Lecture 4:

Object and Class Structuring

H. Gomaa, Chapter 8 - *Software Modeling and Design*, Cambridge University Press, 2011

1

1

How do you decide classes for an OO program?

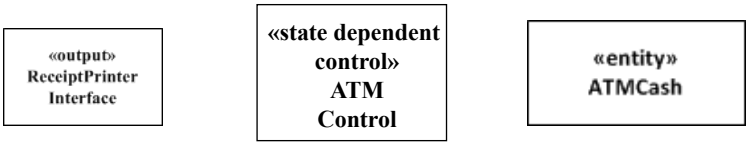
What are the criteria to decide classes?

2

2

Object and Class Structuring Criteria

- Objective
  - Determine all software objects and classes in system
    - Use Object and Class Structuring Criteria
    - Guidelines for identifying objects and classes
- Structuring criteria depicted using stereotypes
- **Stereotype**
  - Defines role of class or object in application
  - Depicted using angle brackets, e.g.,
    - «entity», «output», «control»
- Class has same stereotype as objects instantiated from it



3

3

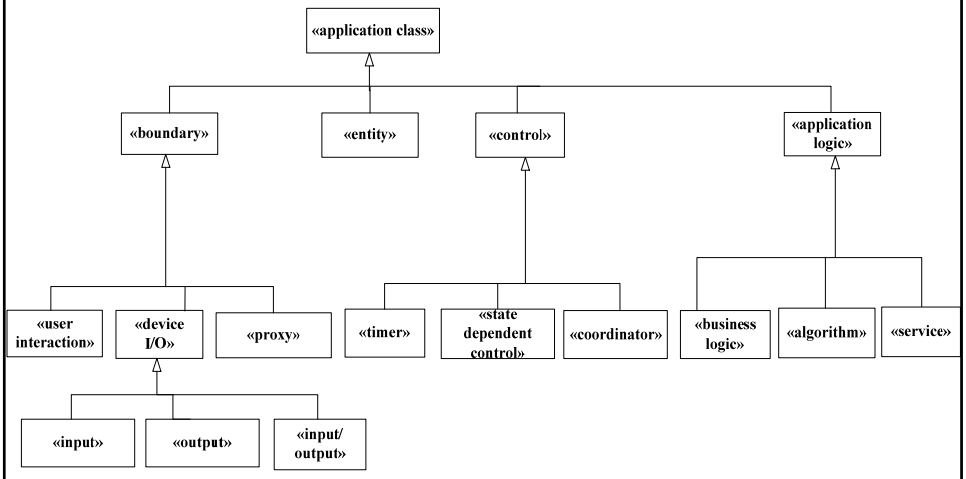
System of Classification

- Objects and classes are categorized
  - A **category** specifically defined division in a system of classification
  - Group together classes with similar characteristics
  - Categorization of application classes by stereotype (Fig. 8.1)

os-4

4

Figure 8.1: Classification of application classes using stereotypes



5

5

Object Structuring Criteria

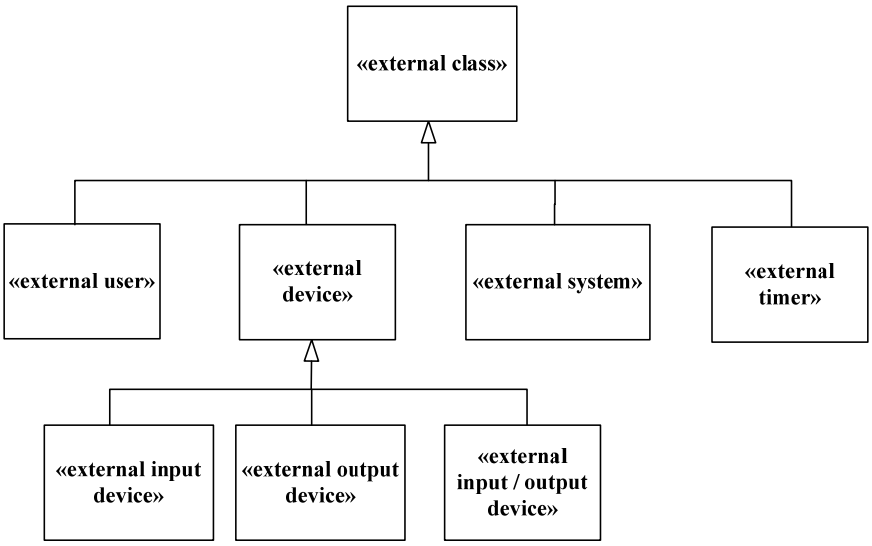
- Boundary object
  - Software object that interfaces to and communicates with external environment
  - Each software boundary object interfaces to an external (real-world) object
    - User interaction object
    - Device I/O object
    - Proxy object
- For each software boundary object
  - There is a corresponding external object

«user interaction» :Operator Interaction

6

6

Figure 7.22 Classification of external classes using stereotypes



7

7

User Interaction Object

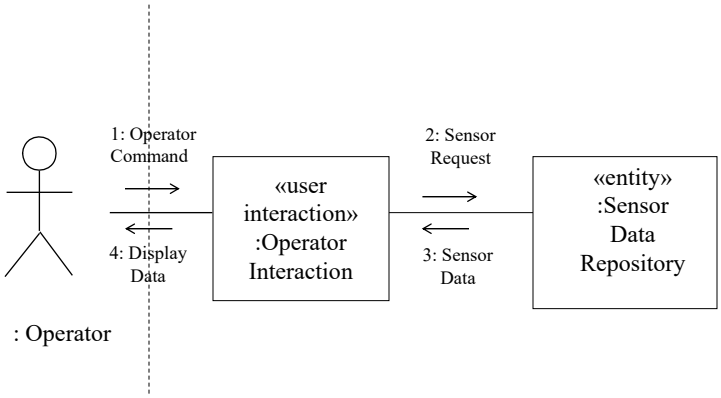
- Interfaces to and interacts with a human user
  - Via standard I/O devices
    - keyboard, visual display, mouse
  - Support simple or complex user interfaces
    - Command line interface
    - Graphical user interface (GUI)
  - Describes how the object interacts with its neighboring objects

«user interaction» :Operator Interaction

8

8

Figure 8.2 Example of user interaction object



Note: The dashed line for the system boundary is for illustrative purposes only and does not conform to the UML notation

9

9

Proxy Object

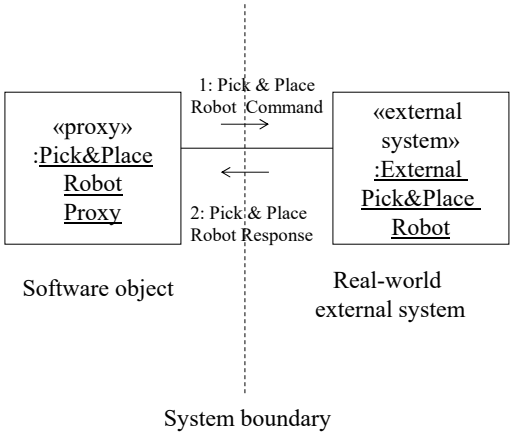
- Software object that interfaces to and communicates with an external system or subsystem
- Hides details of **how** to communicate with external system
  - E.g., Robot Proxy
  - Interfaces to external (real-world) robot

«proxy» :Pick&Place Robot Proxy

10

10

Figure 8.3 Example of proxy object



Note: The dashed line for the system boundary is for illustrative purposes only and does not conform to the UML notation

11

11

Device I/O Boundary Object

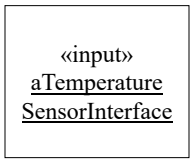
- Provides the software interface to a hardware I/O device
  - Receives input from and/or outputs to a hardware I/O device
- Device I/O Boundary Objects
  - Input object
  - Output object
  - I/O object

12

12

Device I/O Boundary Object
Input object

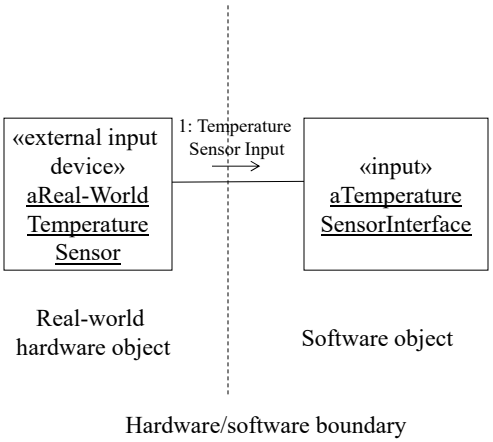
- Input object
  - Device I/O boundary object that receives input from an external input device
  - E.g., Sensor Interface
    - Input object receives sensor input from external hardware object



13

13

Figure 8.4 Example of input object



Note: The dashed line for the hardware/software boundary is for illustrative purposes only and does not conform to the UML notation

14

14

Device I/O Boundary Object
Output object

- Output object
  - Device I/O boundary object that sends output to an external output device
  - E.g., Actuator Interface
    - Outputs to external output device

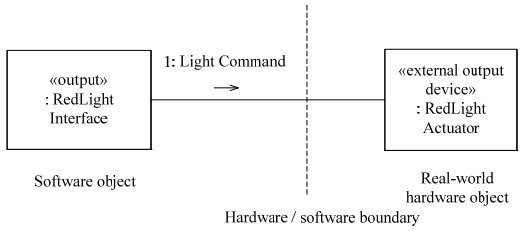


Figure 8.5 Example of output object

15

Device I/O Boundary Object
I/O object

- I/O (Input/Output) object
  - Device I/O boundary object that receives input from and sends output to an external I/O device
  - E.g., ATM Card Reader Interface

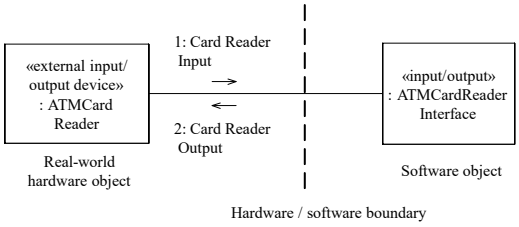


Figure 8.6 Example of I/O object

16

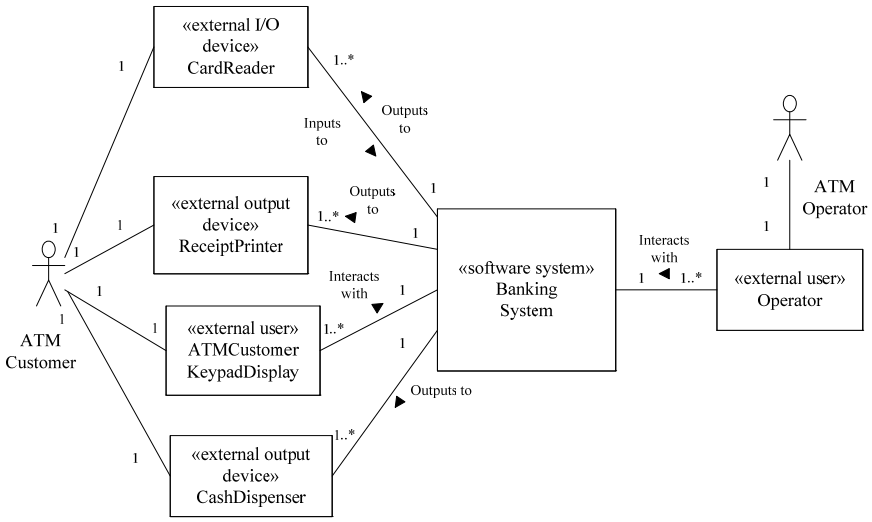
Depicting External Classes and Boundary Classes

- Start from system software context class diagram
  - Shows external classes
  - System (aggregate class)
- Each **external class** must interface to
  - software **boundary class**
- UML
  - Software system shown as aggregate class
  - External classes are outside the software system class
  - Boundary classes are inside the software system class

17

17

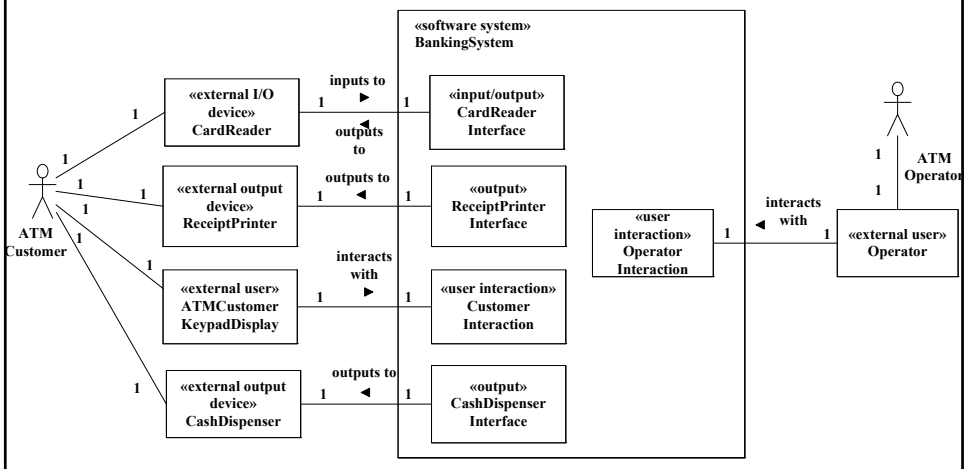
Figure 7.23 Banking System software context class diagram



18

18

Figure 8.7 Banking System external classes and software boundary classes



19

19

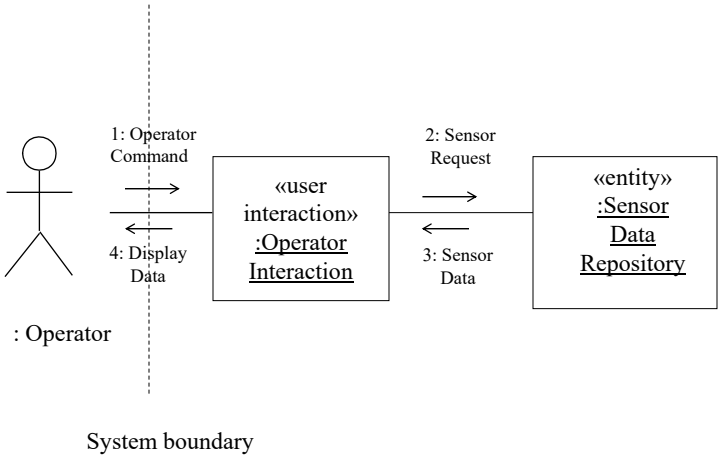
Entity Classes and Objects

- Entity objects
  - Software object that stores information
  - Determined during static modeling
    - Entity classes and relationships shown on class diagram

20

20

Figure 8.2 Example of entity object



Note: The dashed line for the system boundary is for illustrative purposes only and does not conform to the UML notation

21

21

Object Structuring Criteria

- Control objects
  - Provides overall coordination for the execution of a group of objects
  - Makes overall decisions
  - Decides when, and in what order, other objects participate in interaction sequence
    - Entity objects
    - Boundary objects

22

22

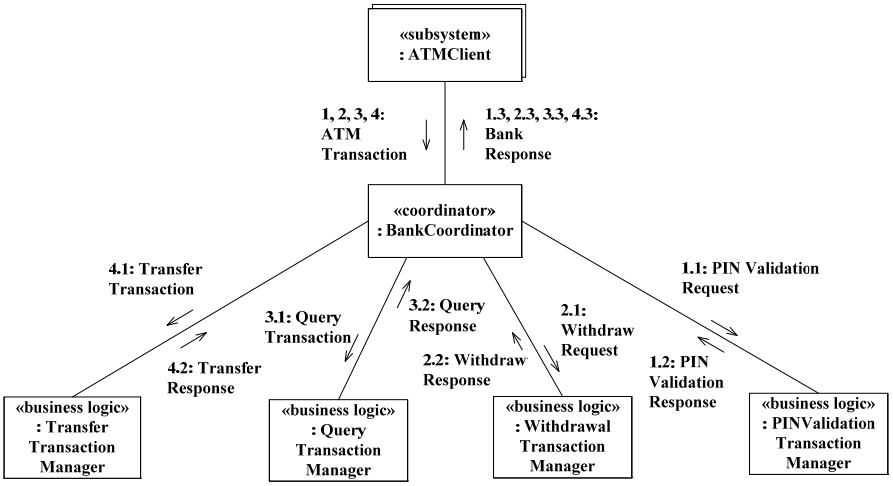
Object Structuring Criteria

- Control objects
  - Coordinator object
  - State dependent control object
  - Timer object

23

Figure 8.10 Coordinator Object

Coordinator object
• Decision making object
• Decision based on input received



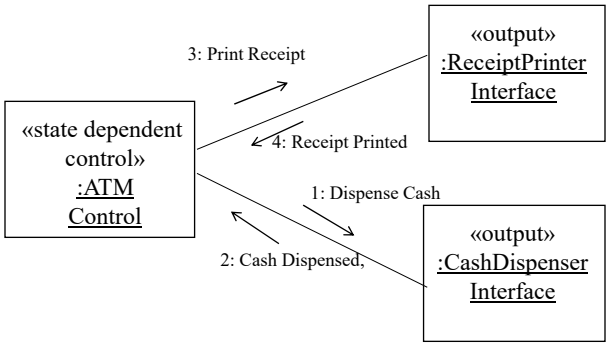
24

State dependent control object

- State dependent control object
  - Controls other objects
  - Defined by finite state machine or state transition table
  - Decision made based on
    - Input received AND Internal state

25

Figure 8.11 State dependent control object



26

Timer object

- Timer object
  - Activated periodically
  - Performs some action

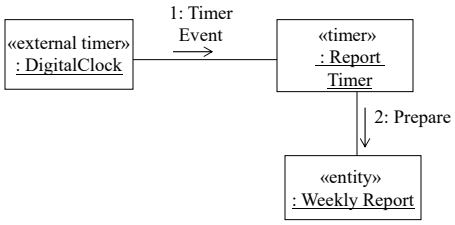


Figure 8.12 Timer object

27

27

Application Logic Objects

- Application Logic Object
  - Software object that contains details specific to an application
- Application Logic Objects
  - Business Logic Object
  - Algorithm Object
  - Service Object

28

28

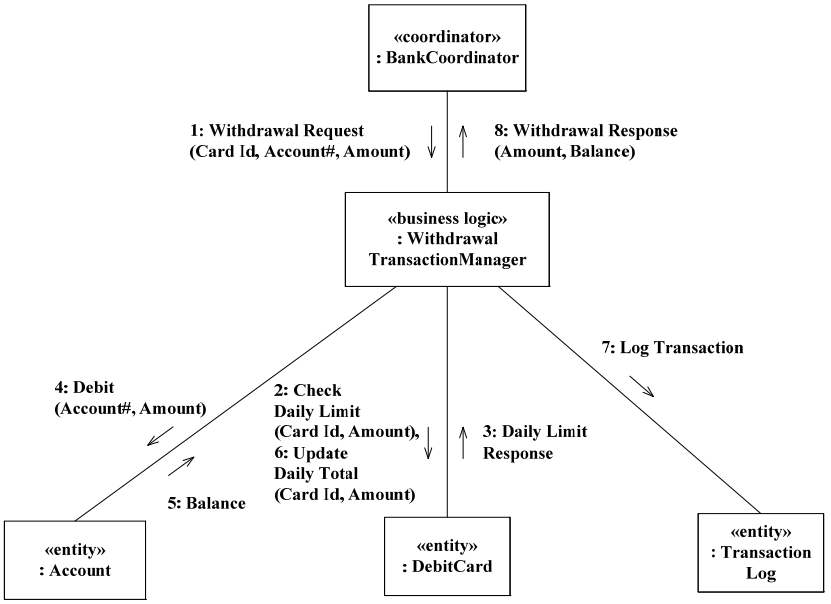
Business logic object

- Defines business specific application logic (rules)
- Usually accesses more than one entity object

29

29

Figure 8.13b Business logic object



30

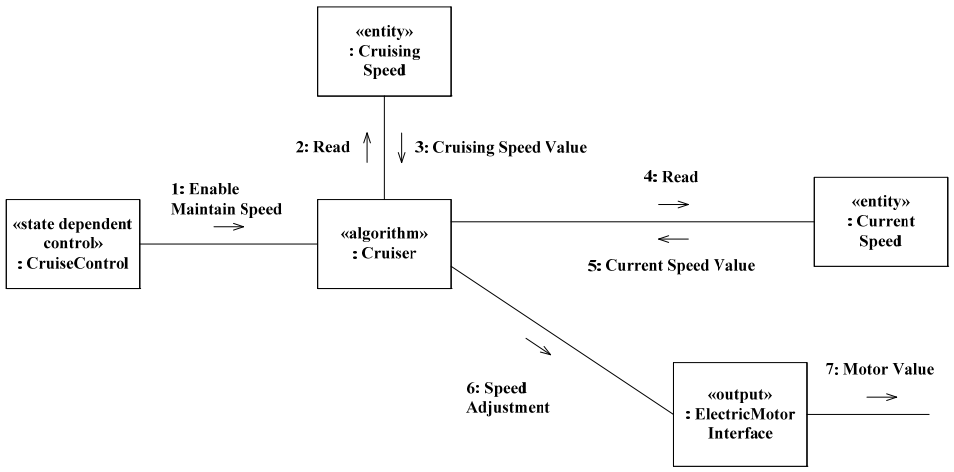
30

Algorithm object

- Encapsulates algorithm used in problem domain
- More usual in scientific, engineering, real-time domains

31

Figure 8.14 Example of algorithm object



32

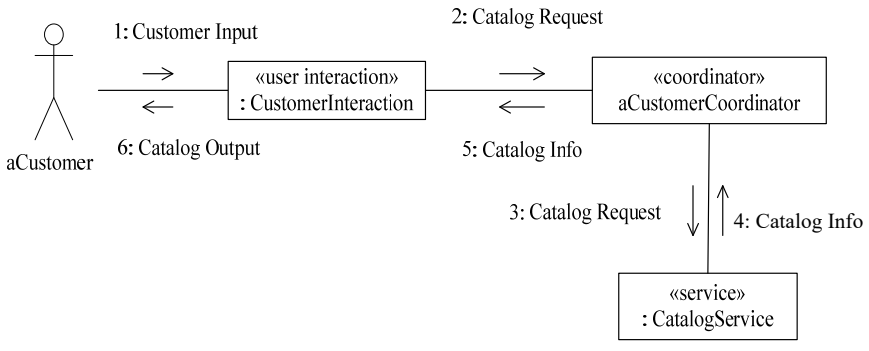
Service object

- Autonomous self-contained object
- Provides a service to other objects
- Responds to requests from client objects
- Operates independently of other services
  - Does not share entity objects with other services
  - E.g., Catalog is only accessed by Catalog Service

33

33

Figure 8.15 Example of Service object



34

34

Browse Catalog use case description

Use case name: Browse Catalog

Summary: Customer browses World Wide Web catalog, views various catalog items from the supplier's catalog, and selects items from the catalog.

Actor: Customer

Precondition: Customer browser is linked to supplier catalog Web site.

Main sequence:

1. Customer requests to browse catalog.
2. System displays catalog information to customer.
3. Customer selects items from catalog
4. System displays an itemized list containing each item description and price, as well as the total price.

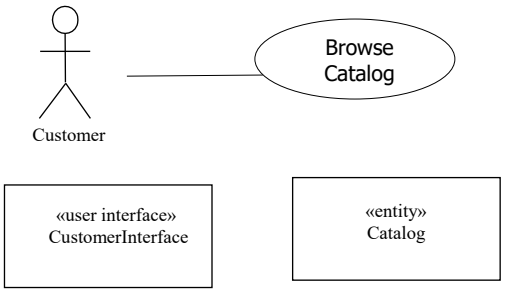
Alternative sequence:

Step 3: Customer does not select item and exits.

Postcondition: System has displayed a list of selected catalog items.

35

Classes for Browse Catalog use case



36

Make Order Request use case description

Use case name: Make Order Request

Summary: Customer enters an order request to purchase catalog items. The customer's credit card is checked for validity and sufficient credit to pay for the requested catalog items.

Actor: Customer, Bank

Precondition: Customer has selected one or more catalog items

Main sequence:

1. Customer provides order request and customer account Id to pay for purchase.
2. System retrieves customer account information, including the customer's credit card details.
3. System requests to a bank checking the customer's credit card for the purchase amount and, if approved, creates a credit card purchase authorization number.
4. System creates a delivery order containing order details, customer Id, and credit card authorization number.
5. System confirms approval of purchase and displays order information to customer.
6. System sends email confirmation to customer.

37

Make Order Request use case description

Alternative sequences:

Step 2: If customer does not have an account, the system prompts the customer to provide information in order to create a new account. The customer can either enter the account information or cancel the order.

Step 3: If authorization of the customer's credit card is denied (e.g., invalid credit card or insufficient funds in the customer's credit card account), the system prompts the customer to enter a different credit card number. The customer can either enter a different credit card number or cancel the order.

Postcondition: Customer has purchased items.

38

Classes for Make Order Request use case

«user interaction» Customer Interface

«entity» Customer Account

«entity» Delivery Order

«entity» Email

«proxy» Bank Proxy