Design Class Diagramming

CSSE 374: Session 11

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

Q1

Plan for the Day

- Pre-break course evaluations
- Design Class Diagrams
  - Abbreviated - pre-course exam showed solid understanding of this
- Design exercise that should help with Homework #4

Help Me Help You

- Pre-break course evaluation on ANGEL
- Please take 10 minutes or so to help me improve the course

ROSE-HULMAN INSTITUTE OF TECHNOLOGY

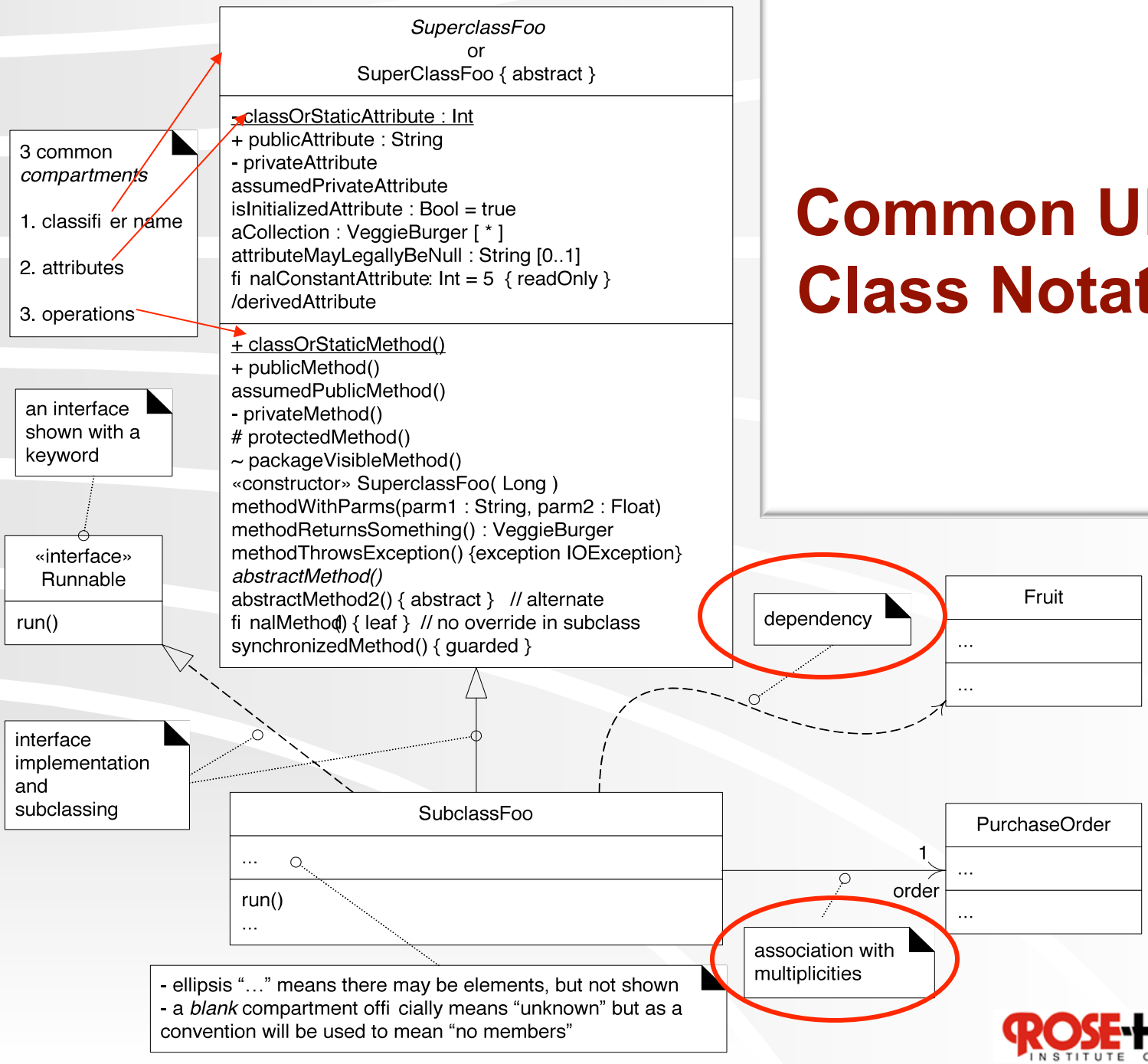
UML Class Diagrams

ROSE-HULMAN INSTITUTE OF TECHNOLOGY

Design Class Diagrams (DCD)

- Creation of DCDs builds on prior:
  - Domain Model (adds detail to the class definitions)
  - Interaction diagrams (identifies class methods)
- Creation of DCDs and interaction diagrams are usually created in parallel
- DCDs illustrates the specifications for software classes and interfaces including:
  - Classes, associations, and attributes
  - Interfaces, with their operations and constants
  - Methods
  - Attribute type information
  - Navigability
  - Dependencies

Common UML Class Notation

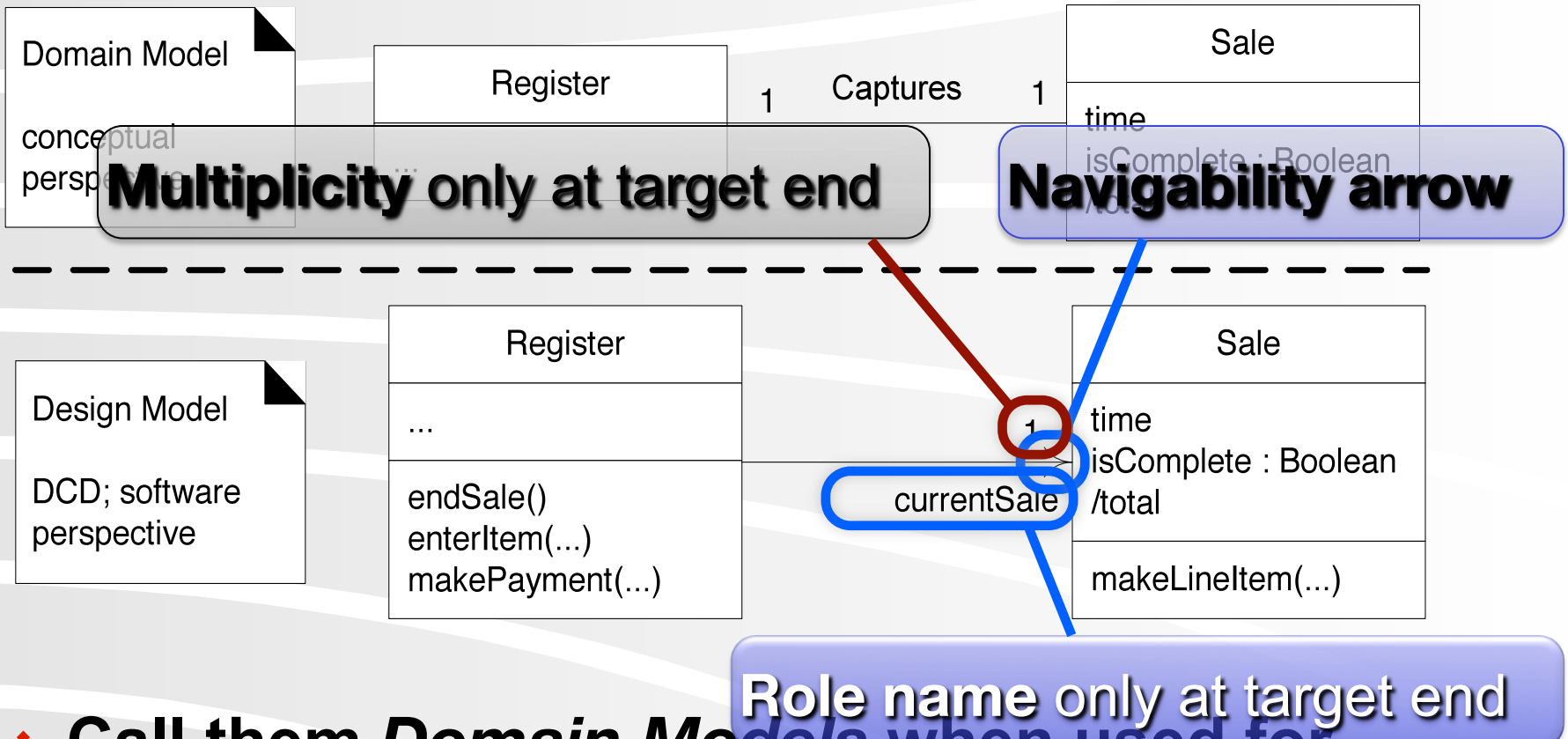


ROSE-HULMAN INSTITUTE OF TECHNOLOGY

Recipe for a Design Class Diagram

1) Identify all the *classes* participating in the software solution by analyzing the interaction diagrams
2) Draw them in a <u>class diagram</u>
3) Duplicate the *attributes* from the associated concepts in the conceptual model
4) Add *method* names by analyzing the interaction diagrams
5) Add *type* information to the attributes and methods
6) Add the *associations* necessary to support the required attribute visibility
7) Add *navigability* arrows to the associations to indicate the direction of attribute visibility
8) Add *dependency* relationship lines to indicate non-attribute visibility

Class Diagrams Do Double Duty

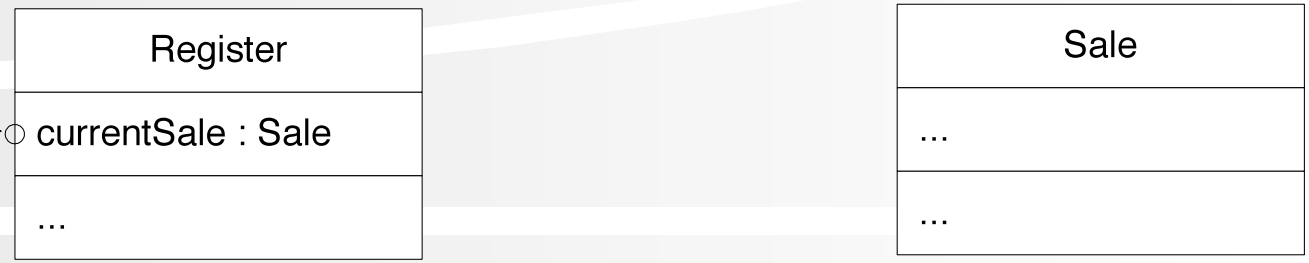


- Call them *Domain Models* when used for analysis at the conceptual level
- Call them *Design Class Diagrams* when used for design

No association name

Attribute Text vs. Association Line Notation

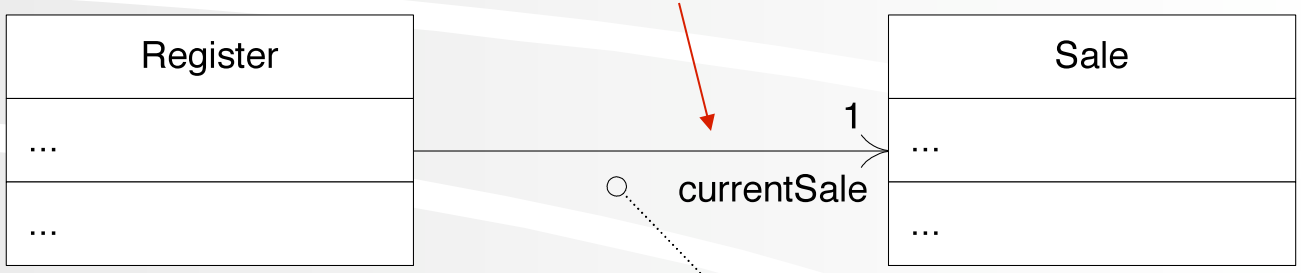
using the attribute text notation to indicate Register has a reference to one Sale instance



Avoid

Role name is attribute name

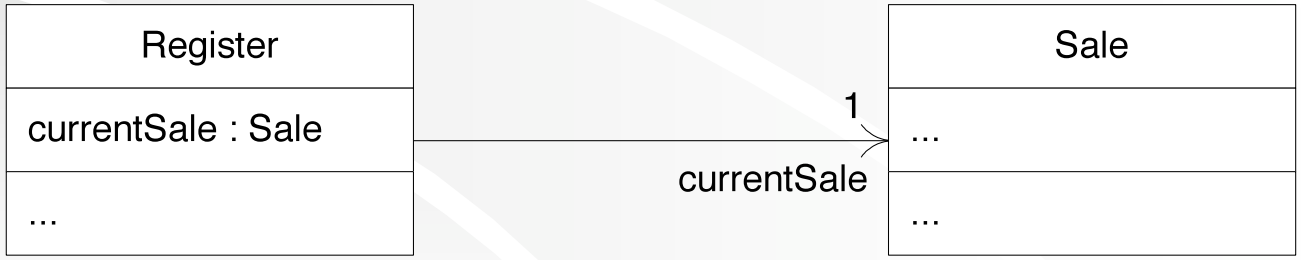
OBSERVE: this style *visually* emphasizes the connection between these classes



Preferred

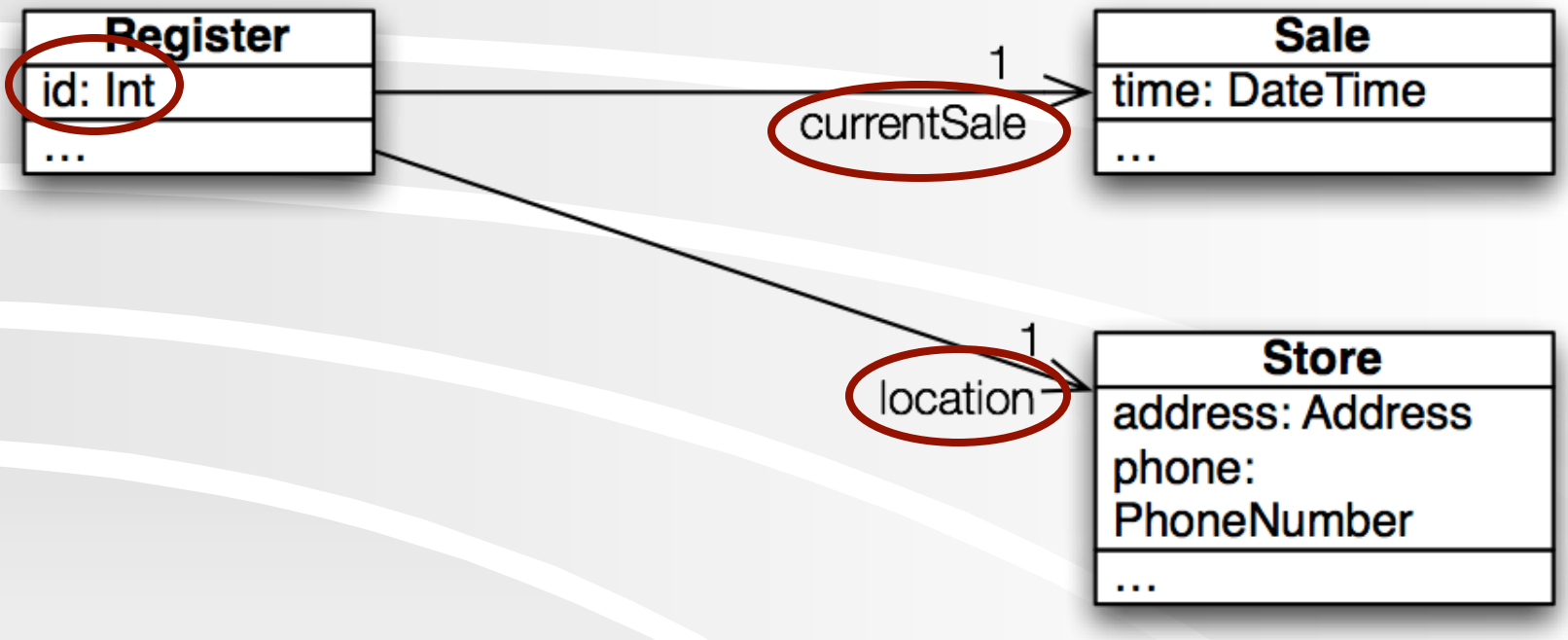
using the association notation to indicate Register has a reference to one Sale instance

thorough and unambiguous, but some people dislike the possible redundancy



Avoid

Guideline Good Practice: Example



Showing Collection Attributes

Multiplicities

1

Sale

time: DateTime
lineItems : SalesLineItem [1..*]
or
lineItems : SalesLineItem [1..*] {ordered}

...

SalesLineItem

...

...

2

Sale

time: DateTime

...

1..*

lineItems
{ordered, List}

SalesLineItem

...

...

Preferred, less visual clutter

Constraints

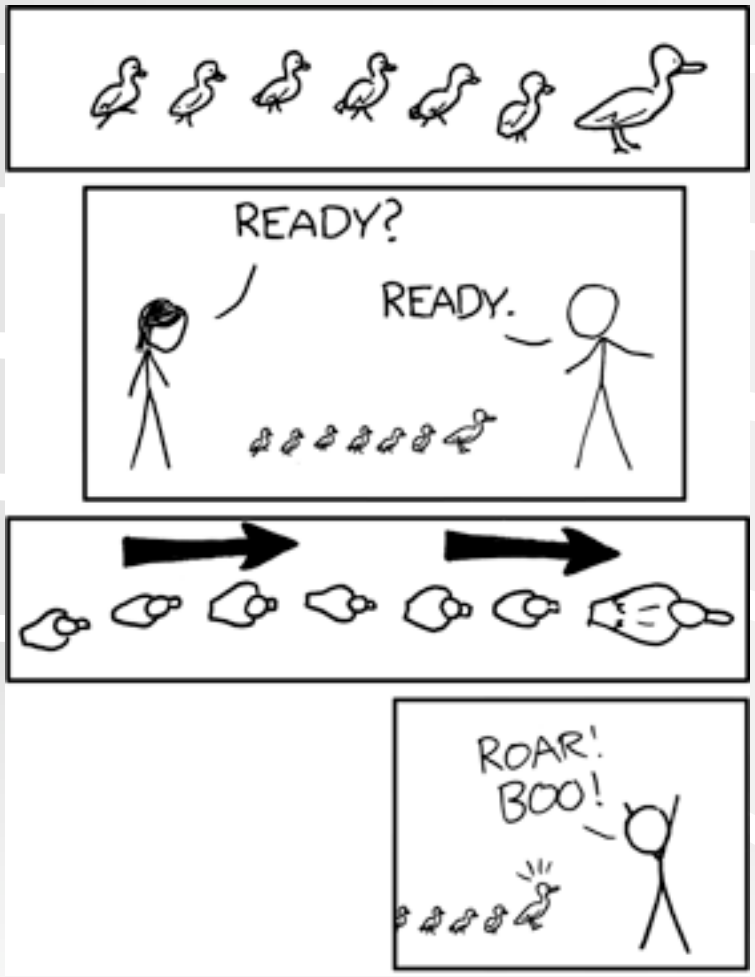
Q5

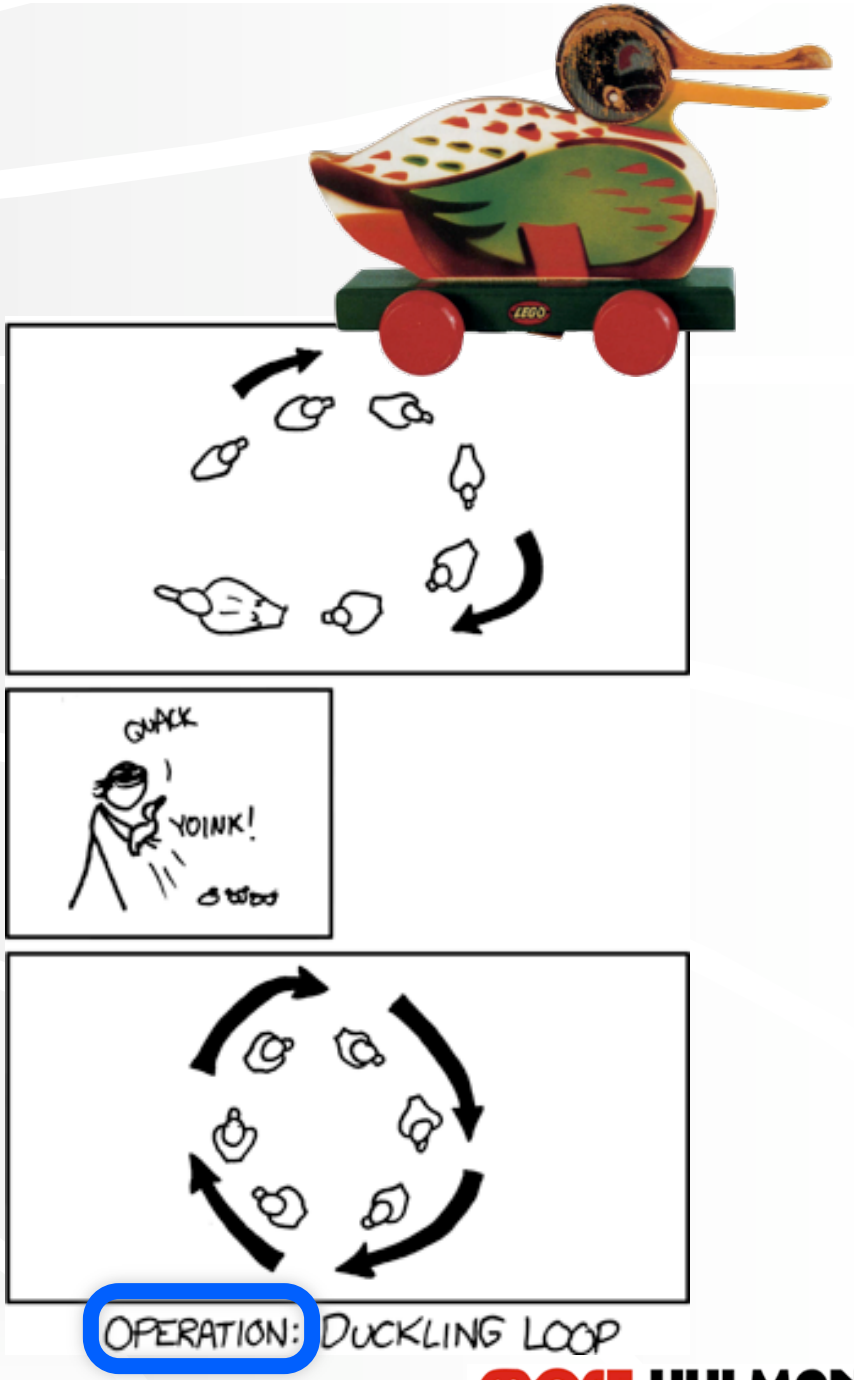
ROSE-HULMAN INSTITUTE OF TECHNOLOGY

Operations

- Syntax:
  - `visibility name(paramName:type, ...) : returnType {properties}`
  - `+ getPlayer(name:String) : Player {exception IOException}`
- Also use syntax of implementation language
  - `public Player getPlayer(String name) throws IOException`
- Operation vs. operation contract vs. method

Cartoon of the Day





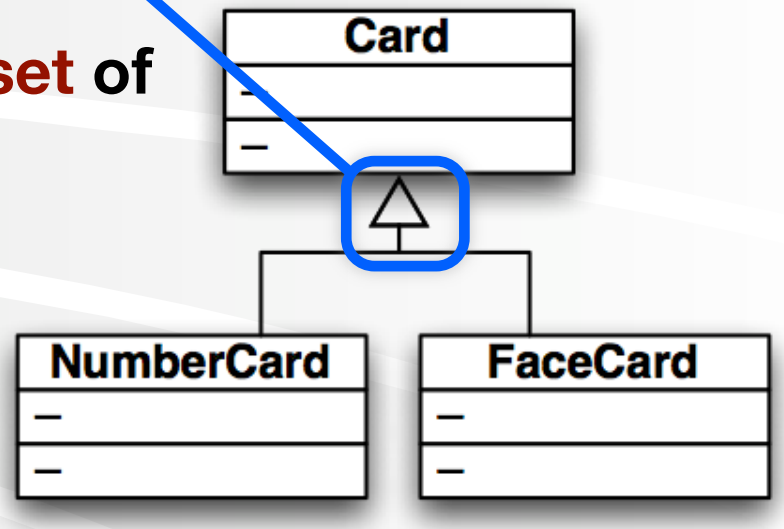
http://www.brickfetish.com/toys/duck.html

Keywords Categorize Model Elements

Keyword	Meaning	Example Usage
«actor»	classifier is an actor	shows that classifier is an actor without getting all xkcd
«interface»	classifier is an interface	«interface» MouseListener
{abstract}	can't be instantiated	follows classifier or operation
{ordered}	set of objects has defined order	follows role name on target end of association
{leaf}	can't be extended or overridden	follows classifier or operation

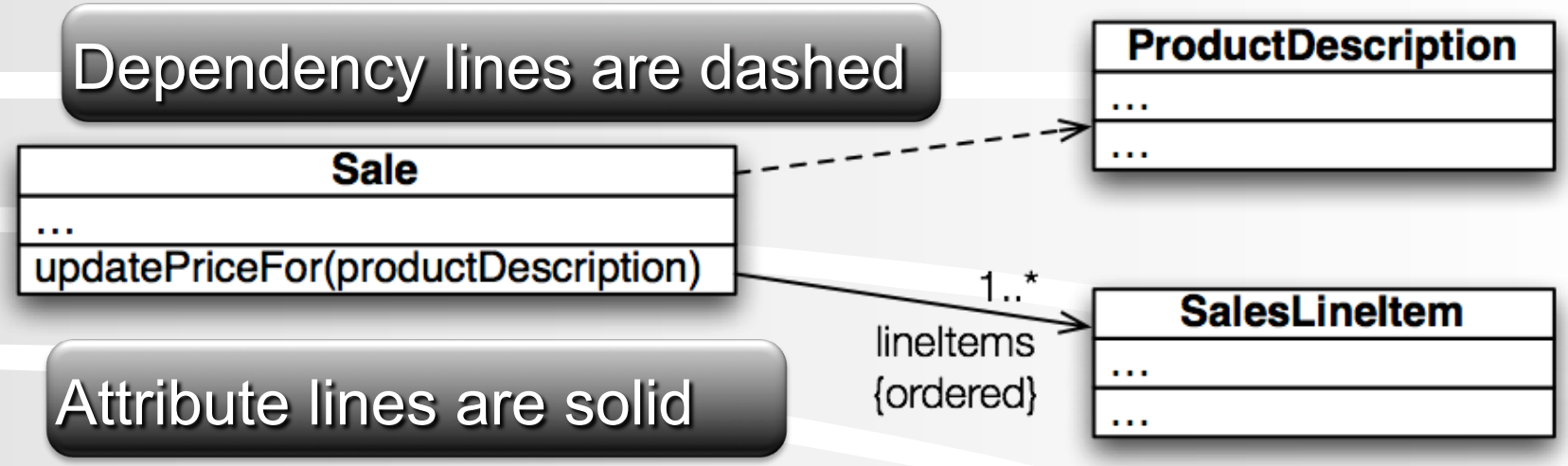
Generalization

- In domain model:
  - Says that the set of all NumberCards is a subset of the set of all Cards
- In DCD:
  - Says that, and that NumberCard inherits from Card



Dependencies

Dependency lines are dashed

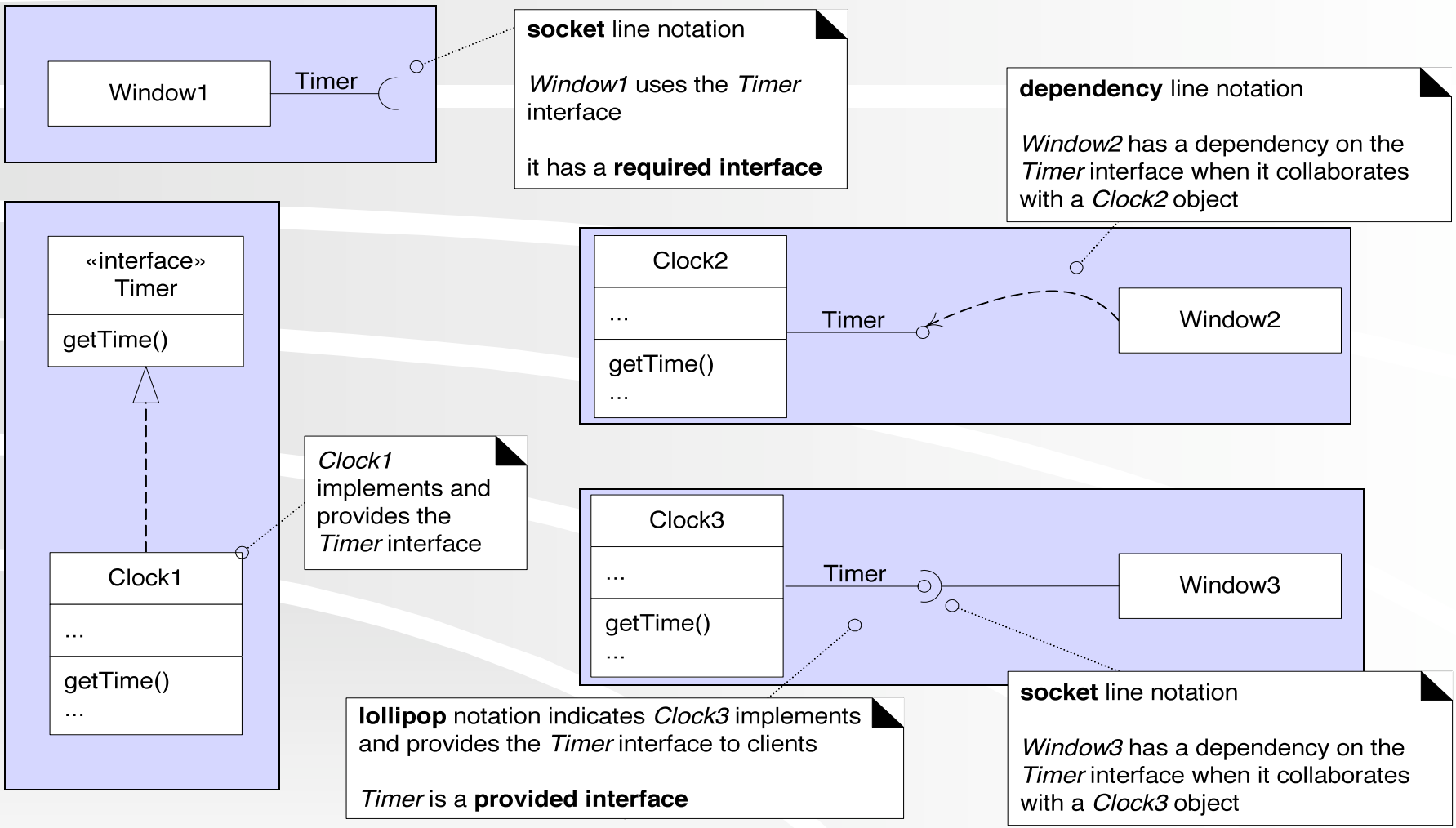


Attribute lines are solid

Use dependency lines when a more specific line type doesn't apply.

Can label dependency arrows: e.g. «call», «create»

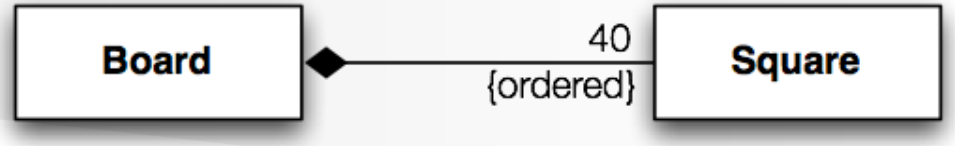
Interfaces in UML



Q7

Composition

- More powerful than an attribute arrow
- Describes whole-part relationship



- Implies
  - Instance of part belongs to only one composite at a time
  - Part always belongs to a composite
  - Composite creates/deletes parts

Association name in composition is always implicitly some “has-part” relation. So, it’s common to omit association or role name with compositions

Interaction Diagrams and Class Diagrams

- Interaction diagrams show dynamic behavior
- Class diagrams show static behavior
- Tips:
  - Draw concurrently
  - Use two adjacent whiteboards, one for static and one for dynamic
  - Sketch communication diagrams, document using sequence diagrams

Example

Q9,10

ROSE-HULMAN INSTITUTE OF TECHNOLOGY

Homework and Milestone Reminders

- Read Chapter 17 on GRASP (through pg. 290)
- Homework 4 – Dog-eDoctor System Preliminary Logical Architecture and Design
  - Due by 5:00pm on Tuesday, January 5th, 2010
  - Extra credit if you get it in by 5:00pm this Friday!
- Milestone 3 – Iteration 1: Junior Project
  - Finish Analysis Model (SSDs, OCs)
  - Logical Architecture - Package Diagrams, and
  - 1st (initial) Version of System
  - Due by 11:59pm on Friday, January 8th, 2009