The University of Iowa

CS:2820 (22C:22)

Object-Oriented Software Development

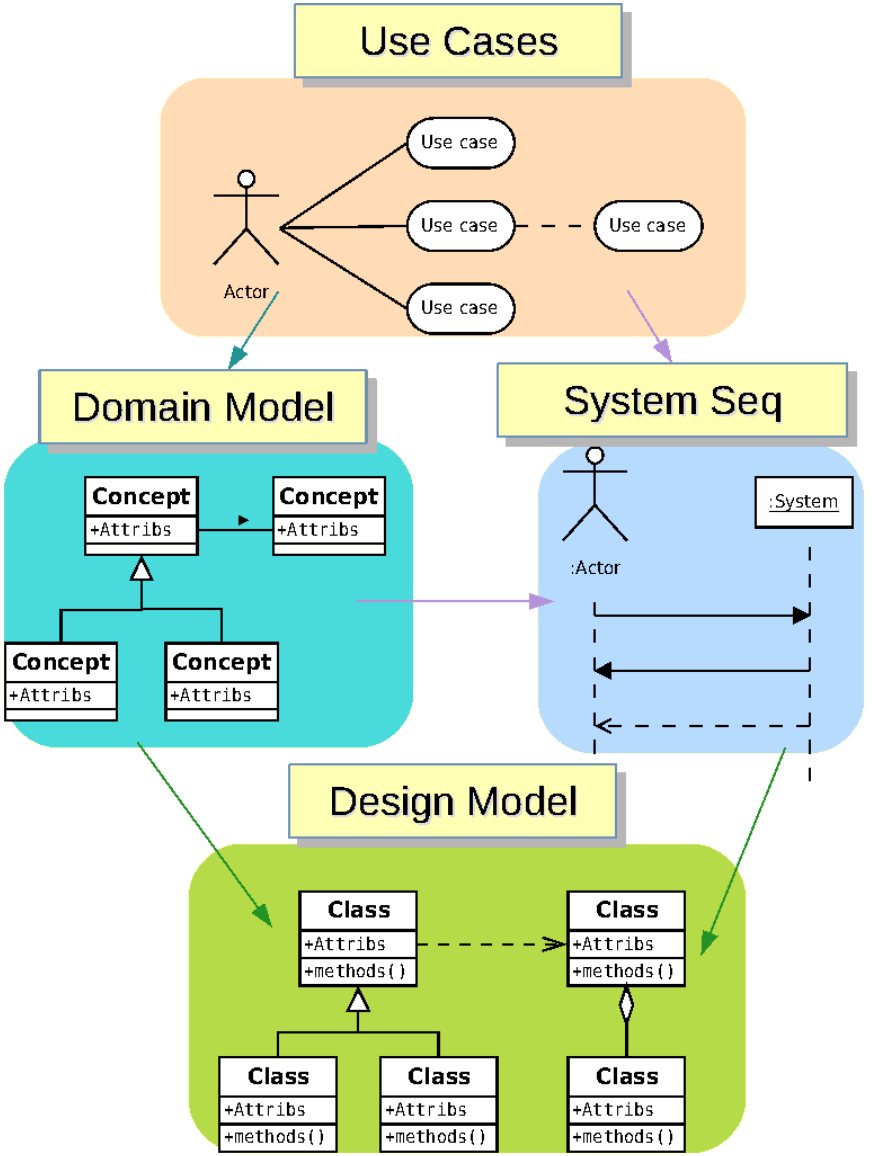
Spring 2015

Operation Contracts

by
Mauricio Monsalve

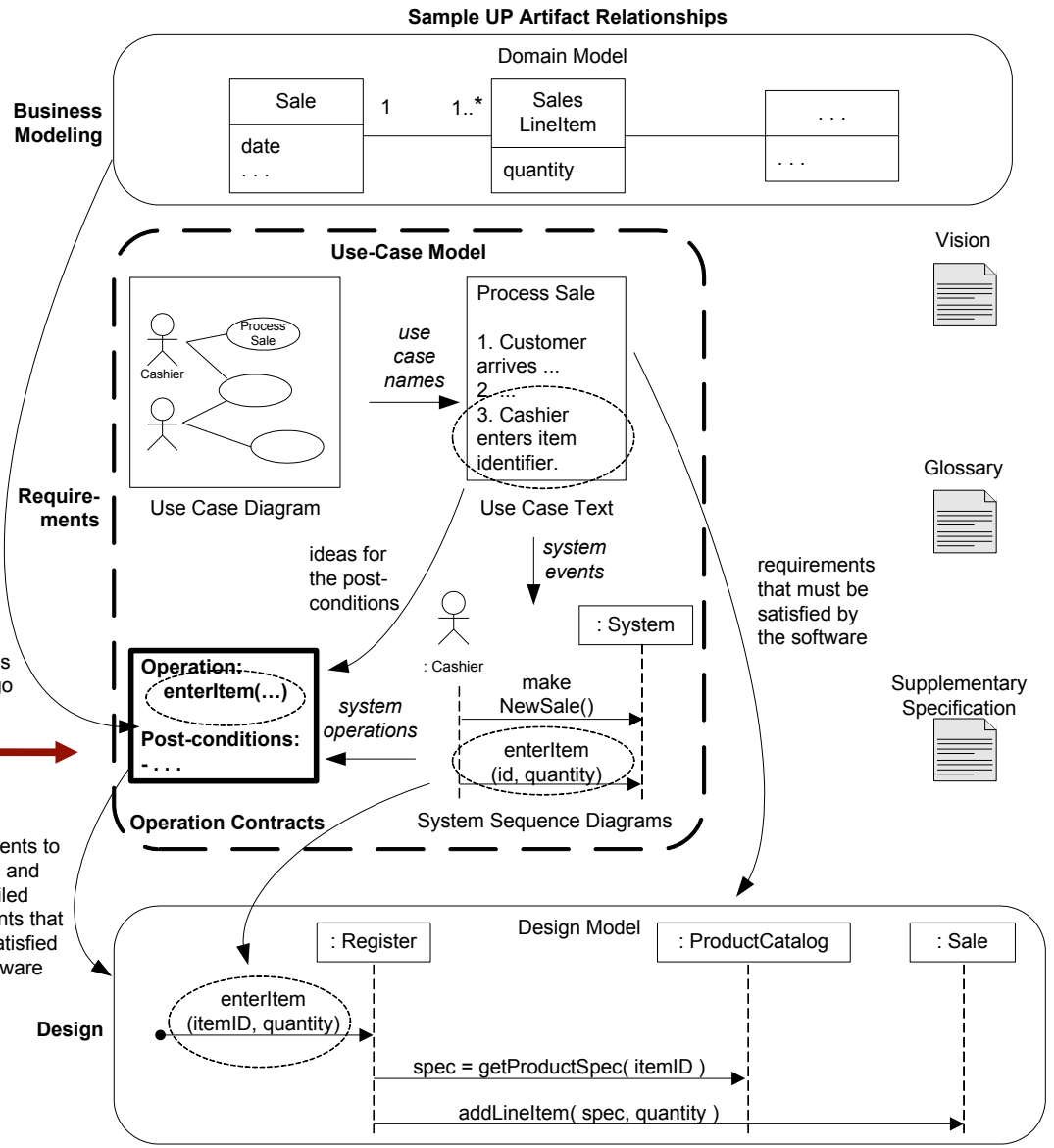
Design road

- We have described:
  - Use Cases
  - Domain Model
  - System Sequence Diagrams
- We now describe Operation Contracts
- Afterwards, we go into the Design Model



Design road

This is where we stand right now



Operation Contracts

- Use Cases often fully describe the behavior of a system
- But they may not be enough
- Operation Contracts describe how the internal state of the concepts in the Domain Model may change
- Operation Contracts are described in terms of preconditions and postconditions

Operation Contracts

This is a sample OC for "enterItem"

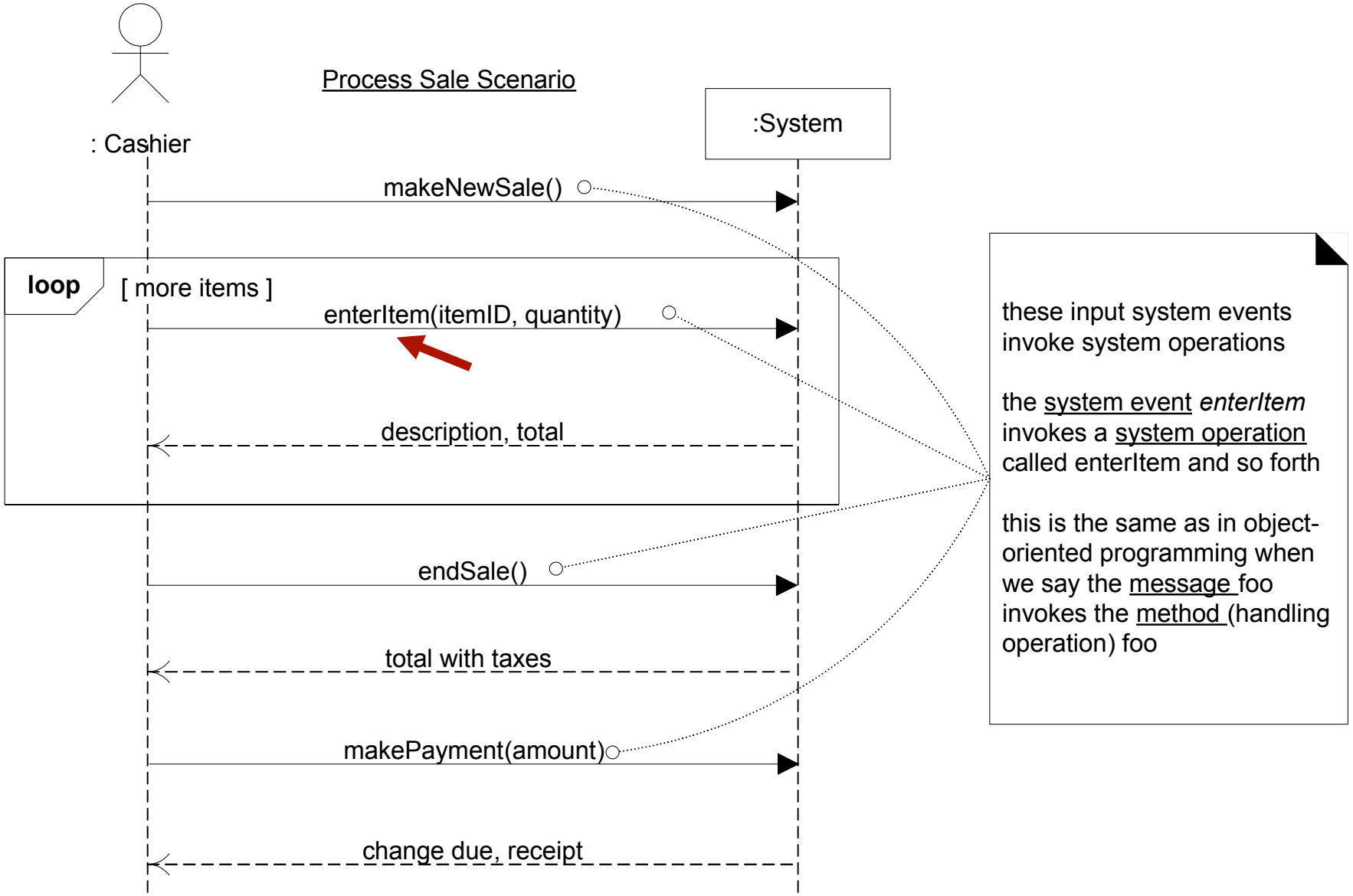
Contract CO2: enterItem

Operation:	enterItem(itemID: ItemID, quantity: integer)
Cross References:	Use Cases: Process Sale
Preconditions:	There is a sale underway.
Postconditions:	– A SalesLineItem instance sli was created *(instance creation)*. – sli was associated with the current Sale *(association formed)*. – sli.quantity became quantity *(attribute modification).* – sli was associated with a ProductDescription, based on itemID match *(association formed).*

Operation Contracts

- Operation Contracts are defined in terms of system operations
  - Operations (say, *methods*) that the system offers as a *whole*
    - The system is still a black box at this stage
- The System Sequence Diagrams show system events
  - I.e., the SSD's messages
- System operations handle system events

Writing Operation Contracts



Writing Operation Contracts

Contract CO2: enterItem

Operation:	enterItem(itemID: ItemID, quantity: integer)
Cross References:	Use Cases: Process Sale
Preconditions:	There is a sale underway.
Postconditions:	– A SalesLineItem instance sli was created *(instance creation)*. – sli was associated with the current Sale *(association formed)*. – sli.quantity became quantity *(attribute modification).* – sli was associated with a ProductDescription, based on itemID match *(association formed).*

Writing Operation Contracts

- Operation—name of operation (parameters)
- Cross Reference—the Use Cases in which the OC occurs
- Preconditions—noteworthy assumptions about state of system or objects in DM before execution
- Postconditions—state of objects in DM *after* execution of operation

Postconditions

- Most important part of OCs!
- Include changes in state of DM
- Book uses categories (note that the names are for reference only):
  - Instance creation or deletion
  - Association formed or broken
  - Attribute modification

Postconditions

- Write in past (passive voice?)
  - *A LineSaleItem was created*
- Readability first
- Common mistake—forgetting that instance creation often implies association formation, and similarly, that instance deletion often implies association breaking

Practical summary

- Identify system operations from the SSDs
- Identify *subtle* or *complex* system operations
- Construct a contract for each of the above; for postconditions, use the following categories
  - *Instance created-deleted*
  - *Attribute modified*
  - *Association formed-broken*

Credits

Notes and figures adapted from

Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development by C. Larman. 3rd edition. Prentice Hall/Pearson, 2005.