

System Sequence Diagrams

1

Overview

- What is System Sequence Diagram?
- UML Sequence Diagram
- Case Study: Simplified "Process Sale"

N. Meng, B. Ryder

2

2

System Sequence Diagram

- Definition
 - A picture that shows, *for a use case*, the events that external actors generate, their order, and inter-system events
 - Happy path + frequent/complex alternatives
- All systems are treated as a black box, focusing on **WHAT** instead of **HOW**

N. Meng, B. Ryder

3

3

Compared with Class Diagram

- Class Diagram describes the **static** structure of software
- Sequence Diagram describes the **dynamic** interactions between actors and the system

N. Meng, B. Ryder

4

4

Roles of SSDs

- Generated from inspection of a use case
 - Illustrate input and output events related to the system
 - Emphasize events cross the boundary between actors and systems
- Input to OOD

N. Meng, B. Ryder

5

5

UML Sequence Diagram

- A notation to illustrate actor interactions and operations initiated by them
- Only the interaction between users and the system is modeled in system sequence diagram

N. Meng, B. Ryder

6

6

Legends: Lifeline

Smith: Student
 |
 |
 |

- Definition
 - Represents either actors or systems that participate by either sending or receiving messages (events)
- Naming convention
 - Instance Name: Class Name
 - Other variants

: Student

 |
 |
 |

Smith

 |
 |
 |

Use a named object only when:

- You refer to it now and then
- You don't mention its type
- There are anonymous same-typed objects to distinguish from

N. Meng, B. Ryder 7

7

Legends: Note, Stereotype, Messages

- Stereotypes can be added to objects to indicate their roles

«actor»

user

 |
 |
 |

«boundary»

editDialog

 |
 |
 |

«control»

editController

 |
 |
 |

«entity»

book

 |
 |
 |

«database»

sales

 |
 |
 |

sender

receiver

sendMessage

returnMessage

- Messages represent events

N. Meng, B. Ryder 8

8

Legends: Combined Fragment

- Definition
 - An interaction fragment which defines a combination of messages between objects
 - Interaction operator(relation) + interaction operands (messages) + interaction constraints (guards)
 - Operators
 - loop - iteration
 - alt - alternatives
 - opt - option

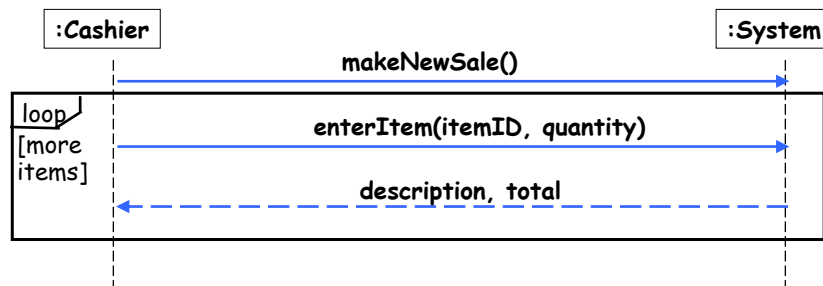
N. Meng, B. Ryder

9

9

Example: Simplified "Process Sale"

1. **Cashier** starts a new sale
 2. **Cashier** enters item id
 3. **System** records sale line item and presents description and running total
- Repeat Steps 2-3 until done*



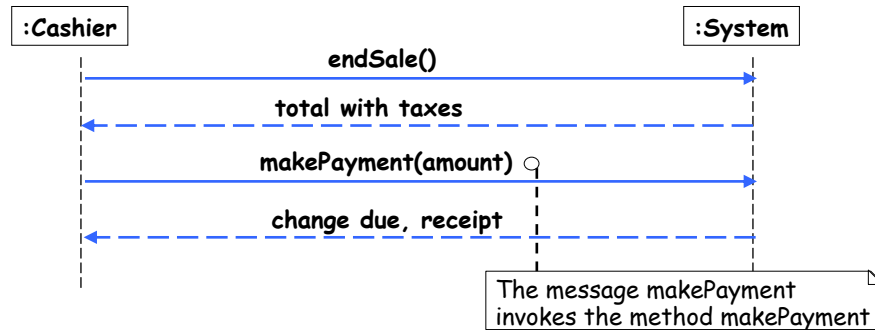
N. Meng, B. Ryder

10

10

Example cont.

4. **System** presents total with taxes calculated.
5. Customer pays and System handles payment



N. Meng, B. Ryder

11

11

Abstractions in SSDs

- **Events** and **return values** are abstractions
 - Independent of mechanism & representation
- *makePayment(amount)*
 - Shows **input info**
 - Looks like a method call, but is really an abstraction of an event
- **Name:** should capture the intent
 - Avoid specifying **implementation choices**
 - *enterItem(itemID)* is better than *scan(itemID)*

N. Meng, B. Ryder

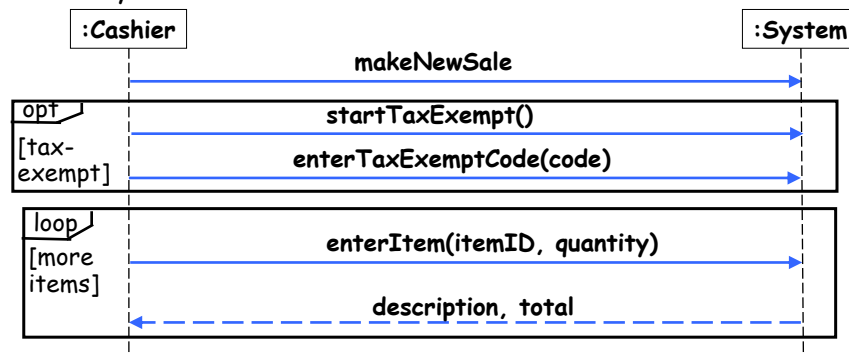
12

12

Alternative Scenario

1a. Customer tells Cashier they have a tax-exempt status (e.g., seniors, native people)

1. Cashier verifies, and then enters tax-exempt status code
2. System records status



N. Meng, B. Ryder

13

13

Homework: Withdraw Money from ATM

- Draw a sequence diagram for the use case description you turned in for HW1
 - Casual use case description
 - SSD
- Due: 10/28/2019 11:59pm

N. Meng, B. Ryder

14

14