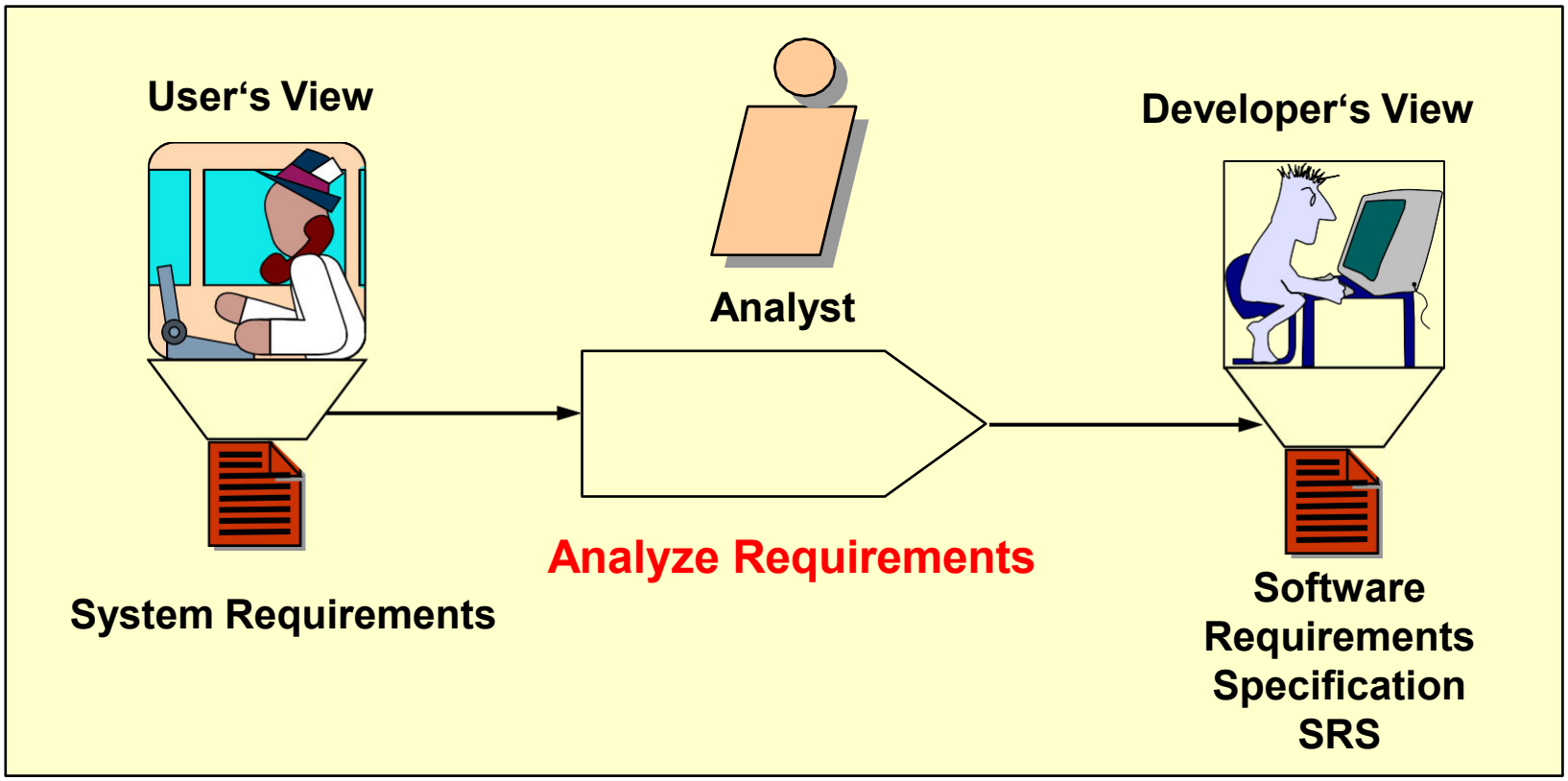
Content

- OOA in the Development Process
- Modeling the Behavior
- Domain Modeling
- Summary

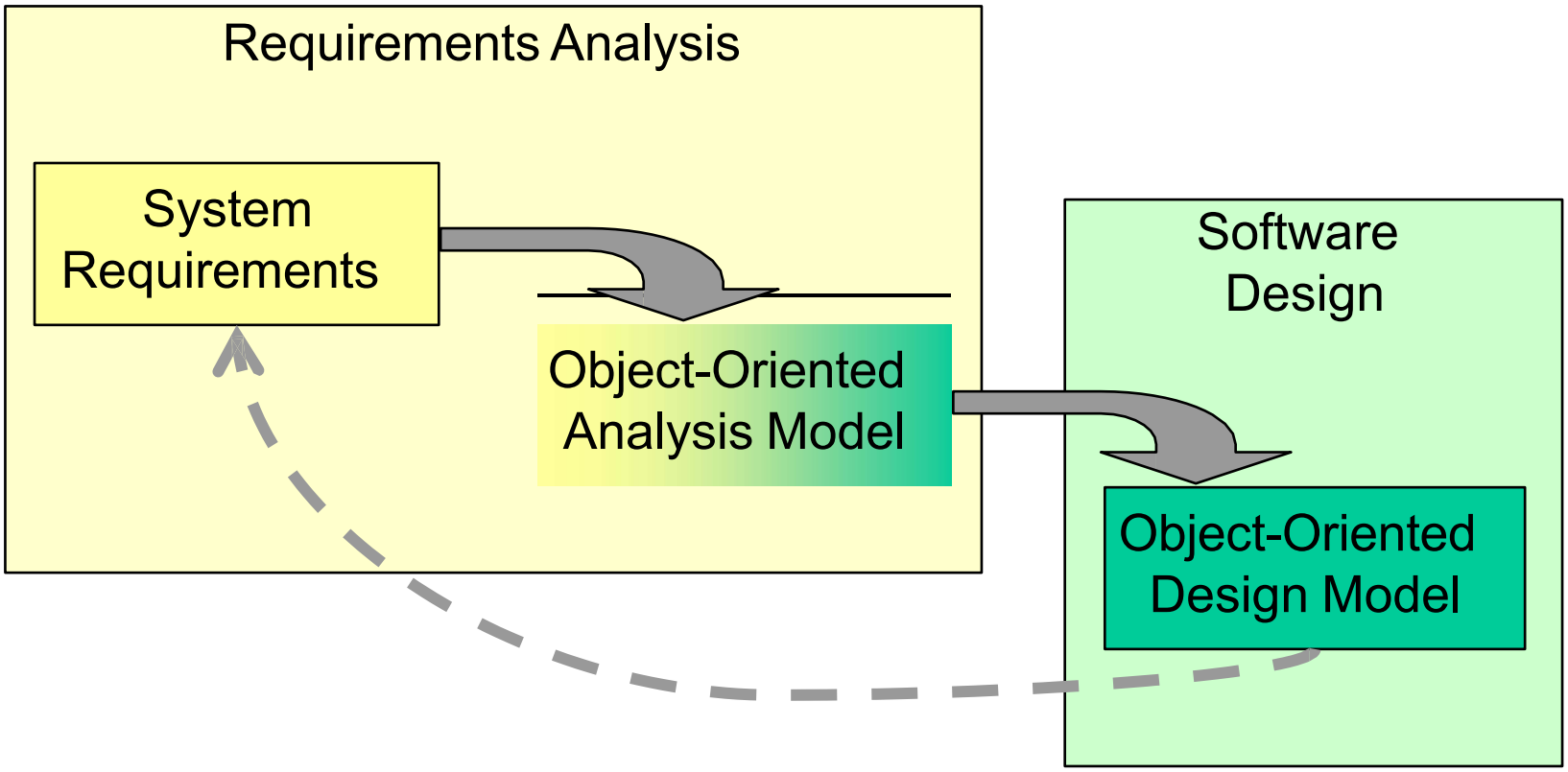
What is Analysis?

- Requirements Analysis

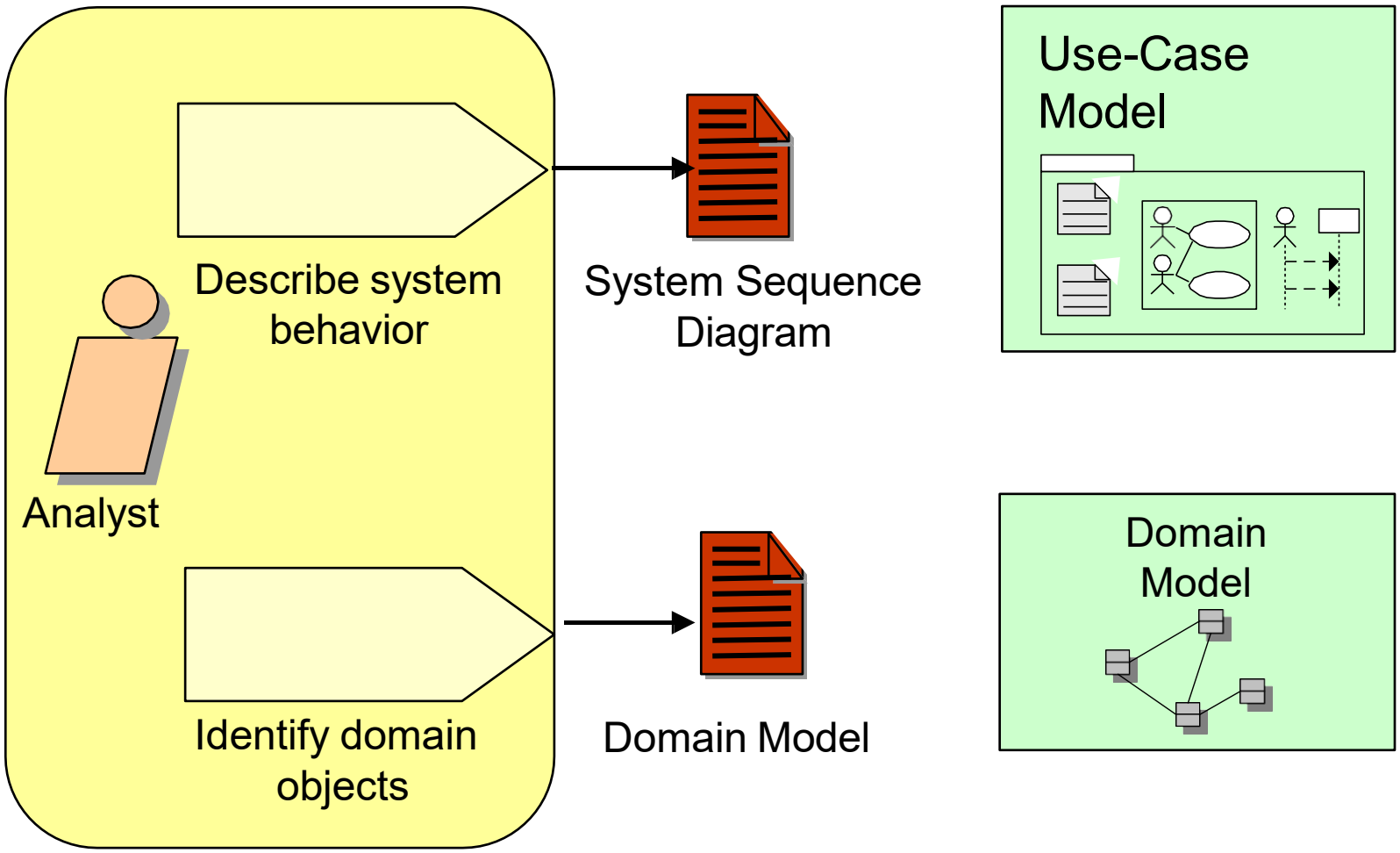


What Is OOA?

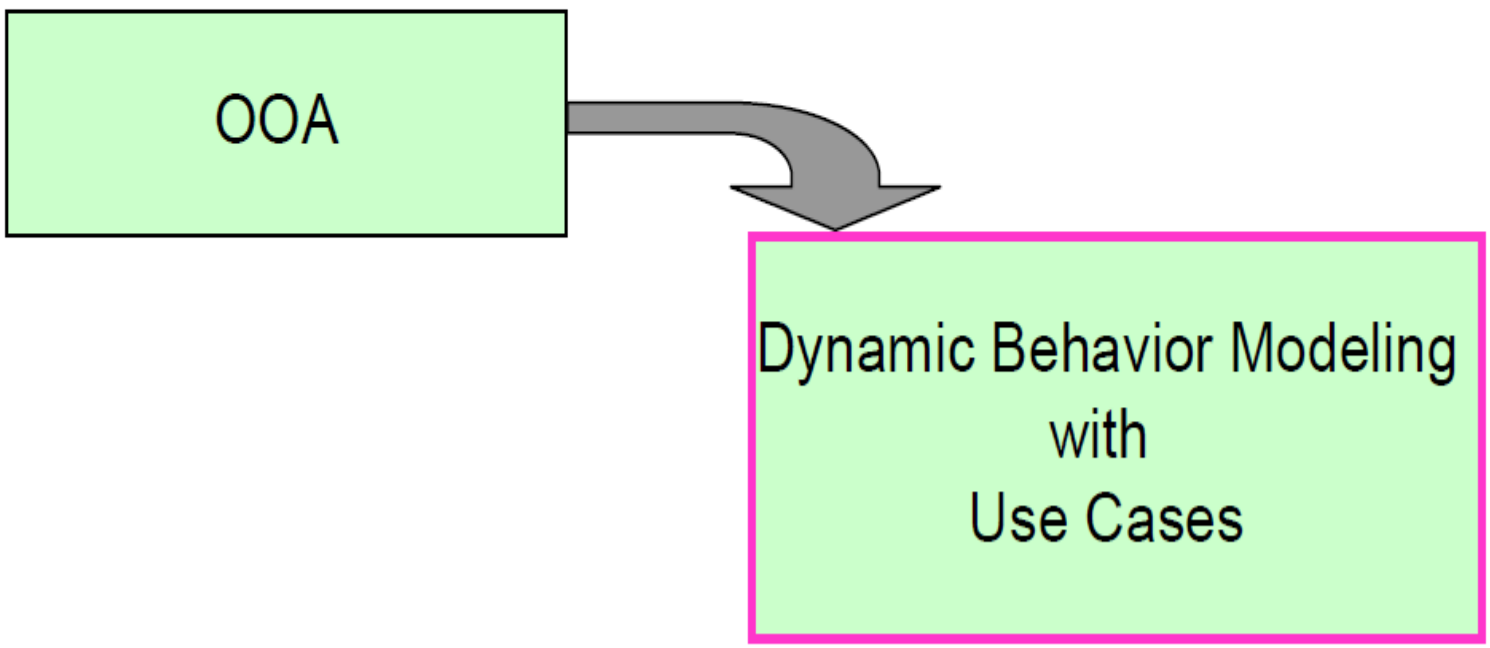
- Bridging the gap



OOA Discipline



Requirements Discipline in RUP


OOA
Dynamic Behavior Modeling
with
Use Cases

IEEE Standard 830-1993

Traditional practice for Software Requirements Specifications

Recommended document structure:

1. Introduction
1. Purpose
2. Scope
3. Definitions, acronyms, and abbreviations ⓪ Glossary!
4. References
5. Overview

2. Overall description
1. Product perspective
2. Product functions
3. User characteristics
4. Constraints
5. Assumptions and dependencies

3. Specific requirements ⓪ List of functional and non-functional requirements

Appendixes

Index

Case Study

- NextGen POS
  - The Next Generation Point-Of-Sale System

Case Study: NextGen POS

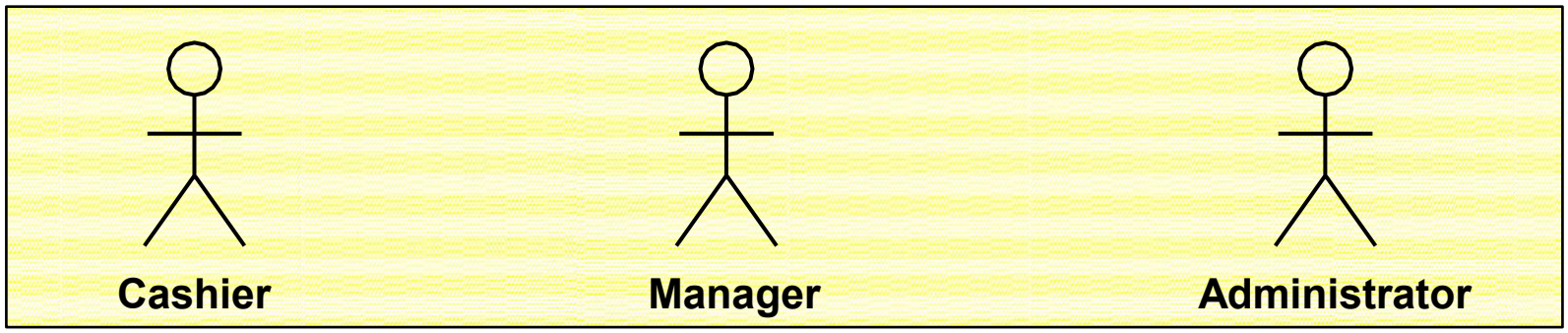
- Tasks
  - Record sales
  - Handle payments
  - Control inventory
  - Print receipts
  - Easy-to-use
  - Touch-screen interface
  - Multi-node vending system
  - Distributed system

NextGen Pos: System Boundary

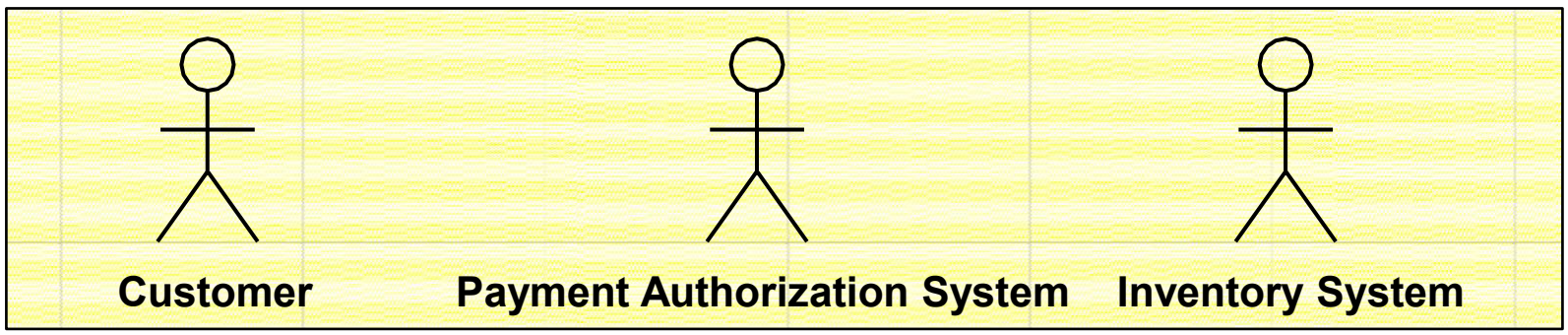
- External
  - Cashier
  - Inventory System
  - Payment Authorization System
- Internal
  - Point-of-sale system
    - Complete payment handling

NextGen POS: Actors

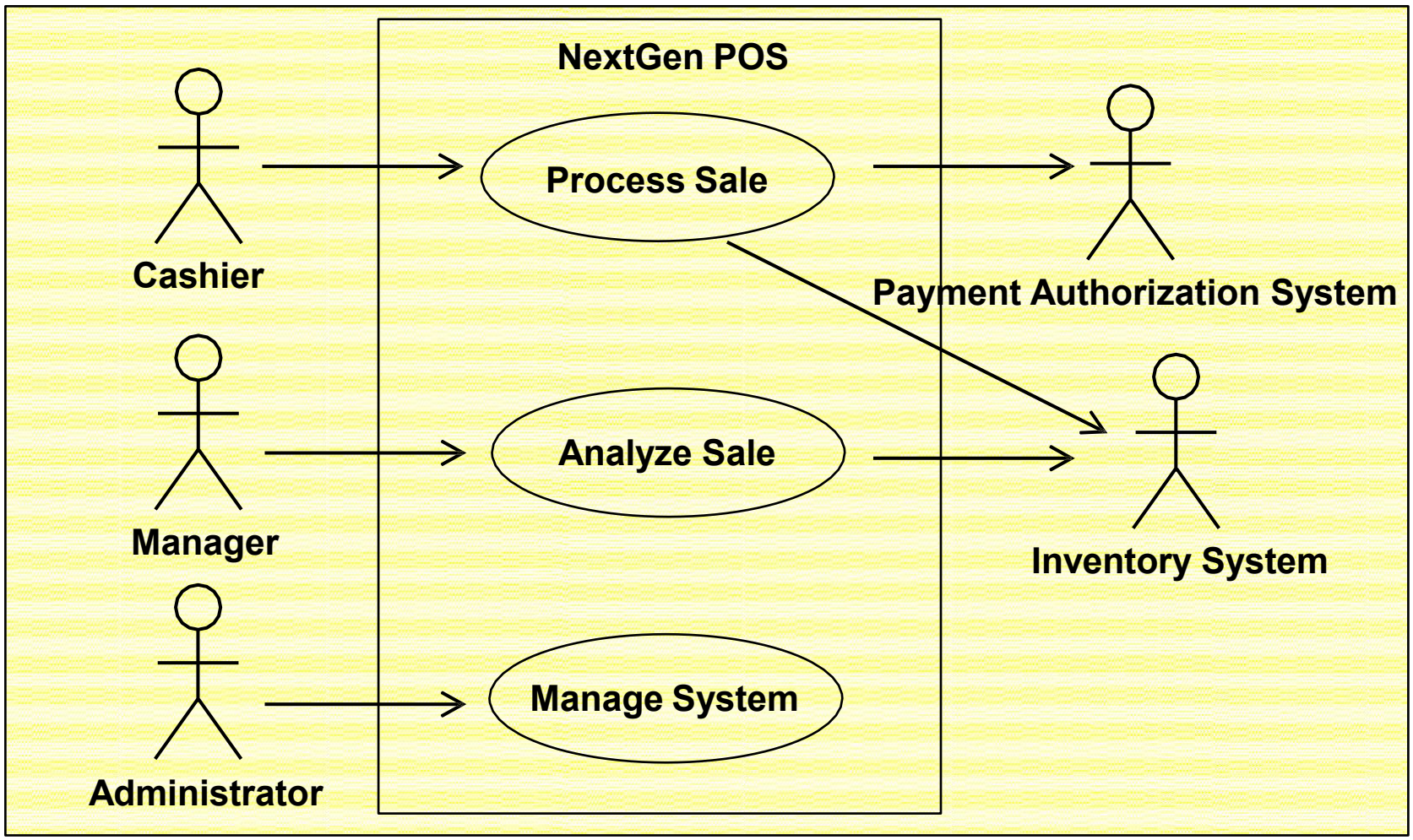
- Primary Actors



- Secondary Actors



NextGen POS: Use Cases



Use Case: Process Sale

Use Case UC1:	Process Sale
Scope:	Point of Sale System
Level:	User Goal
Goal in Context:	Accurate and fast sales process with no payment errors.
Actors:	Primary: - Cashier: processes the sales items and returns changes
	Secondary: -Customer: purchases sales items and gets change -Payment Authorization Service: processes customer authorization

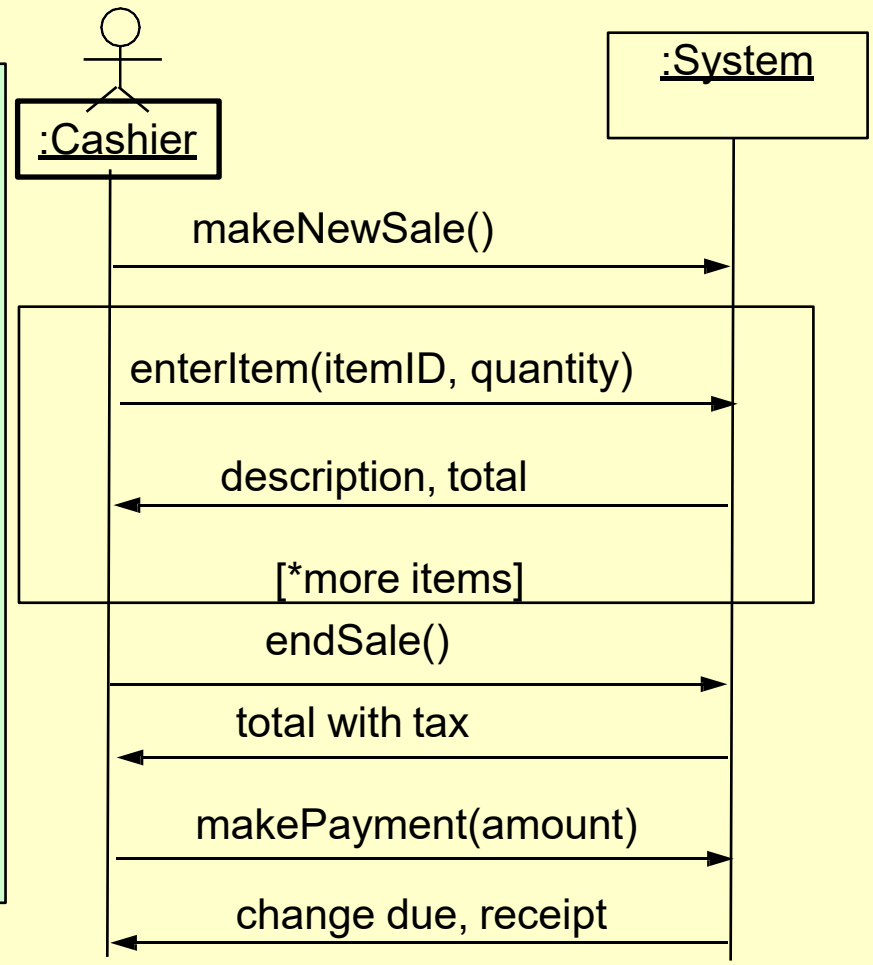
Use Case: Process Sales

Steps (Basic Flow):	1. Customer arrives at POS checkout with goods 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. Price calculated from a set of price rules. 5. Cashier repeats step 3-4 until done with all items. 6. System presents total with taxes calculated. 7. Cashier tells Customer the total, and asks for payment. 8. Customer pays and System handles payment. 9. System logs completed sale and sends sale and payment information to the external Accounting system (for accounting and commissions) and Inventory system (to update inventory). 10. System presents receipt. 11. Customer leaves with receipt and goods.

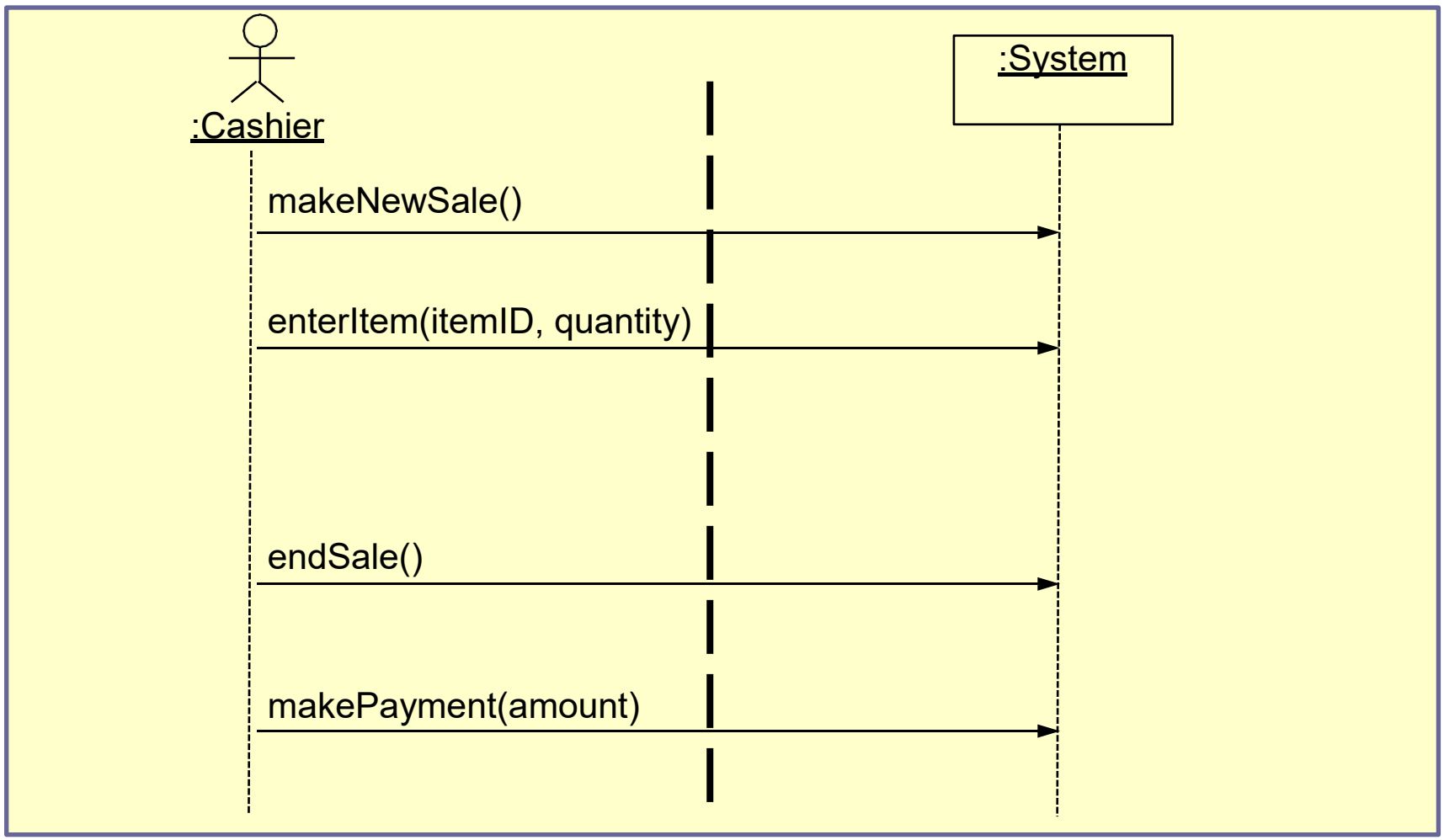
Use Case and System Sequence Diagram SSD

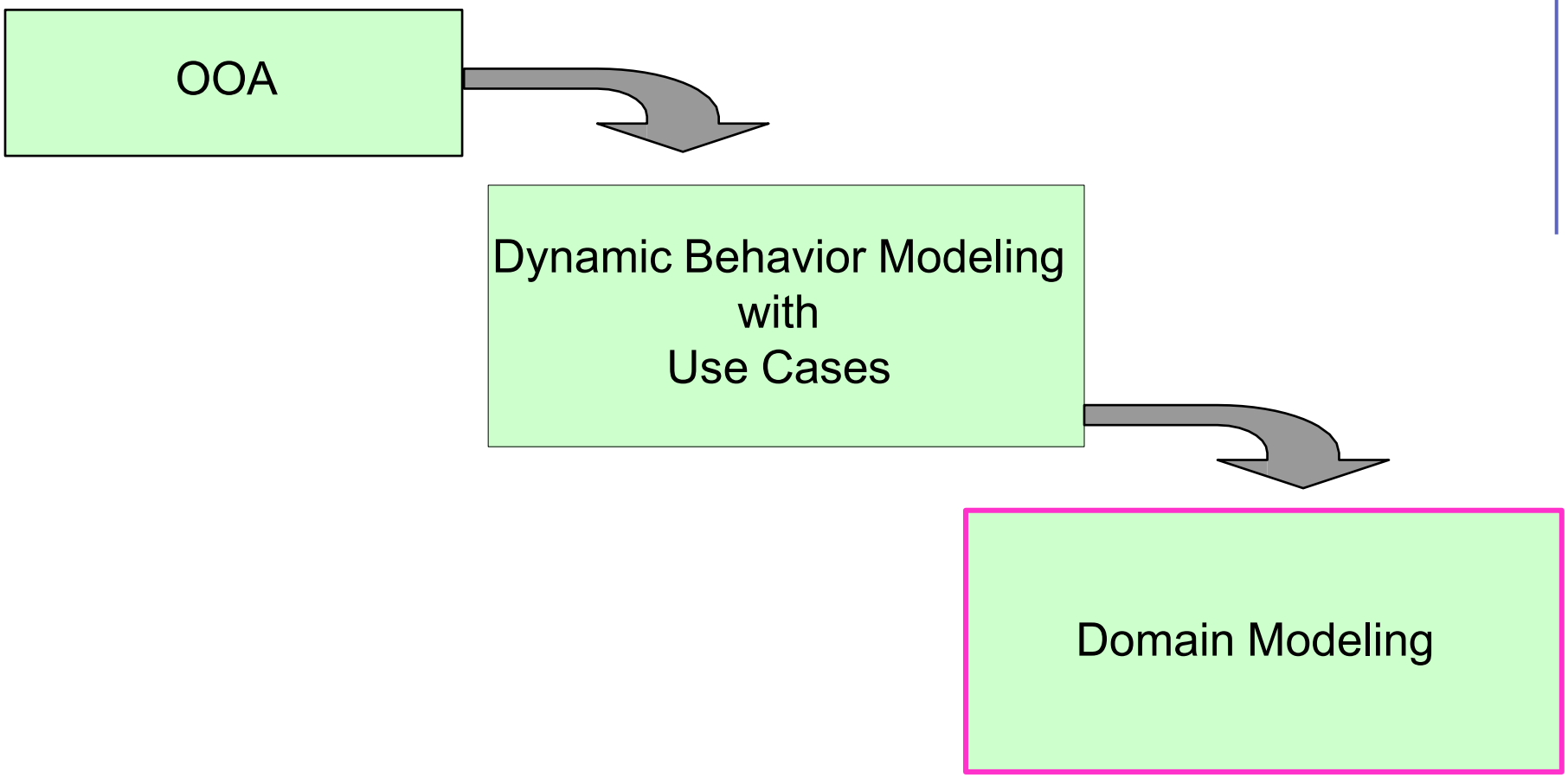
Simple: Process Sale scenario

1. Customer arrives at POS checkout with goods to purchase
2. Cashier starts a new sale
3. Cashier enters item identifier
4. System records sale line itemand presents item decription, price, and running total
   Cashier repeats 3-4 until indicates done.
5. System presents total with taxes
6. Cashier tells customer the total and asks for payment
7. Customer pays and system handels payment



SSD and System Boundary




OOA
Dynamic Behavior Modeling
with
Use Cases
Domain Modeling

Domain Modeling

- Goal
  - The problem domain is captured in a domain model
- Activities
  - Identify the conceptual classes with their attributes and their associations
- Input
  - Use Cases
- Result
  - Conceptual class diagram
    - Domain objects
    - Associations among the objects
    - Attributes of the objects

Domain Model Issues

- No Software Artifacts …
  - No attribute types
  - No methods
- … but Conceptual Class
  - Symbol – the box
  - Intension – described in an annotation
- … and System Decomposition
  - Division by conceptual classes

How To Develop The Domain Model

Apply the following steps:

1. Identify candidate conceptual classes
2. Add associations between the classes
3. Add attributes to the classes

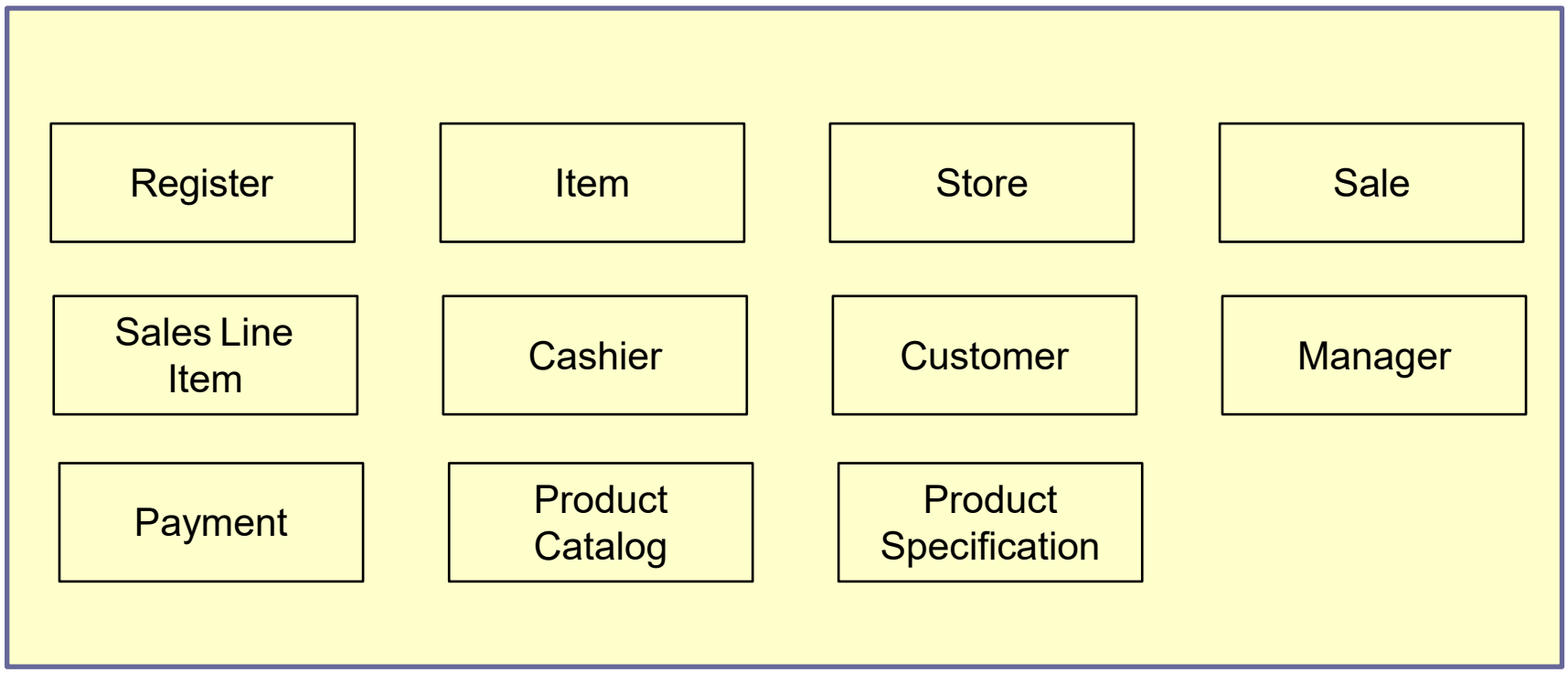
Domain Modeling Guidelines

- Class Naming
  - Use existing names of the problem domain
  - Exclude irrelevant features
  - Do not add things, that are not there (mapmaker strategy)

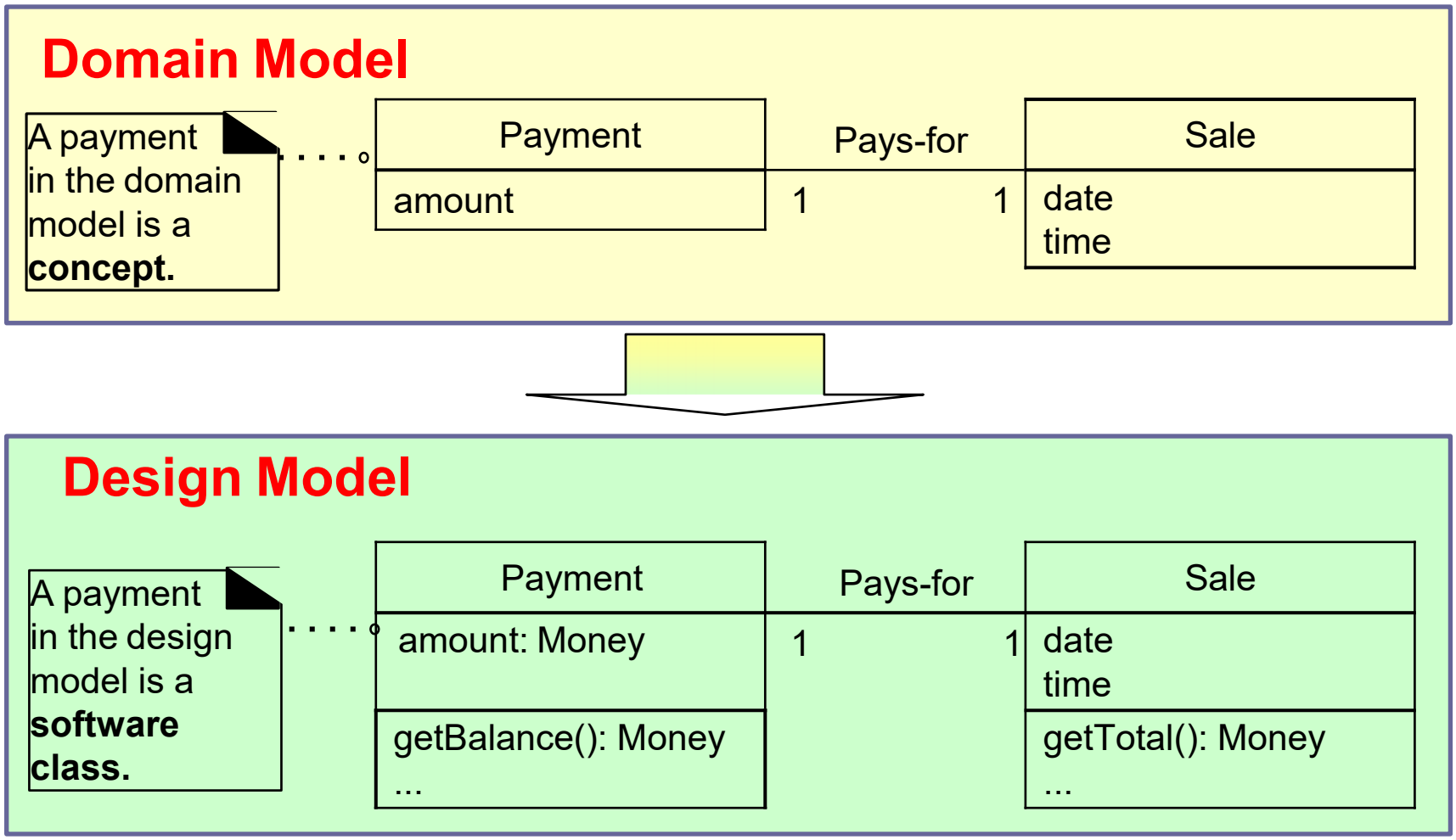
- Modeling of Unreal Elements
- Example of a telecom system:
  Message, Port, Connection
- Example of conceptual classes:
  ProductSpecification, ItemDescription

NextGen POS: Step 1

- Conceptional Classes

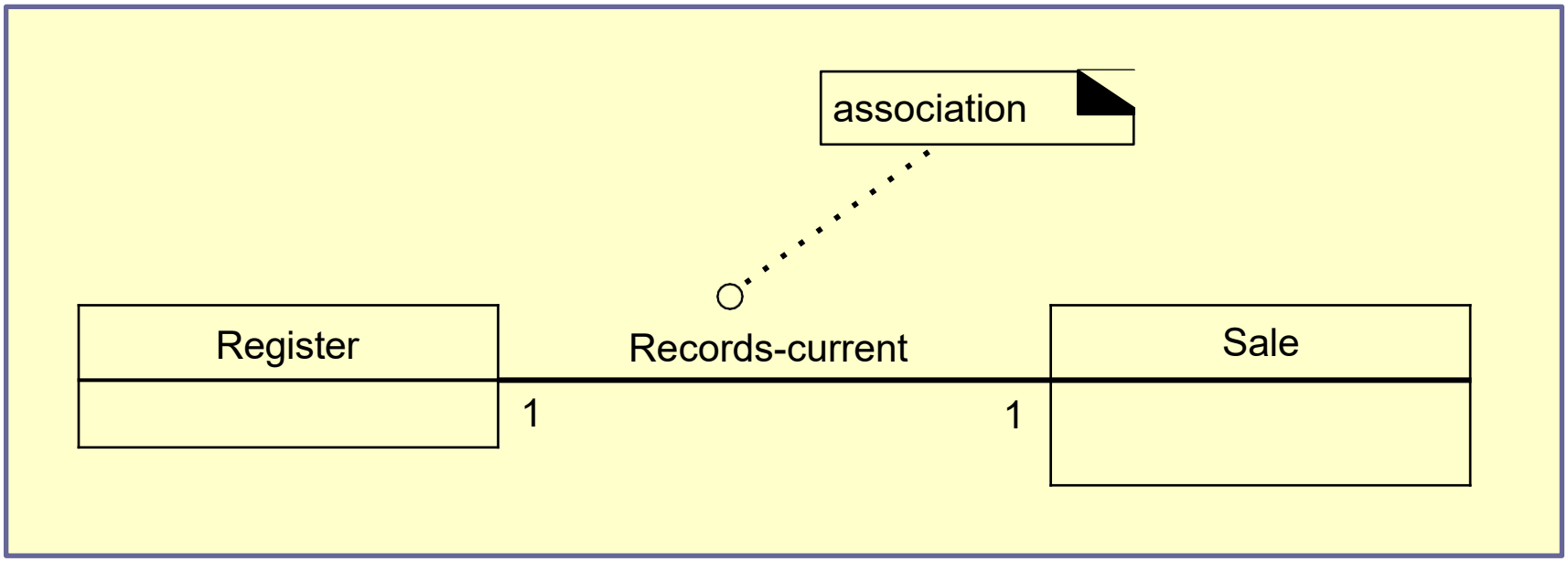


Analysis vs. Design



NextGen POS: Step 2

- An Association is ...
  - ... a relationship between instances of types that indicates some meaningful connection



Finding Associations

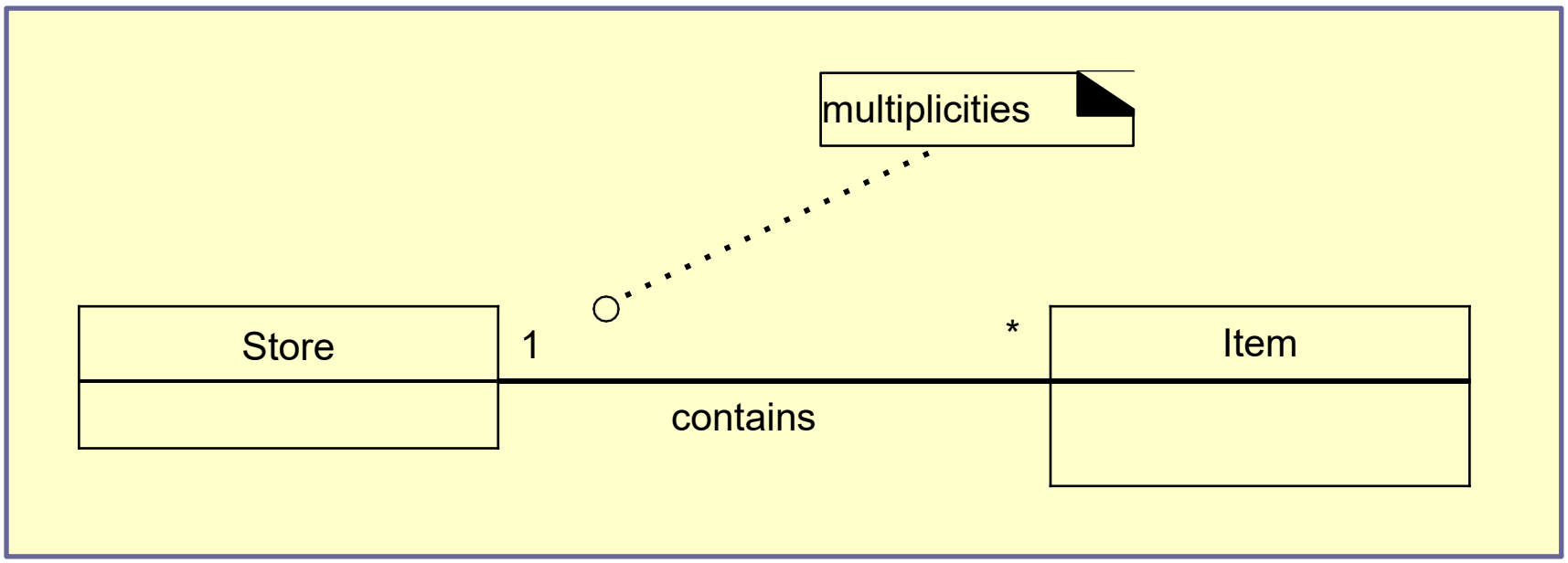
- Common Associations list
  - Summarizes some typical situations, which leads to associations
    - Is physical part of
    - Is logical part of
    - Is physically contained in
    - Is logically contained in
    - ...

Association Guidelines

- Focus on those associations for which knowledge of the relationship needs to be preserved for some duration
  - „*Need-to-know*“ associations
- Too many associations rather confuse than illuminate
- Avoid redundant or derivable associations

Association Guidelines

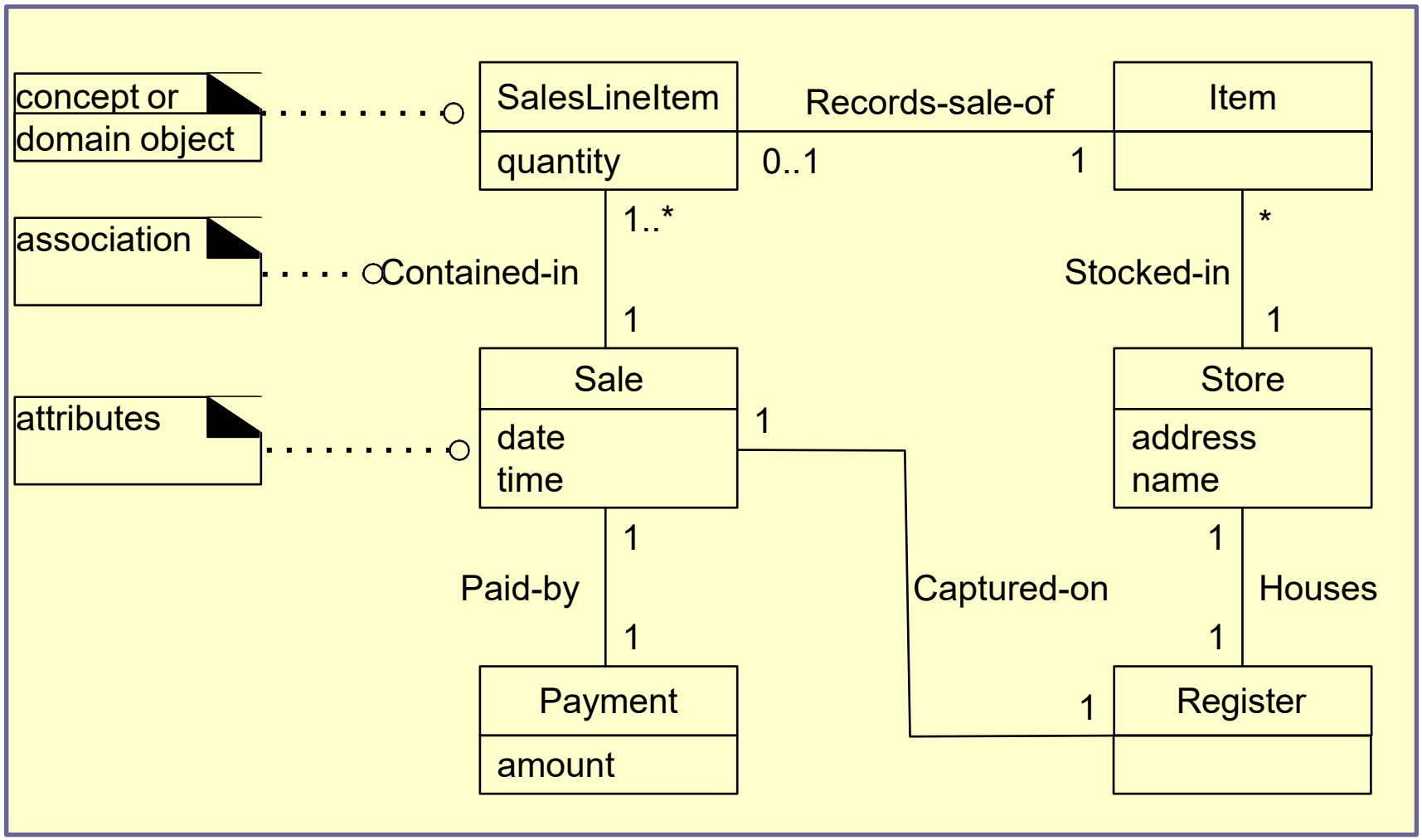
- Multiplicities
  - Specify domain constraints



NextGen POS: Step 2 Associations

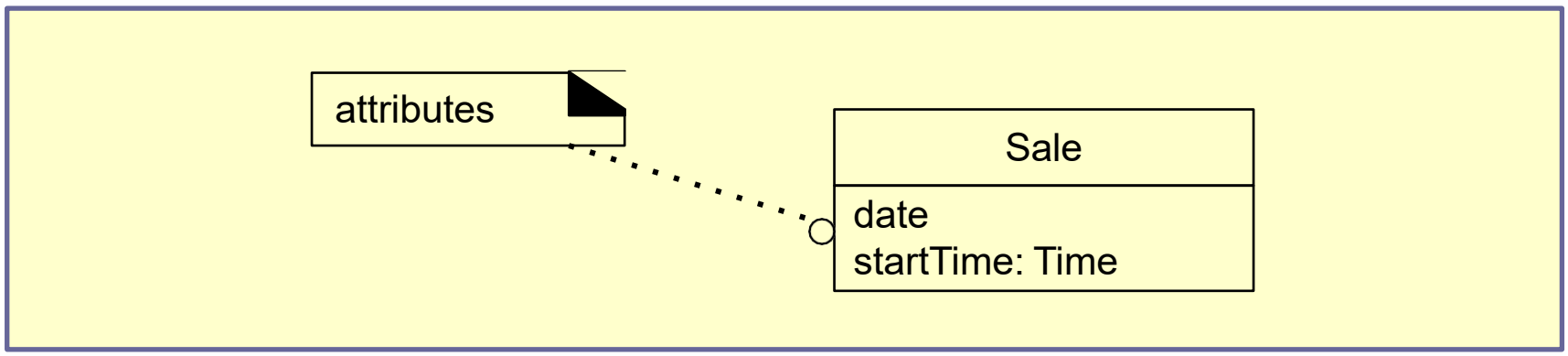
- Unforgettable Relationships
  - *Register Records Sale*
    - To know the current sale, generate total, print receipt
  - *Sale Paid-by Payment*
    - To know if the sale has been paid, relate the amount tendered to the sale total, and print a receipt
  - *ProductCatalog Records ProductSpecification*
    - To retrieve a product specification, given an itemID

Domain Model



NextGen POS: Step 3

- An attribute is ...
  - ... a logical data value of an object



Attribute Guidelines

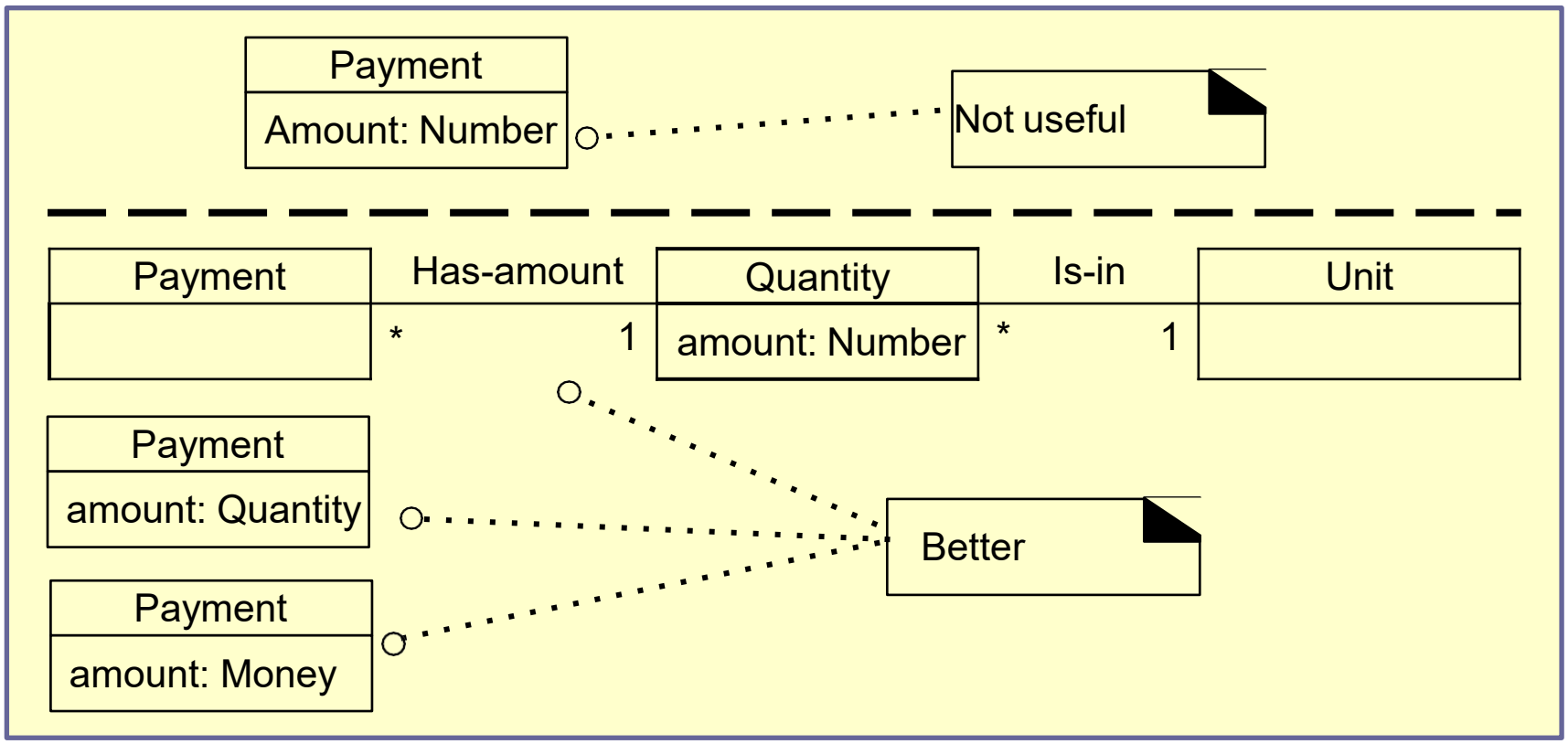
- Criteria
  - Choose those attributes for which requirements suggest or imply a need to remember information
- Keep attributes simple
  - Simple data types
    - Boolean, Date, Number, String, ...
  - Common data types
    - Address, Color, Phone Number, ZIP code,...

More Attribute Guidelines

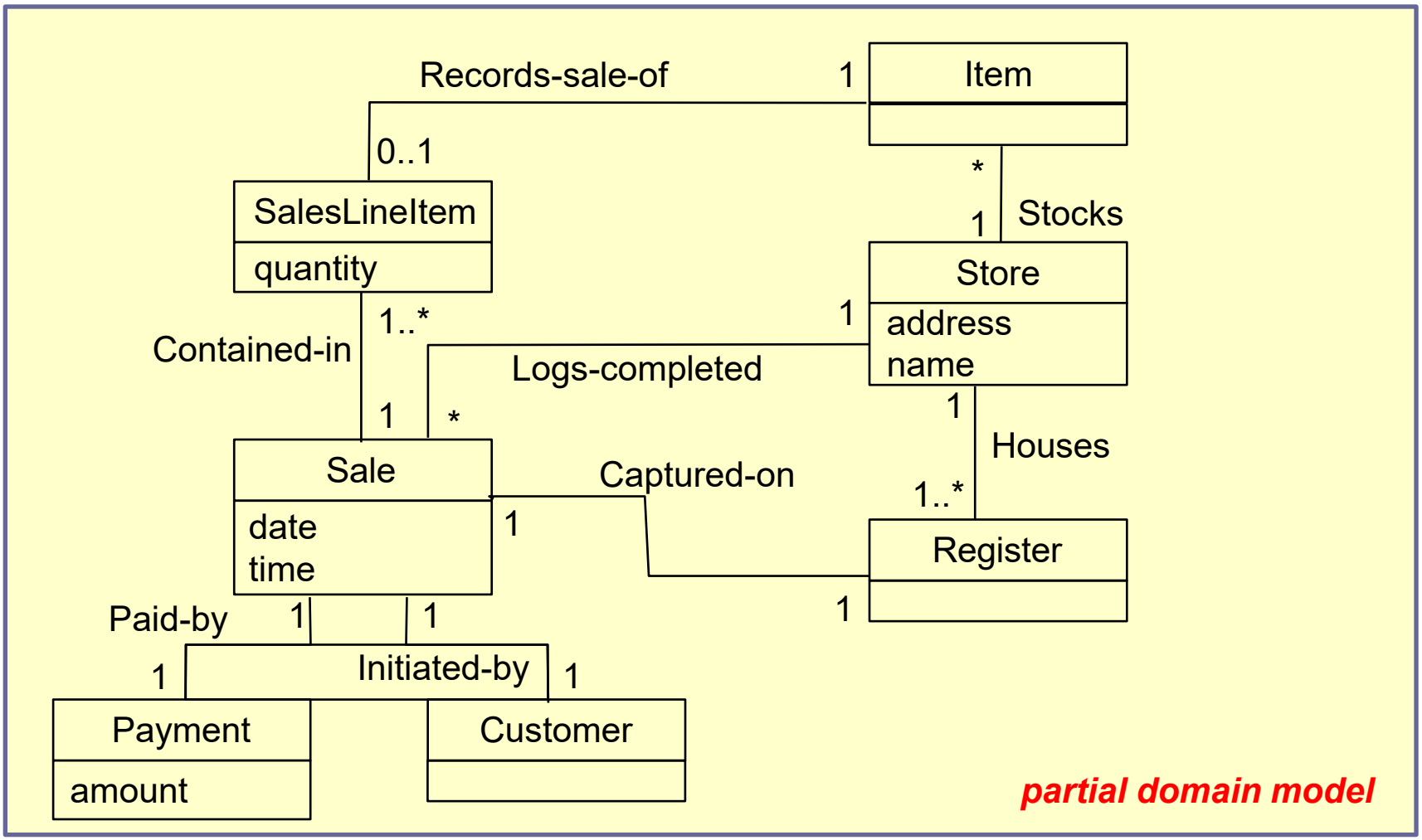
- Make an own conceptual class if a non-primitive data type
  - Is composed of separate sections
    - phone number, name of person
  - There are operations associated with it (parsing or validation)
    - Social security number
  - It has other attributes
    - Promotional prices usually have start and end date
  - It has a quantity with a unit
    - Payment amount with currency unit

More Attribute Guidelines

- Modeling Attribute Quantities and Units



NextGen POS: Step 3



Summary

- Analysis modeling is **not design**
- Handle system as **black-box**
- Model **interaction** between actors and system with System Sequence Diagrams
- Domain objects are **not software classes**
- Domain model is important **input** for design
- Domain objects may **become** software classes