

BridgeSegment behavior:

My understanding of this homework is that cars can arrive asynchronously at the ports of Bridge 3 (west side: westbound_in) and Bridge 1 (east side: eastbound_in), and only one car can be let on the bridge at a time, with a travel time of 10 units. For a bridge to let the car in, it has to satisfy three conditions:

1. `currentCar == null` means that it doesn't currently serve any vehicle.
2. The direction has the green light (`currentLightState`). For example, `currentLightState = WEST_TO_EAST` means the green light is from left to right.
3. And the light remaining has enough time for a car to cross (`lightRemaining >= travelTime`).

After completing services, the bridge outputs the car via the `out()` function at the appropriate output port. Right after outputting the car, `delint()` clears the car and tries to let another car come in if there is enough green light left. When the light remaining reaches 0, the light switches to another direction and resets the light remaining = initial light duration, and it tries to serve another car in the queue.

deltext(double e, message x) - External transition

Continue(e) to synchronize the model's internal timing with the DEVS engine, which also updates the sigma. `lightRemaining -= e` is essential to update the light time left. First, we attempt to queue all the cars arriving at each port (here is `westbound_in` for Bridge 3 and `eastbound_in` for Bridge 1). I check if the bridge is free to serve the car by checking `currentCar == null`, and attempt to serve the car. First, I check the `currentLightState`, then check if there is a car waiting in the queue for that direction, and the green light remaining time is long enough for a car to pass. If it satisfies three conditions, the car can go through the bridge and set the next internal event `holdIn("serving", travelTime)`. The model remains passive if conditions are not met,

out() - Output function

This function is called immediately before the `deltInt()` function takes place. When the phase "serving" sigma time expires, the `out()` function will output the car at the appropriate output port by checking the current light state.

deltInt() - internal transition (handles service completion or switches light direction)

When I use `holdInt("serving", travelTime)`, the DEVS framework updates `sigma = travelTime`. Therefore, to keep the light remaining correctly, sigma must be subtracted from `lightRemaining`, because it must reflect how much real time has passed. Sigma is the time unit passed during the internal event.

I check if the last phase happened was "serving", then I clear the `currentCar`, and check if the bridge can start to serve another car that satisfies three conditions: `lightRemaining >= travelTime`, current light is green, and that car is currently in the queue. If not, the bridge remains passive with the light left. Light switching is performed whenever `lightRemaining` has run out.

lightSwitch() - handle light switch logic

If there is still time left on the light and no car is being served, it remains passive with the time left. Switching `currentLightState`, reset the light to the initial light duration corresponding

with the bridge. After switching the light, I can still attempt to serve the car if it meets conditions like above; if not, it just remains passive.

deltcont(e, x) - Confluent transition

To process the internal transition function first, then the external transition function if both function happens at the same time.

