CS4272 Assignment 1

Due on 20 September 2007

1 Notes

- This is an individual assignment. Acts of plagiarism are subjected to disciplinary action by the university. Please refer to

  http://www.comp.nus.edu.sg/students/plagiarism/

  for details on plagiarism and its associated penalties.

- Submission Instructions: (Failure to follow these instructions may result in deduction of marks.)
  - Create a folder named your matriculation number **YourMatricNumber**, e.g. U123456M. Create the following files in this folder (name these files exactly as instructed.):
    - **assignment1**: Rhapsody project folders of your assignment. The animated sequence diagrams must be included in the projects.
    - **report.doc** or **report.pdf**: please include your particulars (name, matriculation number and NUS email address), and assumptions you made in system modeling (if any).
  - Zip (using WinZip) the entire YourMatricNumber folder (including the folder itself and all files in it) into a file YourMatricNumber.zip.
  - Submit YourMatricNumber.zip to the IVLE Workbin Folder *Assignment 1 Submission*. Submit only *one* copy.

2 Problem Description

A railway system consists of interconnected stations. Shuttles running on the railway bid for orders to transport passengers between certain stations. Successful completion of an order results in a cash payment for the shuttle involved. New orders are made known to all shuttles, thus all shuttles can make an offer. The shuttle with the best, i.e. lowest, offer will receive the assignment.

- Railway network
  - The railway network consists of stations and tracks. Every station has a unique index. Track can be traveled by shuttle in one direction only (which is fixed). There are four stations/tracks labeled from

0 to 3 connected as a ring, and only one track resides between two stations. Specifically, a shuttle can only travel from station 0 to 1, station 1 to 2, ..., and station 3 to 0.

  - A track can only be occupied by one shuttle at a time. Shuttles willing to travel along the occupied track have to wait until the track is free. That is, when a shuttle receives the “depart” event, it should not proceed if the track to be traveled along is occupied.
  - The railway network has a management system. Orders are only available from environment to the management system, and then it should broadcast them to all shuttles. (You can also specify orders in the management system object or during model initialization, and it broadcasts when some event from environment is received.) That is, no order should be sent from environment to shuttle directly.

- Orders
  - Whenever a new order is available to the management system, it sends the order to all shuttles as event “neworder”. An order will contain:
    * The start station (identified by its index).
    * The destination station (identified by its index).
    * The number of passengers to travel - the order size.
  - After the shuttle receives the event “neworder”, it must reply with either an “offer” or “refuse” message. The offer should include the desired charge/fare - the payment it will receive. (You may use event with argument for fare and denote “refuse” with a very large fare.) The shuttle having made the lowest offer will receive the “assignment” from management system. When two shuttles offer the same (lowest) fare, any one can be assigned.

- Shuttles
  - Every shuttle has its capacity (n passengers) and its fare (i dollars per passenger) predefined. These two values are not changed during model execution.
  - When a shuttle is at a station, it can receive the “depart” event to travel to the next station. Then its state changes to reflect that it is on track. Note that it should not proceed if the track is occupied. When it is on a track, it can receive the “arrive” event to change its state as being in the next station. Otherwise it should ignore the event. However, a shuttle cannot move unless it gets an assignment of order.
  - To complete an order a shuttle has to travel to the start station, load the order and then proceed to the destination station to unload. After the shuttle unloads, the assignment is completed. Loading or unloading should be performed automatically at corresponding stations. Loading / unloading at other stations is not permitted. That is, the shuttle should change its state to reflect loading / unloading automatically.

  - When an order is received, a shuttle should make an offer only if (1) the order size does not exceed the capacity, and (2) it does not have an assignment. Otherwise the shuttle should refuse to make an offer.

More details:

- Initially each shuttle should be in a station.
- States in statechart should clearly reflect the status of the corresponding class/object. For example, the status of a shuttle is different when it stays in a station or runs on a track, it is loaded or unloaded, and it has order or no order.
- You may need to write Java code to realize some functionalities.
- For each class/event, you can view the code generated for it by right click it from "Packages / Default", and choose "Edit class".
- Make sure your names for events and classes do not start with the letter "u". It will cause some bugs in code generation.
- An event *e* can be specified with an argument *arg*. The argument *arg* can be used in the action part of transition as: `params.`*arg*.

3 Questions

1. (6 marks) Use "Rhapsody in J" to model the shuttle system according to the specification given above. Your model should contain the class diagram and statechart for all classes. Your class diagram should contain **at least** three classes - the management system, tracks, and shuttles, as well as corresponding multiplicities and associations. Please state clearly in the report any assumptions you made during modeling of the system.

   Your model should comply with the requirement strictly, otherwise marks may be deducted. For example, a shuttle should only unload (by entering "unloading" state) if it is at the destination station.

   Note: "load", "unload", "offer", and "assignment" should not be modeled as external triggers from environment.

2. (2 marks) Suppose there are three shuttles - $s1$ (capacity: 5 passengers; fare: 2 dollars per passenger; initially at station 0), $s2$ (capacity: 10 passengers; fare: 4 dollars per passenger; initially at station 0) and $s3$ (capacity: 8 passengers; fare: 3 dollars per passenger; initially at station 1). Generate an animated sequence diagram named "q2Sequence" for the following scenario, showing **at least** objects of management system, shuttles and tracks.

   (a) An order of 5 passengers from station 1 to 3 is available to all shuttles.

   (b) $s1$, $s2$ and $s3$ make offers and $s1$ is assigned with the order.

   (c) $s1$ departs from the current station 0.

   (d) $s1$ arrives the intermediate station 1, and loads.

(e) An order of 2 passengers from station 1 to 2 is available to all shuttles.

(f) $s2$ and $s3$ make offers while $s1$ refuses to offer.

(g) $s3$ is assigned with the order.

(h) $s3$ loads 2 passengers.

(i) $s3$ departs from current station 1.

(j) $s1$ tries to depart from station 1 but failed.

(k) $s3$ arrives next station 2.

(l) $s3$ unloads 2 passengers.

(m) $s1$ departs from current station 1.

3. (2 marks) For the above system, produce a sequence of events that leads the system into a state that all shuttle are at stations without load, given the following condition:

   - Two orders are available sequentially before any action of shuttles.
     - The first order: 4 passengers from station 0 to 2.
     - The second order: 9 passengers from station 1 to 2.

   Generate an animated sequence diagram named "q3Sequence" corresponding to this scenario.

   Note: if your model requires different settings to generate sequence diagrams for question 2 and 3, please state the settings required clearly in your report.