

Содержание:

Введение

Курсовая работа по предмету «Моделирование систем» должна решить следующее задание:

«Самолеты прибывают для посадки в район аэропорта каждые 5 ± 2 минуты. Если взлетно-посадочная полоса свободна, прибывший самолет получает разрешение на посадку. Если взлетно-посадочная полоса занята, самолет выполняет полет по кругу и возвращается в аэропорт через 4 мин. Если после пятого круга он не получает разрешения на посадку, то отправляется на запасной аэродром. В аэропорту каждые 10 ± 2 мин к взлетно-посадочной полосе вырываются готовые к взлету самолеты и получают разрешение на взлет, если полоса свободна. Время взлета и время посадки (нахождение на полосе) составляют каждое 2 мин. Если при свободной полосе одновременно один самолет прибывает для посадки, а другая – для взлета, полоса предоставляется взлетающему самолету. Необходимо смоделировать работу аэропорта в течение суток и определить число взлетевших самолетов, число самолетов, совершивших посадку число самолетов, ушедших на запасной аэродром коэффициент загрузки взлетно-посадочной полосы».

Такие задачи очень часто встречаются в реальной жизни и имеют очень важное значение для улучшения работы различных объектов для массового обслуживания, оптимизации бизнес-процессов и снижения затрат. Такие задачи решают с помощью моделирования процесса, то есть имитационного моделирования. Аэропорт можно отнести к системе массового обслуживания (СМО) у которой по условиям задачи всего один свободный канал (взлетная полоса).

1. Основная часть

1.1 Обоснование выбора.

Как уже было сказано, аэропорт является системой массового обслуживания и нам необходимо смоделировать его работу. Для этих целей применяются программы

имитационного моделирования, как AnyLogic, MTSS (Manufacturing and Transportation Simulation System) и GPSS Word (GPSSW, General Purpose System Simulation World – Мировая общецелевая система моделирования).

Для решения задачи я остановился на использовании GPSS Word как на мощной и универсальной среде моделирования как дискретных, так и непрерывных процессов, предназначенная для профессионального моделирования самых разнообразных процессов и систем.

1.2 Структурная схема модели системы

Что бы лучше понимать условия задачи часто создают структурные схемы. Схема для нашей задачи показана на рисунке 1.

Накопитель с

условием

Самолет

посадка

ты

Взлет/посадка

Посадочная

полоса

Буфер 2

Самолет взлет

Обслуженные самолёты

Полет по кругу

Рисунок 1- Структурная схема

1.3 Временная диаграмма

Для нашей задачи так же можно построить и временную диаграмму, которая позволит лучше понять какие у СМО могут быть состояния и учесть это при разработке программы имитации.

$t_{ож}$

t_5

t_4

t_2

$t_{обр}$

$t_{об}$

$t_{об}$

$t_{об}$

$t_{ож}$

$t_{об}$

$t_{обр}$

t_1

t_3

t_6

Рисунок 2- Временная диаграмма

На диаграмме:

ось 1 – моменты поступления заявок на взлет;

ось 2 – моменты поступления заявок на посадку;

ось 3 – моменты ожидания освобождение полосы самолётов при взлете;

ось 4 – время полета самолета по кругу;

ось 5 – время занятости взлетной полосы;

1.4 Математическая модель

Данную задачу можно так же решить с помощью математической модели. Такое решение считают не очень точным, и мы его сделаем, что бы сравнить с результатами имитационного моделирования. Для решения мы используем формулу:

где – коэффициенты загрузки полосы

- время обслуживания самолета;

- суммарное время занятости полосы;

T – Общее имитируемое время работы аэропорта. Это сутки в минутах, то есть 1440.

Для того, что бы посчитать суммарное время занятости полосы нам нужно знать количество обслуженных самолетов. Его мы получим с помощью имитационного моделирования.

1.5 Имитационная модель.

Итак, как уже было сказано, для моделирования процесса мы будем использовать бесплатную версию программы GPSS Word для студентов.

Нам нужно подсчитывать количество кругов, которое сделал самолет в случае занятости полосы. Для этого объявим переменную, в которой будем его хранить:

```
INITIAL X$Circle,0
```

В данном случае мы так же присвоили ей первоначальное значение, равное 0.

Нам понадобится еще одна переменная, в которую мы будем использовать в статистической таблице. Кроме этого зададим и ее арифметическое выражение - разность текущего значения времени моделирования и предыдущего значения:

AA1 VARIABLE C1-X1;

В нашей системе только один канал/емкость, которым является взлетная полоса, мы сообщаем об этом программе с помощью команды STORAGE:

```
Runway STORAGE 1
```

Тут Runway является именем емкости, а единица говорит о том что емкость одна.

Еще нам понадобится статистическая таблица, так же ее объявим:

```
TAB1 TABLE V$AA1,5,5,44
```

Дальше мы будем описывать непосредственно программу имитации. Для этого нам понадобится команда, которая будет имитировать процесс взлета и прилета самолетов. В GPSS это команда GENERATE, которая имеет следующий синтаксис:

```
GENERATE A,B,C,D,E,F,G,H
```

Здесь A и B служат для задания интервалов между появлениями заявок, при этом можно использовать один из следующих вариантов:

интервал — равномерно распределенная в диапазоне [AB, A+B] случайная величина;

интервал — значение функции, указанной в B, умноженной на A;

C — задержка в выработке первого транзакта; D — число вырабатываемых источником заявок; E — приоритет заявок. Если D пусто, то число вырабатываемых транзактов неограниченно.

В условиях нашей задачи сказано, что взлетающий самолет имеет больший приоритет перед садящимся. По этому, сначала генерируем вылет с приоритетом 2:

```
GENERATE 10,2,,,2
```

10 у нас по условиям задачи это интервал вылетов, у которого может быть отклонения в 2 минуты, которое мы и указываем во втором параметре.

Дальше табулируем нашу таблицу и говорим программе с помощью операнда ENTER, что мы заняли взлетную полосу:

TABULATE TAB1

ENTER Runway

Так как для взлета нужно некоторое время, а конкретно в нашей задаче оно равно двум минутам имитируем это в нашей программе с помощью команды ADVANCE:

ADVANCE 2

И после этого освобождаем полосу:

LEAVE Runway

После чего уничтожаем отработавший транзакт:

TERMINATE 0

Далее нам нужно описать блок для прилетающих самолетов. Нам так же понадобится и генерация прилетов, и имитация занятости полосы, но кроме этого добавляется условие об ожидании самолетом обсуждения полосы, в случае ее занятости. По этому, нам понадобятся еще условные и безусловные переходы. Сначала генерируем событие прилета самолетов каждые пять минут плюс/минус 2 минуты с приоритетом 1 и табулируем таблицу:

GENERATE 5,2,,,1

TABULATE TAB1

Теперь нужно проверить, не занята ли полоса. Сделать это можно с помощью блока GATE, который проверяет, занят или нет канал. Синтаксис у него следующий:

GATE XX A,B

В поле операции блока GATE записывается слово GATE и через пробел - символ проверяемого условия.

Существует десять условий, которые проверяются в блоке GATE для оборудования:

NU - устройство свободно (т.е. не используется),

U - устройство не свободно (т.е. используется),

NI - устройство не захвачено,

I - устройство захвачено,

SE - память пуста (все единицы памяти свободны),

SNE - память не пуста,

SF - память заполнена (все единицы заняты),

SNF - память не заполнена,

LR - ключ выключен,

LS - ключ включен.

В поле A блока GATE записывается номер или имя проверяемого объекта. Но нам в программе это предстоит делать не однократно. По этому даем блоку имя Free, что бы можно было к нему вернуться по имени:

Free GATE SNF Runway, Busy

В случае если полоса свободна, будут выполняться команды из этого блока расположенные ниже. В противном случае управление передастся на блок Busy, который мы опишем чуть позже. И так команды, если линия свободна:

ENTER Runway; Занимаем очередь на полосу

ADVANCE 2; Задерживаем очередь на 2 минуты

LEAVE Runway; Освобождаем очередь на полосу

TERMINATE 0; Уничтожаем транзакт

Теперь опишем блок Busy. Если полоса занята, на нужно в первую очередь увеличить количество кругов на 1.

Busy SAVEVALUE Circle+, 1

Именуем блок, и с помощью оператора SAVEVALUE увеличиваем значение переменной Circle на единицу. Вернемся к условию задачи. В ней сказано, что самолет не будет все время ждать, когда освободится полоса. Если он сделает 5 кругов и полоса за это время не освободится, он улетит на запасной аэродром. По этому, после захода на следующий круг нам нужно проверить, а сколько же уже

кругов мы сделали? Мы для этого воспользуемся оператором перехода TEST, имеющий следующий синтаксис:

```
TEST XX A,B,C
```

В соответствии с ним переход к оператору, помеченному меткой C, происходит, если не выполняется условие A XX B, где XX {E, NE, L, LE, G, GE}; E — равно; NE — неравно; L — меньше; LE — меньше или равно; G — больше; GE — больше или равно (XX всегда размещается в позициях 13 и 14).

Наш код будет выглядеть:

```
TEST L X$Circle,5,Flies
```

Если значение переменной Circle будет меньше 5, то будут выполняться команды из этого блока расположенные ниже. В противном случае управление передастся на блок Flies, который мы опишем чуть позже. И так команды, если кругов меньше 5:

```
ADVANCE 4;
```

```
TRANSFER ,Free;
```

Указываем программе, что на круг уходит 4 минуты. После чего безусловно переходим к блоку проверки свободна ли полоса, который мы описали раньше.

Нам осталось описать блок, который будет выполняться в случае если самолет не дождался освобождения полосы и улетел на запасной аэродром. Он очень прост, нам нужно всего лишь спросить значение переменной Circle в ноль, для последующего использования и уничтожить транзакт.

```
Flies SAVEVALUE Circle,0
```

```
TERMINATE 0
```

Теперь укажем программе что моделирование нужно провести за полные сутки. Время моделирования указывается в минутах:

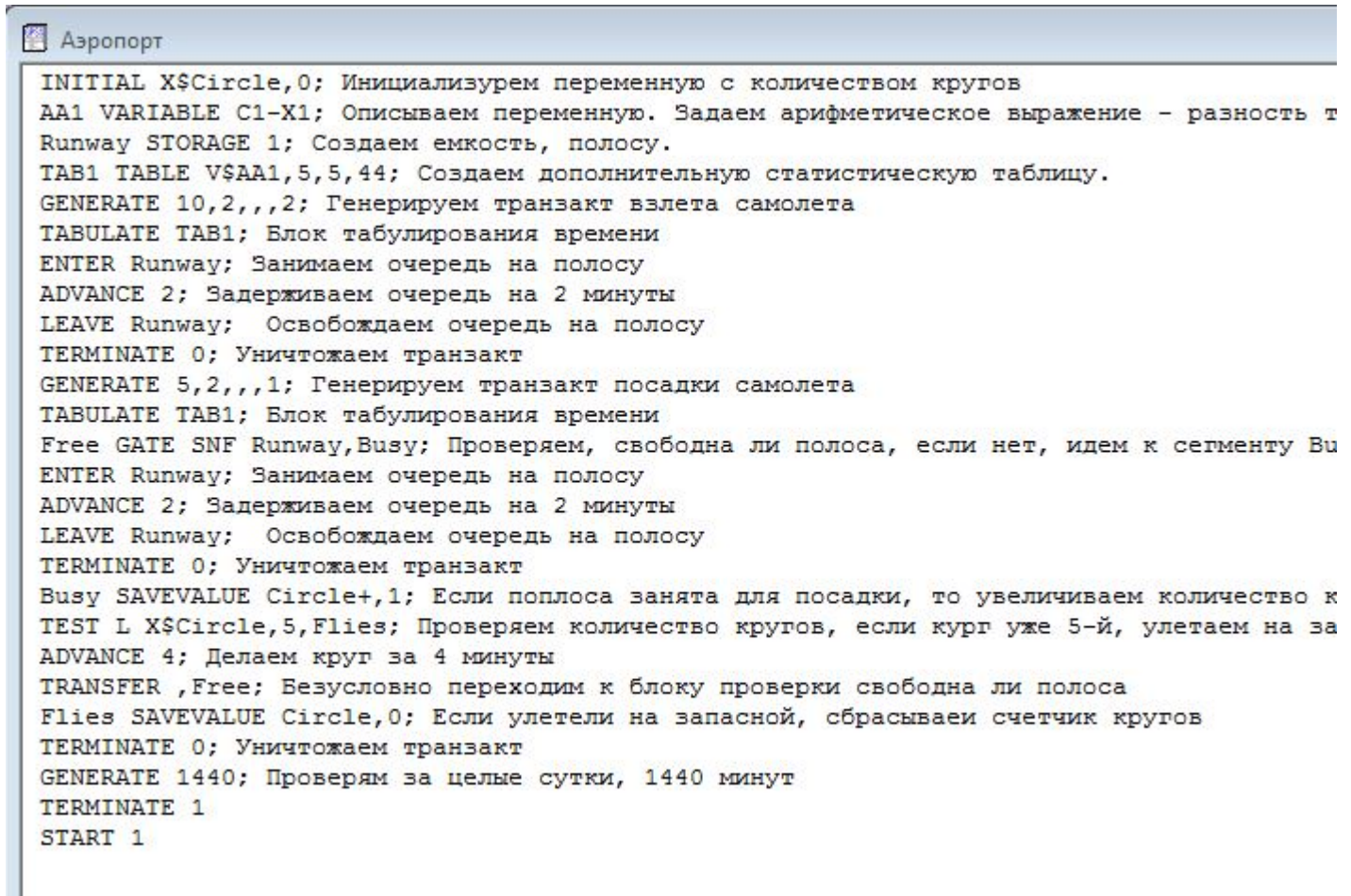
```
GENERATE 1440
```

Теперь можно стартовать моделирование:

```
TERMINATE 1
```


START 1

На рисунке 3 полный листинг нашей программы в рабочем окне программы WPSS World.



```
INITIAL X$Circle,0; Инициализуем переменную с количеством кругов
AA1 VARIABLE C1-X1; Описываем переменную. Задаем арифметическое выражение - разность т
Runway STORAGE 1; Создаем емкость, полосу.
TAB1 TABLE V$AA1,5,5,44; Создаем дополнительную статистическую таблицу.
GENERATE 10,2,,,2; Генерируем транзакт взлета самолета
TABULATE TAB1; Блок табулирования времени
ENTER Runway; Занимаем очередь на полосу
ADVANCE 2; Задерживаем очередь на 2 минуты
LEAVE Runway; Освобождаем очередь на полосу
TERMINATE 0; Уничтожаем транзакт
GENERATE 5,2,,,1; Генерируем транзакт посадки самолета
TABULATE TAB1; Блок табулирования времени
Free GATE SNF Runway,Busy; Проверяем, свободна ли полоса, если нет, идем к сегменту Bu
ENTER Runway; Занимаем очередь на полосу
ADVANCE 2; Задерживаем очередь на 2 минуты
LEAVE Runway; Освобождаем очередь на полосу
TERMINATE 0; Уничтожаем транзакт
Busy SAVEVALUE Circle+,1; Если полоса занята для посадки, то увеличиваем количество к
TEST L X$Circle,5,Flies; Проверяем количество кругов, если круг уже 5-й, улетаем на за
ADVANCE 4; Делаем круг за 4 минуты
TRANSFER ,Free; Безусловно переходим к блоку проверки свободна ли полоса
Flies SAVEVALUE Circle,0; Если улетели на запасной, сбрасываем счетчик кругов
TERMINATE 0; Уничтожаем транзакт
GENERATE 1440; Проверяем за целые сутки, 1440 минут
TERMINATE 1
START 1
```

Рисунок 3.

1.6 Анализ результатов моделирования

Выполняем симуляцию и получаем следующий отчет:

GPSS World Simulation Report - Аэропорт.23.1

Thursday, May 28, 2015 15:51:32

START TIME END TIME BLOCKS FACILITIES STORAGES

0.000 1440.000 21 0 1

LABEL LOC BLOCK TYPE ENTRY COUNT CURRENT COUNT RETRY

1 GENERATE 143 0 0

2 TABULATE 143 0 0

3 ENTER 143 0 0

4 ADVANCE 143 1 0

5 LEAVE 142 0 0

6 TERMINATE 142 0 0

7 GENERATE 289 0 0

8 TABULATE 289 0 0

FREE 9 GATE 417 0 0

10 ENTER 257 0 0

11 ADVANCE 257 0 0

12 LEAVE 257 0 0

13 TERMINATE 257 0 0

BUSY 14 SAVEVALUE 160 0 0

15 TEST 160 0 0

16 ADVANCE 128 0 0

17 TRANSFER 128 0 0

FLIES 18 SAVEVALUE 32 0 0

19 TERMINATE 32 0 0

20 GENERATE 1 0 0

21 TERMINATE 1 0 0

STORAGE CAP. REM. MIN. MAX. ENTRIES AVL. AVE.C. UTIL. RETRY DELAY

RUNWAY 1 0 0 1 400 1 0.554890 0.55489 0 0

TABLE MEAN STD.DEV. RANGE RETRY FREQUENCY CUM.%

TAB1 724.582057 414.400955 0

5.000000 - 10.000000	1	0.23
10.000000 - 15.000000	2	0.69
15.000000 - 20.000000	1	0.93
20.000000 - 25.000000	2	1.39
25.000000 - 30.000000	2	1.85
30.000000 - 35.000000	1	2.08
35.000000 - 40.000000	1	2.31
40.000000 - 45.000000	2	2.78
45.000000 - 50.000000	1	3.01
50.000000 - 55.000000	2	3.47
55.000000 - 60.000000	1	3.70
60.000000 - 65.000000	2	4.17
65.000000 - 70.000000	1	4.40
70.000000 - 75.000000	2	4.86
75.000000 - 80.000000	1	5.09
80.000000 - 85.000000	2	5.56
85.000000 - 90.000000	1	5.79
90.000000 - 95.000000	2	6.25
95.000000 - 100.000000	1	6.48
100.000000 - 105.000000	2	6.94

105.000000 - 110.000000 1 7.18
110.000000 - 115.000000 2 7.64
115.000000 - 120.000000 0 7.64
120.000000 - 125.000000 3 8.33
125.000000 - 130.000000 2 8.80
130.000000 - 135.000000 1 9.03
135.000000 - 140.000000 2 9.49
140.000000 - 145.000000 1 9.72
145.000000 - 150.000000 1 9.95
150.000000 - 155.000000 2 10.42
155.000000 - 160.000000 2 10.88
160.000000 - 165.000000 1 11.11
165.000000 - 170.000000 2 11.57
170.000000 - 175.000000 1 11.81
175.000000 - 180.000000 2 12.27
180.000000 - 185.000000 1 12.50
185.000000 - 190.000000 2 12.96
190.000000 - 195.000000 1 13.19
195.000000 - 200.000000 2 13.66
200.000000 - 205.000000 2 14.12
205.000000 - 210.000000 2 14.58
210.000000 - 215.000000 0 14.58
215.000000 - _ 369 100.00

На рисунке 4 отображены все блоки нашей программы с результатами их выполнения.

Loc	Block Type	Current C...	Entry Co...	Retry Ch...	Line Nu...	Include-...	
1 GEN	GENERATE	0	143	0	5	0	
2 TAB	TABULATE	0	143	0	6	0	
3 ENT	ENTER	0	143	0	7	0	
4 ADV	ADVANCE	1	143	0	8	0	
5 LEA	LEAVE	0	142	0	9	0	
6 TER	TERMINATE	0	142	0	10	0	
7 GEN	GENERATE	0	289	0	11	0	
8 TAB	TABULATE	0	289	0	12	0	
FREE	GATE	0	417	0	13	0	
10 ENT	ENTER	0	257	0	14	0	
11 ADV	ADVANCE	0	257	0	15	0	
12 LEA	LEAVE	0	257	0	16	0	
13 TER	TERMINATE	0	257	0	17	0	
BUSY	SAVEVALUE	0	160	0	18	0	
15 TES	TEST	0	160	0	19	0	
16 ADV	ADVANCE	0	128	0	20	0	
17 TRA	TRANSFER	0	128	0	21	0	
FLIES	SAVEVALUE	0	32	0	22	0	
19 TER	TERMINATE	0	32	0	23	0	
20 GEN	GENERATE	0	1	0	24	0	
21 TER	TERMINATE	0	1	0	25	0	

Рис. 4

Теперь мы видим такие результаты:

- Количество обслуженных самолетов равно 432;
- Взлетевших самолетов 143;
- Севших самолетов 289;
- Самолетов ушедших на следующий круг 160;
- Самолетов улетевших на запасной аэродром 32;
- Коэффициент загрузки полосы 0,55489

И можем сделать вывод, что у полосы еще есть запас загрузки, но тем не менее на запасной аэродром улетаёт около 10% всех прилетевших самолетов.

1.7 Сравнение результатов имитационного моделирования и математической модели

Вычислим по формуле коэффициент загрузки по математической модели:

Мы видим, что результат отличается.

Заключение

Имитационное моделирование в значительной степени облегчает исследование различных процессов. Кроме этого она дает более точные результаты, чем математическое моделирование. И именно моделирование следует променять для решения задач связанных с системами массового обслуживания, которые очень часто встречаются в нашей реальной жизни. Например с помощью имитационного моделирования руководство Call-центра может рассчитать нагрузку на операторов и их минимальное количество при определенном объеме входящих звонков.

GPSS World – это общецелевая система моделирования, значительно облегчающая процесс создания и выполнения программ имитационного моделирования. Она очень гибкая, и с ее помощью можно смоделировать практически любую ситуацию.

Студенческая версия системы GPSS World так же эффективна, как и коммерческая, и на сегодняшних персональных компьютерах выполняется в тысячу раз быстрее, чем работала оригинальная версия GPSS/PC в 1984 году.

Список литературы

1. Кудрявцев Е. М. GPSS World Основы имитационного моделирования различных систем. - М.: ДМК Пресс, 2004.
2. Советов Б.Я., Яковлев С. А. Моделирование систем. - М.:Высш. шк.,1995.

Приложение 1

Полный листинг программы:

INITIAL X\$Circle,0; Инициализируем переменную с количеством кругов

A1 VARIABLE C1-X1; Описываем переменную. Задаем арифметическое выражение - разность текущего значения времени моделирования и предыдущего значения

Runway STORAGE 1; Создаем емкость, полосу

TAB1 TABLE V\$AA1,5,5,44; Создаем дополнительную статистическую таблицу

GENERATE 10,2,,,2; Генерируем транзакт взлета самолета

TABULATE TAB1; Блок табулирования времени

ENTER Runway; Занимаем очередь на полосу

ADVANCE 2; Задерживаем очередь на 2 минуты

LEAVE Runway; Освобождаем очередь на полосу

TERMINATE 0; Уничтожаем транзакт

GENERATE 5,2,,,1; Генерируем транзакт посадки самолета

TABULATE TAB1; Блок табулирования времени

Free GATE SNF Runway,Busy; Проверяем, свободна ли полоса, если нет, идем к сегменту Busy

ENTER Runway; Занимаем очередь на полосу

ADVANCE 2; Задерживаем очередь на 2 минуты

LEAVE Runway; Освобождаем очередь на полосу

TERMINATE 0; Уничтожаем транзакт

Busy SAVEVALUE Circle+,1; Если полоса занята для посадки, то увеличиваем количество кругов на 1

TEST L X\$Circle,5,Flies; Проверяем количество кругов, если круг уже 5-й, улетаем на запасной

ADVANCE 4; Делаем круг за 4 минуты

TRANSFER ,Free; Безусловно переходим к блоку проверки свободна ли полоса

Flies SAVEVALUE Circle,0; Если улетели на запасной, сбрасываем счетчик кругов

TERMINATE 0; Уничтожаем транзакт

GENERATE 1440; Проверяем за целые сутки, 1440 минут

TERMINATE 1

START 1