Software Specification Project

In our project, we developed 5 machines:

Machine 1: Clients. In this machine, we modelled the initial context, related to our system's clients. In this model, we are considering some constant sets EMAILS, CLIENTS, NAMES and PLANS. EMAILS contains all possible emails from everyone who uses our system, CLIENTS contains all possible client Ids, NAMES contains all possible names and PLANS contains VIP and REGULAR, to be able to model the client's possible plan. This static part is stated on the ClientContext. As for the dynamic part, we consider several sets, REG_CLIENTS, REG_EMAILS and REG_NAMES, just like function relating clients with their emails, names and statuses.

Machine 2: Drivers. In this machine, we refined the Clients machine, related to our system's drivers. In this model, we are considering some constant sets DRIVERS and LICENSES. DRIVERS contains all possible driver IDs from every possible driver in our system and LICENSES contains all possible driver licenses. This static part is stated on the DriverContext. As for the dynamic part, we consider the sets REG_DRIVERS and REG_LICENSES, just like function relating drivers with their names and licenses.

Machine 3: Cars. In this machine, we refined the Drivers machine, related to our system's drivers. In this model, we are considering some constant sets CARS and ZOBERSERVICE. CARS contains all possible cars in our system and ZOBERSERVICE contains ZOBER_Y and ZOBER_WHITE. This static part is stated on the CarContext. As for the dynamic part, we consider the sets BANNED_DRIVERS, REG_CARS, etc. (same logic as always) just like function relating cars with their owners, services or drivers.

Machine 4: Rides. In this machine, we refined the Cars machine, related to our system's rides. In this model, we are considering the constant set RIDES, which contains all possible rides. This static part is stated on the RideContext. As for the dynamic part, we consider the sets SCHEDULED_RIDES, BUSY_CARS, AVAILABLE_CARS, etc., just like functions relating rides with their begins, ends, grades and associated cars.

Machine 5: ZoberSharing. In this machine, we refined the Rides machine, modelling some extra features. In this model, we are considering a "magic" function same_route, which tells us, given a ride, which rides are on the same path, on the same beginning time. This static part is stated on the ZoberSharingContext. As for the dynamic part, in order to model the best 20% restriction, we "split" the NEW_RIDE event in 2 events : NEW_RIDE for a single client and NEW_RIDE_SHARED when there are several clients sharing a ride. We also added an event to help us manage the cars that reached its maximum capacity (2 or 4 clients), CAR_MAX_CLIENTS.