующие участки электроремонта, регулировки развала/схождения и работу водителя-диспетчера. В результате модель принимает вид, показанный на рис. 4.5.8.

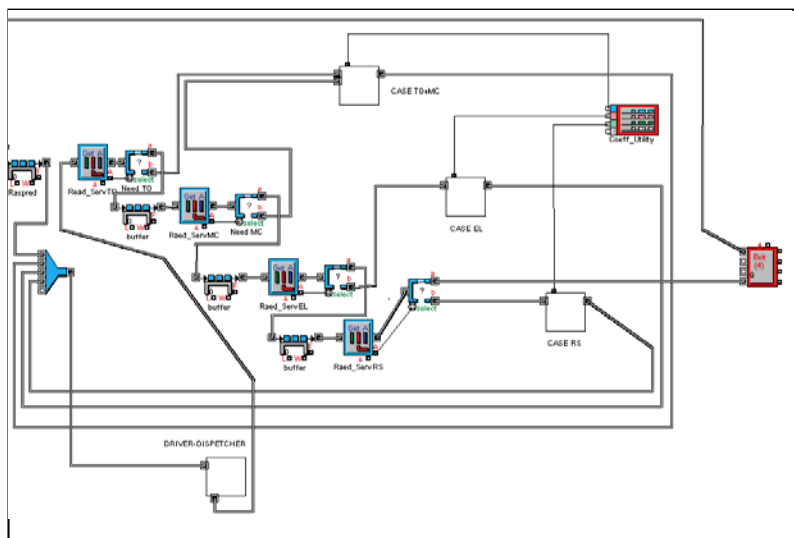


Рис. 4.5.8. Построение иерархических блоков CASE EL, CASE RS и DRIVER-DISPETCHER

После построения иерархических блоков LIVE OCHERED, MOJKA, NAZNACHENIE, RASPRD PLOSH, моделирующих очередь клиентов, мойку, работу оформителя и распределительную площадку соответственно, модель принимает вид, представленный на рис. 4.5.9.

При этом иерархический блок DRIVER-DISPETCHER выглядит, как представлено на рис. 4.5.10.

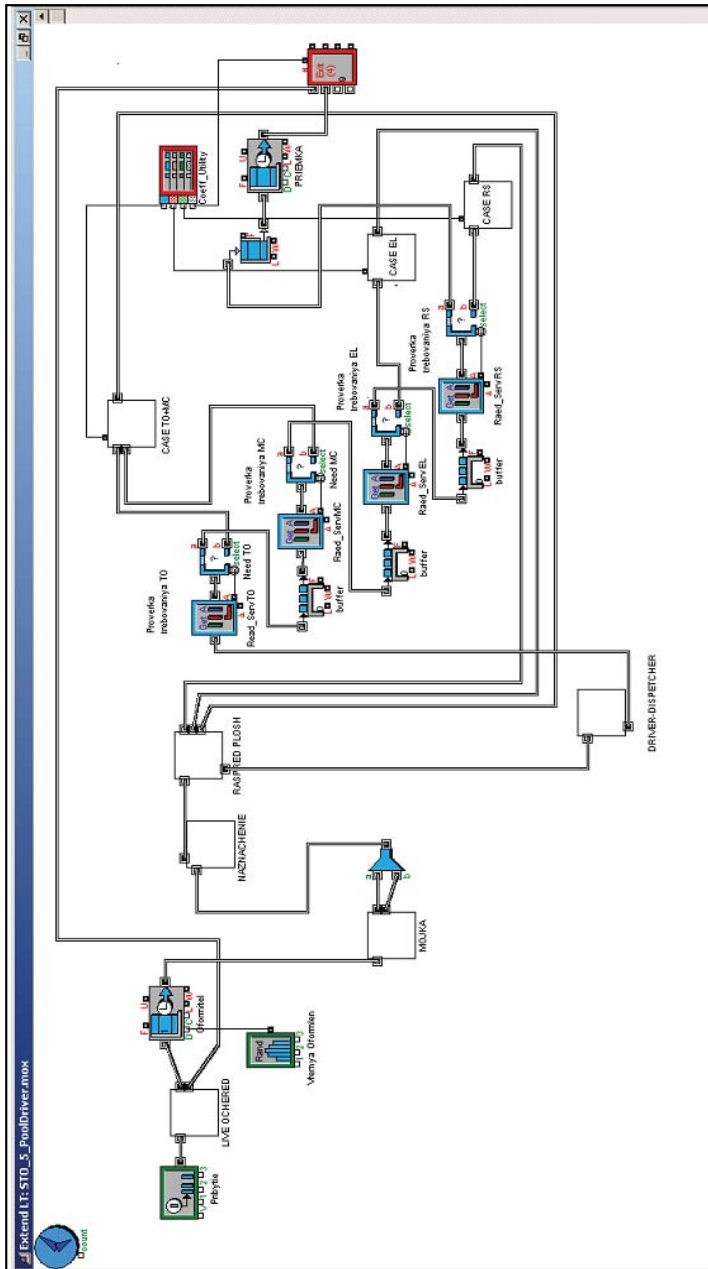


Рис. 4.5.9. Модель СТО с иерархическими блоками участков обслуживания

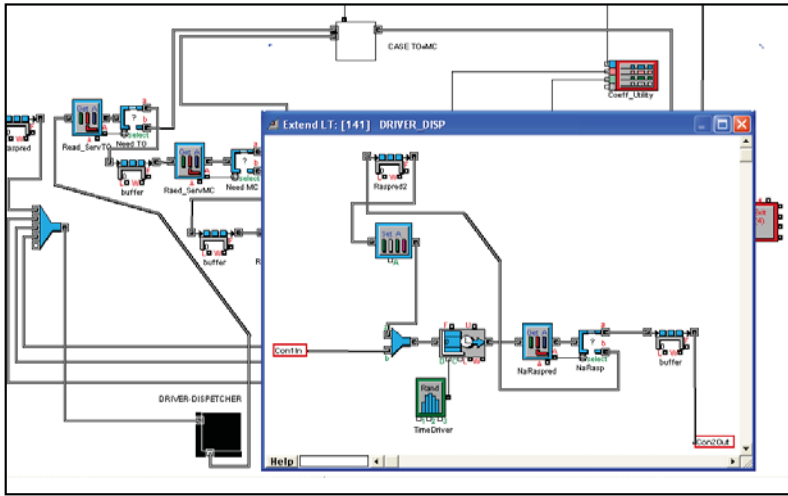


Рис. 4.5.10. Иерархический блок DRIVER-DISPETCHER и его структура

### **2-й способ моделирования работы водителя-диспетчера**

Как видно из рис. 4.5.10, иерархический блок DRIVER-DISPETCHER, моделирующий работу водителя-диспетчера, имеет один входной коннектор и один выходной. Таким образом имитируется поступление автомобиля к водителю (вход транзакта в суперблок) и освобождение водителя после доставки автомобиля по назначению (выход транзакта из суперблока).

Как внутренняя структура, так и настройки элементов суперблока могут изменяться, при этом вход и выход остаются прежними.

Эта возможность демонстрируется с помощью 2-го способа построения модели работы водителя-диспетчера. Новая подмодель будет иметь те же самые входной и выходной коннекторы, но её структура будет отличаться от первого варианта.

В данном случае в суперблок кроме блоков *Activity*, *Multiple*, *Input Random Number*, *Buffer* и *Plotter* включаются блоки *Queue*, *Resource Pool* (рис. 4.5.13); *Resource Pool* (рис. 4.5.14); *Release*, *Resource Pool* (рис. 4.5.15).

В диалоге блока *Queue*, *Resource Pool* (рис. 4.5.13) задается имя пула (*DriverDis*), из которого запрашивается ресурс (водитель-дис-

петчер) для выполнения обслуживания, а также количество ресурсов, запрашиваемых на каждое обслуживание, равное 1. Максимальная длина очереди установлена равной 1, но этот параметр может варьироваться в дальнейшем при исследовании различного числа водителей и параметров очереди на перегон.

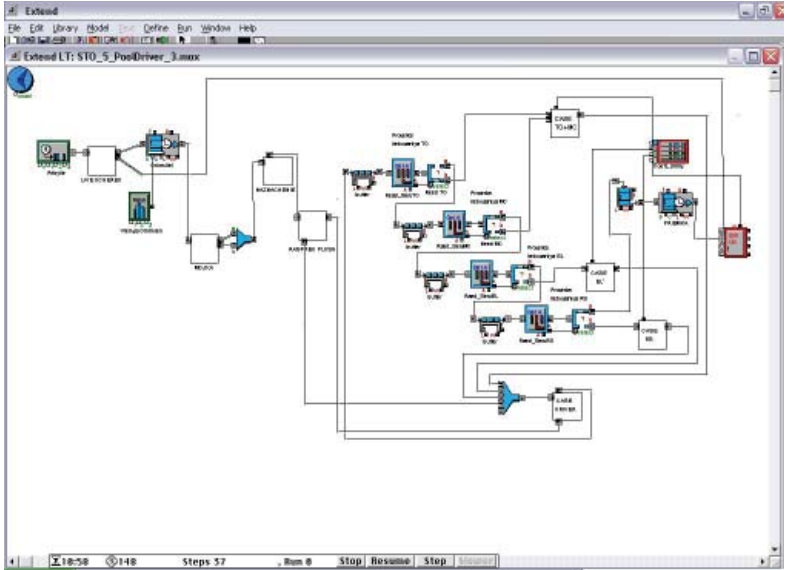


Рис. 4.5.11. Новая модель иерархического блока DRIVER-DISPATCHER

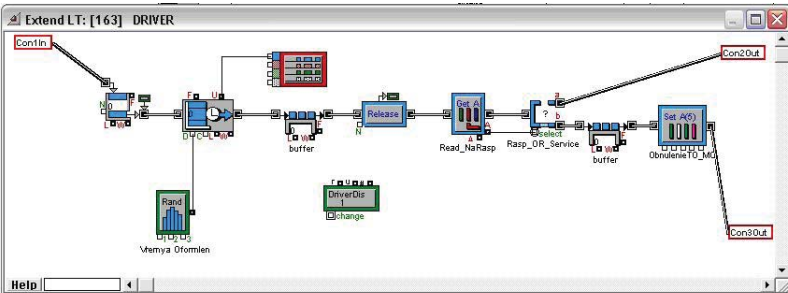


Рис. 4.5.12. Иерархический блок *Driver*



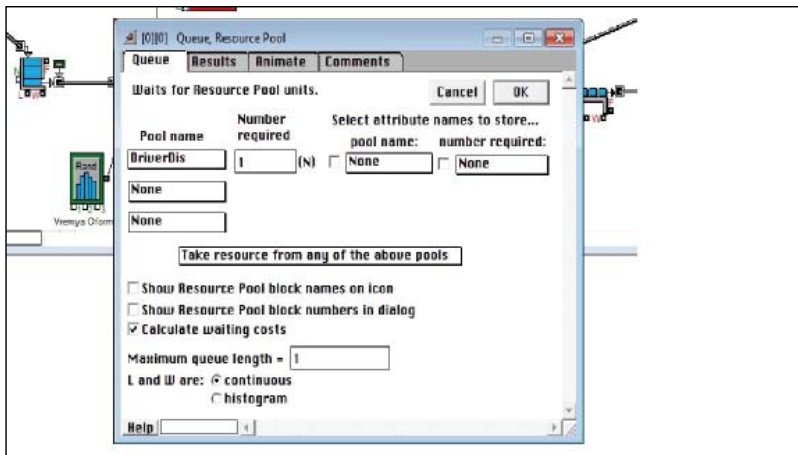


Рис. 4.5.13. Диалог блока *Queue, Resource Pool*

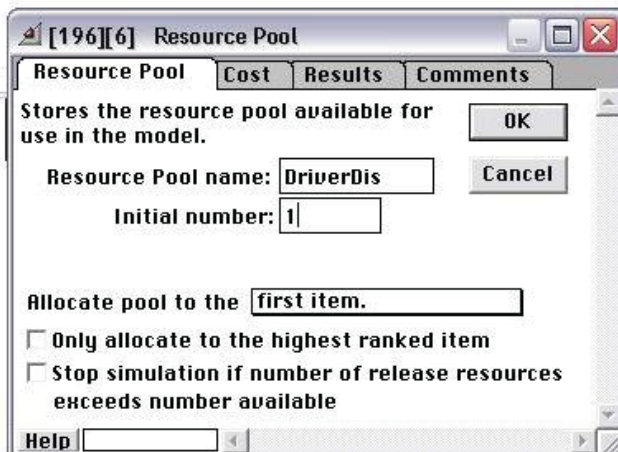


Рис. 4.5.14. Блок *Resource Pool* в составе иерархического блока

Блок *Resource Pool* моделирует заданное количество водителей. В диалоге блока (рис. 4.5.14) задается имя пула (*DriverDis*) и первоначальное количество доступных ресурсов, равное 1.

В диалоге блока *Release Resource Pool* (рис. 4.5.15) задается имя пула (*DriverDis*), в который возвращается ресурс (водитель-диспетчер), освобождаемый после выполнения обслуживания, а также количество освобождаемых ресурсов, равное 1.

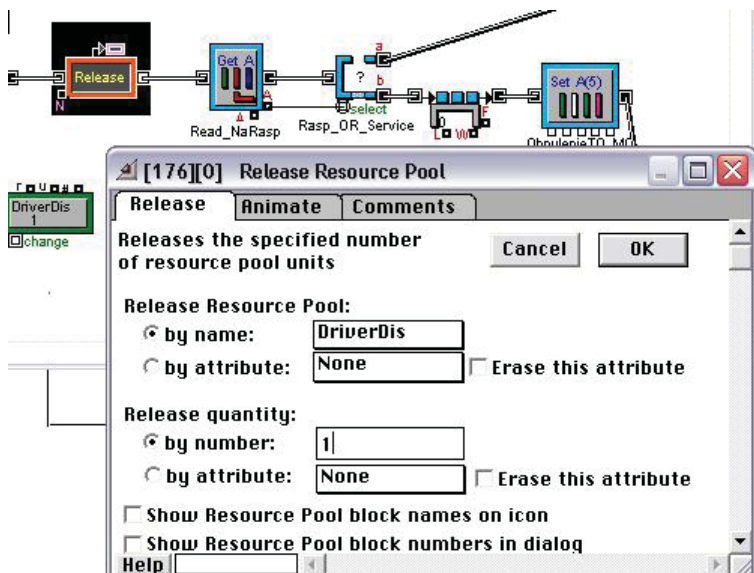


Рис. 4.5.15. Блок *Release, Resource Pool* в составе иерархического блока

В суперблоке *Driver* моделируется перегон водителем автомобилей с распределительной площадки на участки обслуживания и в обратном направлении. Для указания направления движения транзакта в модели используется атрибут *NaRasp*, который создается в блоке, моделирующем оформление заявок *NAZ NACHENIE* (рис. 4.5.9), и первоначально имеет значение 1. Значение атрибута, равное 1, соответствует направлению автомобиля на распределительную площадку, значение, равное 0, – направлению на участки обслуживания.

Значение атрибута *NaRasp* определяется с помощью блока *Get Attribute*, диалог которого представлен на рис. 4.5.16.

Если пришедший транзакт имеет атрибут, равный 1, то он направляется блоком *Select DE Output* на нижний выход суперблока.

В противном случае транзакт направляется на верхний выход, что моделирует перегон автомобиля на участок обслуживания. При этом в случае направления по нижнему выходу транзакт попадает в блок *Set Attribute*, который присваивает значение 0 атрибуту *NaRasp*, что моделирует доставку автомобиля на распределительную площадку.

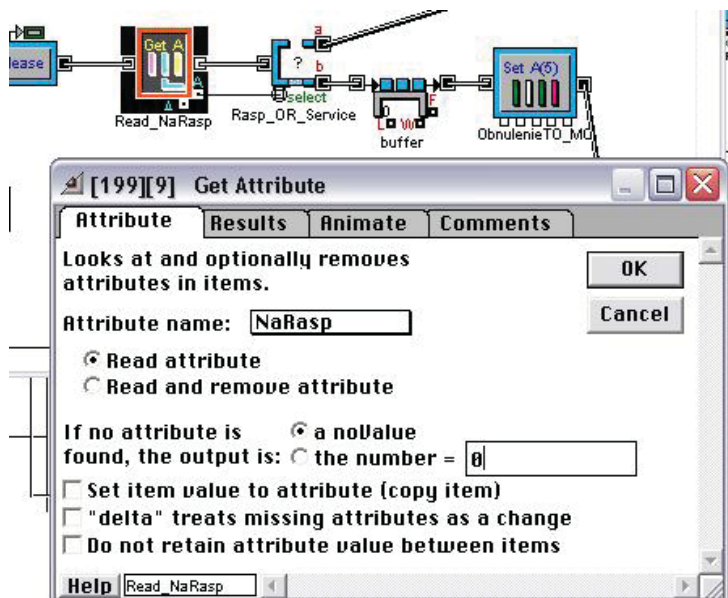


Рис. 4.5.16. Блок *Get Attribute* в составе иерархического блока

Как видно на рис. 4.5.11, в модель внесены некоторые структурные изменения, отличающие её от первого варианта моделирования работы водителя-диспетчера. Модификация модели с помощью блока *Combine* (5) позволяет более логично и наглядно представить маршрут транзактов в модели.

Все потоки транзактов блоком *Combine* (5) направляются в иерархический блок *Driver*, из которого они следуют либо на «распределительную площадку», либо на «участки обслуживания». При этом потребовалось внести изменения и в суперблок *RaspredPlosh*, моделирующий распределительную площадку (рис. 4.5.17).

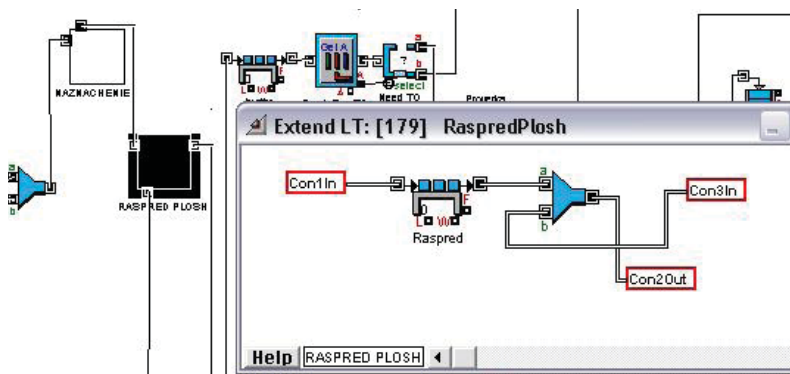


Рис. 4.5.17. Иерархический блок *RaspredPlosh*

Суперблок *RaspredPlosh* стал более компактным и вполне логичным: транзакт входит через два входных коннектора, соответствующих приезду на площадку извне или с участка обслуживания, а выходит через один выходной коннектор, соответствующий поступлению к водителю-диспетчеру.

Результаты моделирования, отображенные плоттером суперблока *Driver* (коэффициент занятости водителя-диспетчера) и плоттером модели (коэффициенты занятости участков обслуживания и количество обслуженных автомобилей), представлены на рис. 4.5.18.

Построенная имитационная модель станции технического обслуживания позволяет имитировать разнообразные станции, различающиеся набором участков обслуживания и порядком приема клиентов, пропускной способностью (количество подъемников) и количеством водителей-диспетчеров, временными характеристиками участков обслуживания.

Как всякая модель, построенная модель СТО не является окончательной и совершенной. Недостатки модели могут быть преодолены на следующих этапах, по мере уточнения структуры реальной системы и статистических данных о её функционировании. Например, моделирование процесса распределения автомобилей с распределительной площадки требует дальнейшего уточнения.

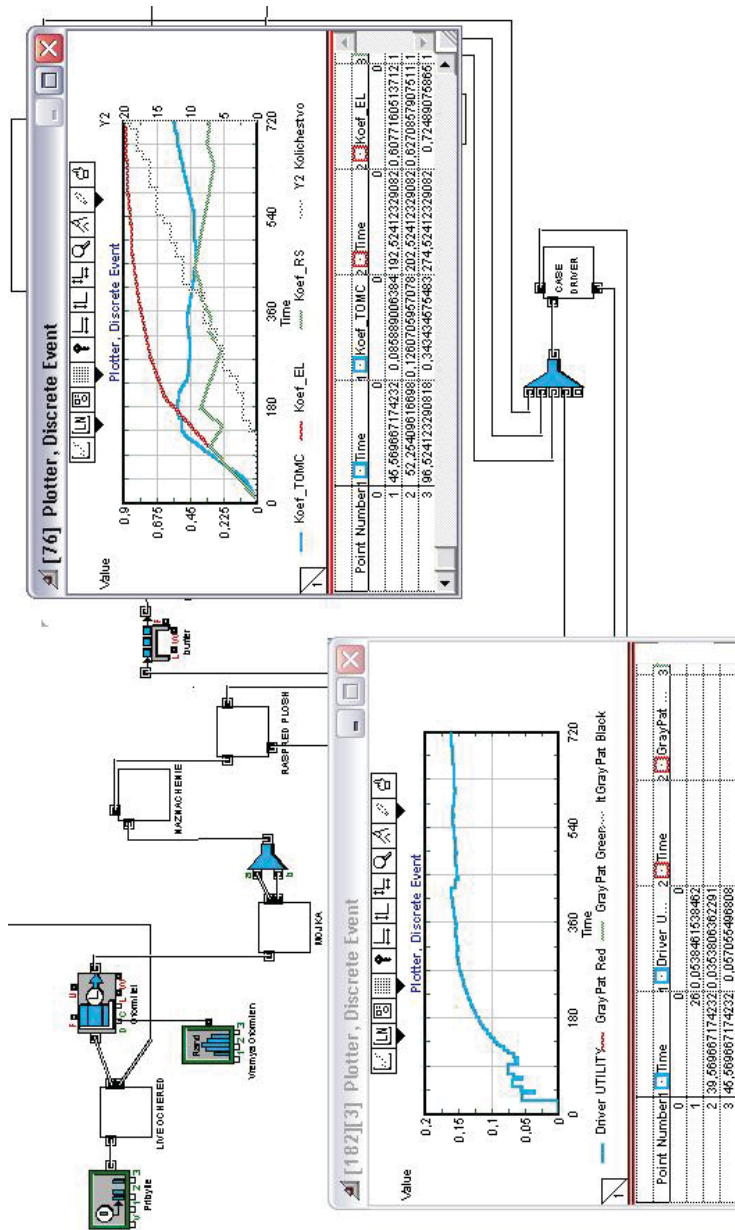


Рис. 4.5.18. Результаты моделирования, отображаемые двумя плоттерами

Действительно, в описанной модели предполагается заданная последовательность проверки наличия заявки на виды обслуживания в следующем порядке: техническое обслуживание (ТО), механический ремонт (МС), ремонт электрооборудования (EL), регулировка развала-схождения (RS). При этом может оказаться, что автомобиль, которому требуется ТО и ремонт электрооборудования, будет поставлен в очередь на участок ТО, хотя в это время будет свободен участок электроремонта. Преодоление указанного недостатка может представлять отдельное задание по моделированию.

Результаты моделирования могут использоваться как для проектирования новой СТО, так и для оптимизации параметров действующих станций, а также для проверки вариантов предлагаемых структурных изменений. Модель позволяет проводить анализ «что-если» и на его основе принимать управленческие решения.

#### **Задания по лабораторным работам 4 и 5**

1. Преобразовать модель к компактному виду, создав иерархические блоки: участок ТО и мехремонта, участок электроремонта и участок регулировки развала/схождения, водитель-диспетчер, приемщик-оформитель.

2. Модифицировать модель в соответствии с предложенными вариантами (табл. 4.5.1. Варианты лабораторных работ) параметров и определить коэффициенты загрузки всех каналов обслуживания.

3. Выявить «узкие места» предприятия по показателям очередей и предложить изменения, позволяющие их расширить.

## Варианты лабораторных работ

Вариант	0	1	2	3	4
Интервал между прибывающими автомобилями	Среднее 25 Экспоненциальное распределение	Среднее 30 Экспоненциальное распределение	Среднее 29 Экспоненциальное распределение	Среднее 33 Экспоненциальное распределение	Среднее 40 Экспоненциальное распределение
Предельная длина очереди	3	4	5	3	5
Доля машин, отправляемых на мойку	40%	50%	60%	80%	90%
Время мойки автомобиля (мин.)	10-20 Равномерное распределение	12-25 Равномерное распределение	Нормальное распределение - 25 Сред. квадр. отклонение - 5	11-22 Равномерное распределение	Треугольное распределение от 15 до 30, среднее 22
Время оформления заказа (мин.)	15-20 Равномерное распределение	12-25 Равномерное распределение	Треугольное распределение от 18 до 28, среднее 24	18-23 Равномерное распределение	Треугольное распределение от 10 до 18, среднее 13
<i>Статистические данные заявки на различные виды обслуживания</i>					
Техническое обслуживание	20%	30%	40%	25%	30%
Механический ремонт	60%	65%	70%	65%	50%
Электроремонт	55%	35%	45%	50%	45%
Регулировка развала/схождения	15%	25%	35%	38%	25%

Окончание табл. 4.5.1

Вариант	0	1	2	3	4
	<i>Время обслуживания на участках (мин.)</i>				
Техническое обслуживание	38-80 Равномерное распределение	Треугольное распределение от 40 до 90, среднее 70	Нормальное распределение Среднее-50 Сред. квадр. отклонение-15	Треугольное распределение от 40 до 90, среднее 70	12-25 Равномерное распределение
Механический ремонт	Треугольное распределение от 40 до 90, среднее 70	Равномерное распределение	Нормальное распределение Среднее-50 Сред. квадр. отклонение-15	40-120 Равномерное распределение	Треугольное распределение от 40 до 90, среднее 70
Электроремонт	40-120 Равномерное распределение	Треугольное распределение от 40 до 90, среднее 70	Нормальное распределение Среднее-50 Сред. квадр. отклонение-15	Треугольное распределение от 40 до 90, среднее 70	12-25 Равномерное распределение
Регулировка вала-схождения	30-50 Равномерное распределение	Нормальное распределение Среднее-50 Сред. квадр. отклонение-15	Треугольное распределение от 40 до 90, среднее 70	40-120 Равномерное распределение	Треугольное распределение от 40 до 90, среднее 70
	<i>Время перегона автомобилей между участками и распределительной площадкой</i>				
	5	6	4	5	7
Время приемки	8	10	9	11	8



## **Лабораторная работа 6**

### **Использование «беспроводных» соединений в моделях Extend LT, создание иерархических блоков, проведение анализа и принятие управленческого решения на его основе**

#### **Исходные условия**

**Объект моделирования:** дистрибьюторская фирма, поставляющая товары различных производителей со своего оптового склада различным получателям.

#### **Описание моделируемой системы**

Потребители вывозят продукцию своим автотранспортом. Транспортные средства представлены большими и малыми грузовиками и легковыми автомобилями (в случае мелких поставок). Получатели приезжают на склад в случайные моменты времени, разделенные интервалами, величина которых имеет случайное распределение со средним значением 8 мин. Автомобиль подъезжает к терминалу для загрузки машин соответствующего типа или становится в очередь к терминалу. Пока загружают товар, водитель-экспедитор оформляет документы у диспетчера.

Статистика показывает, что 15% приезжающих машин являются крупными грузовиками, 55% – средними и 30% – легковым транспортом.

Время загрузки и выезда со склада составляет:

- для крупных грузовиков – от 35 до 50 мин.;
- для малых грузовиков – от 20 до 30 мин.;
- для легковых машин – от 10 до 15 мин.

Случайное время суммарной задержки во всех трех указанных случаях распределено равномерно.

Каждый терминал позволяет загружать одновременно несколько единиц транспорта: крупных грузовиков – 2, малых – 6, легковых машин – 10.

После загрузки и оформления документов транспорт покидает территорию склада. С помощью модели требуется определить коэффициенты занятости терминалов.

Модель строится из 11 типов блоков (в скобках указаны библиотеки, из которых эти блоки берутся):

- 1) *Executive* (DE.LIX)
- 2) *Generator* (DE.LIX)
- 3) *Set Attribute* (DE.LIX)
- 4) *Input Random Number* (GENERIC.LIX)
- 5) *Buffer* (MFG.LIX)
- 6) *Throw* (DE.LIX)
- 7) *Catch* (DE.LIX)
- 8) *Queue, FIFO* (DE.LIX)
- 9) *Activity, Multiple* (DE.LIX)
- 10) *Exit (4)* (DE.LIX)
- 11) *Plotter, Discrete Event* (PLOTTER.LIX)

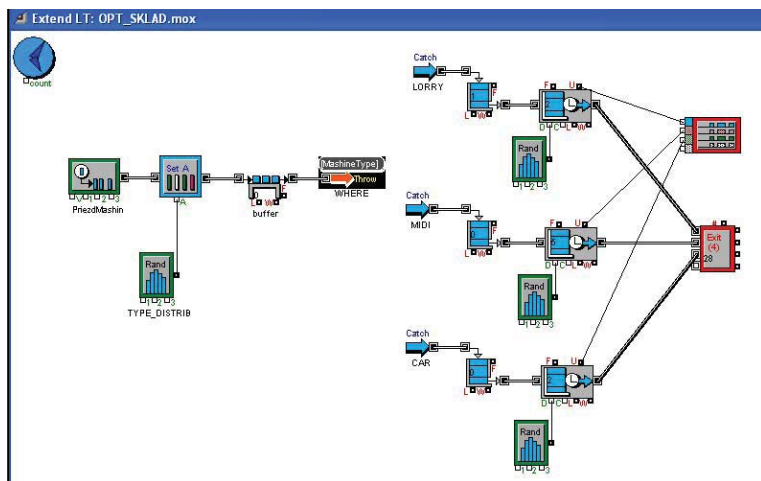


Рис. 4.6.1. Общий вид модели оптового склада

- Блок *Generator* генерирует транзакты. Распределение интервалов времени между порождаемыми транзактами задается экспоненциальным законом со средним значением 8 (в среднем каждые 8 минут на склад прибывают автомобили, причем интенсивность потока примерно постоянна). В модели время исчисляется в целых минутах.

- Блок *Set Attribute* (DE.LIX), представленный на рис. 4.6.2, создает атрибут *MashineType* для вошедшего в него транзакта. Значение атрибута определяет тип транспортной единицы, которую моделирует данный транзакт.

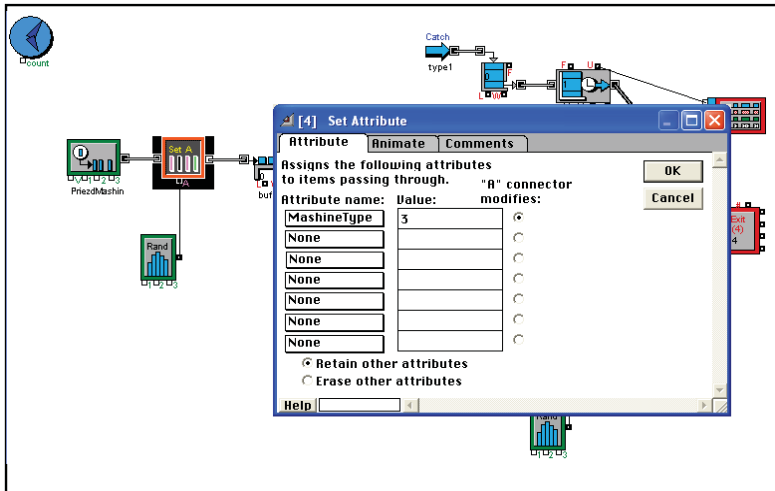


Рис. 4.6.2. Создание атрибута *MashineType*

Имя атрибута при его создании задается в окне, открывающемся при выборе пункта *New Attribute*. Этот пункт, в свою очередь, выбирается из меню, выпадающего при щелчке по полю в столбце *Attribute name* диалога блока.

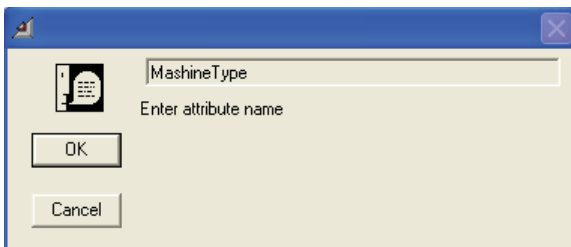


Рис. 4.6.3. Окно задания имени атрибута

Значение атрибута, то есть тип транспорта, задается блоком *Input Random Number* в соответствии с эмпирической таблицей, заданной в описании моделируемой системы.

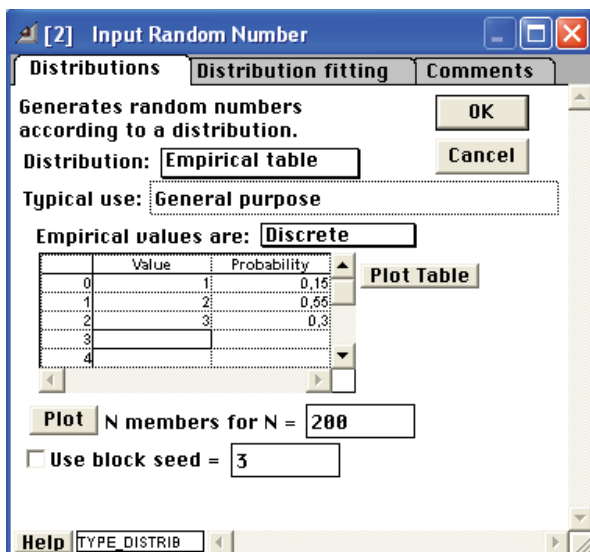


Рис. 4.6.4. Эмпирическая таблица вероятностей значений атрибута *MashineType*

В результате атрибут *MashineType* приобретает значение 1 с вероятностью 0,15, значение 2 – с вероятностью 0,55, значение 3 – с вероятностью 0,3.

- Блок **Buffer** выполняет вспомогательную функцию, обеспечивая промежуточный интервал времени между созданием атрибута и его использованием.

- Новыми в рассматриваемой модели оказываются блоки **Throw** (DE.LIX) и **Catch** (DE.LIX), обеспечивающие «беспроводную передачу транзактов» между блоками.

Блок **Throw** (throw – бросок, бросание) переправляет транзакты в другие части модели, где транзакты «ловятся» блоками **Catch**, словно кэтчерами в бейсболе (catch – ловить). Важно отметить, что блок **Throw** из одного иерархического блока может перебросить транзакты блокам **Catch**, находящимся в других иерархических блоках.

Назначение транзактов производится через соответствие значений атрибутов и номеров блоков **Catch** (рис. 4.6.5).

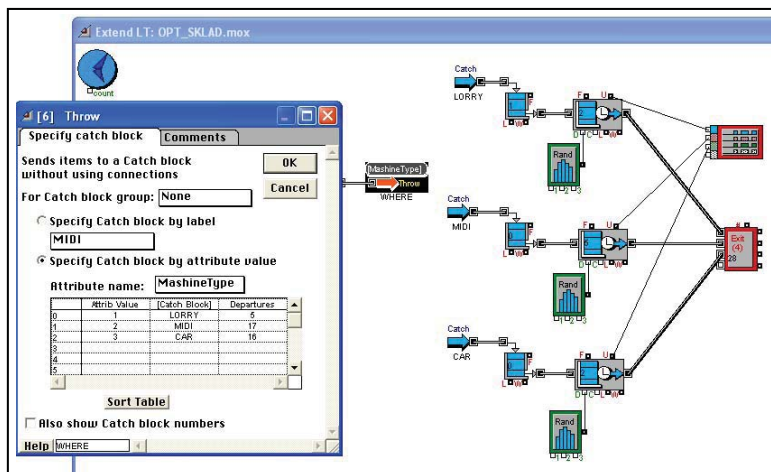


Рис. 4.6.5. Таблица назначений транзактов в зависимости от значения атрибута MashineType

### Настройка блока *Throw*:

- выбирается опция «Specify Catch block by attribute value» (определять блок *Catch* по значению атрибута);
- выбирается имя атрибута из списка, выпадающего при щелчке по полю *Attribute name*;
- в строках таблицы задаются соответствующие друг другу значения атрибута и имена блоков *Catch*.

Блоки *Catch* должны быть созданы и поименованы перед настройкой блока *Throw*. Имена блоков *Catch* задаются в их диалогах в нижнем окне ввода рядом с кнопкой *Help*. При настройке блока *Throw* эти имена выбираются из выпадающего списка, который появляется при нажатии левой клавиши мыши на поле в столбце [Catch Block]. (В диалоге блока *Throw*, показанном на рис. 4.6.5, значению атрибута *MashineType*, равному 1, соответствует блок *Catch* по имени LORRY, значению атрибута *MashineType*, равному 2, – блок *Catch* по имени MIDI, значению атрибута *MashineType*, равному 3, – блок *Catch* по имени CAR.)

Остальные блоки моделируют традиционные части склада: очередь на погрузку (блок *Queue, FIFO*), задержку на погрузку (блок *Activity, Multiple*), задаваемую случайной длительностью (блок *Input Random Number*), как показано на рис. 4.6.5.

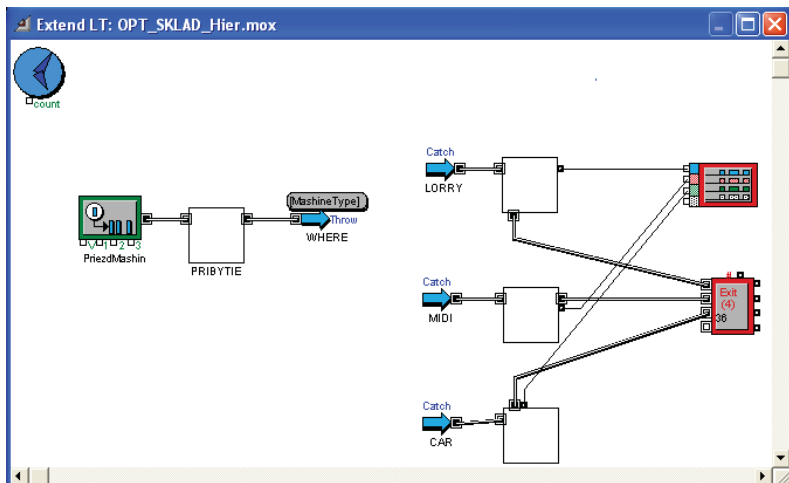


Рис. 4.6.6. Модель оптового склада с иерархическими блоками PRIBYTIE, LORRY, MIDI и CAR

• Компактный вид, представленный на рис. 4.6.6, модель приобретает в результате построения иерархических блоков, моделирующих прибытие трех типов машин с заданной вероятностью (суперблок PRIBYTIE) и работу каждого терминала (суперблоки LORRY, MIDI и CAR).

Блоки *Throw* и *Catch* не включены в иерархические блоки, чтобы оставить видимым механизм передачи транзактов.

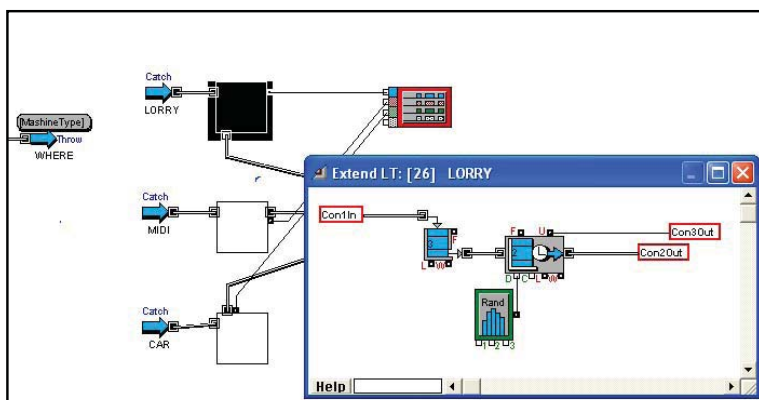


Рис. 4.6.7. Фрагмент модели со структурой иерархического блока LORRY

На рис. 4.6.7 представлен фрагмент модели с открытым иерархическим блоком LORRY, моделирующим пребывание больших грузовиков на складе.

- При запуске модели получают результаты, отображаемые на плоттерах.

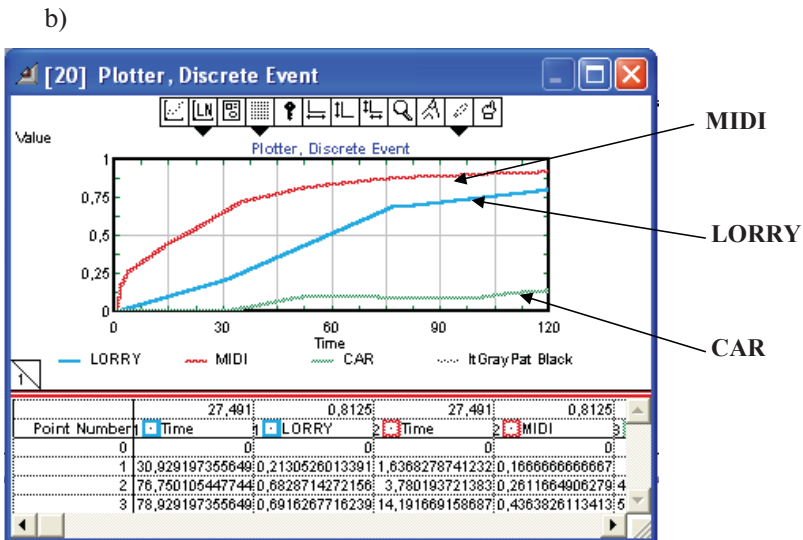
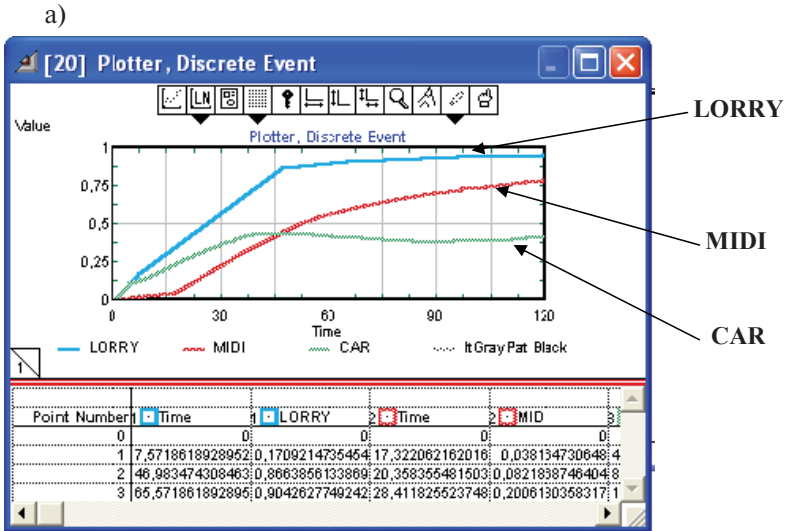


Рис. 4.6.8. Сравнение результатов двух имитационных экспериментов

Графики коэффициентов занятости терминалов, полученные в двух различных экспериментах (запусках имитационной модели), представленные на рис. 4.6.8,а, b, показывают существенный разброс результатов моделирования.

Ввиду этого требуется набрать статистически значимые результаты, для чего запускается многократная имитация. Для сбора результатов многократной имитации в модель вводится блок *Activity Stats* из библиотеки *STATS.LIX*. Перед моделированием в диалоге блока следует установить опцию *Append new updates* (дополнять новые данные в конец списка).

- В пункте *Simulation Setup* задается количество запусков модели: *Numbers of runs* = 50. После запуска произойдет 50-кратный запуск модели. Результаты работы блоков *Activity*, *Multiple* в ходе выполнения всех 50 сеансов моделирования будут собраны в таблицу блока *Activity Stats*. В диалоге блока будет представлена таблица результатов.

- Статистическая обработка полученных результатов для вычисления доверительных интервалов исследуемых параметров выполняется этим же блоком.

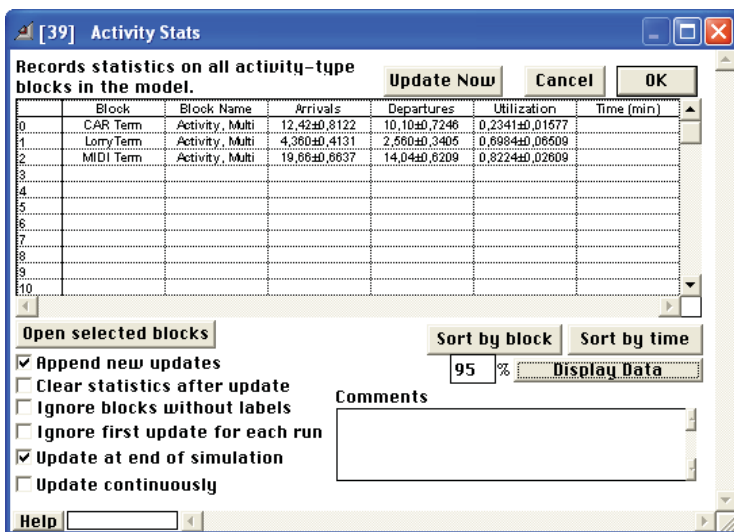


Рис. 4.6.9. Доверительные интервалы характеристик терминалов



95-процентные доверительные интервалы для коэффициентов занятости, вычисленные на основании 50 имитаций, отображаются при нажатии кнопки **Confidence Interval** в диалоге блока *Activity Stats* (рис. 4.6.9).

Коэффициенты занятости терминалов с вероятностью 0,95 падают в следующие интервалы:

LORRY	- от 0,822 – 0,026	до 0,822 + 0,026,
MIDI	- от 0,698 – 0,065	до 0,698 + 0,065,
CAR	- от 0,234 – 0,016	до 0,234 + 0,016.

Таким образом, можно принять решение об уменьшении количества мест и численности персонала на терминале для легковых автомобилей и переводе части грузчиков на терминал для больших грузовиков.

Кроме того, по результатам можно проверить правильность моделирования статистических исходных данных. Из таблицы на рис. 4.6.9 видно, что за промоделированные 2 часа работы через склад прошло следующее количество транспортных единиц:

LORRY	- от 2 до 3, то есть около 10%,
MIDI	- от 14 до 15, то есть около 40%,
CAR	- от 9 до 11, то есть около 50%,

что вполне соответствует заданным параметрам.

В случае необходимости измерения других параметров модели достаточно использовать соответствующие коннекторы блоков, моделирующих очереди и каналы обслуживания. Напомним, что для получения статистических данных по блокам *Queue*, *FIFO*, моделирующим очереди, следует ввести в модель блок *Queue Stats* из библиотеки STATS.LIX.

Варианты лабораторных работ представлены в табл. 4.6.1.

Таблица 4.6.1

## Варианты лабораторных работ (время работы дистрибьюторского центра – 3 часа)

Вариант	Вероятностное распределение интервалов входного потока (мин.)	Тип машины	Параметры распределения времени загрузки
<b>0</b>	Экспоненциальное среднее 18	Грузовик	Равномерное распределение от 20 до 35
		Малый грузовик	Треугольное распределение от 10 до 18, среднее 16
		Легковая	Равномерное распределение от 5 до 10
<b>1</b>	Треугольное распределение от 14 до 20 среднее 16	Грузовик	Треугольное распределение от 25 до 40, среднее 36
		Малый грузовик	Равномерное распределение от 14 до 22
		Легковая	Треугольное распределение от 7 до 12, среднее 9
<b>2</b>	Равномерное распределение от 13 до 22	Грузовик	Равномерное распределение от 20 до 35
		Малый грузовик	Треугольное распределение от 10 до 18, среднее 16
		Легковая	Равномерное распределение от 5 до 10
<b>3</b>	Экспоненциальное среднее 22	Грузовик	Треугольное распределение от 25 до 40, среднее 36
		Малый грузовик	Равномерное распределение от 14 до 22
		Легковая	Треугольное распределение от 7 до 12, среднее 9
<b>4</b>	Равномерное распределение от 12 до 25	Грузовик	Равномерное распределение от 20 до 35
		Малый грузовик	Треугольное распределение от 10 до 18, среднее 16
		Легковая	Равномерное распределение от 5 до 10
<b>5</b>	Треугольное распределение от 17 до 28 среднее 18	Грузовик	Треугольное распределение от 25 до 40, среднее 36
		Малый грузовик	Равномерное распределение от 14 до 22
		Легковая	Треугольное распределение от 7 до 12, среднее 9
<b>6</b>	Равномерное распределение от 18 до 26	Грузовик	Равномерное распределение от 20 до 35
		Малый грузовик	Треугольное распределение от 10 до 18, среднее 16
		Легковая	Равномерное распределение от 5 до 10

Вариант	Вероятностное распределение интервалов входного потока (мин.)	Тип машины	Параметры распределения времени загрузки
<b>7</b>	Экспоненциальное среднее 15	Грузовик	Треугольное распределение от 25 до 40, среднее 36
		Малый грузовик	Равномерное распределение от 14 до 22
		Легковая	Треугольное распределение от 7 до 12, среднее 9
<b>8</b>	Равномерное распределение от 16 до 24	Грузовик	Равномерное распределение от 20 до 35
		Малый грузовик	Треугольное распределение от 10 до 18, среднее 16
		Легковая	Равномерное распределение от 5 до 10
<b>9</b>	Треугольное распределение от 12 до 26 среднее 17	Грузовик	Треугольное распределение от 25 до 40, среднее 36
		Малый грузовик	Равномерное распределение от 14 до 22
		Легковая	Треугольное распределение от 7 до 12, среднее 9

## ЗАКЛЮЧЕНИЕ

Метод имитационного моделирования дискретных событий, проиллюстрированный в представленных главах и лабораторных работах, подчеркивает основы методики моделирования. Они заключаются в том, что следует начинать с простой модели реальной ситуации и постепенно ее усложнять до получения усовершенствованных моделей, более полно отражающих моделируемую ситуацию. При учете в имитационной модели дополнительных факторов реальной жизни полезно использовать иерархические блоки Extend. Создание подмоделей с помощью этих блоков является полезным способом поддержки модульности системы и возможности документирования модели.

Отдельные блоки Extend также позволяют определять стоимостные показатели, предоставляя возможность собирать информацию о совокупных издержках производства. Полная версия Extend обеспечивает оптимизацию модели. Например, оптимизация позволила бы определить количество автомобилей, минимизирующее общие издержки.

Несмотря на ограниченные возможности используемой версии системы Extend, она позволяет подготовить будущих менеджеров и экономистов к самостоятельному моделированию в современных визуальных системах имитационного моделирования, таких как ExtendSim, AnyLogic и им подобных. При этом моделирование представляет собой скорее работу архитектора системы, а не работу программиста, что, в свою очередь, обеспечивает лучшие результаты моделирования и более глубокое понимание закономерностей функционирования исследуемых объектов.

## Библиографический список

1. Уиддет, С. Руководство по компетенциям / С. Уиддет, С. Холлифорд. – М. : ГИППО, 2008. – 240 с.
2. Мур, Дж. Экономическое моделирование в MS Excel / Дж. Мур, Л. Уэдерфорд. – М. : Вильямс, 2004. – 1024 с.
3. Емельянов, А.А. Имитационное моделирование экономических процессов : учеб. пособие/ А.А. Емельянов, Е.А. Власова, Р.В. Дума. – М. : Финансы и статистика, 2002.– 368 с.
4. Шеннон, Р. Имитационное моделирование – искусство и наука / Р. Шеннон. – М. : Мир, 1978. – 418 с.
5. Кобелев, Н.Б. Основы имитационного моделирования сложных экономических систем : учеб. пособие / Н.Б. Кобелев. – М. : Дело, 2003. – 336 с.
6. Карпов, Ю. Имитационное моделирование систем. Введение в моделирование с AnyLogic 5 / Ю. Карпов. – СПб. : БХВ-Петербург, 2005. – 400 с.
7. Томашевский, В.Н. Имитационное моделирование в среде GPSS / В.Н. Томашевский, Е.Г. Жданова. – М. : Бестселлер, 2003. – 416 с.
8. Кельтон, Д. Имитационное моделирование. Классика CS / Д. Кельтон, А. Лоу. – СПб. : Питер ; Киев : Издательская группа ВНУ, 2004. – 847 с.
9. Бочкарев, А.А. Планирование и моделирование цепи поставок : учебно-практическое пособие / А.А. Бочкарев. – М. : Альфа-Пресс, 2008. – 192 с.

## Содержание

Введение .....	3
1. ИМИТАЦИОННОЕ МОДЕЛИРОВАНИЕ КАК ИНСТРУМЕНТ АНАЛИЗА СЛОЖНЫХ СИСТЕМ .....	4
2. EXTEND LT. МОДЕЛИРОВАНИЕ КАК КОНСТРУИРОВАНИЕ .....	6
3. ОСНОВЫ МОДЕЛИРОВАНИЯ В EXTEND .....	10
4. ЛАБОРАТОРНЫЙ ПРАКТИКУМ .....	26
Лабораторная работа 1. Создание имитационной модели средствами Extend LT, проведение анализа и принятие управленческого решения на его основе .....	26
Лабораторная работа 2. Создание имитационной модели средствами Extend LT, проведение статистического анализа, оптимизация параметров .....	41
Лабораторная работа 3. Совершенствование имитационной модели предприятия в Extend LT, создание иерархического блока, проведение анализа и принятие управленческого решения на его основе .....	52
Лабораторная работа 4. Разработка имитационной модели станции технического обслуживания в Extend LT, создание иерархического блока, проведение анализа и принятие управленческого решения на его основе .....	71
Лабораторная работа 5. Совершенствование имитационной модели станции технического обслуживания в Extend LT, создание иерархических блоков, моделирование работы водителя-диспетчера .....	89

Лабораторная работа 6. Использование «беспроводных» соединений в моделях Extend LT, создание иерархических блоков, проведение анализа и принятие управленческого решения на его основе .....	108
Заключение .....	119
Библиографический список .....	120

Учебное издание

*Ярыгин Олег Николаевич*

ИМИТАЦИОННОЕ МОДЕЛИРОВАНИЕ ДЕЯТЕЛЬНОСТИ  
ПРЕДПРИЯТИЯ В EXTEND LT

Учебно-методическое пособие

Редактор *Т.Д. Савенкова*

Техническое редактирование и вёрстка: *Л.В. Сызганцева*

Дизайн обложки: *Г.В. Карасева*

Подписано в печать 03.05.2012. Формат 60×84/16.

Печать оперативная. Усл. п. л. 7,15.

Тираж 100 экз. Заказ № 1-48-11.

Издательство Тольяттинского государственного университета  
445667, г. Тольятти, ул. Белорусская, 14



