Software Specification

Second Project Report

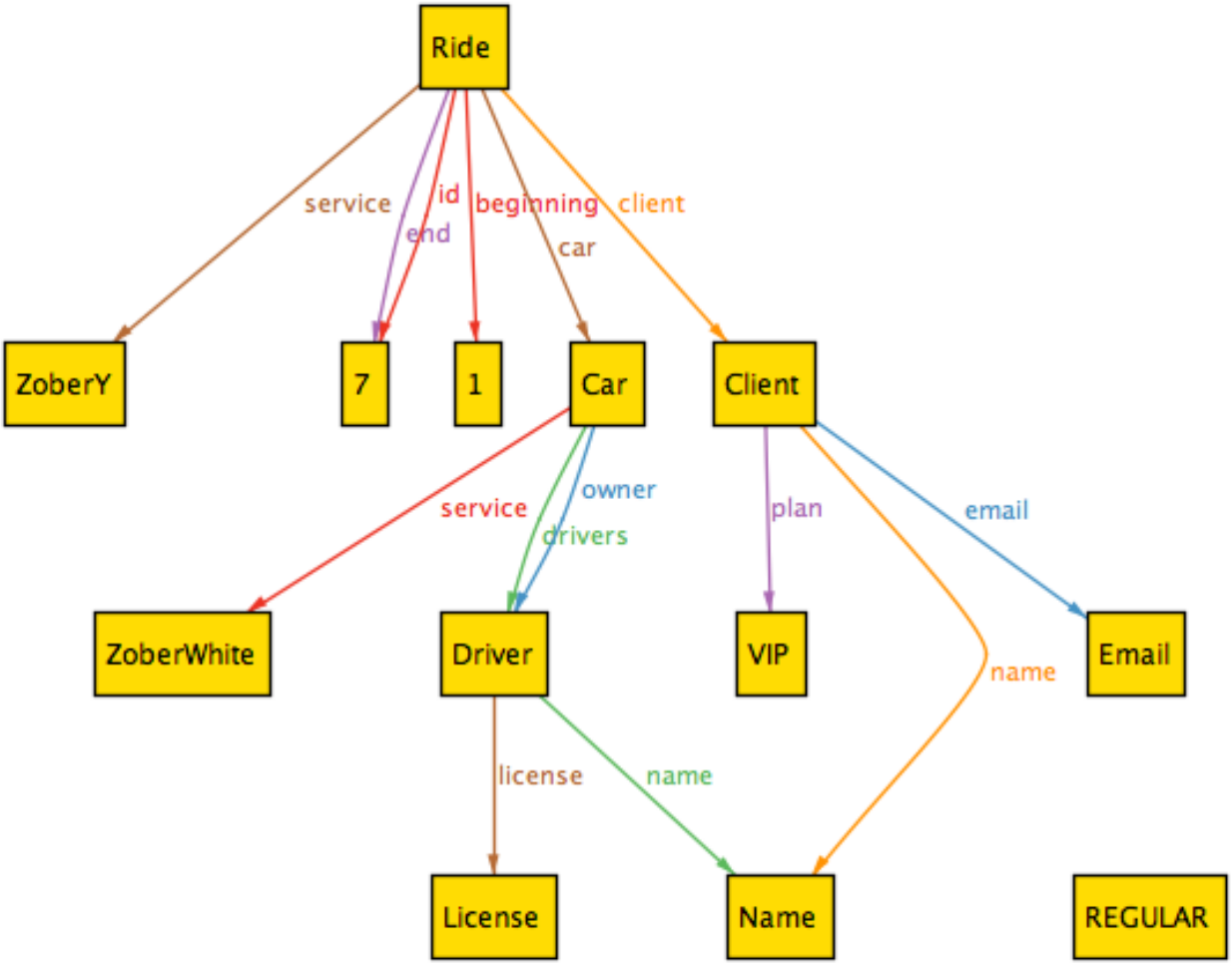
Jorge Veiga (75154) Rodrigo Bernardo (78942) André Santos (84912)

Signatures

1. *Zober*: Elements of this set represent the system possible states. Each state is related to a set of clients, a set of drivers, a set of banned drivers, a set of cars and a set of rides. These states are ordered and its through that ordering that we represent the dynamics of the model.
2. *Name and Email*: These are simple signatures that represent the sets of all names and emails, respectively.
3. *Plan, VIP and REGULAR*: These are simple signatures representing the possible client plans. *Plan* is an abstract signature whose only subsets are *VIP* and *REGULAR*, which are disjoint.
4. *Client*: Represents the set of clients. In our model, each client is related to exactly one instance of *Name*, one instance of *Email* and one of $REGULAR + VIP$ (in this case, it could be related to an instance of *PLAN*).
5. *Driver and License*: *License* represents the set of all possible licenses. Each driver in *Driver* must be related to a license, but must also be related to an instance of *Name*.
6. *Car*: Represents the set of all possible clients. Each car has exactly an owner (which must be an instance of *Driver*), must be driven by at least one driver and provides exactly one level of service.
7. *ZoberService, ZoberY and ZoberWhite*: Akin to the *Plan* case, we have the set of all possible car services represented in *ZoberService*, with *ZoberY* and *ZoberWhite* being its only subsets, which are disjoint.
8. *Ride*: Represents the set of all rides. Its attributes are as described in the specs: each ride is related to an id (Int), the car and client for that ride, the level of service and the time window of the ride. It may also be related to a rate, whether the ride is finished or not. Two things should be noted: the id is completely superfluous right now, as every atom is uniquely identifiable.; also, the *beginning* and *end* functions could map each ride to some ordered type, as we don't rely on any integer properties besides total order.

Dynamics

The dynamic part of the model was achieved using the traces technique. We make the signature *Zober* an ordered type where each element represents a possible state of the system. Moreover, with express as a fact (called *traces* in the model) the properties that the initial state must hold along with all the possible state transitions.



1. One of Zober's possible instances, projected over Zober.