# Monty Hall Problem and Solution(s)

Tuesday, July 4, 2023 7:48 PM

by Garrett A. Hughes
gah@modelingcomplexsystems.net

## Problem Statement

A contestant selects 1 of three doors. There is a car behind one of the doors and a goat behind each of the others. Before the contestant's door is opened, the game show host opens one of the other doors to expose a goat. As the contestant, do you stay with your original choice, or given the opportunity (you will always have this opportunity) do you switch your choice to the remaining door.

## Solution by Simulation

**System Description**
This is a bit of overkill for this problem, but the solution illustrates how simulating a dynamic system can help solve a problem of this nature. Real systems will contain anything from hundreds to millions of nodes and are not practically amenable to anything but solution by simulation.

When I worked for Eastman Kodak Company, I solved problems that arose in complex hardware/software systems by building dynamic models, which were able to reproduce the problem, and thus offer the opportunity to try various solutions. These systems could be characterized as queueing networks that for the most part generated discrete events of a stochastic nature. The Monty Hall Problem can be viewed as a system with these characteristics.

**Modeling Tool**
My modeling tool of choice was a commercial product called SES Workbench. This was a very sophisticated tool with a graphical user interface that included a number of nodes that could be connected with links to form a directed graph. If modeled accurately, this graph was quite good at generating the behaviors of interest. Each node could be parameterized as necessary, and "C" code could be written for each of the nodes in order to modify its default behavior. The node types available are shown in the figure below.

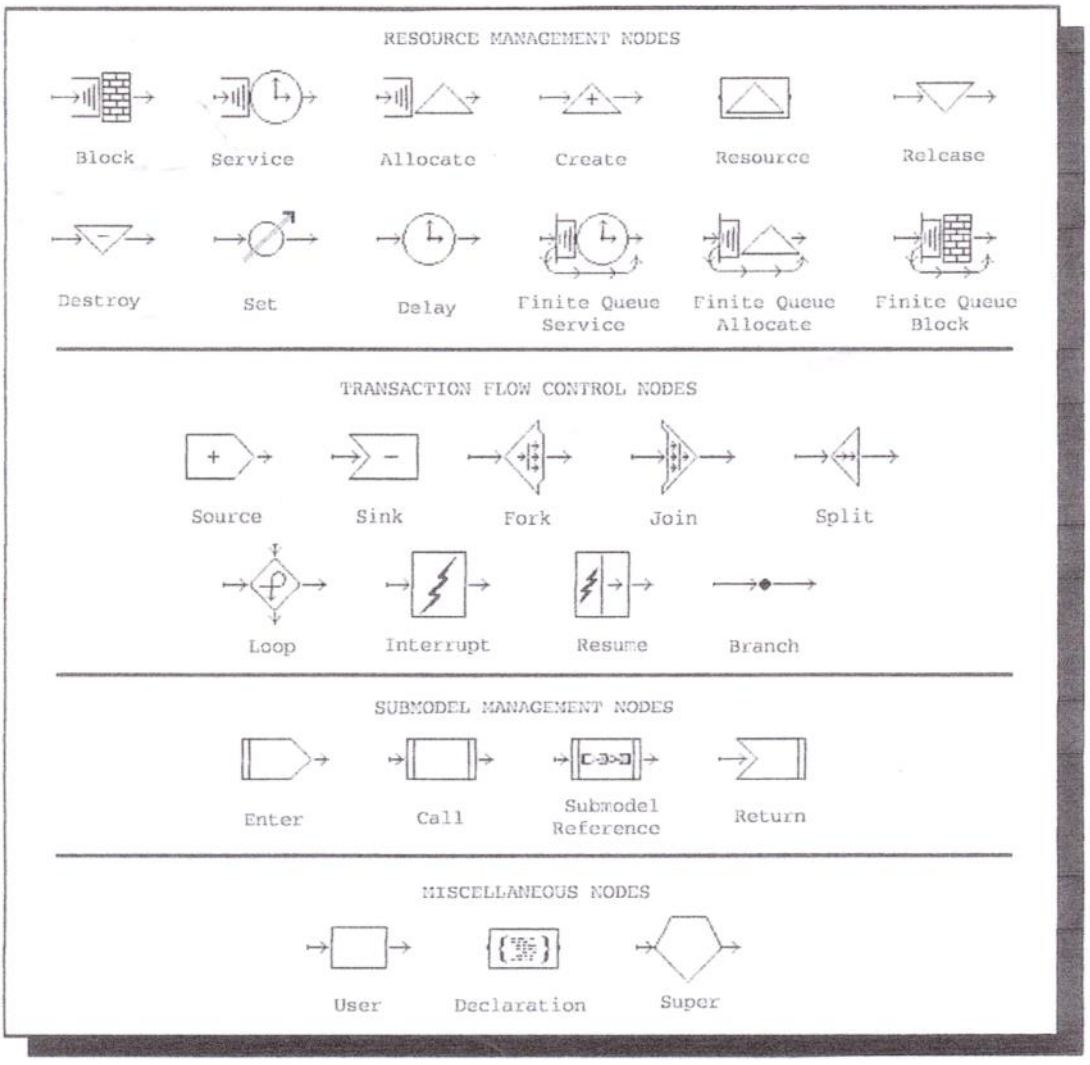



**Figure** : Node Types

**System Diagram**

The queueing network diagram that describes this system is shown in the first of the attached pages

**Interpreting the Diagram**

The first thing to observe it that this is not a logic diagram. The nodes represent locations or time where activities take place in the real world. The network is traversed by the agents in the system from node to node. At the nodes the agents carry out the instructions that are related to the node type based on the parameters and auxiliary code that have been entered by the user.

In the beginning a single transaction is generated that establishes system parameters like how many times the simulation is to be repeated. The detailed statistic reports (see below) indicate that the simulation repeats 10,000 times for each of two runs. At initialization two goats and a car are generated and placed randomly behind the three doors.

When the model is initialized a contestant and a host are also generated. The host tracks a different path than the contestant. A contestant appears first and selects a door, then waits until the host opens a door with a goat behind it. That completes the host's activities. The contestant is then allowed to proceed and either keeps their original door selection or changes it. Then the contestant opens their selected door to reveal either a goat or a car.

This activity repeats 10,000 times. In one run the contestant always stays with their original selection, in the other the contestant always changes their selection. The statistics show that if the contestant changes their selection, that they win a car 67 percent of the time. If they don't change their selection they only win 33 percent of the time.

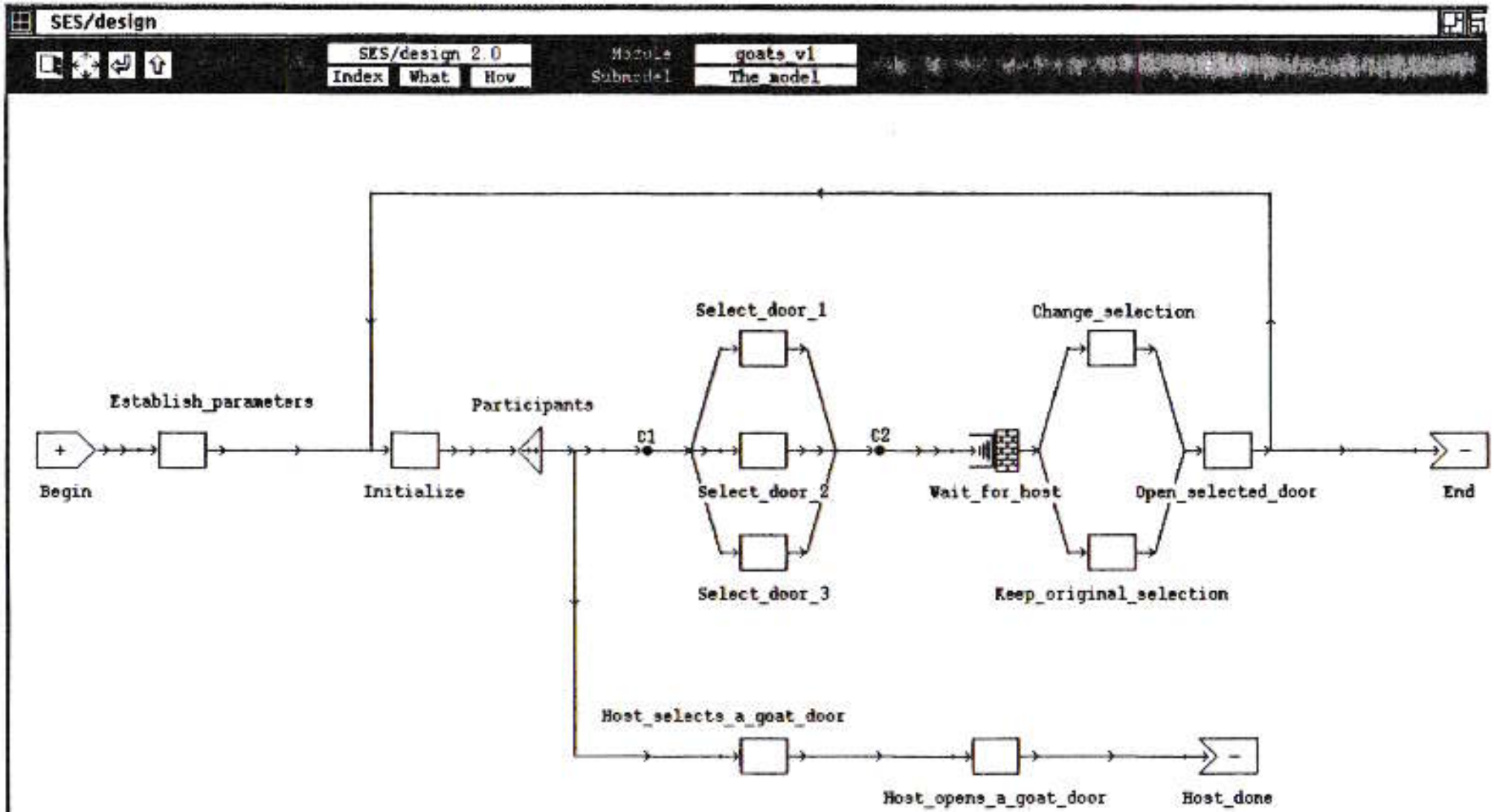
Most people guess that the odds of winning are 50/50 and don't change their selection. This problem has fooled some of the best mathematicians in the world.

## Analytical Solution

I wrote out an analytical solution to this problem years ago. I share it with you in the next of the attached pages

## Enumerated Solution

If both the *Simulation* and *Analytical* solutions seem a little obtuse, the enumerated solution is crystal clear.  A table is shown  for both keeping and changing your selection. All possible cases are enumerated. See the *Enumerated* solution in the attached pages


SES/design
SES/design 2.0
Index
What
How
Module
goats_v1
Submodel
The_model
Establish_parameters
Participants
Select_door_1
Change_selection
C1
C2
Begin
Initialize
Select_door_2
Wait_for_host
Open_selected_door
End
Select_door_3
Keep_original_selection
Host_selects_a_goat_door
Host_opens_a_goat_door
Host_done

```
*****************************************************************************
*********************** DETAILED STATISTIC REPORT ***************************
*****************************************************************************

**Statistic Report:

INTER-ARRIVAL TIMES for node Change_selection

  In module:  goats_v1
  In submodel:  The_model

   category:  ALL
      no of arrivals:       0


**Statistic Report:

result for node Open_selected_door
 (a discrete user-defined statistic)

  In module:  goats_v1
  In submodel:  The_model

   category:  contestant
      MEAN:        0.3321      variance:       0.22183    stdev:      0.47099
      minimum:          0      maximum:              1
      sample count:  10000

                    result            SAMPLES IN      % IN
               INTERVAL               INTERVAL      INTERVAL
     -Infinity < X <=      0.000          6679         66.79
          0.000 < X <=      1.000          3321         33.21
          1.000 < X <   Infinity             0          0.00
                                      ---------
                                total:   10000

   category:  ALL
      MEAN:        0.3321      variance:       0.22183    stdev:      0.47099
      minimum:          0      maximum:              1
      sample count:  10000

                    result            SAMPLES IN      % IN
               INTERVAL               INTERVAL      INTERVAL
     -Infinity < X <=      0.000          6679         66.79
          0.000 < X <=      1.000          3321         33.21
          1.000 < X <   Infinity             0          0.00
                                      ---------
                                total:   10000


**Statistic Report:

INTER-ARRIVAL TIMES for node Keep_original_selection

  In module:  goats_v1
  In submodel:  The_model

   category:  contestant
      MEAN:             0      variance:             0    stdev:             0
      minimum:          0      maximum:              0
      no of arrivals:  10000      no of inter-arrival times:    9999

   category:  ALL
      MEAN:             0      variance:             0    stdev:             0
      minimum:          0      maximum:              0
      no of arrivals:  10000      no of inter-arrival times:    9999
```

```
*****************************************************************************
********************** DETAILED STATISTIC REPORT ****************************
*****************************************************************************

**Statistic Report:

INTER-ARRIVAL TIMES for node Change_selection

  In module:  goats_v1
  In submodel:  The_model

   category:  contestant
      MEAN:              0     variance:              0   stdev:              0
      minimum:           0     maximum:               0
      no of arrivals:  10000       no of inter arrival times:    9999

   category:  ALL
      MEAN:              0     variance:              0   stdev:              0
      minimum:           0     maximum:               0
      no of arrivals:  10000       no of inter arrival times:    9999


**Statistic Report:

result for node Open_selected_door
 (a discrete user-defined statistic)

  In module:  goats_v1
  In submodel:  The_model

   category:  contestant
      MEAN:         0.6679     variance:        0.22183   stdev:        0.47099
      minimum:           0     maximum:               1
      sample count:  10000

                    result                SAMPLES IN      % IN
                 INTERVAL                 INTERVAL      INTERVAL
       -Infinity < X <=      0.000          3321          33.21
            0.000 < X <=      1.000          6679          66.79
            1.000 < X <    Infinity             0           0.00
                                          ---------
                                  total:     10000

   category:  ALL
      MEAN:         0.6679     variance:        0.22183   stdev:        0.47099
      minimum:           0     maximum:               1
      sample count:  10000

                    result                SAMPLES IN      % IN
                 INTERVAL                 INTERVAL      INTERVAL
       -Infinity < X <=      0.000          3321          33.21
            0.000 < X <=      1.000          6679          66.79
            1.000 < X <    Infinity             0           0.00
                                          ---------
                                  total:     10000


**Statistic Report:

INTER-ARRIVAL TIMES for node Keep_original_selection

  In module:  goats_v1
  In submodel:  The_model

   category:  ALL
      no of arrivals:      0
```

## Solution to car and goats problem

? P(win)

Solution

$$P(\text{win}) = 1 - P(\text{Lose})$$

$$P(\text{lose}) = P(C) \cdot P(G|C)$$

$$= \frac{1}{3} \cdot 1 = \frac{1}{3}$$

Theorem $P(\text{win}) = \underline{2/3}$

where

C is the event that car is selected

G is event that goat is selected

discussion:

If you are going to switch choices, then - in order to lose - you must select the car on your first choice. The P(C) is 1/3. After you switch, the probability of selecting a goat, given that the car has already been selected is P(G|C) equal to 1.

$\therefore P(\text{losing}) = 1/3 \cdot 1 = 1/3$ and $1 - P(\text{lose})$

$P(\text{win}) = 2/3$

Remark show that you must select the car on your first choice in order to lose
(to show that the above is true)

Suppose you picked a goat on your first choice, then the host will reveal the other goat, you will switch and since only the car is left, you will automatically win

∴ you must select the car on your first choice in order to lose

comments

it's a very clever ruse (statement) of the problem

enumerated

change selection

| CAR | CHOICE | OPEN | Switch to | RESULT |
|---|---|---|---|---|
| 1 | 1 | 2 or 3 | 2 or 3 | lose |
| 1 | 2 | 3 | 1 | win |
| 1 | 3 | 2 | 1 | win |
| 2 | 1 | 3 | 2 | win |
| 2 | 2 | 1 or 3 | 1 or 3 | lose |
| 2 | 3 | 1 | 2 | win |
| 3 | 1 | 2 | 3 | win |
| 3 | 2 | 1 | 3 | win |
| 3 | 3 | 1 or 2 | 1 or 2 | lose |

win 6/9 lose 3/9

enumerated do not change selection

| car | choice | open | keep | result |
|---|---|---|---|---|
| 1 | 1 | 2 or 3 | 1 | win |
| 1 | 2 | 3 | 2 | lose |
| 1 | 3 | 2 | 3 | lose |
| 2 | 1 | 3 | 1 | lose |
| 2 | 2 | 1 or 3 | 2 | win |
| 2 | 3 | 1 | 3 | lose |
| 3 | 1 | 2 | 1 | lose |
| 3 | 2 | 1 | 2 | lose |
| 3 | 3 | 1 or 2 | 3 | win |

| win | lose |
|---|---|
| 3/9 | 6/9 |