Repair shop activities

Events portray the activities that go by at an automobile repair shop somewhere in Aveiro.

There are five main locations: the *park* where the customers park their cars and retrieve them after the repair has been concluded and where some replacement cars are parked for eventual customer use, the *lounge* where the manager stays and process customers complains, the *repair area* where the mechanics carry out the operations to fix the cars, the *supplier site* where the manager goes to obtain parts that are not presently available at the shop, and the *outside world*, where customers stay performing their day-to-day activities.

There are three kinds of intervening entities: the *manager* who runs the repair shop, the *mechanics* who repair the cars and the *customers* who own the cars.

N customers visit the repair shop during the day. M mechanics are at work to fix the cars. T replacement vehicles for customer use are present. A repair always requires the replacement of one part out of K that in principle are available at the repair area.

The activities are organized as described below

- the customer arrives at the repair shop sometime during the day and parks her/his car;
- the customer goes to the lounge and queues in to talk with the manager;
- the manager deals with customers one at the time: collects the car key, posts the job to the repair area and asks if a replacement vehicle is required; if so, she assigns one to the customer;
- the customer waits for the key of the replacement vehicle, if that was the case, and leaves the repair shop;
- the mechanics, upon receiving a repair order, go to the park to collect the vehicle, check which part is necessary to be replaced and get one, if it is available; if not, they contact the manager to replenish the stock and proceed to the next order, while waiting for the part to arrive;
- when the repair is concluded, they drive the vehicle back to the park and alert the manager of the fact;
- the manager, upon receiving a notice of an empty stock, goes as soon as possible to the suppliers site to replenish it;
- the manager, upon receiving a notice that a service has been concluded, contacts the customer to come and collect her/his car,
- the customer arrives at the repair shop and, if she/he is driving a replacement vehicle, parks it; then, goes to the lounge, queues in to pay for the service and collect her/his car key and leaves the repair shop;
- when all the cars have been repaired, the manager sends the mechanics home and calls the day off.

Assume there are thirty customers, two mechanics, three replacement vehicles and three different types of parts. Write a simulation of the life cycle of the customers, the manager and the mechanics using the client-server model, with server replication, where the customers, the manager and the mechanics are the *clients* and access to the information sharing regions are the services provided to them by the *servers*.

The operations that were previously assigned to activities carried out in the information sharing regions (for the already implemented concurrent version), must now be assigned to independent requests performed on the servers through message passing. In each case, a connection has to be established, a request has to be made, waiting for the reply will follow and the connection has to be closed.

One aims for a solution to be written in Java, to be run in Linux under TCP sockets, either in a concentrated manner (on a single platform), or in a distributed fashion (up to 9 different platforms) and to terminate (it must contemplate service shutdown). A *logging* file, that describes the evolution of the internal state of the problem in a clear and precise way, must be included.

Guidelines for solution implementation

1. Specify for each representative server of an information sharing region the structure of the messages to be exchanged.
2. Specify the general organization of the servers architecture.
3. Specify the general organization of the clients architecture.
4. Sketch the interaction diagram which describes in a compact, but precise, way the dynamics of your solution. Go back to steps 1, 2 and 3 until you are satisfied the description is correct.
5. Proceed to its coding in Java as specific reference data types.
6. Specify the mapping of the servers and the clients onto multiples nodes of the parallel machine and write the shell scripts which enable the deployment and the execution of the different modules the application is composed of.
7. Validate your solution by taking several runs and checking for each, through the detailed inspection of the logging file, that the output data is indeed correct.