



FC-MD

Class diagram x Sequence diagrams

Contents

- ▶ 1 OORT Concepts
- 2 Sequence Diagram x Use cases
- 3 Class Diagram x Sequence Diagram

- Example design: Gas Station Control System (GSCS)
 - **Recall our description of the system:**
 - “... The gas station allows customers to purchase gas (self-service) or to pay for maintenance work done on their cars. Some local businesses may have billing accounts set up so that the business is sent a monthly bill, rather than paying for each transaction at the time of purchase. There will always be a cashier on-duty at the gas station to accept cash payments or perform system maintenance, as necessary.”
 - The requirements in this excerpt “...concern how the system receives payment from the customer. A customer has the option to be billed automatically at the time of purchase, or to be sent a monthly bill and pay at that time. Customers can always pay via cash or credit card.”

Slide 1



FC-MD

OORT Concepts

Contents

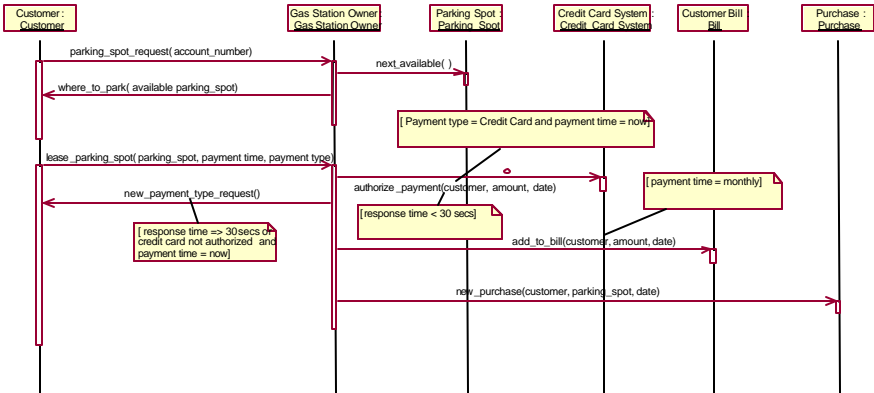
- ▶ 1 OORT Concepts
- 2 Sequence Diagram x Use cases
- 3 Class Diagram x Sequence Diagram

- Defining levels of functionality
 - **Functionality:** *Functionality* describes the behavior of the system. Typically, functionality is described from the user's point of view.
 - **Service:** Services are low-level actions performed by the system; they are the “atomic units” out of which system functionalities are composed. Users do not typically consider services an end in themselves; rather, services are the steps by which some larger goal or functionality is achieved.
 - **Message:** *Messages* are the very lowest-level behaviors out of which system services and, in turn, functionalities are composed. They represent the communication between objects that work together to implement system behavior. Messages may be shown on sequence diagrams and must have associated class behaviors.

Slide 2

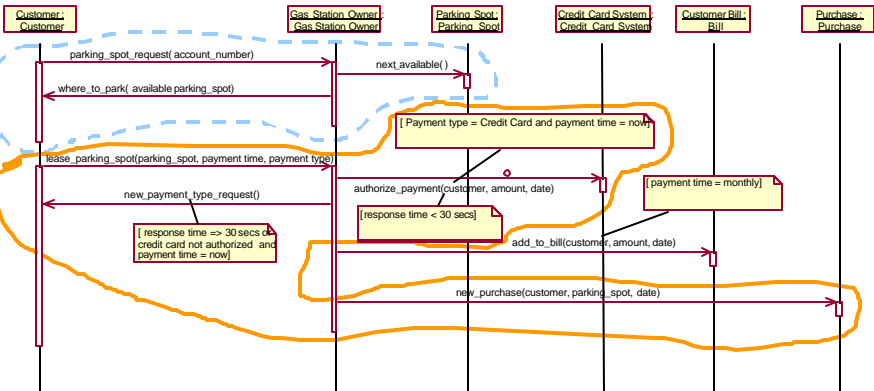
OORT Concepts

- Example: Abstracting messages to services (to functionality) in gas station



OORT Concepts

- Example: Abstracting messages to services (to functionality) in gas station





FC-MD

OORT Concepts

Contents

- ▶ 1 OORT Concepts
- 2 Sequence Diagram x Use cases
- 3 Class Diagram x Sequence Diagram

- Constraints/Conditions
 - Condition: describes what must be true for the functionality to be executed.
 - Constraint: must be always be true for system functionality

Slide 5



FC-MD

OORT Concepts

Contents

- ▶ 1 OORT Concepts
- 2 Sequence Diagram x Use cases
- 3 Class Diagram x Sequence Diagram

- Example of identifying constraints/conditions in gas station
 - If the time of payment is the same as the purchase time and Customer decides to pay by Credit Card then Credit Card system should be used. The Customer can only wait for 30 seconds for the authorization process otherwise this payment should be made by cash or personal check to avoid other Customers waiting on the lane. The Gas Station Owner should ask the Customer for a new payment type.

Slide 6



FC-MD

OORT Concepts

Contents

1 OORT Concepts

2 Sequence Diagram x Use cases

3 Class Diagram x Sequence Diagram

- Example of identifying constraints/conditions in gas station

– If the time of payment is the same as the purchase time and Customer decides to pay by Credit Card then Credit Card/system should be used. The Customer can only wait for 30 seconds for the authorization process otherwise this payment should be made by cash or personal check to avoid other Customers waiting on the lane. The Gas Station Owner should ask the Customer for a new payment type.

Condition

Constraint

Slide 7



FC-MD

OORT Concepts

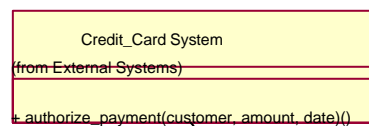
Contents

1 OORT Concepts

2 Sequence Diagram x Use cases

3 Class Diagram x Sequence Diagram

- Example of checking constraints/conditions are fulfilled, in gas station



[response time should be less than 30 secs for all Credit Card Systems]

Slide 8



FC-MD

Sequence diagram x Use cases

Contents

- 1 OORT Concepts
- 2 Sequence Diagram x Use cases
- 3 Class Diagram x Sequence Diagram

- The components:
 - A *use case* that describes important concepts of the system and the services it provides.
 - One or more *sequence diagrams* that describe the objects of a system and the services it provides.
 - The *class descriptions* of all classes in the sequence diagram.
- The goal:
 - To verify that sequence diagrams describe an appropriate combination of objects and messages that work to capture the functionality described by the use case.

Slide 9



FC-MD

Sequence diagram x Use cases

Contents

- 1 OORT Concepts
- 2 Sequence Diagram x Use cases
- 3 Class Diagram x Sequence Diagram

- **Step 1**
 - **Inputs:** Use case.
 - **Outputs:** System concepts; Services provided by system; Data necessary for achieving services.
 - **General instructions:** Identify the functionality described by a use case, and important concepts of the system that are necessary to achieve that functionality.
 - **Specific instructions:**
 - Find system concepts in the use case.
 - For each concept, identify the services in which it is involved. Look for associated constraints and conditions.
 - Also identify any information or data that is required to be sent or received in order to achieve the services.

Slide 10



FC-MD

Sequence diagram x Use cases

Contents

- 1 OORT Concepts
- 2 Sequence Diagram x Use cases
- 3 Class Diagram x Sequence Diagram

- **Step 2**
 - **Inputs:** Use case, sequence diagram.
 - **Outputs:** System concepts; Services provided by system; Data exchanged between objects.
 - **General instructions:** Identify and inspect the related sequence diagrams, to identify if the corresponding functionality is described accurately and whether behaviors and data are represented in the right order.
 - **Specific instructions:**
 - For each sequence diagram, identify the system objects.
 - Identify the *services* described by the sequence diagrams.
 - Identify the information (or data) that is exchanged between system classes.

Slide 11



FC-MD

Sequence diagram x Use cases

Contents

- 1 OORT Concepts
- 2 Sequence Diagram x Use cases
- 3 Class Diagram x Sequence Diagram

- **Step 3**
 - **Inputs:** Use case, sequence diagram.
 - **Outputs:** Discrepancy reports.
 - **General instructions:** Compare the marked-up diagrams to determine whether they represent the same domain concepts.
 - **Specific instructions:**
 - For each of identified concepts on the use case, search the sequence diagram to see if it is represented.
If not => omission.
 - For each object on the sequence diagram search the use cases to see if it is represented.
If not => extraneous info.

Slide 12



FC-MD

Sequence diagram x Use cases

Contents

- 1 OORT Concepts
- 2 Sequence Diagram x Use cases
- 3 Class Diagram x Sequence Diagram

• Step 3 (cont.)

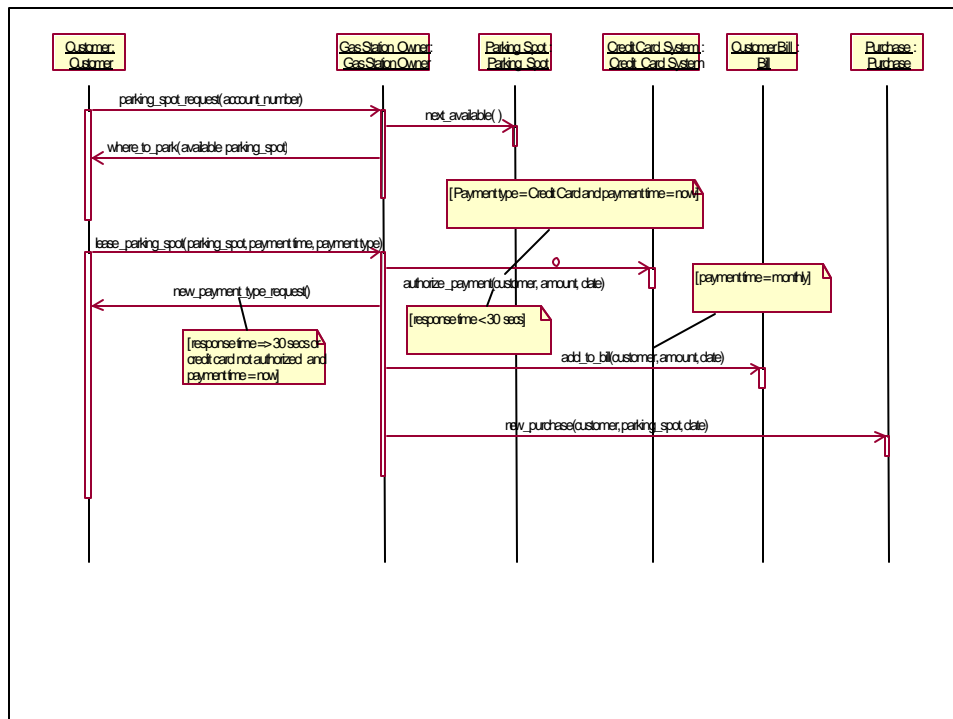
→ Identify the services described by the sequence diagram, and compare them with the description used on the use case.

Do the classes exchange messages in the same specified order? Are all transported data in the right message? Is data being sent between the right two classes? => incorrect fact

→ Are all the constraints and conditions from the use case being observed in this sequence diagram?

Were the constraints observed? Was all of the behavior and data on the sequence diagram directly concerned with the use-case? => incorrect fact

Slide 13





FC-MD

Class diagram x Sequence diagrams

Contents

- 1 OORT Concepts
- 2 Sequence Diagram x Use cases
- 3 Class Diagram x Sequence Diagram

- The components:
 - A *class diagram* (possibly divided into packages) that describes the classes of a system and how they are associated.
 - *Sequence diagrams* that describe the classes, objects, and possibly actors of a system and how they collaborate to capture services of the system.
- The goal:
 - To verify that the class diagram for the system describes classes and their relationships in such a way that the behaviors specified in the sequence diagrams are correctly captured.

Slide 15



FC-MD

Class diagram x Sequence diagrams

Contents

- 1 OORT Concepts
- 2 Sequence Diagram x Use cases
- 3 Class Diagram x Sequence Diagram

- **Step 1**
 - **Inputs:** Sequence diagram.
 - **Outputs:** System objects; Services of the system; Conditions on the services.
 - **General instructions:** Take a sequence diagram and read it to understand the system services described and how the system should implement those services.
 - **Specific instructions:**
 - Identify the system objects and classes, and any actors.
 - Identify the information exchanged between objects (the horizontal arrows). If the information exchanged is very detailed, at the level of messages, you should abstract several messages together to understand the services they work to provide.
 - Identify any constraints on the messages and services.

Slide 16



FC-MD

Class diagram x Sequence diagrams

Contents

- 1 OORT Concepts
- 2 Sequence Diagram x Use cases
- 3 Class Diagram x Sequence Diagram

• Step 2

- **Inputs:** Sequence diagrams, with objects, services, and constraints marked; Class diagrams.
- **Outputs:** Discrepancy reports.
- **General instructions:** Identify and inspect the related class diagrams, to identify if the corresponding system objects are described accurately.
- **Specific instructions:**
 - Verify that every object, class, and actor used in the sequence diagram is represented by a concrete class in a class diagram.
 - If a class or object or actor cannot be found on the class diagram => inconsistency.**

Slide 17



FC-MD

Class diagram x Sequence diagrams

Contents

- 1 OORT Concepts
- 2 Sequence Diagram x Use cases
- 3 Class Diagram x Sequence Diagram

• Step 2 (cont.)

- Verify that for every service or message on the sequence diagram, there is a corresponding behavior on the class diagram.
Verify that there are class behaviors in the class diagram that encapsulate the higher-level services provided by the sequence diagram.
Verify that for each service, the messages described by the sequence diagram are sufficient to achieve that service.
- Is there a message on the sequence diagram for which the receiving class does not contain an appropriate behavior on the class diagram? => inconsistency**
- Are there appropriate behaviors for the system services? => omission**
- Is there an association on the class diagram between the two classes between which the message is sent? => omission**

Slide 18



FC-MD

Class diagram x Sequence diagrams

Contents

- 1 OORT Concepts
- 2 Sequence Diagram x Use cases
- 3 Class Diagram x Sequence Diagram

- **Step 2 (cont.)**

→ Verify that the constraints identified in the sequence diagram can be fulfilled according to the class diagram.

Could you find the data but they do not conform to the behavior arguments? => inconsistency

Could you find the constraints but they do not completely agree in both documents? => inconsistency

Slide 19



FC-MD

Class diagram x Sequence diagrams

Contents

- 1 OORT Concepts
- 2 Sequence Diagram x Use cases
- 3 Class Diagram x Sequence Diagram

- **Step 2 (cont.)**

→ For each class, message, and data identified above, think about whether, *based on your previous experience*, it results in a reasonable design.

Does it make sense for the class to receive this message with these data? Could you verify if the constraints are feasible? Are all of the necessary attributes defined? => incorrect facts

For the classes specified in the sequence diagram, do the behaviors and attributes specified for them on the class diagram make sense? Is the *name* of the class appropriate for the domain, and for its attributes and behaviors? Are the relationships with other classes appropriate? Are the relationships of the right *type*? => incorrect facts

Slide 20