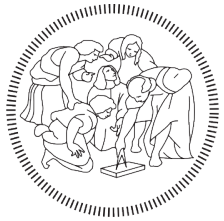


POLITECNICO
MILANO 1863

Software Engineering 2

Exercises from WE2 exams

M Camilli, E Di Nitto, M Rossi

AdmissionManager

POLITECNICO
MILANO 1863

- Your company has been tasked to develop a system that handles the admission applications that parents send, on behalf of their children, to the high schools of a metropolitan area.
- Parents can send admission applications to multiple schools.
- Before sending an application, they must register their child in the system; the registration includes login credentials (username and password), the personal data of the child (first name, last name, birth date, etc.), the name of at least one parent, contact information (which must include an email address and a phone number), the name of the last school they attended, and the list of grades (which includes the obtained score, from 1 to 10, for each subject).
- Each application is assigned an identifier by the system, to allow parents to check its status after sending it (which can be “accepted”, “rejected”, or “not evaluated”).
- Parents can withdraw applications previously sent.
- They can also ask the system to be notified by email when the outcome of the evaluation of an application is available.

AdmissionManager (2)

POLITECNICO MILANO 1863

- School administrators use the system to check the applications sent to their schools and to approve/reject them.
- In particular, they can retrieve the list of applications sent to their schools that have yet to be evaluated;
- they can also leave comments on the applications, and they can decide to accept or reject the applications.
- Administrators can also set a preference to receive a notification, in the form of an email, when a new application is sent to their school.

First question

- Define the goals for the AdmissionManager system.

Discussion

POLITECNICO
MILANO 1863

- Are the following ones well-formed goals?
  - G1: User sends an application
  - G2: User withdraws an application
  - G3: School administrator evaluates an application
  - G4: User is notified about an application evaluation

Our goals formulation

- G1: Parents can manage (send and then monitor) applications to schools on behalf of their children.
- G2: School administrators can manage (check and approve/reject) applications sent to their schools.

Second question

- Select one of the goals defined in the previous point and define in natural language suitable domain assumptions and requirements to guarantee that the AdmissionManager system fulfills the selected goal.

Discussion

POLITECNICO
MILANO 1863

- Can the following sentence be considered a domain assumption?
  - "As soon as an application arrives to the system, a status needs to be assigned to it"

Discussion (cont.)

POLITECNICO
MILANO 1863

- Are the following requirements well formulated?
  - Parents must be registered into the system to issue an application
  - The system must allow parents to register by providing their email address and personal information

Our requirements and domain assumptions – G1

- G1: Parents can manage (send and then monitor) applications to schools on behalf of their children.
  - R1.1: AdmissionManager allows system administrators to open application windows for their schools
  - R1.2: AdmissionManager allows parents to register into the system and provide contact information and information about their child (name of the child, name of last attended school, etc.)
  - R1.3: AdmissionManager allows parents to log into the system using the credentials input at registration time
  - R1.4: AdmissionManager allows parents to indicate in their profile that they want to be notified when the outcome of an application is available

Our requirements and domain assumptions – G1 (cont.)

- R1.5: AdmissionManager allows parents to send an application to a school
- R1.6: AdmissionManager assigns a unique identifier to each application received
- R1.7: AdmissionManager allows parents to see the list of applications sent
- R1.8: AdmissionManager allows parents to withdraw an application previously sent
- A1.1: Parents provide correct information (in particular, contact information) when registering

Our requirements and domain assumptions – G2

- G2: School administrators can manage (check and approve/reject) applications sent to their schools.
  - R2.1: AdmissionManager allows system administrators to insert new administrators in the system and associate them with the corresponding school
  - R2.2: AdmissionManager allows school administrators to log into the system using the credentials assigned to them by system administrators
  - R2.3: AdmissionManager allows school administrators to indicate in their profile that they want to be notified when new applications for their schools are received

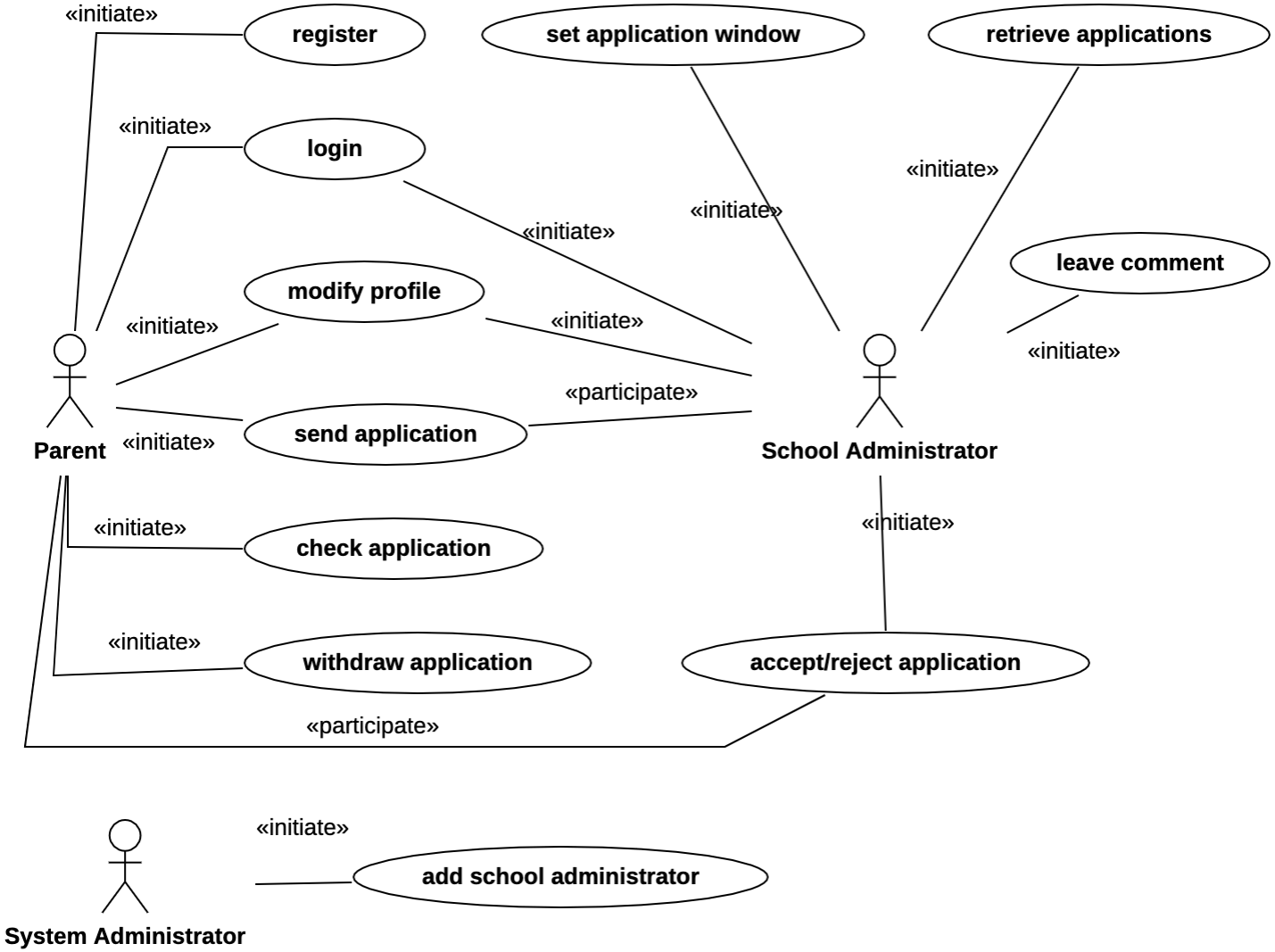
Our requirements and domain assumptions – G2 (cont.)

- R2.4: AdmissionManager allows school administrators to retrieve applications (related to their schools) that have yet to be evaluated
- R2.5: AdmissionManager allows school administrators to select an application yet to be evaluated and leave a comment in it
- R2.6: AdmissionManager allows school administrators to accept/reject an application
- A2.1: School administrators periodically evaluate applications and guarantee to explicitly accept/reject all applications arrived within the notification window.

Third question – part 1

- Draw a UML Use Case Diagram describing the main use cases of the AdmissionManager system.

Use Case Diagram



Third question – part 2

- Pick one of the use cases, and define it (name, pre-condition, event flow, etc.).

A use case description

Name	Send Application
Actor	Parent, School Administrator
Entry condition	Parent has registered child and is logged in with the corresponding account. He/she has all necessary information
Event Flow	1. Parent selects school to which application must be sent 2. If required by the school, parent fills out additional information concerning child 3. Parent clicks “submit” button 4. System checks application and responds with application number 5. If administrator of selected school has asked to receive a notification of the application, email is sent to school administrator
Exit Condition	Application is received by the system, and email is sent to school administrator if he/she asked to be notified
Exceptions	Data provided in application is invalid or missing, user is notified that it should be fixed. School does no longer accept application (the application window has expired), so the application is immediately rejected

PaasPopCoin

POLITECNICO
MILANO 1863

- The private security and event organisation company “HSG” from The Netherlands wants to build an application---PaasPopCoin---that handles the coin emission and transactions in the scope of a medium-size music festival they are organizing. The goal of the system is to allow festival-goers and operators to spend an allotted amount of money in relative safety and without the need to bring wallets and other assets around the event. The software in question needs to handle at least three scenarios:
    1. Emission of coins in exchange of money through appropriate cashier desks and Automated Teller Machines (ATMs).
    2. Cash-back, that is, exchange of coins with cash in the same locations (we assume that people at the festival may be willing to receive back the money corresponding to the coins they have not used).
    3. Tracking of coin expenditure transactions at the various festival shops.
- In the scope of the above scenarios, there are several special conditions to be considered.

PaasPopCoin (2)

POLITECNICO
MILANO 1863

- First, in the scope of coin emissions, there exist four classes of coin “buyers”, namely: (a) VIPs (e.g., event artists, shop-owners) who receive a 30% discount on the coins they buy; (b) event organisation people (e.g., security guards, event managers) who receive a 50% discount; (c) event ticket holders class A, who receive a 20% discount; (d) finally, regular ticket holders who receive no discount. When buying coins, users first need to authenticate themselves by inserting their own ID card in the ATM or by giving it to the cashier; this allows the system to determine the class to which each coin buyer belongs. After authentication, buyers get the coins upon inserting into the ATM or giving to the cashier the corresponding amount of money.
- Second, also in the context of cash-back, users need to authenticate with their ID card to make sure the appropriate amount of money is given back, considering their role and privileges.
- Third, during the event, every shop clerk keeps track through the PaasPopCoin system of the sales of products and the coins received.
- PaasPopCoin relies on a third-party analytics service to periodically check whether the festival is earning money or not (cost-benefit analysis). Such check is performed with respect to costs of products being sold during the event, as well as the overhead to cover all event organisation and management expenses. Note that reasoning and evaluating these aspects is beyond the scope of the exercise.

First Question

POLITECNICO
MILANO 1863

- With reference to the Jackson-Zave distinction between the world and the machine, identify the relevant world phenomena for PaasPopCoin, including the ones shared with the machine, providing a short description if necessary. For shared phenomena specify whether they are controlled by the world or the machine. Focus on phenomena that are relevant to describe the requirements of the system.

Possible world-only phenomena

POLITECNICO
MILANO 1863

- User buys Class A ticket
- User buys regular ticket
- VIP is contracted for event
- Event organization is started and contractors registered
- Event starts
- User gives money to cashier (to be converted in coins)
- User gives coins to cashier (to be converted in money)
- User gives ID card to cashier
- User buys some product at festival
- The external analytics service checks the success of an event

Possible shared phenomena

POLITECNICO
MILANO 1863

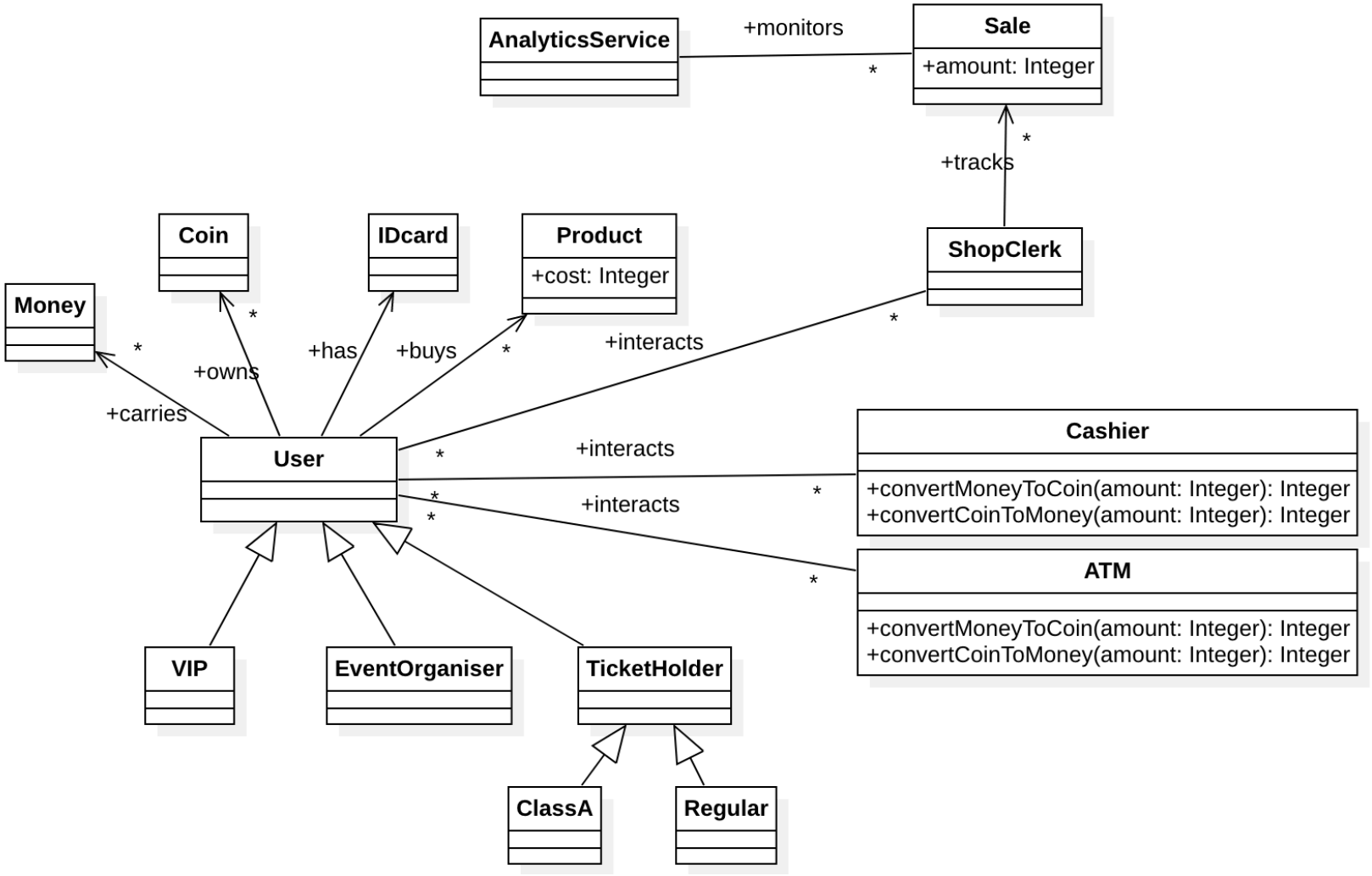
- World-controlled
  - User inserts money into an ATM machine
  - ID Card is inserted into ATM
  - User inserts coins into an ATM
  - Cashier inserts in the system an ID card number
  - Cashier inserts in the system the amount of money handed by a certain user
  - Cashier inserts in the system the amount of coins returned by a certain user
  - Store clerk inputs in system the amount of coins spent by user in shop
- Machine-controlled
  - The system enables coin emission after checking ID card and inserted amount of money
  - The system enables cash-back after checking ID card and inserted number of coins
  - The system sends data about purchases to the external analytics service

Second Question

POLITECNICO MILANO 1863

- Describe through a UML Class Diagram the main elements of the PaasPopCoin domain.

Domain model

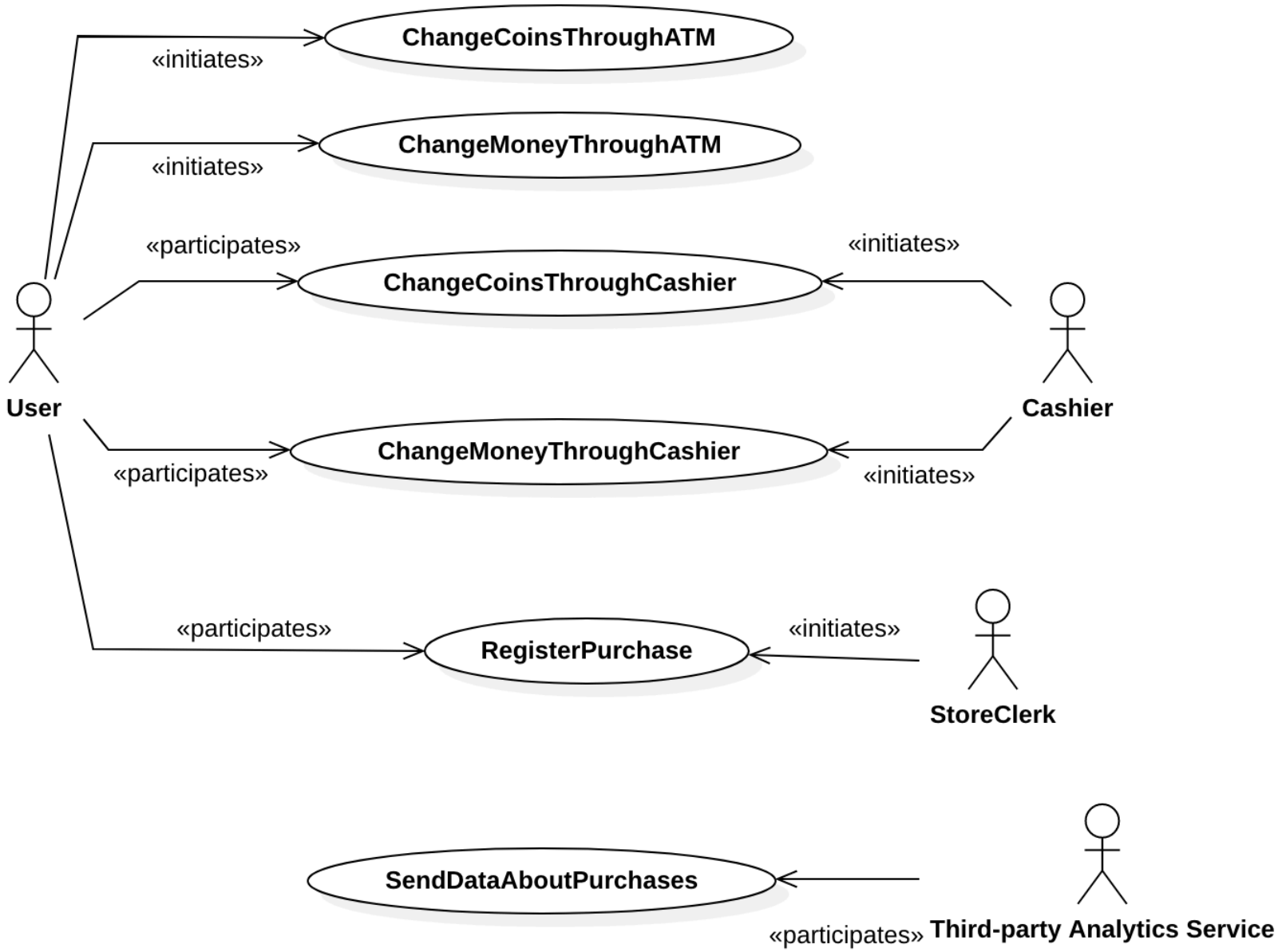


Third Question

POLITECNICO
MILANO 1863

- Define a UML Use Case Diagram describing the relevant actors and use cases for PaasPopCoin.
  - You can provide a brief explanation of the Use Case Diagram, especially if the names of the use cases are not self-explanatory.

Use Case Diagram



Milk vending machine - problem description

- A milk vending machine accepts 3 types of coins: 25¢ (cents), 50¢, and 1$. It accepts coins only in ascending order, for example, after inserting a 50¢ coin, it can only accept 50¢ or 1$ coins and it cannot accept 25¢ coins until the process is restarted by asking for the remaining change.
- Every time the amount of money in the machine reaches (or surpasses) 1$, it produces a bottle of milk, subtracts 1$ from the amount of money that is in the machine, and leaves the rest in the machine.

Milk vending machine - problem description

- At any time, the user can ask for the money still in the machine (and not used to buy a bottle of milk) to be returned (this can occur even if there is no money remaining in the machine); the effect of this is that the process is restarted, so smaller coins can be introduced again.
- The vending machine also accepts fidelity cards. When the user inserts a fidelity card, he/she can buy a bottle of milk for 75¢ instead than 1$. Thus, if the machine receives the fidelity card and the current amount of money is equal to or surpasses 75¢, it produces the bottle of milk, subtracts 75¢ from the amount of money that is in the machine, and leaves the rest in the machine.

The task

- Referring to the Jackson-Zave distinction between the world and the machine, identify world, machine and shared phenomena. For these last ones specify which part (world or machine) is controlling them

Phenomenon	Shared	Who controls it
User wants to buy some milk	N	W
User inserts a coin in the machine	Y	W
The machine compares the inserted coin with the last received one	N	M
The machine rejects the inserted coin	Y	M
The machine accepts the inserted coin	Y	M
User inserts a fidelity card	Y	W
The machine checks and accepts the fidelity card	Y	M
The machine sees that amount needed to buy a bottle of milk is reached	N	M
The machine delivers the bottle of milk	Y	M
The machine updates the current amount of money	Y	M
The user goes home with the milk	N	W
The user wants to receive the money back	N	W
The user asks for the money back	Y	W
The machine delivers the amount of money to the user	Y	M
The machine resets the money count	N	M
The operator sets the current number of bottles in the machine	Y	W
A milk sensor signals the milk in the machine is finishing	Y	W
The machine decreases the counter of the current number of bottles	N	M
The machine goes out of service	Y	M