Requirements Elicitation with Use Cases

Jörg Kienzle & Shane Sendall
School of Computer Science
McGill University, Montreal, QC, Canada

Problems with Version 1

- What is the meaning of “lift” and how does it relate to the system?
- It could state better who has to do what, e.g., "user requests a floor" would be better written: “User requests System…”
  - Remember: each base interaction has to be an *input* or *output* interaction between the System and an actor
- It fails to state whether the extensions join the main scenario, succeed, or fail.
- It fails to state sufficient detail on system responsibilities, at least to give to developers! This may be remedied by making some steps sub-”use cases”.
- Secondary actors do not appear (functionality might get lost).
- It fails to make extension conditions verifiable, e.g., steps (2-3)IIa, 3a, and 6a

Problems with Version 2

- Use case fails to get the system boundary right (door closing functionality should not be part of system according to project description).
- It has the wrong kind of system detail (algorithmic), e.g. FIFO, floor lights, etc.
- It is long, hard to read, and unnecessarily complex. Each step should describe a single input or output interaction.

1. Auction System Actors

- Primary
  - User (or Customer)
    - Can be split into Buyer / Seller
  - (System Administrator)
- Secondary
  - Credit Card Company
  - (Email Application)
  - (Postal Service)

1. Stakeholder Interest

- User expects
  - Fairness
  - Transparent bidding process
  - Prompt notification
  - Highest bidder wins
  - Reliable service
  - No money lost
  - High availability
  - Privacy
- ...

2. Summery-Level Use Case (1)

Use Case: Buy and Sell Goods by Auction

Scope: Auction System

Level: Summary

Intention in Context: The intention of the User is to buy and sell goods by auctions over time.

Multiplicity: Multiple users can interact with the auction system concurrently. A User can be involved in multiple auctions at any one time.

Primary Actor: User (becomes Customer, once s/he has identified him/herself with the System)

2. Summery-Level Use Case (2)

Main Success Scenario:

All Users must first enrol with the System before they have the right to use the system

1. *User* enrols with *System*, providing System with registration information.

Steps 2-5 can be repeated many times.

2. *User* identifies him/herself to *System*.
3. *System* presents *Customer* with a welcome message.

The user-goal level use cases of step 4 can be performed in parallel and individually repeated. A customer may bid and sell in many auctions at any one time.

4. *Customer* increases credit with *System*

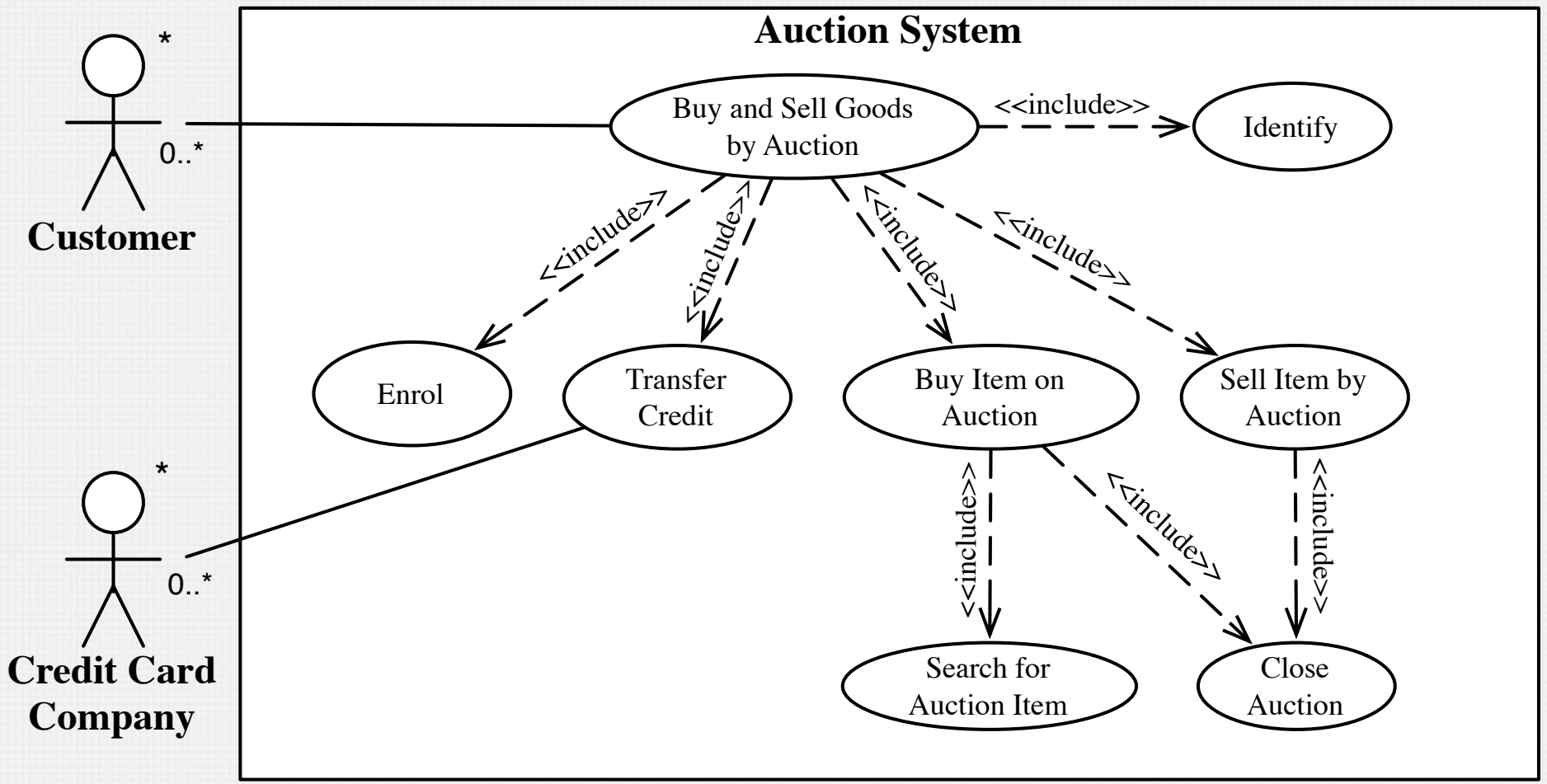
   or *Customer* buys an item on auction

   or *Customer* sells an item by auction
5. *Customer* exits *System*.
6. *Customer* requests to cancel his/her enrollment.

Extensions:

3a. *System* fails to identify *User*; use case continues at step 2.

3. Auction System Use Case Diagram



4. Buy Item Use Case (1)

Use Case: Buy Item on Auction

Scope: Auction System

Level: User Goal

Intention in Context: The intention of the *Customer* is to follow the auction, which may then evolve into an intention to buy an item by auction, i.e., he/she may then choose to bid for an item.

Multiplicity: Several *Customers* can place bid simultaneously. A given *Customer* may bid in many different auctions at any one time.

Primary Actor: Customer

Precondition: The Customer has already identified her / himself to the System

4. Buy Item Use Case (2)

Main Success Scenario:

Customer may leave the auction and come back again later to look at the progress of the auction, without effect on the auction; in this case, the Customer is required to join the auction again.

1. *Customer* searches for an item under auction.
2. *Customer* requests *System* to join the auction of the item.
3. *System* presents a view of the auction to *Customer*.

Steps 4-5 can be repeated according to the intentions and bidding policy of the Customer.

4. *Customer* makes a bid on the item to *System*.
5. *System* validates the bid.
6. *System* closes the auction with a winning bid by *Customer*.

4. Buy Item Use Case (3)

Extensions:

2a. *Customer* requests *System* not to pursue item further; use case ends in failure.

3a. *System* informs *Customer* that auction has not started: use case ends in failure.

3b. *System* informs *Customer* that auction is closed: use case ends in failure.

5a. *System* determines that bid does not meet the minimum increment.

5a.1. *System* informs *Customer*; use cases continues at step 4.

5b. *System* determines that *Customer* does not have sufficient credit to guarantee bid:

5b.1. *System* informs *Customer*; use cases continues at step 4.

6a. *Customer* was not the highest bidder:

6a.1. *System* closes the auction; use case ends in failure.

6b. Bid did not meet reserve price.

6b.1. *System* closes the auction; use case ends in failure.

Buy Drink Use Case (1)

Use Case: Buy Drink

Scope: Vending Machine

Level: User Goal

Intention in Context: The intention of the *Customer* is to buy a drink in exchange of money.

Multiplicity: There can always be only one *Customer* interacting with the system at a given time.

Primary Actor: Customer

Secondary Actors: Selector Button, Coin Slot, Shelf, Sensor, Money Box, Drink Light, Cancel Button, Display, Terminal

Precondition: The system is in service, filled with drinks and change, and the Money Box is not full.

Buy Drink Use Case (2)

Main Success Scenario:

Customer selects drink by pushing appropriate drink selector button.

1. *Button* notifies *System* of selected drink.
2. *System* displays the price of the selected drink on *Display*.

Customer inserts a coin into Coin Slot.

3. *Coin Slot* notifies *System*.
4. *System* recognizes the coin, and updates the remaining price on *Display*.

Steps 3 and 4 are repeated until the amount of inserted money reaches or exceeds the price of the drink.

5. *System* validates that there are sufficient funds for the selection and notifies *Shelf* to start dispensing the drink.
6. *Sensor* informs *System* that the drink has been dispensed.
7. *System* asks *Money Box* to collect the specified amount of money and, if necessary, provide the change.

Customer collects the drink and optionally the change.

Buy Drink Use Case (3)

Extensions:

2a. *System* ascertains that the selected drink is not available and flashes *Drink Lights*; use case ends in failure.

4a. *System* fails to identify the coin; *System* asks *Money Box* to eject coin; use case continues at step 3.

(3-4)a. *Customer* informs *System* to abort the sale by hitting the *Cancel* button;

(3-4)a.1 *System* asks *Money Box* to eject coins; use case ends in success.

(3-4)b. *System* times out.

(3-4)b.1 *System* asks *Money Box* to eject the inserted coins; use case ends in failure.

5a. *System* ascertains that the inserted money exceeds the price for the drink and that there is not enough change;

5a.1 *System* asks *Money Box* to eject inserted coins.

5a.2 *System* displays “no change” on *Display*; use case ends in failure.

7a II. The *Money Box* is full.

7a II.1 *System* displays “no service” on *Display* and goes out of service; use case ends successfully.

7b II. The delivered drink was the last one of that kind.

7b II.1 *System* turns on the appropriate *Drink Light*; use case ends successfully.

Service Machine Use Case (1)

Use Case: Service Machine

Scope: Vending Machine

Level: User Goal

Intention in Context: The intention of the Service Person is to maintain the machine by ensuring that there are drinks available, modifying drink pricing, and by collecting the money earned.

Multiplicity: There can be only one service person servicing the machine at a given time.

Primary Actor: Service Person

Precondition: No Customer is currently using the system.

Buy Drink Use Case (2)

Main Success Scenario:

Service Person interacts with the system by using the Terminal.

1. *Service Person* identifies himself with *System*.

Steps 2-3 can be repeated for each shelf, in any order.

2. *Service Person* informs *System* of new price for a shelf.

3. *Service Person* replenishes a shelf and informs *System* of new number of drinks for that shelf.

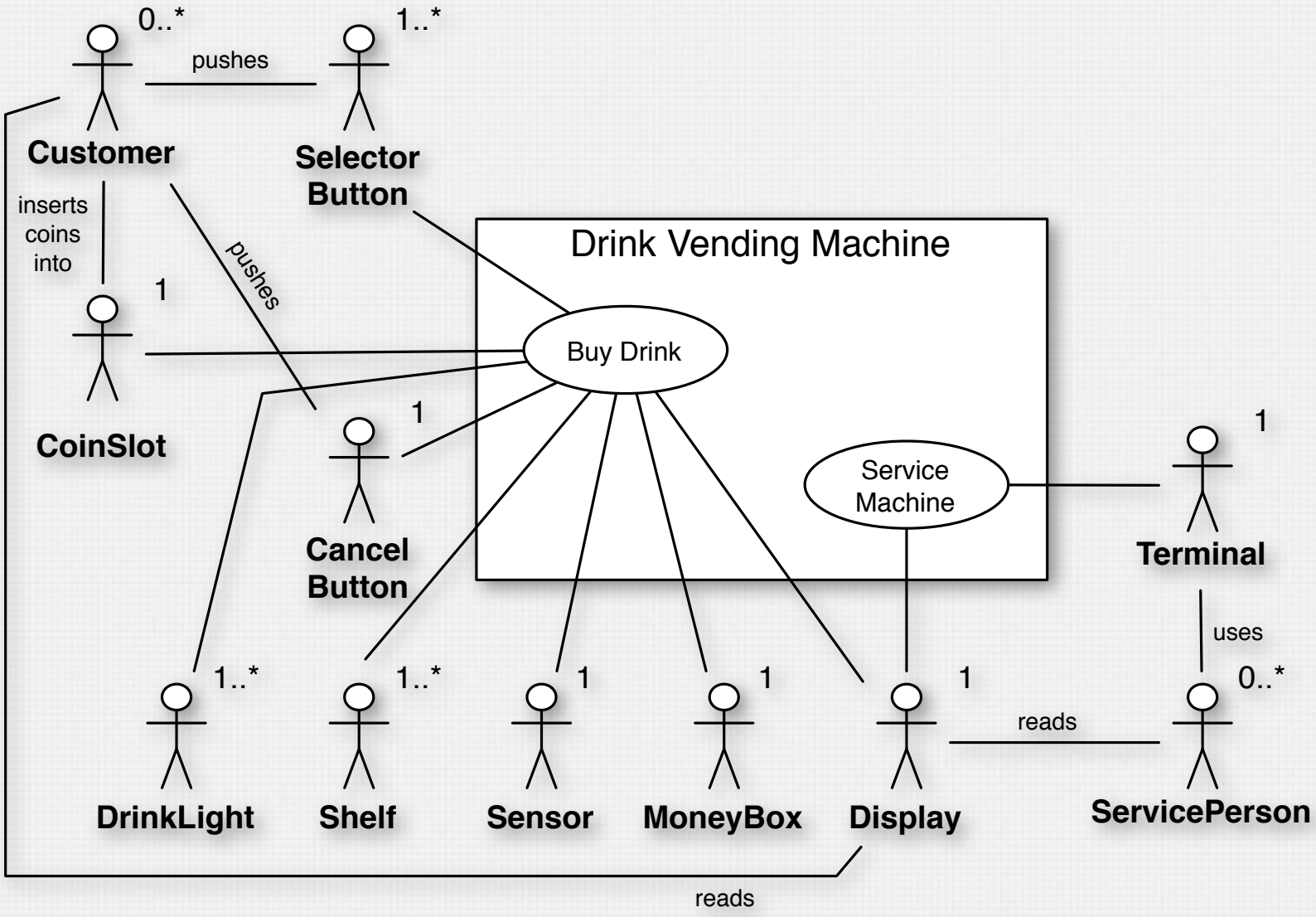
4. *Service Person* empties the *Money Box*, replenishes the change and informs the *System*.

5. *Service Person* informs *System* that maintenance is over.

Extensions:

2a. *System* fails to identify the *Service Person*; use case ends in failure.

Vending Machine Use Case Diagram



Drink Vending Machine Questions

1. Create a URN model for the *BuyDrink* use case.
   - If you decide to group several basic interaction steps into one URN responsibility, please use the description field of the responsibility to document which use case steps it represents
2. Create a URN model for the *ServiceMachine* use case