

Simulation of Physical Examination (PE) Services

W. M. Song 桑慧敏

Tsing Hua Univ. 清華大學

2015.11.11

1 Physical Examination (PE) Service

2 Performance Measures

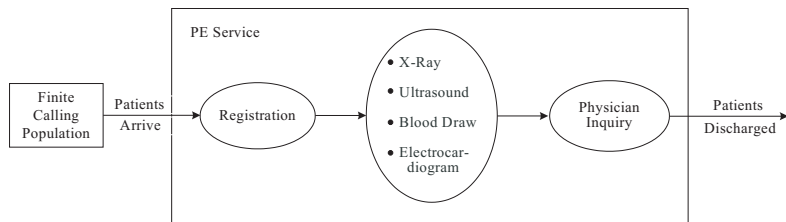
3 Decision Variables

4 Input Analysis

5 Flexsim Modeling

6 Output Analysis

Physical Examination (PE) Service



- Discussion: performance measures and decision variables.

Performance Measures

Performance Measures

- $E(W)$: The mean patient wait time
- $P(W > 150)$: Patient prolonged wait rate
- $E(D)$: The mean physician total shift time
- p_u : The physician utilization

Decision variables

- Three patient **dispatching rules**:LPT,SPT,Random

- **Physician inquiry starting times**

$t = 0, 5, 10, 15, 20, \dots, 150$, where $t = 0$ denotes a physician inquiry starting time of 8:00 am

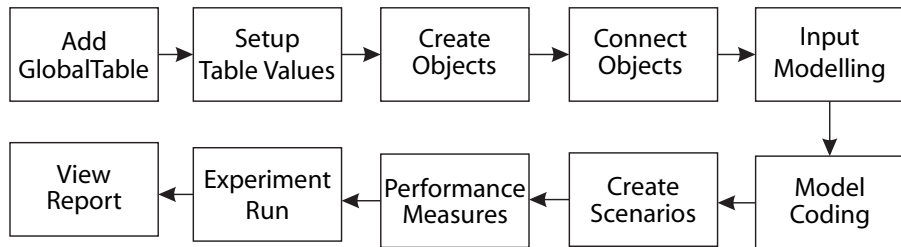
- Two types of **patient arrival policies**

- One-group Policy: 20 patients register at 8:00 am
- Staggered Policy: 10 patients register at 8:00 am the rest register at 9:00 am

The Fitted Distribution of PE services

	Random variable	Fitted Distribution	Mean(min.)	Variance(min ²)
Arrival process	Patient inter-arrival time	$0.01 + 28.13 \text{ beta}(0.74, 3.4)$	5.05	22.63
Service process				
Stage 1	Registration	$6.19 + 5.82 \text{ beta}(0.52, 0.97)$	8.22	3.09
Stage 2	X-ray	Uniform(5.43, 8.74)	7.09	0.91
	Ultrasound	Uniform(3.53, 11.08)	7.31	4.75
	Blood draw	$1.34 + 4.87 \text{ beta}(0.38, 1.18)$	2.53	1.71
	Electrocardiogram	$3.34 + 3.79 \text{ beta}(0.63, 0.94)$	4.96	1.34
Stage 3	Doctor inquiry	$4.86 + 5.21 \text{ beta}(0.64, 0.78)$	7.21	2.77

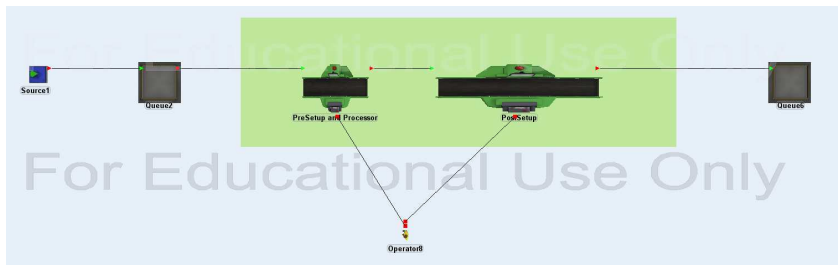
Model Construction via Flexsim



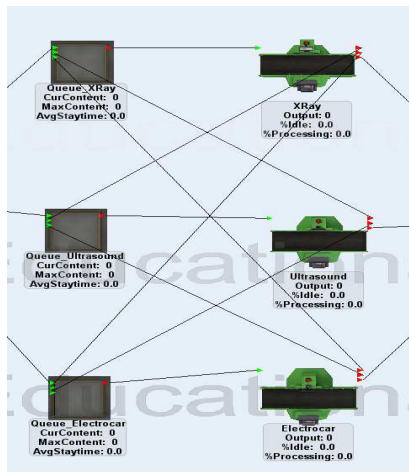
Global Tables

- **Each_Patient_Wait_Time**
 - 20 Rows for 20 patients
 - 4 Columns for Arrive Time, Leave Time, Stay Time, $W > 150$
- **Doctor_QueryTime**
 - 20 Rows for 20 patients
 - 3 Columns for Patient Arrive Time, Patient Leave Time, Query Time
- **Performance_Measures**
 - 4 Rows for $E(W)$, $P(W > 150)$, $E(D)$, p_u
 - 1 Column
- **Total_Arrivals**
 - For controlling total arrival patients
 - Input 20 in (1,1)
- **Query_Start_Time**
 - For controlling physician starting time
 - Input 0 in (1,1)

Flexsim Model Overview



Connect PE Stations



- Each station's Output Port connect to the queue of the another two station's Input Port
- Each station's Output Port connect to Queue_Inquiry

Set Patients Dispatch Rule

- Registration > Flow > Send To > Random Port >
 - `duniform(1, nrop(current))`
- XRay > Flow > Send To > Custom Code >

```

XRay - Send To Port
1 /**Custom Code*/
2 treenode item = parnode(1);
3 treenode current = ownerobject(c);
4
5 int x=getlabelnum(item,"XRay_Label");
6 int u=getlabelnum(item,"Ultrasound_Label");
7 int e=getlabelnum(item,"Electrocar_Label");
8
9 int value=x+u+e;
10 colorarray(item, value);
11 if((x+u+e)==3)
12 {
13     return 1;
14 }
15 else
16 {
17     if(u+e==0)
18     {
19         return duniform(2, 3);
20     }
21     else
22     {
23         if(u==0)
24         {
25             return 2;
26         }
27         else
28         {
29             return 3;
30         }
31     }
32 }
33
  
```

XRay Properties

Processor Breakdowns Flow Triggers Labels General Statistics

Appearance

3D Shape fs3d/Processor/Processor.3ds

Shape Factors Edit Reset

2D Shape ***

3D Texture ***

Color [Green]

Visuals/Animations Load Save Edit

Position, Rotation, and Size

X 15.00 RX 0.00 SX 4.00

Y 9.57 RY 0.00 SY 3.00

Z 0.00 RZ 0.00 SZ 2.00

Ports

Input Ports: 1: Queue_Inquiry

Central Ports: 2: Queue_Ultrasound

Output Ports: 3: Queue_Electrocar

Rank ^

Rank v

Delete

Properties

Apply OK Cancel

Set Patients Dispatch Rule

```
int x=getlabelnum(item,"XRay_Label");
int u=getlabelnum(item,"Ultrasound_Label");
int e=getlabelnum(item,"Electrocar_Label");
int value=x+u+e;
colorarray(item, value);
if((x+u+e)==3)
{
    return 1;
}
else
{
    if(x+u==0)
    {
        return duniform(2, 3);
    }
    else
    {
        if(x==0)
        {
            return 2;
        }
        else
        {
            return 3;
        }
    }
}
```

Set Patients Dispatch Rule

- Ultrasound > Flow > Send To > Custom Code >

```

Ultrasound - Send To Port
1 /**Custom Code*/
2 treenode item = parnode(1);
3 treenode current = ownerobject(c);
4
5 int x=getlabelnum(item,"XRay_Label");
6 int u=getlabelnum(item,"Ultrasound_Label");
7 int e=getlabelnum(item,"Electrocar_Label");
8 int value=x+u+e;
9
10 colorarray(item, value);
11 if((x+u+e)==3)
12 {
13     return 1;
14 }
15 else
16 {
17     if (x+e==0)
18     {
19         return duniform(2, 3);
20     }
21     else
22     {
23         if (x==0)
24         {
25             return 2;
26         }
27         else
28         {
29             return 3;
30         }
31     }
32 }
33
  
```

Ultrasound Properties

Processor Breakdowns Flow Triggers Labels General Statistics

Appearance

3D Shape fs3d/Processor/Processor.3ds

Shape Factors Edit Reset

2D Shape ***

3D Texture ***

Color

Visuals/Animations Load Save Edit

Position, Rotation, and Size

X 15.13 RX 0.00 SX 4.00

Y 0.79 RY 0.00 SY 3.00

Z 0.00 RZ 0.00 SZ 2.00

Ports

Input Ports 1: Queue_Inquiry Rank ^

Central Ports 2: Queue_XRay Rank v

3: Queue_Electrocar

Properties Delete

Apply OK Cancel

Set Patients Dispatch Rule

- Electrocar > Flow > Send To > Custom Code >

```

Electrocar - Send To Port
1 /**Custom Code*/
2 treenode item = parnode(1);
3 treenode current = ownerobject(c);
4 int x=getlabelnum(item,"XRay_Label");
5 int u=getlabelnum(item,"Ultrasound_Label");
6 int e=getlabelnum(item,"Electrocar_Label");
7 int value=x+u+e;
8
9 colorarray(item, value);
10 if((x+u+e)==3)
11 {
12     return 1;
13 }
14 else
15 {
16     if(x+u==0)
17     {
18         return duniform(2, 3);
19     }
20     else
21     {
22         if(x==0)
23         {
24             return 2;
25         }
26         else
27         {
28             return 3;
29         }
30     }
31 }
32

```

Electrocar Properties

Processor Breakdowns Flow Triggers Labels General Statistics

Appearance

3D Shape fs3d/Processor/Processor.3ds

Shape Factors Edit Reset

2D Shape ***

3D Texture ***

Color

Visuals/Animations Load Save Edit

Flags

Show Name

Show Ports

Show 2D Shape

Show 3D Shape

Show Contents

Scale Contents

Protected

Position, Rotation, and Size

X 15.07 RX 0.00 SX 4.00

Y -8.94 RY 0.00 SY 3.00

Z 0.00 RZ 0.00 SZ 2.00

Ports

Input Ports: 1: Queue_Inquiry, 2: Queue_XRay, 3: Queue_Ultrasound

Central Ports: Dispatch Ports

Rank ^ Rank v Delete

Properties

Apply OK Cancel

Input Modelling

- Source1
 - Inter-Arrival Time: $\text{beta}(0.01, 28.14, 0.74, 3.4, 0)$
- Registration
 - Process Time: $\text{beta}(6.19, 12.01, 0.52, 0.97, 0)$
- XRay
 - Process Time: $\text{uniform}(5.43, 8.74, 0)$
- Ultrasound
 - Process Time: $\text{uniform}(3.53, 11.08, 0)$
- Electrocar
 - Process Time: $\text{beta}(3.34, 7.13, 0.63, 0.94, 0)$
- Doctor_Inquiry
 - Process Time: $\text{beta}(4.86, 10.07, 0.64, 0.78, 0)$
- Count_Time
 - Process Time: 5

Source1

Add Labels: Source1 > Triggers > OnCreation

- Label Name: "arrive_system" , Value:0
- Label Name: "leave_system" , Value:0
- Label Name: "arrive_doc_query" , Value:0
- Label Name: "leave_doc_query" , Value:0
- Label Name: "XRay_Label" , Value:0
- Label Name: "Ultrasound_Label" , Value:0
- Label Name: "Electrocar_Label" , Value:0

Control Total Arrivals: Source1 > Triggers > OnCreation > CustomCode

```
int n_patients = gettablenum("Total_Arrivals",1,1);  
if(getoutput(current)==n_patients) { stopoutput(current); }
```


Set Label Num in Objects

- Queue_Register > Triggers > OnEntry > CustomCode
 - `setlabelnum(item,"arrive_system",time());`
- Doctor_Inquiry > Triggers > OnEntry > CustomCode
 - `setlabelnum(item,"arrive_doc_query",time());`
- Doctor_Inquiry > Triggers > OnExit > CustomCode
 - `setlabelnum(item,"leave_system",time());`
 - `setlabelnum(item,"leave_doc_query",time());`
- XRay > Triggers > OnExit > CustomCode
 - `setlabelnum(item,"XRay_Label",1);`
- Ultrasound > Triggers > OnExit > CustomCode
 - `setlabelnum(item,"Ultrasound_Label",1);`
- Electrocar > Triggers > OnExit > CustomCode
 - `setlabelnum(item,"Electrocar_Label",1);`

Queue_End > OnEntry > CustomCode(1/2)

```
int n = getinput(current); //ith patient
double arrive_system_time = getlabelnum(item,"arrive_system");
double leave_system_time = getlabelnum(item,"leave_system");
double wait_time_in_system = leave_system_time - arrive_system_time;
double arrive_doc_time = getlabelnum(item,"arrive_doc_query");
double leave_doc_time = getlabelnum(item,"leave_doc_query");
double wait_time_in_query = leave_doc_time - arrive_doc_time;
// Set Each_Patient_Wait_Time Table
settablenum("Each_Patient_Wait_Time",n,1,arrive_system_time);
settablenum("Each_Patient_Wait_Time",n,2,leave_system_time);
settablenum("Each_Patient_Wait_Time",n,3,wait_time_in_system);
if(wait_time_in_system > 150)
{settablenum("Each_Patient_Wait_Time",n,4,1);}
// Set Doctor_QueryTime Table
settablenum("Doctor_QueryTime",n,1,arrive_doc_time);
settablenum("Doctor_QueryTime",n,2,leave_doc_time);
settablenum("Doctor_QueryTime",n,3,wait_time_in_query);
```

Queue_End > OnEntry > CustomCode(2/2)

```

//Performance Measures
if(getinput(current)==gettablenum("Total_Arrivals",1,1))
{ //*****E(W) Start*****
double time_sum=0;
double average_time;
for(int i=1;i<=gettablenum("Total_Arrivals",1,1);i++)
{ time_sum = time_sum + gettablenum("Each_Patient_Wait_Time",i,3); }
average_time = time_sum/gettablenum("Total_Arrivals",1,1);
settablenum("Performance_Measures",1,1,average_time);
//*****E(W) Stop*****
//*****P(W> 150) Start*****
int num_wait_over_150=0;
for(int i=1;i<=gettablenum("Total_Arrivals",1,1);i++)
{ num_wait_over_150 = num_wait_over_150 + gettablenum("Each_Patient_Wait_Time",i,4); }
double p_num_wait_over_150 = num_wait_over_150/gettablenum("Total_Arrivals",1,1);
settablenum("Performance_Measures",2,1,p_num_wait_over_150);
//*****P(W> 150) Stop*****
//*****E(D) Start *****
double query_time_sum=0;
for(int i=1;i<=gettablenum("Total_Arrivals",1,1);i++)
{ query_time_sum = query_time_sum + gettablenum("Doctor_QueryTime",i,3); }
settablenum("Performance_Measures",3,1,query_time_sum);
//*****E(D) Stop *****
//***** Utilization Start*****
double query_start_time = gettablenum("Query_Start_Time",1,1);
double utilization = (query_time_sum)/(time() - query_start_time);
settablenum("Performance_Measures",4,1,utilization);
//***** Utilization Stop*****
endreplication(1);

```

Use Source(Count_Time) to Control Query Starting Time

- Close output port before Query when simulation start
 - Queue_Inquiry > Triggers > OnReset > Close and Open Ports > Action: closeoutput
- Open output port when time() > Query Starting Time
 - Count_Time > Triggers > OnCreation > Custom Code >

```
if(time())>gettablenum("Query_Start_Time",1,1)) {  
  openoutput(node("/Queue_Inquiry",model()));  
}
```

Set Scenarios and Performance Measures

31 Scenarios of Physician Inquiry Starting Time -
 $t = 0, 5, 10, 15, 20, \dots, 150$

Experimenter > Scenarios

- Scenarios:31 , Experiment variables: 1
- Variable1:/Tools/GlobalTables/Query_Start_Time>variables/data/1/1
- Scenario 1 - Scenario 31 > From 0,5,10,15,20,...,150

Experimenter > Performance Measures

- E(W), Global table value:"Performance_Measures",Row:1,Col:1
- P(W> 150), Global table value:"Performance_Measures",Row:2,Col:1
- E(D), Global table value:"Performance_Measures",Row:3,Col:1

Output Analysis

Skip here. Leave as homework problems.

- Estimate the performance measure θ via replications.
- Report the estimated value via LDR or confidence interval.
- Make conclusions

Home Work

- Due Date: 2015.11. 18
- Consider the complete system with 3 inspection services (X-Ray, Ultrasound, Blood Draw and Electrocardiogram)
- Choose one of the following task
 1. Comparing dispatching rules (LPT, SPT, and Random)
 2. Comparing Physician's inquiry starting times
 3. Comparing two types of patient arrival policies