**Host Agent**

Data Messages

int ntables= 4; IWantFood(CustomerAgent cust) {

List<CustomerAgent> waitingCustomers; waitingCustomer.add(cust);

Collection<Table> tables; }

List<MyWaiter> waiters;

String name; TableFree(int t) {

Semaphore atTable; if Ǝ a table in tables ϶

CookAgent cook; table.tableNumber = t

table.setUnoccupied();

}

class MyWaiter { WantToGoOnBreak(WaiterAgent waiter) {

WaiterAgent waiter; if Ǝ myWaiter in waiters ϶

WaiterState state; myWaiter.waiter = waiter

WaiterEvent event; myWaiter.event = requestingBreak;

} }

class State { GoingOnBreak(WaiterAgent waiter) {

CustomerAgent occupiedBy; if Ǝ myWaiter in waiters ϶

int xCoord; myWaiter.waiter = waiter

int yCoord; myWaiter.state = onBreak;

} }

GoingOffBreak(WaiterAgent waiter) {

if Ǝ myWaiter in waiters ϶

myWaiter.waiter = waiter

myWaiter.event = breakOver;

}

Scheduler

if Ǝ t in tables ϶ t.occupied = false

and waitingCustomesr.empty = false

and Ǝ w in waiters ϶ w.state = available

seatCustomer(waitingCustomers[0], t, w)

if Ǝ myWaiter in waiters ϶ myWaiter.state =

Available and myWaiter.event = RequestingBreak

replyToBreakRequest(myWaiter)

if Ǝ myWaiter in waiters ϶ myWaiter.state =

OnBreak and myWaiter.event = BreakOver

waiterBreakIsOver(myWaiter)

Actions

seatCustomer(CustomerAgent customer, Table table, WaiterAgent waiter) {

customer.setWaiter(waiter);

table.setOccupant(customer);

mwaiter.PleaseSeatCustomer(this, customer, table.tableNumber);

waitingCustomer.remove(customer);

}

replyToBreakRequest(MyWaiter myWaiter) {

boolean waiterAvailable = false;

if Ǝ waiter in waiters ϶ waiter != myWaiter

if waiter.state = Available

waiterAvailable = true

if (waiters.size > 1 and waiterAvailable)

myWaiter.waiter.BreakOkay();

myWaiter.state = WaitingForBreak;

myWaiter.event = none;

else

myWaiter.waiter.BreakNotOkay();

myWaiter.state = Available;

myWaiter.event = none;

}

waiterBreakIsOver(MyWaiter myWaiter) {

myWaiter.state = Available;

myWaiter.event = none;

myWaiter.waiter.BreakOver();

}