BDSA 2017 Assignment 03

Martin Winther CÃęcilie Vildenfeldt Iversen

September 15, 2017

1 Exercise 1

1.1

Can the system under consideration be represented as an actor? Justify your answer

Answer: No, the system under considereation cannot be represented as an actor when performing requirements elicitation.
An actor is an external entity that interacts with the system and therefore the system itself cannot be considered an actor. The questions for determining an actor are:

- Which users are supported by the system to perform work?
- Which users execute the main functions?
- Which users perform secondary functions (maintenance, operation, etc.)?
- With what external system (hw/sw) will the system interact?

None of these questions conceivably cover the system itself.

1.2

What is the difference between a scenario and a use case? When do you use each construct?

Answer: Definitions:

- **A scenario**
  - is an instance of a use case describing a concrete set of actions.
  - is a narrative description of what people do and experience as they try to make use of computer systems and applications.
  - is used as examples for illustrating common cases; their focus is on understandability.
- **A use case**
  - is an abstraction that describes all possible scenarios involving the described functionality.
  - represents a complete flow of events through the system in the sense that it describes a series of related interactions that result from its initiation.
  - describes the behavior of the system as seen from an actorâĂŹs point of view.
  - is used to describe all possible cases; their focus is on completeness.

The difference between a scenario and use case is one of perspective.
A scenario is a very narrow description of a set of interactions with the system, while the use case describes the flow of events initiated by an actor in detail.
A scenario is used for communicating with the client, they are understandable and do not use much technical language.
A use case is used for development and is based upon the requirements set forth by the scenario. It is created as a collaborative effort by the client and developer.

1.3

What is the difference between a state machine diagram and an activity diagram?

Answer:

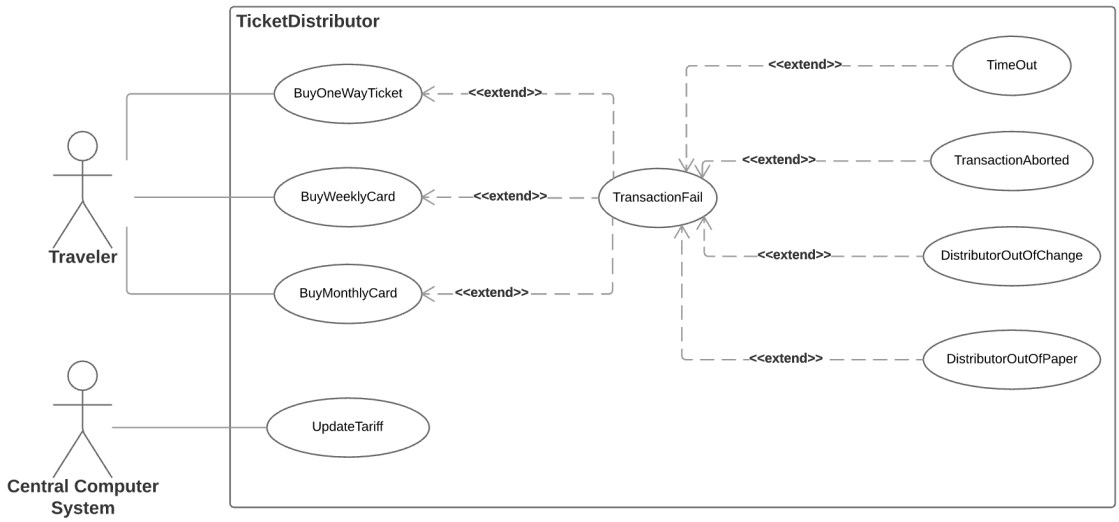
- **A state machine diagram**
  - is a notation for describing the sequence of states an object goes through in response to external events.
  - describes the dynamic behavior of an individual object as a number of states and transitions between these states.
  - focuses on the transitions between states as a result of external events for an individual object.
- **An activity diagram**
  - describes the behavior of a system in terms of activities.
    - Activities are modeling elements that represent the execution of a set of operations.
  - can be used to represent control flow and data flow.
  - model the dynamic behaviour of a system - show decisions.

The difference between a state machine diagram and an activity diagram is the level of abstraction. While an activity diagram deals with lower level behaviour and decisions, a state machine diagram deals with the different states the system is in during operation.

2 Exercise 2

Draw a use case diagram for a ticket distributor for a train system.
The system includes two actors:
a traveler, who purchases different types of tickets, and a central computer system, which maintains a reference database for the tariff.
Use cases should include: BuyOneWayTicket, BuyWeeklyCard, BuyMonthlyCard, UpdateTariff.
Also include the following exceptional cases: Time-Out (i.e., traveler took too long to insert the right amount), TransactionAborted (i.e., traveler selected the cancel button without completing the transaction), DistributorOutOfChange, and DistributorOutOfPaper.

Answer:

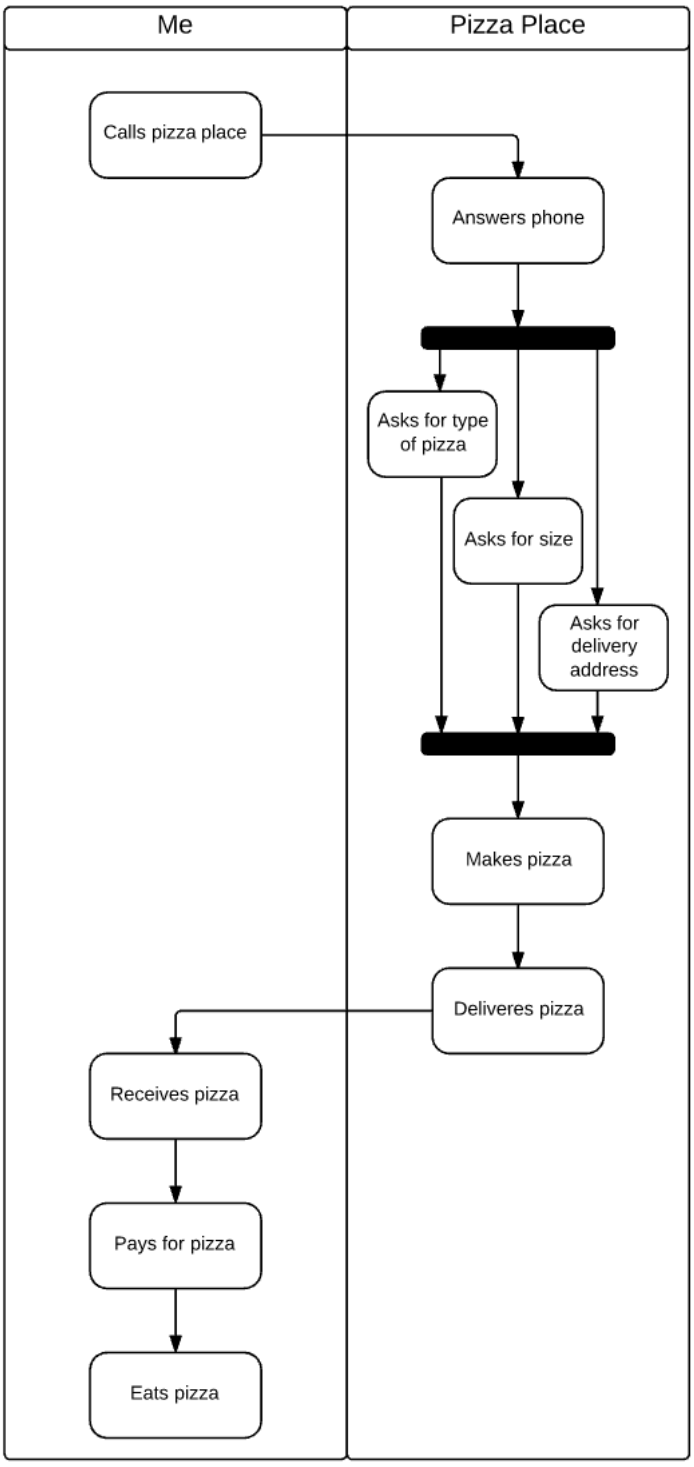


3 Exercise 3

3.1

Consider the process of ordering a pizza over the phone. Draw an activity diagram representing each step of the process, from the moment you pick up the phone to the point where you start eating the pizza. Do not represent any exceptions. Include activities that others need to perform.

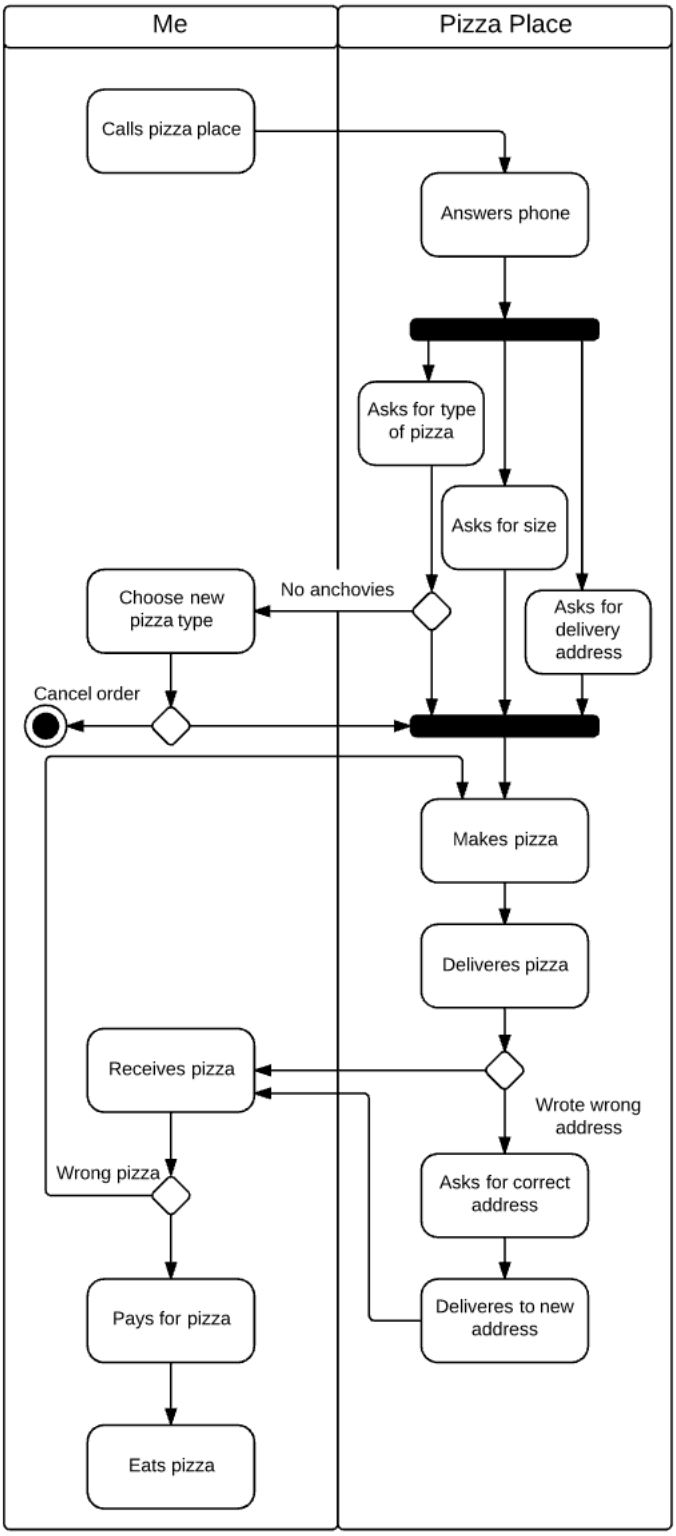
Answer:



3.2

Add exception handling to the activity diagram you developed in the previous exercise. Consider at least three exceptions. You can for instance use the following: delivery person wrote down wrong address, deliver person brings wrong pizza, store out of anchovies.

Answer:



4 C#

Github link: https://github.itu.dk/marw/BDSA2017.Assignment03