

Domain Modeling

CSSE 374: Session 5

Shawn Bohner
Office: Moench Room F212
Phone: (812) 877-8685
Email: bohner@rose-hulman.edu



ROSE-HULMAN
INSTITUTE OF TECHNOLOGY

Q1

© 2009 Shawn A. Bohner

Description Classes

❖ A description class contains information that describes something else (e.g., *ProductDescription*)

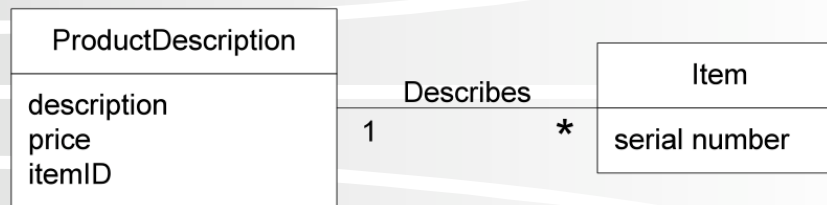
❖ Example: An *Item* instance represents a physical item in a store

- Item data only recorded within *Item* instances
- When a real-world item is sold, we remove the software *Item* goes away...
- Problem: last *Item* sold, how much for the *Item*?
- Duplication also a problem...

Item
description
price
serial number
itemID

ROSE-HULMAN
INSTITUTE OF TECHNOLOGY

Solution: Use A Description Class



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

Creating a Domain Model

- ❖ Identify Candidate Conceptual classes
- ❖ Draw them in the initial Domain Model
- ❖ **Add requisite associations to record key relationships**
- ❖ **Add attributes to preserve relevant information**
- ❖ Apply existing Analysis Patterns
 - Use existing names for things, the vocabulary of the domain
 - Exclude irrelevant features
- ❖ Rules of thumb for Conceptual Classes
 - 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.



Associations

- ❖ A relationship between classes that indicate some meaningful relationship between instances of the classes
 - Says that we need some memory of the relationship
 - A memory in the real world, not a software need
- ❖ Not about data flows, foreign key relationships, instances variables, or software pointers

Be Parsimonious

Q2

ROSE-HULMAN
INSTITUTE OF TECHNOLOGY

Association Notation

Association name:

- ✓ Use verb phrase
- ✓ Capitalize
- ✓ Typically camel-case or hyphenated
- ✓ Avoid "has", "use"

Reading direction:

Can exclude if association reads left-to-right or top-to-bottom



Multiplicity (Cardinality):

- ✓ "*" means "many"
- ✓ x..y means from x to y inclusively

Q3

ROSE-HULMAN
INSTITUTE OF TECHNOLOGY

Multiplicity (AKA Cardinality)

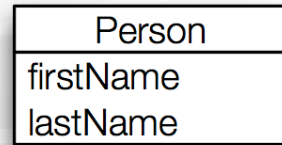
*	T	zero or more; "many"
1..*	T	one or more
1..40	T	one to 40
5	T	exactly 5
3, 5, 8	T	exactly 3, 5, or 8

Common Association Lists

Association Category	POS Examples
A is a transaction related to another transaction B	CashPayment PaysFor Sale
A is a line item of a transaction B	SalesLineItem ContainedIn Sale
A is known/logged/recorded in/on B	Sale CapturedOn Register
...	...

Attributes

- ❖ Include attributes that the requirements suggest need to be remembered



- ❖ Notation (“[]” indicate optional parts):

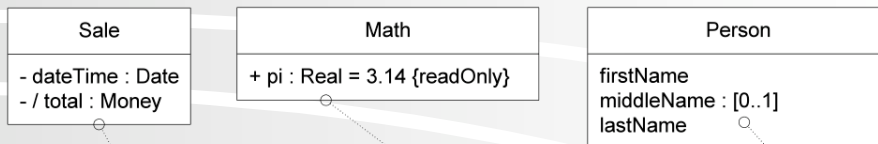
- [+|-] [/] name [: [type] [multiplicity]] [= default] [{property}]

Visibility

Derived

e.g., readOnly

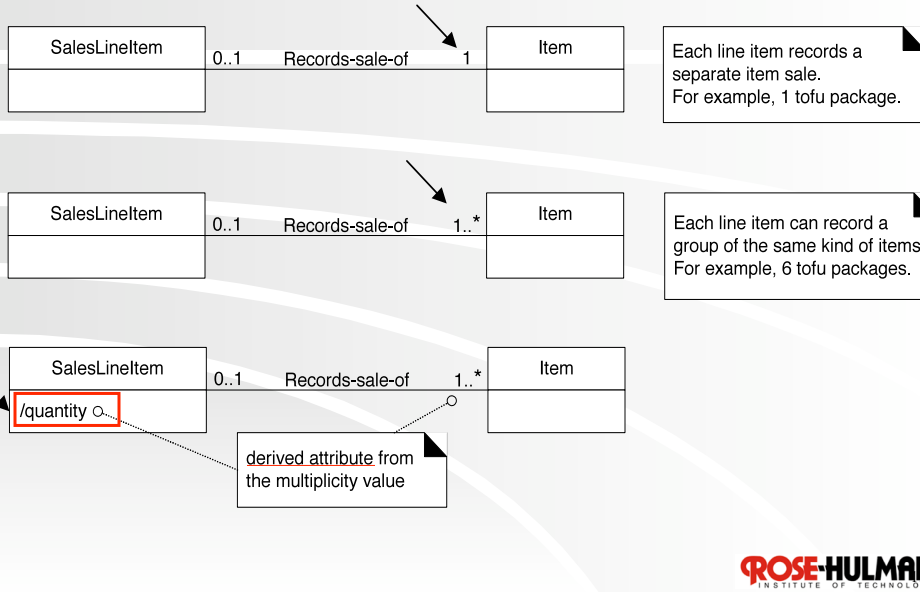
Attribute Examples



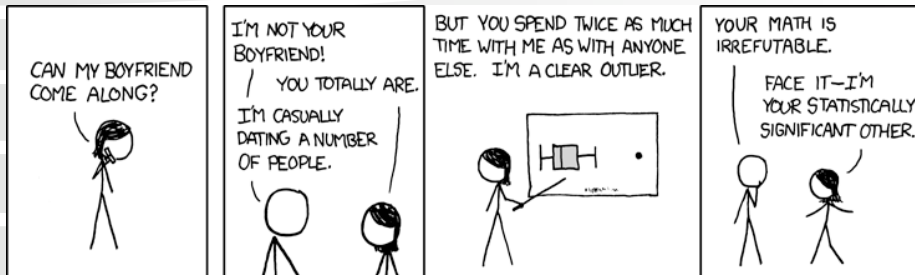
- ❖ What does each part mean?

Note: While visibility is possible in Domain Models, use it sparingly.

Recording Quantity of Items



Cartoon of the Day



... okay, but because you said that, we're breaking up.

Related?

ROSE-HULMAN
INSTITUTE OF TECHNOLOGY

In Domain Model, Use Data Type Attributes

- ❖ Primitive data types:
 - Boolean, String, Real, Integer, ...
- ❖ Sometimes more complex, but not domain specific:
 - Address, Color, Phone Number, ...
- ❖ If it's domain specific, use a class and association

Intuition from code: a "data type" is a primitive type, or a complex type where for instances a and b , $a.equals(b)$ doesn't imply $a == b$

Q6

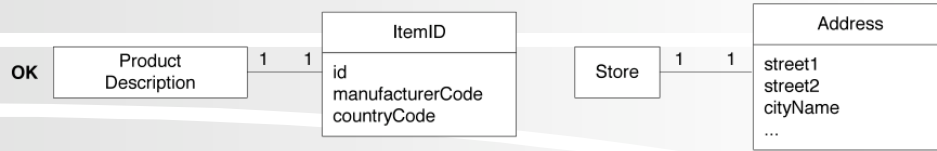
ROSE-HULMAN
INSTITUTE OF TECHNOLOGY

Create Your Own Complex Data Type When

- ❖ It has attributes of its own
- ❖ There are operations associated with it (e.g., validation)
- ❖ It's a quantity with a unit

ROSE-HULMAN
INSTITUTE OF TECHNOLOGY

Showing Data Type Attributes



Choose the representation that best communicates with the stakeholders

Q7

ROSE-HULMAN
INSTITUTE OF TECHNOLOGY

Example... Let's do one!

Q8

ROSE-HULMAN
INSTITUTE OF TECHNOLOGY

Domain Model Guidelines, Summarized

- ❖ **Classes first, then associations and attributes**
- ❖ **Use existing models, category lists, noun phrases**
- ❖ **Include “report objects”, like Receipt, if they’re part of the business rules**
- ❖ **Use terms from the domain**
- ❖ **Don’t send an attribute to do a conceptual class’s job**
- ❖ **Use description classes to remember information independent of instances and to reduce redundancy**
- ❖ **Use association for relationship that must be remembered**
- ❖ **Be parsimonious with associations**
- ❖ **Name associations with verb phrases, not “has” or “uses”**
- ❖ **Use common association lists**
- ❖ **Use attributes for information that must be remembered**
- ❖ **Use data type attributes**
- ❖ **Define new data types for complex data**
- ❖ **Communicate with stakeholders**



Homework and Milestone Reminders

- ❖ **Read Chapter 10**
- ❖ **Homework 2 – Dog-eDoctor Domain Model**
 - **Due by 5:00pm on Tuesday, December 8th, 2009**
- ❖ **Milestone 2 – Junior Project Domain Model**
 - **Due by 11:59pm on Friday, December 11th, 2009**



NexGen POS Domain Model...

