

Олимпиада школьников «Шаг в будущее»
Научно-образовательное соревнование «Шаг в будущее, Москва»

регистрационный номер

Факультет «Робототехника и комплексная автоматизация»
Кафедра «Компьютерные системы автоматизации производства» (РК-9)

**«АНАЛИЗ ПРОПУСКНОЙ СПОСОБНОСТИ ШКОЛЬНОЙ СТОЛОВОЙ С
ПОМОЩЬЮ ИМИТАЦИОННОГО МОДЕЛИРОВАНИЯ»**

Автор:

Янишин Кирилл Романович,
ГОУ Лицей №1580 (при МГТУ им.
Баумана), 11 класс

Научный руководитель:

Урусов Андрей Витальевич, старший
преподаватель кафедры «Компьютерные
системы автоматизации производства»

Содержание

| | |
|---|----|
| Введение..... | 3 |
| Постановка проблемы..... | 4 |
| Цели и задачи..... | 4 |
| Объект моделирования..... | 5 |
| Сбор статистических данных..... | 6 |
| Описание состояния модели..... | 9 |
| Описание образцов..... | 11 |
| Событийный подход к построению имитационной модели..... | 13 |
| Связь системного и модельного времени..... | 15 |
| Сбор показателей..... | 15 |
| Проведение экспериментов..... | 19 |
| Выводы..... | 21 |
| Приложение..... | 21 |

Введение

«Сложные системы», «управление сложными системами», «системность», «бизнес процессы», «модели» - эти термины в настоящее время все чаще встречаются и используются практически в любой сфере деятельности человека. Причиной этого является обобщение накопленного опыта и результатов в различных сферах человеческой деятельности и естественное желание найти и использовать некоторые общесистемные принципы и методы. Именно системность решаемых задач в перспективе должна стать той базой, которая позволит работать исследователю с любой сложной системой, независимо от ее физической сущности. Именно модели и моделирование систем являются тем инструментом, которое обеспечивает эту возможность.

Имитационное моделирование (ИМ) на ЭВМ находит широкое применение при исследовании и управлении сложными дискретными системами (СДС) и процессами в них. К таким системам можно отнести экономические и производственные объекты, морские порты, аэропорты, комплексы перекачки нефти и газа, программное обеспечение сложных систем управления, вычислительные сети и многие другие. Моделирование применяется в случаях, когда проведение экспериментов над реальной системой невозможно или нецелесообразно: например, по причине хрупкости или дороговизны создания прототипа либо из-за длительности проведения эксперимента в реальном масштабе времени.

Имитационная модель — это компьютерная программа, которая описывает структуру и воспроизводит поведение реальной системы во времени. Имитационная модель позволяет получать подробную статистику о различных аспектах функционирования системы в зависимости от входных данных.

Постановка проблемы

Многие сложные системы представляют собой системы массового обслуживания (СМО). Для подобных систем характерны случайный входной поток клиентов, наличие приборов обслуживания, случайное время обслуживания и очереди, возникающие при ожидании клиентами обслуживания. Одним из примеров таких систем служит обычная школьная столовая, которая и была взята мною в качестве объекта моделирования.

Во время перемен в столовой наблюдается скопление очередей, состоящих из значительного числа клиентов. При этом в системе отсутствует наличие единой централизованной очереди, а это непременно сопровождается различного рода толчками и ругательствами, что вызывает крайне неприятные ощущения. Все это позволяет усомниться в правильности организации процесса обслуживания. Именно этим вопросом я и решил заняться с инженерной точки зрения.

Цели и задачи

Для решения данной проблемы я решил воспользоваться имитационной средой РДО. Преимущества данной среды моделирования заключаются в том, что в ней можно создавать не только модели процессов, но и системы управления ими. Дальнейшее проведение экспериментов в системе РДО позволяет проанализировать модель с измененными параметрами и предложить свой вариант решения поставленной проблемы.

Основная цель моего проекта заключается в создании адекватной (отражающей реальное состояние системы) модели, имитирующей работу школьной столовой для оценки эффективности организации процесса обслуживания.

Сформулированная цель определяет следующие задачи:

1. Выявление реальных предложений по уменьшению очередей

2. Определение рационального числа продавцов
3. Определение основных характеристик очереди, таких как:
 - максимальная длина
 - среднее время ожидания и др.
4. Моделирование должно обеспечиваться с учетом многих случайных факторов.

Объект моделирования

Объектом моделирования служит пищеблок Лицея №1580. Он подразделяется на буфет и столовую. Таким образом, объект моделирования представляет собой двухканальную СМО с двумя возможными очередями (рис. 1).

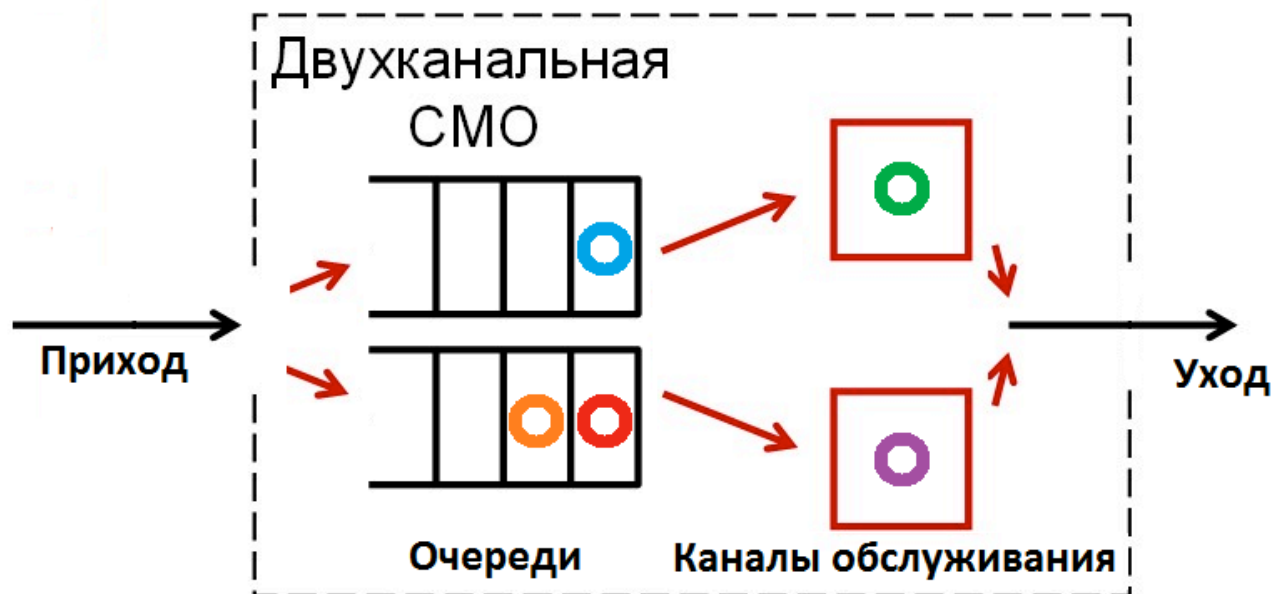


рис.1

Клиенты заходят в пищеблок, выбирают один из каналов обслуживания (в обоих из которых расположено по одному продавцу), после возможного ожидания обслуживаются и затем уходят.

Сбор статистических данных

В рамках подготовки исходных данных мною были проведены определенные замеры и собрана следующая статистика:

- Времени между приходами клиентов
- Направления клиентов (т.е. было определено, как часто клиент идет в столовую и в буфет)
- Времени обслуживания продавцом

В частности было замечено, что приход клиентов в систему не является регулярным (т.е. клиенты не приходят в пищеблок строго через какой-то определенный промежуток времени). Таким образом, мною было сделано предположение, что поток клиентов на входе представляет собой поток с некоторым случайным распределением. На рис. 2 представлена гистограмма прихода клиентов в пищеблок за первую переменную на основе собранных статистических данных.

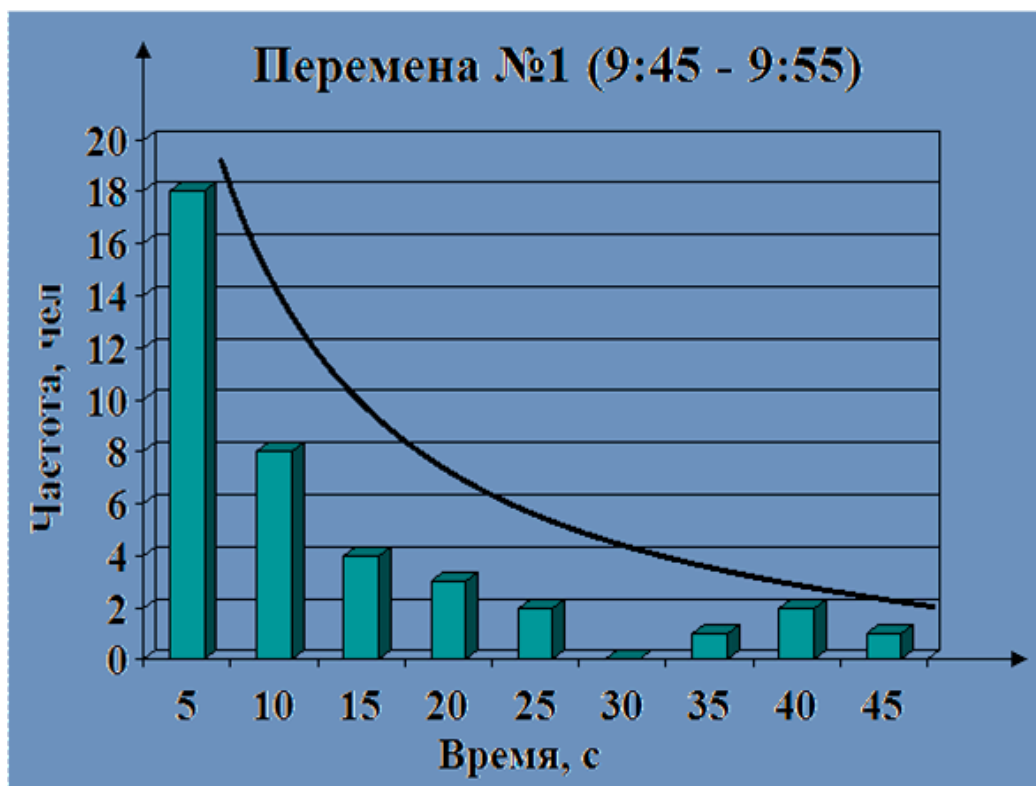


рис. 2

Как видно из графика, время прихода клиентов подчиняется экспоненциальному закону распределения случайных чисел.

Рис. 3 представляет собой гистограмму обслуживания продавцом, стоящим за прилавком буфета.



рис. 3

Из графика видно, что время обслуживания клиентов подчиняется равномерному закону распределения случайных чисел.

В дальнейшем было определено, что приход клиентов подчиняется экспоненциальному закону вне зависимости от номера перемены. А равномерный закон обслуживания также характерен и продавцу столовой. Все эти выводы легли в основу разработанной программы.

В РДО имеется возможность моделировать потоки с различными законами распределения. Для этого в объекте констант, функций и последовательностей модели должны быть описаны соответствующие последовательности.

В моей программе описания последовательностей имеет следующий вид:
\$Sequence Равномерный_закон: integer

\$Type = uniform 123456789

\$End

\$Sequence Экспоненциальный_закон: real

\$Type = exponential 123456789

\$End

Описание последовательности начинается с зарезервированного слова **\$Sequence**, за которым следует ее имя. После имени через двоеточие помещается тип значения последовательности.

Последовательности имеют следующие типы (**\$Type =**):

- **uniform** – последовательность псевдослучайных чисел, распределенных по равномерному закону;
- **exponential** – последовательность псевдослучайных чисел, распределенных по экспоненциальному закону;

Обращение к последовательности осуществляется по ее имени. Примеры вызовов последовательностей:

Равномерный_закон (18.12, 34.35);

Экспоненциальный_закон (13.43);

Здесь, для равномерного закона в скобках указываются границы нижнего и верхнего диапазона времени обслуживания клиентов. В моей модели времена обслуживания клиентов для продавцов будут равномерно распределены в интервале 18.12 – 34.35 секунд. Для экспоненциального закона в скобках указывается математическое ожидание, значение которого, я вычислял на основе собранных данных. Математические ожидания для каждой перемены представлены ниже.

| Номер и время перемены | Математическое ожидание прихода клиентов (λ) |
|---------------------------|---|
|---------------------------|---|

| | | |
|---------------------------|---------------|---------------|
| <i>Открытие</i> | 8.30 – 9.00 | 163.63 |
| <i>Перемена №1</i> | 9.45 – 9.55 | 15.00 |
| <i>Перемена №2</i> | 10.40 – 10.55 | 13.43 |
| <i>Перемена №3</i> | 11.40 – 12.00 | 15.78 |
| <i>Перемена №4</i> | 12.45 – 13.15 | 18.94 |
| <i>Перемена №5</i> | 14.00 – 14.15 | 18.75 |
| <i>Перемена №6</i> | 15.00 – 15.10 | 26.08 |

В реальной системе, после того как прозвонит звонок на урок, обслуживание клиентов прекращается. Также следует отметить, что если в пищеблоке, в момент начала урока все еще присутствуют клиенты, то они будут удалены из него, только после принятия пищи.

Описание состояния модели

Исходя из понятий РДО, вначале определим ресурсы модели, т.е. те элементы пищеблока, которые участвуют в процессе обслуживания клиентов и которые должны быть обслужены. В качестве таковых использованы сам пищеблок, клиенты и продавцы. Остановимся более подробно на каждом из них.

1) Пищеблок.

Первый тип ресурса, определенного в модели, имеет имя ***Пищеблоки***. Это постоянный ресурс, т.е. он всегда присутствует в модели. Данный тип ресурса имеет два параметра целого типа (типа integer), которым присваиваются имена (служашие идентификаторами) – ***Количество_зашедших_клиентов*** и ***Количество_ушедших_клиентов***, и один параметр перечислимого типа – ***Местоположение***. Они описываются в модельном объекте с именем **dining_room.rtp** (рис.4).

```

$Resource_type Пищевблоки : permanent
$Parameters
    Количество_зашедших_клиентов: integer
    Местоположение                : (Столовая, Буфет) = Столовая
    Количество_ушедших_клиентов : integer
$End

```

рис.4

В модели присутствует лишь один ресурс данного типа. Он имеет имя *Пищевблок*, начальное значение его параметров описывается в объекте *dining_room.rss*. Сам объект показан на рис.5 (красным цветом выделено описание данного ресурса). При описании ресурсов можно указать признак трассировки. Трассировка служит для сбора информации об изменениях состояний ресурсов. Этот признак задается зарезервированным словом **trace**.

```

$Resources
    Пищевблок : Пищевблоки trace 0 * 0
    Продавец_1: Продавцы Столовая *
    Продавец_2: Продавцы Буфет *
    Timeinfo   : TypeTimeinfo 1 30600.0 0
$End

```

рис.5

2) Клиенты

Второй тип ресурса, определенного в модели, имеет имя *Клиенты*. Это временный ресурс, т.е. он может быть создан во время прогона модели и уничтожен при выполнении операций, правил и совершении нерегулярных событий. Данный тип ресурса имеет три параметра перечислимого типа: *Направление*, *Состояние*, *Размещение*, *Время_прихода* и *Время_ожидания* (рис. 6).

```

$Resource_type Клиенты : temporary
$Parameters
    Направление      : such_as Пищевблоки.Местоположение
    Состояние        : (Пришел, Ест, Поел, В_очереди, Покупает, Купил, Разговаривает, Ушел)
    Размещение       : (Стол_1, Стол_2, Стол_3, Стол_4, Стол_5, Стол_6, Еще_нету)
    Время_прихода    : real
    Время_ожидания   : real
$End

```

рис. 6

Этот тип ресурса создается во время прогона модели, следовательно, начальное значение его параметров описано быть не должно.

3) Продавцы

Третий тип ресурса, определенного в модели, имеет имя *Продавцы*. Это постоянный ресурс. Данный тип ресурса имеет два параметра перечислимого типа – *Расположение* и *Занятость_продавца*. (рис.7).

```
$Resource_type Продавцы : permanent
$Parameters
    Расположение      : such_as Пищевые.Местоположение
    Занятость_продавца: (Обслуживает, Свободен) = Свободен
$End
```

рис. 7

В модели присутствует два ресурса данного типа – *Продавец_1* (продавец столовой) и *Продавец_2* (продавец буфета). На рис.5 синим цветом выделено их описание.

Описание образцов

Приведенные выше объекты описывают состав ресурсов моделируемого пищеблока. После того, как ресурсы описаны, необходимо формализовать закономерности их взаимодействия при выполнении действий, происходящих в моделируемой системе. Для этого служат такие объекты, как образцы и события. Остановимся более подробно на одном из образцов моей модели – операции обслуживания клиентов.

Операция обслуживания клиентов моделируется с помощью образца с именем *Образец_Обслуживание_клиента*, описывающего действие протяженное во времени и имеющее два события – начала и конца (т.е. имеющего тип **operation**). Образец находится в объекте **dining_room.pat**. Описание образца имеет представлено на рис. 8.

Рассмотрим более подробно конструкцию данного образца.

Описание начинается со служебного слова **\$Pattern**, за которым через двоеточие записывается тип данного образца. Служебное слово **trace**, следующее за типом образца, является признаком трассировки событий.

С образцом связано два релевантных ресурса, с именами *_Клиент* и *_Продавец*. За именем релевантного ресурса после двоеточия следует описатель

релевантного ресурса. В качестве описателя для релевантного ресурса, с именем **_Клиент** указан ресурс **Клиенты**, а для релевантного ресурса, с именем **_Продавец** указан ресурс **Продавцы**, что означает, что в процессе операции могут использоваться лишь эти ресурсы. Следом за описателем релевантного ресурса следует два статуса конвертора – конвертор начала и конвертор конца. Статус конвертора начала описывает, что происходит в начале, статус конвертора конца – в конце операции. Т.к. оба из ресурсов не исчезают и не образуются в начале и конце операции, то статусы конверторов их образцов одинаковые – **keep**.

```
$Pattern Образец_Обслуживание_клиента : operation
$Relevant_resources
    _Клиент : Клиенты keep keep
    _Продавец: Продавцы keep keep
    with_min _Клиент.Время_прихода
$Time = Равномерный_закон (18.12, 34.35)
$Body
    _Клиент
        Choice from _Клиент.Состояние == В_очереди
        Convert_begin
            Состояние = Покупает;
        Convert_end
            Состояние = Купил;
            Время_ожидания = time_now - _Клиент.Время_прихода;

    _Продавец
        Choice from _Продавец.Занятость_продавца == Свободен and
            _Продавец.Расположение == _Клиент.Направление
        Convert_begin
            Занятость_продавца = Обслуживает;
        Convert_end
            Занятость_продавца = Свободен;
$End
```

рис.8

Строка **\$Time = Равномерный_закон (18.12, 34.35)** образца, представляет собой выражение, определяющее время обслуживания клиента.

Тело приведенного образца начинается с зарезервированного слова **\$Body**. Тело содержит два имени релевантных ресурсов – **_Клиент** и **_Продавец** и правила их использования. Правила использования состоят из предусловий, начинающихся зарезервированными словами **Choice from** и двух конверторов, начинающихся зарезервированными словами **Convert_begin** и **Convert_end**.

Предусловия образца содержат логические выражения. Обслуживание клиента возможно, если имеются клиенты в очереди (**_Клиент.Состояние == В_очереди**), одновременно с этим продавец свободен (**_Продавец.Занятость_продавца == Свободен**) и расположение продавца совпадает с направлением клиента (**_Продавец.Расположение == _Клиент.Направление**).

Конвекторы начала данного образца показывают, что события начала соответствующей операции приводят к изменению параметров **Состояние** и **Занятость_продавца**.

Конвектор конца данного образца показывает, что в результате реализации события окончания операции параметр ресурса **Состояние** принимает значение «**Купил**», параметр ресурса **Занятость_продавца** принимает значение «**Свободен**», а параметр ресурса **Время_ожидания** принимает значение «**time_now - _Клиент.Время_прихода**».

Событийный подход к построению имитационной модели

При описании функционирования СДС могут быть использованы события, действия и процесс. Выделяют три альтернативных подхода к построению ИМ: событийный, подход сканирования активностей и процессно-ориентированный. Моя модель основана на событийном подходе. Рассмотрим его более подробно, для лучшего понимания работы ИМ.

Событие происходит мгновенно, т. е. это некоторое действие с нулевой длительностью (рис. 9).

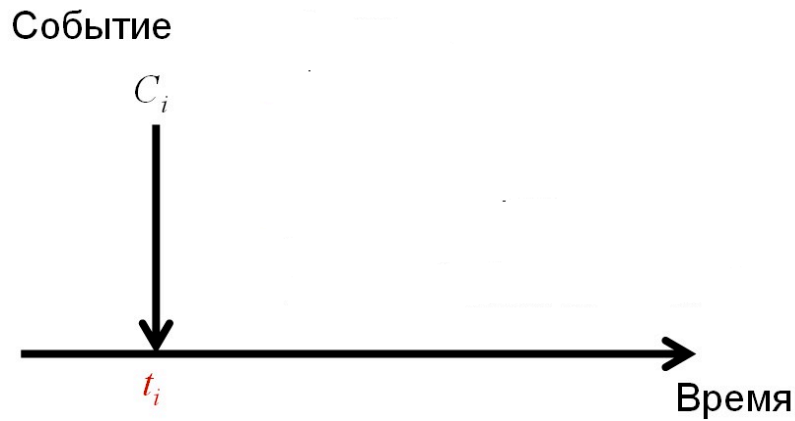


рис. 9

При событийном подходе описываются события, которые могут изменять состояние системы и определяются логические взаимосвязи между ними. Имитация происходит путем выбора из списка будущих событий ближайшего по времени и его выполнении. Выполнение события приводит к изменению состояния системы и генерации будущих событий, логически связанных с выполняемым. Эти события заносятся в список будущих событий и упорядочиваются в нем по времени наступления. Алгоритм выполнения событий в ИМ показан на рис.10.



рис.10

Также следует отметить, что имитационное время продвигается от события к событию и между событиями состояние модели не изменяется.

Связь системного и модельного времени

Одной из основных проблем, с которыми мне пришлось столкнуться, в ходе построения имитационной модели – являлась связь системного (реального) и модельного времени. Режим работы пищеблока – с 8:30 до 15:30. Мною было принято решение, что не стоит моделировать то время, когда столовая закрыта. Т.е. к 15:30 первого дня я присваивал 8:30 второго и т.д. (рис. 11).

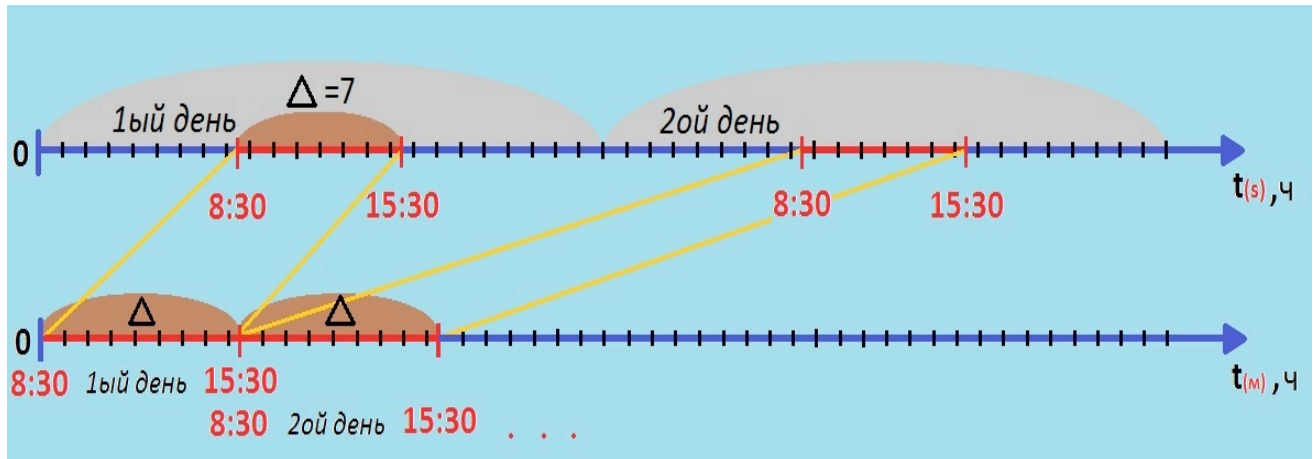


рис. 11

Для связи этих двух времен были выведены следующие формулы:

1) $K(T) = \text{int}(T/7) + 1$, где K – номер дня, а T – это произвольный момент времени.

Результатом функции $\text{int}(x)$ является наибольшее целое число, не превосходящее x .

$$2) t_{(s)} = t_{(m)} (8:30 + (T - 7 * (K - 1)) + 24 * (K - 1));$$

$$3) t_{(m)} = t_{(s)} ((K - 1) * 7 + T - (8:30 + 24 * (K - 1)));$$

Данные формулы были использованы в модели для осуществления перехода к новому рабочему дню.

Сбор показателей

Трассировка содержит подробную информацию о ходе имитации, включая все события, происходящие в модели и временах этих событий, все изменения состояния ресурсов и времена этих изменений. Она может использоваться для

отладки модели, однако, в процессе имитационных исследований обычно требуется получить не данные об изменениях состояния системы в течение прогона, а набор показателей, связанных с этими изменениями, то есть результирующие статистические данные, характеризующие имитируемую систему.

В результате моделирования пищеблока, были собраны такие показатели, как занятость продавцов столовой и буфета, количество клиентов в очереди в столовую и буфет и время клиентов в очереди в столовую и в буфет.

Показатели, которые требуется собрать в процессе прогона модели, описываются в отдельном объекте – **dining_room.pmd** (рис. 12).

```
$Results
    Занятость_продавца_столовой          : watch_state
    Продавец_1.Занятость_продавца == Обслуживает
    Занятость_продавца_буфета            : watch_state
    Продавец_2.Занятость_продавца == Обслуживает
    Количество_клиентов_в_очереди_в_столовую : watch_quant
    Клиенты (Клиенты.Направление == Столовая) and
    (Клиенты.Состояние == В_очереди or Клиенты.Состояние == Покупает)
    Количество_клиентов_в_очереди_в_буфет   : watch_quant
    Клиенты (Клиенты.Направление == Буфет) and
    (Клиенты.Состояние == В_очереди or Клиенты.Состояние == Покупает)
    Время_клиентов_в_очереди_в_столовую      : watch_value
    Клиенты Клиенты.Направление == Столовая Клиенты.Время_ожидания
    Время_клиентов_в_очереди_в_буфет         : watch_value
    Клиенты Клиенты.Направление == Буфет Клиенты.Время_ожидания
$End
```

рис. 12

В моей модели собираются показатели следующих видов:

- **watch_state** – наблюдать состояние системы;
- **watch_quant** – наблюдать количество временных ресурсов;
- **watch_value** – наблюдать значение параметра временного ресурса в момент уничтожения ресурса

РДО – имитатор создает в результате прогона объект результатов (рис. 13), имеющий расширение **dining_room.pmv**.

```
Занятость_продавца_столовой    FALSE    779 0.115576    550083 18.1241 34.2901
Занятость_продавца_буфета      FALSE    1585    0.236661    1.13637e+006 18.1248 34.3466
Количество_клиентов_в_очереди_в_столовую  0    1559    0.196664    1.6198e+006 0    7
Количество_клиентов_в_очереди_в_буфет    0    3171    1.12072 2.92383e+007 0    21
Время_клиентов_в_очереди_в_столовую  779 44.5335 25.7531 0.922702    18.3007 157.25
Время_клиентов_в_очереди_в_буфет    1585    124.729 91.2134 2.2911    18.139 565.154
```


рис. 13

Для показателей **watch_state** и **watch_quant** в строке результатов выводятся:

- Имя показателя – имя данного показателя, описанного в объекте показателей.
- Текущее значение – это значение логического выражения для данного показателя в момент окончания моделирования. TRUE – ИСТИНА, FALSE – ЛОЖЬ.
- Количество наблюдений – это число раз, когда логическое выражение приобретало значение ИСТИНА, а затем - ЛОЖЬ.
- Среднее значение – это отношение времени, в течение которого состояние системы на интервале сбора удовлетворяло условию, к длительности интервала сбора показателя.
- Сумма квадратов – это сумма квадратов длительностей интервалов, в течение которых состояние системы удовлетворяло условию.
- Минимальное значение, максимальное значение – это минимальная и максимальная длительность интервала за время наблюдения.

Таким образом, в момент окончания моделирования (моделировалось 7 дней работы пищеблока) продавец столовой обслужил **779** человек, продавец буфета – **1585** человек. Минимальное время обслуживания продавца столовой составило **18.1241** секунд, продавца буфета – **18.1248** секунд. Максимальное время обслуживания продавца столовой – **34.2901** секунд, продавца буфета – **34.3466** секунд. Максимально зафиксированная в процессе моделирования очередь в столовую состояла из **7** человек, в буфет – из **21** человека. Коэффициент занятости (η) продавца столовой составил **0.115576**, продавца буфета – **0.236661**. Однако следует учесть, что данный коэффициент рассчитывался для всего времени моделирования, нас же интересует информация о занятости продавцов непосредственно во время перемен. Для этого необходимо воспользоваться следующей формулой:

$$\eta' = \eta \cdot \frac{T}{T'}$$

Здесь, η' – коэффициент занятости продавца во время перемен;

T – длительность всего моделирования;

T' – длительность всех перемен;

$$T' = T_1 + T_2 + T_3 + \dots + T_n = \sum_{i=1}^n T_i$$

В итоге, средняя занятость продавца столовой во время перемен при данном прогоне составила **48, 54%**, продавца буфета – **99, 39%**.

Для показателя **watch_value** выводятся:

- Имя показателя – имя данного показателя, описанного в объекте показателей

· Число наблюдений – это число уничтоженных временных ресурсов, состояние которых удовлетворяло условию.

· Среднее значение показателя – это сумма наблюдений, деленная на число наблюдений.

· Среднеквадратическое отклонение – это корень квадратный из суммы квадратов разностей среднего значения и каждого наблюдения, деленной на число наблюдений минус 1.

· Отклонение среднего – это среднеквадратичное отклонение, деленное на корень квадратный из числа наблюдений.

· Минимальное значение, максимальное значение – это минимальное и максимальное значения арифметического выражения за время наблюдения.

Эти результаты интерпретируются следующим образом: в момент окончания моделирования столовую посетило **779** человек, буфет – **1585** человек. Среднее время ожидания в очереди в столовую составило **44.5335** секунд, в буфет – **124.729** секунд, минимальное время ожидания в очереди в столовую – **18.3007** секунд, в буфет – **18.139** секунд, максимальное время ожидания в очереди в столовую – **157.25** секунд, в буфет – **565.154** секунд.

Однако, для получения более точной статистики, необходимо провести не один прогон модели, а несколько (мною было проведено 20), каждый раз изменяя значение базы генератора. Оно представляет собой целое число в диапазоне -2147483648...2147483647. Изменение базы генератора позволяет получать различные выборки случайной последовательности чисел.

Таким образом, были получены следующие окончательные результаты для времени, когда идут перемены (т.к. нас не интересует все время моделирования):

- Максимальная длина очереди в столовую составила **11** человек
- Максимальная длина очереди в буфет составила **26** человек
- Средняя длина очереди в столовую составила **0.79** человек
- Средняя длина очереди в буфет составила **4.24** человек
- Среднее время ожидания в очереди в столовую составило **43.132** секунд
- Среднее время ожидания в очереди в буфет составило **115.479** секунд
- Средняя занятость продавца столовой составила **47, 57%**
- Средняя занятость продавца буфета составила **96, 76%**

Проведение экспериментов

Одной из основных задач моей работы, как уже указывалось ранее, является поиск предложений по уменьшению очередей в пищеблоке. Будем анализировать выдаваемую статистику, изменяя некоторые параметры модели.

1) При добавлении еще одного продавца в буфет, модель выдает следующие результаты (следует отметить, что значения всех показателей для продавца столовой не изменятся):

- Максимальная длина очереди в буфет составила **9** человек
- Средняя длина очереди в буфет составила **0.98** человек
- Среднее время ожидания в очереди в буфет составило **31.732** секунд

- Средняя занятость первого продавца буфета составила **58,15%**
- Средняя занятость второго продавца буфета составила **38,91%**

2) Теперь проанализируем модель, в случае совместной работы двух продавцов (т.е. предположим, что пищеблок уже не разделяется на столовую и буфет, а представляет собой единое место обслуживания с двумя продавцами). Для этого пришлось немного изменить код программы. Результаты моделирования следующие:

- Средняя длина очереди составила **1.411** человек
- Среднее время ожидания в очереди составило **45.9184** секунд
- Средняя занятость первого продавца составила **78,86%**
- Средняя занятость второго продавца составила **65,86%**

На рис. 15 показано среднее значение времени ожидания в очереди.

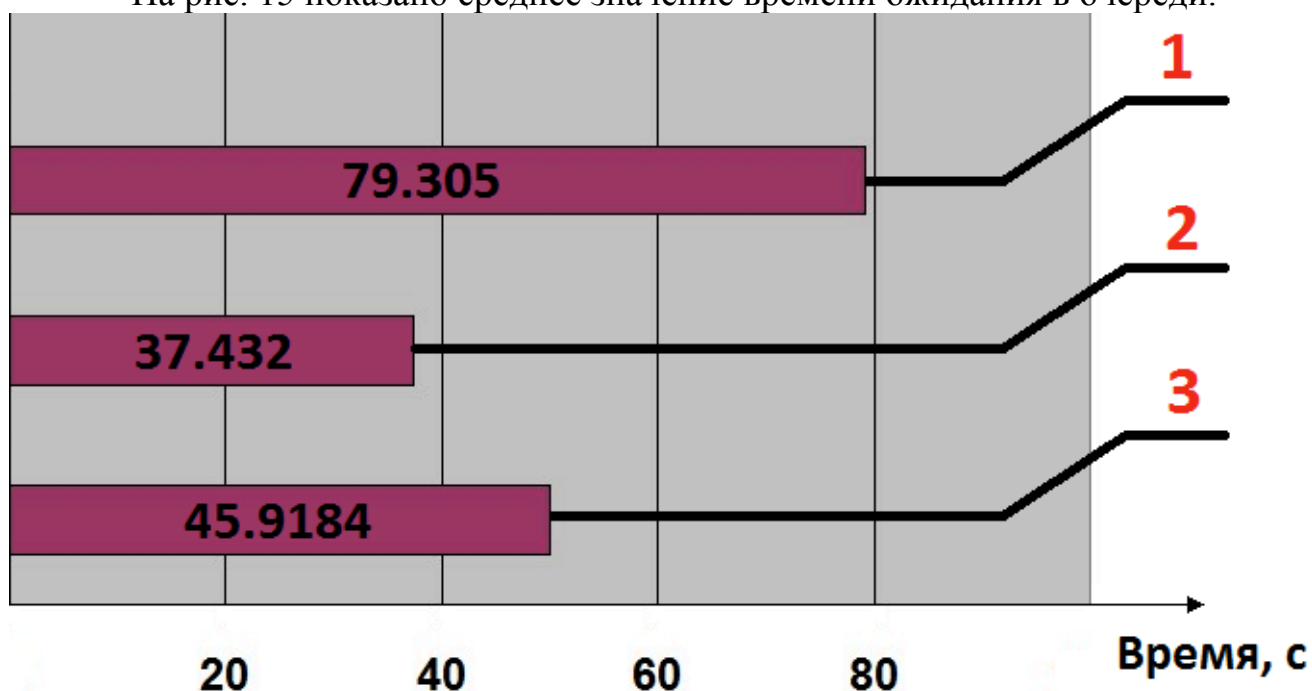


рис. 15

Здесь, и в дальнейшем, под цифрой 1 подразумевается первоначальная система (без внесения каких – либо изменений); под цифрой 2 – система после добавления еще одного продавца в буфет; под цифрой 3 – система, при совместной работе двух продавцов.

Таким образом, результаты моделирования показали, что при добавлении еще одного продавца в буфет, среднее время ожидания в очереди сократится на **52,79%**, а при совместной работе двух существующих продавцов на **42,09%**.

На рис. 16 представлено значение такого показателя, как *средняя занятость продавцов*.

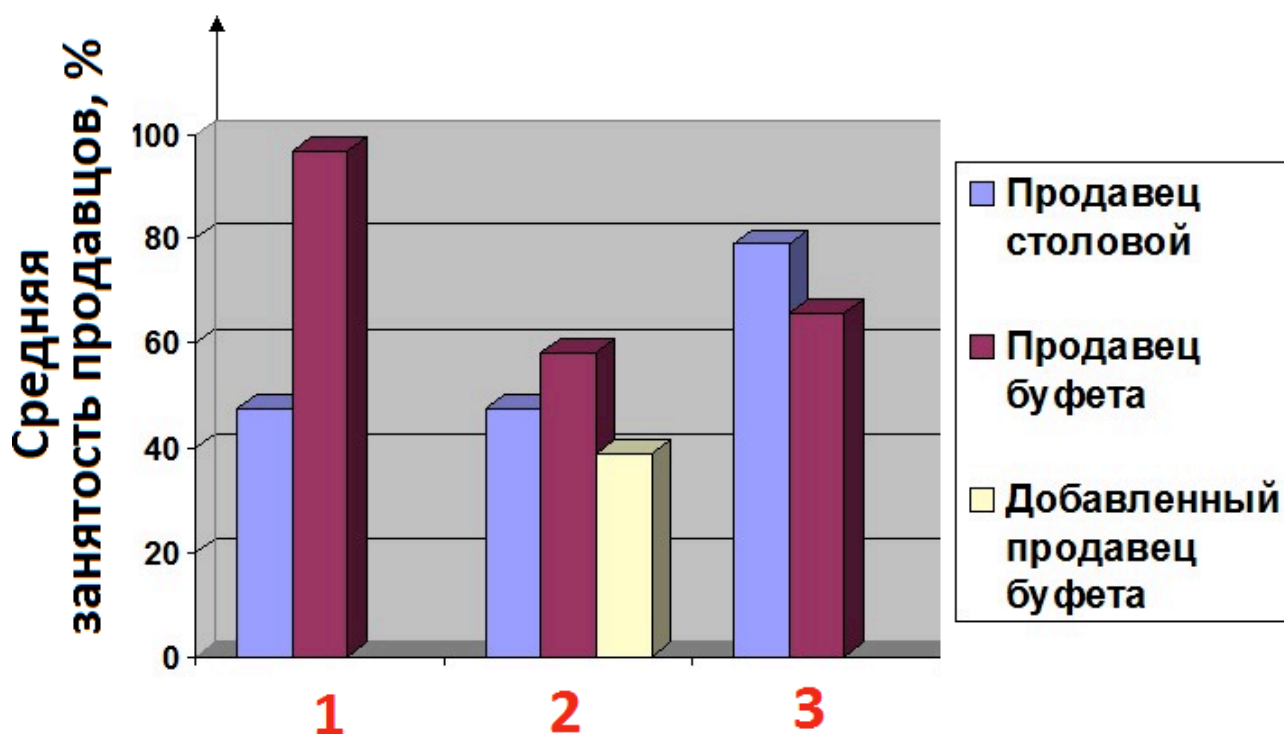


рис. 16

На рис. 17 представлено значение такого показателя, как *средняя длина очереди*.

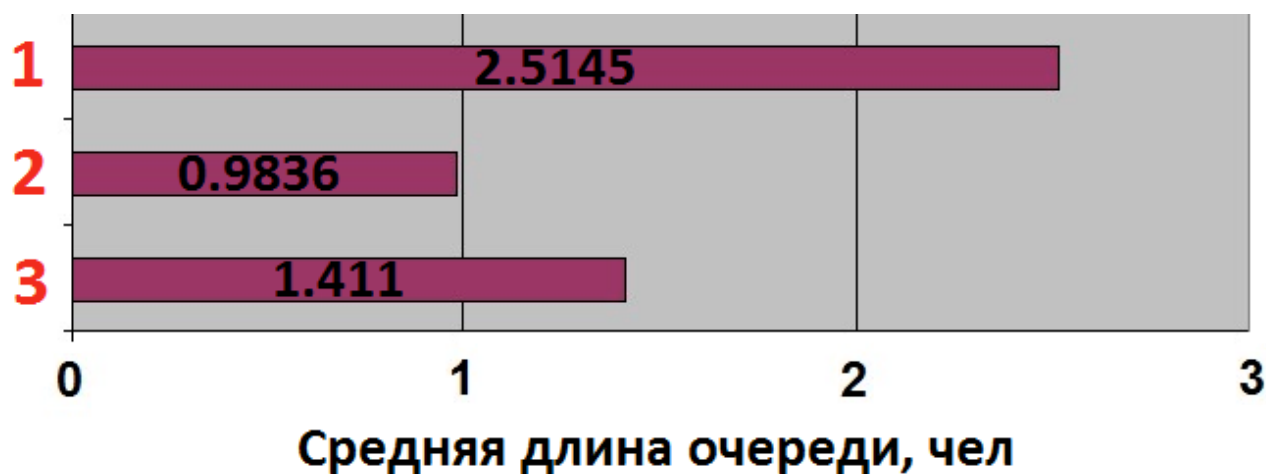


рис. 17

Таким образом, результаты моделирования показали, что при добавлении еще одного продавца в буфет, средняя длина очереди сократится на **60, 88%**, а при совместной работе двух существующих продавцов на **43, 88%**.

Выводы

В результате проведенной работы, была создана модель на языке имитационного моделирования РДО, отражающая реальное поведение системы – школьного пищеблока. Данная модель работает на основе собранных статистических данных. После проведения ряда тестов, был сделан вывод, что она работает адекватно. Был проведен полный анализ пропускной способности системы. Проведение над моделью серии экспериментов показало, что

уменьшению очередей, возникающих в системе, можно способствовать двумя различными способами. Первый из них – добавление еще одного продавца за прилавок буфета. Это можно осуществить, расширив штаб сотрудников пищеблока. Второй вариант, менее затратный, – создать единое место обслуживания с двумя существующими продавцами. Это можно осуществить, в случае незначительного изменения планировки пищеблока.

Приложение

К данной работе прилагается диск с версией программы. Для ее просмотра необходимо выполнить следующие действия:

- 1) открыть архив **RAO-studio-bin-5600**;
- 2) установить файл **RAO-studio.exe**, после чего нажать **файл – открыть – DVD RW дисковод (D:) Янишин Кирилл – dining room – dining_room**;
- 3) для включения анимации необходимо нажать **F5**.