**Call Centre DES Architecture changes**

**Altered Classes**:

**Sales Manger**

* Computing busy reps | Method

I added this method to prevent the need to foreach over all the reps checking who is busy within the simulation methods themselves making the Sales Manager responsible for its sales reps

**Stats**

* Calc Average Time In Queue | Method

I added a method to compute the average time of queue because I didn’t realize I would need it until later on when I focused more on the stats.

* Increment Excessive Wait | Method

I added a method to compute the excessive wait time, because I didn’t think of this during the initial design stages.

**Entity (Call)**

* Calc Time In Queue | Method

I was originally going to add this method in the stats but then though it belongs more to the entity, therefore I moved it the entity.

* Calc Time In System | Method

I was originally going to add this method in the stats but then though it belongs more to the entity, therefore I moved it the entity.

* Compute Call Type | Method

I was originally going to use the simulation to compute the entity’s call type, but decided it was better for the entity to this meaning that the simulation never has to alter the call type property of the entity.

**Event Comparer**

Removed the duplicate comparers and the event uses it rather than the calendar (this was a mistake when rushing to hand in).

**Simulation**

Initialize | Method

I added an initialize method as there was no way to seed the calendar to get the system to actually run.

**Global Parameters**

* + SPEED\_INTERVAL | Variable
  + NUM\_STEREO \_REPS | Variable
  + NUM\_OTHER\_REPS | Variable
  + SECS\_IN\_MIN | Variable

**Added Classes:**

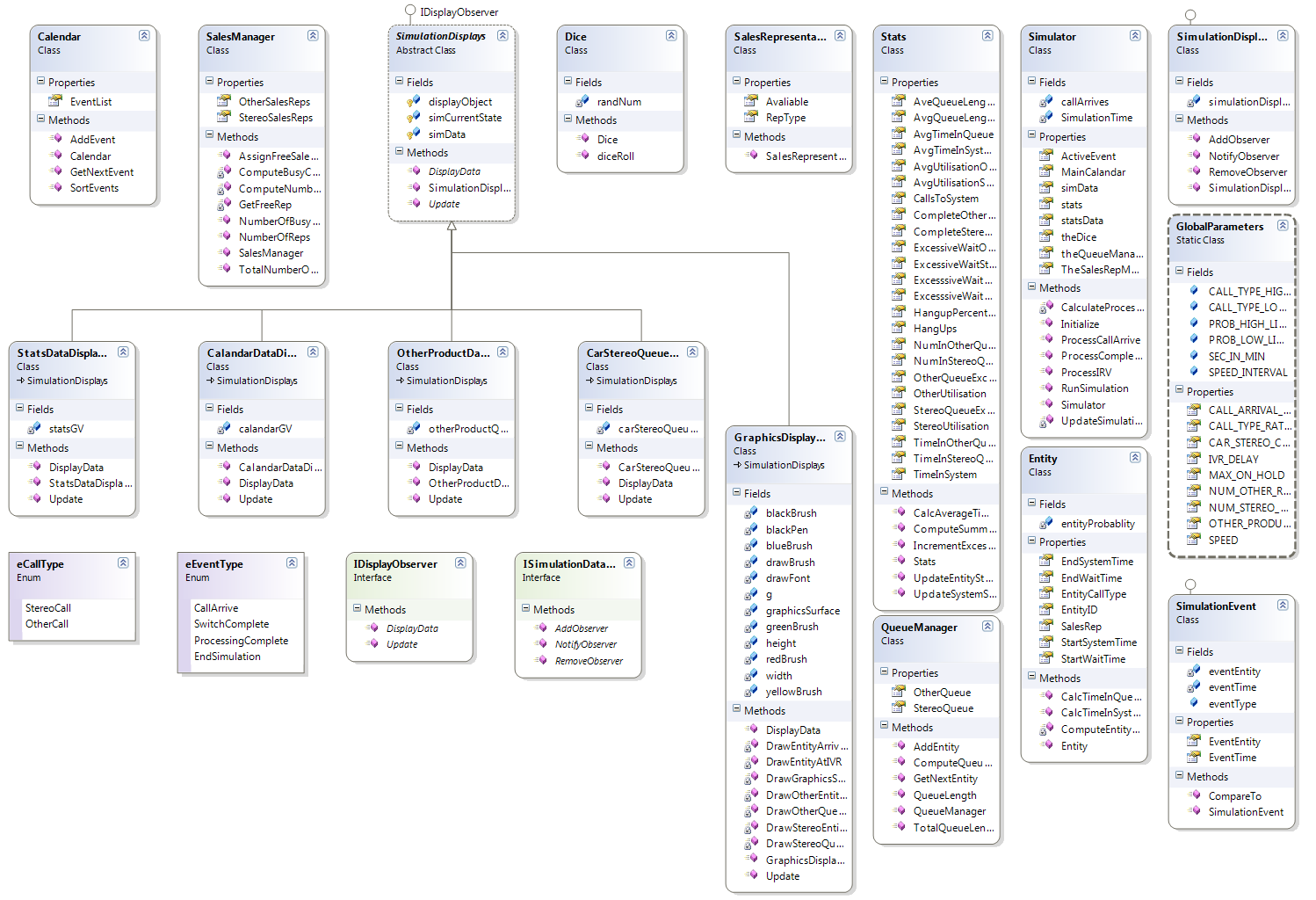
I was originally going to have the methods for displaying the stats and information to the screen within the stats class, but then I though why is the stats class handling the displaying.

This also meant that the simulation would of need the whole form. So I used the observer pattern and a display hierarchy which allowed for ease of extension on the displays for the simulator. The object passed into the display observer classes is of the type Object which allows for more flexible classes as I can now pass and object of any type through the constructor as long as you cast the object to the correct type. I viewed this as being a good approach in the sense that the user never interacts with what gets passed into the construer.

Below are the classes added for the simulations display.

* ClandarDisplayObserver
* CarStereoQueueDataDisplayObserver
* OtherProductDataDisplayObsever
* GraphicsDisplayObserver
* ISimulationDataSubject
* SimulationDisplayDataSubject
* IDisplayObserver
* SimulationDisplays
* StatsDisplayObsever
* Dice

I ended up adding a dice class because I forgot that I needed one when first looking at the program, although it was mentioned in the pseudocode.

**New Class Diagram**