

# **System Sequence Diagram (SSD)**

CSE 3311 & 5324

Christoph Csallner

University of Texas at Arlington (UTA)

Other undergrad UTA courses may already cover SSD

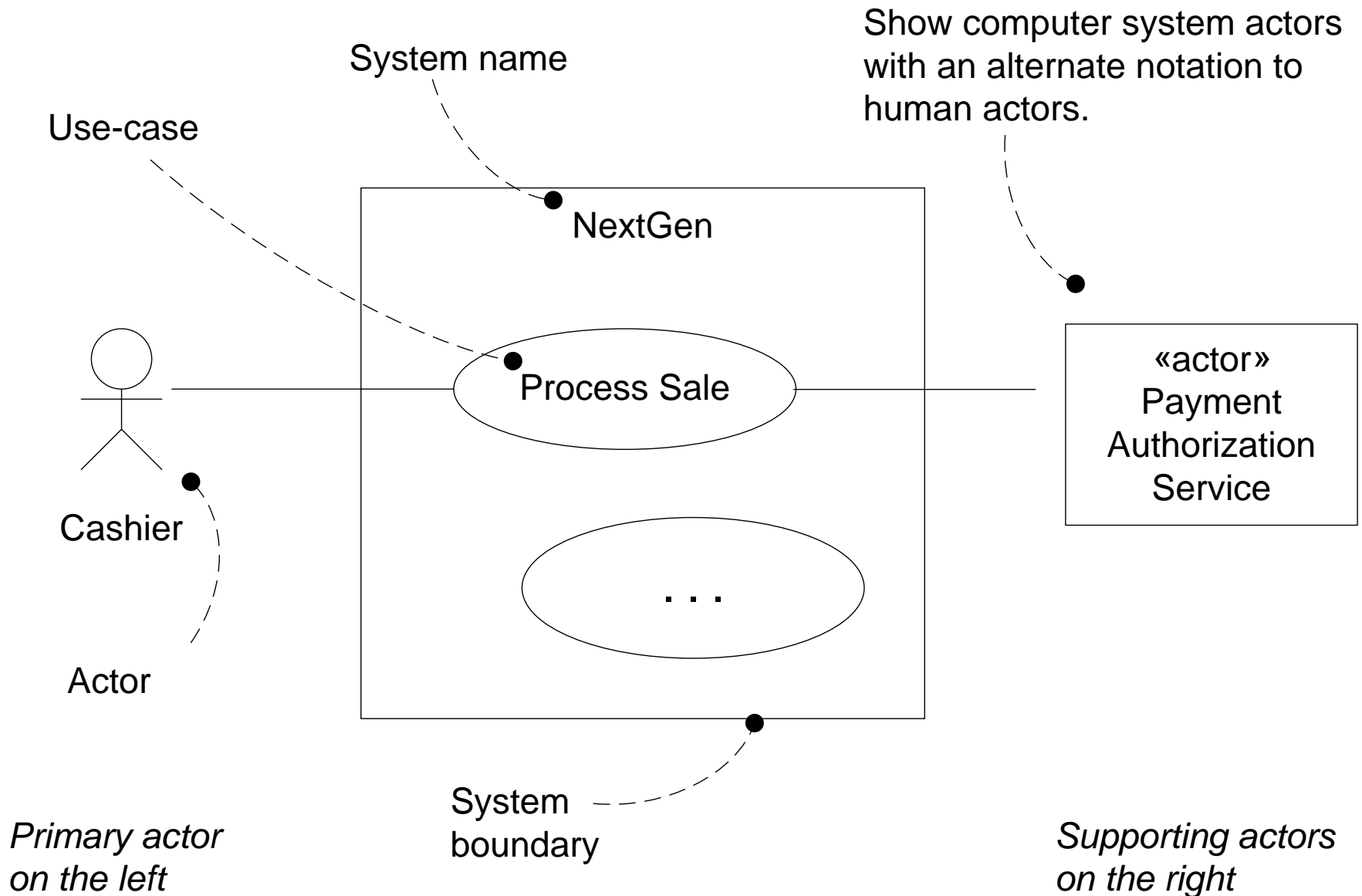
# Motivation

- SSD = System Sequence Diagrams
- CL, Chapter 10
- System Sequence Diagrams
  - **Are part of use-case modeling**
  - Are a set of UML sequence diagrams
  - Each sequence diagram rephrases an interesting use case scenario

# Review

- Use-case = Text story
- Main success scenario = Basic flow
  - Should be easy to understand, defer branches to:
- Error scenarios (extensions) = Alternate flows
  - Largest part of a use-case
- Scenario = Sequence of steps or actions
- Treat system as a single black box

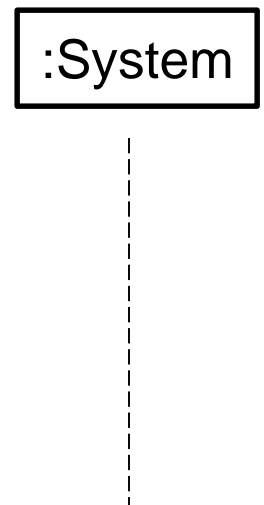
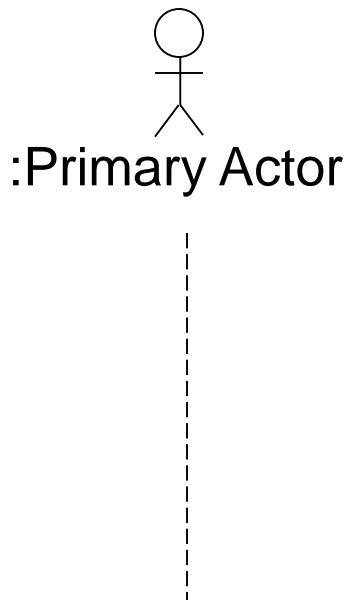
# Review: Use-Case Diagram



CL Figure 6.4

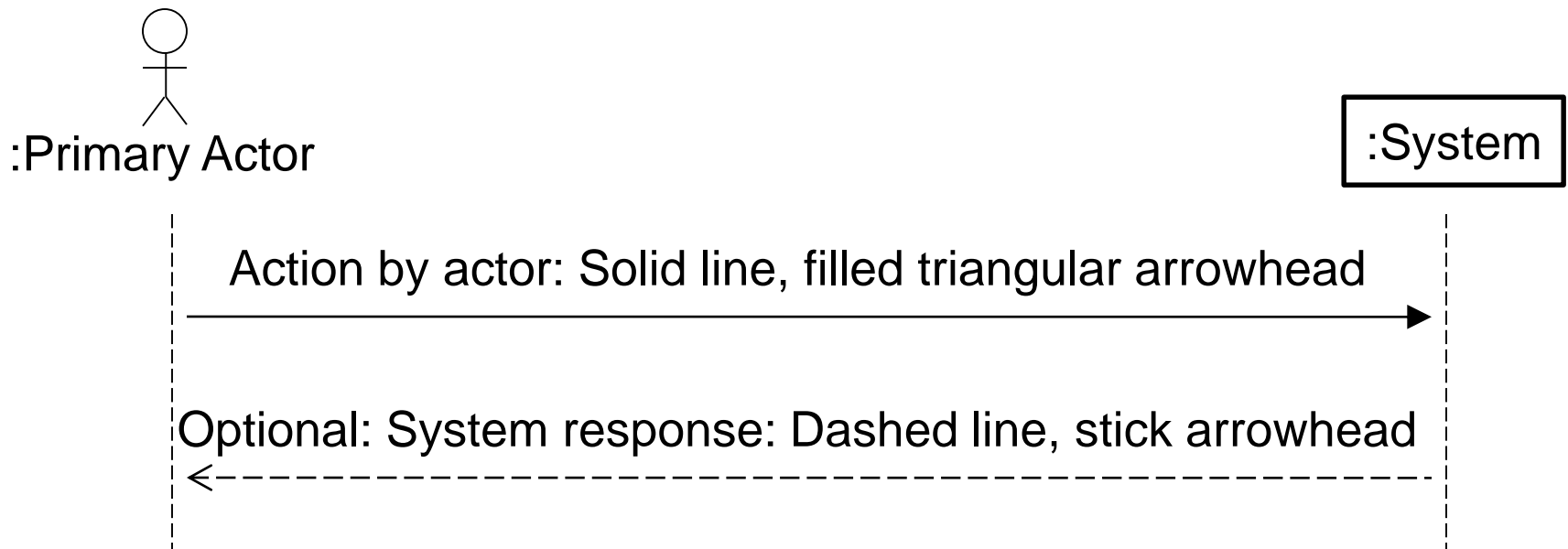
# System Sequence Diagram

- For each use-case: Pick an interesting scenario
  - Typically: The success scenario
- Map scenario to an UML sequence diagram
  - Actors ordered left to right in decreasing importance
  - Each actor: Vertical dashed line: timeline (“lifeline”)



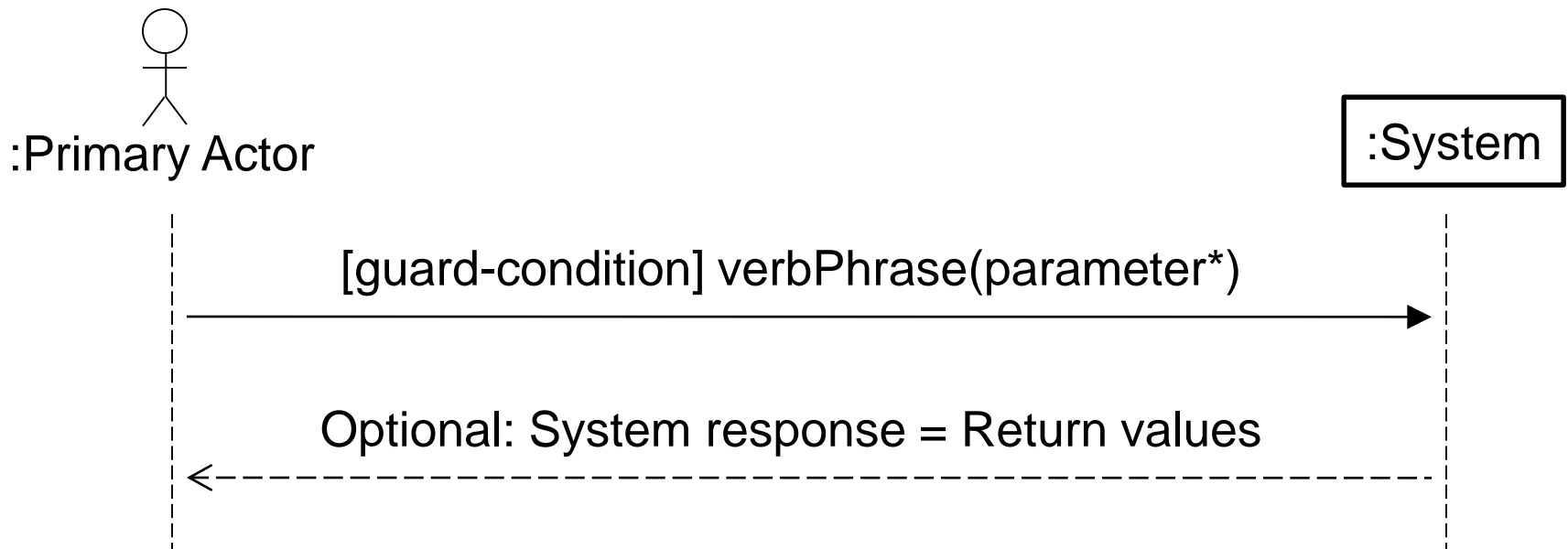
# Map Use-case Steps to Arrows

- Recall: Scenario = Sequence of steps or actions
  - Ping-pong between primary actor and system
  - (Action by primary actor, [system response])\*
- Step: Directed arrow between two lifelines

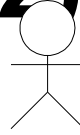


# Use-case Step as API Call

- Action by primary actor written as a verb phrase
  - Invent a good verb phrase
  - Written like a method signature
  - **System event (“public API” of system)**
  - Optional guard condition, default = true



# Example: POS (1/2)



: Cashier

!

Process Sale Scenario

:System

!

Simple cash-only *Process Sale* scenario:

1. Customer arrives at a POS checkout with goods and/or services to purchase.
2. Cashier starts a new sale.
3. Cashier enters item identifier.
4. System records sale line item and presents item description, price, and running total.  
Cashier repeats steps 3-4 until indicates done.
5. System presents total with taxes calculated.
6. Cashier tells Customer the total, and asks for payment.
7. Customer pays and System handles payment.

...



# Example: POS (2/2)

system as black box

the name could be "NextGenPOS" but "System" keeps it simple

the ":" and underline imply an instance, and are explained in a later chapter on sequence diagram notation in the UML

external actor to system

*Process Sale Scenario*

: Cashier

:System

a UML loop **interaction frame**, with a boolean **guard** expression

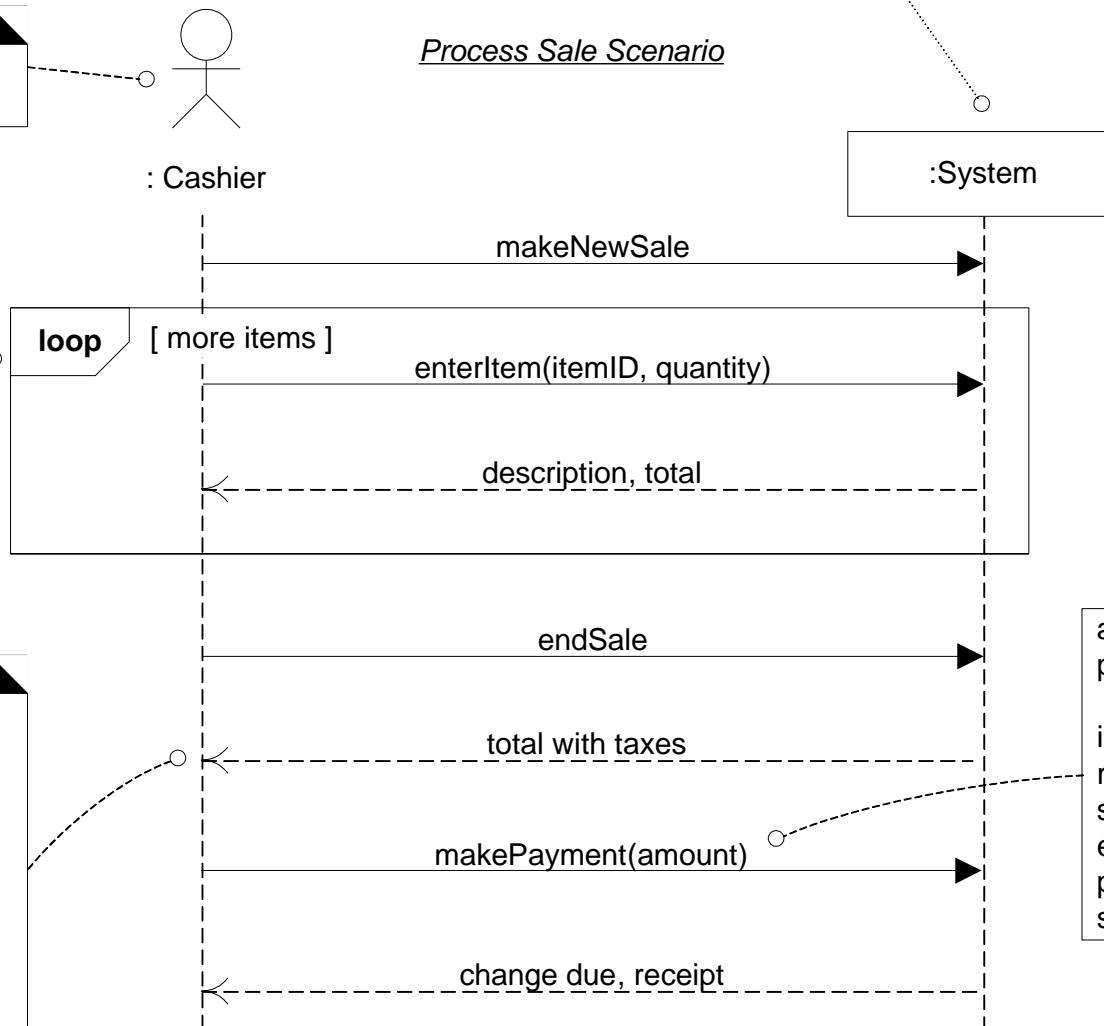
return value(s) associated with the previous message

an abstraction that ignores presentation and medium

the return line is optional if nothing is returned

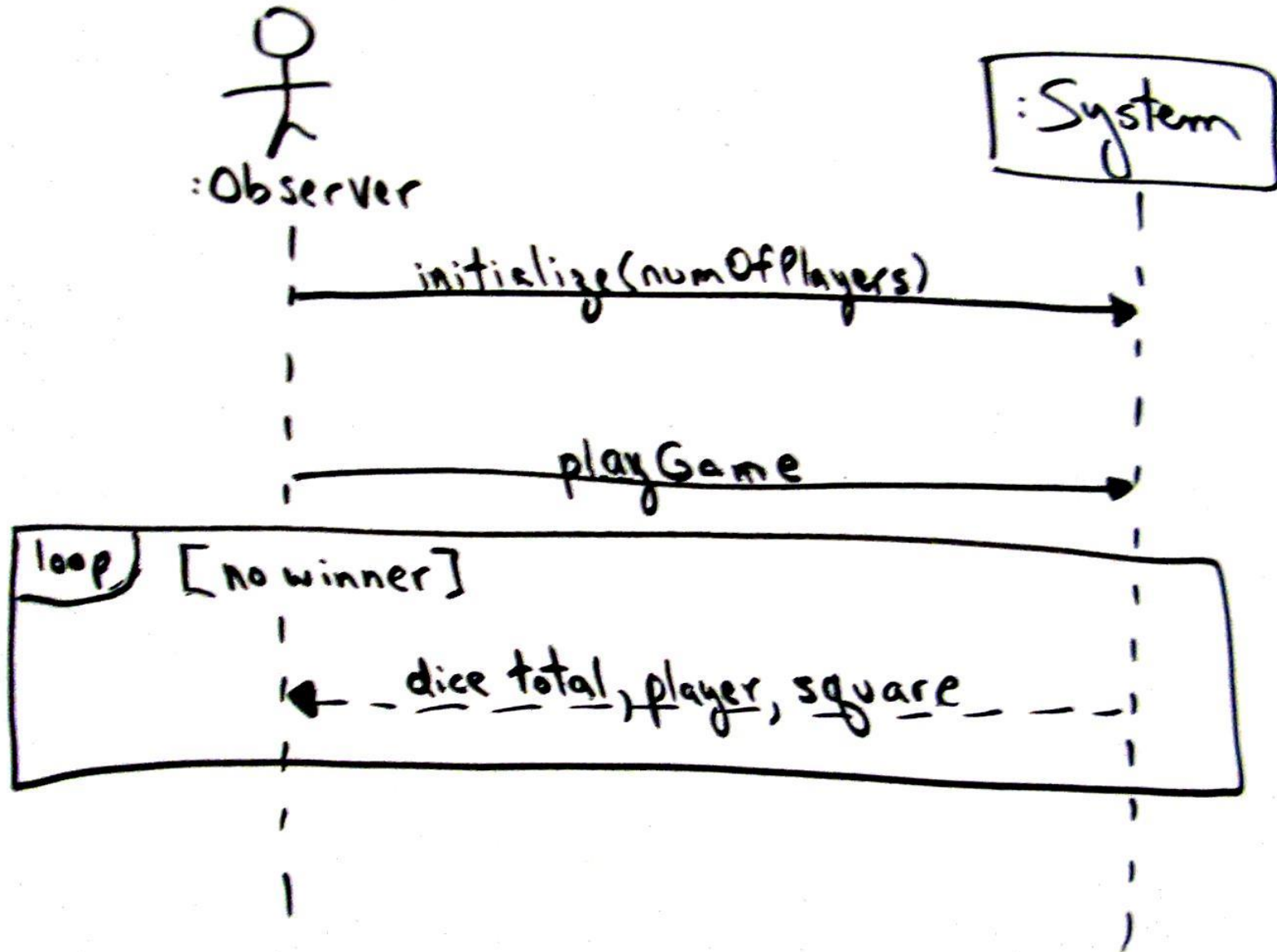
a message with parameters

it is an abstraction representing the system event of entering the payment data by some mechanism



CL Figure 10.2

# Example: Monopoly

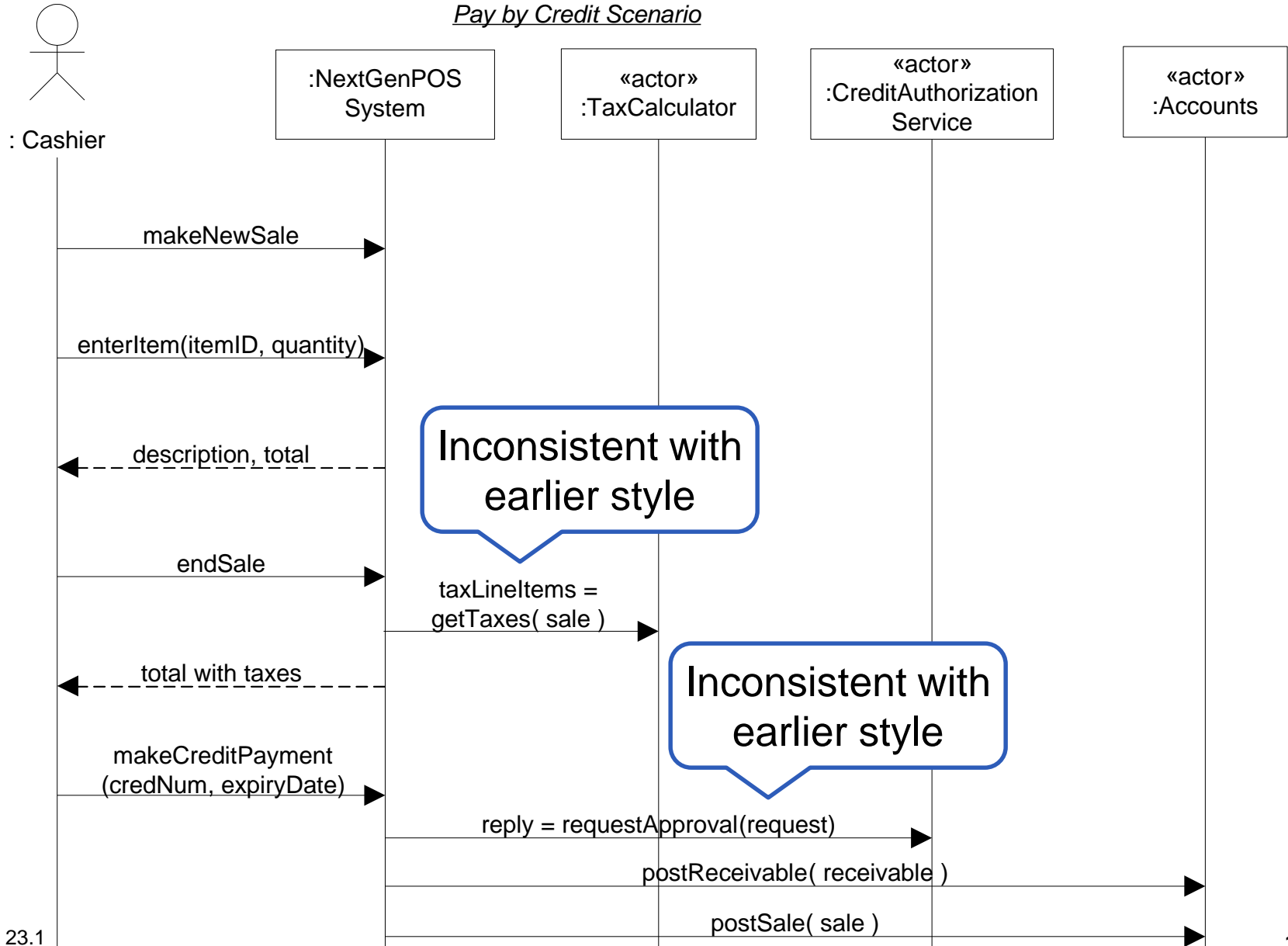


CL, Chapters 23, 32

# **SYSTEM MAY COMMUNICATE WITH THIRD-PARTY SYSTEMS**

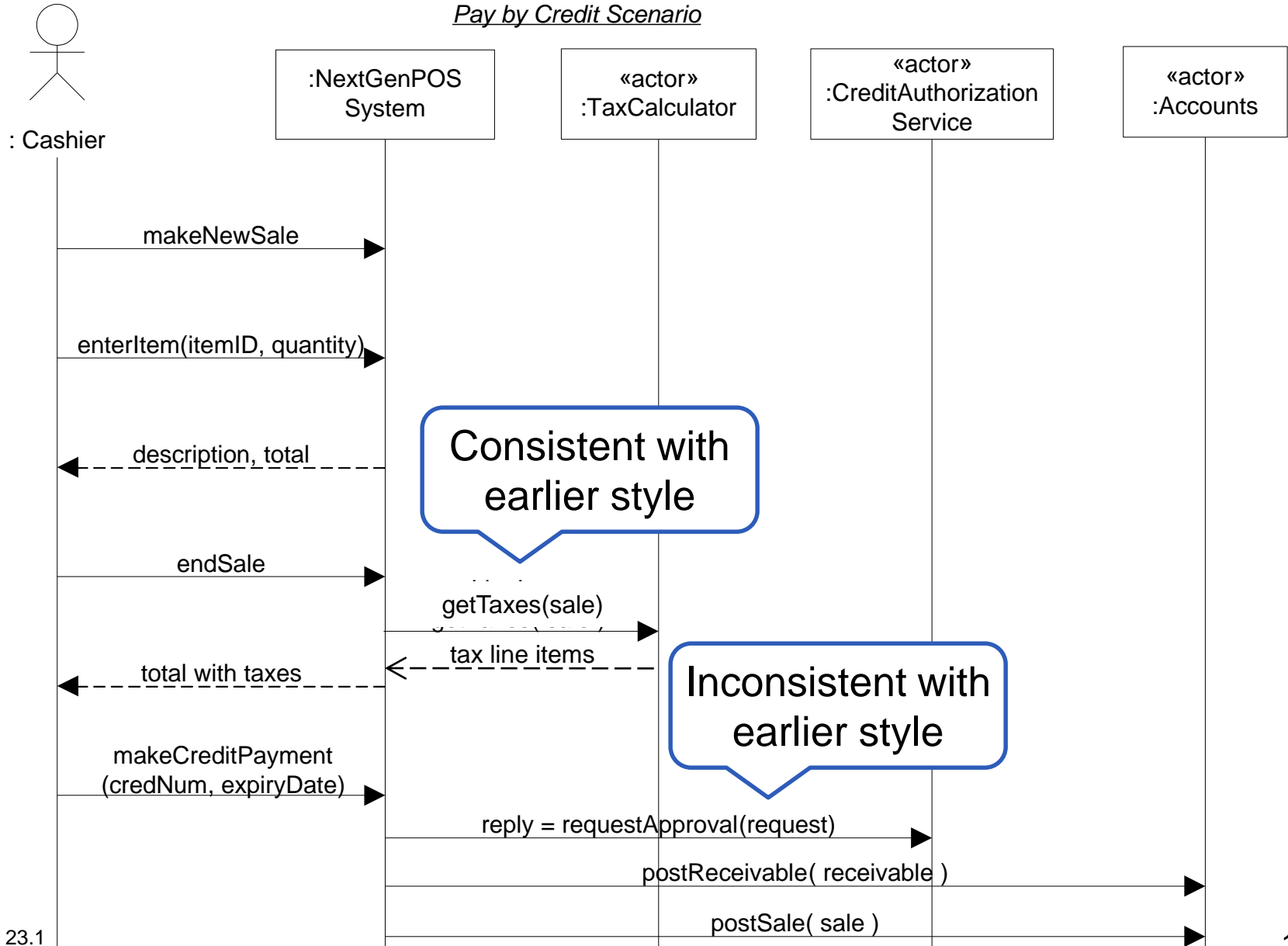
# POS Example From Textbook

*Process Sale*  
*Pay by Credit Scenario*



# POS Example Fixed

*Process Sale*  
*Pay by Credit Scenario*





# **IN-CLASS EXERCISE: CREATE A SYSTEM SEQUENCE DIAGRAM**

# Create a System Sequence Diagram

- Get together with your team
- Complete a system sequence diagram
  1. Pick the main success scenario of a use-case you defined for your project
  2. Convert the main success scenario into a system sequence diagram
- Be prepared to present your results