# Patient Flow Model 01

One patient type, one unit with infinite capacity



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
| Patient arrival rate |  | 6 | patients per day | |
| Average length of stay |  | 2.0 | days |  |
|  |  |  |  |  |
|  |  |  |  |  |
| Average occupancy in postpartum unit |  | ? | patients |  |
|  |  |  |  |  |

1. What is the average number of total patients in postpartum?
2. If the average LOS increased to 2.5 days, what is the resulting average number of total patients in postpartum?
3. What would the average LOS need to be reduced to in order to maintain an average occupancy of 10 patients in postpartum?
4. Create a simulation model in Simio that will allow you to confirm your answers for Question 1-3.
   1. Assume that the interarrival time of patients is exponentially distributed. You can figure out the mean interarrival time from the patient arrival rate (24 hours/6 patients = 4hrs/pat). Call your entity type, Patient.
   2. For now, assume that the length of stay is deterministic.
   3. Create an infinite capacity (set capacity to a large number such as 100) resource called Postpartum.
   4. Make appropriate runs of your simulation model to confirm your answers to Questions 1-3.
      1. First let’s run it for two weeks. What is the average number of patients in the postpartum unit? Now run it for 4, 8, 16, 128, 500 and 1000 weeks, noting the average number of patients in postpartum, the number of arrivals, the number of discharges and the average length of stay in postpartum for each of these runs. Track the results in an Excel table.
      2. First let’s look at the number of arrivals to confirm we are generating the right number of patients. *Compute average number of arrivals per day in Excel*.
      3. Arrivals and departures and conservation of flow?
      4. Average length of stay?
      5. Now, look at relationship between run length and average occupancy of post-partum.
         1. What do you think is going on?
         2. How does the average postpartum occupancy compare to what we predicted via Little’s Law?
      6. So, let’s set *warmup time = 200* and the run time to *1000*  
         for this analysis.
      7. **New simulation concept**: steady state vs transient analysis.
         1. *10 replications*
      8. Now use the simulation model to confirm our analytical predictions we made in Questions 2 and 3 above.
      9. So, now we’ve seen how Little’s Law works. What happens if did NOT have an infinite number of beds? Does Little’s Law still apply to the number of FULL BEDS in postpartum?
         1. Set the average length of stay back to 2 days and the number of beds to 15 (we have to set the number of beds to at least 13, why?). Now, when patients arrive and all the beds are full, we’ll let them wait in a queue (what happens in real life?).
         2. Run the simulation. Let’s see if we can find out if and how Little’s Law still applies.
         3. Let’s add a Status Label from the Animation tab to show the current number of patients in process at the Postpartum unit.
            1. Set the Expression property of the Status Label to **Postpartum.Processing.Contents.NumberWaiting**. Can build this piece by piece by exploring the lists in the Expression builder.

# Patient Flow Model 02

Two patient types with different average length of stay

One unit with infinite capacity



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
| Patient arrival rate |  | 6 | patients per day | |
| C-Section percentage |  | 20% |  |  |
|  |  |  |  |  |
| Average length of stay |  |  |  |  |
| C-section |  | 3.5 | days |  |
| Regular delivery |  | 2.0 | days |  |
|  |  |  |  |  |
| Overall average length of stay |  | ? | days |  |
| Average occupancy |  | ? | patients |  |
|  |  |  |  |  |

What are the overall average length of stay and average occupancy in the postpartum unit?

Assume that the length of stay distribution for each patient type is a normal distribution with the means given above. Assume that the standard deviation is 0.5 days.

1. Build a Simio model and use it to confirm the overall average length of stay and the average occupancy that you found above. As we do this, we’ll learn a bunch of new basic modeling techniques.
   1. Different entity types with different processing times and graphics.
   2. Using tally statistics to get statistics by patient type.
   3. Using system variables like TNOW.
   4. Using entity states (aka attributes) to store timestamps and other quantities of interest.
   5. Using multiple Dispose blocks to get automatic statistics by patient type.
2. If the C-section rate decreases to 18% what will be the average number of C-section patients in postpartum? Use process physics and then confirm with simulation.
3. So far, we’ve seen how Little’s Law is pretty useful for exploring average occupancy in some system. Now, let’s make things a little harder. Save a copy of your model and we’ll make some changes to the copy.
   1. Set the capacity of the postpartum unit to 15.
   2. Run the model.
      1. What is the average wait time for each patient type?
      2. What is the overall utilization of the beds?
   3. Our goal is to find the number of beds needed so that no more than 5% of patients have to wait for a bed at all. For 15 beds, do we know what percentage waited at all? How might you add to your model so that we can figure out what percentage of patients had to wait to get a bed?  
        
      HINTS: We need to know the wait time for each patient

## Different entity types with different processing times and graphics

There are a few different strategies we can take in terms of number of Source objects. However, since multiple patient types will visit the same Server objects (e.g. post-partum unit) and will have different LOS distributions, we need to have a general approach to managing different parameters for different patient types. In Simio, the easiest way to do this is through a Data Table (Chapter 7). Tables can contain any number of columns and the allowable data types includes a wide variety of Standard Properties, Element References or Object References. Once the table is created, it can be referenced in a variety of ways (p219) in the model. Row selection from tables can be done randomly based on user specified probabilities or some rule. Often each entity will simply be referencing a specific row every time. Simio provides an easy way to implement this by setting a Table Reference Assignment in the Source object.

### Creating a Data Table for Patient Type data

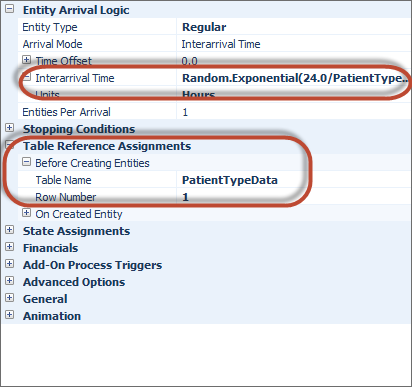
From Data tab (Model), add a Data Table and renamed it PatientTypeData. Added following columns:

PatientType – integer

LOS\_dist\_PP – Expression with Simio distribution such as Random.Normal(2.0,0.5)\*24.0.

AvgArrivalsPerDay – Real Poisson arrival rate. In this simple model, the arrival rate for each of the two patient types depends on the overall arrival rate (e.g. 6.0 pats/day) and the percent CSection parameter (e.g. 0.20). We could either specify the patient type specific arrivals rates directly in the Data Table or use an Expression that references the overall arrival rate and CSection probability. In general, for the OBSched model, we want to have a data driven way to specify the overall arrival rates for main streams and the various branching probabilities. We’ll do that in the next version of this model. For now, let’s just hard code in the 4.8 and 1.2 values for the arrival rates.

The Table Reference Assignment has to be made BEFORE the entities are created. I tried putting the mean arrival rates in the table but Simio complained that it couldn’t reference the correct table row.



PatientTypeData.LOS\_dist\_PP

Useful SimBits

Logic Based on Entity Property

Logic Based on Entity State

Entity Follows Sequence With Table

Entity Follows Sequence With Table2

Strategy 1: One source and one sink per patient type

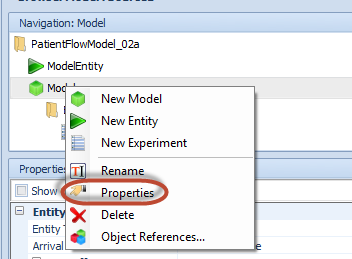
Model - PatientFlowModel\_02a.spfx

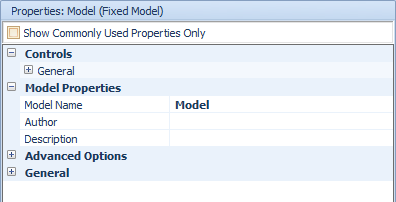
Individual sources makes it easy and transparent to see the different patient types. We can use a Data Table to store patient type related information such as the PP LOS distribution and use a Table Reference in the source nodes to sync the table with the sources.

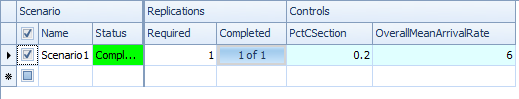
Individual sinks gives us built in stats by patient type. If we had a single sink and wanted patient type specific counts at the source, we’d have to do the work ourselves somehow.

## Using Model Properties for Input Controls

Turns out to be rather easy to define model properties for key inputs like PctCSection and OverallAvgArrivalRate. Add them as Properties to the Model object. Give them a default value in their Properties (yes Properties of a user defined Property) window. Then reference them in places like the Mean Interarrival Time property of a Source object. Model runs fine. However, if you run from the Experiment interface, all Model Properties get automatically added as Controls and you’ll see them in the scenario window. For some reason they don’t get the default value and instead default to 0. You can set them to whatever you’d like for each scenario. Voila.





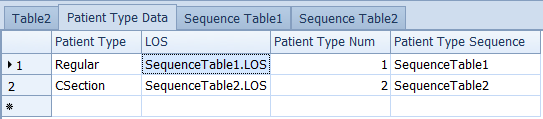


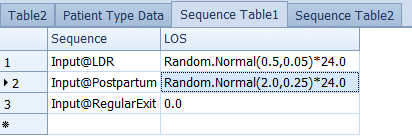
Instead of adding Tally Statistics and any other additional features to this model, let’s build a more complicated model in which different patient types have different sequences through the system. We can add a Tally Statistic to count the number of units visited by each patient (ideally, by patient type). This will force us to learn how to use Sequence Tables. In addition, let’s figure out how to write out transaction logs.

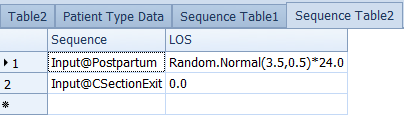
# Patient Flow Model 03

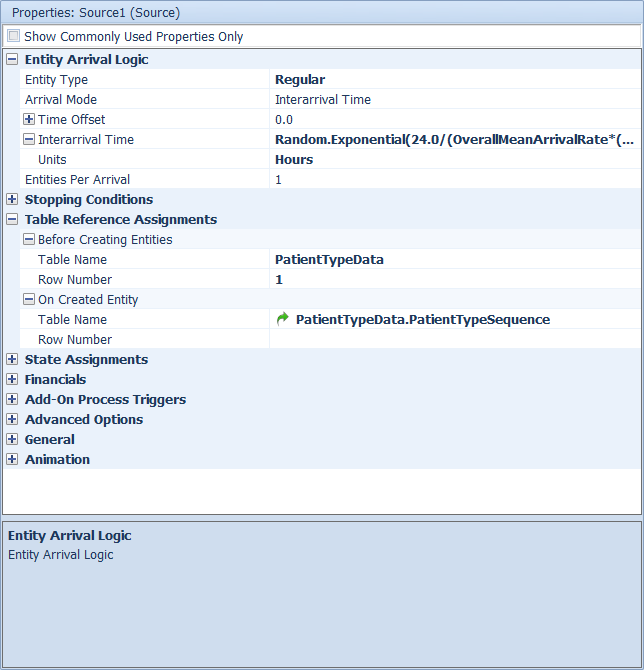
|  |  |  |
| --- | --- | --- |
| Patient Type | Route |  |
| 1 | LDR -> PP |  |
| 2 | PP |  |

Both units infinite capacity for now





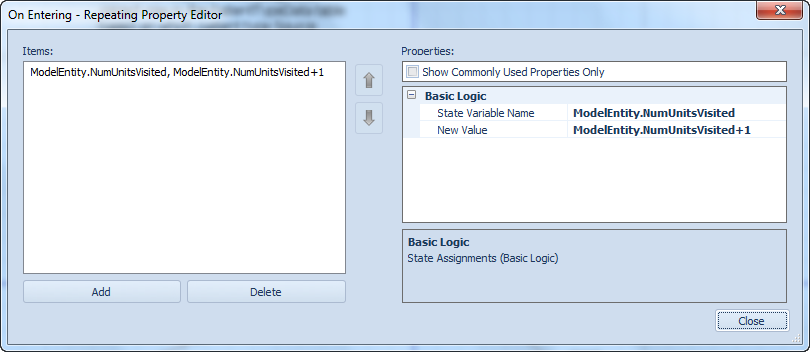






### State – NumUnitsVisited

Added a state to ModelEntity called NumUnitsVisited. Then did a State Assignment for each unit



### Tally – tlyNumUnitsVisited

When each patient exits, we want to tally NumUnitsVisited but we want to have separate tallies by patient type. One approach is described in the Simbit entitled Tally Statistics In Tables. The basic idea is to create the various tallies by part type and store references to them in a Data Table. Then actually do the tallies at the Sinks via an Add-On Process for each. See the Simbit example – it’s pretty straightforward.

We can also get automatic tallies with the new Interactive Logging feature. However, this only seems to work for Interactive Runs and not for Experiments.

### Manual Logging

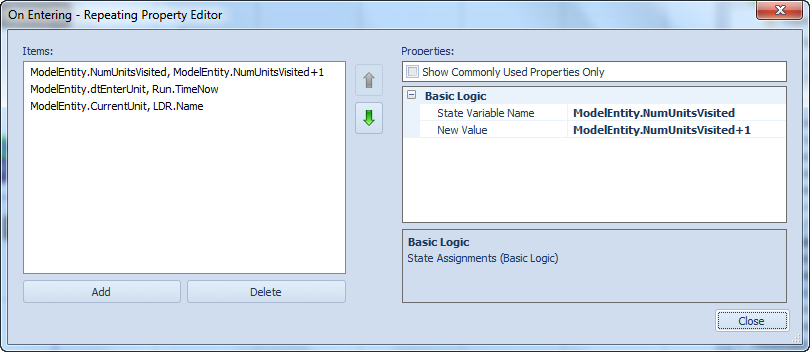
Let’s write out a file suitable for Hillmaker.

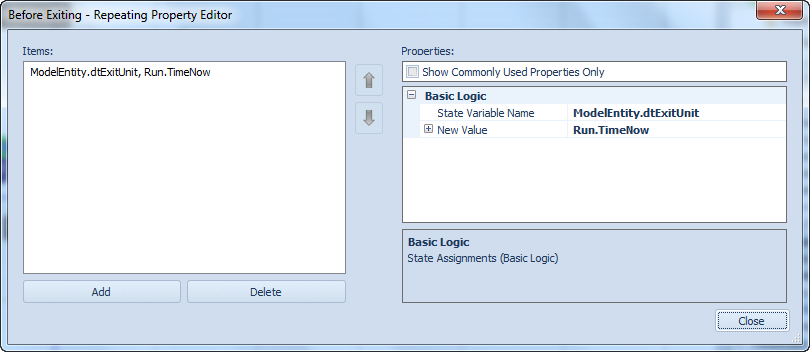
Define a File Element from Model | Definitions | Element | User Defined. Set the Name and FilePath properties of this new new object.

Define three new State variables for ModelEntity object:

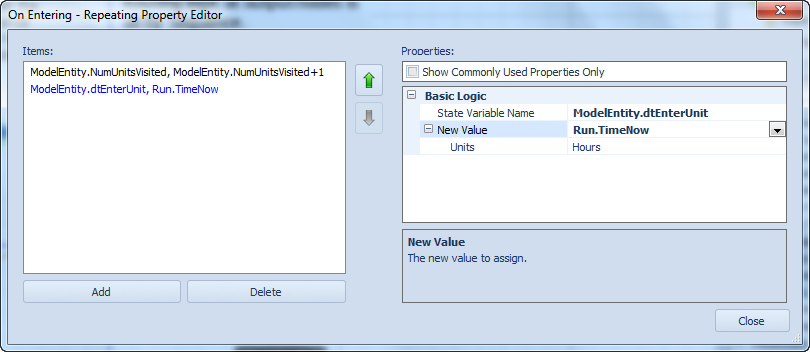
dtEnterUnit, dtExitUnit and CurrentUnit (string)

Can use the State Assignment property to set these on entry and exit from the unit. Getting the current simulation time is Run.TimeNow.



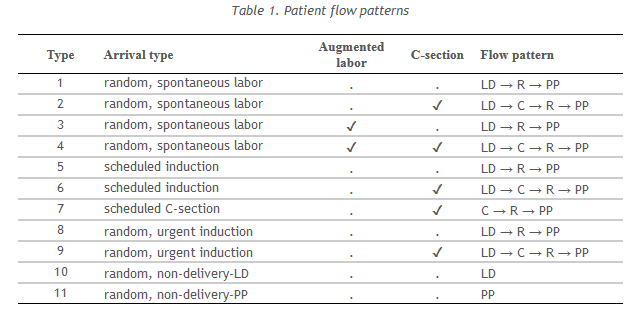


Since we reuse these for each visit to a unit, we need to write them out as we go. For that we use an Add-On Process. Need to be careful where that process is triggered. Originally I triggered it when Exit the output node of the server (e.g. Postpartum). However, the process wasn’t getting triggered until after the entity entered the next input node, which then caused the entry time to get updated before the stop record written. So, moved the trigger to the Exit of the server object itself.



# Patient Flow Model 04

Now, we are pretty much ready to rebuild the inpatient OB model to match the OBSched regime and Arena validation model.

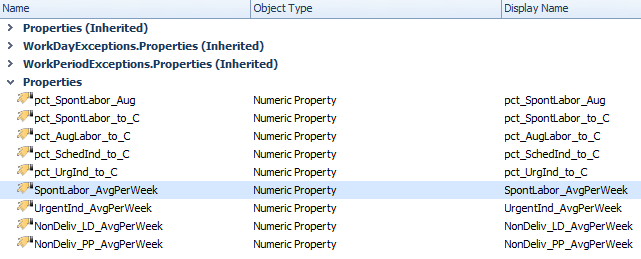


## Arrival Streams

There are six arrival streams – four of which are random arrivals and two are scheduled. Let’s build the random arrival streams first.

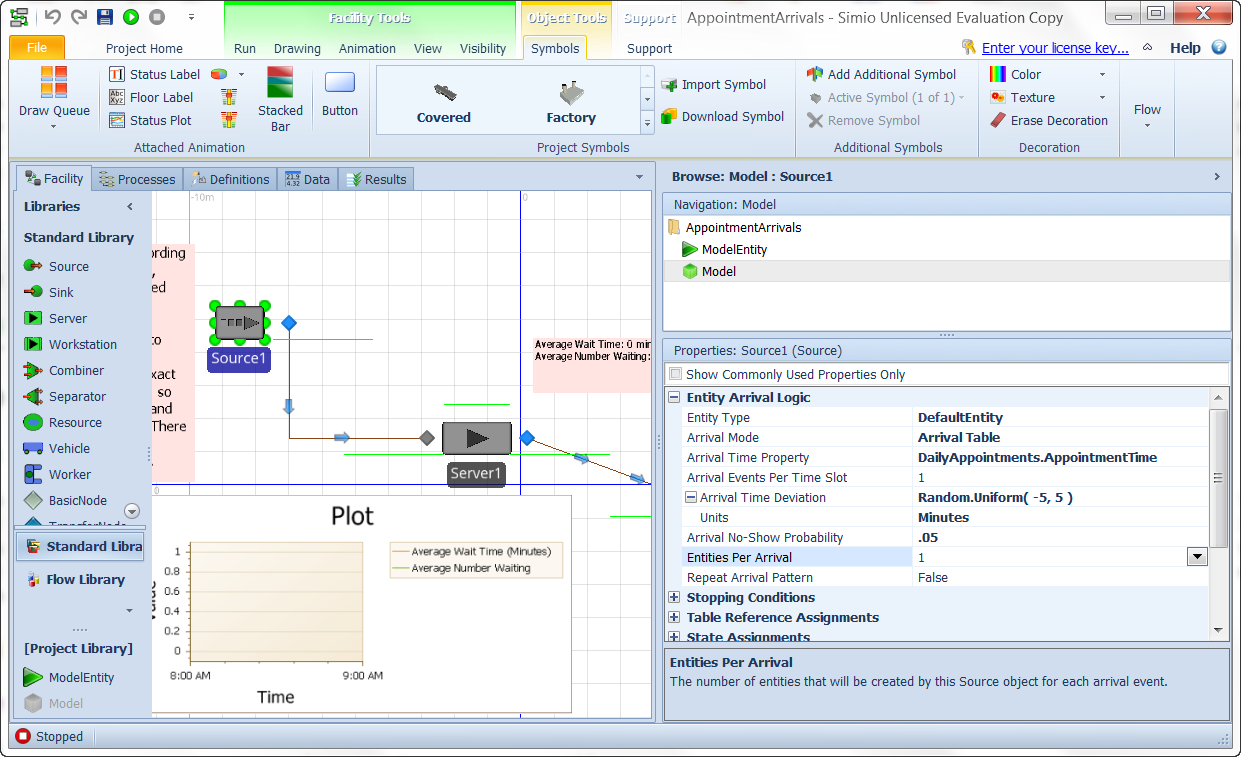
### Random arrival streams

Each random arrival stream gets its own Source object and a Model Property defining the average volume per week. In addition, the branching probabilities are implemented as Model Properties.



### Scheduled arrivals

Simio has some nice features built in for doing scheduled arrivals – deviation around scheduled time, no-show probability, appointment times in a Data Table. See Appointment Arrivals Simbit. I’m assuming we can set the EntitiesPerArrival via another column in the arrival table.



# Patient Flow Model 05

Now that

* we’ve got all the arrival streams (both random and scheduled) and all 11 patient types implemented
* Routing is done via a relational Sequence Table
* LOS distributions are in the Sequence Table
* Stop log is being written
* Scheduled arrival files are read in (CSV)

# OBSim02 – this was derived from Patient Flow 05 a while ago

Upon entry to each unit, the planned LOS is sampled based on the LOS distribution in the Sequence table. It’s stored in states like PlannedLOS\_LDR, PlannedLOS\_PP. So at discharge each entity knows their planned LOS at each location.

Our last major task it to implement unit blocking and LOS adjustments for blocked patients.

Set Output Buffer of LDR to 0 and Input Buffer of PP to 0.

Currently to timestamp the time a patient tries to enter a location, I’m using an Add-On Process at the On Enter event. However, if the input buffer capacity is 0, I’m not sure that this event is firing at the right time. It really needs to fire at the completion of the upstream processing.

# OBSim04

Add-on processes and entity states used to capture blocking due to insufficient capacity in downstream units. LOS in both LDR and PP are adjusted by subtracting off the time blocked.

Many, many experiments run with OBSim04.

Then, I talked to TJW. He had many ideas for improvement.

# OBSim05

TJW suggested numerous improvements

|  |  |
| --- | --- |
| TJW | Resolution |
| LOS in LDR should be adjusted by time blocked in triage. However, once baby is born, time blocked in LDR waiting for PP is largely irrelevant. | Manually removed the subtraction of blocked time from the PP processing time property. To make things more flexible, I could add a toggle to control this. |
| The best way to model PP LOS, if  you ask me, is this:  -  patient arrives on PP  whenever they get there on the first PP day  -   LOS in days is best modeled by a distribution of 1, 2 or 3  days for vaginal or 2, 3, or 4 days for csec.     -  on the discharged day, the discharge  time is selected from the appropriate LOS distribution for  time of day.  It's the same no matter how many days the  patient was on the PP unit.  So, interestingly  enough, arriving early in the AM at PP actually increases  LOS.  Arriving late, say 1800 or so, decreases overall PP  LOS.  If you want to shorten PP LOS, the best thing to do  is schedule procedures later in the afternoon and move the  patient to PP between 1800 and 2100 in the evening. | Simio has a Random.Discrete() function. However, we need to use this just to determine the “discharge day” and then use some other method to decide on the actual discharge time. Seems like we should do the math logic within a callable process (i.e. to mimic a UDF).  NOT DONE YET |
| I would rather have birth vol in 1,000 birth increments from 1,000 thru 8, or 9,000 |  |
| I would rather c-section rates of 20, 25, 30, 35 and 40% |  |
| Labor LOS  is very dependent on labor type  -  spontaneous  labor, vaginal birth/csec  -  augmented labor,  vaginal birth/csec  -  induced labor, vaginal  birth/csec    -  -  should be a variable 10,  20, 30 and 40% of total birth vol    -  -   the non-induced patient volume should be split evenly  between spontaneous and augmented labor    -   -  can have different probabilities for vag birth vs csec  delivery for the three labor types aboveOf  course, scheduled csec patients do not spend any time in  labor.  Rather, these patients go straight to the pre-op  area. |  |
| I  have a few very descriptive LOS distributions for each of  the patient types noted in c aboveFor PP, there  are only two patient types, vag birth and csec delivery.   PP LOS is independent on anything that happened in labor  except how did the baby come out. |  |
| Mark, the big problem these days is the  exploding induction rates.  Induced patients have more than  double the LDR LOS in labor.  This is primarily one-on-one  nursing and LDR room consuming for an additional 11 or 12  hours, on average.  So, induced patients consume LDR rooms  and csec patients consume PP rooms.  Spontaneous labor,  un-augmented labor, vaginal birth patients - the natural way  - is far more efficient and frugal regarding resource  consumption.  Induced labor that results in a csec is the  most expensive patient type on the planet. | Question is how much volume to schedule. |
|  |  |
|  |  |
|  |  |
|  |  |