

System Sequence Diagrams

CSSE 374: Session 6

Shawn Bohner
Office: Moench Room F212
Phone: (812) 877-8685
Email: bohner@rose-hulman.edu

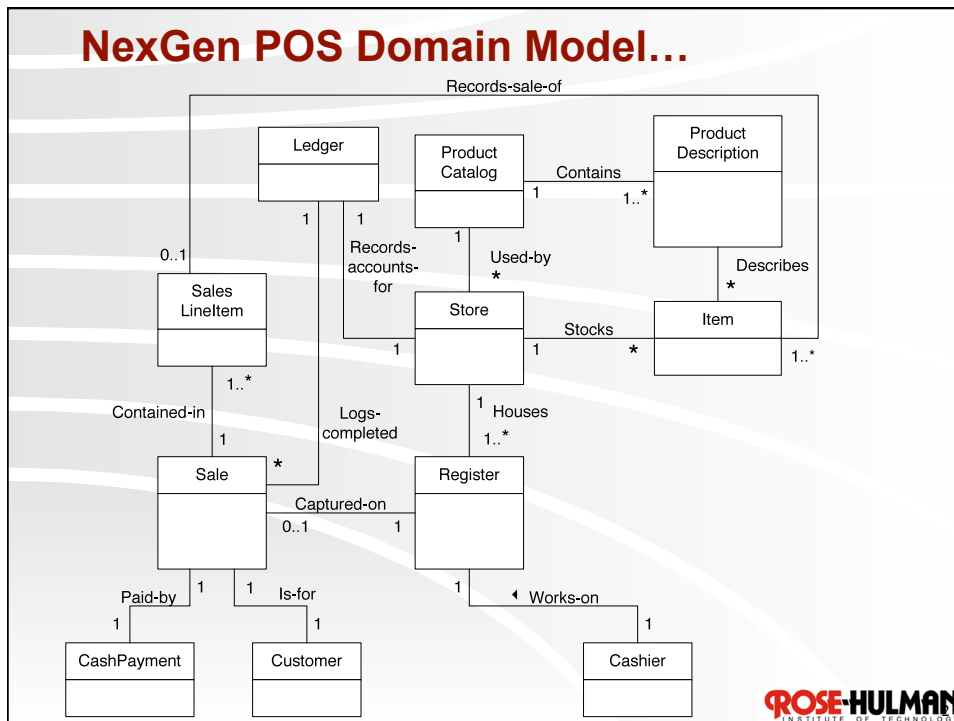


ROSE-HULMAN
INSTITUTE OF TECHNOLOGY

Q1

© 2009 Shawn A. Bohner

NexGen POS Domain Model...



ROSE-HULMAN
INSTITUTE OF TECHNOLOGY

We have Use Cases and Domain Model: So what's Next?

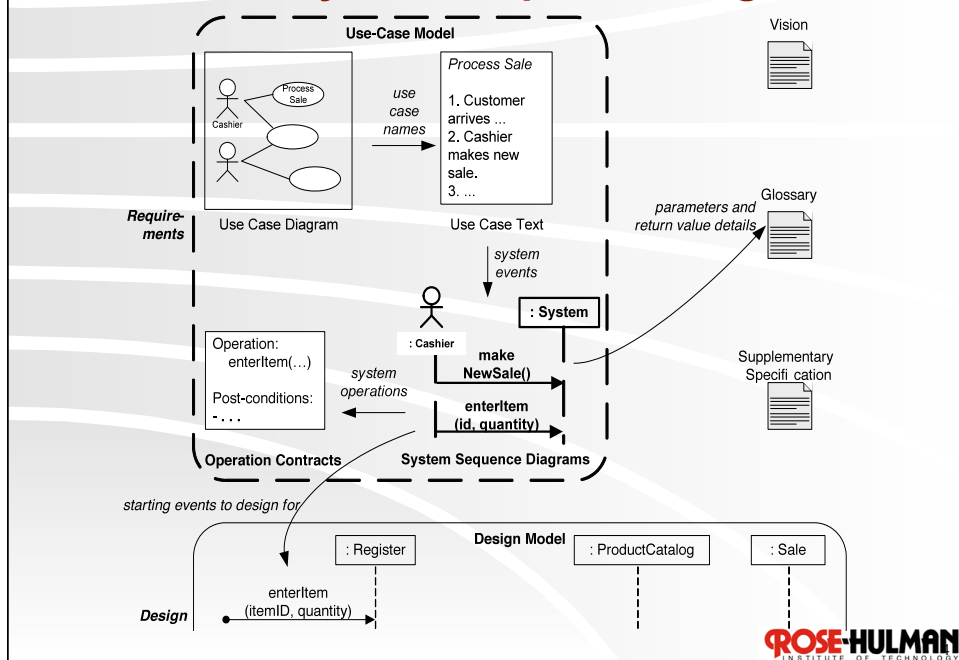
- ❖ **System Sequence Diagrams (SSD): A specialization of sequence diagrams that describe system behaviors**



- ❖ **SSDs typically modeled for the main use case scenario, and for frequent and alternative scenarios**

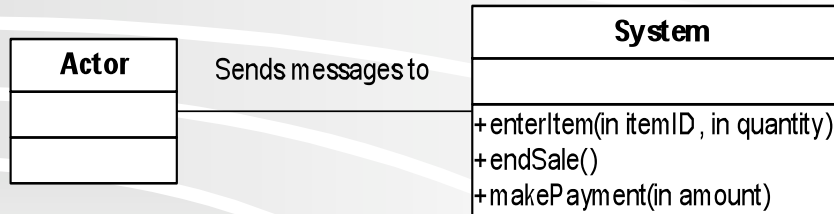
ROSE-HULMAN
INSTITUTE OF TECHNOLOGY

Where are System Sequence Diagrams?



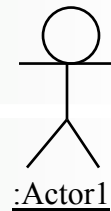
Modeling Behavior from a System Perspective

A **Use Case Scenario** is an ordered series of operations (functions) that Actors invoke on the System

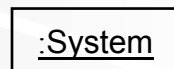


SSD Notation

❖ **Actor:** An Actor is modeled using the ubiquitous stick figure symbol



❖ **Object:** is represented as a rectangle which contains the name of the object underlined



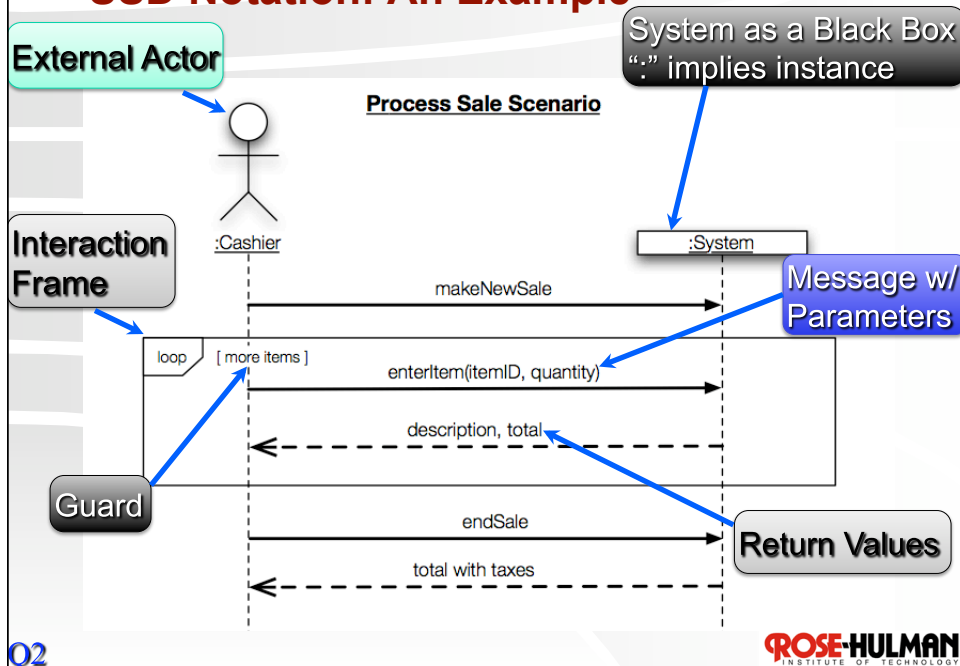
SSD Notation (continued)

- ❖ **Lifeline:** is depicted as a vertical dotted line extending from an object that identifies the existence of the object over time
- ❖ **Message:** modeled as horizontal arrows between activations, indicate the communications between objects

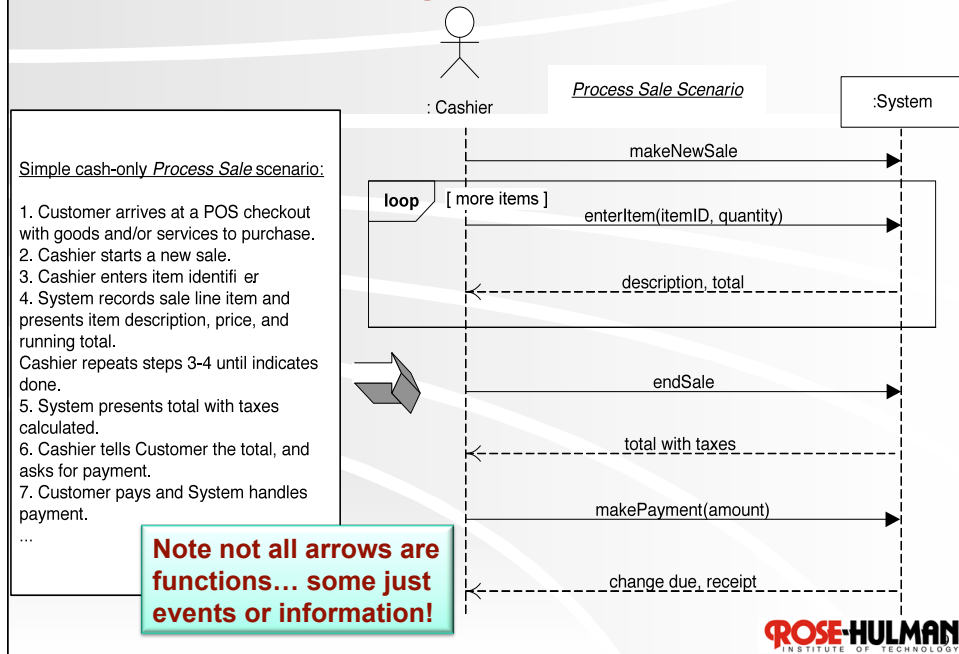
messageName(argument) →

ROSE-HULMAN
INSTITUTE OF TECHNOLOGY

SSD Notation: An Example



Relating UC and SSD



From Use Case to SSD

- ❖ Use cases describe how **external actors** will interact with our system
- ❖ Actors generate **system events** requesting some **system operation**
- ❖ For a **single scenario** of a use case, SSD shows **system events and their order**
- ❖ All systems treated as black boxes; only show events that **cross system boundaries**

Also inter-system events

Why Draw an SSD?

- ❖ Software systems react to three things:
 1. External input events (a.k.a., **system events**) from actors
 2. Timer events
 3. Faults or exceptions

- ❖ SSD captures **System Behavior**: a description of what a system does, **NOT** how it does it

Q4.5

ROSE-HULMAN
INSTITUTE OF TECHNOLOGY

How To “Tips” on Creating SSDs

- ❖ Show **one scenario** of a use case
- ❖ Show events as **intentions**, not physical implementation
 - E.g., *enterItem* not *scanItem*
 - E.g., *presentCredentials*, not *enterPassword*
- ❖ Start system event names with **verbs**
- ❖ Can model collaborations between systems
- ❖ Give **details in the Glossary**

ROSE-HULMAN
INSTITUTE OF TECHNOLOGY

Key Idea: SSDs are a Bridge

❖ Challenge: To Make the functional-OO transition

- Without losing any requirements
- Delivering a correct, robust system



❖ System Sequence Diagram is the key

- Links UCs with OO models (e.g., class & sequence)
- Supported by operation contracts
- Provides traceability of requirements into OO models

Let's do some Examples...

Homework and Milestone Reminders

- ❖ **Read Chapter 11 on Operations Contracts**
- ❖ **Homework 2 – Dog-eDoctor Domain Model**
 - Due by 5:00pm on Tuesday, December 8th, 2009
- ❖ **Milestone 2 – Junior Project Domain Model**
 - Due by 11:59pm on Friday, December 11th, 2009
- ❖ **Homework 3 – Dog-eDoctor SSDs and Operations Contracts**
 - Due by 5:00pm on Tuesday, December 15th, 2009