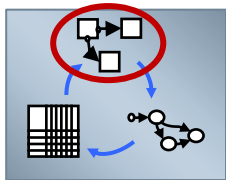
This is an exercise from the Merode course at KU Leuven. The exercise is used as an optional take-home assignment.

The solution is given as an EDG (part of a Merode model)

Some of the mistakes from student solutions are specific to Merode, others are universal

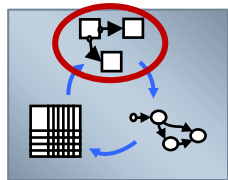
EDG

3. GAS STATION



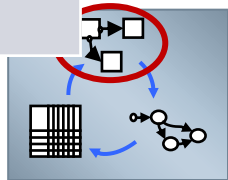
Gas Stations

- An oil company possesses a number of gas stations spread across the country. Customers can pay their refuel turn in cash at the counter or with a personal credit card, but they can also ask for a refuel card.. When using their refuel card, customers do not pay the refuel turn immediately. Rather, at the end of each month, the holder of the fuel card receives a monthly invoice with the amounts of all the refuel turns paid with the fuel card. On the monthly invoice discounts are given depending on the turnover of the customer during the invoiced month. Invoicing and paying occurs in any order: one can already receive a second invoice before having paid the first one. But obviously, each invoice must first have been sent before a payment can be received. Each refuel turn only appears on the invoice of the month of the refuel turn. If a customer doesn't pay the invoice, the unpaid refuel turns are not added to the next month's invoice. Instead, the invoice is simply sent again to the customer to remind him/her of the payment due. If a customer repeatedly does not pay the invoices (e.g. two months in a row), then the customer is suspended and his/her card is blocked until the bills have been paid. The exact threshold to block a customer is not 100% fixed: very good customers are given a bit more slack than new customers. If the bills are settled, a customer may return to the normal state.
- Each gas station has a number of pumps that each contain a particular type of gasoline. Obviously only one customer can refueling his/her car at a time at a given pump. When the nozzle is taken, the refueling can start. When the nozzle is put back the refueling is stopped. Between those two moment, no other customer can use this pump. As soon as the nozzle is put back, the pump becomes available for the next customer.
- Each pump has a separate reservoir that must be refilled when the quantity of gasoline drops below a certain preset reordering level. Refilling can occur at any time. The system should produce a daily report indicating which pump needs to be refilled and for each pump the current level of gasoline and the quantity to order. As the bottom of a tank can contain dirt, when the fuel drops below a critical level, the pump is automatically blocked so that no refuelling can occur with this pump. For the same reason, when refilling a tank that was still above the critical level, the pump is temporarily blocked by the gas station responsible for approximatively an hour, so that dirt can sink to the bottom. The pump is released when the gas station responsible deems that refuelling with this pump is safe again.



Analysis of the first three paragraphs

Text	Enterprise Layer object	Enterprise Layer event	Information system service	Business Process Layer
An oil company possesses a number of gas stations spread across the country.	Oil Company = Universe of Discourse Gas station = Object Type Country = If only one country, then not a **class**. Part of the UoD	default event types for Gas Station	Default services to register new, view, and end gas station	
Customers can pay their refuel turn in cash at the counter or with a personal credit card, but they can also ask for a refuel card.	Customer = Object Type Refuel Turn = Object Type Cash = Not an object type, but a way of paying Counter = physical location of where to pay, not an OT Personal Credit Card = Could be stored as OT, but in this case we do not want to save this information as we don't want to deal with security issues Refuel Card = Card given to customer to be monthly invoiced.	default event types for Customer and Refuel Turn + Pay as event of refuel turn	Default services to register new, view, and end Customer Service to handle refuel turn Service to register payment Service to ask for Refuel Card	business process for - Refueling - Asking for a Refuel Card
When using their refuel card, customers do not pay the refuel turn immediately. Rather, at the end of each month, the holder of the fuel card receives a monthly invoice with the amounts of all the refuel turns paid with the fuel card.	Fuel Card = synonym for Refuel Card Holder of the Card = synonym of Customer Invoice = Object Type	default event types for Fuel Card and Invoice	Service to pay with Refuel Card Service to create & send invoices	business process for - Refueling - Monthly invoicing



Defining Object Types

- Examples from student solutions

Somewhat acceptable

GasStation

The gas stations that the oil company owns. We say it cannot exist without a pump.

Station

Place where pump is located at

GasStation

A gas station is owned by the oil company. It has pumps at which customers can fill up their tanks.

Gas Station

The providing party of pump.

Inacceptable

Gas Station

The Universe of Discourse is an oil company which owns/has many gas stations. This is a master object.

Gas Station

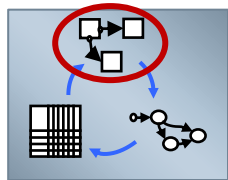
Is the central business object type

Object Type 5: GasStation

GasStation is the Master object of CustomerRelationship and of Pumps.

GasStation

Each gas station is a different object.



Defining Object Types

Object Type 2

Definition

...

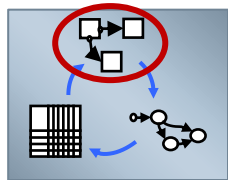
I'm sorry but I don't understand what we are asked to do in this exercise...

a business object

- is a business component relevant only in business domain
- can be described by means of attributes
- has an identity
- corresponds to a real-world phenomenon
- exists for a certain period of time
- and hence, is involved in at least two business events : one for its creation, one for its life ending

business objects are abstracted into object TYPES. An object TYPE is a template for an individual object. a business object CLASS is a set of business objects with the same characteristics, that conform to the same TYPE.

- Go back to the definition of what a business object is
  - Explain why it is relevant in the business domain, what attributes it could have, and how each object has a unique identity
  - More importantly: reflect on the two business events that delineate the life of the objects: when to they enter the universe of discourse, and when do they leave it ?



Defining Object Types

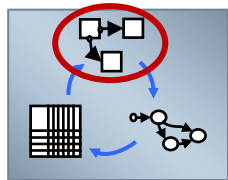
a business object

- is a business component relevant only in business domain
- can be described by means of attributes
- has an identity
- corresponds to a real-world phenomenon
- exists for a certain period of time
- and hence, is involved in at least two business events : one for its creation, one for its life ending

business objects are abstracted into object TYPES. An object TYPE is a template for an individual object. a business object CLASS is a set of business objects with the same characteristics, that conform to the same TYPE.

I am under the assumption that every gas station is identical, as such we are operating in the universe of gas stations and the EDG describes the state of one singular gas station.

- The object type "Gas Station" describes a single gas station



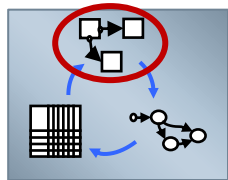
Defining Object Types

a business object

- is a business component relevant only in business domain
- can be described by means of attributes
- has an identity
- corresponds to a real-world phenomenon
- exists for a certain period of time
- and hence, is involved in at least two business events : one for its creation, one for its life ending

business objects are abstracted into object TYPES. An object TYPE is a template for an individual object. a business object CLASS is a set of business objects with the same characteristics, that conform to the same TYPE.

- Gas Station:
  - A Gas Station is a point of sale for fuel. It has a specific address, offers a number of pumps that can be used by the customers to refuel their cars.
  - The Gas Station enters the UoD as soon as its opening is planned. It leaves the UoD when it is dismantled and the site has been decontaminated and released for other use.



Defining Object Types

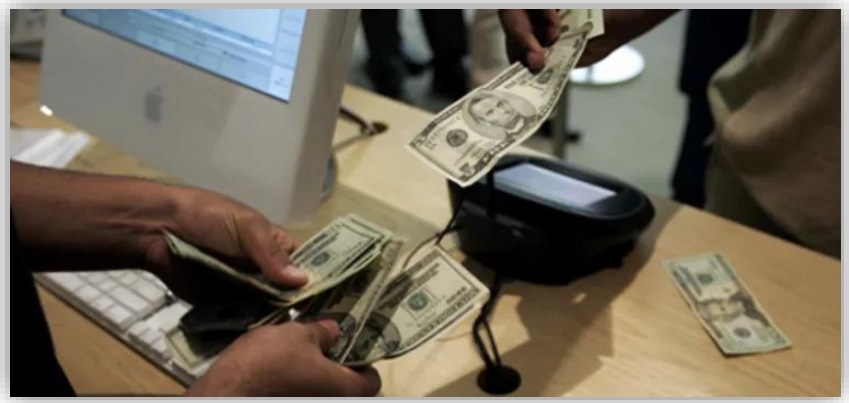
a business object

- is a business component relevant only in business domain
- can be described by means of attributes
- has an identity
- corresponds to a real-world phenomenon
- exists for a certain period of time
- and hence, is involved in at least two business events : one for its creation, one for its life ending

business objects are abstracted into object TYPES. An object TYPE is a template for an individual object. a business object CLASS is a set of business objects with the same characteristics, that conform to the same TYPE.

- Customer
  - A Customer is a person who refuels their car at one of the Gas Stations of the Oil Company

- A Customer enters the UoD
  (= is registered)
  when ... ?



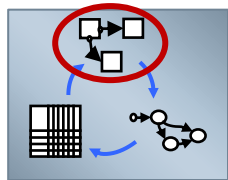
Defining Object Types

a business object

- is a business component relevant only in business domain
- can be described by means of attributes
- has an identity
- corresponds to a real-world phenomenon
- exists for a certain period of time
- and hence, is involved in at least two business events : one for its creation, one for its life ending

business objects are abstracted into object TYPES. An object TYPE is a template for an individual object. a business object CLASS is a set of business objects with the same characteristics, that conform to the same TYPE.

- Customer
  - A Customer is a person who has asked for and possibly holds a Refuel Card of the Oil Company.
    - A person who pays the refuel turn with cash or by credit card remains anonymous for the Oil Company and is therefore not registered as Customer.
  - A person becomes known as a Customer to the Oil Company when the customer asks for a fuel card and provides their details (name, address, etc.). The life of a customer is ended when the customer has had no active fuel card for at least 10 years, and has no outstanding invoices at the Oil Company.



Defining Object Types

- Examples from student homeworks

Object Type 3: Customer

The customer is another object type as he is the one performing the process of refueling.

Customer

The customers that visit the gas stations.

Customer

A customer is a person that can fill up his/her tank at the gas station at a specific pump. Customers can have a refuel card and have a refuel turn each time they fill up their tank. A customer can be suspended if (s)he does not pay his/her bills and unsuspended when (s)he pays.

Customer

The one using the pump

Customer

Generic customer object, has attributes like Name, Surname, Adress, AmtOfRefuels etc...

(amount of refuels attribute can then be used for calculating 'non-payment slack'-terms in case of unpaid invoices.)

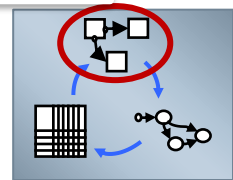
Customer

Each customer is a separate object, with each refueling event having exactly one customer.

Good starting point, but missing the explanation of when the objects enter and leave the Universe of Discourse

--> Reflecting on start and end of life helps to define the object types more precisely

Do not describe the object type in terms of relationships to other elements or usage of attributes

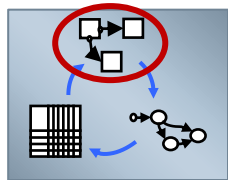


Defining Object Types

- (Re)Fuel Card
  - Card held by a customer that he can use to ask for delayed payment of a refuel turn via invoice
  - Is created upon request by the customer
    - One can imagine that this will require some approval process
  - End when the life of the customer ends
    - unhappy path: ends when lost by the customer?

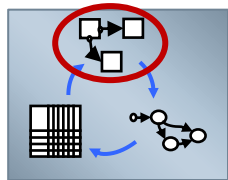
Defining Object Types

- **CUSTOMER vs CARD**
  - A card could be just a means to identify a customer. In that case, CUSTOMER and FUEL CARD are one and the same object type. The company will not keep track of multiple cards per customer. If a card is lost, a new card is issued independently of the previous card.
  - This choice has obviously consequences on the possible functionality that is implied by the domain model. By keeping the concepts of CARD and CUSTOMER in one class, several simultaneous or consecutive cards of a single customer are in principle not related with each other. This means for example that the customer will receive separate mails for each of his/her cards. The only way to discover which cards belong to the same person is by querying the set of cards for identical name of card-holder or identical address.



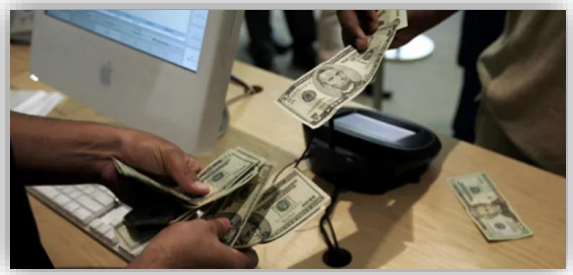
Defining Object Types

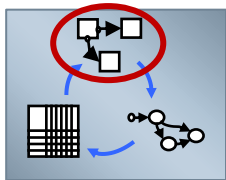
- **CUSTOMER vs CARD**
  - Alternatively, a customer might have several cards, either in sequence or simultaneously. In that case, CUSTOMER and FUEL CARD are distinct object types.
  - In the following slides, the solution assumes that "CARD HOLDER" is a single object type to capture the concepts of customer and fuel card.
  - If a separate customer object type is needed, it can be added as master of "Card Holder".



Defining Object Types

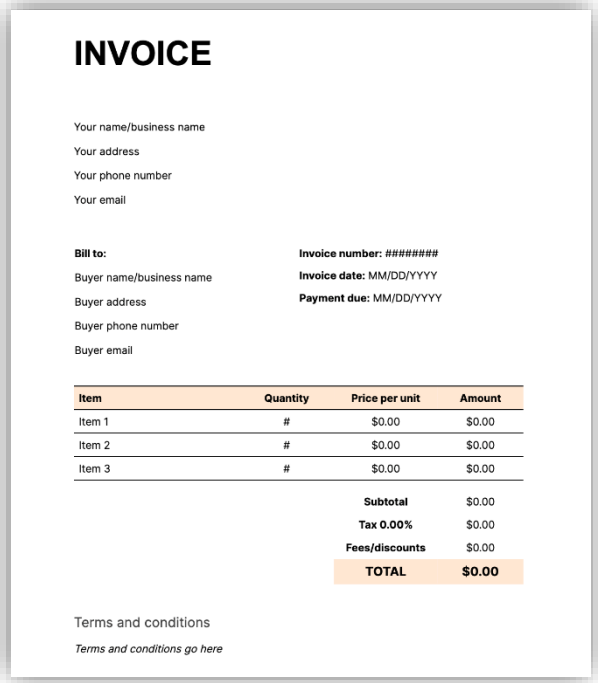
- Refuel Turn
  - This is what the actual business is about
    - generates income for the business
    - offers value for both the customer and the business
  - Starts when the customer takes the nozzle to refuel
  - Ends when?
    - Consider the process of refueling: when does it end? When you put it back ? Think again ...





Defining Object Types

- Invoice
  - Request to a customer to pay for the refuel turns of the past month
  - Is created by the system on a monthly basis
  - Life ends when it has been paid.
    - unhappy path: ends if the Oil Company abendons obtaining payment



INVOICE

Your name/business name
Your address
Your phone number
Your email

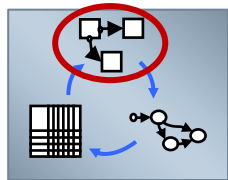
Bill to:
Buyer name/business name
Buyer address
Buyer phone number
Buyer email

Invoice number: ########
Invoice date: MM/DD/YYYY
Payment due: MM/DD/YYYY

Item	Quantity	Price per unit	Amount
Item 1	#	$0.00	$0.00
Item 2	#	$0.00	$0.00
Item 3	#	$0.00	$0.00
		Subtotal	$0.00
		Tax 0.00%	$0.00
		Fees/discounts	$0.00
		TOTAL	**$0.00**

Terms and conditions

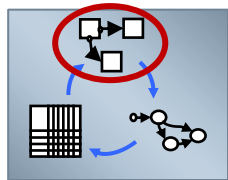
Terms and conditions go here



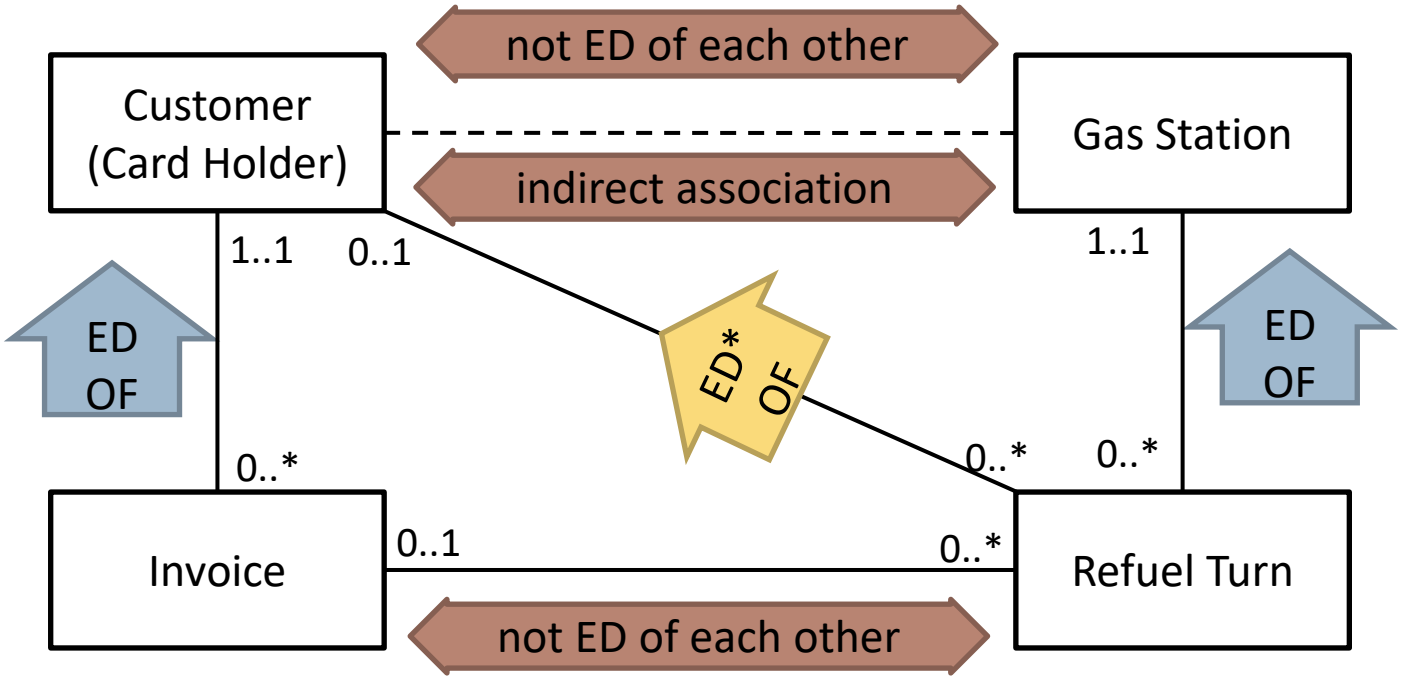
Are NOT object types

- You don't create and end
  - Cash
  - Car
  - Country

- IS Services: Report + Report Line
- Events: Refill, Pay, Refuel

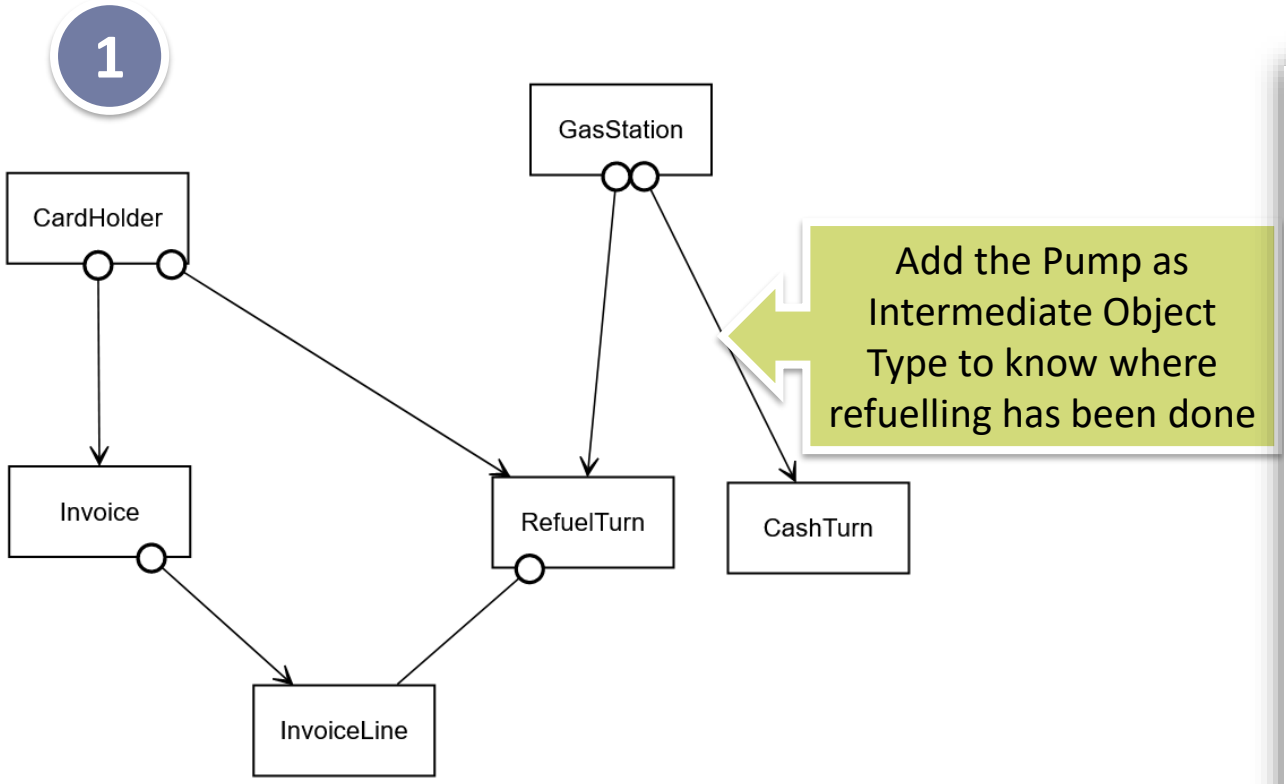


Relating the Object Types

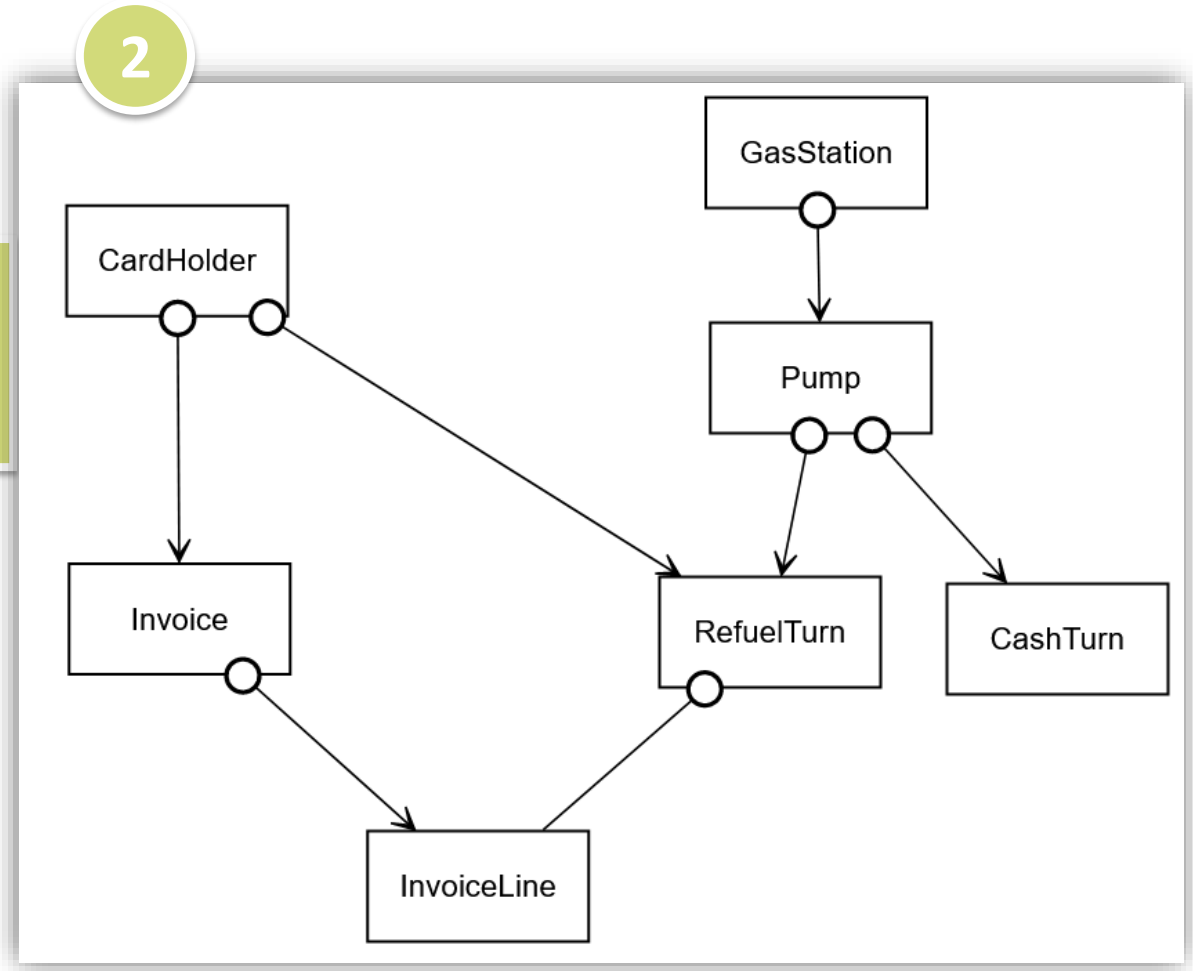


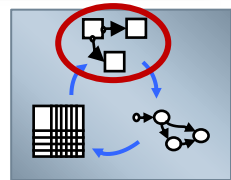
* If the Customer is known

Model Solution



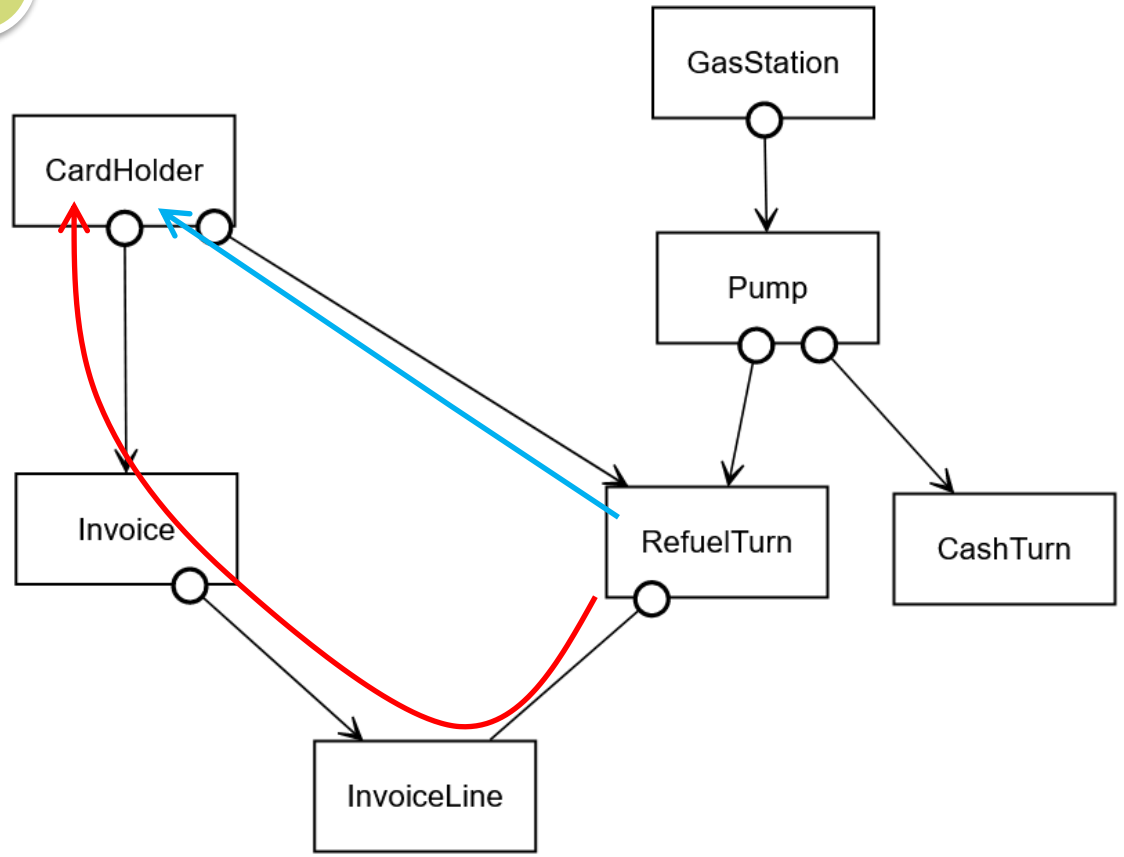
Add the Pump as Intermediate Object Type to know where refuelling has been done





Model Solution

2

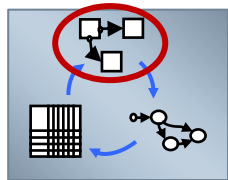


Two paths between RefuelTurn and CardHolder.

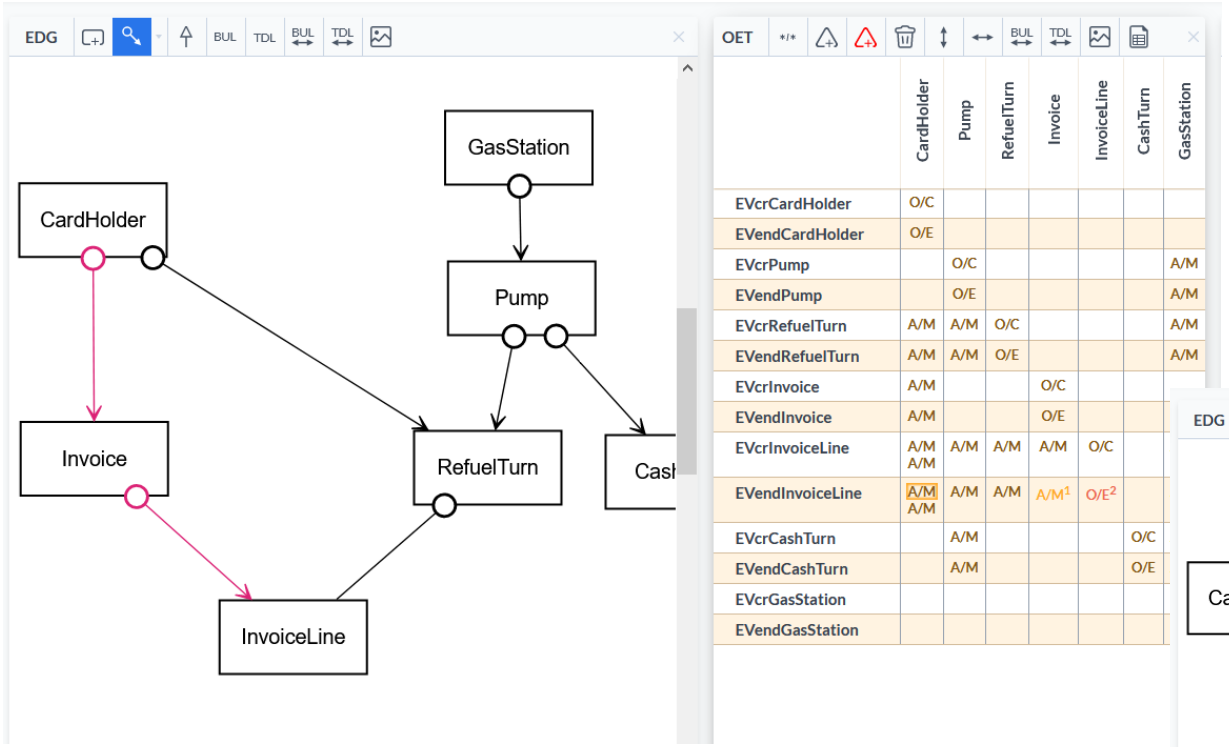
Both are needed:

- The blue path exists first
- The red path is created later, when the invoice is created

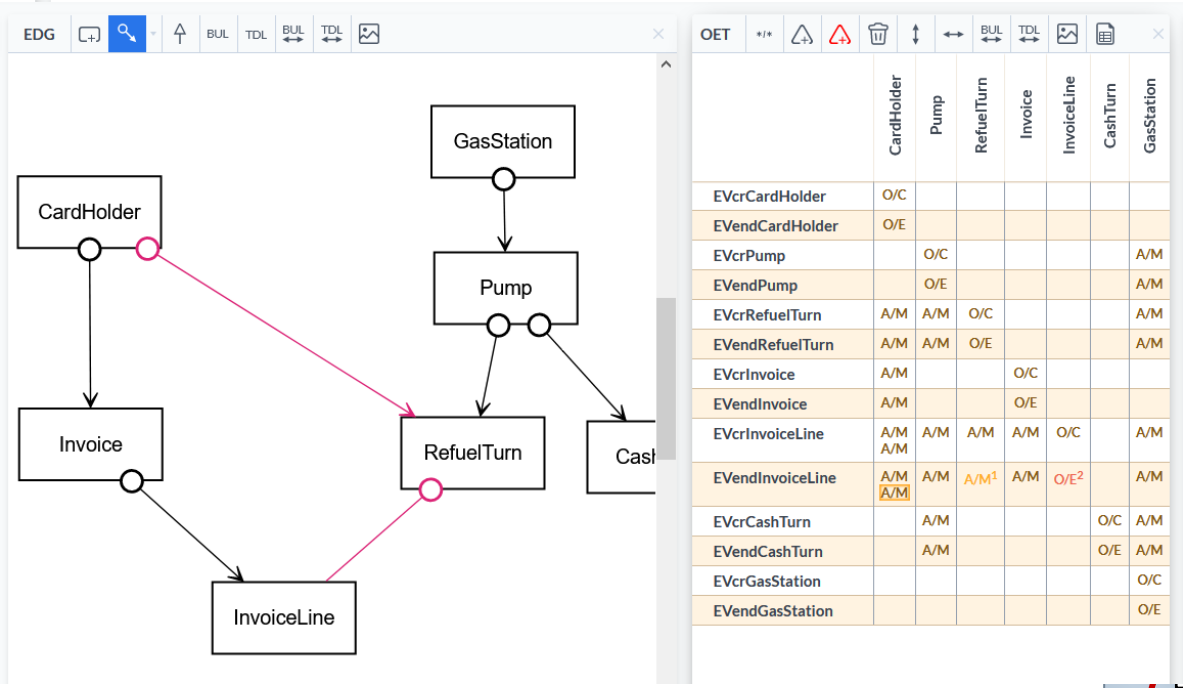
The blue one is needed in order to invoice the right person



OET



	CardHolder	Pump	RefuelTurn	Invoice	InvoiceLine	CashTurn	GasStation
EVcrCardHolder	O/C						
EVendCardHolder	O/E						
EVcrPump		O/C					A/M
EVendPump		O/E					A/M
EVcrRefuelTurn	A/M	A/M	O/C				A/M
EVendRefuelTurn	A/M	A/M	O/E				A/M
EVcrInvoice	A/M			O/C			
EVendInvoice	A/M			O/E			
EVcrInvoiceLine	A/M A/M	A/M	A/M	A/M	O/C		
EVendInvoiceLine	A/M A/M	A/M	A/M	A/M[1]	O/E[2]		
EVcrCashTurn		A/M				O/C	
EVendCashTurn		A/M				O/E	
EVcrGasStation							
EVendGasStation							

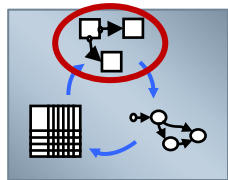


	CardHolder	Pump	RefuelTurn	Invoice	InvoiceLine	CashTurn	GasStation
EVcrCardHolder	O/C						
EVendCardHolder	O/E						
EVcrPump		O/C					A/M
EVendPump		O/E					A/M
EVcrRefuelTurn	A/M	A/M	O/C				A/M
EVendRefuelTurn	A/M	A/M	O/E				A/M
EVcrInvoice	A/M			O/C			
EVendInvoice	A/M			O/E			
EVcrInvoiceLine	A/M A/M	A/M	A/M	A/M	O/C		A/M
EVendInvoiceLine	A/M A/M	A/M	A/M[1]	A/M	O/E[2]		A/M
EVcrCashTurn		A/M				O/C	A/M
EVendCashTurn		A/M				O/E	A/M
EVcrGasStation							O/C
EVendGasStation							O/E

Decomposing the Enterprise Model

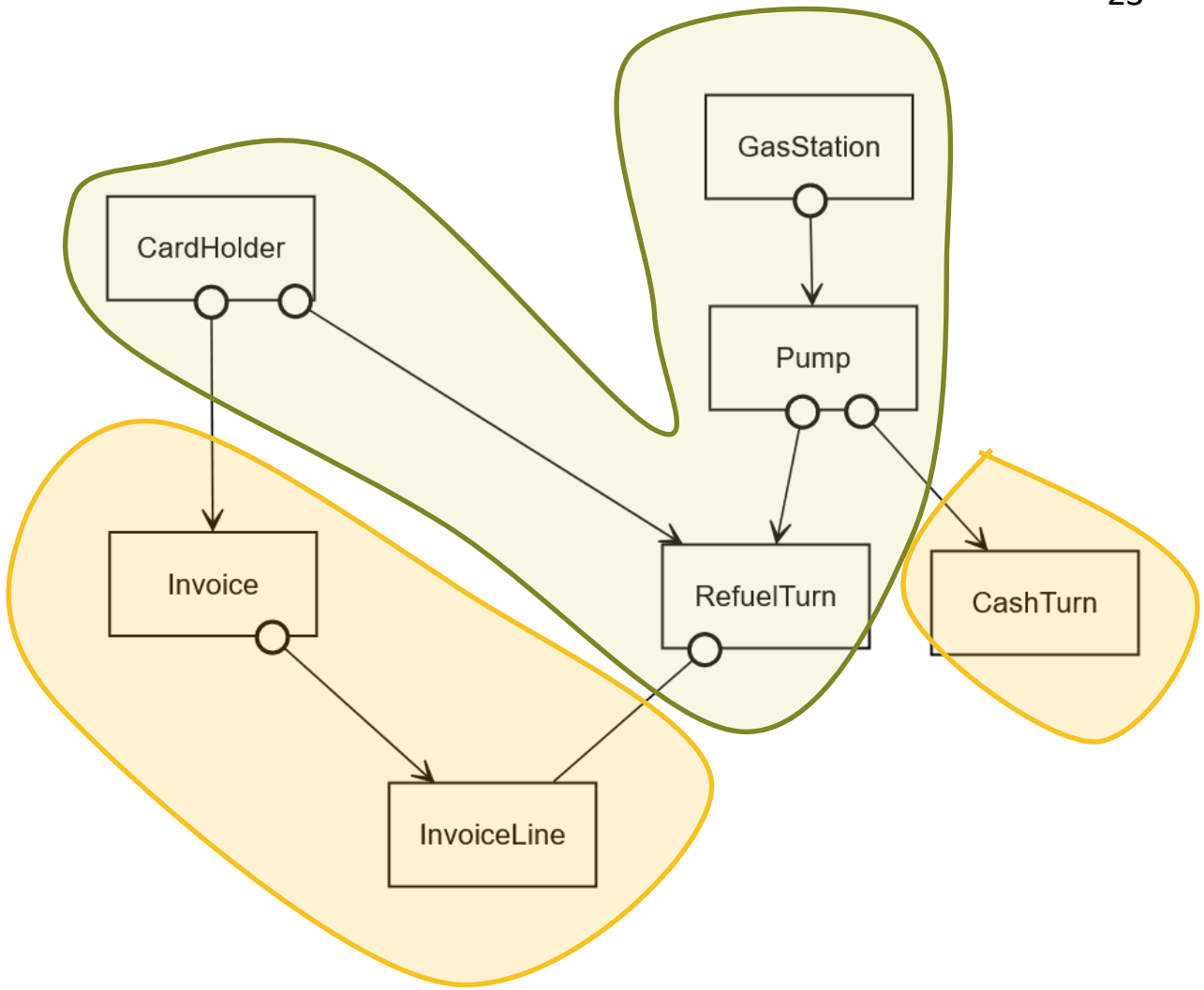
- What is the core business object? What is this business selling ?
  - This business is about selling fuel.
  - What is invoiced/paid for is a "Fuel take" or "Refuel Turn"

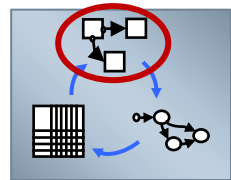
- All other information is needed to manage the business
  - Infrastructure: where did it happen (pump ==> gas station)
  - By whom: card holder, or customer paying cash



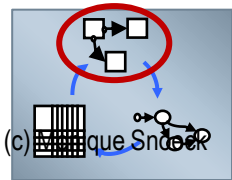
Grading

- Approximation
  - 7 objects + 7 associations
  - added elements: 0 if OK, -1 if not OK.
- Satisfactory
  - **Core** is OK
  - No major mistakes in further elements
- Good
  - **Core** is OK
  - **Additional elements** also OK
- Fail
  - Core not OK
  - Core OK, but too many other major mistakes



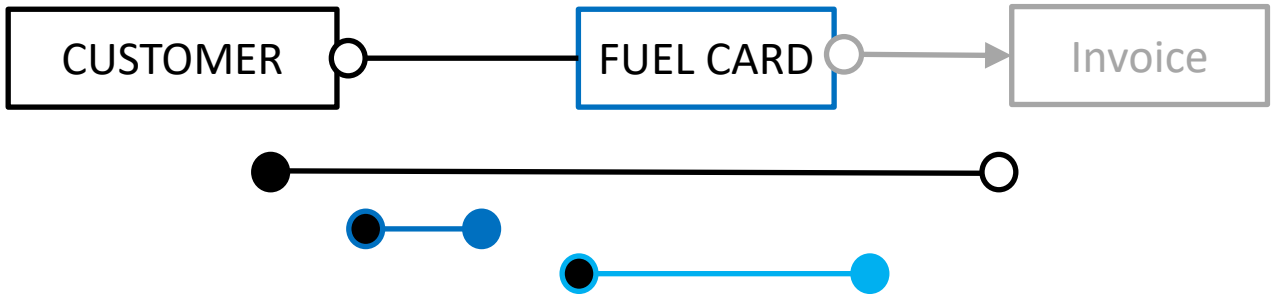


Additional Object Types

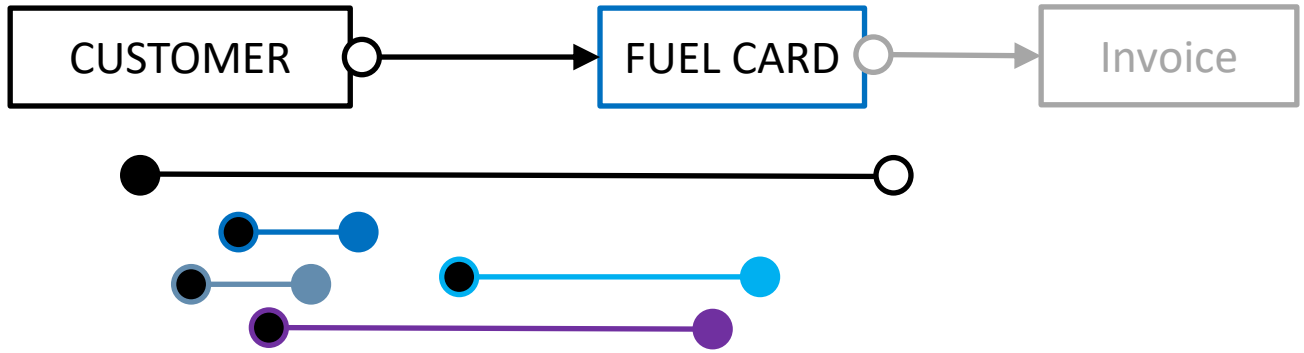


Adding Fuel Card

- Customer versus Fuel Card:
  - This model allows to keep track of multiple consecutive cards belonging to the same customer.

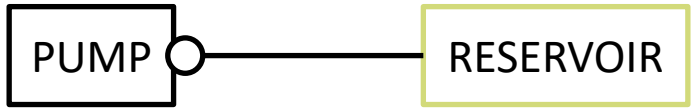


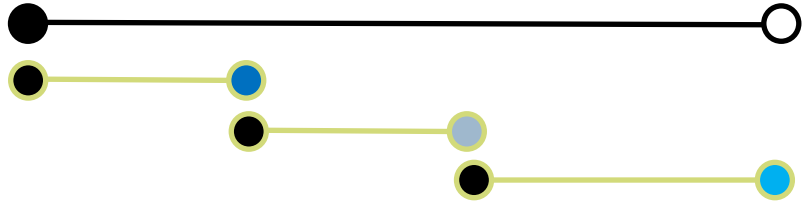
  - An optional-many association would be a good solution for a company having many cards for its employees.



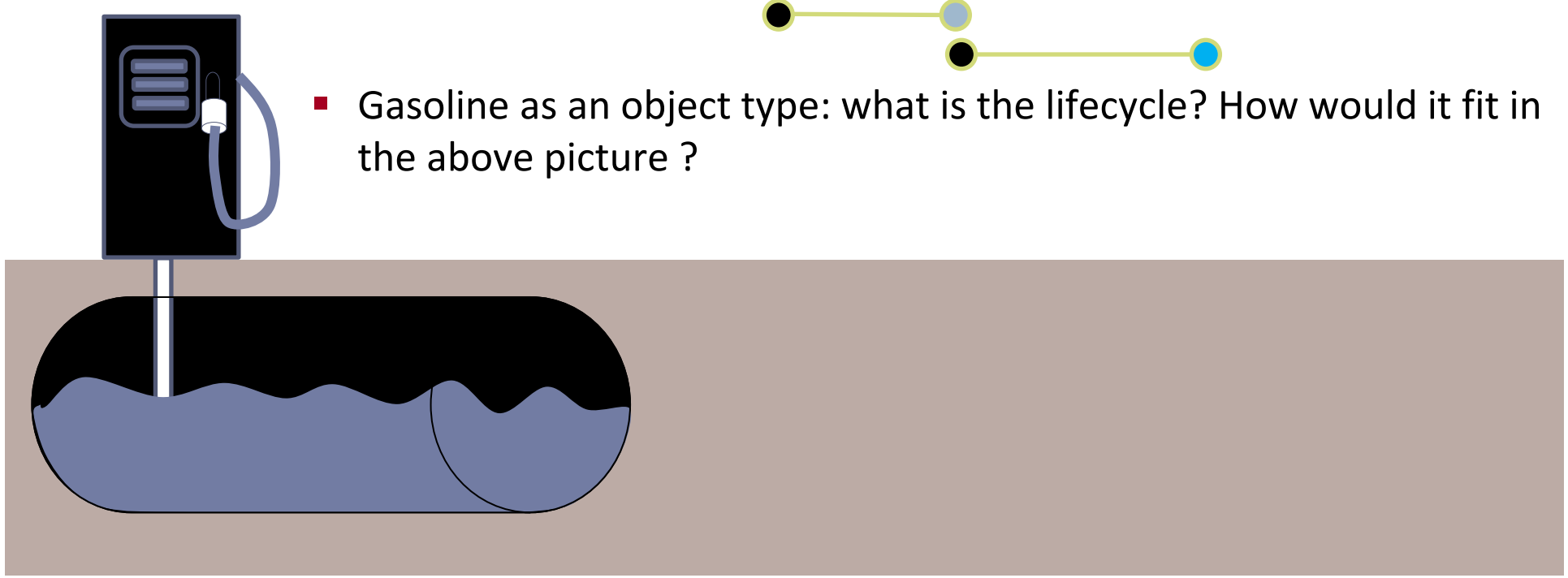
Adding Reservoir: Frequent Student Solutions

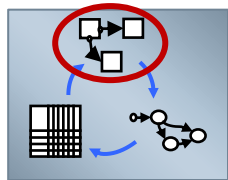
- Pump versus Reservoir:





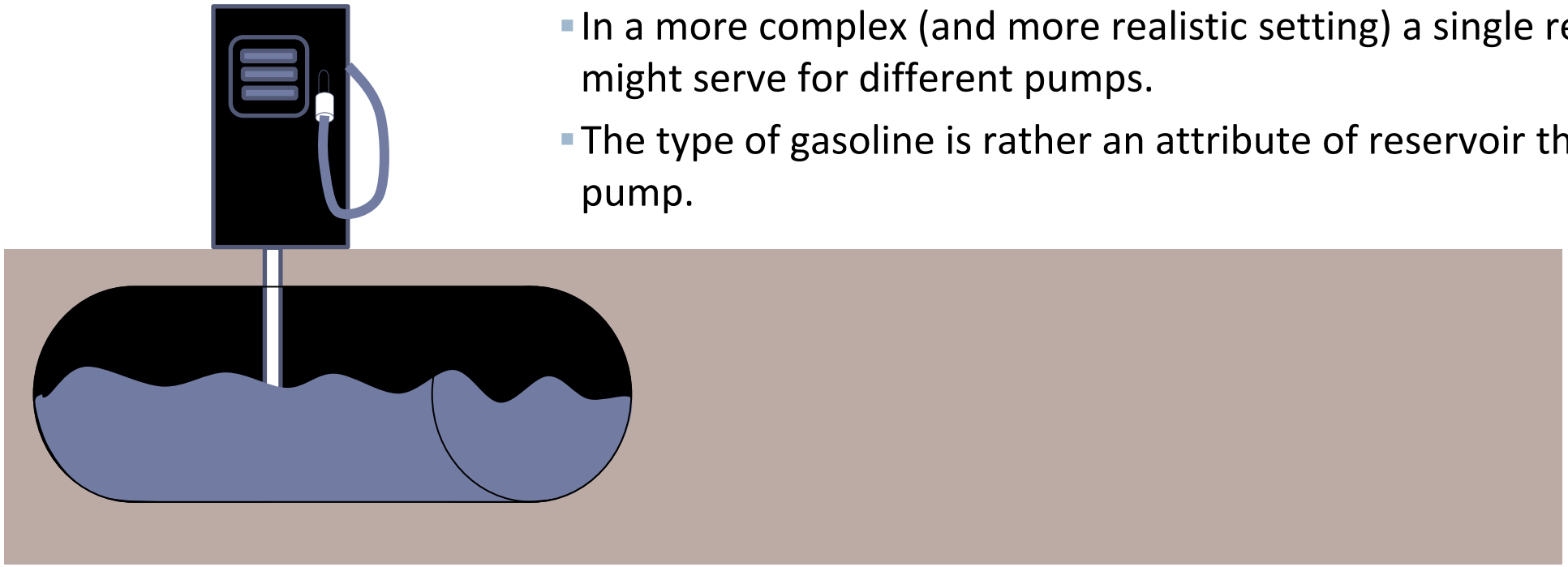
- Gasoline as an object type: what is the lifecycle? How would it fit in the above picture ?

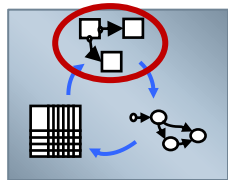




Adding Reservoir

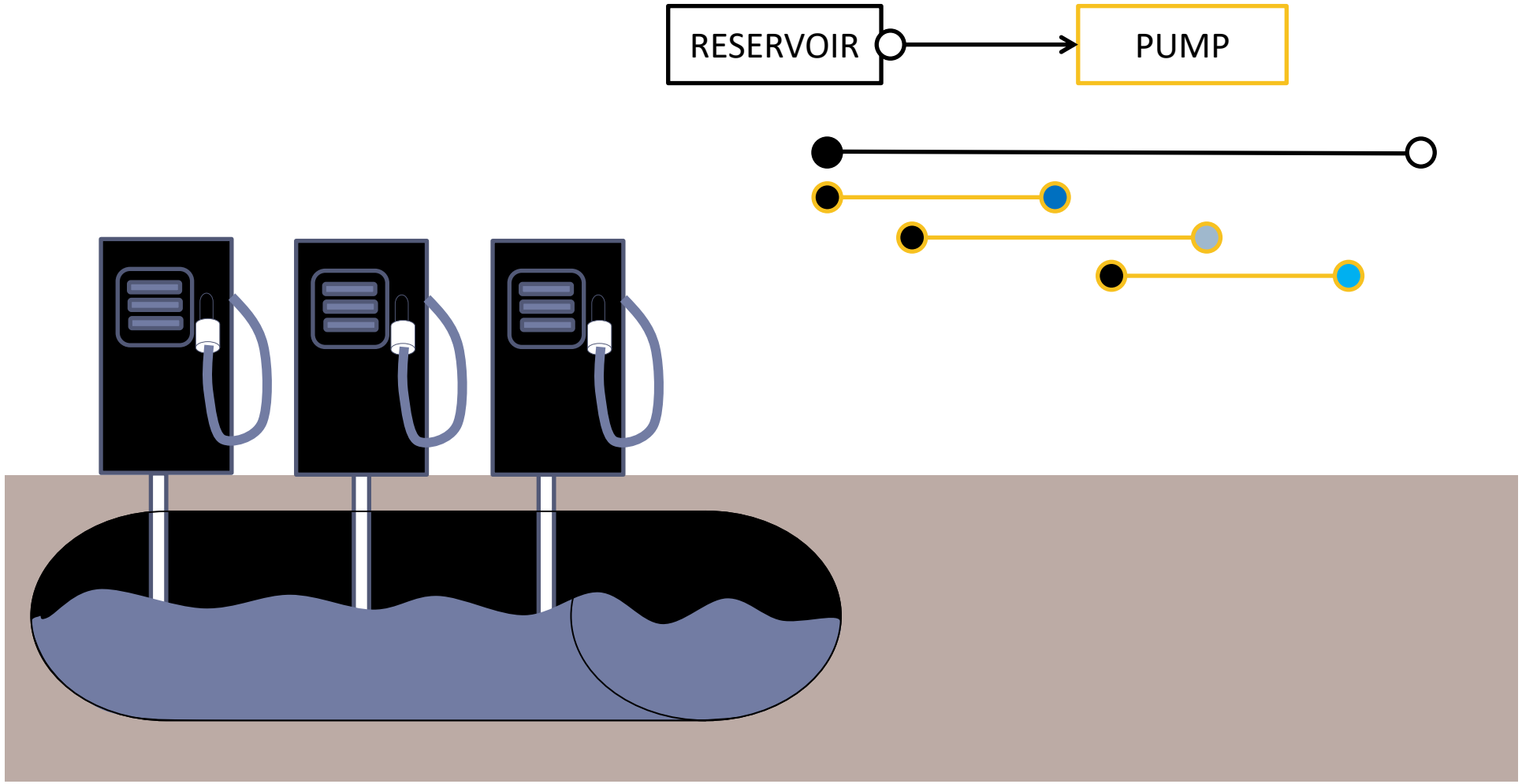
- Pump versus Reservoir:
  - You can consider these as a single object.
  - If you consider these as separate objects, it's more likely that the pump will change over time and the tank will have a longer life cycle than the pump
  - In a more complex (and more realistic setting) a single reservoir might serve for different pumps.
  - The type of gasoline is rather an attribute of reservoir than of pump.





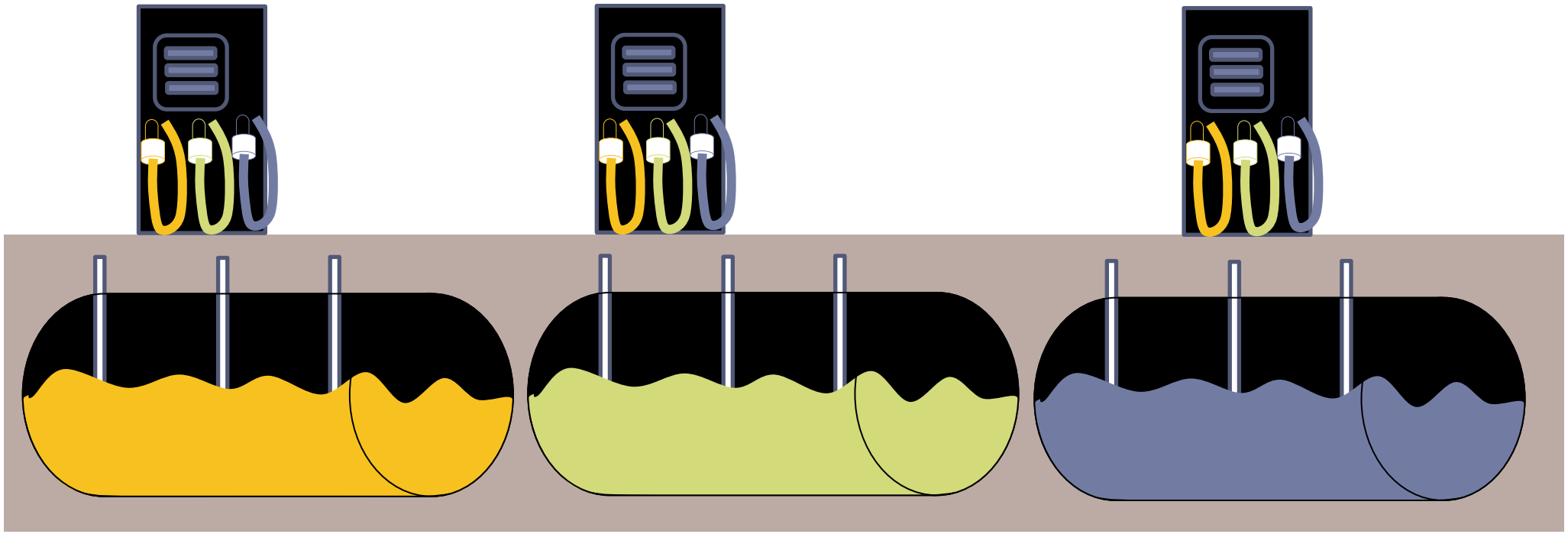
Adding Reservoir: Frequent Student Solutions

- Pump versus Reservoir:

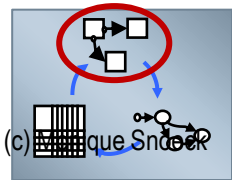


Pumps with several nozzles (one per type of gazoline)

- Pump versus Reservoir:

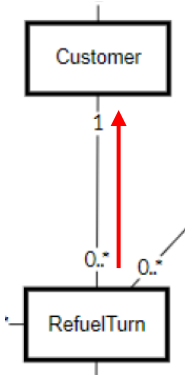


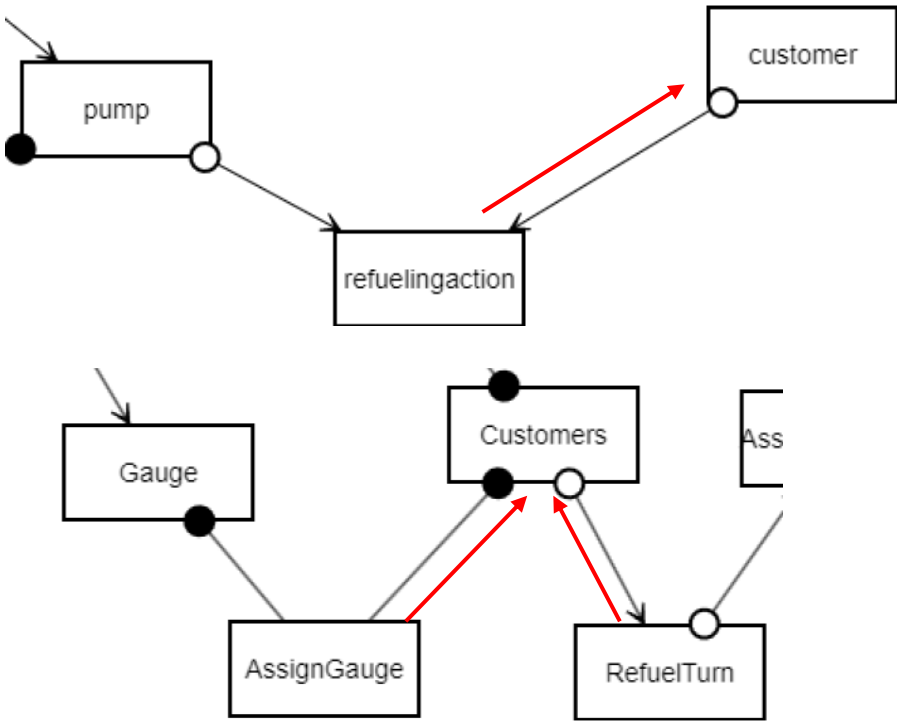
Student Solutions 2023

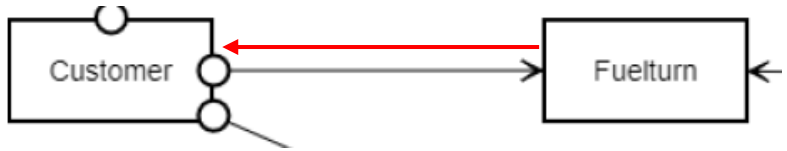


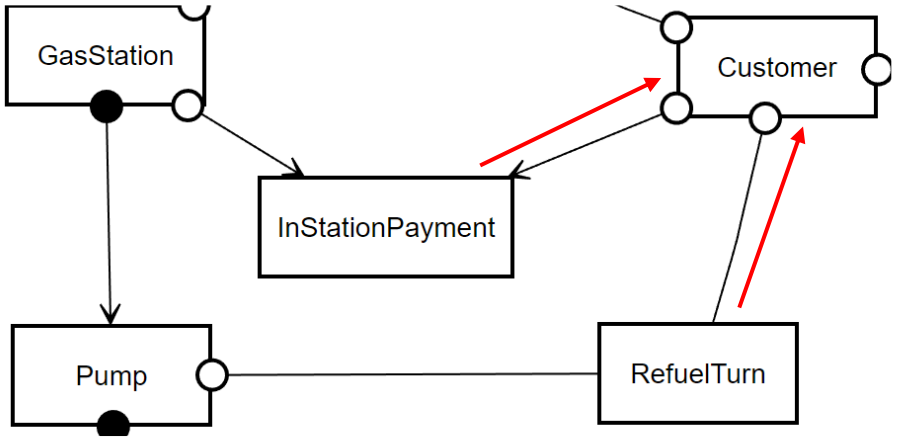
Cash Turn

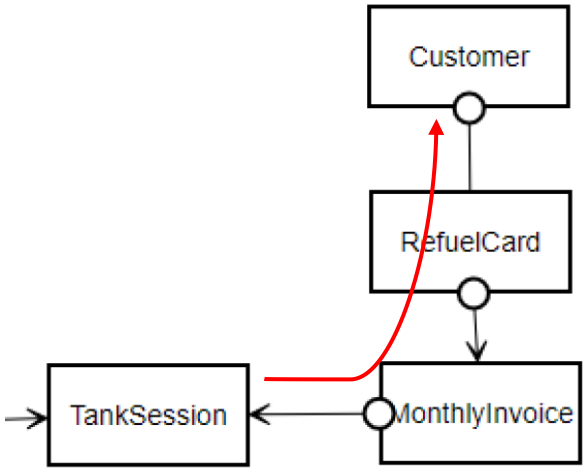
- Any refueling action is ED on Customer. Therefore, anonymous refuels are not possible.

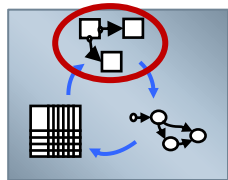






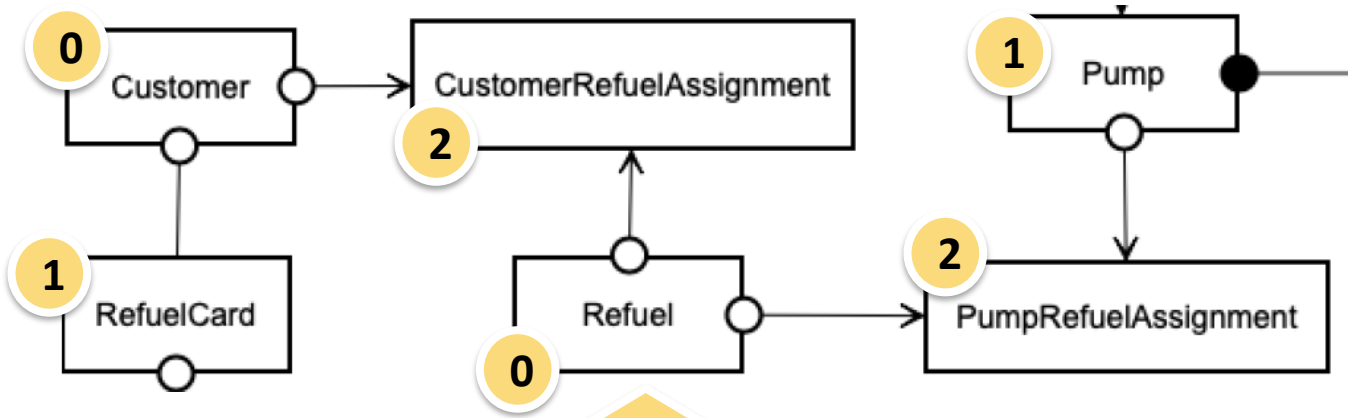


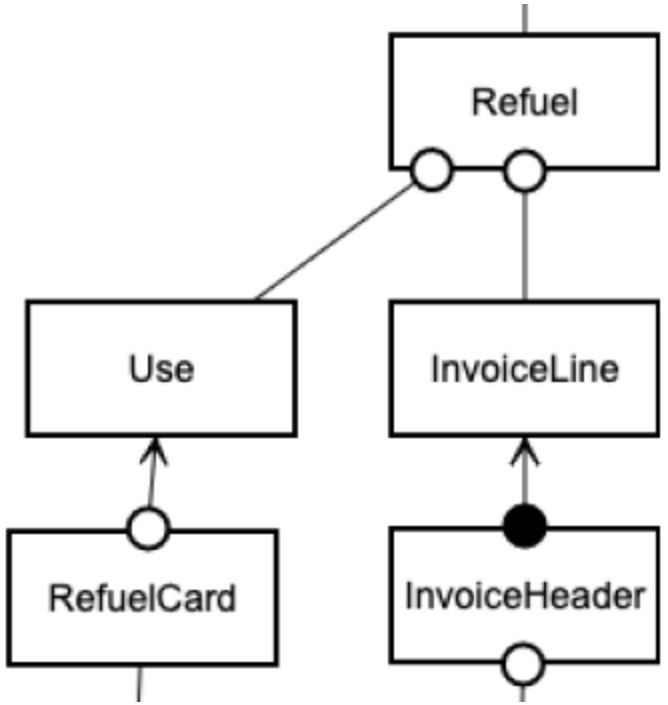




CashTurn

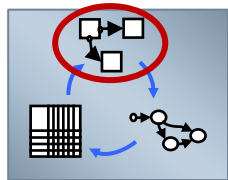
- Linking a RefuelTurn to a Customer/Invoice/Card is optional. How do you know when a RefuelTurn is already paid, or when it still needs to be invoiced?





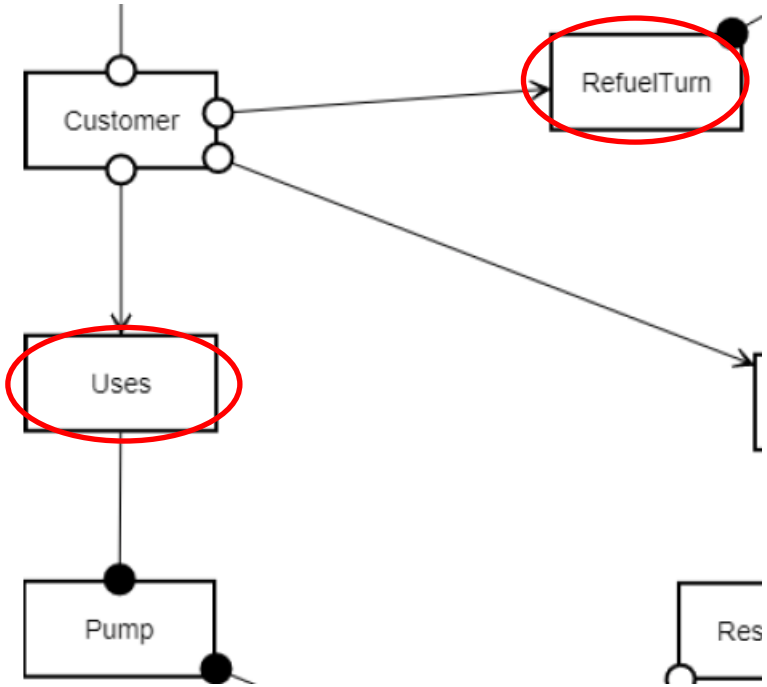
<u>Check the definition of a Business Object Type</u>: Analysing creating and ending events of each of this proposed object types, will help to see where this model can be improved.

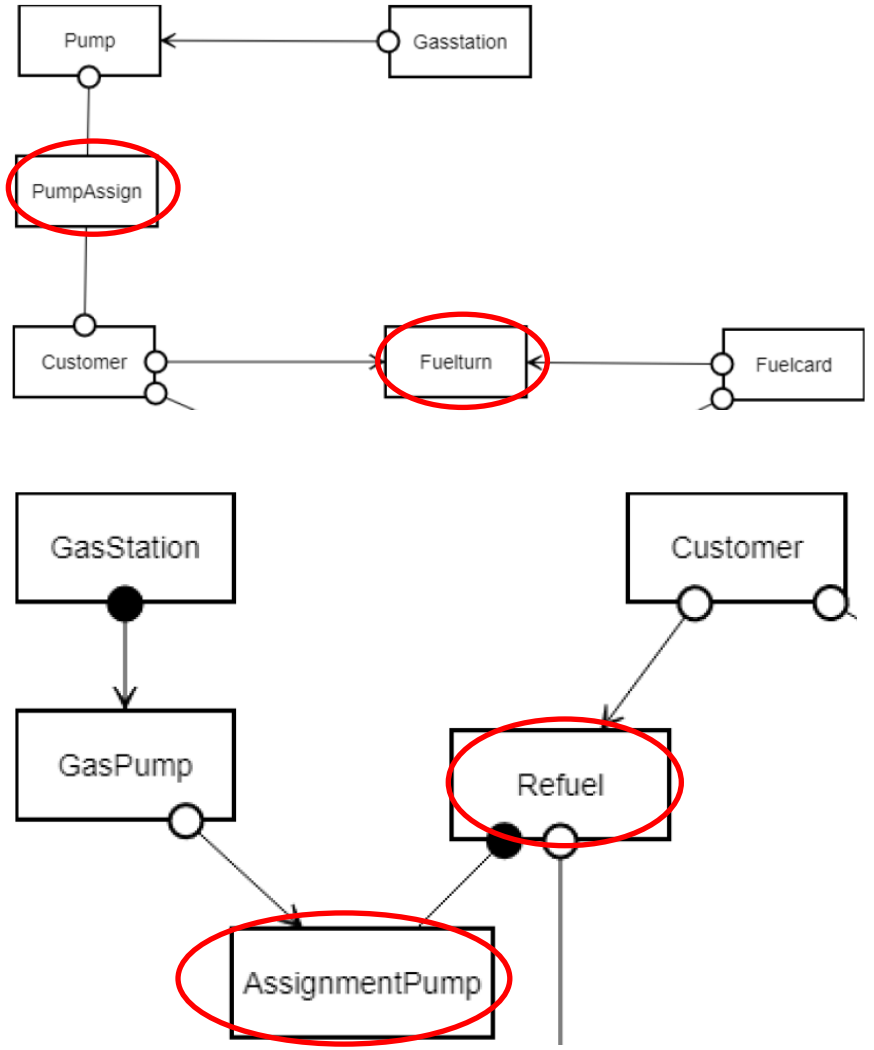
Check the TDL too !!



FuelTurn

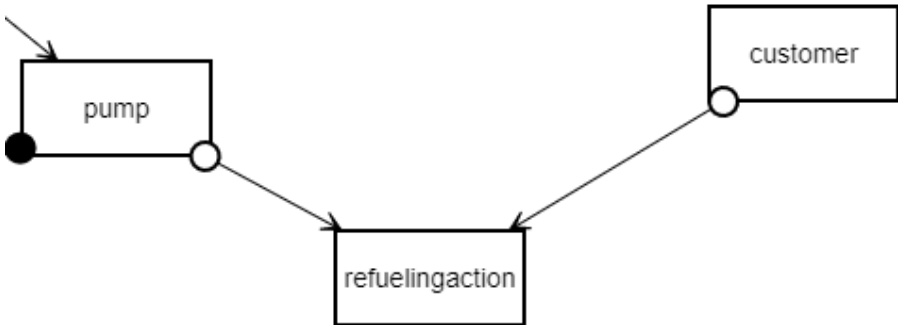
- 2 different object types that represent FuelTurn



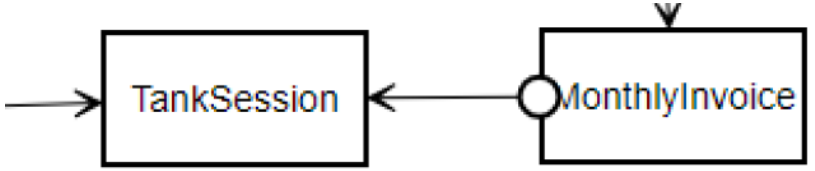


Invoice & InvoiceLine

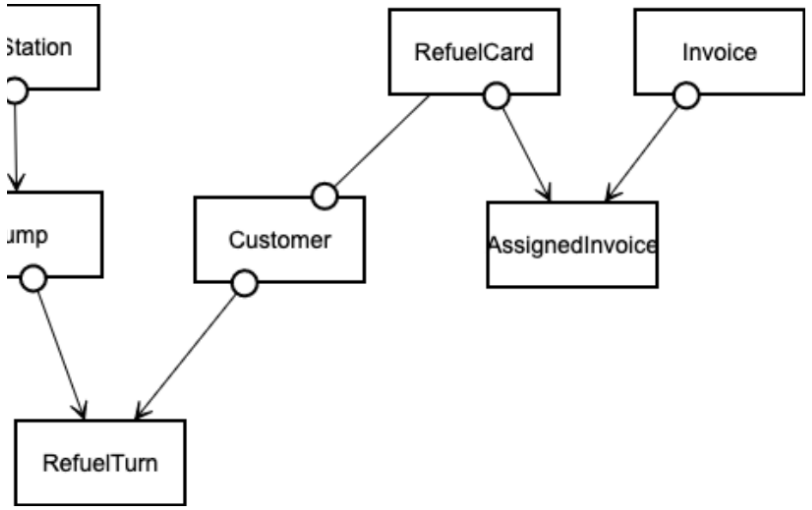
- Invoice should be included.



- What exists first, the TankSession or the Invoice?

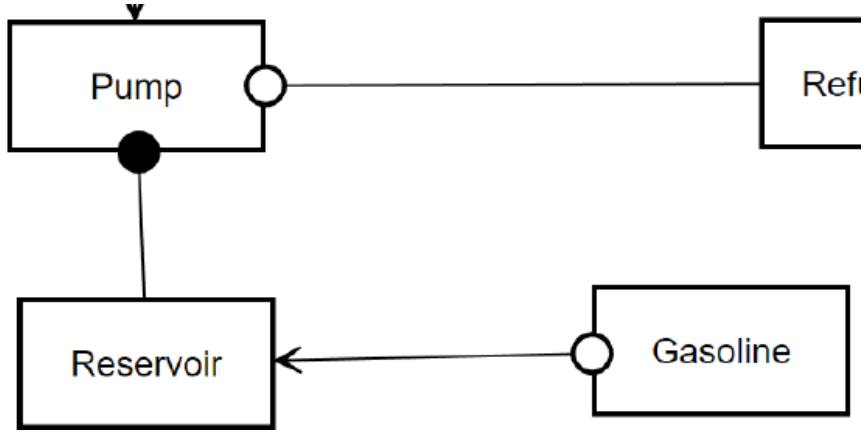


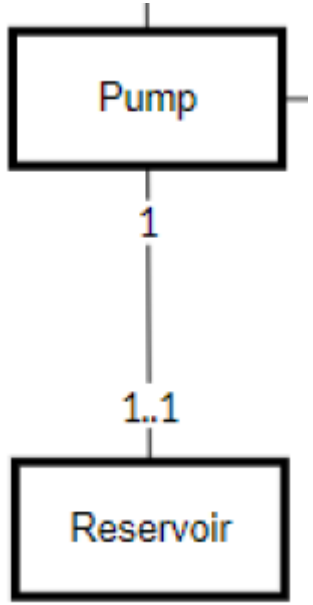
- Can the Customer/RefuelCard of the Invoice change over time?
- Which RefuelTurns are invoiced?



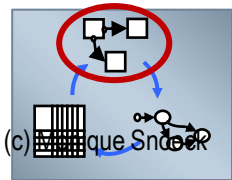
Pump & Reservoir

- Reservoir is not required as a separate object type. But if it is, which exists first, the pump or the reservoir? See previous slides

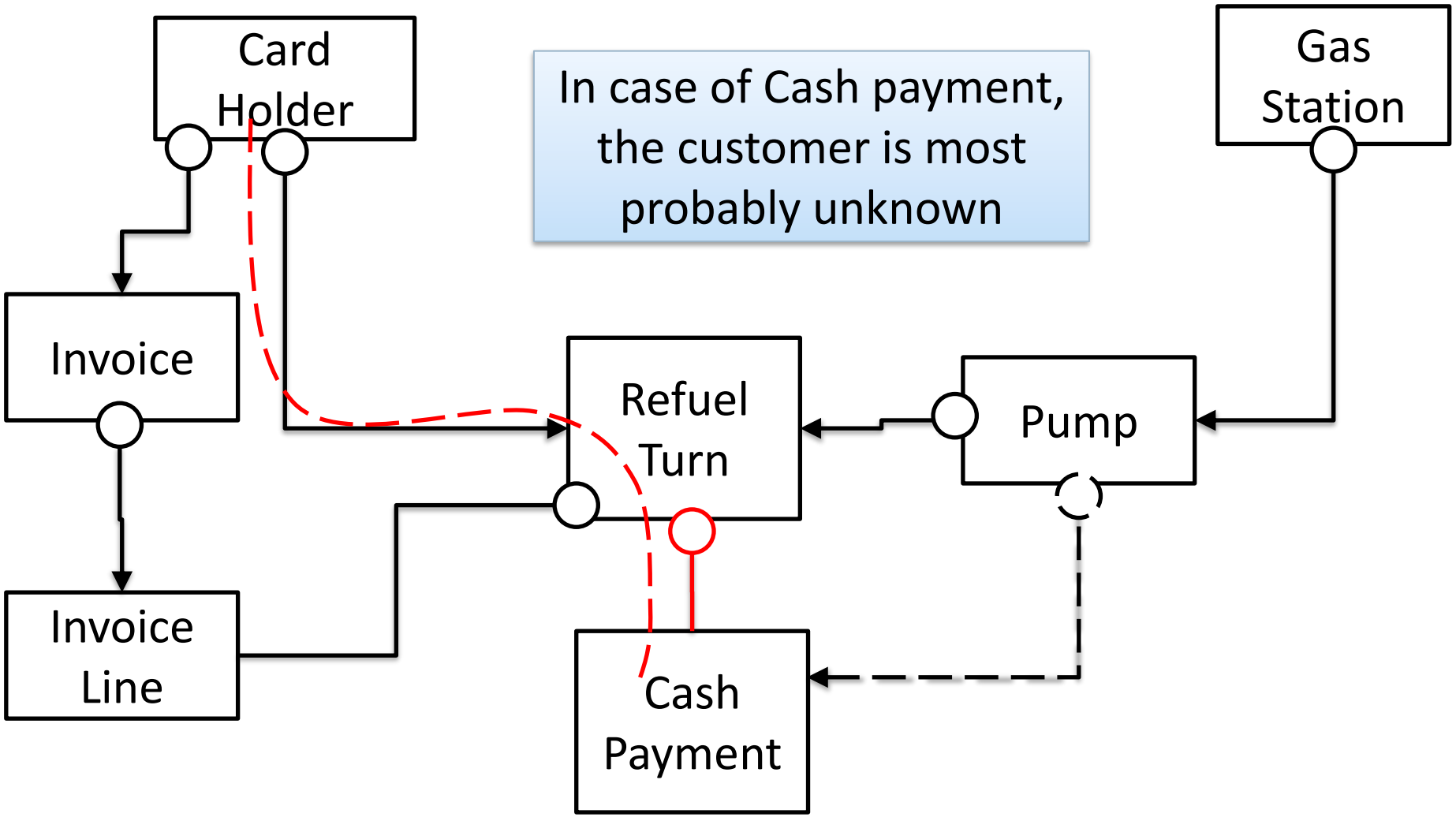




Older Student Solutions with comments

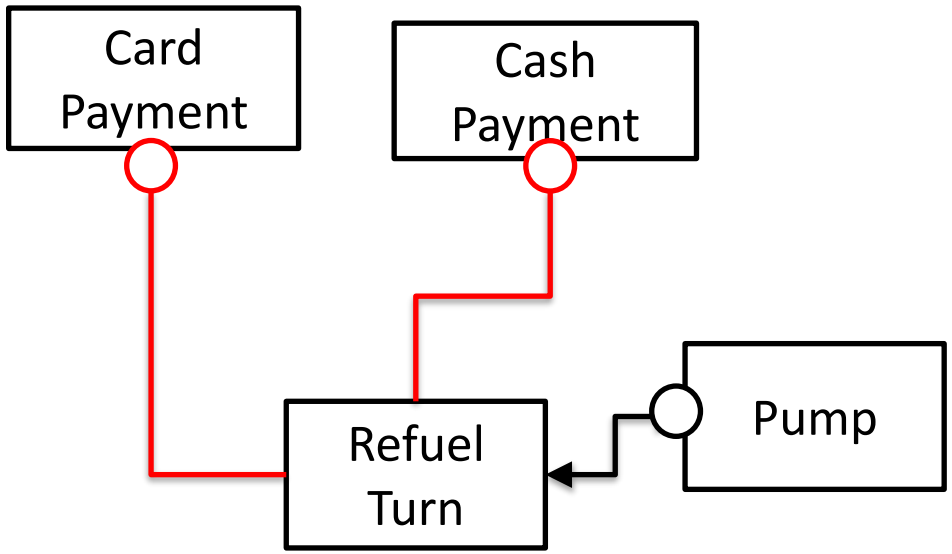


Student Solutions



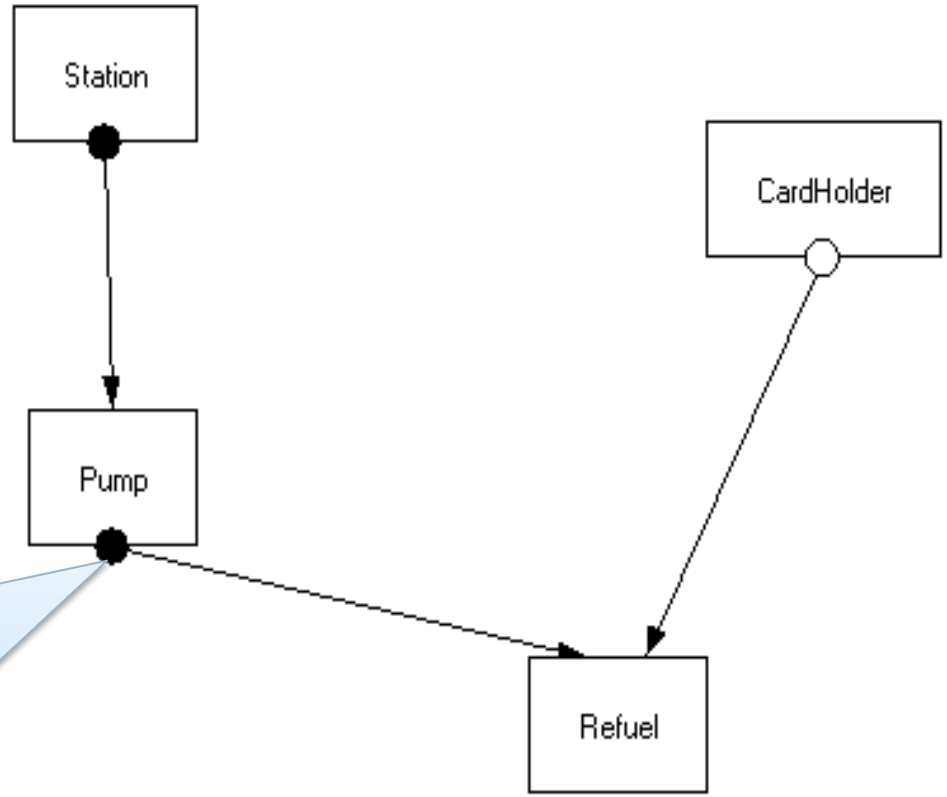
Student Solutions

- Assumes pre-payment.
- But, as masters are mandatory ==> this solution requires a refuel turn to have both a cash payment and a card payment

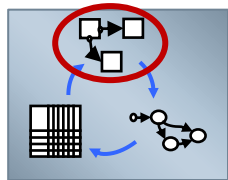


Student solutions

Most simple model; fairly ok

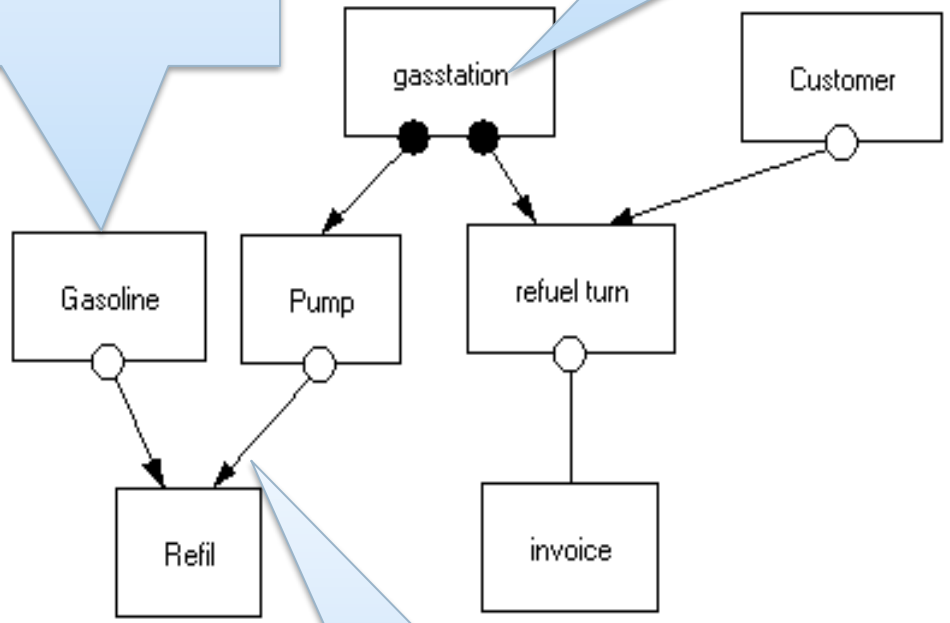


If the relationship is mandatory this implies there must be at least one refuel turn in order for the pump to exist ...



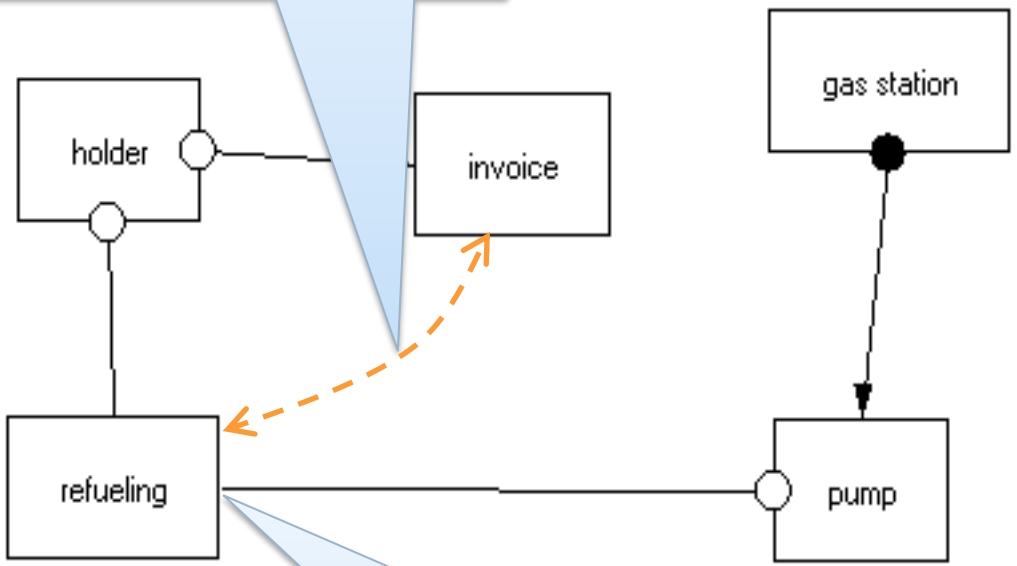
If Refuel Turn is made existence dependent from Gas-station, how do we know at which pump the refuel took place ? How do we know what type of Gasoline was taken and what price to invoice ?

Should Gasoline be modelled as an Object type or as an Attribute of Pump? see further discussion on this issue in Chapter 7.

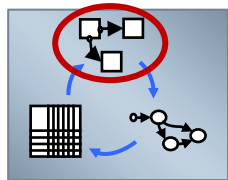


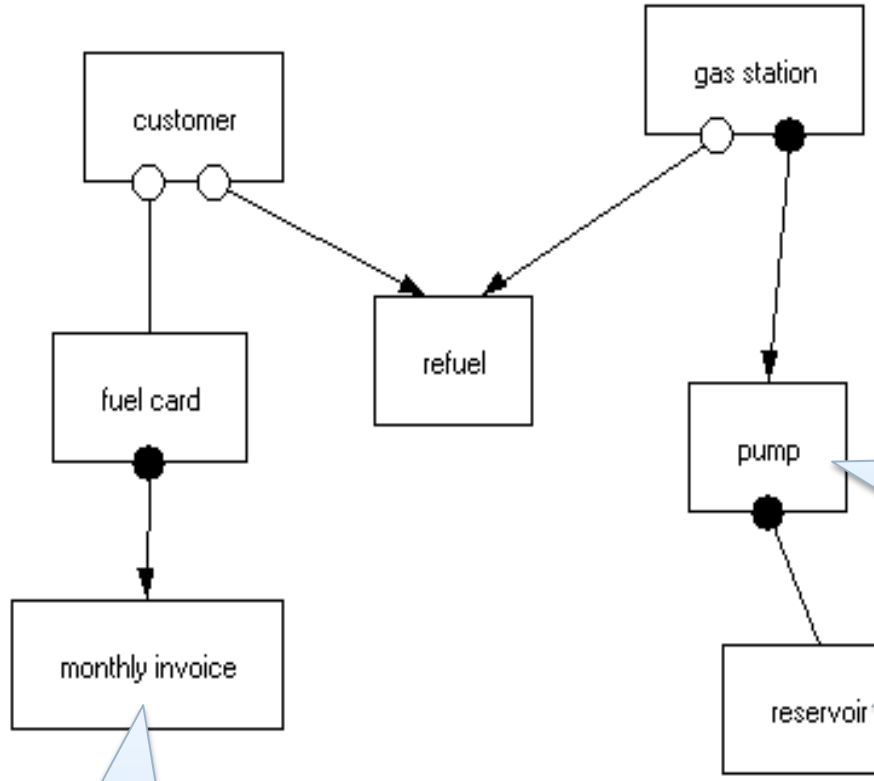
Many refills at 1 point in time?

With this model, how can you know when or if a refuel turn has been invoiced?



Only one refueling at one point in time, but what if not paid?
→ means that you destroy refuel-info before next refuel turn starts

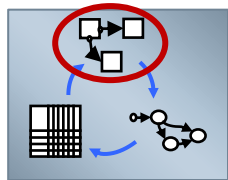


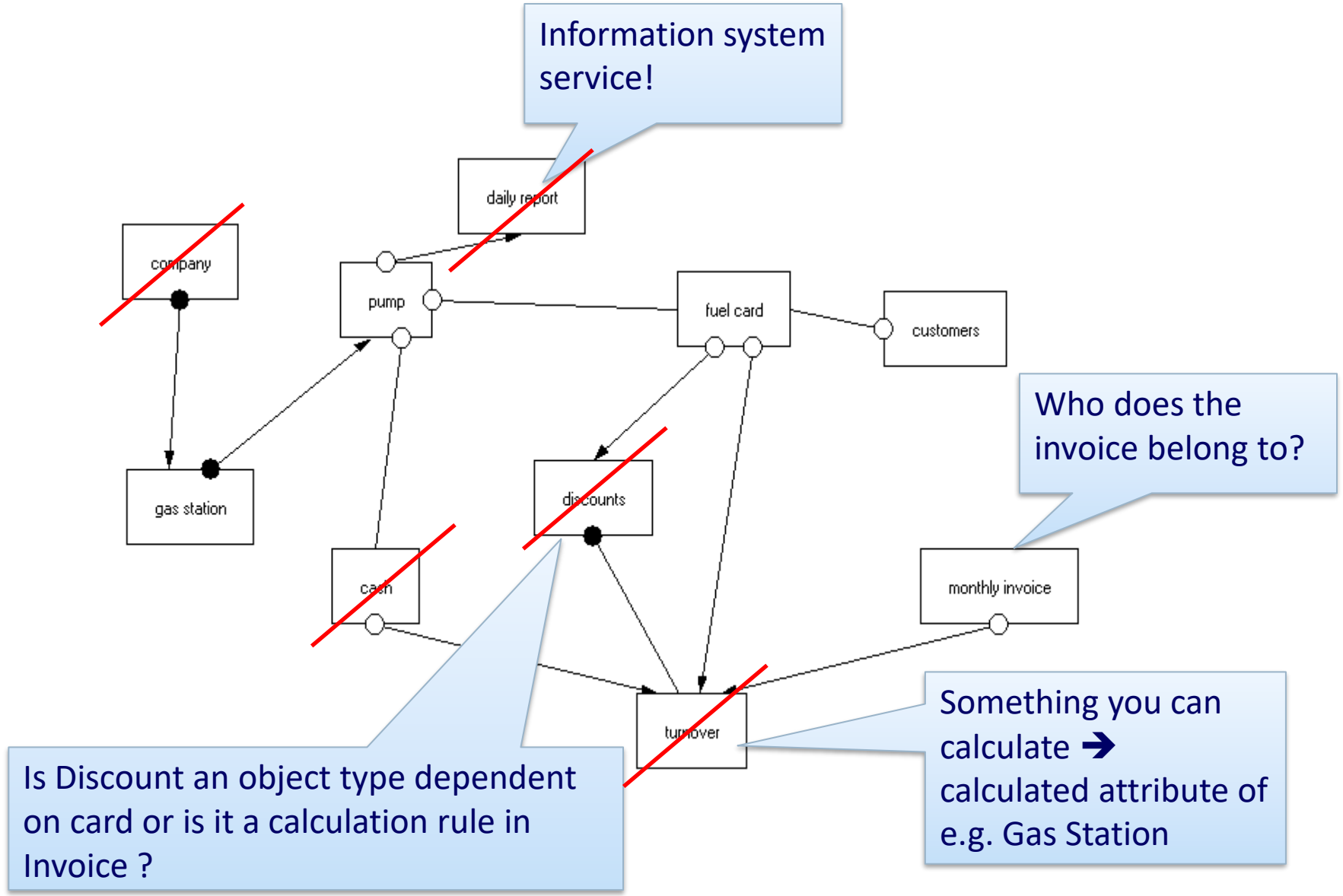


Same lifecycle? If it is the same lifecycle, you probably have a single object type instead of two different object types.

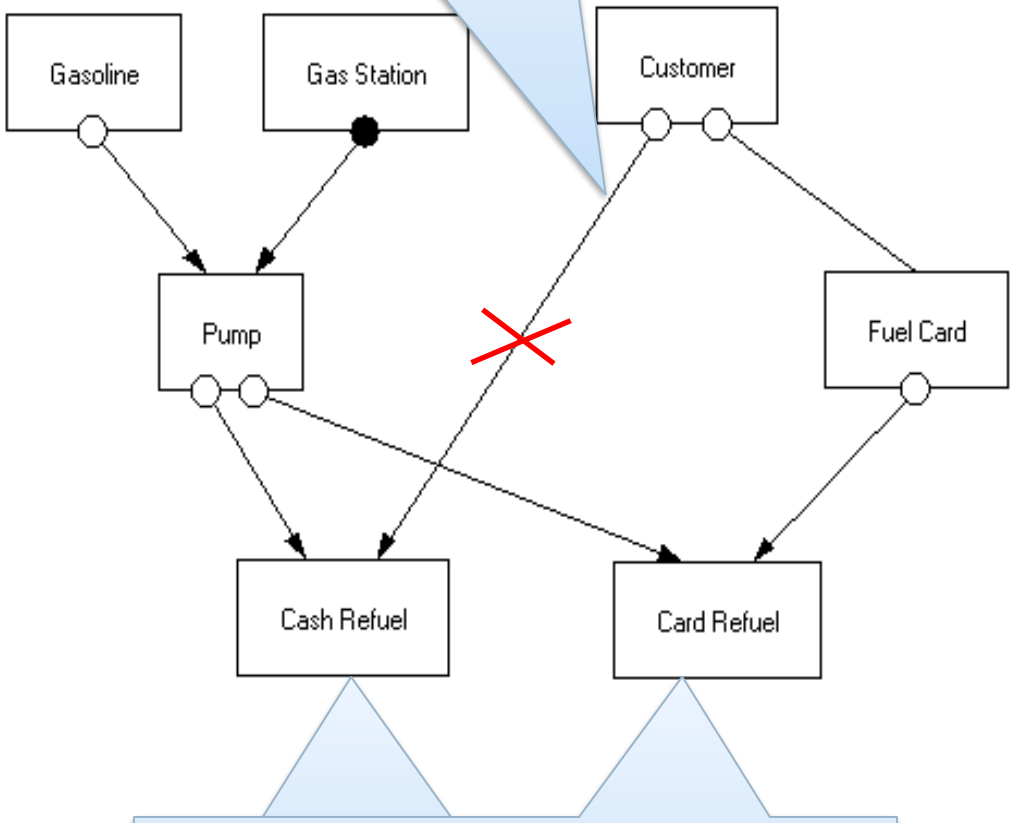
What is on the invoice? which refuel turn?

Lifecycle of Reservoir vs. Lifecycle or Pump

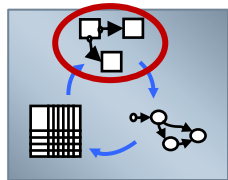


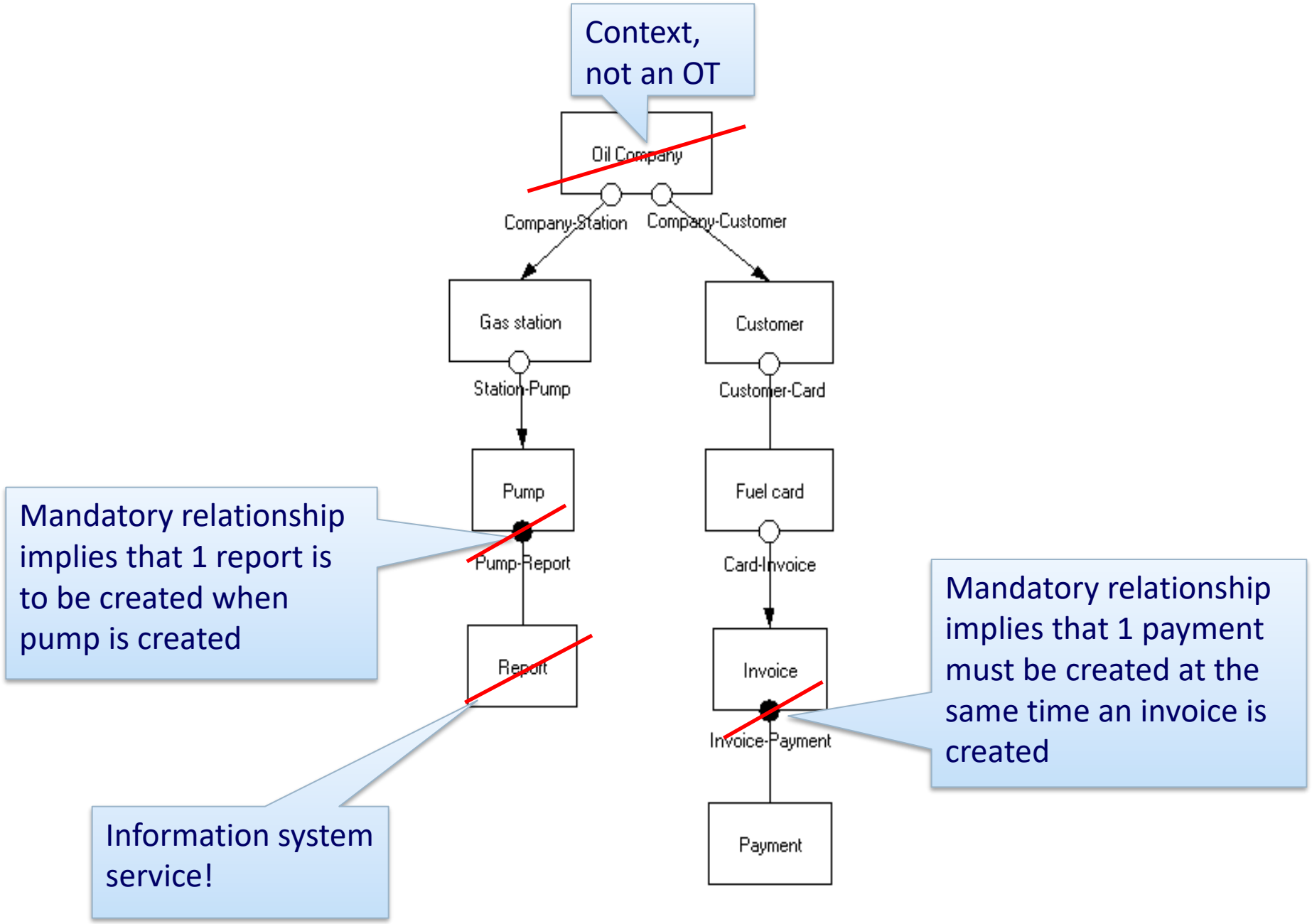

Information system service!
daily report
company
pump
fuel card
customers
gas station
discounts
cash
monthly invoice
turnover
Who does the invoice belong to?
Is Discount an object type dependent on card or is it a calculation rule in Invoice ?
Something you can calculate ➔ calculated attribute of e.g. Gas Station

By removing this dependency we allow anonymous cash refuel

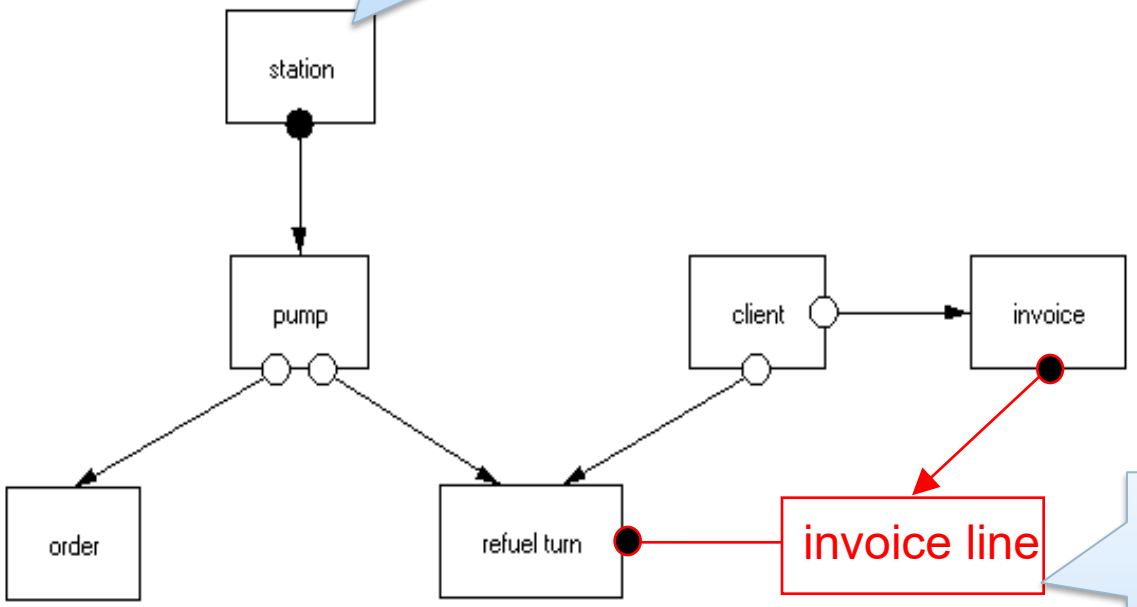


Later on, we might consider the use of Inheritance to model common aspects. See Chapter 8.




Context, not an OT
Oil Company
Company-Station
Company-Customer
Gas station
Customer
Station-Pump
Customer-Card
Pump
Fuel card
Pump-Report
Card-Invoice
Report
Invoice
Invoice-Payment
Payment
Mandatory relationship implies that 1 report is to be created when pump is created
Information system service!
Mandatory relationship implies that 1 payment must be created at the same time an invoice is created

Mandatory? Is it really necessary for a station to have at least one pump to exist ? Think of the consequences of this constraint in terms of user friendliness of the resulting information system.



This object type establishes the link between the invoice and the refuel turns that are on the invoice.