Queue

listEvents: ArrayList<Evnt>
rewindList: ArrayList<RewindEvnt>
currentEventId: int
simulation: Simulation
replay_flag: boolean

addEventToPool(int eventIndex, int eventRank,
        String eventType, int objectId)
addRewindEvnt(int id, int busId, int rank,
        String type, int stopIndex,
        int numPassengersAtStation,
        int numPassengersOnBus,
        int oldCapacity,
        double oldSpeed)
chooseNextEvent( )
choosePreviousEvent( )
updateEventExecutionTimes(int eventIndex,
        int eventRank)

Event

id: int
rank: int
type: string
busId: int

RewindEvnt

id: int
busId: int
rank: int
type: string
stopIndex: int
oldNumPassengersAtStation: int
oldNumPassengersOnBus: int
oldCapactiy: int
oldSpeed: double

Location

latitude: double
longitude: double

Bus

id: int
routeId: int
routeIndex: int
newRouteId: int
newRouteIndex: int
numPassengersRiding: int
numPassengersUnloading: int
numPassengersLoading: int
maxCapacity: int
avgSpeed: double
travelTimeNextStop: int
distanceNextStop: double
stopId: int
previousRouteIndex: int
simulation: Simulation
isFirstStop: boolean
isNewRte: boolean
lastStopLastRte: int
previousStops: List

getPreviousStopId( ): int
getRewindPrevStopId( ): int
getCurrentStop(): Stop
getNextStop( ): int
subtractRiders(int num)
addRiders(int num)
getNumAvailableSeats( ): int
calculateDistance(int nextStopId): double
calculateTravelTime(double distance): int
changeRoute(int routeId, int stopIndex)

BusRouteChange

new_route_id: int
new_route_index: int

BusCapacityChange

new_capacity: int

BusSpeedChange

new_speed: double

BusChange

ChangeType: enum
type: ChangeType
bus_id: int

Route

id: int
number: int
name: string
listStopIds: list<int>

getStopIdbyIndex(int stop_index): int
addStopIdtoRoute(int new_stop_id)

Stop

id:int
name: string
location: Location
numPassengersWaiting: int
numNewPassengersArrivals: int
numPassengerDepartures: int
ridersArriveHigh: int
ridersArriveLow: int
ridersOffHigh: int
ridersOffLow: int
ridersOnHigh: int
ridersOnLow: int
ridersDepartHigh: int
ridersDepartLow: int
random: Random

getLocation( ): double[ ]
setStopDistributions(int ridersArriveHigh,
        int ridersArriveLow,
        int ridersOffHigh,
        int ridersOffLow,
        int ridersOnHigh,
        int ridersOnLow,
        int ridersDepartHigh,
        int ridersDepartLow)
newPassengersArrive( ): int
unloadPassengersfromBus( ): int
loadPassengersfromStop( ): int
passengersDepartStop( ): int

Efficiency

consts_speed: double
consts_waiting: double
consts_capacity: double
consts_buses: double
consts_combined: double

waiting_passengers(): int
bus_cost(): double
system_efficiency(): double

UserInterface

main_simulation_frame: JFrame
world_layout: JPanel
button_layout: JPanel
status_layout: JPanel
stop_boxes: Map<Integer, StopBox>
APP_WIDTH: int
APP_HEIGHT: int
system_efficiency_text: JTextField
simulation: Simulation
k_speed_text: JTextField
k_speed_button: Jbutton
k_capacity_text: JTextField
k_capacity_button: Jbutton
k_waiting_text: JTextField
k_waiting_button: Jbutton
k_combined_text: JTextField
k_combined_button: Jbutton
efficiency_button: Jbutton
currEventTime_text: JtextField

add_bus_GUI(int bus_index)
add_stop_GUI(int stop_index)
move_bus( )
move_bus_rewind( )
updateSystemEfficiency(String text)
setupListenersForButtons()

Simulation

buses: map<Integer, Bus>
stops: map <Integer, Stop>
routes: map <Integer, Route>
bus_changes: ArrayList<BusChange>
efficiency: Efficiency
current_bus_processing: int
next_stop_id: int
next_time: int
next_passengers: int
next_distance: double
event_index: int
queue: Queue
routeIDs: List
ui: UserInterface

setup(String[] args)
evaluateChanges(int bus_id)
createRewindEvent(int eventIndex)
updateBusStop(Bus bus)
execute_next()
rewind( )
passengerExchange(Bus bus, Stop stop)
addBusSpeedChange(int busId, int newSpeed)
addBusCapacityChange(int busId, int newCapacity)
addBusRouteChange(int busId, int rteId, int rteIndex)

BusChangeUI

busId: int

actionPerformed( )

StopBox

stop_icon: ImageIcon
bus_icon: ImageIcon
c: GridBagConstraints
bus_stop_info: JTextField
bus_stop_img: JLabel
bus_img: JLabel
pax_info: JTextField
busTextFields: Map<Integer,
JTextField>
name: string
stopID: int

add_busTextField(int bus_id, String s)
updateBusInfo(int bus_id)
show_buses( )
set_pax_info(String s)
init_bus( )
resize_box( )

Main

main(String[] args)

Creates
Moves
Rewinds
Changes
Travels Along
Stores
Travels to
Stores
Stores
Represents
Calculates
Displays
Controls
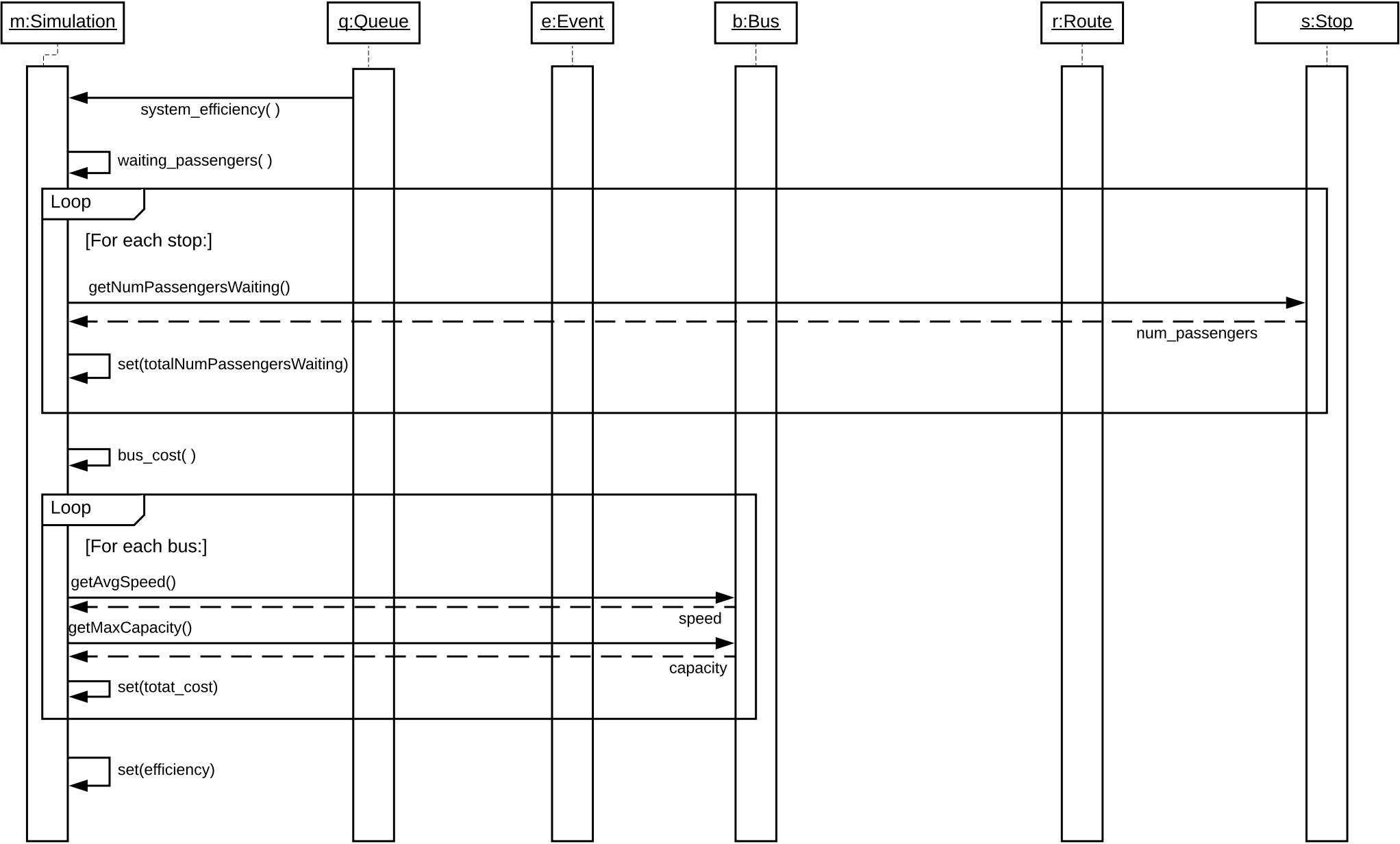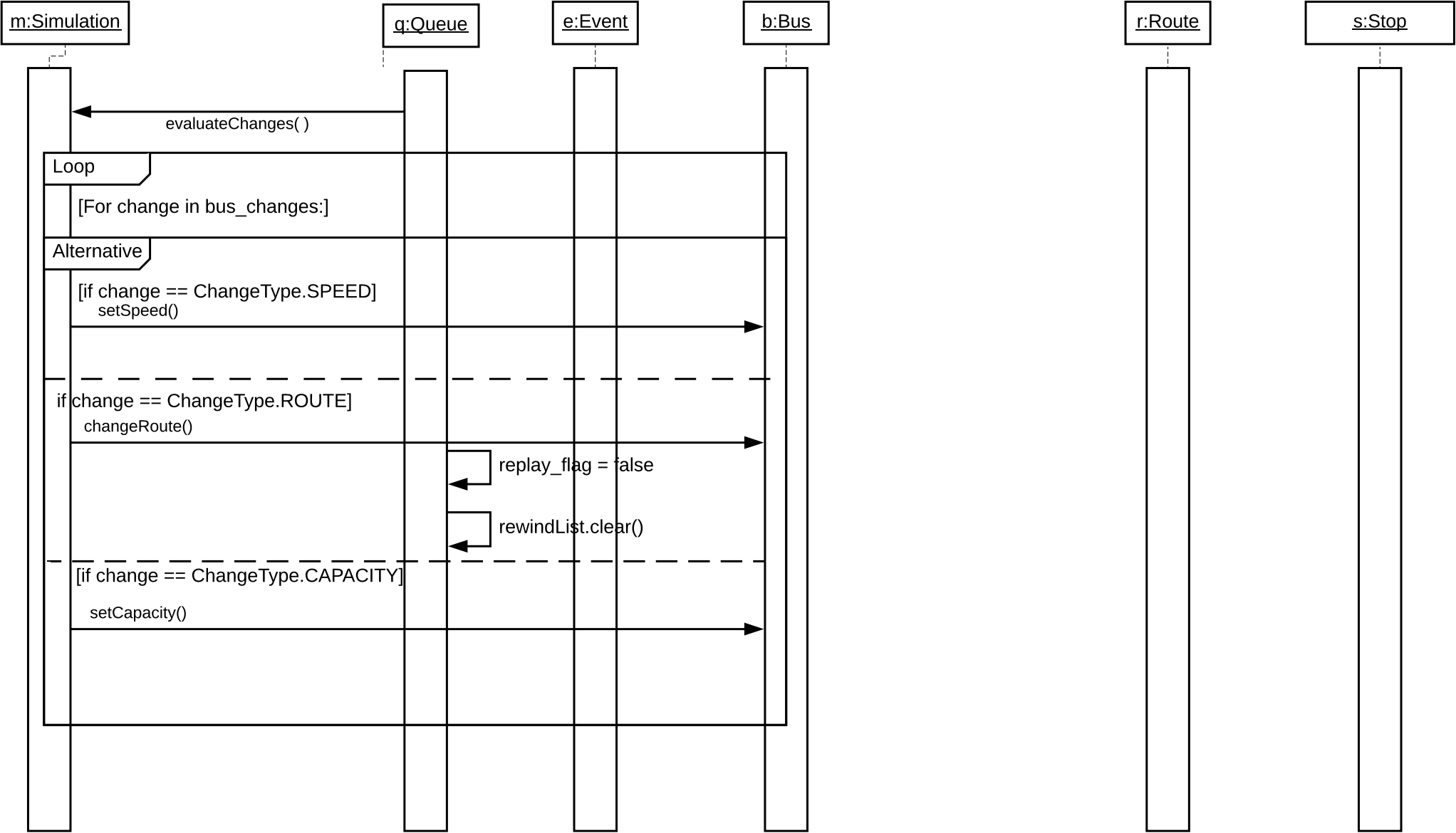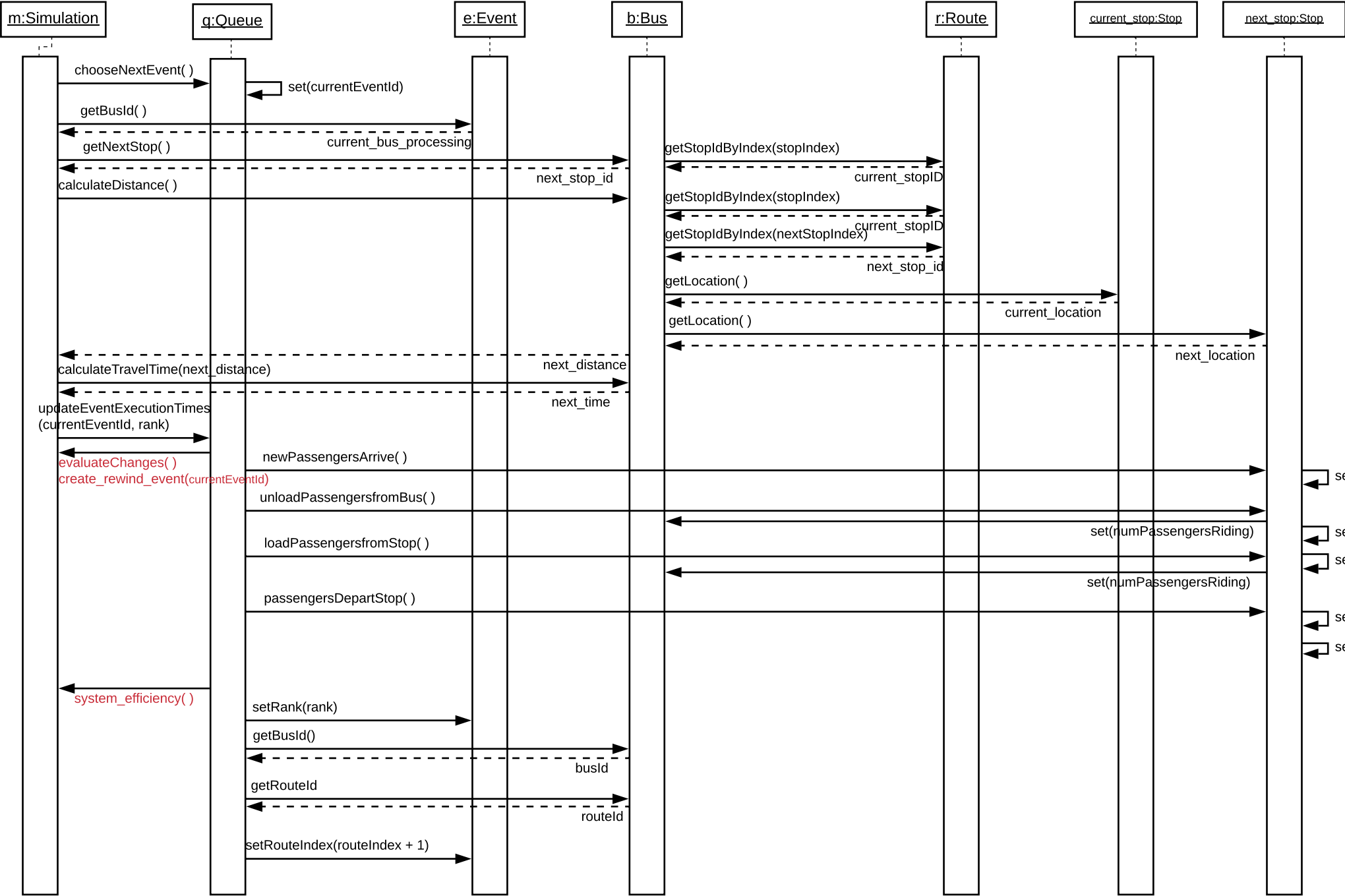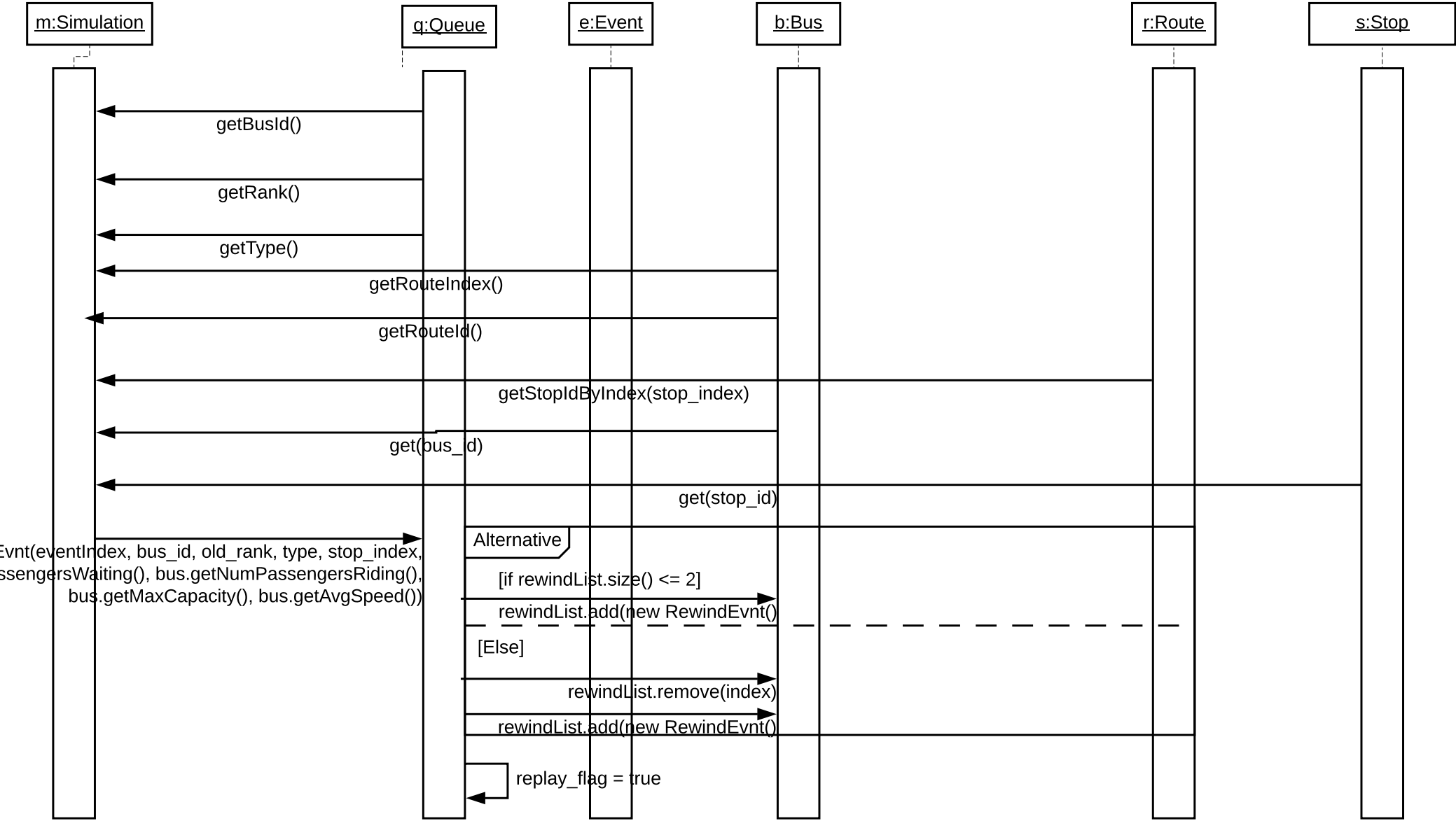Contains
Represents
Starts
Identifies

Kristen Beaufait, Navjot Kamal, Yifan Guo, Tyler Comte, Terrence Bunkley - Assignment #9 - Mass Transit Simulation System UML Sequence Diagram - System Efficiency   **(UML Version 1.4)**
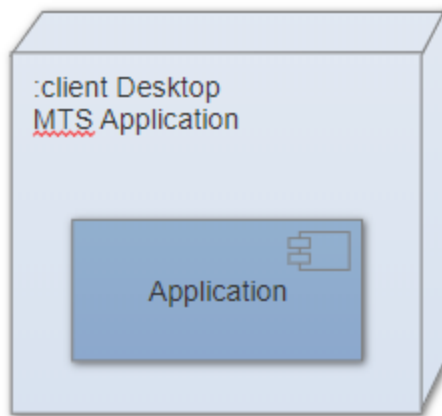
m:Simulation	q:Queue	e:Event	b:Bus	r:Route	s:Stop

system_efficiency( )

waiting_passengers( )

Loop

[For each stop:]

getNumPassengersWaiting()

num_passengers

set(totalNumPassengersWaiting)

bus_cost( )

Loop

[For each bus:]

getAvgSpeed()

speed

getMaxCapacity()

capacity

set(totat_cost)

set(efficiency)

Kristen Beaufait, Navjot Kamal, Yifan Guo, Tyler Comte, Terrence Bunkley - Assignment #9 - Mass Transit Simulation System UML Sequence Diagram - evaluate changes   (UML Version 1.4)

m:Simulation	q:Queue	e:Event	b:Bus	r:Route	s:Stop

evaluateChanges( )

Loop

[For change in bus_changes:]

Alternative

[if change == ChangeType.SPEED]
setSpeed()

if change == ChangeType.ROUTE]
changeRoute()

replay_flag = false

rewindList.clear()

[if change == ChangeType.CAPACITY]
setCapacity()

**Kristen Beaufait, Navjot Kamal, Yifan Guo, Tyler Comte, Terrence Bunkley - Assignment #9 - Mass Transit
Simulation System UML Sequence Diagram - Move Next Bus   (UML Version 1.4)**

Text in red is broken down in a separate sequence diagram

m:Simulation	q:Queue	e:Event	b:Bus	r:Route	s:Stop

getBusId()

getRank()

getType()

getRouteIndex()

getRouteId()

getStopIdByIndex(stop_index)

get(bus_id)

get(stop_id)

Evnt(eventIndex, bus_id, old_rank, type, stop_index, ssengersWaiting(), bus.getNumPassengersRiding(), bus.getMaxCapacity(), bus.getAvgSpeed())

Alternative

[if rewindList.size() <= 2]

rewindList.add(new RewindEvnt()

[Else]

rewindList.remove(index)

rewindList.add(new RewindEvnt()

replay_flag = true

UML Deployment Diagram:
MTS by Kristen Beaufait, Navjot Kamal, Yifan Guo, Tyler Comte, Terrence Bunkley

:client Desktop
MTS Application

Application

It's a jar file that runs within a VM environment, and there is no server, DB, or any other sources. It runs the application and executes the UI components according to the user input.

User

Move Next Bus

Rewind

Update Speed

Update Capacity

Update Waiting

Update Buses

Update Combined

Refresh Efficiency

Change Bus

Speed

Capacity

Route