RAO-studio

Руководство пользователя

Содержит описание принципов дискретного имитационного моделирования, основные положения РДО-метода, работы с инструментальной средой RAO-studio и описание модели простейшей парикмахерской

2007

rdo.rk9.bmstu.ru

11.10.2007

Оглавление

[Общие сведения 2](#_Toc211012245)

[Назначение программного комплекса RAO-studio 3](#_Toc211012246)

[Дискретное имитационное моделирование 8](#_Toc211012247)

[Принцип функционирования 8](#_Toc211012248)

[Процессно-ориентированный подход 10](#_Toc211012249)

[Событийный подход 11](#_Toc211012250)

[Подход сканирования активностей 12](#_Toc211012251)

[Использование подходов 13](#_Toc211012252)

[РДО-метод 14](#_Toc211012253)

[Основные понятия 14](#_Toc211012254)

[Событие 16](#_Toc211012255)

[Действие 17](#_Toc211012256)

[Продукционное правило 19](#_Toc211012257)

[Запуск программного комплекса 21](#_Toc211012258)

[Создание модели 24](#_Toc211012259)

[Редактирование модели 27](#_Toc211012260)

[Редактирование текста модели 27](#_Toc211012261)

[Автозавершение 30](#_Toc211012262)

[Шаблоны и вставка синтаксических конструкций 30](#_Toc211012263)

[Поиск фразы по всей модели 32](#_Toc211012264)

[Навигация с помощью закладок 33](#_Toc211012265)

[Компиляция и запуск модели 34](#_Toc211012266)

[Режимы моделирования 37](#_Toc211012267)

[Результаты моделирования 40](#_Toc211012268)

[Построение графиков 41](#_Toc211012269)

[Модель простейшей парикмахерской 43](#_Toc211012270)

[Работа с моделью 43](#_Toc211012271)

[Описание модели 50](#_Toc211012272)

# Общие сведения

Программный комплекс RAO-studio предназначен для разработки и отладки имитационных моделей на [языке РДО](ms-its:RAO-language.chm::/html/rdo_intro.htm). Основные цели данного комплекса - обеспечение пользователя легким в обращении, но достаточно мощным средством разработки текстов моделей на языке РДО, обладающим большинством функций по работе с текстами программ, характерных для сред программирования, а также средствами проведения и обработки результатов имитационных экспериментов.

В соответствии с основной целью программный комплекс решает следующие задачи:

* синтаксический разбор текста модели и настраиваемая подсветка [синтаксических конструкций языка РДО](ms-its:RAO-language.chm::/html/rdo_base_intro.htm);
* открытие и сохранение [моделей](ms-its:RAO-language.chm::/html/rdo_terms.htm#model);
* расширенные возможности для редактирования текстов моделей;
* автоматическое завершение ключевых слов языка;
* поиск и замена фрагментов текста внутри одного модуля модели;
* поиск интересующего фрагмента текста по всей модели;
* навигация по тексту моделей с помощью закладок;
* наличие нескольких буферов обмена для хранения фрагментов текста;
* вставка синтаксических конструкций языка и заготовок (шаблонов) для написания элементов модели;
* настройка отображения текста моделей, в т.ч. скрытие фрагментов текста и масштабирование;
* запуск и остановка процесса моделирования;
* изменение режима моделирования;
* изменение скорости работающей модели;
* переключение между кадрами анимации в процессе моделирования;
* отображение хода работы модели в режиме реального времени;
* построение графиков изменения интересующих разработчика характеристик в режиме реального времени;
* обработка синтаксических ошибок при запуске процесса моделирования;
* обработка ошибок во время выполнения модели;
* обеспечение пользователя справочной информацией.

Программный комплекс состоит из двух частей:

* среды разработки (RAO-studio.exe);
* файлов справок (RAO-language.chm - справка по языку РДО, RAO-studio.chm - справка по программному комплексу).

Для обеспечения работы программного комплекса необходимо, чтобы модуль среды разработки и файлы справок находились в одной директории на диске. Других ограничений нет. Пользователь может сам настроить и расположить, например, на Рабочем столе, ярлык для быстрого запуска программного комплекса.

# Назначение программного комплекса RAO-studio

Задачи системного анализа и синтеза объектов различной природы и назначения часто решаются с использованием имитационных моделей. Эти модели позволяют исследовать динамические аспекты поведения сложных дискретных систем и процессов. Имитация в частности позволяет выполнить анализ функционирования объекта, прогнозирование, организационное управление, поддержать принятие решений при проектировании и управлении.

RAO-studio является средством имитационного моделирования, позволяющим воспроизводить на ЭВМ динамику объекта, принятие решений сложной системой управления, и даже моделировать деятельность человека при принятии решений. В основе имитатора лежит РДО-метод формализации знаний о дискретных системах и процессах. Знания представляются в форме модифицированных продукционных правил. При этом сохраняются такие достоинства продукционных систем, как универсальность, гибкость и наличие формальных механизмов логического вывода. Традиционные продукционные правила являются частным случаем модифицированных, поэтому в имитационную модель легко могут быть включены, например, экспертные системы.

Язык описания объектов, алгоритмов управления и задач в RAO-studio это по существу язык представления знаний. Он требует от пользователя лишь знаний в предметной области, а не в программировании. Пользователь описывает ресурсы, правила функционирования, требуемые показатели и анимационные кадры непосредственно в терминах предметной области, не прибегая при этом к представлению своей системы в терминах какого-либо известного метода (системы очередей, сети Петри, автоматы) или языка типа SLAM-II, ARENA, SIMPLE++ и других. Это резко повышает гибкость, мощность и наглядность. РДО - язык высокого уровня, использующий символические имена, арифметические и логические выражения и функции, генераторы псевдослучайных чисел, модифицированные и простые продукции.

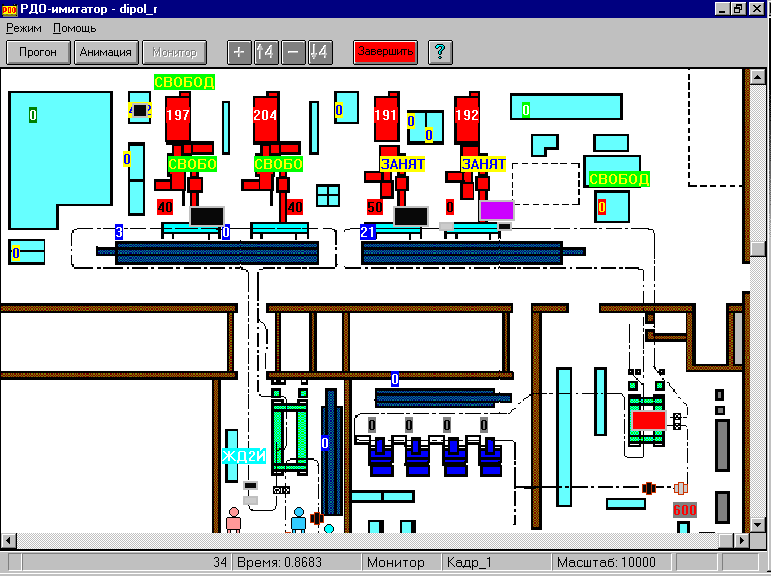
Основные элементы RAO-studio - это модифицированная продукционная система и аппарат событий. Действия инициируются системой вывода, а нерегулярные события специальным блоком. При имитации состояние системы изменяется в соответствии с описанием нерегулярного события либо действия, которое началось или завершилось. После любого изменения состояния, т.е. при каждом событии, вызывается система вывода. Она просматривает в базе знаний продукционные правила и проверяет по предусловиям, могут ли начаться какие-либо действия. При нахождении таких действий инициируются события их начала.

Продукционная система, блок имитации нерегулярных событий и аппарат ведения событий совместно осуществляют построение имитационной модели процесса. На основании анализа результатов имитации вычисляются требуемые показатели функционирования системы. Система трассировки выводит подробную информацию о событиях в специальный файл, который затем обрабатывается для детального анализа процесса. Система анимации отображает на экране во время имитации поведение моделируемого объекта.

RAO-studio может быть применен для создания имитационных моделей, систем планирования, игр и тренажеров, экспертных систем реального времени и гибридных систем, включающие экспертные системы, имитационные модели и алгоритмы оптимизации. Непрерывные процессы также могут быть описаны, поскольку формулы интегрирования переменных состояния можно записать в виде модифицированных продукций.

С использованием РДО были проведены исследования ряда объектов различного назначения. Среди моделей - линия разлива пищевых жидкостей фирмы Перье (Франция), участок разделки бревен фирмы Хейдельберг (Германия), участок механообработки завода экспериментального машиностроения г. Протвино, участок сборки МНТК Робот и ряд других. Модели используются для исследования как самих производственных процессов, так и построения систем управления ими.

#### Исследование проекта гибкого автоматизированного участка



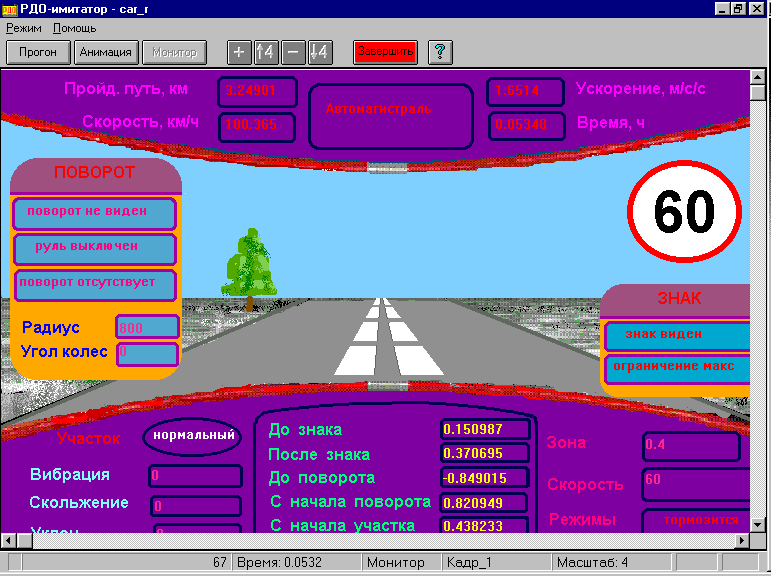
Гибкий автоматизированный участок предназначен для изготовления деталей сверхпроводящих магнитов. Участок включает более 70 единиц различного оборудования, в том числе прессы листовой штамповки, склады, моечные и галтовочную машины, позиции сборки, монорельсовые дороги.

Задача: оценить технический проект участка, определив пропускную способность, узкие места, интенсивность транспортных потоков и другие характеристики.

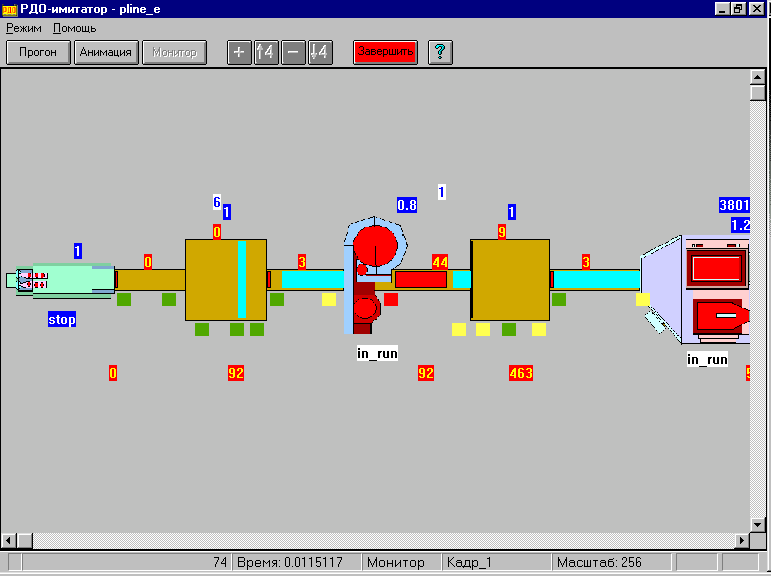
Применение РДО позволило вскрыть ряд недостатков в разработанном проекте, и выработать конкретные рекомендации по изменению проекта. Проект разработан в Институте Физики Высоких Энергий в рамках программы создания ускорительно-накопительного комплекса заряженных частиц на энергию 3000 ГэВ.

#### Экспертная система управления автомобилем

Управление автомобилем, как известно сложная и тяжелая работа. Водитель должен быстро принимать множество решений в подчас чрезвычайно сложной и быстро меняющейся обстановке. Правильные решения может принять лишь водитель, обладающий большим опытом, знаниями и автоматизмом поведения в сложных ситуациях. Помочь принять решения водителю может экспертная система, содержащая знания специалистов и всегда готовая подсказать решение водителю, там, где его опыта не хватает.

Задача: определить состав и объем знаний для принятия решений, выбрать способы их приобретения, хранения и использования. С помощью РДО разработана сложная система, включающая имитационные модели дороги, движения автомобиля и экспертную систему для принятия решений. В экспертную систему были заложены знания опытного водителя. Результаты показали принципиальную возможность реализации подобных систем.

#### Разработка алгоритмов управления линии разлива жидкостей

Линия разлива пищевых жидкостей в бутылки состоит из рабочих станций, производящих или контролирующих продукты (мойка, наполнение, упаковка), связанных между собой многоленточными аккумулирующими конвейерами. Основная цель управления - это балансировка производственных потоков и емкости буферных накопителей. Система управления линией должна воспринимать и обрабатывать информацию о текущем состоянии машин, датчиков и конвейеров и в зависимости от ситуации менять скорость машин и конвейеров.

Задача: разработать алгоритмы управления линией разлива, обеспечивающие максимальную производительность.

Разработанная модель позволила испытать различные алгоритмы управления линией разлива фирмы Перье (Франция) и оценить альтернативные способы размещения датчиков, контролирующих поток бутылок на линии. Работа проводилась в лаборатории разработки промышленного программного обеспечения (LGIL) Института науки и техники (ISTV) Университета г.Валансьенн и ЭноКамбрези (UVHC) (Франция).

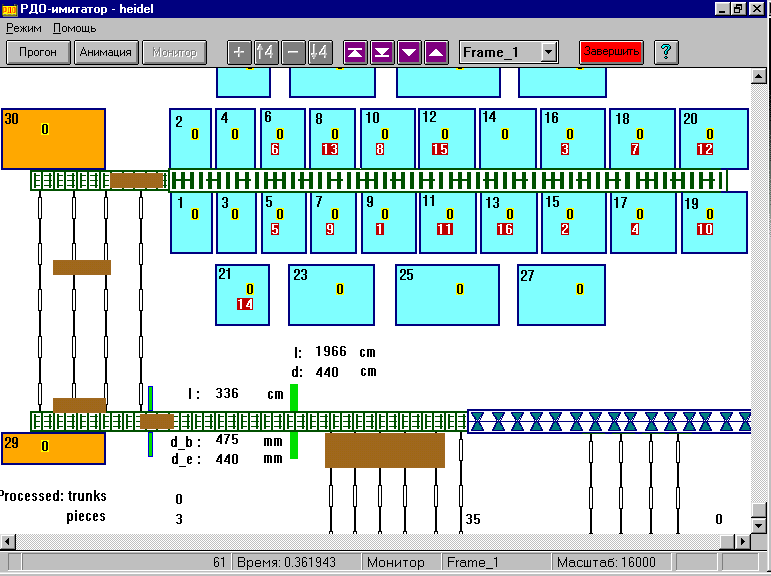
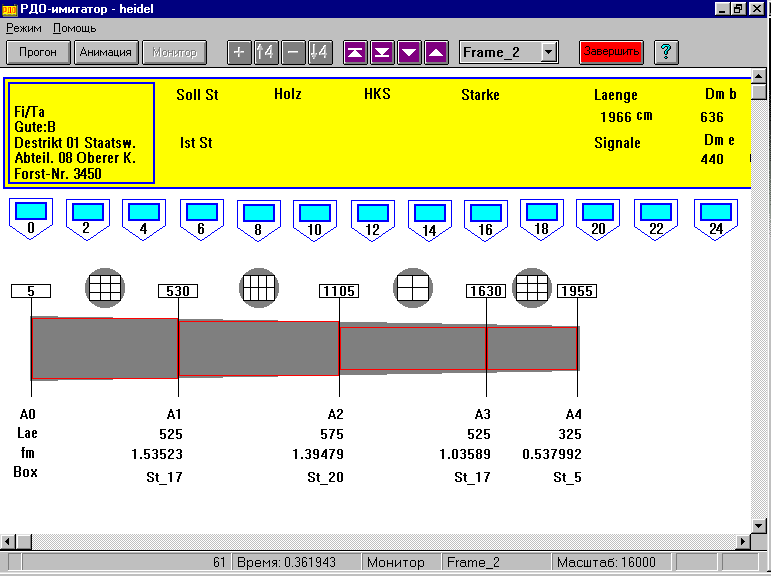
#### Планирование работы участка мебельной фабрики

Важнейшее требование к производству мебели - быстрое выполнение индивидуальных заказов на мебель. Производство мебели из массового все более становится индивидуальным. Заказы отличаются не только составом, но и цветом отделки. Рассматриваемая линия изготовляет комплекты деталей под имеющийся портфель заказов.

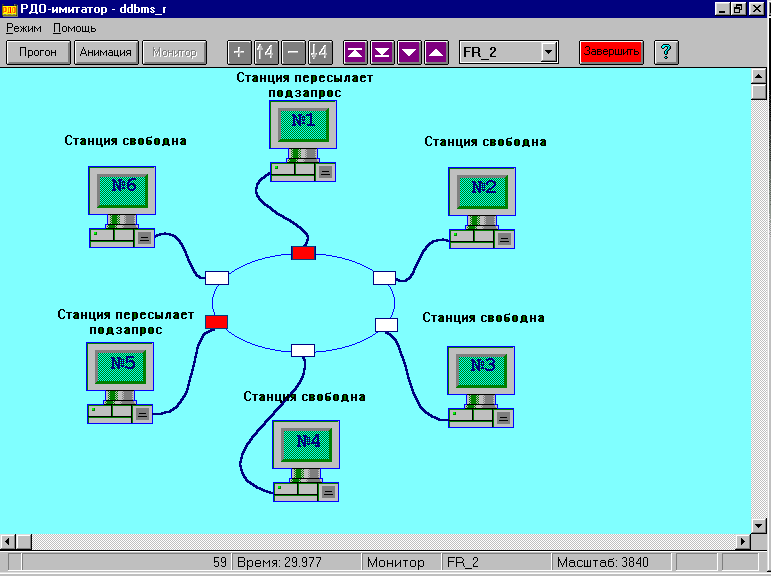
Задача: обеспечить своевременное и быстрое изготовление отдельных заказов, уменьшить переналадки оборудования. Для этой цели необходимо эффективное планирование работы участка и управление оборудованием.

Имитационная модель позволила исследовать различные алгоритмы планирования и управления выпуском деталей, обеспечить комплектацию заказов на выходе линии. Кроме того получены оценки максимальной производительности линии и загрузки отдельных ее элементов.

#### Оптимизация раскроя бревен на балки и брусья

Для минимизации отходов древесины при разделке хлыстов на балки и брусья необходимо разработать алгоритм оптимального раскроя бревна. Этот алгоритм по результатам обмера очередного бревна и по имеющемуся портфелю заказов определяет места распилов и вписывание в куски бревна балок и брусьев. Так как при раскрое необходимо учитывать плановые сроки изготовления заказов и запаздывание самой программы раскроя, свойства алгоритма раскроя не могут быть определены без имитации процесса функционирования участка распилки. Разработанная модель и различные варианты алгоритмов раскроя (эвристические, генетические, поиска решений в продукционной системе) позволили обеспечить рациональный раскрой. Работа выполнялась для предприятия фирмы Heidelberger Holzverarbeitung GmbH (ФРГ).

#### Проектирование распределенной базы данных



Распределённая база данных (РБД) представляет собой базу данных, отдельные части которой размещены на нескольких ЭВМ локальной вычислительной сети. Система управления РБД обеспечивает определение ЭВМ, на которой хранятся требуемые в запросе данные; декомпозицию распределённых запросов на подзапросы; планирование обработки запросов; передачу частных подзапросов и их исполнение на удалённых ЭВМ и многое другое.

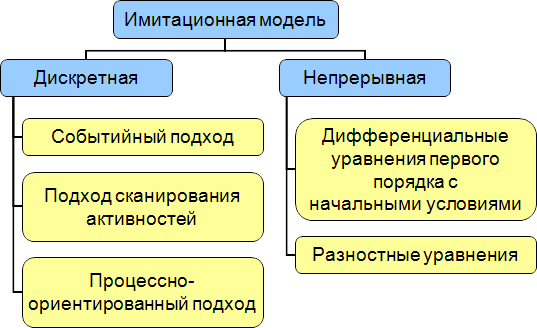
Задача: сократить время обработки сложных распределённых запросов, выполняющих ряд операций над данными на различных ЭВМ сети, за счет эффективного размещения файлов и эффективного распределения этапов процесса обработки данных между ЭВМ сети.

Имитируется появление сложных запросов, планируется процесс обработки запросов, их декомпозиция на подзапросы, передача данных по сети, обработка на ЭВМ, поиск и чтение данных из РБД, композиция общего результата, обеспечение целостности РБД, выявление тупиковых ситуаций. Определяются основные характеристики системы управления РБД.

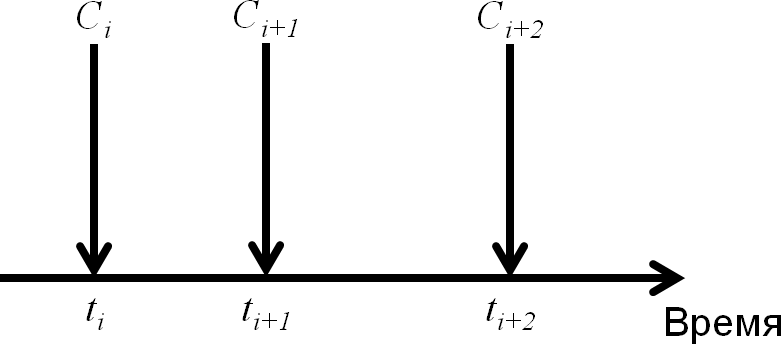
# Дискретное имитационное моделирование

## Принцип функционирования

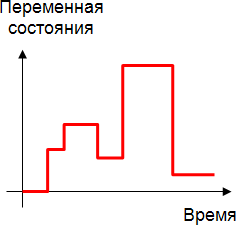
Алан Прицкер в своей книге “Введение в имитационное моделирование и язык СЛАМ II”[[1]](#footnote-2) рассматривает разные подходы к имитационному моделированию, которые делятся им на дискретные и непрерывные:



Рассмотрим более подробно дискретное моделирование. В нём состояние модели изменяют события, которые происходят в дискретные моменты времени **t1, t2,…, tn**, при этом, состояние системы на интервале **[ti, ti+1)** остается неизменным. Само состояние определяется набором переменных состояния, по совокупности которых и можно судить о текущем состоянии модели на интервале **[ti, ti+1)**. Эти переменные можно сравнить с фотографией модели, а сам процесс дискретного моделирования с кинофильмом, в котором последовательно прокручиваются (рассчитываются) состояния-кадры. События не имеют длительности, т.е. мгновенно изменяют состояние модели, с точки зрения модельного времени.



Событие можно представить как подпрограмму, написанную на специальном или алгоритмическом языке программирования. Система имитационного моделирования занимается продвижением модельного времени и вызовом соответствующих событий. От разработчика модели требуется описать изменения, которые производят события с переменными состояния модели. Из-за дискретности график изменения переменной состояния имеет вид:



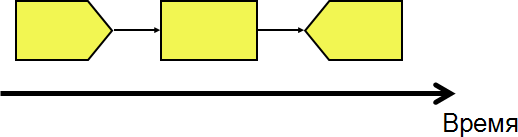
При описании модели сложной дискретной системы (СДС) можно выделить модель объекта, которая описывает процесс преобразования вектора входных воздействий **X** в реакцию системы **Y**. Если в качестве примера системы выбрать производственный участок, то под вектором входных воздействий можно подразумевать заготовки, а под реакцией – готовые изделия. Процесс обслуживания является управляемым. За управление отвечает система управления. Как правило, эта часть имитационной модели состоит их правил продукции (конструкции типа **если…то…**), которые не имеют длительности, но позволяют гибко описать логику управления. И наконец, в модели СДС можно выделить взаимодействие с внешней средой. Оно осуществляется, во-первых, через вектор входных воздействий **X**, во-вторых – вектор внешних возмущений **E**, который описывает возникновение внештатных ситуаций, например, отказы оборудования.



Рассмотрим предложенные А. Прицкером подходы дискретного моделирования относительно предложенной схемы имитационной модели СДС.

## Процессно-ориентированный подход

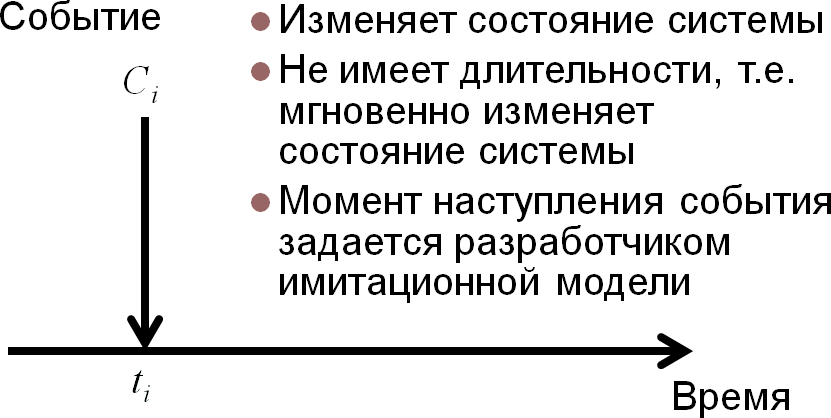
Очень распространенный и часто используемый подход. Лучше всего подходит для описания модели процесса. Он предлагает простую и наглядную логику описания связанных операций, из которых создается процесс обслуживания. С формированием вектора входных воздействий у него тоже всё в порядке. Но процессно-ориентированный подход имеет ограниченные возможности описания внешний возмущений и модели системы управления. В некоторых системах процессно-ориентированного имитационного моделирования встроена поддержка моделирования отказов оборудования, что отчасти снимает потребность в векторе **E**, но встает вопрос моделирования других внештатных ситуаций. Тоже относится и к модели системы управления. Если необходимо смоделировать жесткую систему управления процессом, то возможностей этого подхода вполне достаточно: процесс состоит из связанных операций, можно настраивать логику дисциплин очередей и отказы оборудования. Но за рамки этих возможностей выйти бывает проблематично, например, смоделировать процессы с не жесткими связями между операциями, или логику принятия решений с использованием генетических алгоритмов или нейронных сетей. Для последних задач процессно-ориентированных подход просто не предназначен, и необходимо использовать гибридные системы, где имитационная модель будет одним из элементов. Он предназначен для моделирования систем массового обслуживания (СМО). Разработку моделей можно сравнить с конструированием домиков Lego из готовых блоков. Некоторые системы моделировании позволяют разрабатывать новые блоки, что увеличивает гибкость подхода.



Алгоритм работы основан на предположении, что имеются заявки на обслуживание (часто называются транзактами), которые получает его, перемещаясь в модели от одного блока (оператора) к другому. Одновременно может быть выполняться несколько блоков, если в них есть транзакты. Еще выделают такие элементы модели, как ресурсы. Они используются для обслуживая транзактов. Примеры ресурсов: станок, кассовый аппарат и т.п.

## Событийный подход

В событийном подходе разработчику предлагается описать поведение системы не как набор взаимосвязанных операций, а в виде отдельных событий. Подход очень трудоемок, но позволяет добиться очень большой гибкости в описании процесса функционирования. Плохо описывает систему управления.

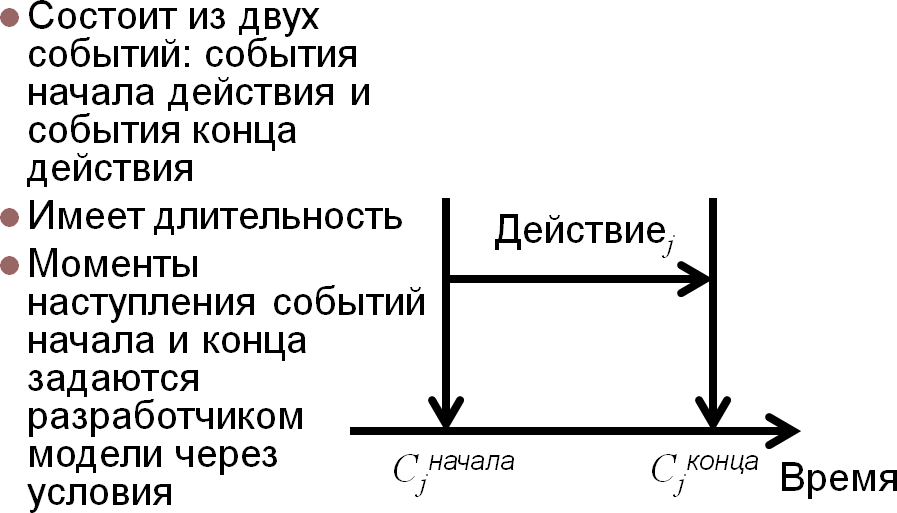


Алгоритм работы приведен ниже.

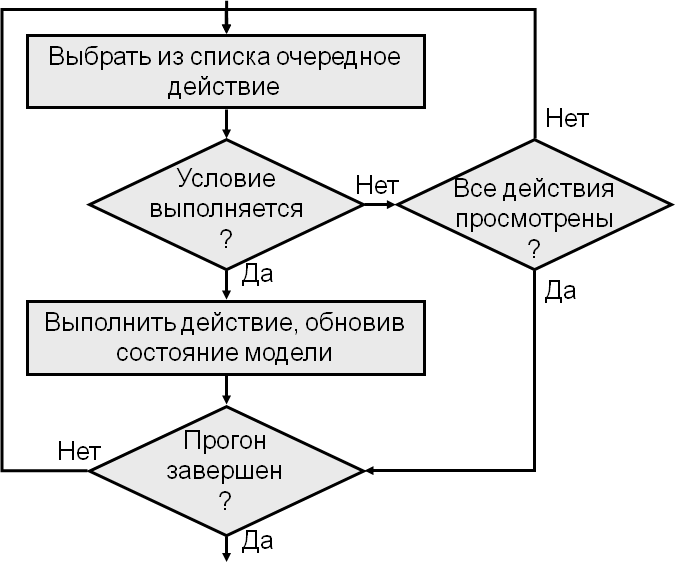


## Подход сканирования активностей

В подходе сканирования активностей (от англ. activity, действие) разработчику предлагается описать поведение системы как набор действий. Действие, в отличие от события, имеет длительность. Причем, в отличие от событийного процессно-ориентированного подхода, моменты наступления действий задаются не через выражение времени, а через логические выражения. Это очень удобно, когда заранее неизвестен момент начала, но понятно его условие. Тоже относится и к определению длительности. Подход довольно трудоемок, но позволяет добиться очень большой гибкости в описании процессов функционирования, процессов реагирования на внешние воздействия и работу системы управления. Если требуется, то можно описать последовательность действий, как и в процессном подходе, но из-за отсутствия транзакта как элемента модели, жесткую связь между действиями необходимо прописывать самостоятельно через дополнительные условия, что усложняет процесс разработки. Также с его помощью неудобно описывать взаимодействие с внешней средой. Подход довольно ресурсно-затратный, т.к. необходимо постоянно проверять условия выполнения действий.

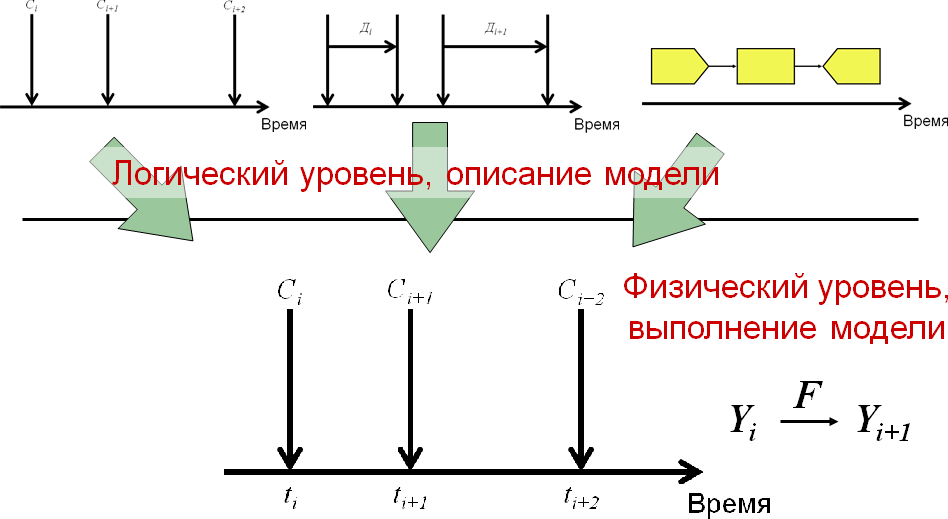


Алгоритм работы приведен ниже.



## Использование подходов

Приведенные выше подходы имеют отношение только к способу описания разработчиком процесса функционирования СДС: или через события, или через действия, или через набор процессов. В системе имитационного моделирования эти понятия переводятся в события, которые в дискретные моменты времени изменяют состояние модели, а саму модель можно определить как оператор преобразования **F**, с помощью которого рассчитывается новое состояние модели, на основании предыдущего.

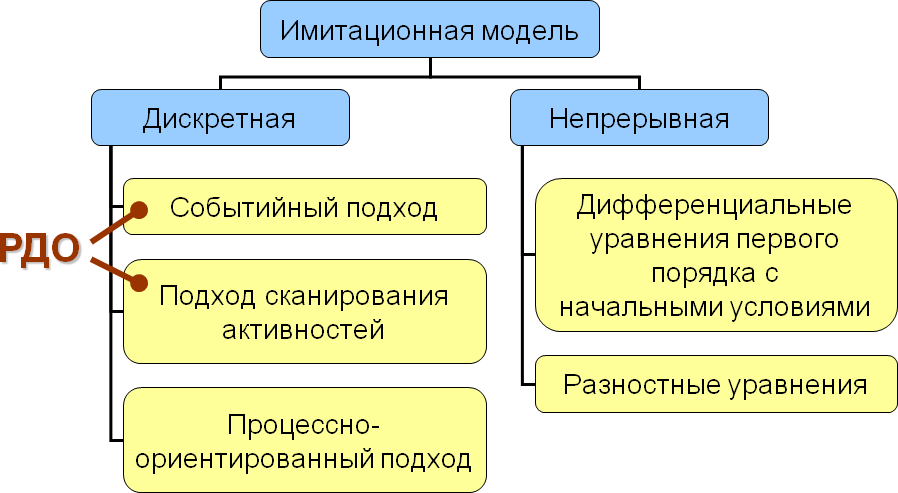


Каждый из походов может показать свои сильные стороны при моделировании различных аспектов СДС. Если СДС относится к классу СМО, то будет достаточно процессно-ориентированного подхода для её моделирования. Но если по каким-либо причинам систему не удается представить как СМО, то её придется моделировать, используя два других подхода. При этом выигрывают те инструментальные средства разработки имитационных моделей, которые позволяют в рамках одной модели использовать различные подходы, что позволяет разные её части описывать наиболее адекватным ситуации подходом.

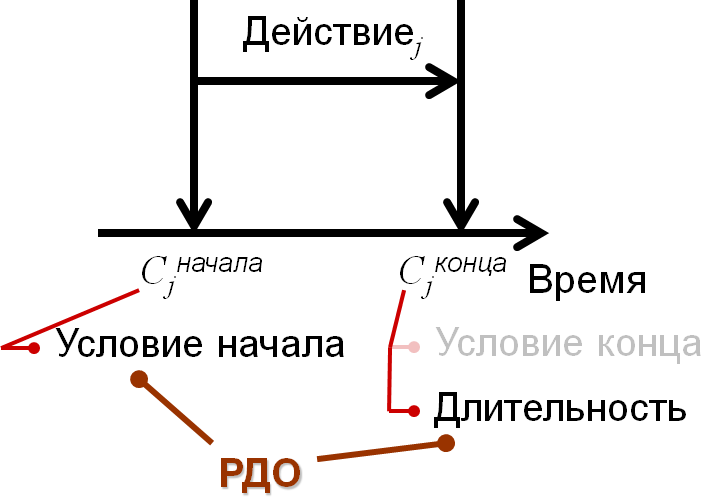
# РДО-метод

## Основные понятия

RAO-studio является инструментальным средством разработки имитационных моделей на языке РДО. В основе языка РДО лежит РДО-метод, который подробно описан в книге Емельянов В.В., Ясиновский С.И. “Введение в интеллектуальное имитационное моделирование. Язык РДО.”[[2]](#footnote-3) Рассмотрим отличия метода от подходов, предложенных А. Прицкером.

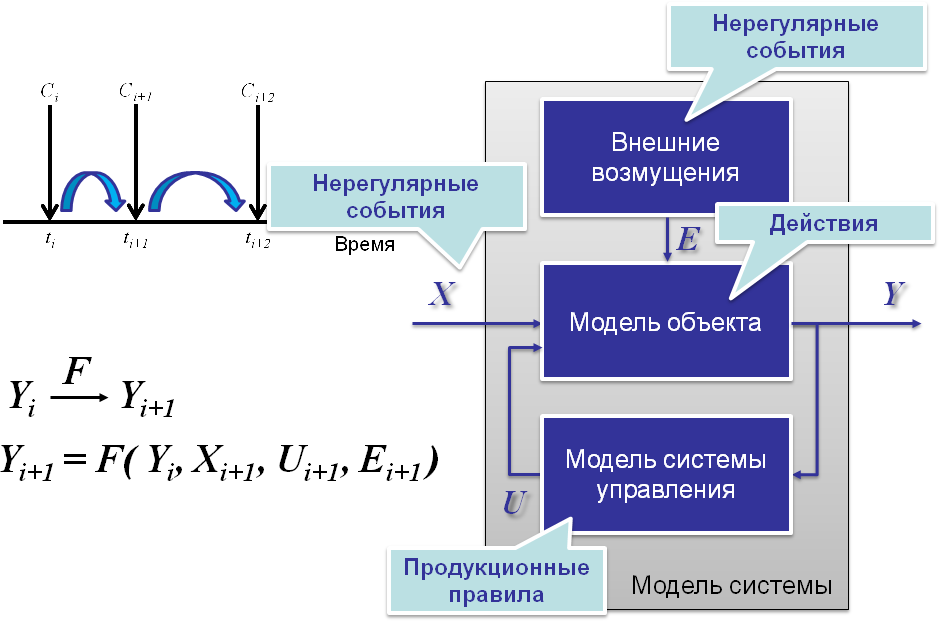


В РДО есть поддержка событийного подхода и подхода сканирования активностей, но последний имеет небольшое отличие: событие конца действия задается не через условие, а через длительность.



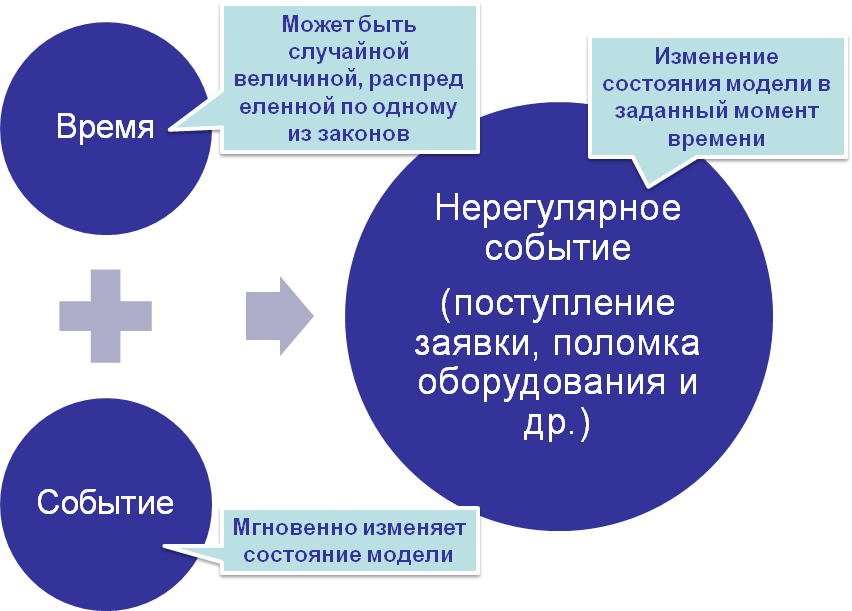
Помимо понятий событие и действия, в РДО есть возможность описать продукционное правило. Оно также как событие не имеет длительности и мгновенно изменяет состояние системы, но запускается как действие через условие. С помощью продукционных правил довольно легко можно описывать сложные системы управления. Все эти три конструкции могут взаимодействовать и обмениваться информацией через общие ресурсы модели, которые описывают объекты моделируемой системы. Каждый ресурс состоит из набора параметров. Совокупное состояние всех параметров всех ресурсов определяет состояние модели. К достоинствам РДО-метода стоит отнести гибкость описания процессов функционирования СДС и системы управления. К недостаткам: отсутствие процессно-ориентированного описание, что приводит к усложнению написания модели объекта через действия.





## Событие

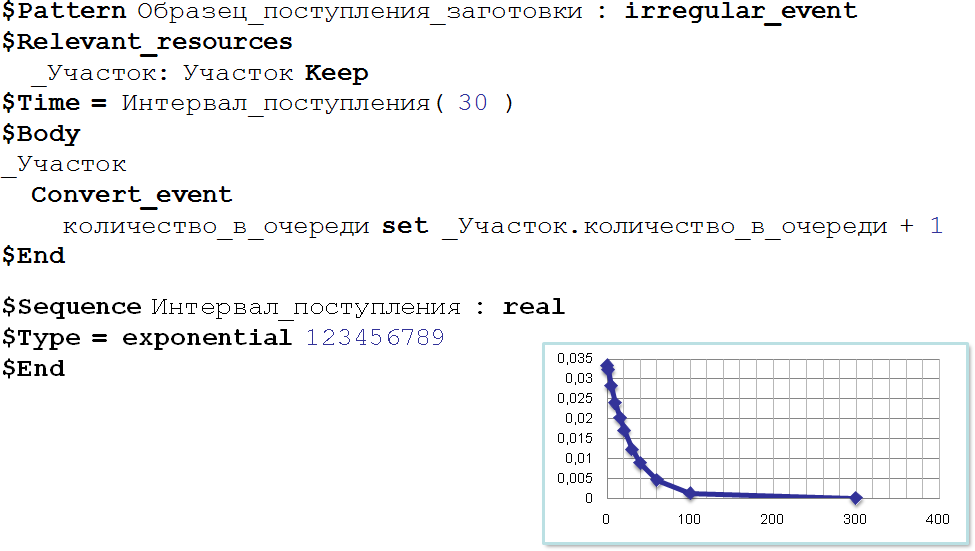
Общий вид



Пример события



Пример события на РДО

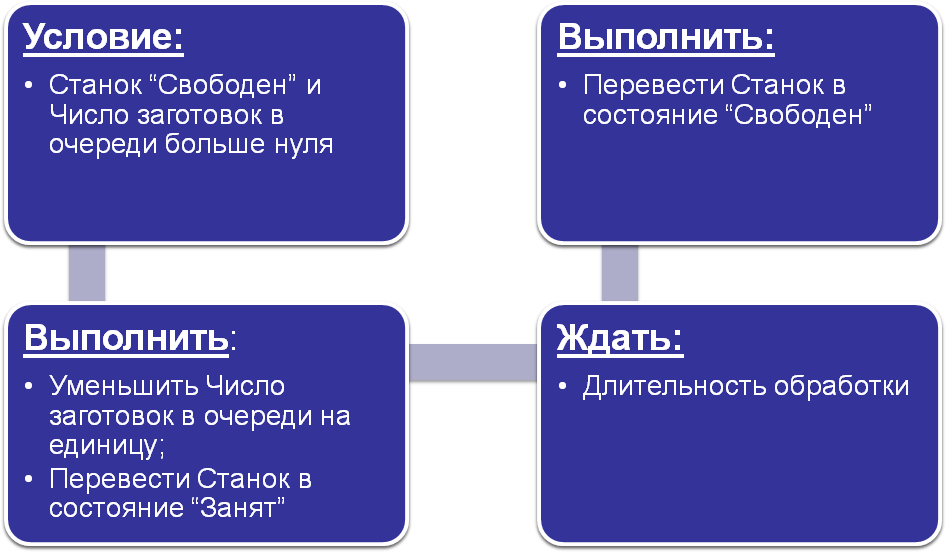


## Действие

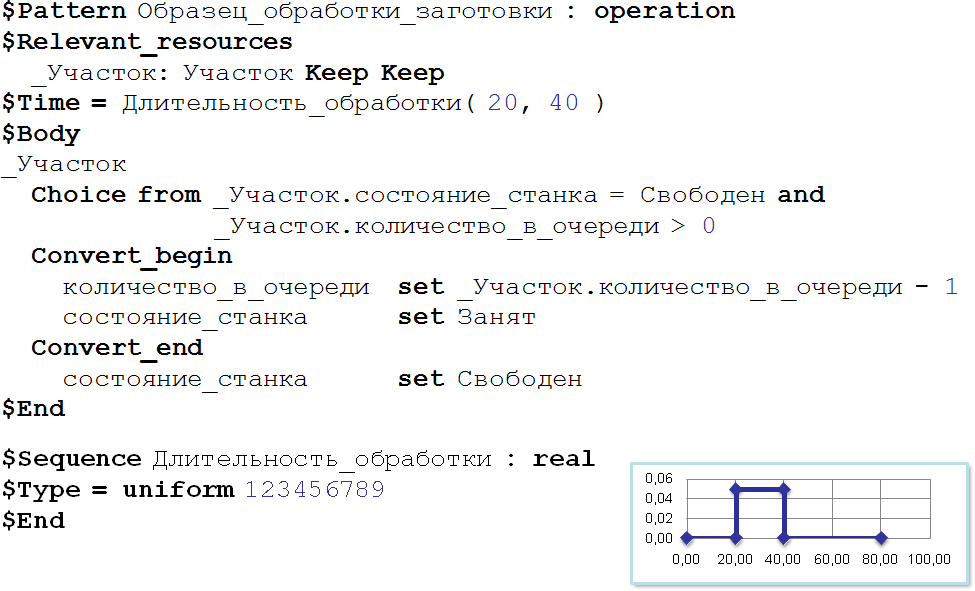
Общий вид



Пример действия

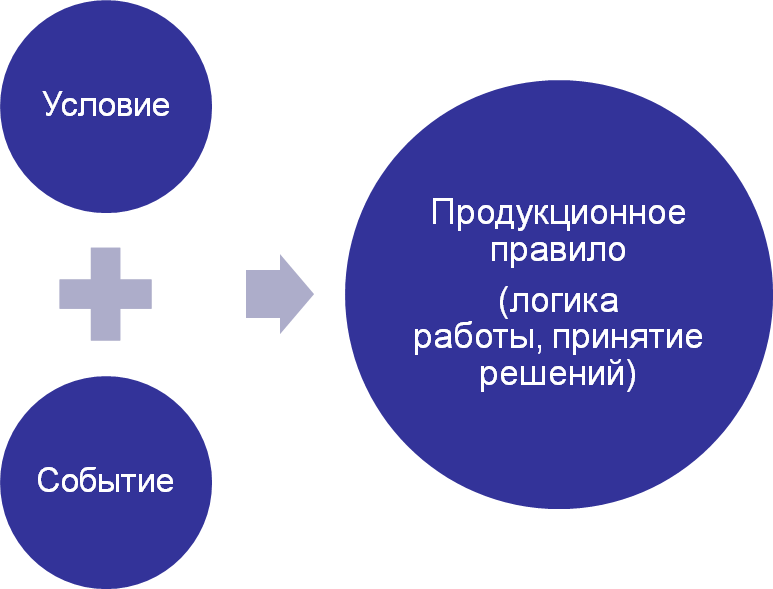


Пример действия на РДО

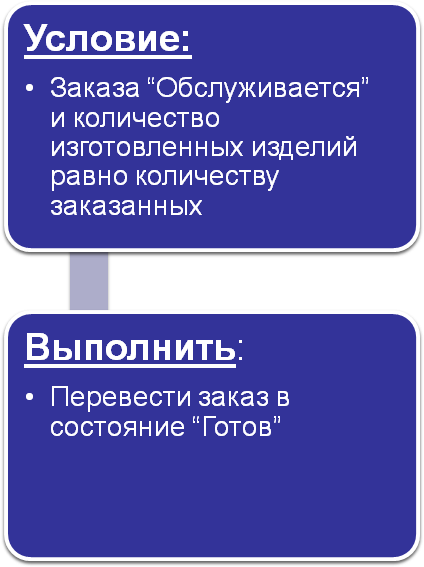


## Продукционное правило

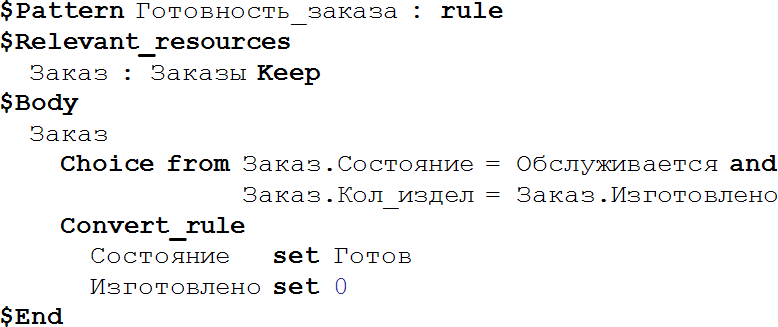
Общий вид



Пример продукционного правила



Пример продукционного правила на РДО



# Запуск программного комплекса

Для начала работы системы необходимо запустить модуль RAO-studio.exe. После загрузки программы откроется основное окно (см. Рис. 1). Оно имеет [главное меню](mk:@MSITStore:D:\rdo\rdo_studio\rdo_help\rdo_studio_rus\RAO-studio.chm::/html/work_menu_intro.htm), кнопки панели управления, "плавающие" окна объектов и вывода со своими закладками, строку состояния и окно с [набором закладок](mk:@MSITStore:D:\rdo\rdo_studio\rdo_help\rdo_studio_rus\RAO-studio.chm::/html/work_model_tabsheets.htm) ("PAT", "RTP", "RSS", "OPR", "FRM", "FUN", "DPT", "SMR", "PMD"), соответствующих [объектам модели](ms-its:RAO-language.chm::/html/rdo_terms.htm#object) для редактирования новой модели.

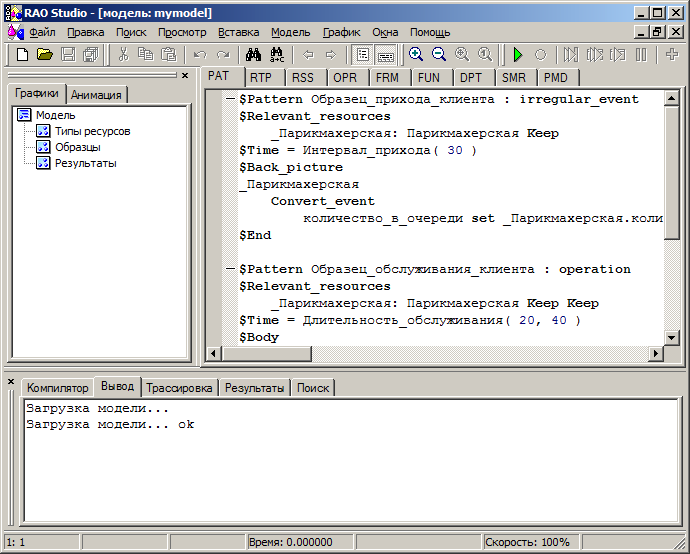
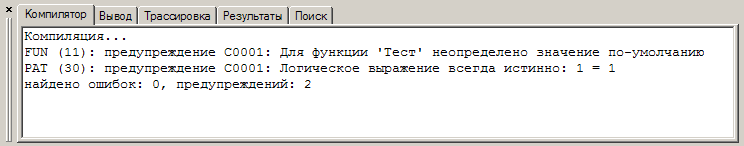


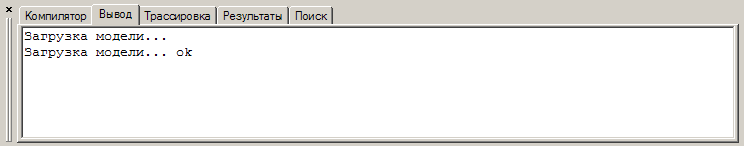
Рис. Главное окно

Снизу располагается окно вывода с закладками:

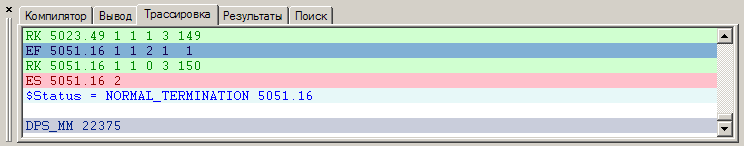
* Компилятор: отображает информацию о процессе компиляции модели, выводит сообщения об ошибках компиляции, позволяет переключиться в текст модели на место возникновения выбранной ошибки



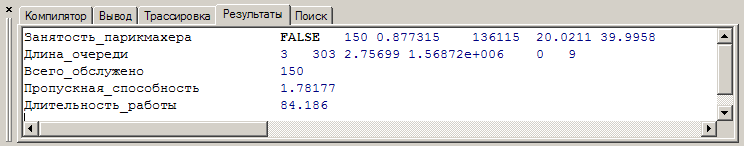
* Вывод: окно, в которое любой модуль может вывести отладочную информацию, например, здесь отображается список закруженных для анимации картинок, при запуске модели на исполнение



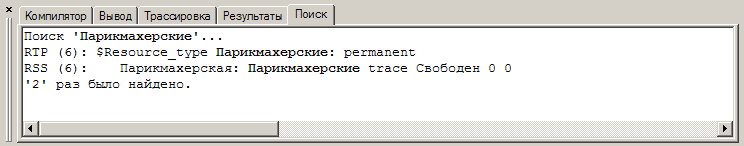
* Трассировка: отображает файл трассировки процесса моделирования в режиме реально времени



* Результаты: отображает результаты моделирования по завершении прогона



* Поиск: отображает результаты поиска подстроки по всей модели, позволяет переключиться в текст модели на найденный фрагмент



Сбоку от главного окна располагается плавающее окно объектов модели с закладками Графики и Анимация (см. Рис. 2). На первой отображает список трассируемых объектов в процессе моделирования, что позволяет отобразить [график](mk:@MSITStore:D:\rdo\rdo_studio\rdo_help\rdo_studio_rus\RAO-studio.chm::/html/work_model_chart.htm) по выбранному элементу. На второй - список доступных окон анимации в процессе моделирования.

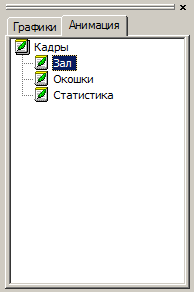
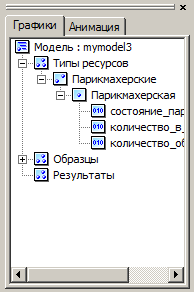
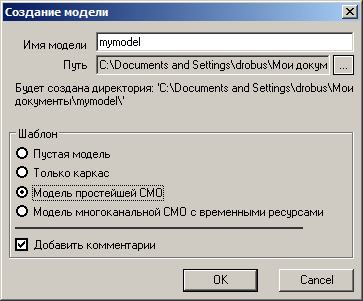


Рис. Панели графиков и анимации

# Создание модели

Создать новую [модель](ms-its:RAO-language.chm::/html/rdo_terms.htm#model) можно с помощью [пункта меню](mk:@MSITStore:C:\Documents%20and%20Settings\drobus\Мои%20документы\1\RAO-studio.chm::/html/work_menu_file.htm#file_new) **Файл/Новая модель**, комбинации клавиш **Ctrl+N** или кнопки на панели инструментов проекта - . Если перед созданием новой модели редактировалась другая модель, и изменения не были сохранены, то выдается запрос на сохранение изменений.

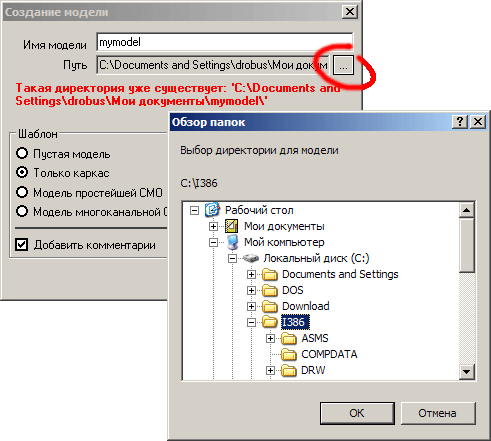
Новая модель будет создана на основе варианта, который выбирается в диалоге:



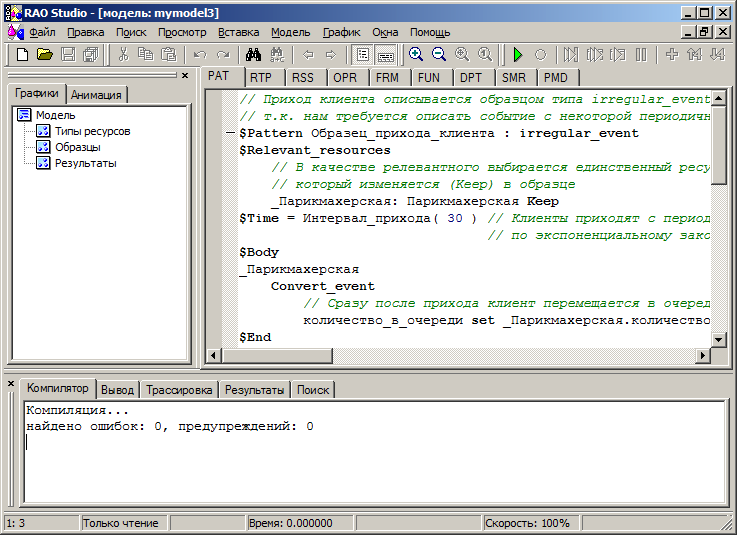
|  |  |
| --- | --- |
| Шаблон | Описание |
| Пустая модель | Создается пустая модель с заполненным файлом прогона имя\_модели.smr |
| Только каркас | Файлы модели будут содержать только [шаблоны](mk:@MSITStore:C:\Documents%20and%20Settings\drobus\Мои%20документы\1\RAO-studio.chm::/html/work_model_insert.htm) типовых конструкций ([объектов](ms-its:RAO-language.chm::/html/rdo_terms.htm#object)) модели |
| Модель простейшей СМО | Модель простейшей парикмахерской с одним парикмахером. Модель компилируется, запускается, выводятся собираемые показатели. |
| Модель многоканальной СМО с временными ресурсами | Модель простейшей парикмахерской с тремя парикмахерами, клиенты создаются при появлении в парикмахерской и удаляются после обслуживания. Модель компилируется, запускается, выводятся собираемые показатели. |

При установленном флаге **Добавить комментарии** текс модель будет дополнен пояснениями.

Все файлы создаваемой модели имеют одно указанное имя, но разные расширения, создаются они в папке с одноимённым названием и имеют текстовый формат. Если модель с указанным именем уже существует, то можно сменить имя в поле ввода имени или директорию создаваемой модели, с помощью соответствующей кнопки под именем модели:



Ниже приводится пример созданной модели на основе шаблона простейшей СМО с включенными пояснениями:



# Редактирование модели

## Редактирование текста модели

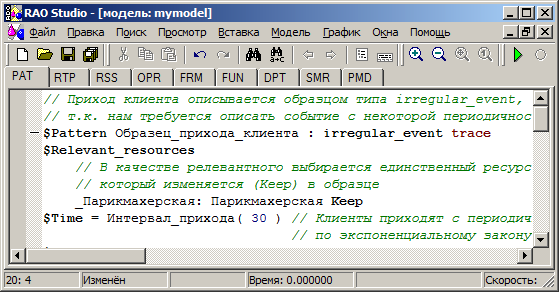
Редактирование [объектов модели](ms-its:RAO-language.chm::/html/rdo_terms.htm#object) осуществляется в полях редактирования, находящихся на соответствующих [закладках](mk:@MSITStore:C:\Documents%20and%20Settings\drobus\Мои%20документы\1\RAO-studio.chm::/html/work_model_tabsheets.htm) главного окна программного комплекса. При редактировании используются стандартные клавиши и сочетания клавиш WINDOWS.

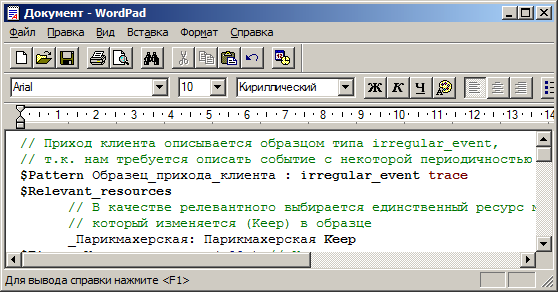
В процессе редактирования некоторая информация отображается в строке состояния:

* В первой позиции отображается текущее положение каретки (позиция в строке, номер строки) в редактируемом [объекте](ms-its:RAO-language.chm::/html/rdo_terms.htm#object).
* Во второй - состояние редактируемого объекта. В том случае, если имело место изменение текста объекта, в данной панели отображается слово "Изменён". Если объект доступен только по чтению, то на панели отображается фраза "Только чтение".
* Третья – индикатор режима редактирования текста (вставка или замена) в текущем объекте. Переключение режимов осуществляется клавишей Ins. Если для данного объекта включен режим замены, то на панели отображается слово "Заменя" и изменяется форма каретки.



Скопировать выделенный фрагмент в буфер обмена в формате RTF с сохранением форматирования и цветового выделения [синтаксических конструкций языка](ms-its:RAO-language.chm::/html/rdo_base_intro.htm) можно путем выбора [пункта меню](mk:@MSITStore:C:\Documents%20and%20Settings\drobus\Мои%20документы\1\RAO-studio.chm::/html/work_menu_edit.htm#edit_copyasrtf) **Правка/Копировать как RTF**:





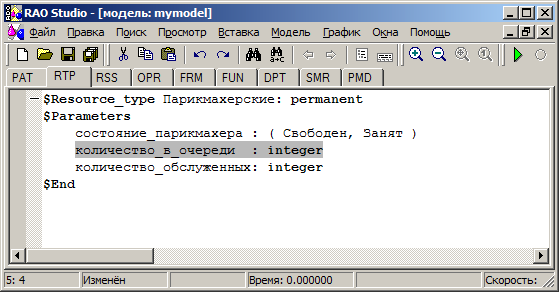
Существует возможность изменения регистра символов в выделенном фрагменте. Для изменения регистра на верхний (все прописные буквы) используется [пункт меню](mk:@MSITStore:C:\Documents%20and%20Settings\drobus\Мои%20документы\1\RAO-studio.chm::/html/work_menu_edit.htm#edit_uppercase) **Правка/В верхний регистр** (комбинация клавиш **Ctrl+Shift+U**), а для изменения регистра на нижний - [пункт меню](mk:@MSITStore:C:\Documents%20and%20Settings\drobus\Мои%20документы\1\RAO-studio.chm::/html/work_menu_edit.htm#edit_lowercase) **Правка/В нижний** регистр (комбинация клавиш **Ctrl+U**):

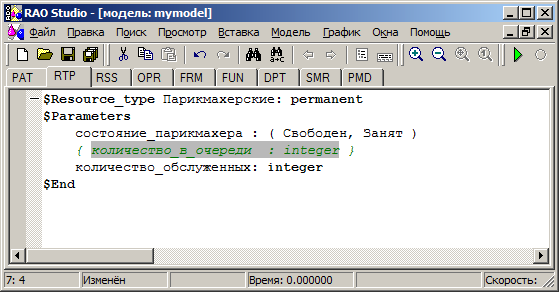






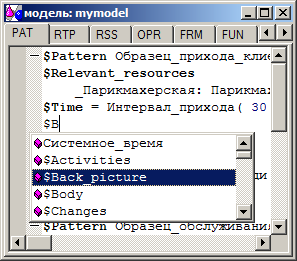
Можно закомментировать выделенный фрагмент текста. Для этого необходимо выбрать [пункт меню](mk:@MSITStore:C:\Documents%20and%20Settings\drobus\Мои%20документы\1\RAO-studio.chm::/html/work_menu_edit.htm#edit_comment) **Правка/Закомментировать выделенное** или нажать комбинацию клавиш **Ctrl+Q**. При этом перед началом выделенного фрагмента будет установлен символ начала комментария, а после окончания - символ конца комментария:





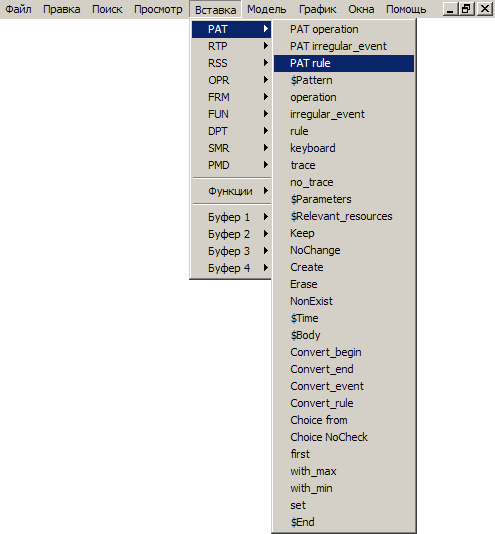
## Автозавершение

В данной версии программного комплекса имеется механизм автоматического завершения (автозавершения) [синтаксических конструкций языка](ms-its:RAO-language.chm::/html/rdo_base_intro.htm). Автозавершение позволяет пользователю не вводить полностью самостоятельно синтаксические конструкции языка. Для пользователя достаточно ввести один или несколько начальных символов конструкции и нажать сочетание клавиш **Ctrl+Space** или выбрать [пункт меню](mk:@MSITStore:C:\Documents%20and%20Settings\drobus\Мои%20документы\1\RAO-studio.chm::/html/work_menu_edit.htm#edit_complete) **Правка/Завершить слово**. При этом рядом с кареткой появится выпадающий список [зарезервированных слов языка](ms-its:RAO-language.chm::/html/rdo_lex_div_res_words.htm), а курсор будет позиционирован на ближайшем слове, начало которого совпадает по написанию с символами, введенными пользователем. Далее необходимо выбрать требуемое слово из списка с помощью клавиатуры (клавиши стрелок и клавиша **Enter**) или мыши. Выбранное слово заменит введенные пользователем начальные символы. Убрать выпадающий список слов без внесения изменений в текст [модели](ms-its:RAO-language.chm::/html/rdo_terms.htm#model) можно с помощью клавиши **Esc**. Если в списке возможных замен находится только значение, то оно подставляется автоматически, а сам список не отображается.

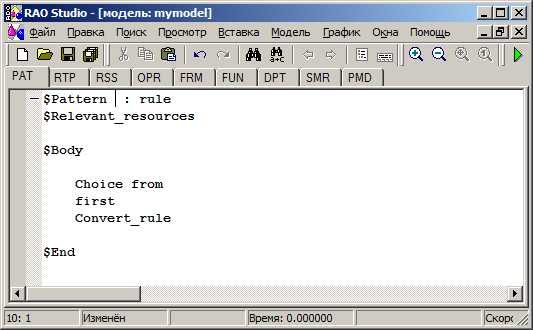


## Шаблоны и вставка синтаксических конструкций

Для обеспечения возможности быстрого ввода текста [модели](ms-its:RAO-language.chm::/html/rdo_terms.htm#model) с минимальным количеством ошибок пользователь имеет возможность работы с шаблонами элементов модели и автоматически вставлять [синтаксические конструкции языка](ms-its:RAO-language.chm::/html/rdo_base_intro.htm). Для этого необходимо воспользоваться [меню](mk:@MSITStore:C:\Documents%20and%20Settings\drobus\Мои%20документы\1\RAO-studio.chm::/html/work_menu_insert.htm) **Вставка**. В данном меню шаблоны и синтаксические конструкции сгруппированы по [объектам модели](ms-its:RAO-language.chm::/html/rdo_terms.htm#object). Группы названы исходя из тех же соображений, что и [закладки](mk:@MSITStore:C:\Documents%20and%20Settings\drobus\Мои%20документы\1\RAO-studio.chm::/html/work_model_tabsheets.htm) в главном окне программного комплекса.



При выборе группы раскрывается подменю, которое содержит шаблоны элементов данного объекта и синтаксические конструкции, употребляемые в данном объекте. Шаблоны находятся в верхней части подменю и их наименования начинаются с наименования подгруппы. Далее, если необходимо, следуют уточняющие предложения. Например: при выборе пункта меню Вставка/PAT/PAT rule в текст будет вставлен шаблон для описания [образца типа продукционное правило](ms-its:RAO-language.chm::/html/rdo_obj_pat.htm#pat_type):



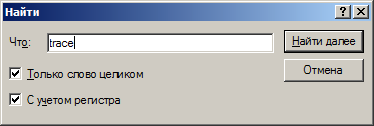
Синтаксические конструкции следуют за шаблонами, и выбор конкретной конструкции обеспечивает ее вставку в текст модели без синтаксических ошибок.

Также существует возможность вставки [стандартных арифметических и логических функций](ms-its:RAO-language.chm::/html/rdo_base_std_var_func.htm) языка. Эти возможности предоставляют подпункты [пункта меню](mk:@MSITStore:C:\Documents%20and%20Settings\drobus\Мои%20документы\1\RAO-studio.chm::/html/work_menu_insert.htm#insert_function) **Вставка/Функции**.

Меню **Вставка** также имеется в выпадающем меню поля редактирования, активизируемом при нажатии на правую кнопку мыши.

## Поиск фразы по всей модели

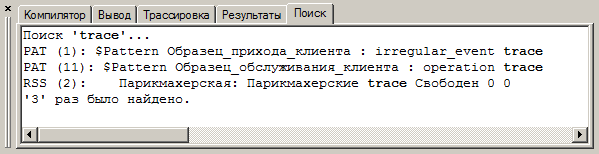
Пользователь имеет возможности поиска подстрок по всей [модели](ms-its:RAO-language.chm::/html/rdo_terms.htm#model) целиком. Для общего поиска подстроки необходимо воспользоваться [пунктом меню](mk:@MSITStore:C:\Documents%20and%20Settings\drobus\Мои%20документы\1\RAO-studio.chm::/html/work_menu_search.htm#search_in_model) **Поиск/Найти в моделе**. После этого будет выведено стандартное диалоговое окно WINDOWS для поиска подстрок, где пользователю необходимо указать подстроку для поиска, задать параметры поиска и нажать кнопку **Найти далее** (**Find Next**).



После этого будет произведен поиск подстроки в текстах файлов модели, результаты которого будут отображены на закладке Поиск окна Панель вывода. Результаты представляются в виде:

* тип файла модели (PAT, RPT, RSS и т.п.)
* номер строки
* строка целиком, в которой была найдена искомая фраза

Отчет о найденных подстроках заканчивается сообщением о количестве найденных подстрок:



При двойном щелчке мыши по строке с результатами поиска, курсор устанавливается в начало найденной подстроки в тексте модели.

## Навигация с помощью закладок

Поставить/снять закладку в определенном месте текста модели, можно вызвав [пункт меню](mk:@MSITStore:C:\rdo\rdo_studio\rdo_help\rdo_studio_rus\RAO-studio.chm::/html/work_menu_search.htm#search_book) **Поиск/Поставить/снять закладку** или нажав сочетание клавиш **Ctrl+F2**. При этом строка, на которой находится закладка, помечается маркером на поле слева от поля редактирования, а в случае, если [поле не видимо](mk:@MSITStore:C:\rdo\rdo_studio\rdo_help\rdo_studio_rus\RAO-studio.chm::/html/work_options_editor.htm#margin), то соответствующим цветом подсвечивается вся строка.



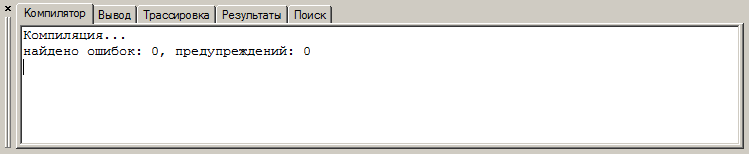


Закладки могут быть установлены во всех [объектах модели](ms-its:RAO-language.chm::/html/rdo_terms.htm#object). Программный комплекс обеспечивает навигацию по всем установленным закладкам. Для перемещения к следующей закладке после позиции каретки в тексте используется [пункт меню](mk:@MSITStore:C:\rdo\rdo_studio\rdo_help\rdo_studio_rus\RAO-studio.chm::/html/work_menu_search.htm#search_nextbook) **Поиск/Перейти к след. закладке**, клавиша **F2** или кнопка  на панели инструментов редактирования. Для перемещения к предыдущей закладке можно воспользоваться [пунктом меню](mk:@MSITStore:C:\rdo\rdo_studio\rdo_help\rdo_studio_rus\RAO-studio.chm::/html/work_menu_search.htm#search_prevbook) **Поиск/Перейти к пред. закладке**, комбинацией клавиш **Shift+F2** или кнопкой  на панели инструментов редактирования.

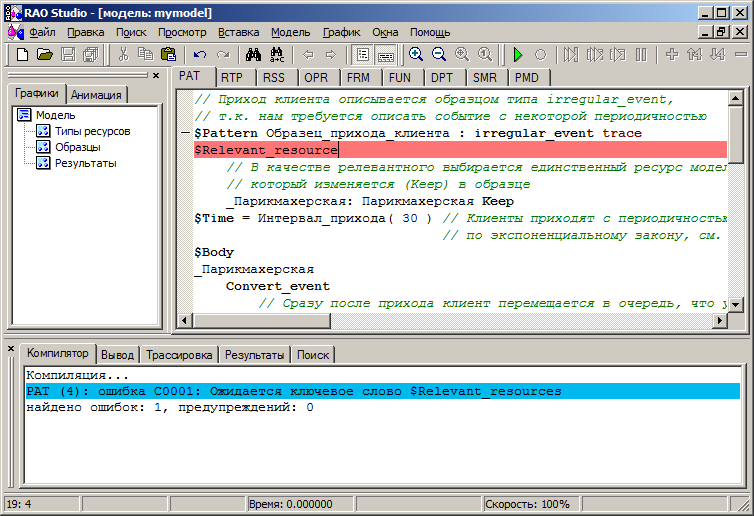
Снять все установленные закладки можно с помощью [пункта меню](mk:@MSITStore:C:\rdo\rdo_studio\rdo_help\rdo_studio_rus\RAO-studio.chm::/html/work_menu_search.htm#search_clearbook) **Поиск/Очисть все закладки** или нажав комбинацию клавиш **Ctrl+Shift+F2**.

# Компиляция и запуск модели

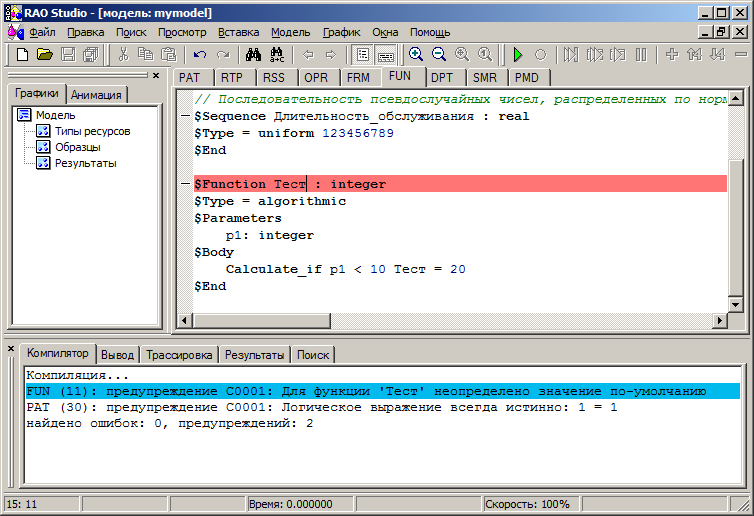
Текст [модели](ms-its:RAO-language.chm::/html/rdo_terms.htm#model) перед запуском компилируется с помощью [пункта меню](mk:@MSITStore:C:\rdo\rdo_studio\rdo_help\rdo_studio_rus\RAO-studio.chm::/html/work_menu_model.htm#rdo_run) **Модель/Скомпилировать** или нажатием клавиши **F7**. На этой стадии происходит обнаружение синтаксических ошибок. Отчет о компиляции выводится на закладку Компилятор [окна вывода](mk:@MSITStore:C:\rdo\rdo_studio\rdo_help\rdo_studio_rus\RAO-studio.chm::/html/work_run.htm#output):

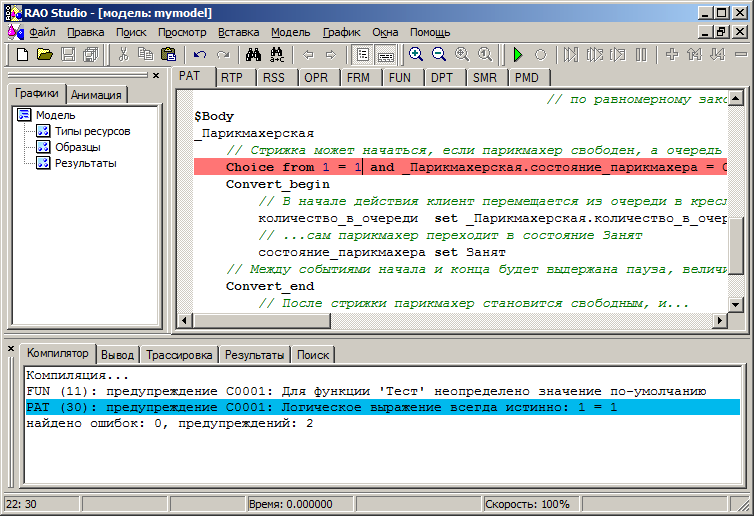


Если была обнаружена ошибка, то курсор автоматически на неё устанавливается:



Так же могут быть выявлены несущественные нарушения, которые называются **предупреждениями**. На них курсор автоматически не устанавливается. Это можно сделать, нажимая на каждое предупреждение мышкой:





Запуск [модели](ms-its:RAO-language.chm::/html/rdo_terms.htm#model) осуществляется путем выбора [пункта меню](mk:@MSITStore:C:\rdo\rdo_studio\rdo_help\rdo_studio_rus\RAO-studio.chm::/html/work_menu_model.htm#rdo_run) **Модель/Запустить**, нажатием клавиши **F5** или кнопки  панели инструментов проекта. Перед запуском модель автоматически записывается и компилируется. Если возникает синтаксическая ошибка, то выдается соответствующее сообщение, и запуск отменяется.

Остановить прогон (запущенную на исполнение модель), можно выбрав [пункт меню](mk:@MSITStore:C:\rdo\rdo_studio\rdo_help\rdo_studio_rus\RAO-studio.chm::/html/work_menu_model.htm#rdo_stop) **Модель/Остановить**, нажатием комбинации клавиш **Shift+F5** или кнопки  панели инструментов проекта.

# Режимы моделирования

После [запуска](mk:@MSITStore:C:\rdo\rdo_studio\rdo_help\rdo_studio_rus\RAO-studio.chm::/html/work_model_run.htm) модели на исполнение, у пользователя есть возможность выбрать режим моделирования.

|  |  |  |
| --- | --- | --- |
| Режим | Кнопка | Описание |
| Максимальная производительность |  | Режим максимальной производительности системы моделирования. Скорость моделирования определяется производительностью компьютера. Масштабный коэффициент не учитывается, регулятор скорости тоже. Отключается вывод анимации, графиков и трассировки на экран (данные по трассировке продолжают собираться). |
| Дискретная имитация |  | В режиме дискретной имитации переходы между событиями осуществляются как можно быстрее без синхронизации с таймером операционной системы, т.е. интервалы времени между событиями в модели учитываются в расчетах модельного времени, но никак не проявляются в анимации и считаются равными друг другу (с точностью до сложности расчета каждого события). Скорость моделирования определяется производительностью компьютера и регулятором скорости имитатора. Если модель содержит кадры анимации, то они могут быть выведены на экран. Доступно построение графиков и просмотр данных трассировки в режиме реального времени. Если сравнивать с режимом синхронной имитации, то для дискретной имитации масштабный коэффициент времени равен бесконечности. Т.е. модель в этом режиме работает с максимальным быстродействием, учитывающим регулятор скорости имитатора, и позволяет отображать анимацию. |
| Синхронная имитация |  | В режиме синхронной имитации переходы между событиями осуществляются по синхронизации с таймером операционной системы, т.е. интервалы времени между событиями в модели учитываются не только в расчетах, но и при выводе анимации. Скорость моделирования определяется пользователем через масштабный коэффициент. Масштабный коэффициент устанавливает отношение единицы модельного времени к одному часу реального времени, который рассчитывается по таймеру операционной системы. Изменяется с помощью соответствующих команд. Регулятор скорости работы имитатора тоже учитывается. Если модель содержит кадры анимации, то они могут быть выведены на экран. Доступно построение графиков и просмотр данных трассировки в режиме реального времени. При большом значении масштабного коэффициента стирается разница между дискретным и синхронным режимами моделирования, и быстродействие модели начинает определяться мощностью компьютера. |
| Пауза |  | Режим паузы. Процесс моделирования приостанавливается. Чтобы его продолжить, достаточно переключиться на любой другой режим моделирования. |

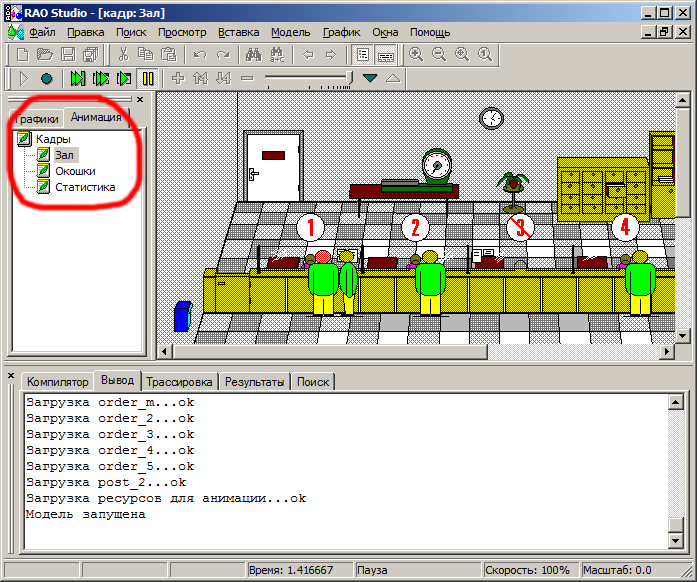
Поменять скорость имитации можно с помощью команд:

|  |  |  |
| --- | --- | --- |
| Команда | Кнопка | Описание |
| Увеличить масштаб |  | Увеличить скорость анимации на 50% процентов. Используется только в режиме синхронной имитации. |
| Увеличить масштаб в 4-ре раза |  | Увеличить скорость анимации в 4 раза. Используется только в режиме синхронной имитации. |
| Уменьшить масштаб в 4-ре раза |  | Уменьшить скорость анимации в 4 раза. Используется только в режиме синхронной имитации. |
| Уменьшить масштаб |  | Уменьшить скорость анимации на 50% процентов. Используется только в режиме синхронной имитации. |
| Скорость имитатора |  | Логарифмическая шкала, регулирующая общую скорость имитатор. Используется в моделях, основанных на продукционных правилах, в которых модельное время остается равным нулю, но имеется анимация. Снизив значение скорости можно подробнее рассмотреть процесс моделирования. Используется в режимах дискретной и синхронной имитации, не используется в режиме максимального быстродействия. |

Текущее модельное время, режим моделирования, скорость и масштабный коэффициент отображаются в строке состояния:



В режиме анимации, пользователь может перемещаться по кадрам анимации, выбирая нужный из списка кадров на закладке Анимация [окна объектов:](mk:@MSITStore:C:\rdo\rdo_studio\rdo_help\rdo_studio_rus\RAO-studio.chm::/html/work_run.htm#workspace)



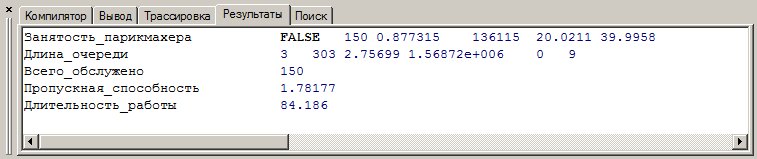
Или с помощью команд:

|  |  |  |
| --- | --- | --- |
| Команда меню Модель | Кнопка | Описание |
| След. кадр анимации |  | Перейти к следующему кадру анимации |
| Пред. кадр анимации |  | Перейти к предыдущему кадру анимации |

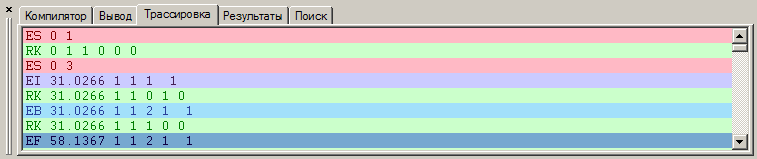
# Результаты моделирования

Результатами моделирования являются объекты:

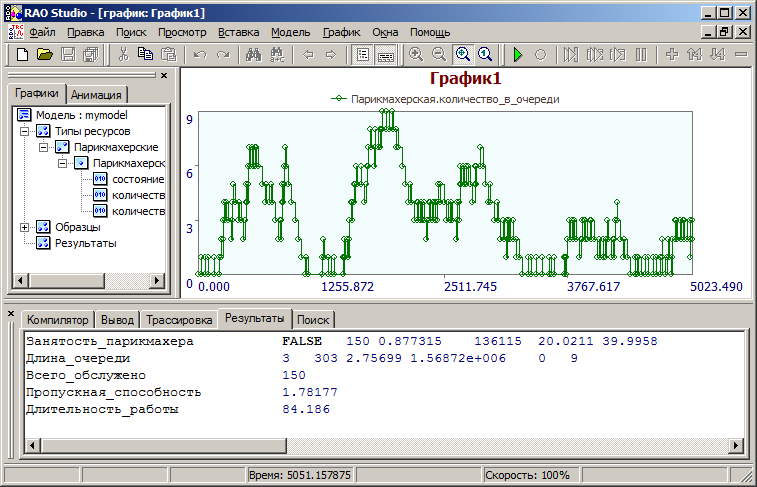
* [объект результатов](ms-its:RAO-language.chm::/html/rdo_res_pmv.htm), который выводится на закладку Результаты [окна вывода](mk:@MSITStore:C:\rdo\rdo_studio\rdo_help\rdo_studio_rus\RAO-studio.chm::/html/work_run.htm#output) автоматически при завершении [прогона](ms-its:RAO-language.chm::/html/rdo_terms.htm#run):



* [объект трассировки](ms-its:RAO-language.chm::/html/rdo_res_trc.htm), который отображается в режиме реального времени в процессе моделирования на закладке Трассировка окна вывода:



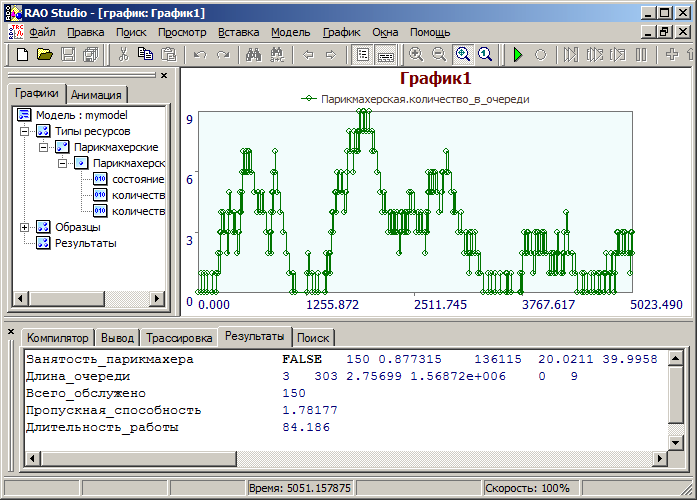
* [графики](mk:@MSITStore:C:\rdo\rdo_studio\rdo_help\rdo_studio_rus\RAO-studio.chm::/html/work_model_chart.htm) изменения трассируемых величин, которые отображается в режиме реального времени в процессе моделирования на закладке Графики [окна объектов](mk:@MSITStore:C:\rdo\rdo_studio\rdo_help\rdo_studio_rus\RAO-studio.chm::/html/work_run.htm#workspace):



# Построение графиков

В данной версии программного комплекса имеется возможность визуально отображать процессы, происходящие в модели, в виде графиков. Добавление графиков состояния ресурса становится возможным только после запуска модели. Графики состояния [ресурса](ms-its:RAO-language.chm::/html/rdo_base_resources.htm) можно добавлять как в процессе работы модели в режиме анимации, так и после завершения моделирования. График состояния ресурса может быть отображен только в том случае, если конкретный ресурс трассируется.

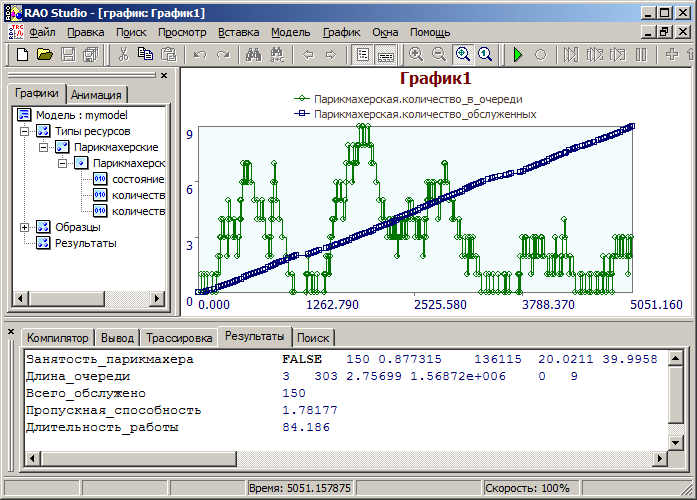
Для добавления графика, во вкладке Графики [окна объектов](mk:@MSITStore:C:\rdo\rdo_studio\rdo_help\rdo_studio_rus\RAO-studio.chm::/html/work_run.htm#workspace) в дереве модели найдите необходимый вам параметр ресурса . Далее, чтобы создать новое окно графика, нужно щелкнуть два раза по выбранному параметру ресурса левой кнопкой мыши или один раз правой и в выпадающем меню выбрать команду Добавить на новый график. По оси абсцисс графика откладывается время наступления событий, при которых изменяется значение параметра выбранного ресурса.



При первом построении графика, масштаб по оси ординат устанавливается автоматически. Для изменения масштаба по оси абсцисс можно воспользоваться панелью инструментов Масштаб или использовать выпадающее меню при правом щелчке мышки по области графика. Основными функциями масштабирования являются:

|  |  |  |  |
| --- | --- | --- | --- |
| Пункт меню | Комбинация клавиш | Кнопка | Функция |
| Увеличить масштаб | Ctrl+Num + |  | Увеличение масштаба |
| Уменьшить масштаб | Ctrl+Num - |  | Уменьшение масштаба |
| Автоматический масштаб | Ctrl+Num \* |  | Автоматический подбор масштаба по ширине области графика |
| Восстановить масштаб | Ctrl+Num / |  | Восстановление начального масштаба |

Имеется возможность добавлять на уже существующий график другие. Для этого необходимо мышкой перетащить имя параметра ресурса на область уже построенного графика.



# Модель простейшей парикмахерской

## Работа с моделью

Рассмотрим основные команды программного комплекса на примере модели простейшей парикмахерской, которая является представителем систем массового обслуживания (СМО). Исходные тексты модели можно найти на сайте http://rdo.rk9.bmstu.ru. Опишем систему и исходные данные. Имеется парикмахерская с одним парикмахером. Клиенты заходят в парикмахерскую и встают в очередь. Интервал между прибытиями – случайное число, распределенное по экспоненциальному закону со средним значением 40 минут. Длина очереди не ограничена. После ожидания в очереди они обслуживаются. Время обслуживания - случайное число, распределенное по равномерному закону в интервале от 20 до 40 минут. Затем клиенты покидают систему. В начальный момент времени парикмахер свободен и очередь пуста. Время завершения работы – одна 7‑ми дневная рабочая неделя с длительность смены в 12‑ть часов. Схема парикмахерской представлена на .

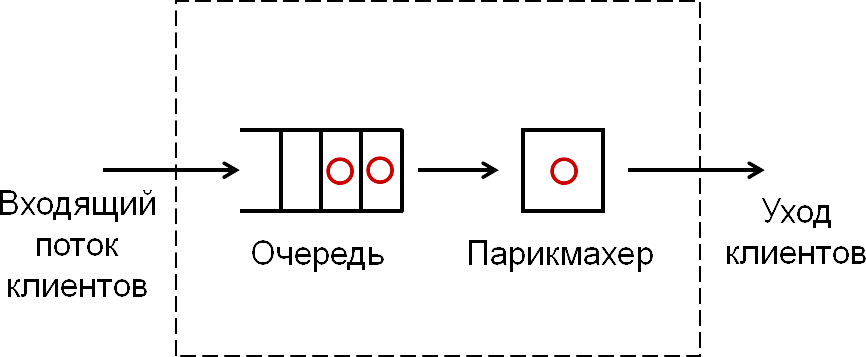


Рис. Схема парикмахерской

После запуска открывается главное окно программы. Загрузить модель парикмахерской можно через меню **Файл/Открыть**. После этого текст модели будет отображен на закладках (см. Рис. 4).

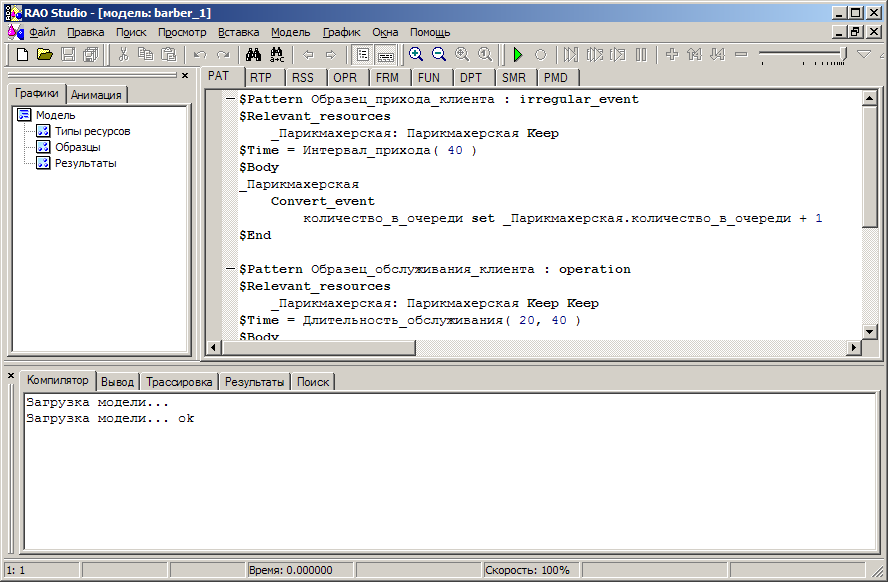


Рис. . Модель после открытия

Перемещаться по различным объектам модели (ресурсы, правила функционирования, функции, собираемые показатели и др.) можно с помощью закладок. Ниже приведены названия закладок с краткими комментариями:

|  |  |  |
| --- | --- | --- |
| Имя закладки | Тип РДО-объекта | Краткое описание |
| PAT | Паттерны, они же образцы операций | С помощью паттернов описываются правила функционирования системы. |
| RTP | Типы ресурсов | Типы описывают параметры ресурсов, определяют структуру базы данных модели. |
| RSS | Ресурсы | Содержимое базы данных модели. Ресурсы изменяют значения своих параметров в процессе моделирования. Временные ресурсы могут динамически создаваться и удаляться. |
| OPR | Операции | Содержимое базы знаний модели. Если от описанного паттерна не создается операция, то паттерн в модели не будет использован. |
| FRM | Кадры анимации | Анимация отображает текущее состояние базы данных. Используется как механизм отладки моделей. |
| FUN | Функции | На этой закладке описываются константы, последовательности псевдослучайных чисел и функции. |
| DPT | Точки принятия решений | Развивает идеи закладки OPR. Так же используется для определения содержимого базы знаний модели, но оперирует не операциями, а активностями. Позволяет группировать паттерны в логические группы, временно отключать некоторые из них. В дополнении к этому на закладке DPT могут быть описаны точки принятия решений. |
| SMR | Файл прогона | Закладка содержит перечень файлов модели (содержимое каждой закладки хранится в виде отдельного текстового файла), режимы моделирования и условие окончания. |
| PMD | Описание показателей | Описываются собираемые показатели, например, подсчет загрузки устройства или средняя длина очереди. |

Запустить модель на исполнение можно командой **Модель/Запустить (F5)**. По окончании будут выведены результаты на закладку Результаты, которая находится внизу главного окна. Результаты представляют собой рассчитанные показатели (см. Рис. 5). Список собираемых показателей определяет сам разработчик модели, формируя его на закладке PMD. На приведенном рисунке текст модели занимает больше места внутри главного кона, т.к. была скрыта панель объектов, которая находилась с левой стороны окна. Для этого можно использовать команду **Просмотр/Показать панель объектов**. Повторный вызов команды вернет панель на место. Аналогично можно убрать и панель вывода, которая находится внизу окна.

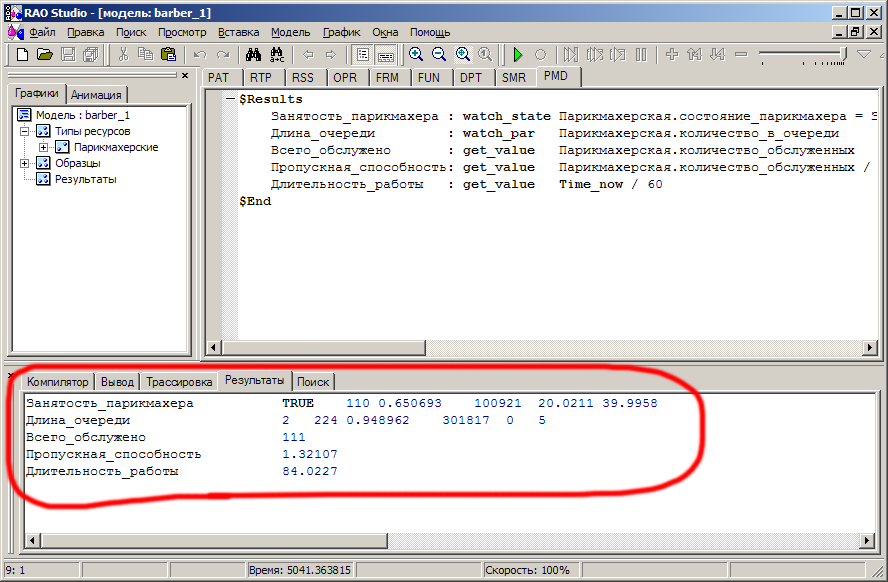


Рис. . Результаты моделирования

В процессе моделирования изменяются параметры ресурсов. Можно отобразить зависимость параметров ресурсов от времени в виде графиков. Для этого на закладке Графики надо выбрать интересующий параметр двойным кликом. Важно отметить, что графики строятся на основе трассировки, в которую сбрасываются информация о процессе моделирования. Объем этой информации достаточно большой, поэтому трассировка по-умолчанию отключена. Чтобы установить признак трассировки используется ключевое слово trace. Его достаточно добавить после имени типа ресурса, чтобы начать записывать в трассировку изменения параметров этого ресурса (см. Рис. 6). Можно трассировать и паттерны, чтобы узнать моменты их наступления. Если трассировка отключена в SMR или нет признака, то и графиков построить нельзя. А так как файл трассировки имеет текстовый формат, документация на которую можно найти в справке по языку, то имеет возможность обрабатывать результаты моделирования в сторонних программах, например электронных таблицах.

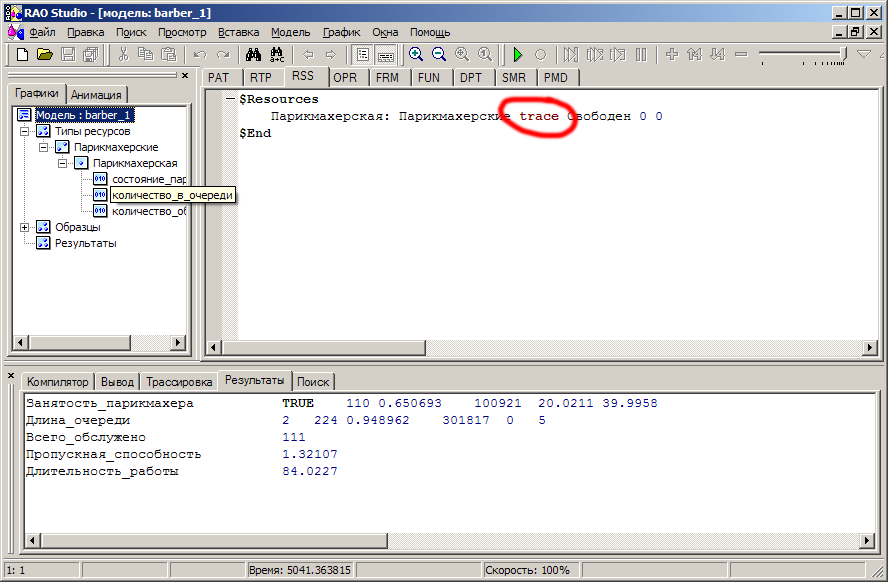


Рис. . Добавление признака трассировки

Графики строятся в режиме реального времени, но могут быть просмотрены и после прогона. Информации, как правило, много. Чтобы уместить её на одном экране надо, стоя на графике, вызвать выпадающее меню и выбрать пункт **Автоматический масштаб** (см. Рис. 7). На один лист можно вывести несколько графиков, если на уже существующий перетащить другой параметр (см. Рис. 8).

Трассировка выводится на соответствующую закладку панели вывода (см. Рис. 10). Хочется обратить внимание, что в процессе имитационном моделирования в один момент времени может произойти несколько событий и измениться сразу несколько параметров. На графике такие изменения будут накладываться друг на друга (см. Рис. 9). Чтобы увидеть последовательность изменения параметров в один момент времени нужно убрать галочку **‘Схлопнуть’ время** у выпадающего меню графика. Тогда станет видно, что в нашей модели в момент времени 1199.520 мин. сначала состояние парикмахера стало Свободен (он закончил стрижку i-го клиента), потом уменьшилась длина очереди на единицу (i+1-ый клиент перешел из очереди в кресло парикмахера), и только потом состояние парикмахера опять стало Занят (началась стрижка i+1-го клиента). См. Рис. 10.

При изучении и отладки модели часто бывает нужно узнать, где используется то или иное выражение. Например, нам понадобится узнать, где описана Длительность\_обслуживания, используемая для задания длительности стрижки на закладке паттернов. Выделяем её, выполняем команду **Поиск/Найти в модели** и видим все найденные вхождения на закладке Поиск панели вывода (см. Рис. 11). Нажимая на строчки с результатами поиска, можно перемещаться сразу к найденному месту.

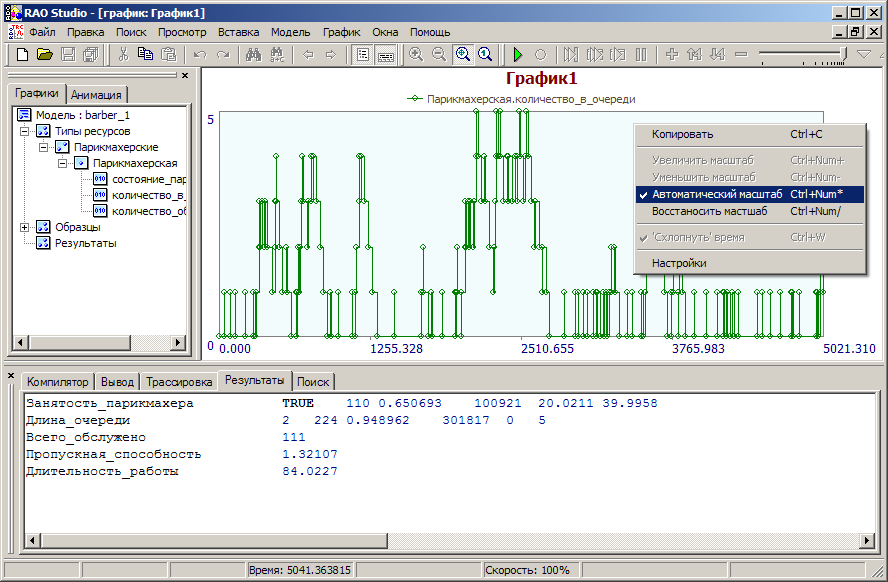


Рис. . График изменения длины очереди

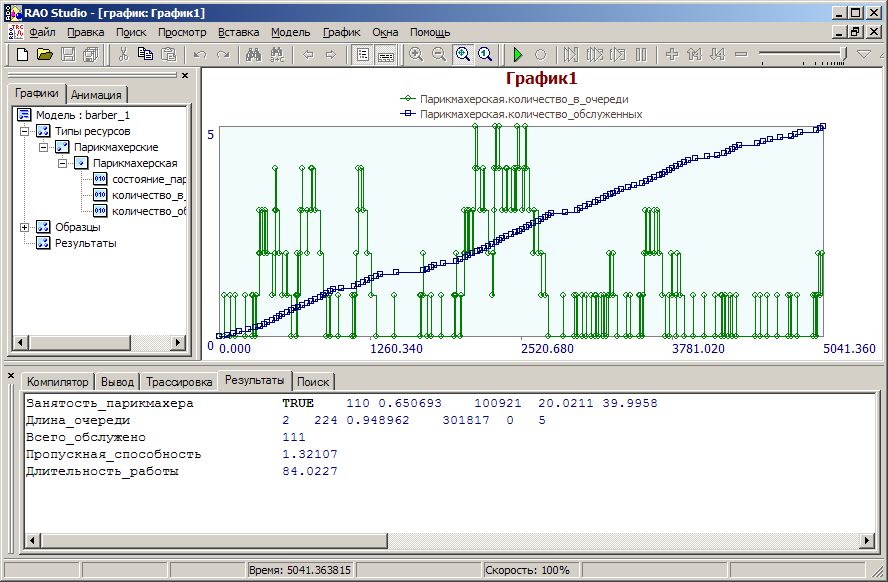


Рис. . Несколько параметров на одном графике

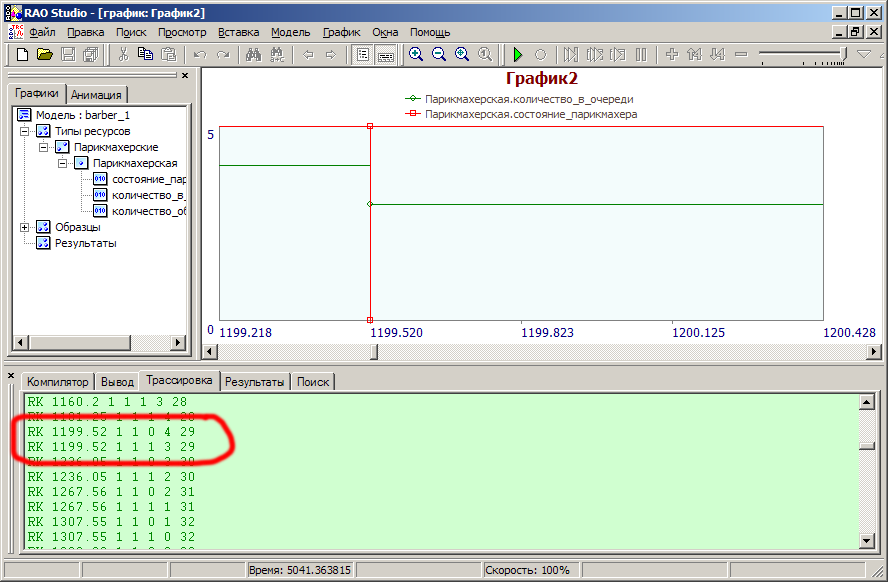


Рис. . Одновременное изменение двух параметров

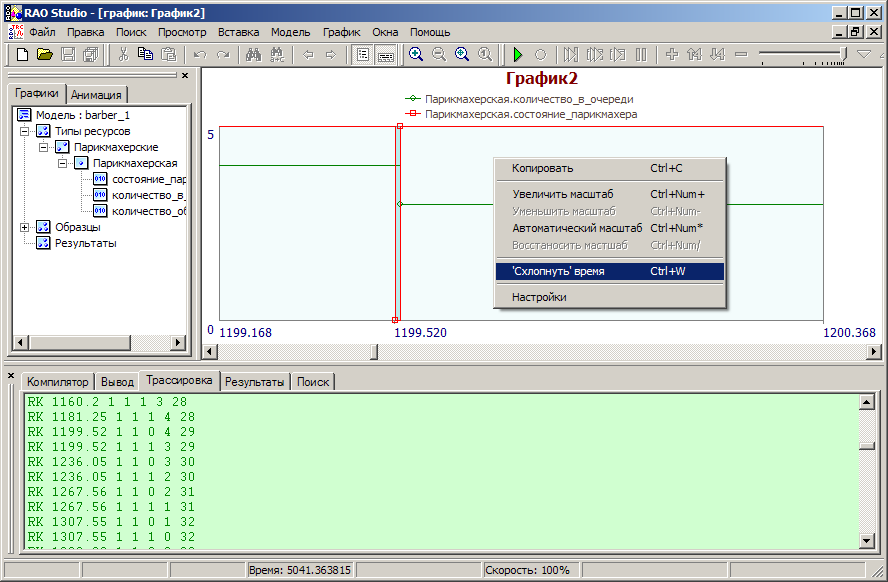


Рис. . Последовательное изменение двух параметров в один момент времени

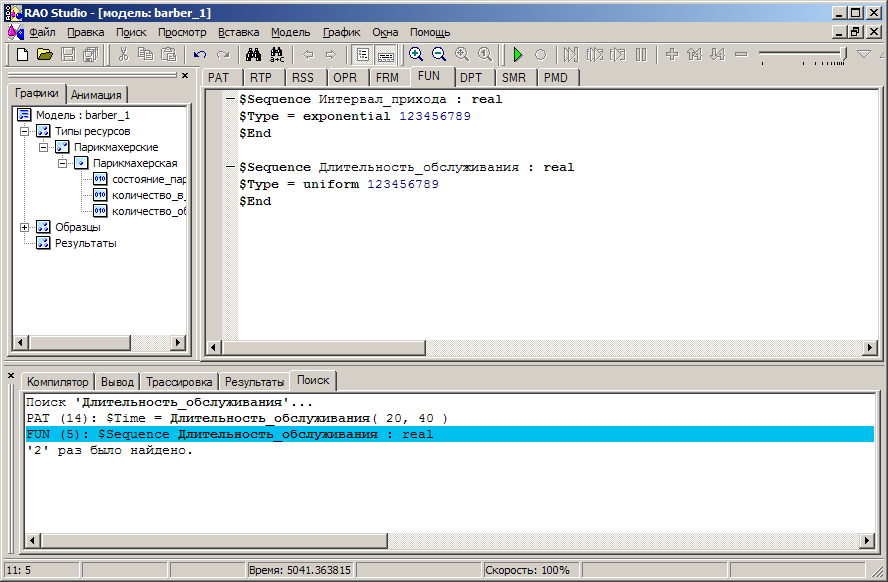


Рис. . Поиск по модели

## Описание модели

Начнем изучать модель с её ресурсов. Через ресурсы описываются объекты реальной системы. Совокупность всех ресурсов часто называют базой данных (БД) модели. Для начала определим их типы. Ресурс состоит из параметров, значения которых изменяются в процессе прогона модели, а сами параметры указывается в описании типа. От одного типа можно создать несколько ресурсов, отличающихся значениями параметров, при этом, сами ресурсы описываются на закладке RSS. Для нашей задачи определим тип ресурса постоянного вида (**permanent**): созданные от такого типа ресурсы можно изменять в процессе прогона, но нельзя удалять или создавать динамически. В качестве параметров опишем состояние парикмахера, количество клиентов в очереди и общее количество обслуженных клиентов.

|  |
| --- |
| Типы ресурсов, закладка RTP |
| **$Resource\_type** Парикмахерские: **permanent**  **$Parameters**  состояние\_парикмахера : ( Свободен, Занят )  количество\_в\_очереди : **integer**  количество\_обслуженных: **integer**  **$End** |

Теперь можно перейти на закладку RSS и создать ресурс, по определенному выше типу.

|  |
| --- |
| Ресурсы, закладка RSS |
| **$Resources**  Парикмахерская: Парикмахерские Свободен 0 0  **$End** |

Приведенная форма записи означает, что будет создан один ресурс Парикмахерская, с начальными значениями параметров:

* состояние\_парикмахера = Свободен
* количество\_в\_очереди = 0
* количество\_обслуженных = 0

, т.е. в начале система пуста.

Теперь опишем правила функционирования нашей системы, они хранятся в базе знаний (БЗ) модели. Создадим для этого два шаблона: событие прихода в парикмахерскую и операцию стрижки. Приход клиента описывается образцом типа **irregular\_event**, т.к. нам требуется описать событие с некоторой периодичностью. В качестве релевантного выбирается единственный ресурс модели (см. RSS), который изменяется (**Keep**) в образце. Клиенты приходят с периодичностью в 40 минут по экспоненциальному закону (см. FUN, Интервал\_прихода). Сразу после прихода клиент перемещается в очередь, что увеличивает её на 1.

|  |
| --- |
| Образец прихода клиента, закладка PAT |
| **$Pattern** Образец\_прихода\_клиента : **irregular\_event**  **$Relevant\_resources**  \_Парикмахерская: Парикмахерская **Keep**  **$Time** = Интервал\_прихода( 40 )  **$Body**  \_Парикмахерская  **Convert\_event**  количество\_в\_очереди **set** \_Парикмахерская.количество\_в\_очереди + 1  **$End** |

Действие по обслуживанию клиента, описывается образцом типа **operation**, т.к. операция обслуживания имеет длительность и условие начала: стрижка может начаться, если парикмахер свободен, а очередь не пуста. В качестве релевантного опять выбирается единственный ресурс модели (см. RSS), который изменяется (**Keep**/**Keep**) и в начале и в конце образца. Продолжительность стрижки выбирается в интервале от 20 до 40 минут по равномерному закону (см. FUN, Длительность\_обслуживания). В начале действия клиент перемещается из очереди в кресло парикмахера, а сам парикмахер переходит в состояние Занят. После стрижки парикмахер становится свободным, и увеличивается счетчик обслуженных клиентов на 1 для сбора статистики.

|  |
| --- |
| Образец обслуживания клиента, закладка PAT |
| **$Pattern** Образец\_обслуживания\_клиента : **operation**  **$Relevant\_resources**  \_Парикмахерская: Парикмахерская **Keep** **Keep**  **$Time** = Длительность\_обслуживания( 20, 40 )  **$Body**  \_Парикмахерская  **Choice** **from** \_Парикмахерская.состояние\_парикмахера = Свободен **and** \_Парикмахерская.количество\_в\_очереди > 0  **Convert\_begin**  количество\_в\_очереди **set** \_Парикмахерская.количество\_в\_очереди - 1  состояние\_парикмахера **set** Занят  **Convert\_end**  состояние\_парикмахера **set** Свободен  количество\_обслуженных **set** \_Парикмахерская.количество\_обслуженных + 1  **$End** |

Описанные на закладке PAT образцы не будут использоваться в модели, если по ним не определены операции на закладке OPR. PAT и OPR, находятся в такой же логической зависимости, как и RTP и RSS, т.е. существует возможность от одного образца создать несколько операций, которые могут отличаться значениями параметров (операций), но это не используется в данном примере. Определим по одной операции от каждого образца.

|  |
| --- |
| Операции, закладка OPR |
| **$Operations**  Приход\_клиента : Образец\_прихода\_клиента  Обслуживание\_клиента: Образец\_обслуживания\_клиента  **$End** |

Если в модели появляется хотя бы одна операция типа **operation**, то она будет запускаться параллельно себе самой до тех пор, пока истинно её условие **Choice from**. Т.е. для создания параллельных процессов нет необходимости плодить несколько операция от одного образца, достаточно всего одной. Но иногда это удобно использовать. Например: четыре операции перемещения по четырем направлениям создаются на основе одного образца, в котором описывается перемещению в любую сторону. При этом, конкретное направление задается через параметры каждой операции:

|  |
| --- |
| Пример: несколько операций от одного образца |
| Прокрутка\_Вправо: \_Прокрутка 'RIGHT' 1 0 -1  Прокрутка\_Влево : \_Прокрутка 'LEFT' 1 0 +1  Прокрутка\_Вверх : \_Прокрутка 'UP' 0 1 +1  Прокрутка\_Вниз : \_Прокрутка 'DOWN' 0 1 -1 |

Теперь рассмотрим используемые в модели функции. Они описываются на закладке FUN. В нашей модели были использованы две функции, точнее последовательности, для определения интервалов прихода между клиентами и длительности обслуживания. Первая последовательность представляет собой последовательность псевдослучайных чисел, распределенных по экспоненциальному закону, вторая – по нормальному. Это задается через ключевые слова **exponential** и **uniform** соответственно. После типа указывается база генератора, от которой зависит содержимое последовательности. Если базу не изменять, то получаемы последовательности псевдо случайных чисел будут одинаковыми, что позволяет повторить имитационных эксперимент в точности[[3]](#footnote-4). В зависимости от типа последовательности определяются параметры вызова. Для экспоненциальной – это одно число, означающие математическое ожидание, для нормальной – два числа, описывающих диапазон: минимальная и максимальная границы получаемых значений.

|  |
| --- |
| Последовательности, закладка FUN |
| **$Sequence** Интервал\_прихода : **real**  **$Type** = **exponential** 123456789  **$End**  **$Sequence** Длительность\_обслуживания : **real**  **$Type** = **uniform** 123456789  **$End** |

Одной из целью проведения имитационного эксперимента является сбор интересующих показателей. Как правило, это информация о загрузке узлов системы, по которой можно определить “узкое” место, пропускная способность, длина и/или время ожидания клиентов в очереди. Для подсчета загрузки используется показатель типа **watch\_state**, который наблюдает за состоянием системы. Интересующее состояние описывается через логическое выражение. В нашем случае: Парикмахерская.состояние\_парикмахера = Занят. Подсчет статистики ведется для тех моментов времени, когда значение логического выражения истинно. Для сбора статистики по очереди используется показатель типа **watch\_par**, который наблюдает за изменениями указанного параметра ресурса. Другие показатели в нашей модели не собираются, их значения вычисляются в самом конце моделирования и выводятся в отчет.

|  |
| --- |
| Показатели, закладка PMD |
| **$Results**  Занятость\_парикмахера : **watch\_state** Парикмахерская.состояние\_парикмахера = Занят  Длина\_очереди : **watch\_par** Парикмахерская.количество\_в\_очереди  Всего\_обслужено : **get\_value** Парикмахерская.количество\_обслуженных  Пропускная\_способность: **get\_value** Парикмахерская.количество\_обслуженных / **Time\_now** \* 60  Длительность\_работы : **get\_value** **Time\_now**  **$End** |

После расчета модели с указанными выше базами будут получены результаты:

|  |
| --- |
| Результаты моделирования, закладка Результаты |
| Занятость\_парикмахера **TRUE** 149 0.877315 136115 20.0211 39.9958  Длина\_очереди 3 303 2.75699 1.56872e+006 0 9  Всего\_обслужено 150  Пропускная\_способность 1.78177  Длительность\_работы 84.186 |

Результаты можно интерпретировать следующим образов:

* Занятость парикмахера : 0.877315 (т.е. 87.73%)
* Длина очереди : средняя 2.75699, минимальная 0, максимальная 9
* Всего обслужено : 150 клиентов
* Пропускная способность системы : 1.78 клиента в час
* Длительность работы системы : 84 часа

Более подробно см. справку по ключевому слову PMV (результаты моделирования).

Осталось рассмотреть содержимое закладки SMR, на которой указывается информация о содержимом модели, режимах моделирования и условие окончание работы модели.

|  |
| --- |
| Информация о прогоне, закладка SMR |
| **Model\_name** = barber\_1  **Resource\_file** = barber\_1  **OprIev\_file** = barber\_1  **Statistic\_file** = barber\_1  **Results\_file** = barber\_1  **Trace\_file** = barber\_1  **Show\_mode** = **NoShow**  **Show\_rate** = 3600.0  **Terminate\_if** **Time\_now** >= 12 \* 7 \* 60 |

В первых строчках указываются имена файлов модели. Далее идет описание режима анимации (**NoShow**) и скорости работы (3600.0). Внизу, после ключевого слова **Terminate\_if**, указывается логическое выражение, выполнение которого приводит к остановке модели. В данном случае: **Time\_now** >= 12 \* 7 \* 60, что означает остановку после моделирования одной 7‑ми дневной рабочей недели с длительность смены в 12‑ть часов. В качестве дополнений к приведенному примеру:

* Если файл трассировки закомментировать, то отключится вся трассировка в модели;
* **Show\_rate** указывает скорость анимации. Используется только в режиме анимации (**Show\_mode** = **Animation**). По своей сути является масштабным коэффициентом: показывает отношение реального времени к модельному. Значение, равное 1.0, означает, что одна единица модельного времени, например, **$Time** = 1, будет длиться 1 час реального времени. Это так называемая синхронная имитация, когда время модели синхронно реальному времени. Чем больше значение, тем быстрее прогон. Ограничивается вычислительными возможностями процессора, т.е. при больших значениях теряет смысл. Масштабный коэффициент 3600.0 означает, что одна единица модельного времени будет длиться 1 секунду реального.

1. Прицкер А. Введение в имитационное моделирование и язык СЛАМ II. –М.: Мир, 1987. [↑](#footnote-ref-2)
2. Емельянов В.В., Ясиновский С.И. Введение в интеллектуальное имитационное моделирование. Язык РДО. –М.: АНВИК, 1998. [↑](#footnote-ref-3)
3. На самом деле результаты могут различаться, если использовать разные версии одной и той же системы моделирования или еще по каким-то причинам. Но если одна и та же модель будучи подряд запущенной дважды без изменений выдала разные результаты, то впору задуматься о причинах и почитать документацию. К имитационному эксперименту необходимо относиться так же как и к натурному: если при одних и тех же исходных данных реакция системы изменяется, значит присутствуют неконтролируемые факторы. Возможно, система моделирования сама опционально назначает/изменяет базы генераторов. [↑](#footnote-ref-4)