Beastly Heis v1.5

Kolbjørn Austreng, Andreas Våge

January 29, 2017

Diagrams

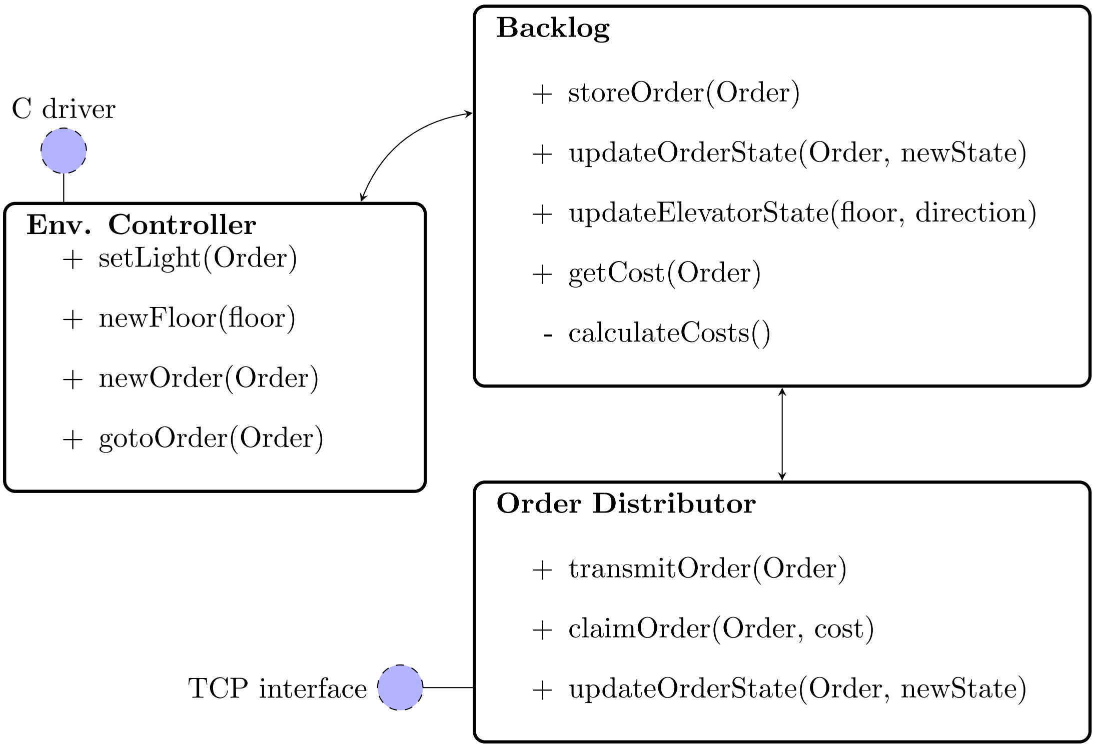

Figure 1: System module diagram.

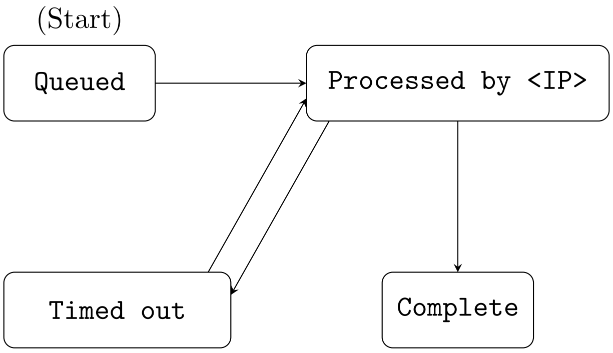

Figure 2: Order life stages.

Order object

Order	Comment
+ type	Internal / External
+ floor	Destination floor
+ timestamp	Set by computer that first received order
+ origin IP	Set by computer that first received order
+ state	Queued, In progress, Timed out, Complete

Environment Controller

+ setLight(Order)

Sets the light corresponding to the floor of an Order object.

+ newFloor(int floor)

Call from C driver to communicate to *Environment Controller* that a new floor has been reached.

+ newOrder(Order)

Call from C driver to communicate to *Environment Controller* that a new order has been created.

+ gotoOrder(Order)

Sends the elevator to the floor of a specific order.

Backlog

+ storeOrder(Order) ok

Saves an order from either *Environment Controller* or *Order Distributor* to the backlog. Returns acknowledgement.

+ updateOrderState(Order, newState) ok

Changes the state of a specific order. Returns acknowledgement.

+ updateElevatorState(int floor, enum direction

Communicates the position and direction of the elevator to the *Backlog*.

+ getCost(Order) cost

Returns the cost of taking a specific order for this elevator.

- calculateCosts()

Calculates the costs of all the orders in the backlog for this elevator.

Order Distributor

+ transmitOrder(Order) ok

Transmits an Order object to all the other nodes in the network. Acknowledges if at least one other elevator received the order transmit, or there are no other elevators in the network.

+ claimOrder(Order, cost) ok

Attempts to claim an order in the *Backlog*. Transmits own cost of taking on this order. Acknowledges if no other elevators have a lower cost on the specified order.

+ updateOrderState(Order, newState) ok

Broadcasts an order state update to ensure that the backlogs are identical.