CSSE 374: Introduction to Domain Modeling

Shawn Bohner

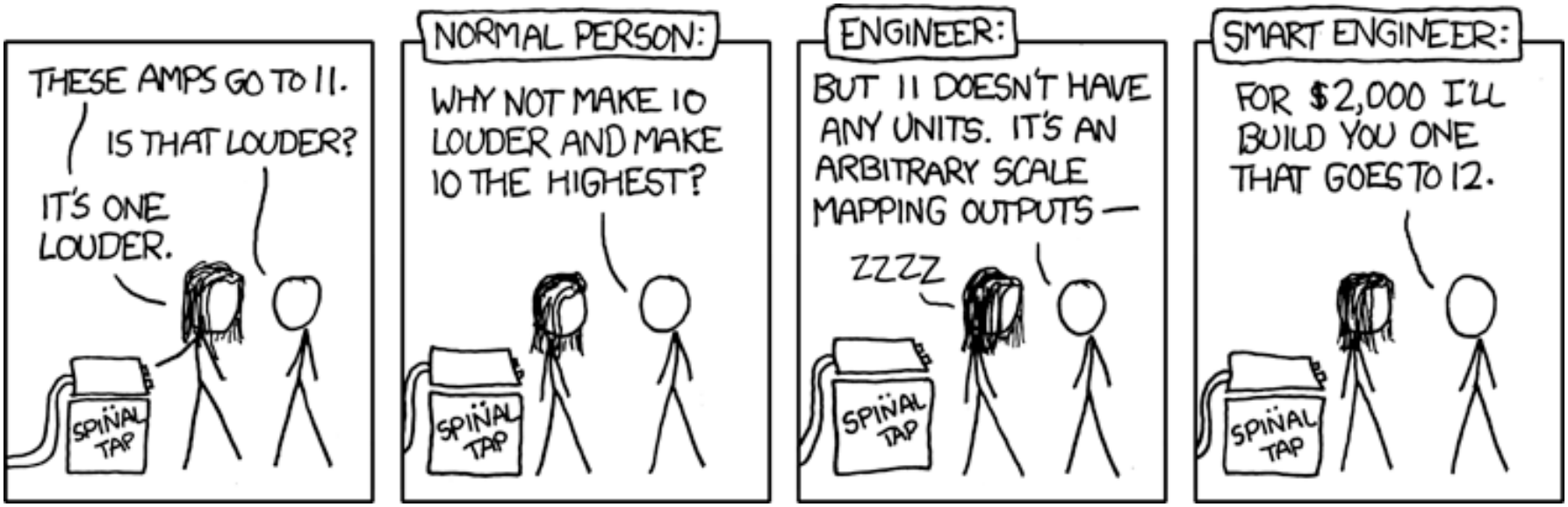
Office: Moench Room F212

Phone: (812) 877-8685

Email: bohner@rose-hulman.edu

ROSE-HULMAN INSTITUTE OF TECHNOLOGY

It's important to understand your customer's domain…



Wow, that's less than $200 per … uh … That's a good deal!

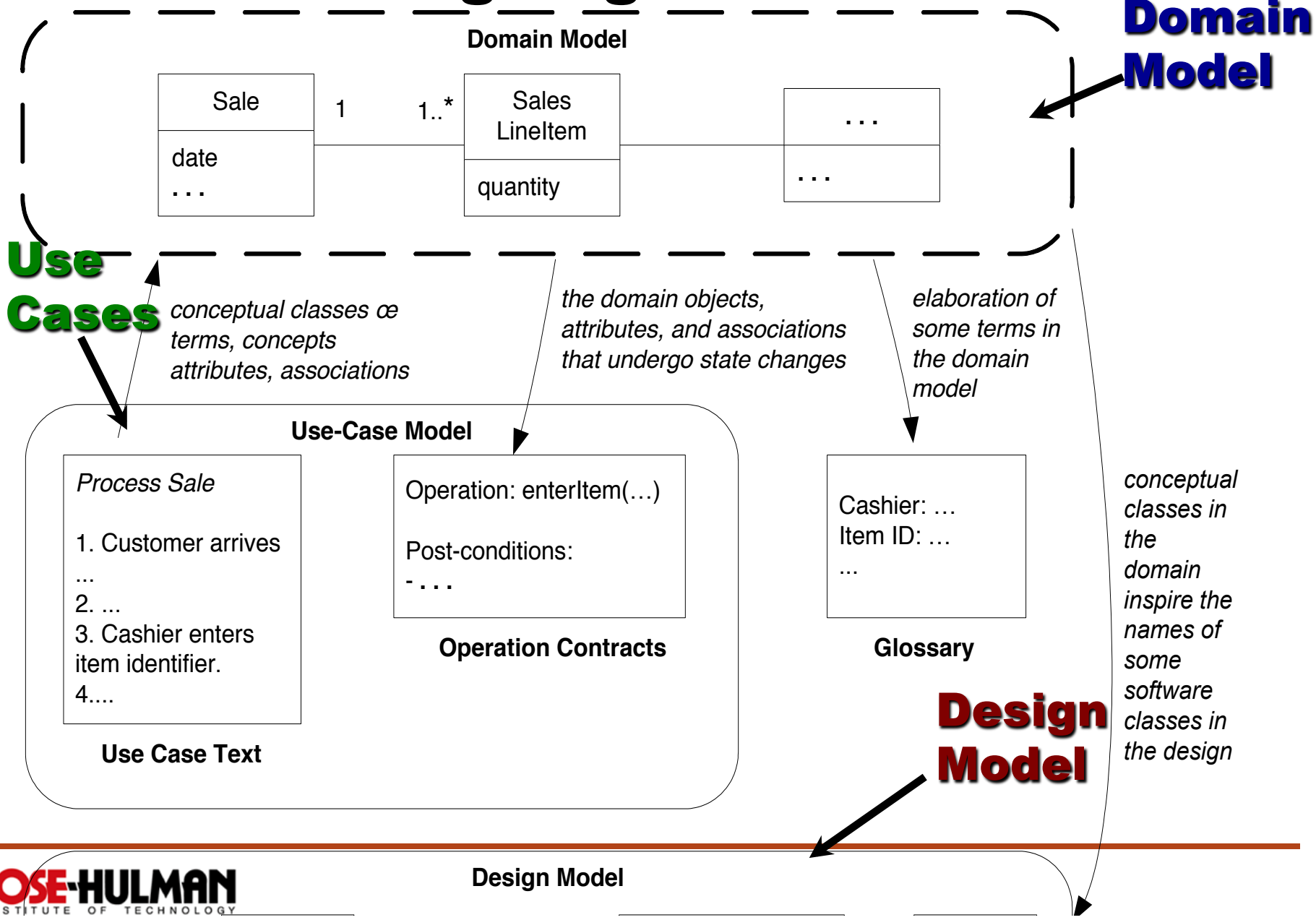
Learning Outcomes: O-O Design

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

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

- **Introduce and define Domain Model**
- **Examine Concept Classes and Description Classes**
- **Outline some domain modeling guidelines**

Where we're going...

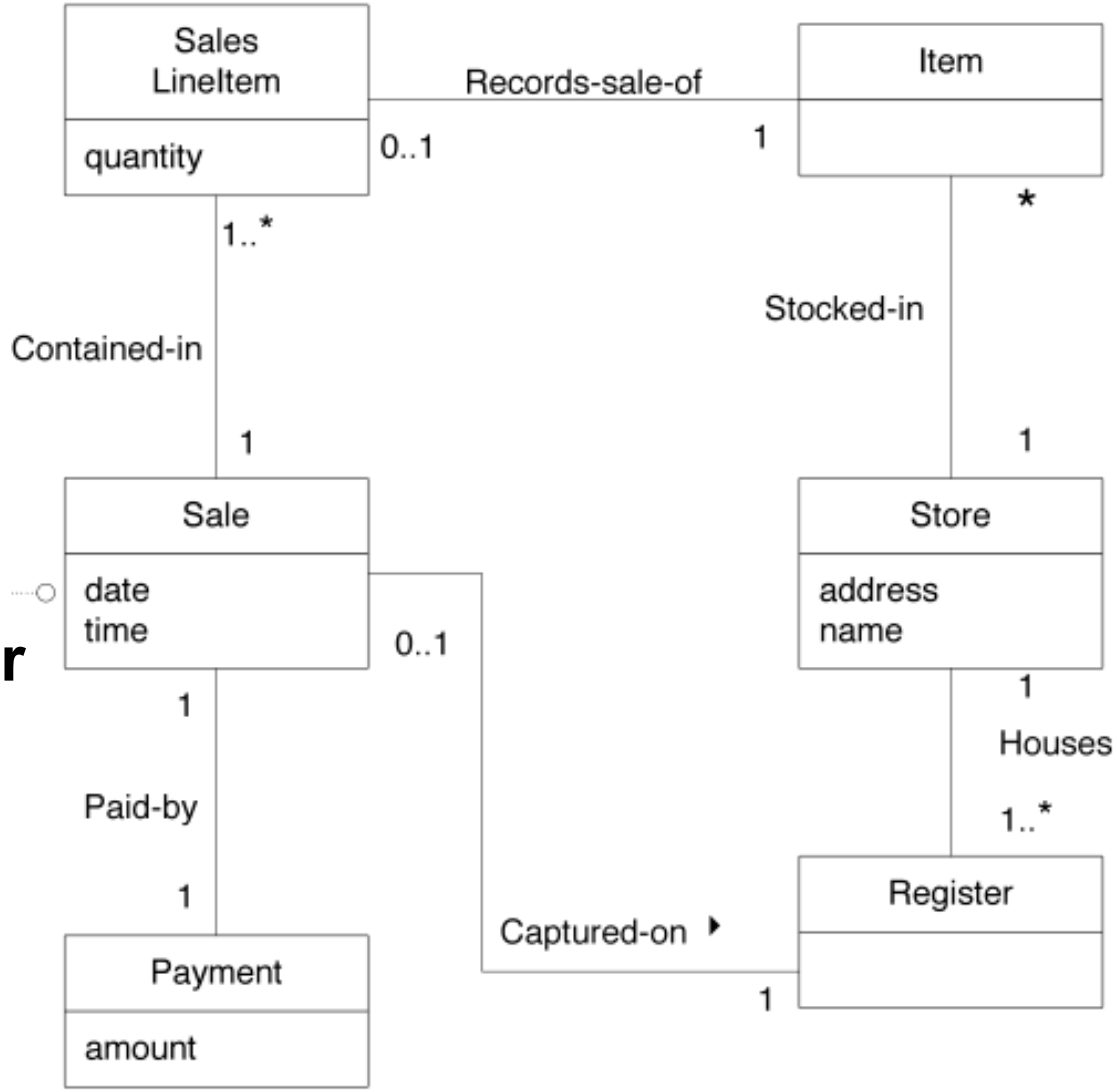


How do you relate the problem domain to the design (i.e., where do names for objects come from and the relationships between them)?

- Again, think for 15 seconds…
- Turn to a neighbor and discuss it for a minute

What is a Domain Model? 1/2

- Key Object-Oriented <u>Analysis</u> Model
- Abstraction of Conceptual Classes
- Illustrates noteworthy domain <u>concepts</u>
- Provides inspiration for naming design objects
- Notation trivial, but it takes practice to build a useful model

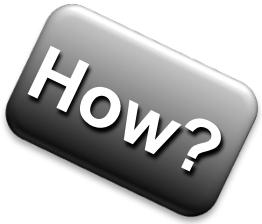


What is a Domain Model? 2/2

- <u>Visual</u> representation of conceptual classes and their relationships
- Focuses on one domain
- Illustrated using UML class diagrams without operations

Why Create a Domain Model?

- Domain Model Easier for Users to Understand
- Names from domain model move into the domain layer in the software
- Goal: lower representational gap
  - Think like a mapmaker!
- Helps us:
  - Understand the software
  - Maintain the software



How to Create a Domain Model

Bounded by the current requirements

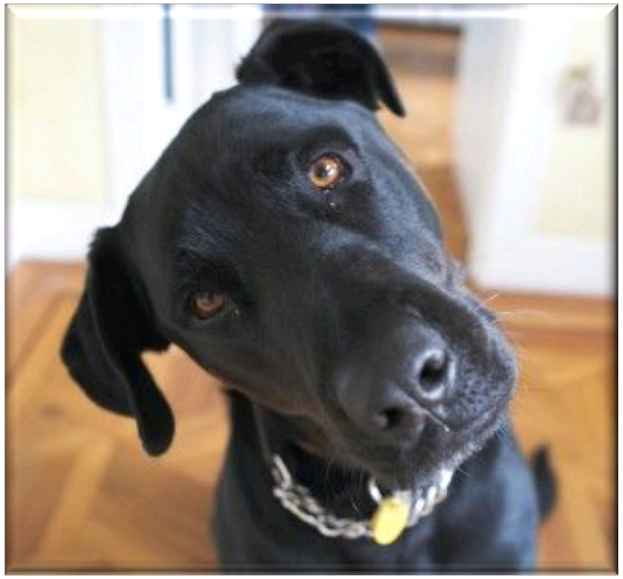
1. Find the conceptual classes

2. Draw them as classes in a UML class diagram

3. Add associations and attributes (but not operations)

Conceptual Classes

- A conceptual class is an idea, thing, or object
- Formally, a conceptual class can be represented as:
  - a symbol, it's intension, or it's extension



- Rules of thumb
  - If it takes up space, then it is probably a conceptual class
  - If you can't think of a thing as a number or text, then it is probably a conceptual class

Conceptual Class, More Formally

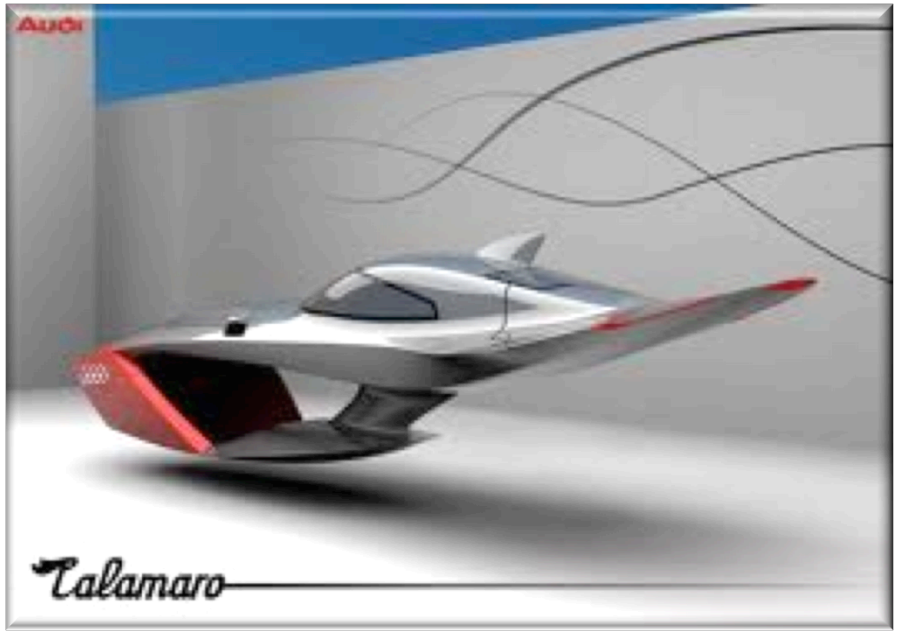
	Example	Not Just an OO Idea
Symbol	Sale date time	$\mathbb{N}$
Intension	"A sale represents the event of a purchase transaction. It has a date and time."	$\{x \in \mathbb{Z} \mid x \geq 0\}$
Extension	*sale-1, 198702131914* *sale-2, 19691216084212* *sale-3, 19920815202830*	$\{0, 1, 2, 3, \ldots\}$

Strategies to Find Conceptual Classes

1. Reuse or modify existing models

2. Identify noun phrases; linguistic analysis

3. Use a category list



Category Lists for Conceptual Classes

Conceptual Class Category	POS Examples
Business transactions *Here's where the $ is!*	Sale, Payment
Physical objects *Important for control systems, simulations*	Item, Register
Containers of things	Store, Aisle, Bin
…	…

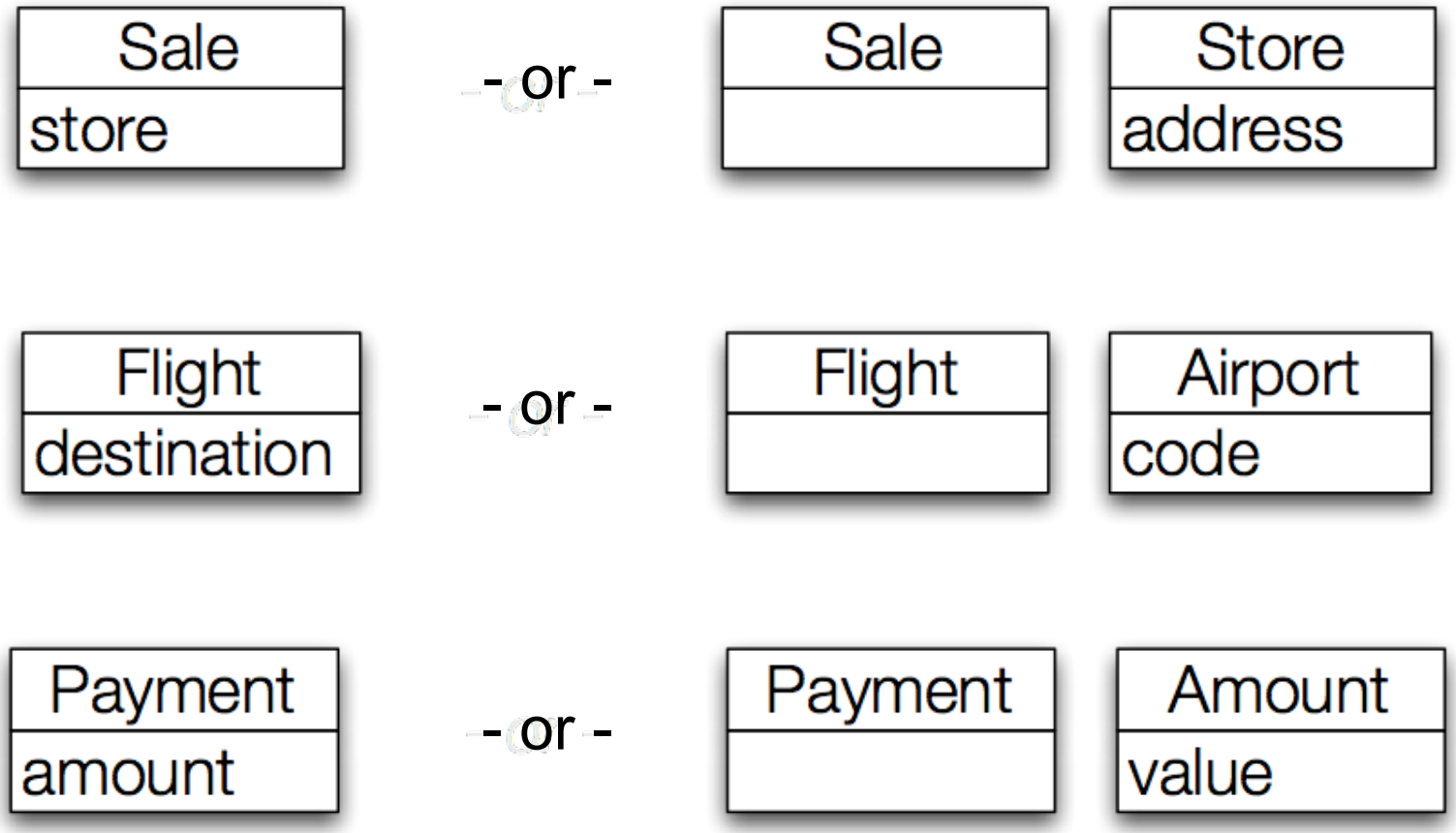
How not to be sensitive about design reviews…

Modeling the Unreal World

- Some domains are inherently abstract
  - Telecommunications
  - Server Management
  - Log File Analysis

- Guideline: listen carefully to the vocabulary and concepts used by the domain experts

Attribute or Class?



Description Classes

- A description class contains information that describes something else

- *For example, ProductDescription*

Consider…

Item
description price serial number itemID

- Assume an *Item* instance represents a physical item in a store
- Item data only recorded within *Item* instances
- When a real-world item is solid, we remove the software *Item* from a collection and it's garbage collected

Amps that go to 11 are sold out

How much for an Amp that goes to 11?

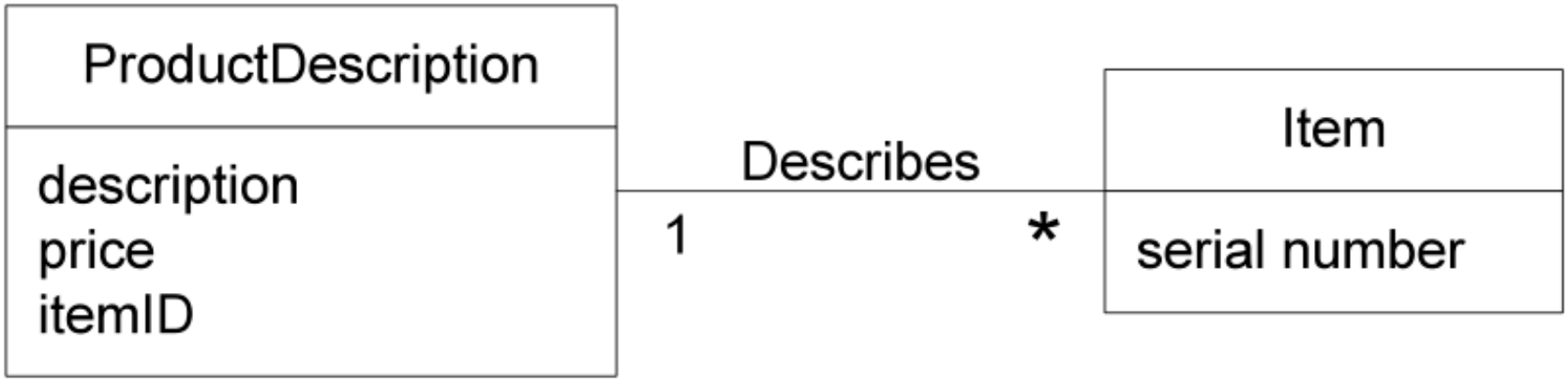
Problems

Item
description price serial number itemID

- Lose memory of the price, etc., if no *Item* instances remain in the system

- Duplicate data
  - Wasted space
  - Error-prone

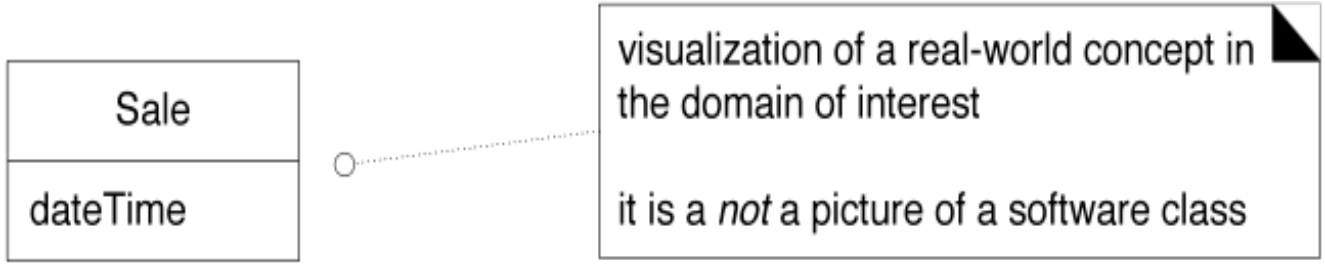
Solution: Use Description Class



- When information must be retained independent of existence of instances of the described item
- When deleting the described item could result in information loss
- When it reduces redundant information

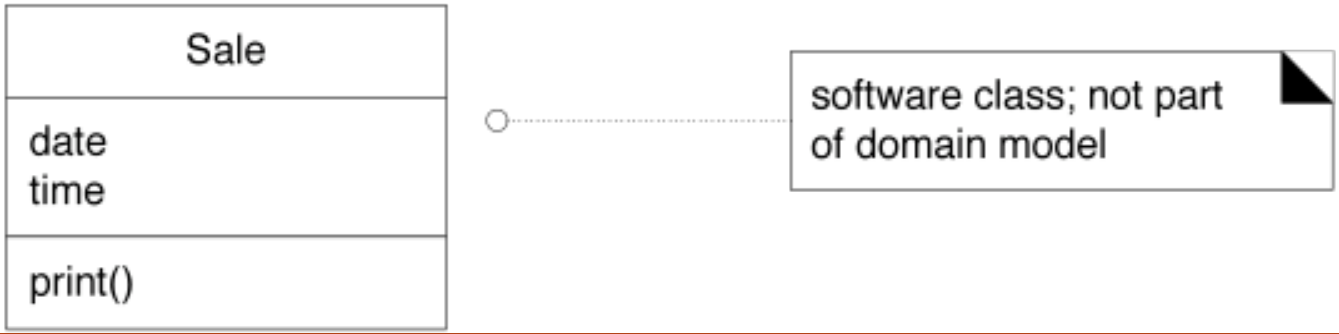
Avoid Premature Design

Good

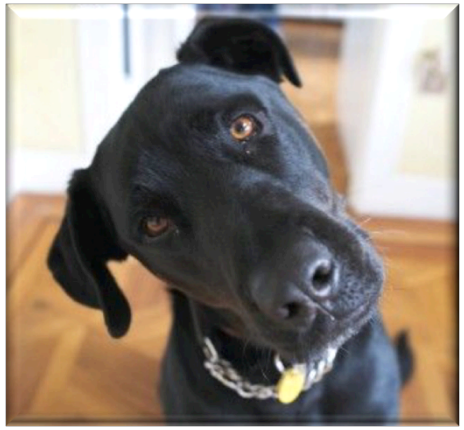


Avoid

Avoid



Confusion with Databases



- Domain model ≠ data model
- Data models:
  - Only show persistent data
  - Exclude classes that don't have attributes
- Domain models may include:
  - External actors, transient data, any real-world classes
  - Also classes without attributes/data that have a purely behavioral role

Team Information

- Team 2.1 Interactive Syllabus
  - Connor Freeman, Alec Manke, Karl Heigtbrink, Michael Frank
  - Meets with Instructor on Friday: 1st half of 3rd period (9:55am)
  - Project Manager: Eric Stokes
- Team 2.2 Rovio
  - Colin DeClue, Phil Scherer, Tim Hollingshead, Matthew Moore
  - Meets with Instructor on Friday: 2nd half of 3rd period (10:20am)
  - Project Manager: Tim Ekl
- Team 2.3 Evaluation GUI Tool
  - Eric Reed, Bryan Watts, Matt Sickler, Calvin Mlynarczyk
  - Meets with Instructor on Friday: 1st half of 7th period (1:35pm)
  - Project Manager: Tim Ekl
- Team 2.4 Observatory Control Software
  - William Lester, Ryan Fuller, Arjun Comar, Rob Adams
  - Meets with Instructor on Friday: 1st half of 4th period (10:50am)
  - Project Manager: Sam Varga
- Team 2.5 Academic Paper Cataloging System
  - William Anderson, Josh Jones, Ted Stamp
  - Meets with Instructor on Friday: 2nd half of 1st period (8:30am)
  - Project Manager: Eric Stokes

Homework and Milestone Reminders

- Read Rest of Chapter 9

- Milestone 1 – Infrastructure for Junior Project
  - Due by 11:55pm on Friday, December 3rd, 2010
- Homework 1 – Video Store Domain Model
  - Due by 5:00pm on Tuesday, December 7th, 2010
- Milestone 2 – Junior Project Domain Model
  - Due by 11:55pm on Friday, December 10th, 2010