Domain Model Refinement, continued

Curt Clifton

Rose-Hulman Institute of Technology

Q1

Looking Ahead

Final Exam

- Monday, Feb. 22, 8am
- **Optional**
  - If you don't take the exam, we'll use your exam 1 grade as your final exam grade
  - Sign-up for exam during 10th week
  - If you sign-up, you have to take the exam
  - Taking the exam can lower your grade

Design Studio: An Experiment

- A chance to get more help on your projects
- A chance to participate in more design sessions without taking more of your time
- 20 minutes in each session, starting Thursday

- Format:
  - ~5 minutes: team describes problem and current solution (if any)
  - ~3 min.: class thinks about questions, alternative approaches
  - ~12 min.: on-board design

Suggestions for 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

Design Studio Sign-up

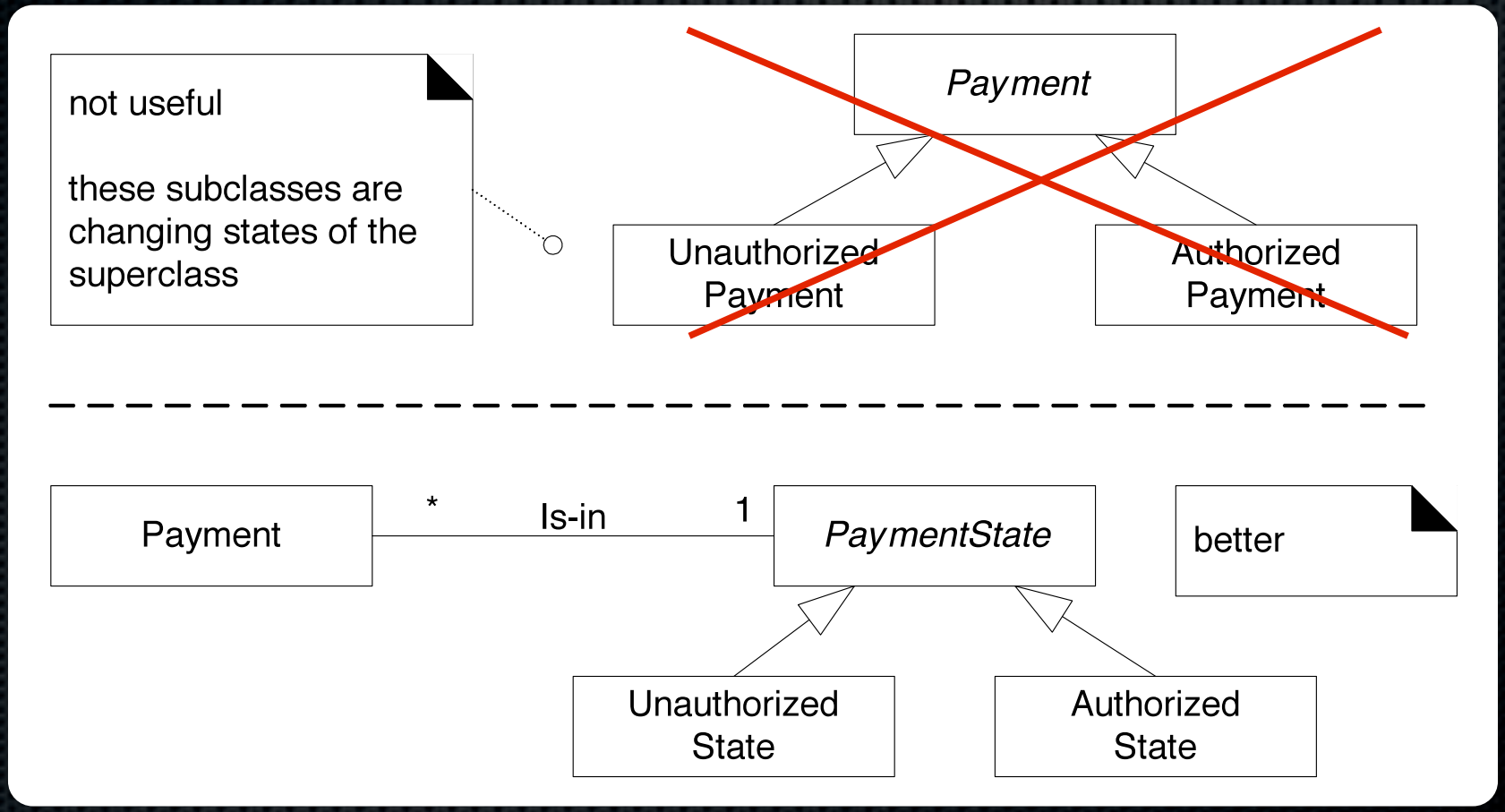
	Monday	Tuesday	Thursday
8th week	Yesterday	Today	
9th week			
10th week			Course Wrap-up

More on Domain Modeling

Modeling Changing States

- Suppose a concept X has multiple states
  - E.g., Draft vs. Sent Email
- Do not model the states as subclasses of X!
- Instead:
  - Define a State hierarchy and associate X with State
  - Or, ignore the states in the domain model

State Example



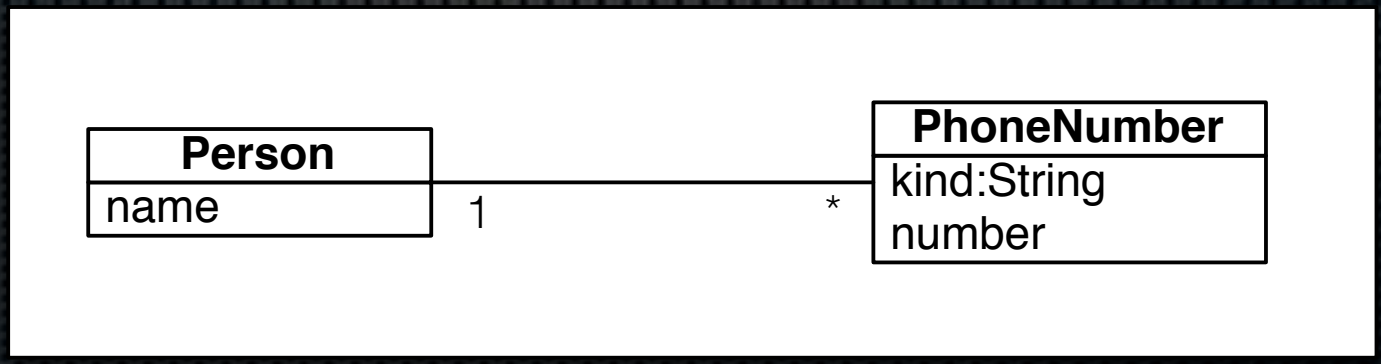
Association Classes, Consider…

- 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
- Example:

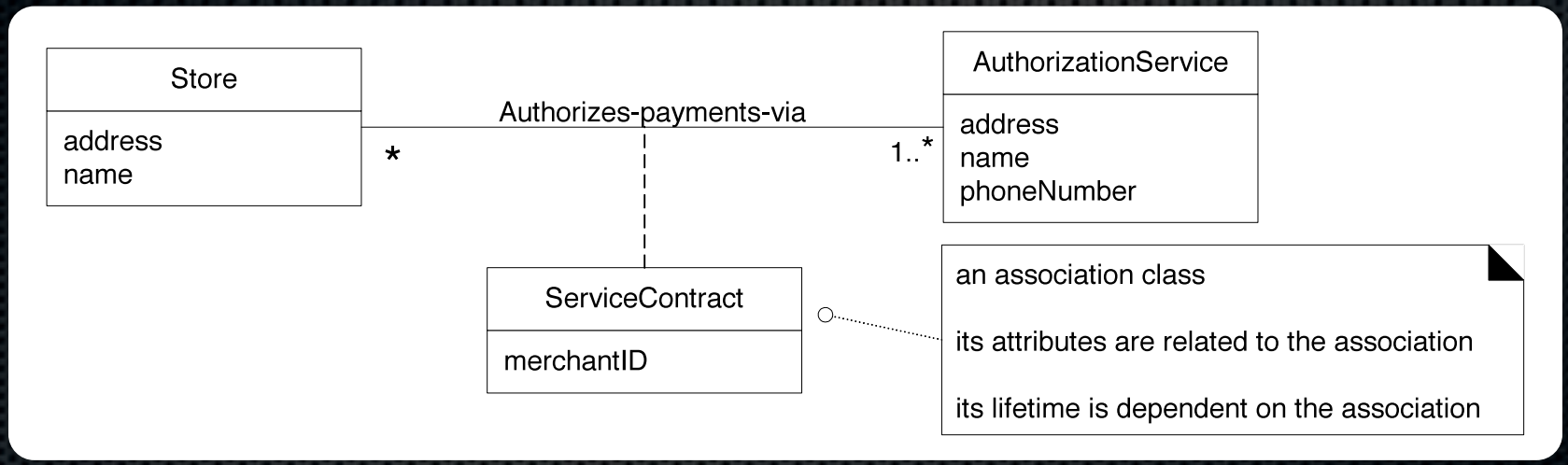


Guideline

- 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

Q2

Example Association Class

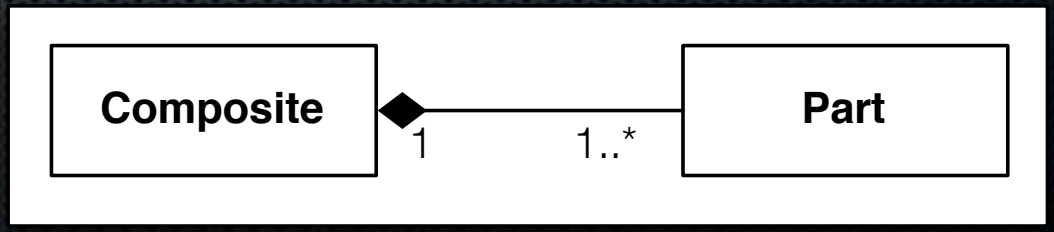


Q3

Recall, Composition

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

- 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

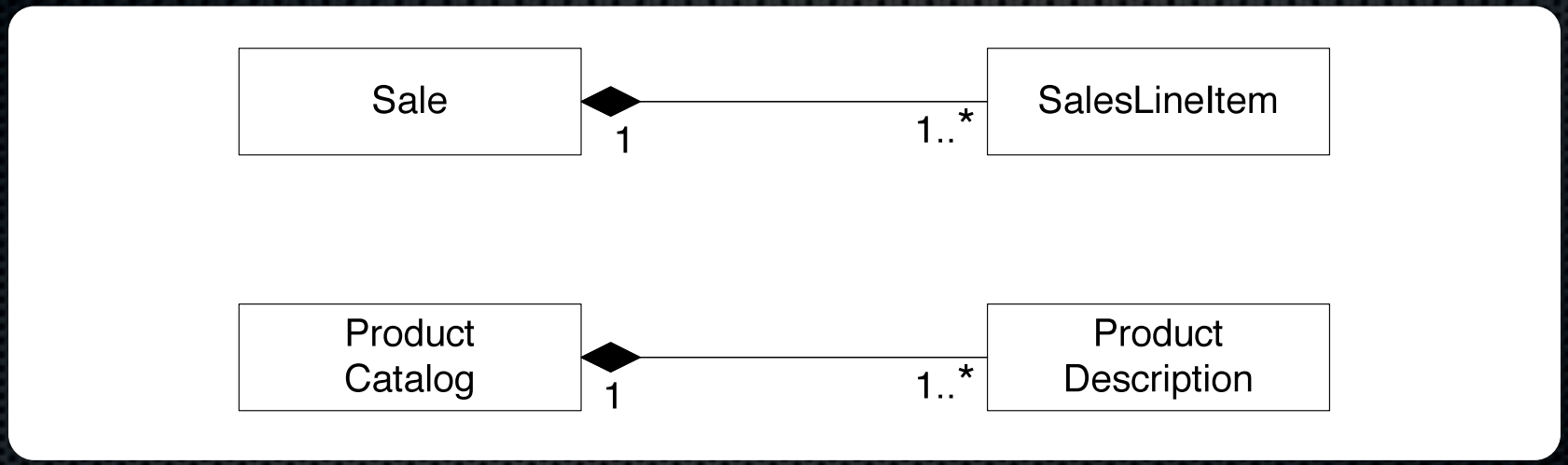


Composition in Domain Models

- Guideline: **If in doubt, leave it out.**
- But, consider showing composition when:
  - The lifetime of the part is bounded within the lifetime of the composite
  - There is an obvious whole-part physical assembly
  - Properties of the composite propagate to the parts
  - Operations on composite propagate to the parts

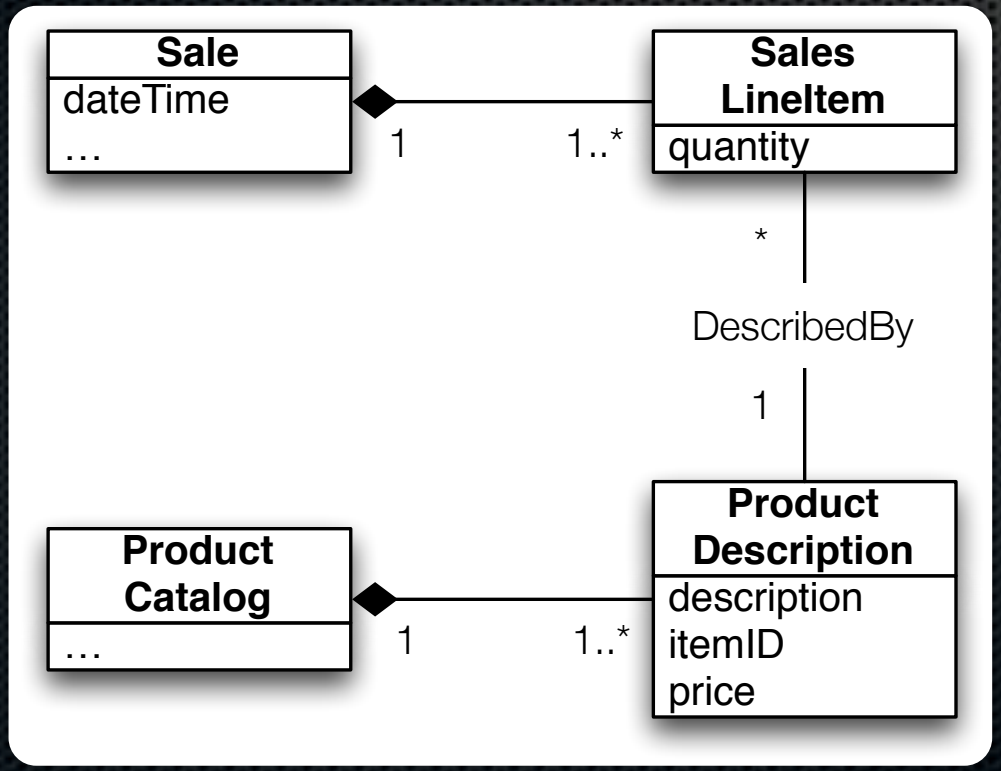
Q4

Composition in NextGen Domain Model



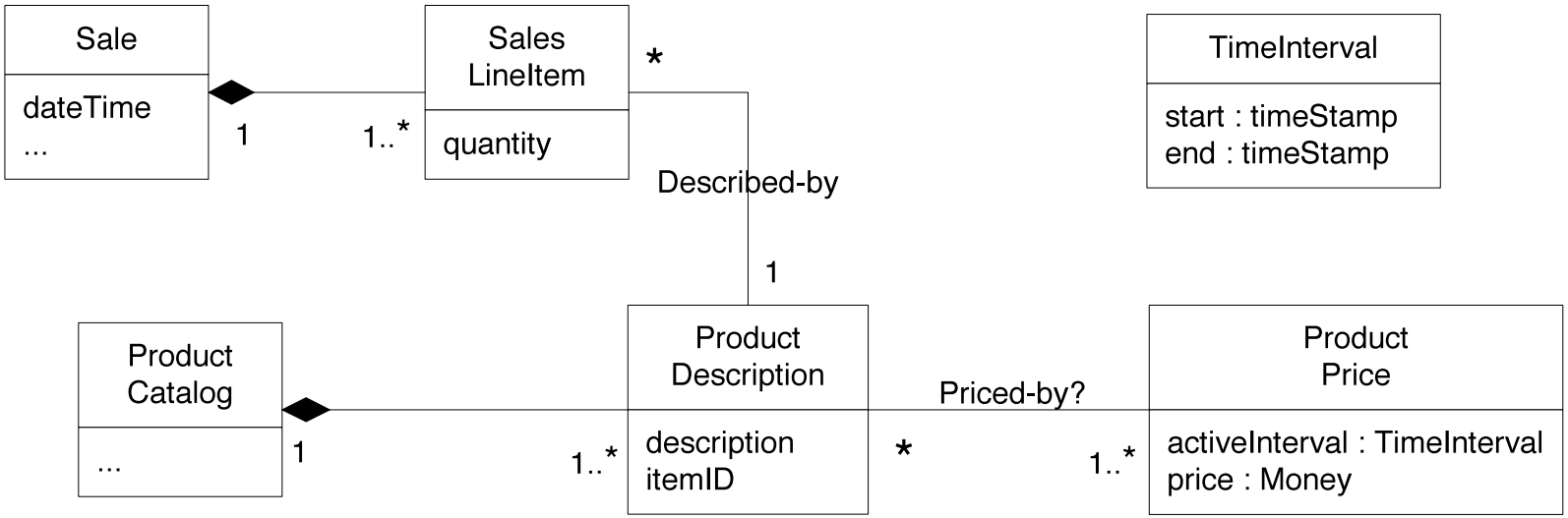
Problem

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



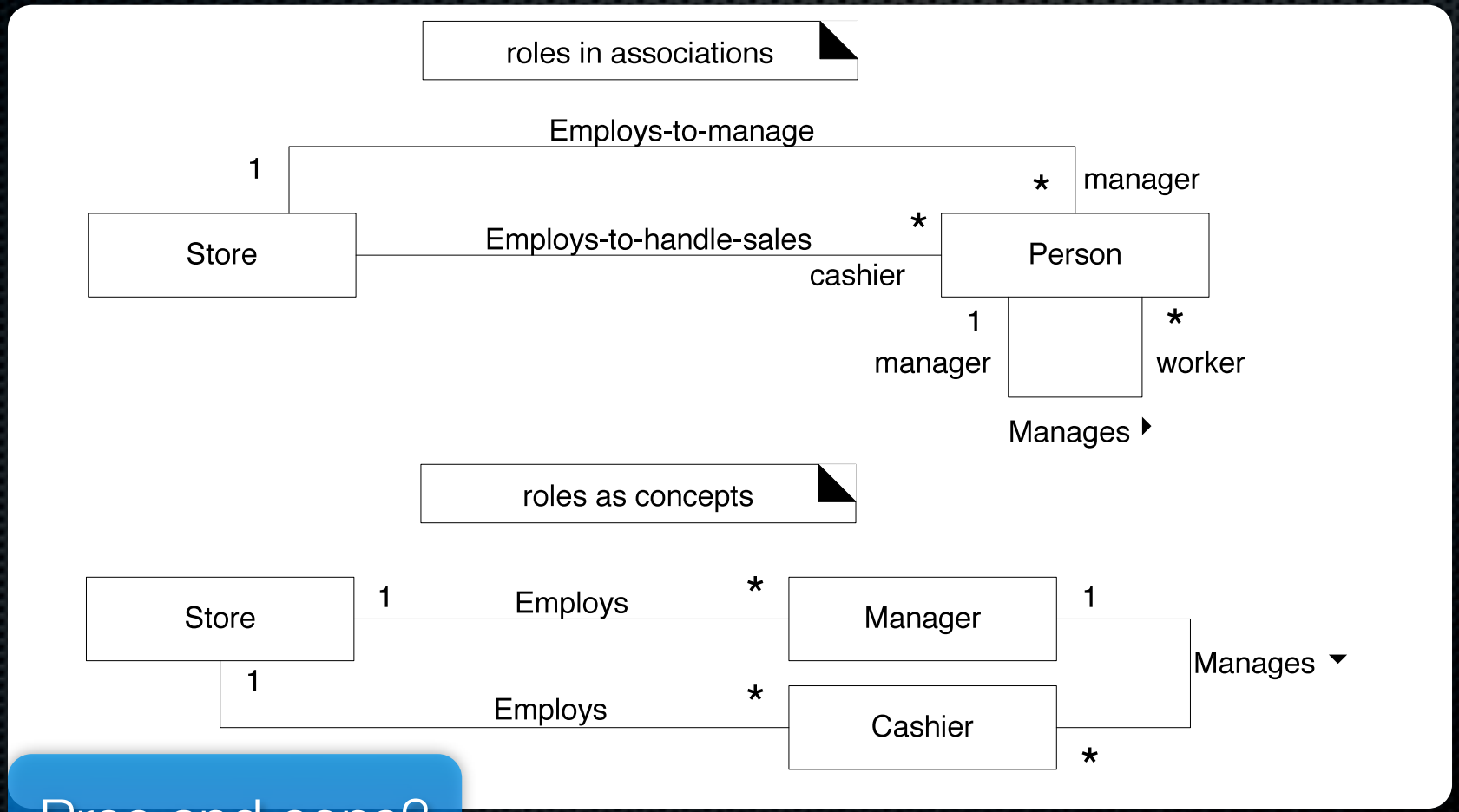
Solutions?

Time Intervals



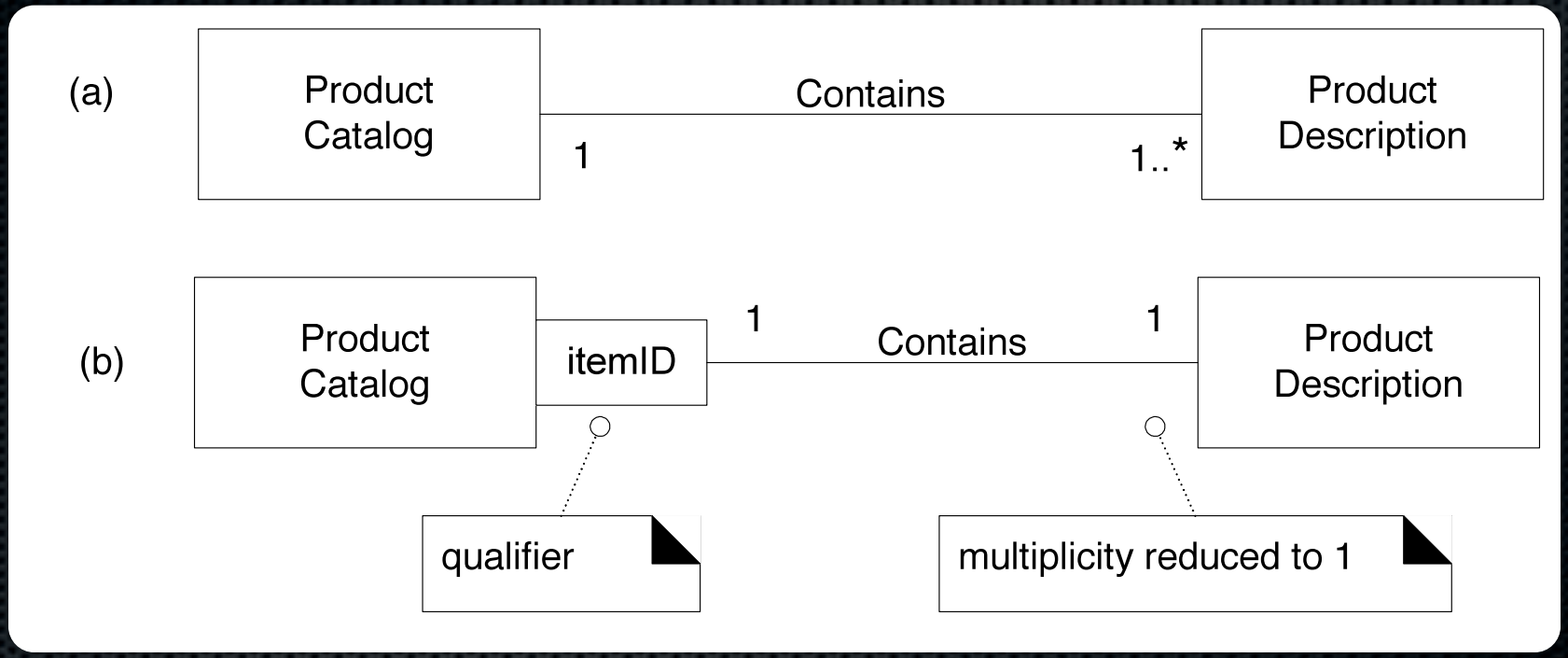
Rolls…

Er, Roles…



Pros and cons?

Qualified Associations

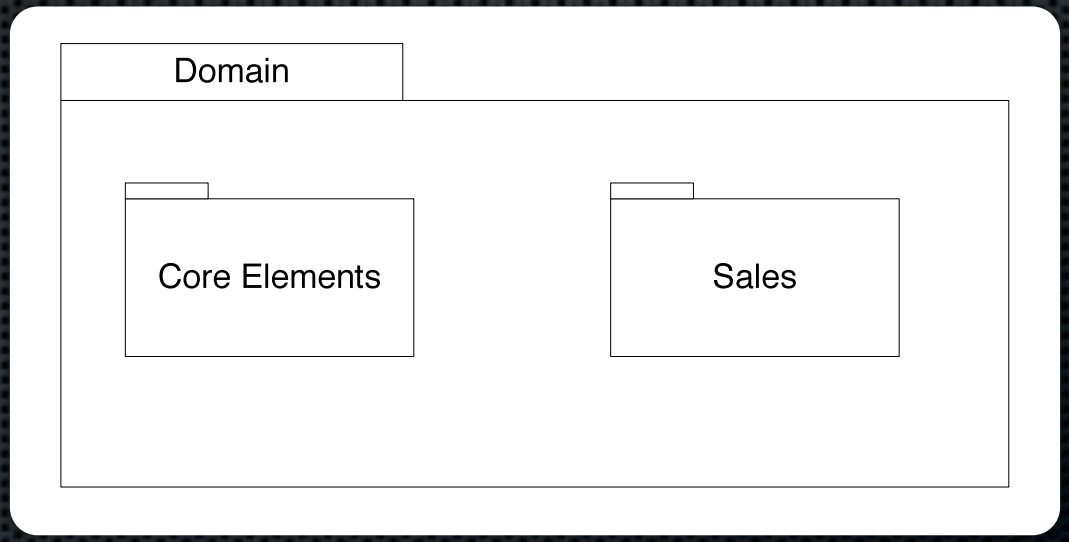


Don’t over do it

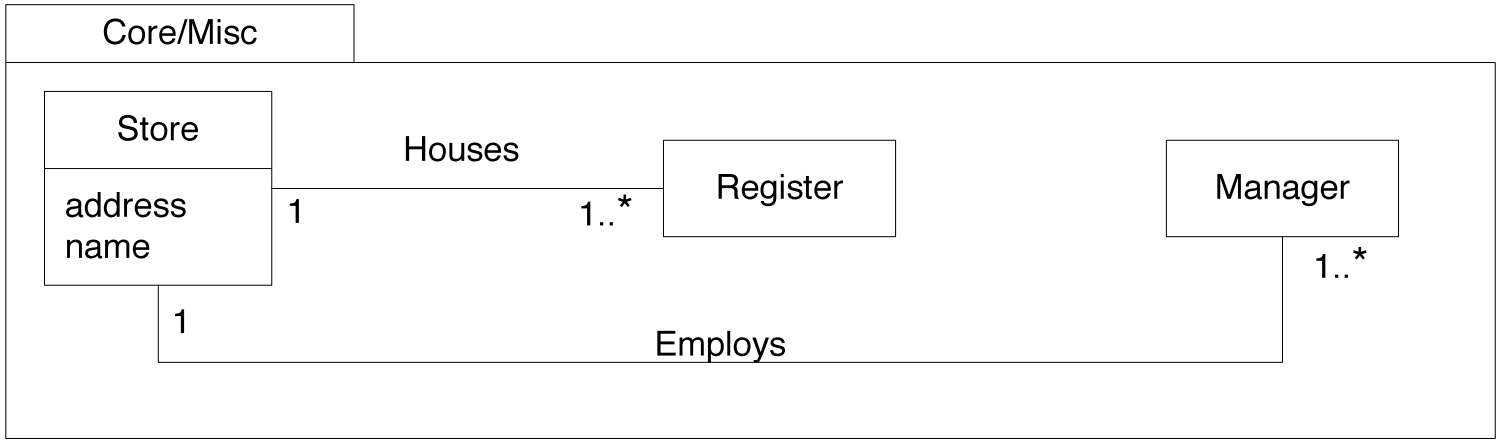
Q6

Splitting Domain Model into Packages

- Supports parallel analysis work
- NextGen POS example:

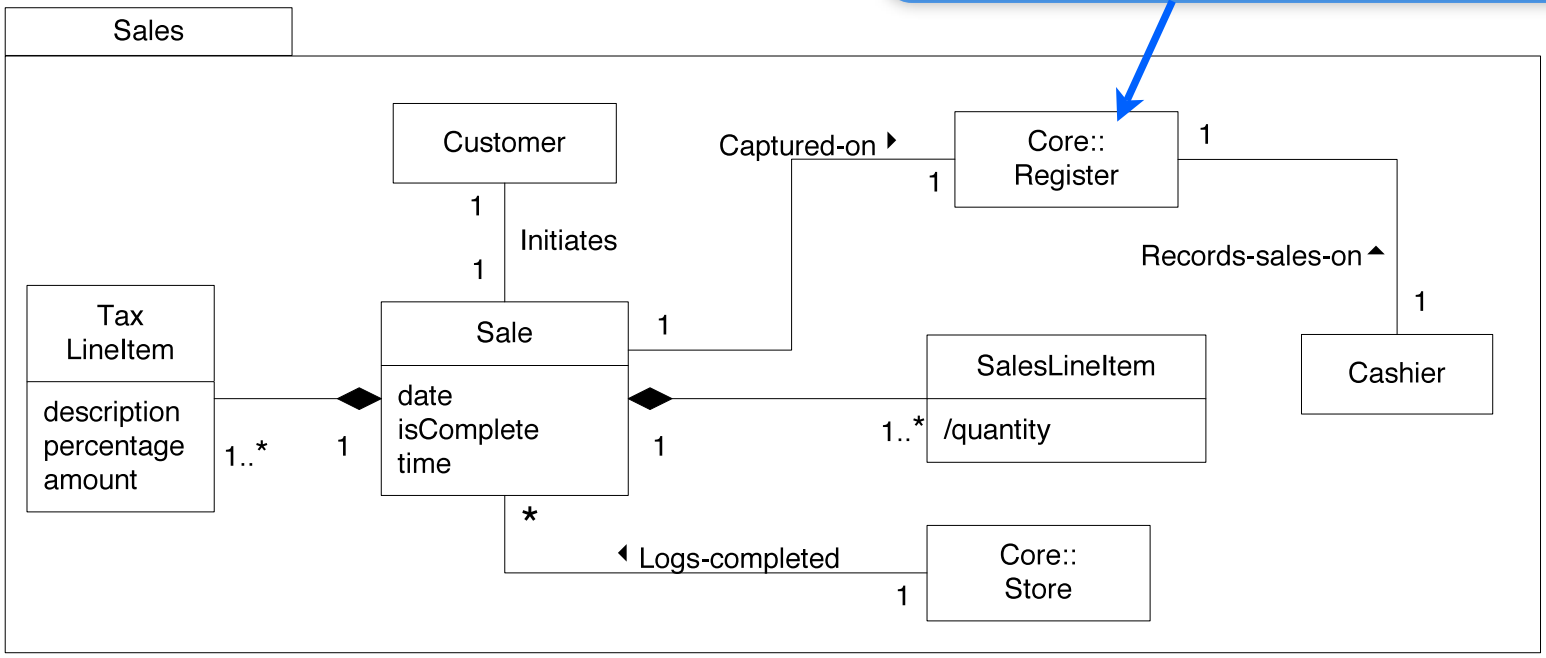


NextGen POS
Core Package



NextGen POS Sales Package

Register 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