New changes to sequence and class diagram

All class diagram changes for Assignment6 are highlighted in bold.

Class Diagram Changes and Explanations

1. How is replay implemented?
   It is implemented using collection (**busCollection hash in BusStatusHistory class).** Bus object store route and route has all the stops. Since Bus object is cloned and added busCollections, as soon as replay is requested, the data can be retrieved from busCollections and restore to previous bus state.

2. How are ”bus changes” designed?
   Bus parameter changes are updated once bus reach a stop. **uml_seq_diagram_movebus** has that updated at the beginning of the move_next_bus process. It then goes through the normal process recalculating the time taken to next stop and select the bus based on the logical time using method populateTimeStops(). During this calculation populateTimeStops() functions is used to populate the time taken to next stop and arrival times to different stops for a bus. getTravelTime() in the MoveBus() section is only getting travel time and priority calculated earlier part of sequence. populateTimeStops() calculation is required for move_next_bus, since route, speed and passenger capacity can be changed(can be effective) at any stop during the simulation.

3. How are “passenger exchanges” designed?
   Passenger changes happens at the stop level and all of 4 “uniform distribution functions” are part of stop class. Those function will be called when bus reach a stop. Data is stored in the stop class used.

4. How is “system efficiency” implemented?
   Waiting passenger information is updated to stops collection of TransportSystem. Bus speed and bus capacity are updated to bus collection in the transport system. This information is updated at every stop of every bus at the end of processing at stop. Waiting passengers are calculated from **SystemEfficiency class.** SystemEfficiency class will use its methods and TransportSystem collections to calculate the efficiency.

Sequence Diagram Changes and Explanations

There are 3 sequence diagrams.

uml_seq_diagram_movebus.pdf : This is the main sequence diagram for moving the next bus.

uml_seq_diagram_reset_and_replay.pdf: This one contains both replay process and reset process for assignment6. The approach used is the same explained in the class diagram section. For the simplicity of the design, resetAll() will restart the process by reloading the file.

uml_seq_diagram_systemefficieny.pdf: This one contains sequence for calculating system efficiency as part of move bus process.

Other information

Busmovement is a simple class used to keep track of completed bus movements. It is used as a collection in SystemSimulator class.

Other Corrections

Besides "Assignment2" classes Bus, Route, Stop and TransportSystem has been corrected a little as well and those changes are not highlighted. Collection initiation methods and maintenance methods are moved to Transport system. Some of new attributes and methods were required to maintain the simulation events and state of the simulation and those were not added in the initial diagram, and that is added now. This is based of application developed during assignment5.