Data

enum CustomerState {waiting, seated, readyToOrder, asked, ordered, orderGiven, done};

class MyCustomer {

CustomerAgent c;

int t;

string c;

CustomerState s;}

List<MyCustomer> customers;

List<WaiterOrder> readyOrders;

WaiterGui waiterGui;

HostAgent host;

Class WaiterOrder {

String choice;

int table;

}

Scheduler

if there exists c in customers such that c.s == ordered then GiveOrderToCook(c);

if there exists c in customers such that c.s == waiting then seatCustomer(c);

if there exists c in customers such that c.s == readyToOrder then TakeOrder(c);

if there exists o in readyOrders then TakeFoodToCustomer();

Messages

msgSitAtTable(CustomerAgent c, int table) {

customers.add(new MyCustomer(c,table, waiting));

}

msgImReadyToOrder(CustomerAgent c) {

MyCustomer mc = customers.find(c);

mc.s = readyToOrder;

}

msgHereIsMyChoice(String choice, CustomerAgent c) {

MyCustomer mc = customers.find(c);

mc.choice = choice;

mc.state = ordered;

}

msgOrderIsReady(String choice, int table) {

readyOrders.add(new WaiterOrder(choice, table);

}

msgDoneEatingAndLeaving(CustomerAgent c) {

MyCustomer mc = customers.find(c);

host.msgTableIsFree(mc.table);

customers.remove(mc);

}Actions

\*\* Included animations in design since they are closely tied with the semaphores that dictate waiter’s processes \*\*

seatCustomer(MyCustomer c) {

c.c.FollowMe(new Menu());

DoSeatCustomer(c); // animation

c.s = seated;

DoLeaveCustomer(); //animation

}

TakeOrder(MyCustomer c) {

DoGoToTable(c.t); // animation

c.c.WhatWouldYouLike();

c.s = asked;

}

GiveOrderToCook(MyCustomer c) {

c.s = orderGiven;

DoGoToCook() // animation

cook.msgHereIsAnOrder (this, c.choice, c.t);

c.s = orderGiven;

}

TakeFoodToCustomer(MyCustomer c)

{

WaiterGui.GiveFoodToCustomer(c); // animation

if there exists c in customers such that c.s != done and readyOrders.get(0).table = c.table then c.HereIsYourFood();

ProcureFood(); // animation

DoGoToTable(); // animation

serveFood() // animation

c.msgHereIsYourFood

readyOrders.remove(0);

DoLeaveCustomer // animation

}

DoSeatCustomer(CustomerAgent customer, int tableNumber)

{

WaiterGui.DoBringToTable(c.getGui(), tableNumber);

}