Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное

учреждение высшего образования

«Севастопольский государственный университет»

Институт информационных технологий и управления в технических системах

Кафедра ИС

ОТЧЁТ

По лабораторной работе №3

«Исследование технологии дискретно-событийного имитационного моделирования»

Выполнил:

Ст. гр. ИС/б-20-1-о

Хроменко Д.А.

Проверил:

Хохлов В.В.

Севастополь

2022

**3.1 Цель работы**

Исследование технологии дискретно-событийного имитационного моделирования. Изучение базовых блоков программы моделирования Anylogic и получение практических навыков программирования имитационных моделей.

**3.2 Ход работы**

Были запрограммированы имитационные модели в Anylogic с помощью библиотеки моделирования процессов согласно варианту.

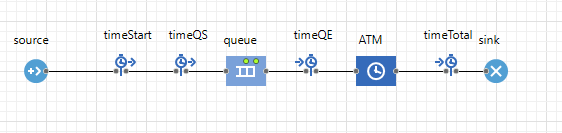


Рисунок 1 – Имитационная модель для задачи № 1

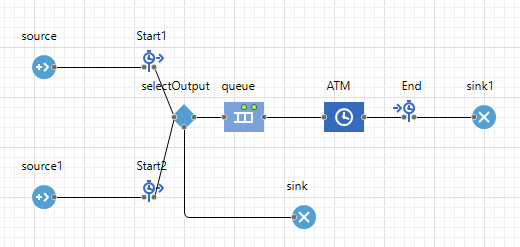


Рисунок 2 – Имитационная модель для задачи №2

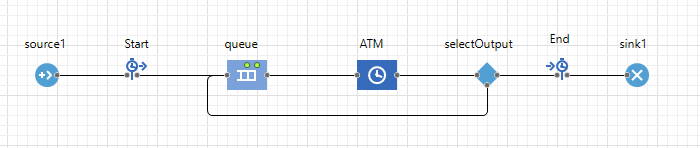


Рисунок 3 – Имитационная модель для задачи №3

В моделях был организован вывод статистики: были выведены гистограммы среднего времени пребывания заявки в системе, загрузки системы, средней длины очереди и среднего времени ожидания заявки.

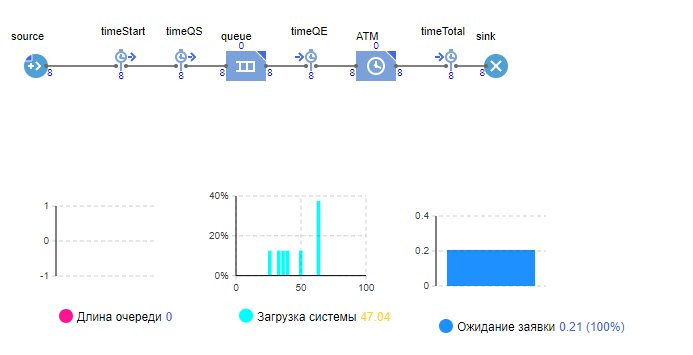


Рисунок 4 – Вывод статистики для задачи №1

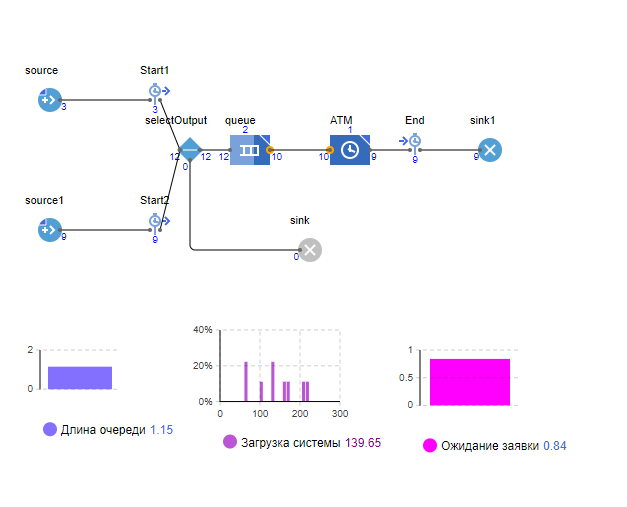


Рисунок 5 – Вывод статистики для задачи №2

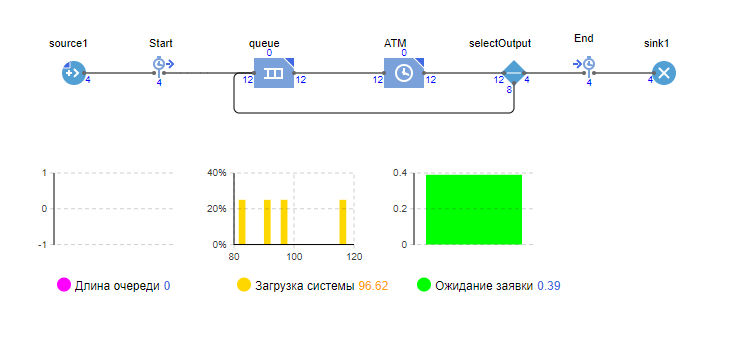


Рисунок 6 – Вывод статистики для задачи №3

Были запрограммированы имитационные модели в Anylogic с помощью библиотеки диаграмм состояний.

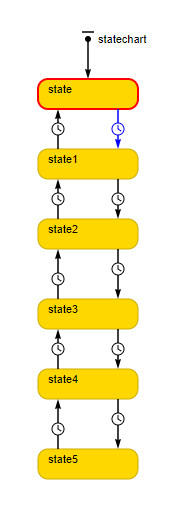


Рисунок 7 – Диаграмма состояний для задачи №1

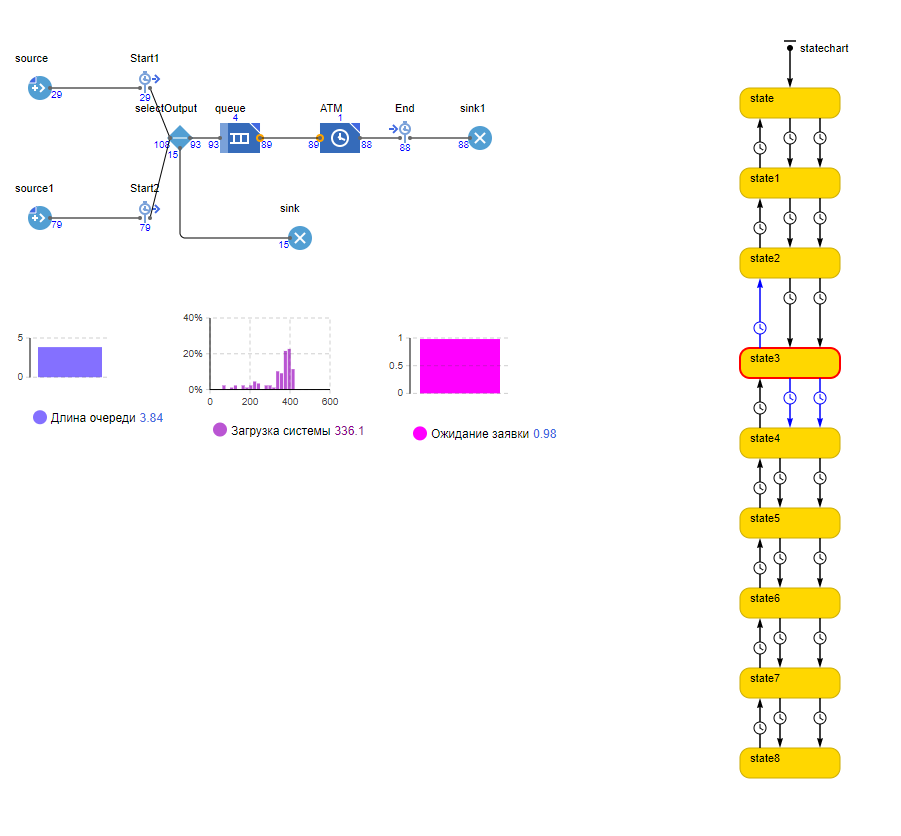


Рисунок 8 – Диаграмма состояний для задачи №2

**Выводы**

В ходе выполнения данной лабораторной работы были исследованы технологии дискретно-событийного имитационного моделирования. Изучены базовые блоки программы моделирования Anylogic и получены практические навыки программирования имитационных моделей.

**Ответы на контрольные вопросы**

1. Библиотека моделирования процессов позволяет моделировать сложные системы, разбивая их на мелкие части и представляя в виде отдельных компонентов (подпроцессов). На верхнем уровне можно соединить эти компоненты в одну систему.

2. Если у агента можно выделить несколько состояний, выполняющих различные действия при происхождении каких-то событий, или если у агента есть несколько качественно различных поведений, последовательно сменяющих друг друга при происхождении определенных событий, то поведение такого объекта может быть описано в терминах диаграммы состояний. Диаграмма состояний позволяет графически задать пространство состояний алгоритма поведения объекта, а также события, которые являются причинами срабатывания переходов из одних состояний в другие, и действия, происходящие при смене состояний.

С помощью диаграмм состояний можно графически задать дискретные поведения объектов любой сложности, куда более разнообразные, чем элементарные состояния свободен/занят (idle/busy), открыт/закрыт (open/closed), исправен/неисправен (up/down) и т.п., предлагаемые большинством блочных инструментов моделирования. Диаграммы состояний AnyLogic соответствуют стандарту UML. Они сохраняют графический вид, атрибуты и семантику выполнения, определенную в UML.

3. Объект сбора статистики Статистика вычисляет основную статистическую информацию (среднее значение, минимум, максимум и т.д.) для последовательности измеренных значений. Объект работает по-разному в зависимости от того, считаются ли собираемые данные непрерывными или дискретными.

При работе с непрерывными данными, измерения имеют длительность во времени, то есть значение остается актуальным в течение какого-то времени, и изменяется только в дискретные моменты времени (например, длина очереди). В объект непрерывной статистики можно добавлять только измерения с возрастающими значениями временных меток. Методы вычисления среднего значения и т.д. подразумевают, что последнее добавленное значение было актуально до времени вызова метода. Собранная объектом статистика в этом случае доступна посредством программного интерфейса класса StatisticsContinuous.

При работе с дискретными данными, измерения не имеют длительности во времени (например, цена продуктов или длительность пребывания пациентов). Они считаются изолированными, дискретными точками во времени, так что среднее значение является просто суммой измерений, поделенной на их общее количество. Собранная объектом статистика в этом случае доступна посредством программного интерфейса класса StatisticsDiscrete.

4. Заявки, транзакты, или же сообщения являются абстрактными подвижными элементами, которые моделируют объекты реального мира: клиентов, покупателей, сообщения, программы, сбои.

Каждый транзакт занимает некоторый объем памяти ЭВМ. По­сле того, как он закончит свое движение по блокам модели, его необходимо уничтожать для освобождения памяти, чтобы избежать ее переполнения. Поскольку транслятору не известно, сколько транзактов одновременно будет находиться в модели, то память под транзакты выделяется динамически.

Таким образом, при начале моделирования в GPSS -модели не существует ни одного транзакта. В процессе моделирования транзакты входят в модель в определенные моменты времени, соответствующие логике функционирования моделируемой системы. Таким же образом транзакты покидают модель в зависимости от специфики моделирования. В общем случае в модели существует несколько транзактов, но в каждый момент времени движется только один из них.

Если транзакт начал свое движение, он передвигается от блока к блоку по пути, указанному блок-схемой (логикой работы модели). В тот момент, когда транзакт входит в блок, вызывается соответствующая этому блоку подпрограмма. Далее транзакт (в общем случае) пытается войти в следующий блок. Его перемещение продолжается до тех пор, пока не выполнится одно из таких возможных условий:

1. Транзакт входит в блок, функцией которого является задержка транзакта на определенное время.

2. Транзакт входит в блок, функцией которого является удаление транзакта из модели.

3. В соответствии с логикой модели транзакт пытается войти в следующий блок, но блок не принимает этот транзакт. В этом случае транзакт остается в том блоке, в котором в данное время находится, но позже будет повторять попытки войти в следующий блок. Когда условия в модели изменятся, одна из таких попыток может быть успешной. После этого транзакт продолжит свое перемещение по моде­ли.

Если выполняется одно из указанных условий, транзакт остается на месте и в модели делается попытка перемещения другого транзакта.

5. Продукт получил название AnyLogic, потому что он поддерживал все три известных метода моделирования: системная динамика; дискретно-событийное моделирование, дискретно-событийное (процессное) моделирование; агентное моделирование.

6. Объект Source генерирует заявки определенного типа. Обычно он используется в качестве начальной точки диаграммы процесса, формализующей поток заявок. Например, заявки могут быть запросами на обработку сервером, а объект Source будет моделировать их поступление.

Объект Queue моделирует очередь заявок, ожидающих приема объектами, следующими за данным в диаграмме процесса. В нашем случае он будет моделировать очередь запросов, ожидающих освобождения сервера.

Объект Delay задерживает заявки на заданный период времени. В нашей модели это сервер, обрабатывающий запросы.

Объект Seize отвечает за захват ресурсов.

Объект Release отвечает за освобождение ресурсов.

Объект Sink уничтожает поступившие заявки. Обычно он используется в качестве конечной точки потока заявок (и диаграммы процесса соответственно).

Объект RecourcePool задает ресурсы определенного типа.

Объект SelectOutput производит проверку счетчика числа повторений (циклов) для каждой заявки.

Объект Split используется для создания копии заявки.

Объект Combine используется для объединения двух заявок в одну.

Объект Service моделирует занятие заявкой ресурса на определенное время.

7. Объекты Библиотеки моделирования процессов самостоятельно производят сбор основной статистики. Для активации сбора статистики требуется в свойствах объекта установить флажок «Включить сбор статистики» на интересующем объекте.

8.

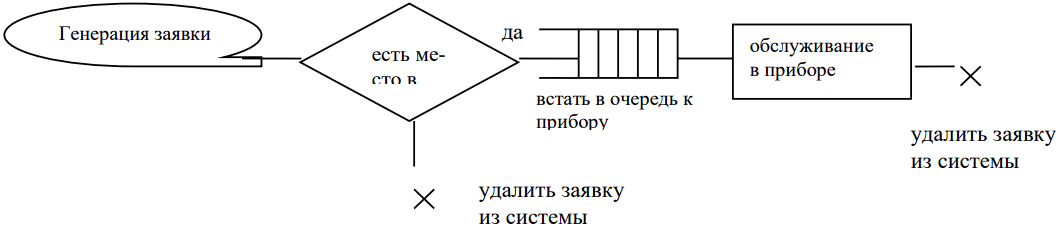


Рисунок 1 – Схема функционирования модели системы с потерями

9.

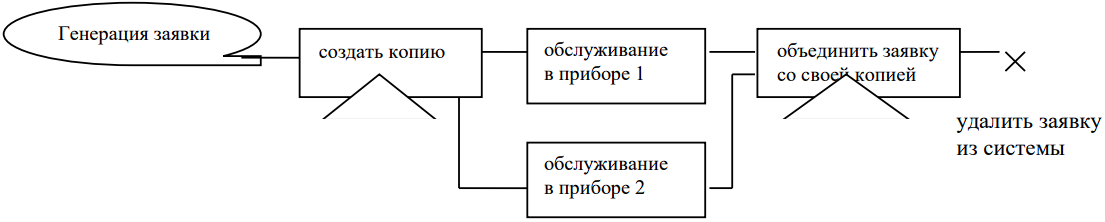


Рисунок 2 – Схема функционирования модели системы с параллельной обработкой заявки двумя приборами

10.

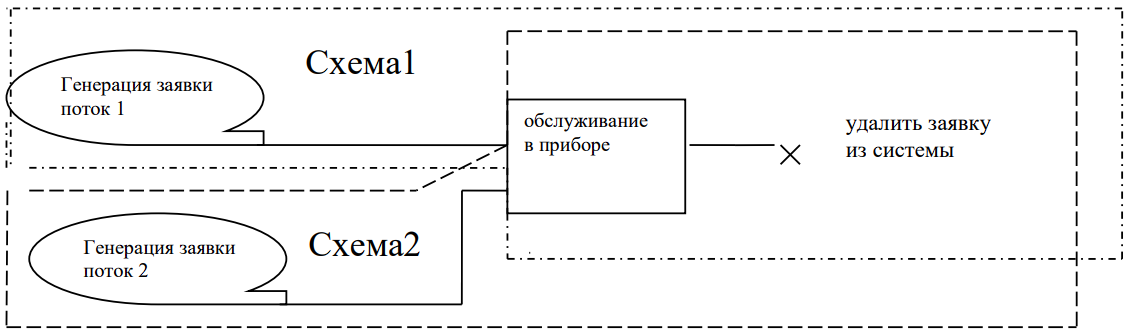


Рисунок 3 – Схема функционирования модели системы с двумя потоками к одному обслуживающему прибору

11.

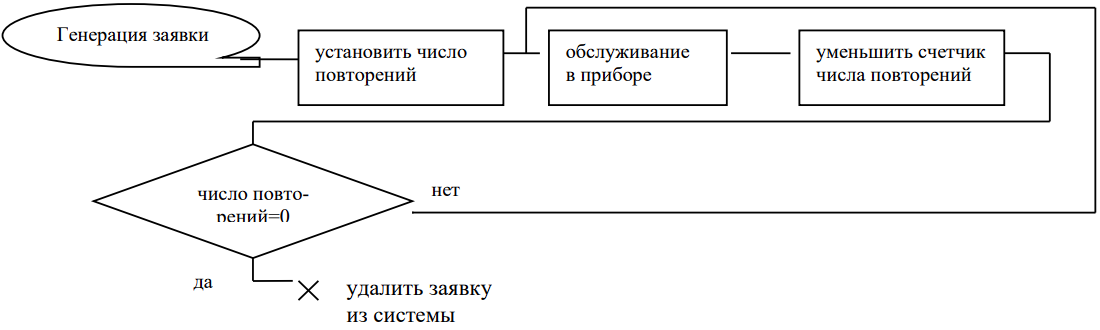


Рисунок 4 – Схема функционирования модели системы с заданным количеством циклов обработки заявки в устройстве

12.

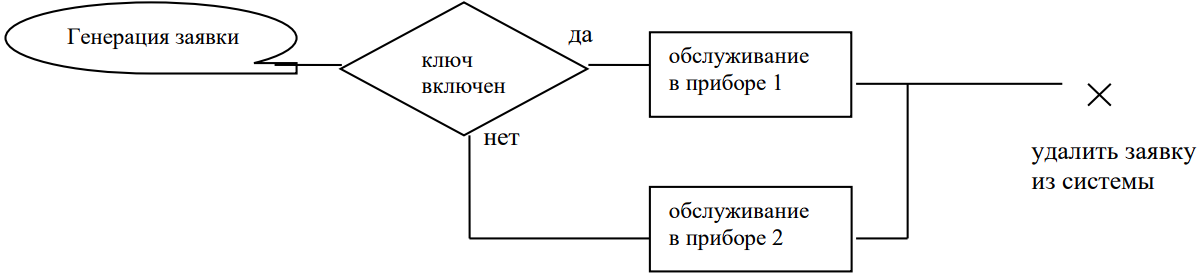


Рисунок 5 – Схема функционирования модели системы с перенаправлением потока на два прибора