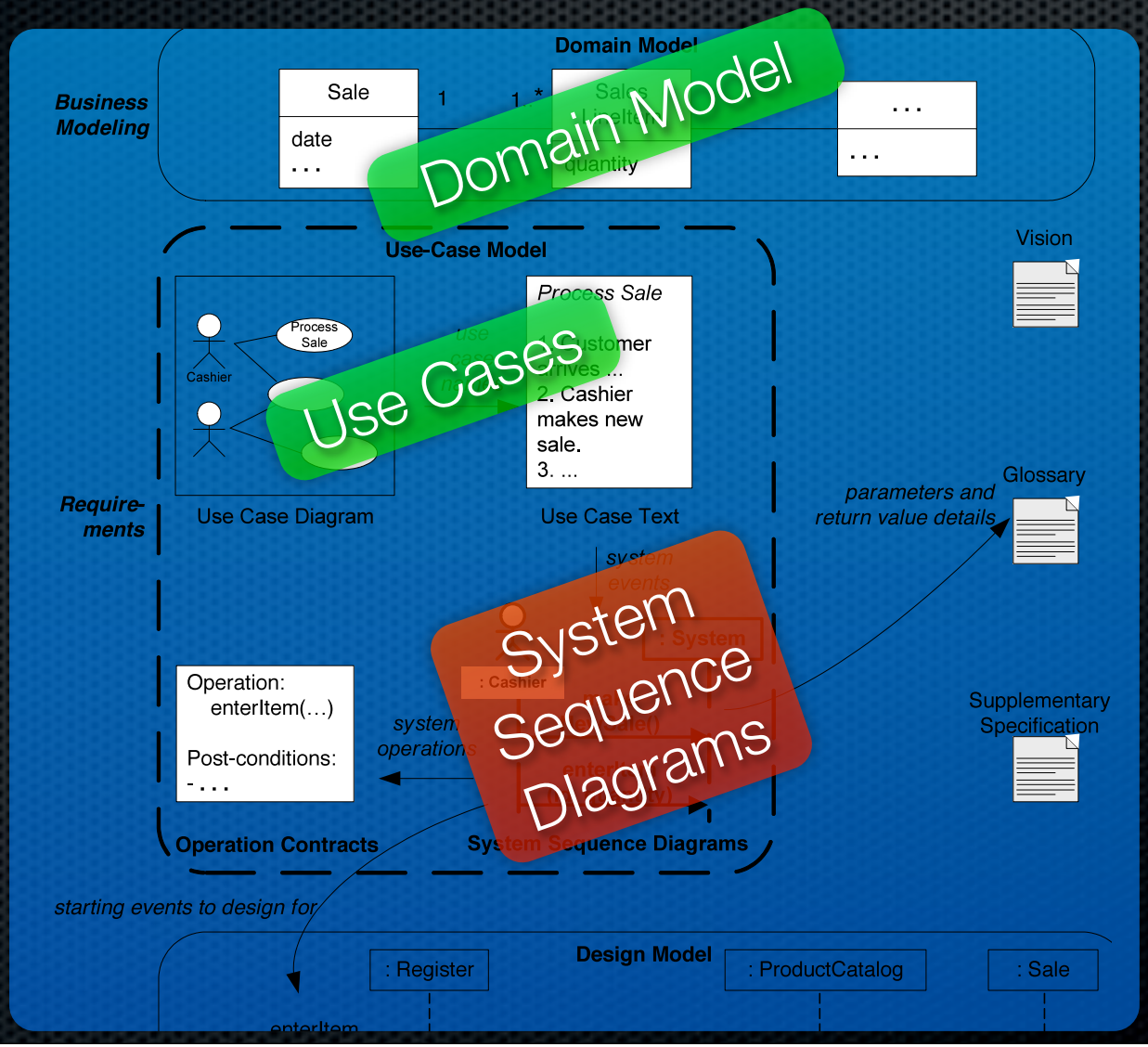
System Sequence Diagrams

Curt Clifton

Rose-Hulman Institute of Technology

Q1

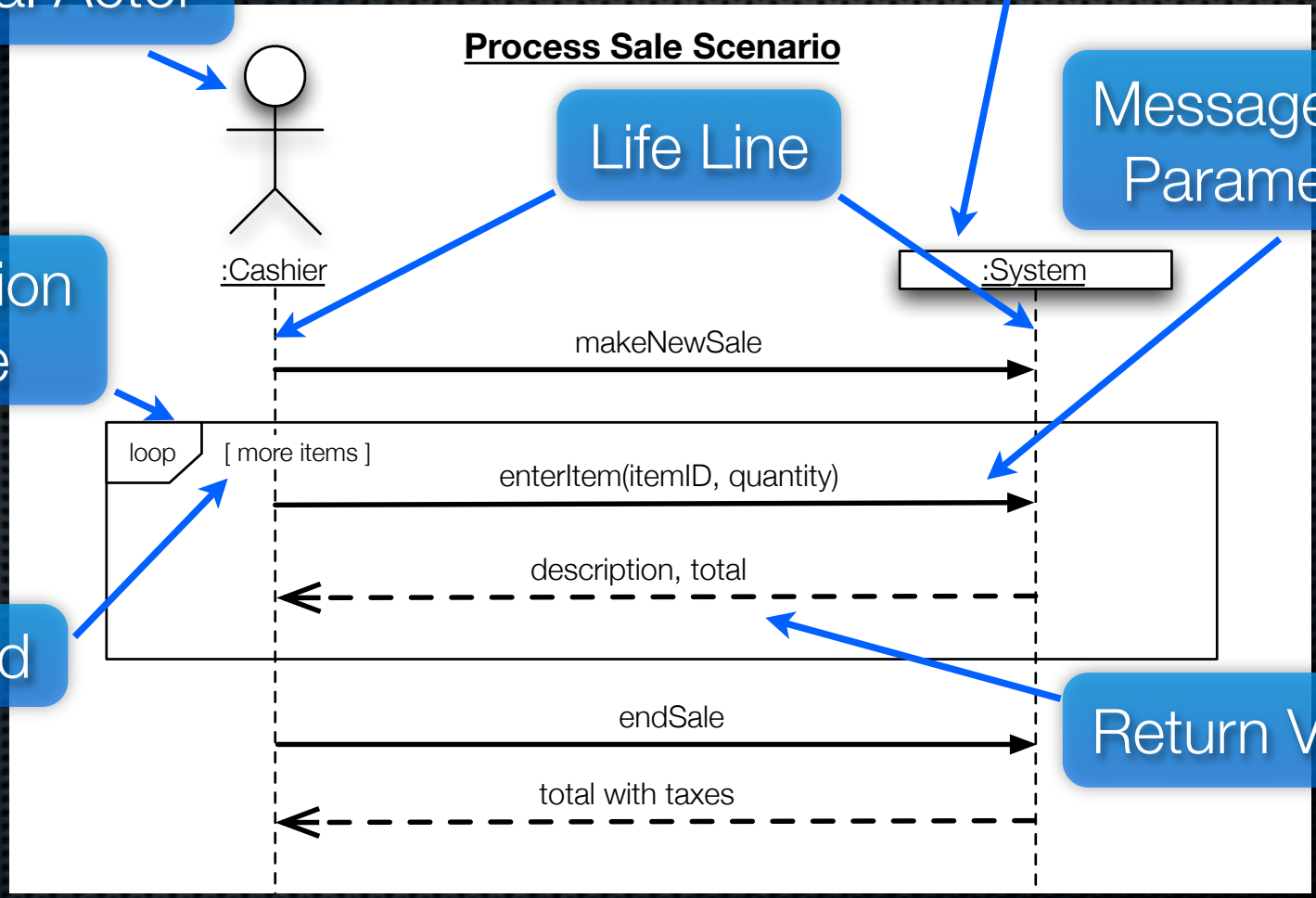
Where Are We?


Domain Model
Business Modeling
Sale
date
...
1
1..*
...
...
Domain Model
Use-Case Model
Vision
Process Sale
Cashier
Use Case Diagram
Process Sale
2. Cashier makes new sale.
3. ...
Use Case Text
Use Cases
Glossary
parameters and return value details
Require-ments
Operation:
enterItem(...)
Post-conditions:
- ...
Operation Contracts
system operations
System Sequence Diagrams
System Sequence Diagrams
Supplementary Specification
starting events to design for
Design Model
: Register
: ProductCatalog
: Sale
enterItem

System Sequence Diagrams

- Illustrate the **input and output events** related to the systems under discussion
- Show large-scale operations of systems
  - Starting point for designing collaborating objects

SSD Notation



Black Box Object

External Actor

Life Line

Message with Parameters

Interaction frame

Guard

Return Values

Q2

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**

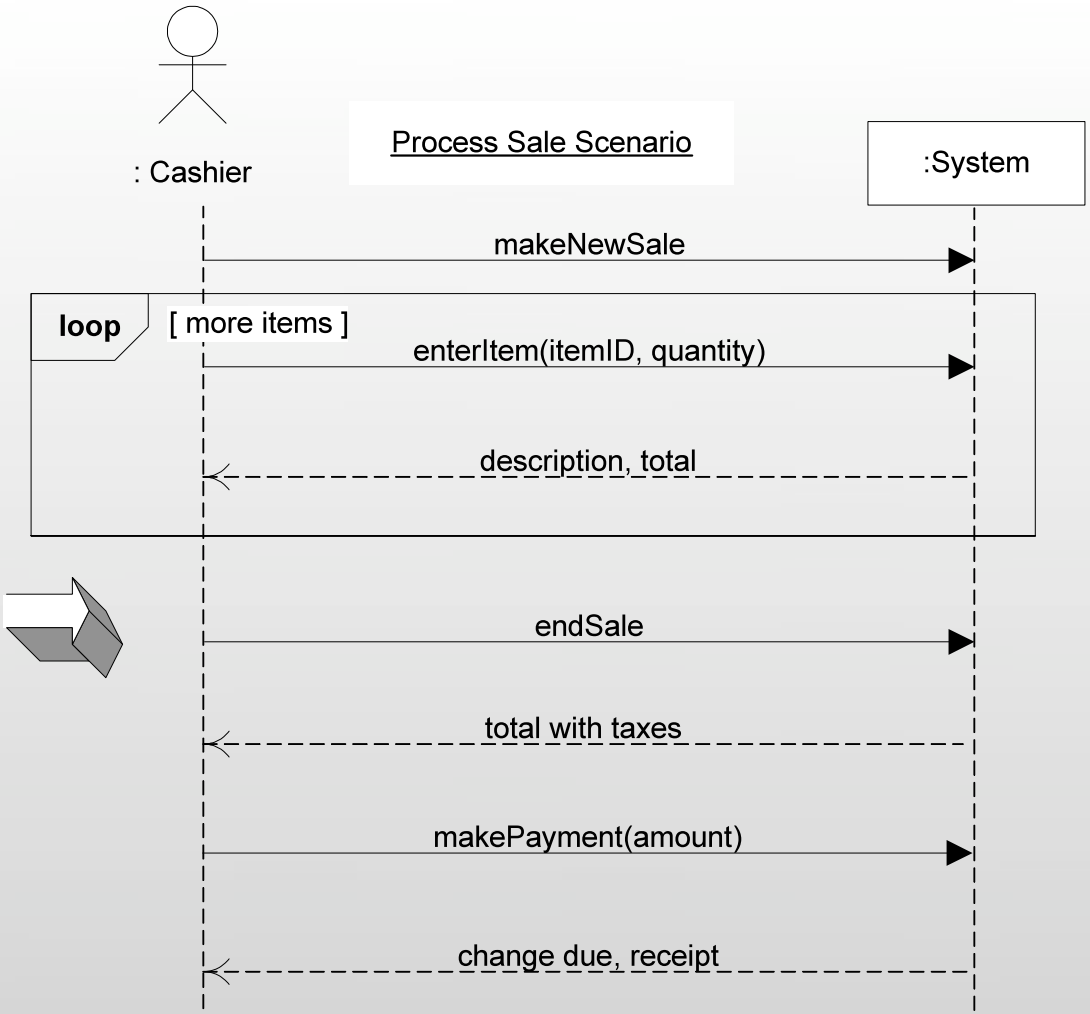
Also inter-system events

Q3

Example

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.
...



Why Draw an SSD?

- Software systems react to three things:
  - External input events (a.k.a., **system events**) from actors
  - Timer events
  - Faults or exceptions
- SSD captures **system behavior**: a description of *what* a system does, *not how* it does it

Q4,5

SSD Tips

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

When To Create SSDs

- Just draw them for the scenarios in the next iteration
- Useful for:
  - Understanding external interface
  - Understanding collaboration with existing systems
  - Documenting system architecture

Key Idea

- System Sequence Diagrams are **a bridge**
  - **From functional** Use Cases
  - **To** an **object-oriented** System Model
  - Providing **requirements traceability**

http://justinsomnia.org/2007/10/golden-gate-bridge-at-night/

Examples…

Q6,7