CSSE 374: More Domain Model Refinements

Shawn Bohner

Office: Moench Room F212

Phone: (812) 877-8685

Email: bohner@rose-hulman.edu

Q1

ROSE-HULMAN INSTITUTE OF TECHNOLOGY

Learning Outcomes: O-O Design

Demonstrate object-oriented design basics like domain models, class diagrams, and interaction (sequence and communication) diagrams.

http://enterprisegeeks.com/blog/2009/07/

- Look ahead to finishing the term strong
- Discuss more Domain Model Refinements
- Design Studio with Team 2.4

Final Exam

- 8:00am on Wednesday, Feb. 23rd
  - Room G317
- Exam is *Optional*
  - If you don't take the exam, we'll use your first exam grade as your final exam grade
  - Sign-up for exam by Tuesday of 10th week
    - If you sign-up, you must take the exam
    - Taking the exam can improve or lower your grade

Design Studio Calendar

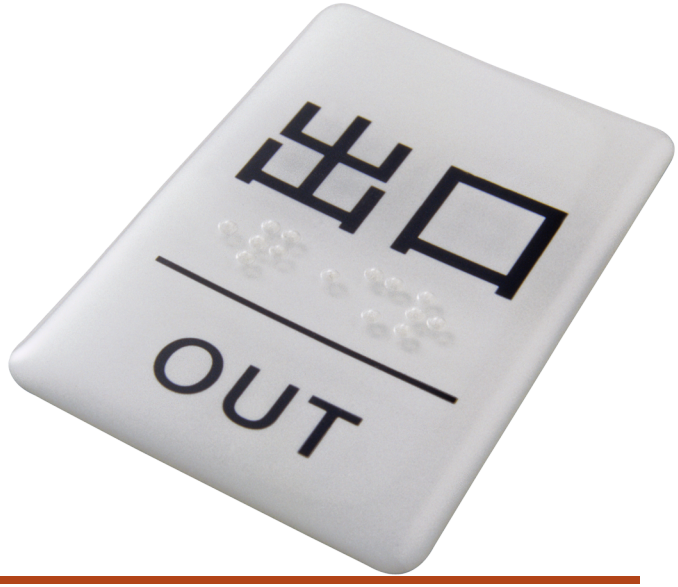
	Monday	Tuesday	Thursday
8th week	Yesterday	Today Team 2.4	Team 2.1
9th week	Team 2.2	Team 2.3	Team 2.5
10th week	Team 2.4	Team 2.1	Course Wrap-up

Potential Design Studio Topics

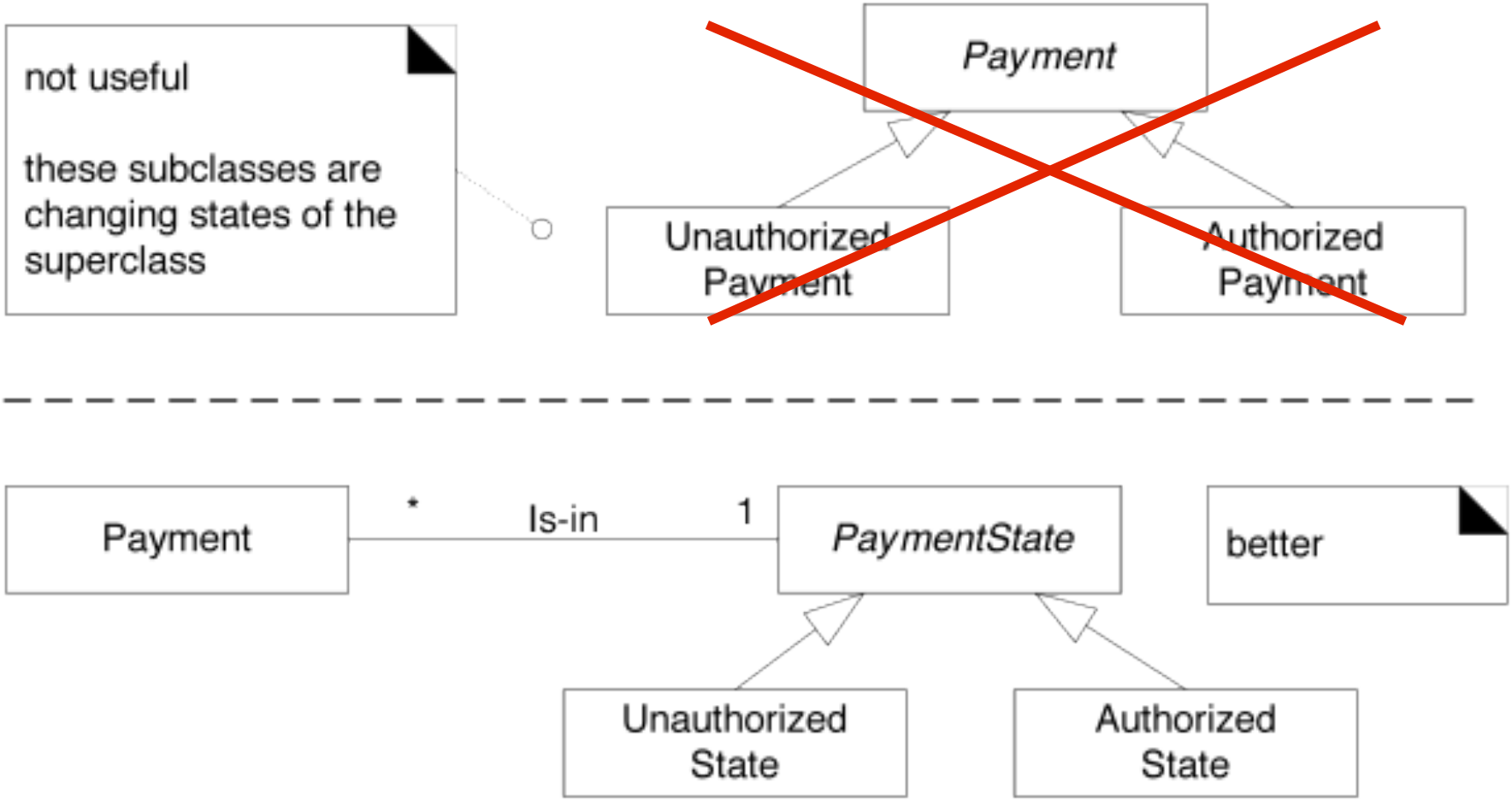
- Look for problems where there are lots of "which class should be responsible for X" questions
- Challenging design problems you've already faced and how you solved them
- Current design problems that you haven't solved yet
- Future extensions that will need to be considered

Modeling Changing States

- Suppose a concept X has multiple states
  - Draft vs. Sent Email
  - Purchase Request vs. Order
- Do not model the states as subclasses of X!
- Instead:
  - Define a State hierarchy and associate X with State
    or,
  - Ignore showing the states in the domain model



State Example



Association Classes: Consider…

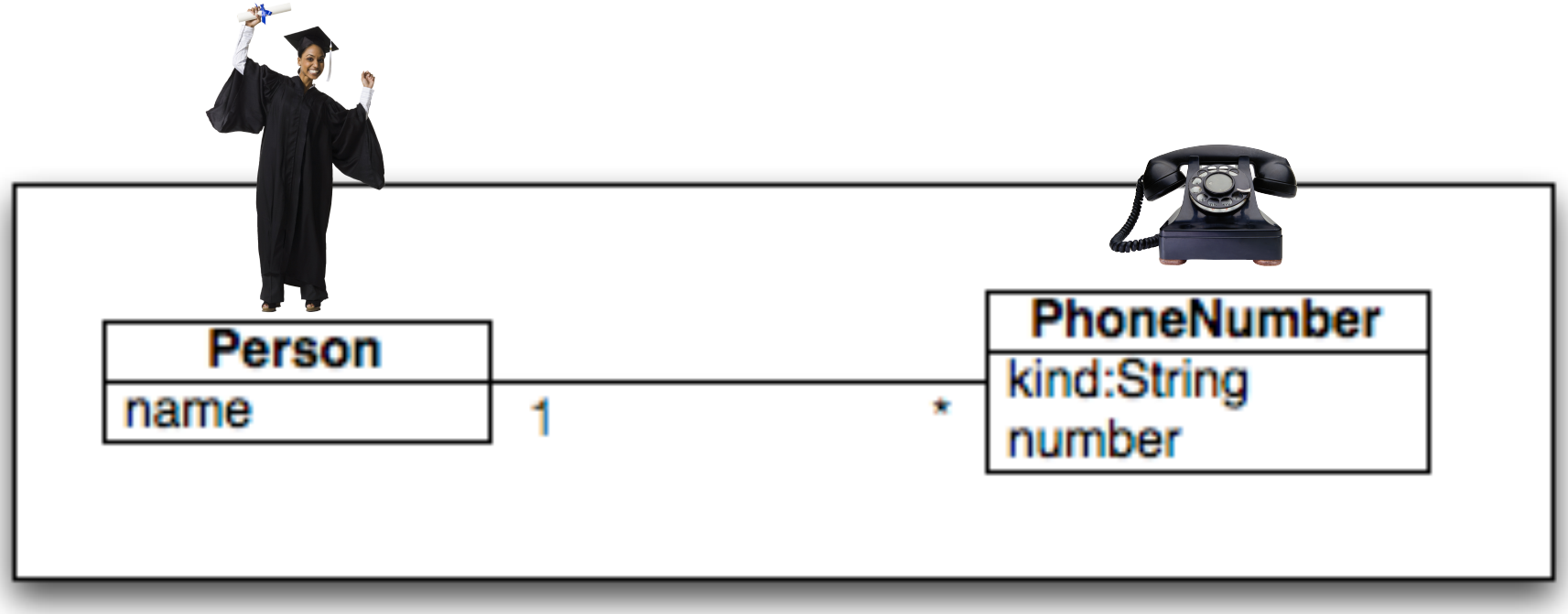
In NextGen POS:

- Authorization Services assign a *merchant ID* to each store
- Payment Authorization Request from store to service must use the *merchant ID*
- A store has a different *merchant ID* for each service

Where should the *merchant ID* appear in the domain model?

Guideline

If a class C can simultaneously have many values for the same attribute A, put A in another class associated with C

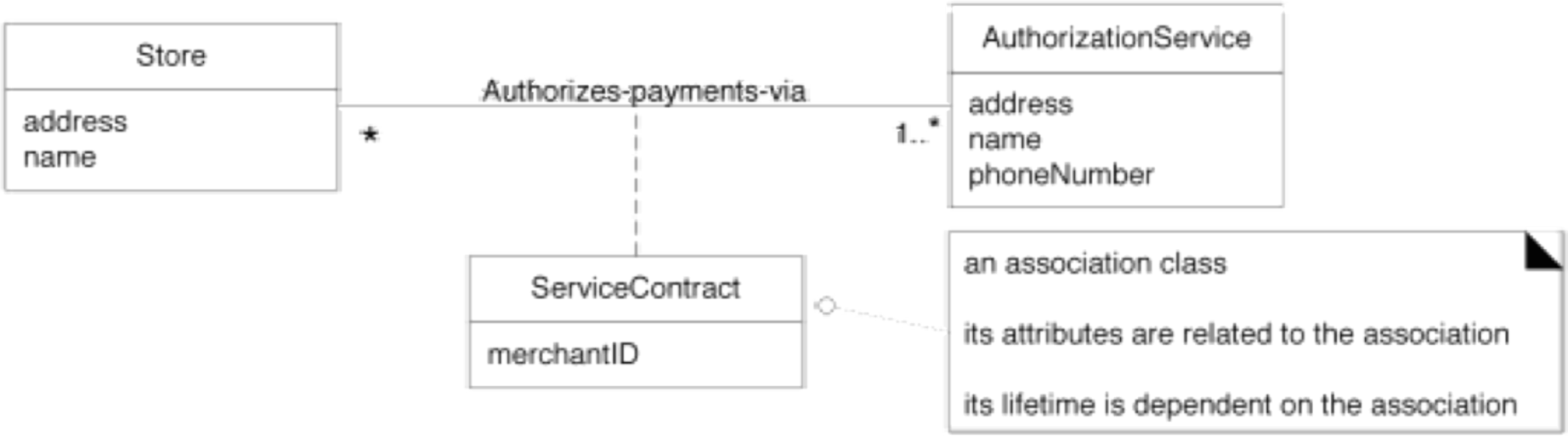


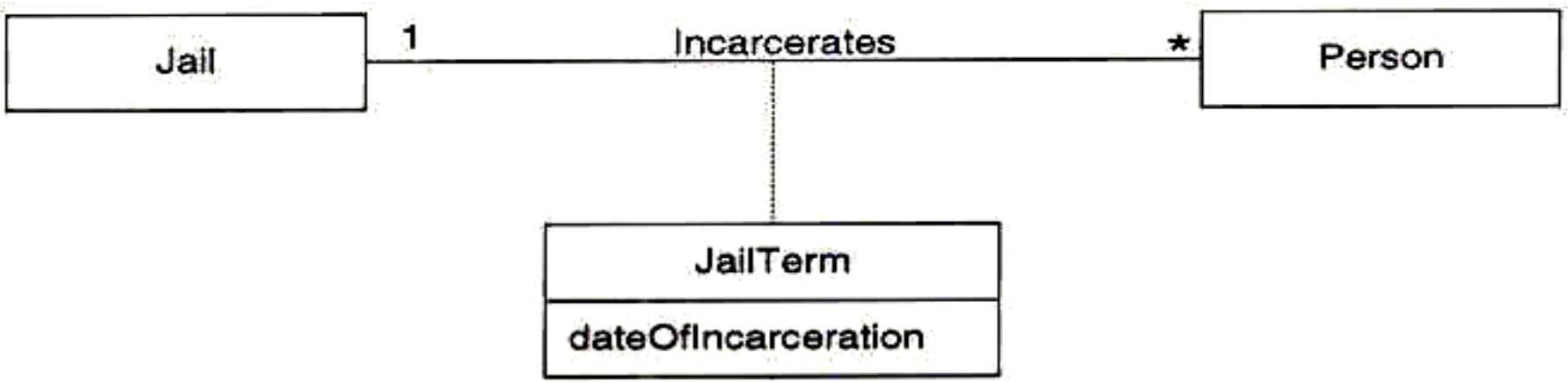
Association Class Guideline

Association class might be useful in a domain model if:

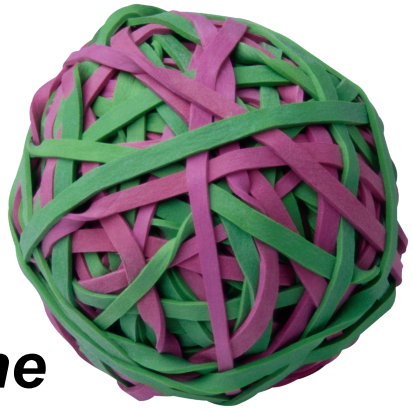
- The association has a **related attribute**
- Instances of the association class can only **last as long as the association does**
- There is a **many-to-many association** between two concepts and information is needed to distinguish the pairs

Association Class Examples





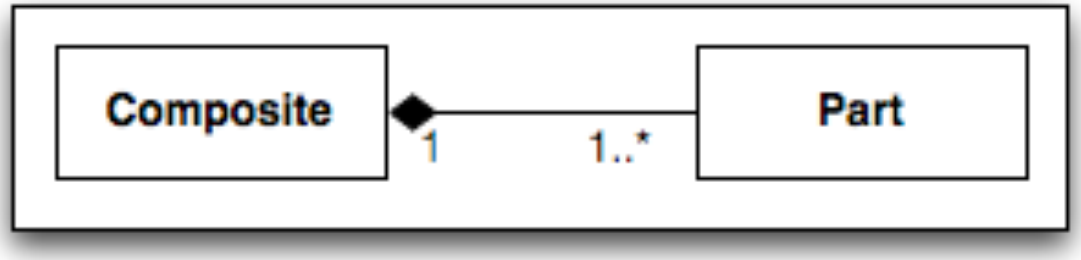
Recall, Composition



A composition relationship implies:

- An instance of the part belongs to only *one* composite instance at a time
- The part must *always belong* to a composite
- The composite is responsible for creating/deleting the parts

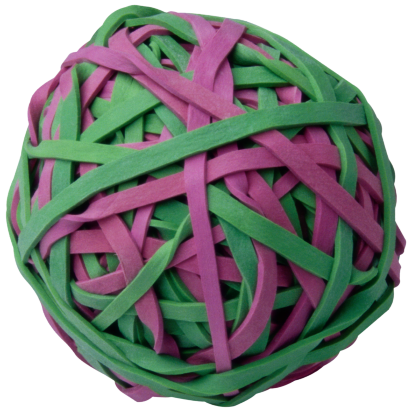
Not to be confused with the composite pattern, which is similar. (sigh)

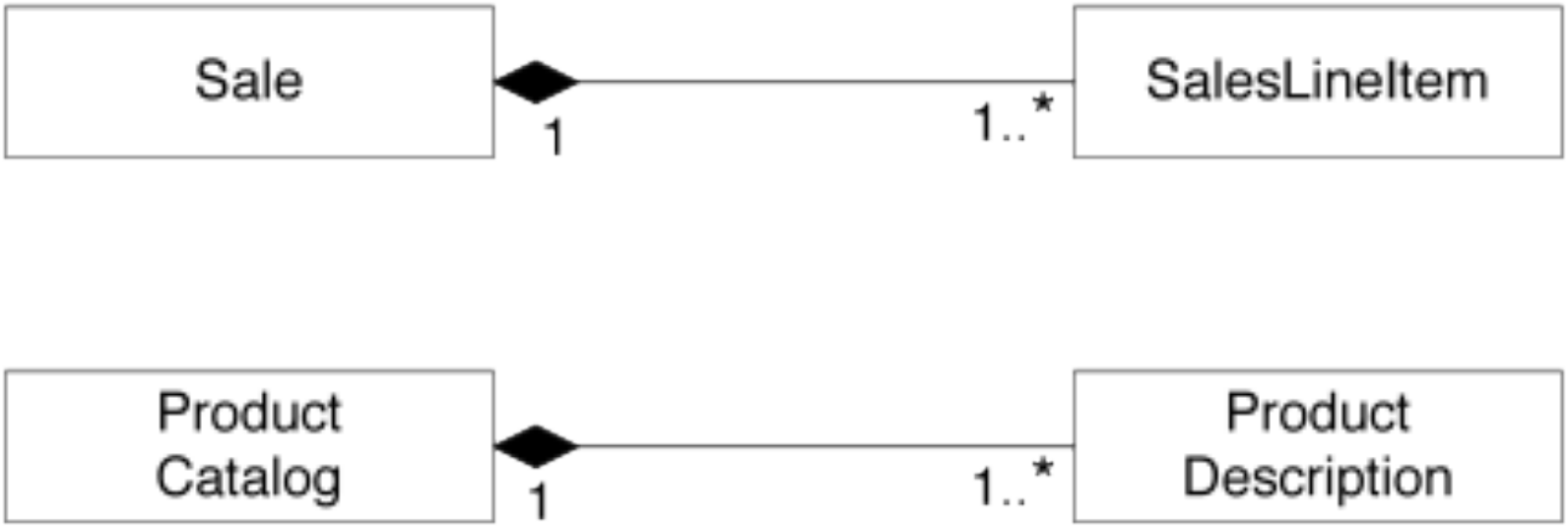


Composition in Domain Models

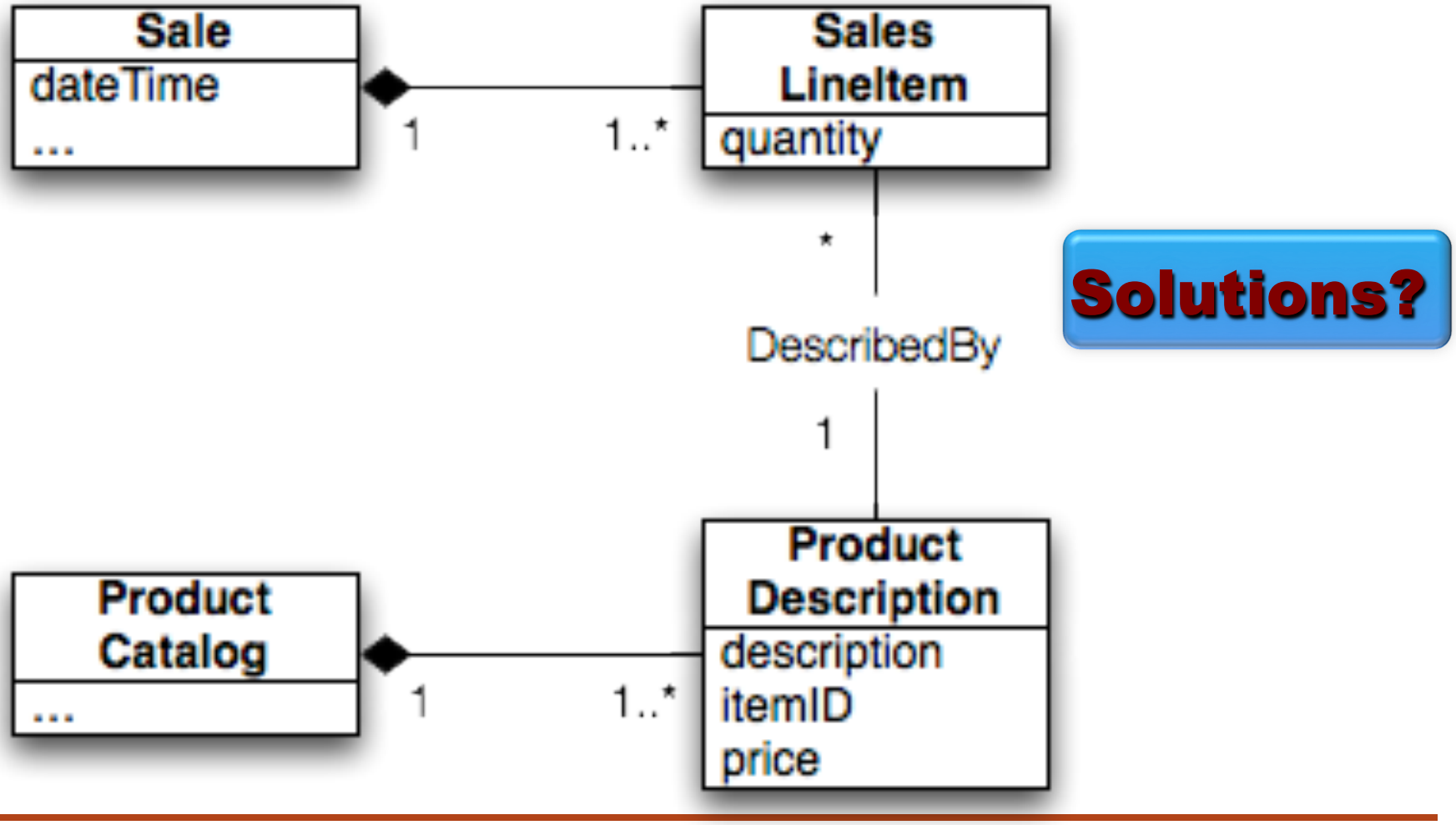
- Guideline: *If in doubt, leave it out.*
- But, consider showing composition when:
  - Lifetime of part is bounded within lifetime of composite
  - An obvious whole-part physical assembly exists
  - Composite properties propagate to the parts
  - Composite operations propagate to the parts

Examples of Composition in NextGen Domain Model

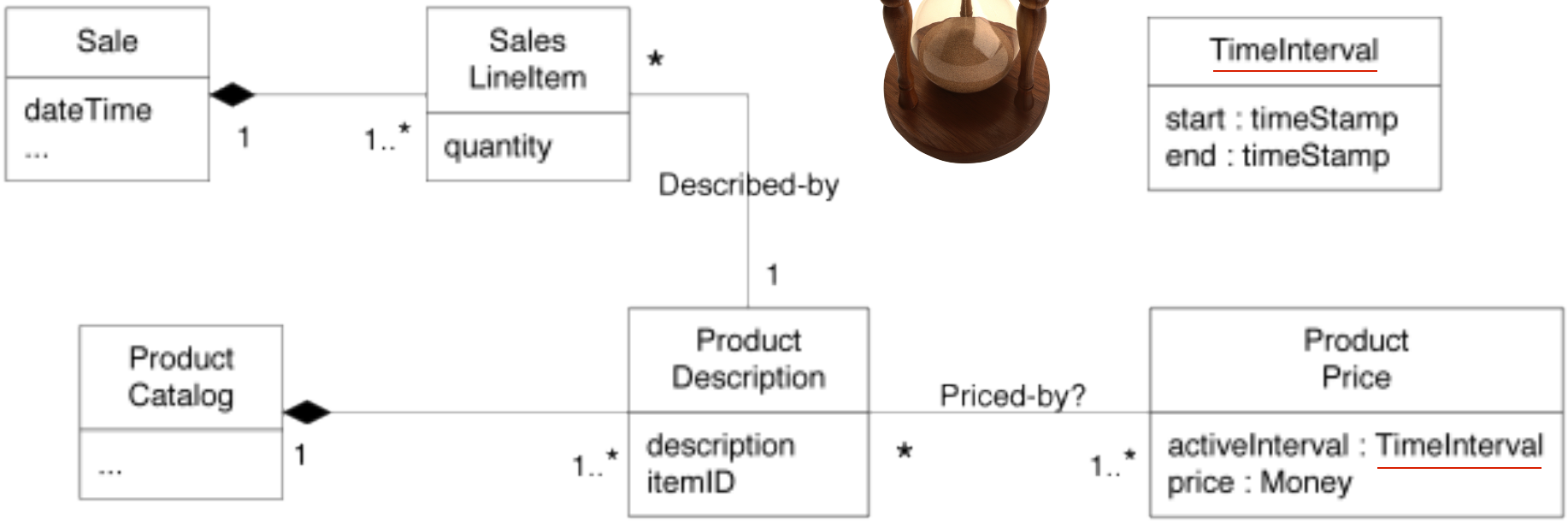




Problem: What happens to old Sales when a product's price changes?



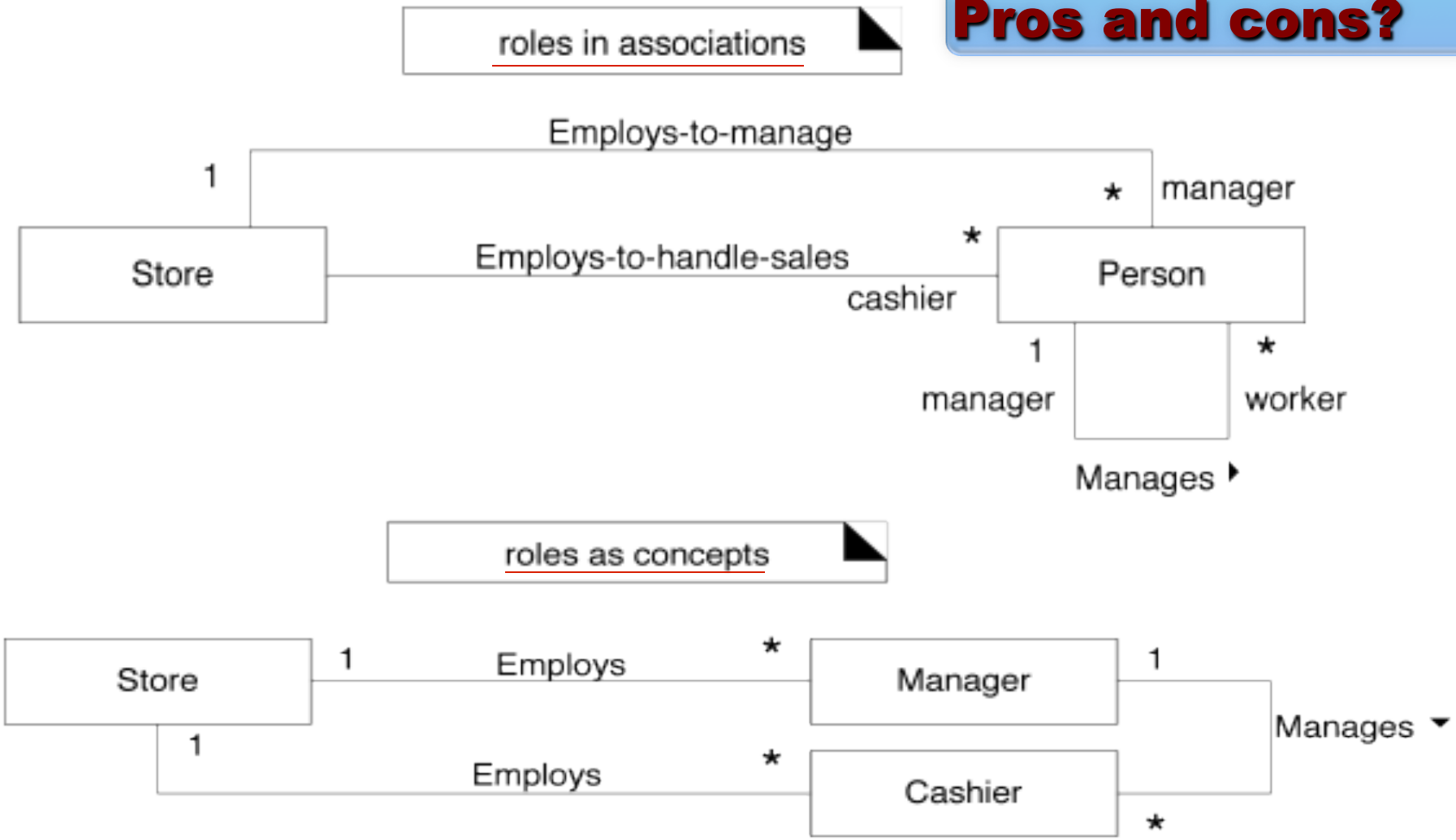
Time Intervals



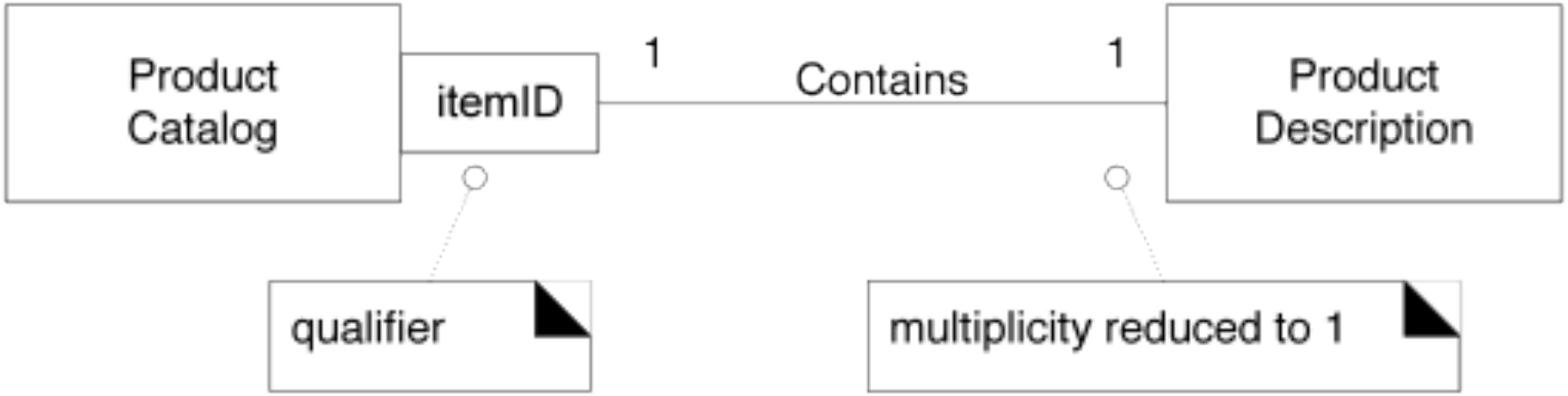
Rolls… I'm HHUUUNNNNGRY!

Er, Roles…

Pros and cons?



Qualified Associations



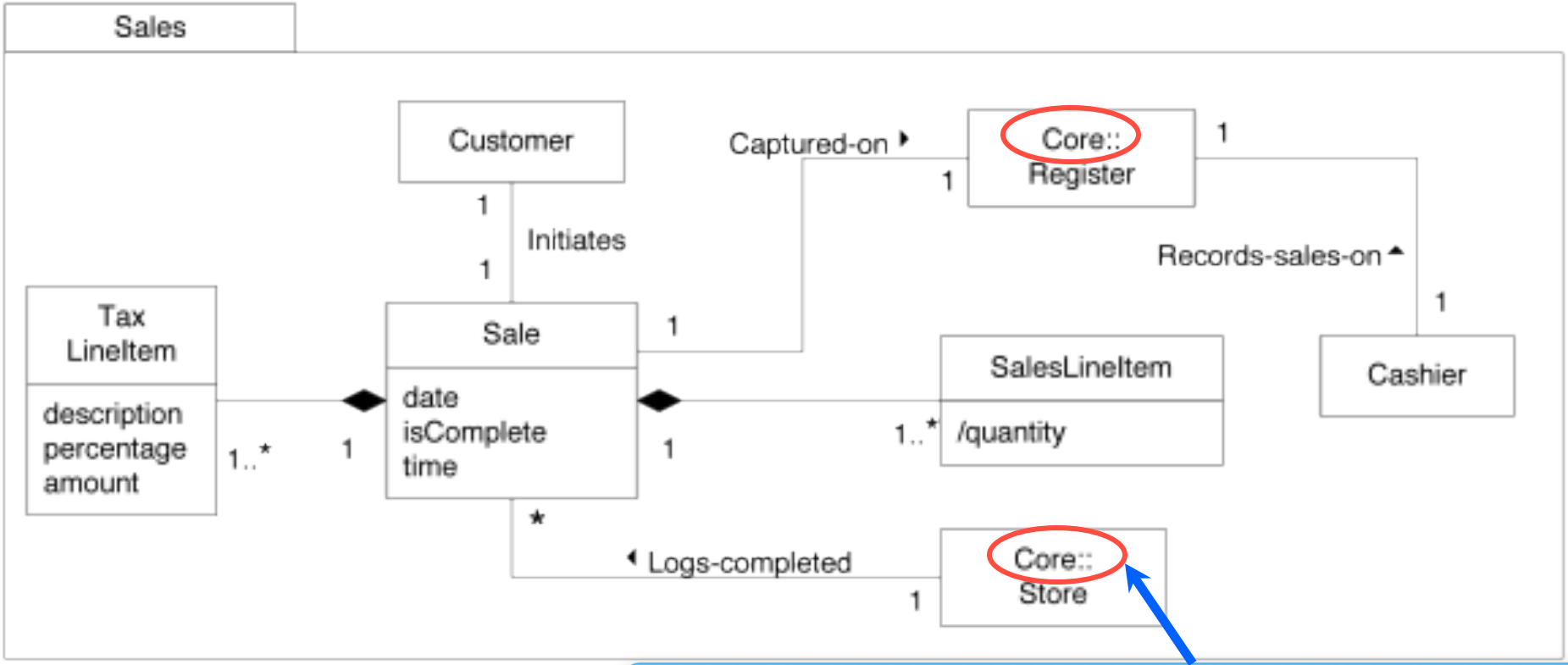
Don't overdo it...

Splitting Domain Model into Packages

- Supports parallel analysis work
- NextGen POS example:

NextGen POS Core Package

NextGen POS Sales Package with Assocations from Core Package



Store is "owned" by Core package, but also shown here to illustrate associations

If Domain is big enough to partition into packages, Group conceptual Classes that:

- Are in the same subject area
- Are in the same class hierarchy
- Participate in the same use cases
- Are strongly associated

Design Studios

Objective is to share your design with others to communicate the approach or to leverage more eyes on a problem.

- **Minute or so to set up…**
- **5-6 minute discussion**
- **1-2 minute answering questions**

1. **Team 2.4 – Observatory Tracking System**

Homework and Milestone Reminders

- Read Chapters 32 and 33
- Milestone 5 – Final Junior Project System and Design
  - Draft due by 11:59pm on Friday, February 11th, 2011
  - Final due by 11:59pm on Friday, February 18th, 2011
- Homework 6 – BBVS Design using GoF Patterns
  - Due by 11:59pm Tonight, Tuesday, February 1st, 2011
- Team 2.1 –Interactive Syllabus on Thursday