CSSE 374: Applied Domain Modeling – Associations and Attributes

Shawn Bohner

Office: Moench Room F212

Phone: (812) 877-8685
Email: bohner@rose-hulman.edu

Q1

ROSE-HULMAN
INSTITUTE OF TECHNOLOGY

Don't do Design like Marketing

Learning Outcomes: O-O Design

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

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

- **Domain Model Associations**
- **Domain Model Attributes**
- **Apply Domain Modeling in Exercise**

Associations in domain modeling provide **knowledge** that should be **preserved**. Why would you NOT want to record all associations?

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

Domain Model: Associations

Register		Records-current ▸	Sale
	1		0..1

association name

multiplicity

Be Parsimonious
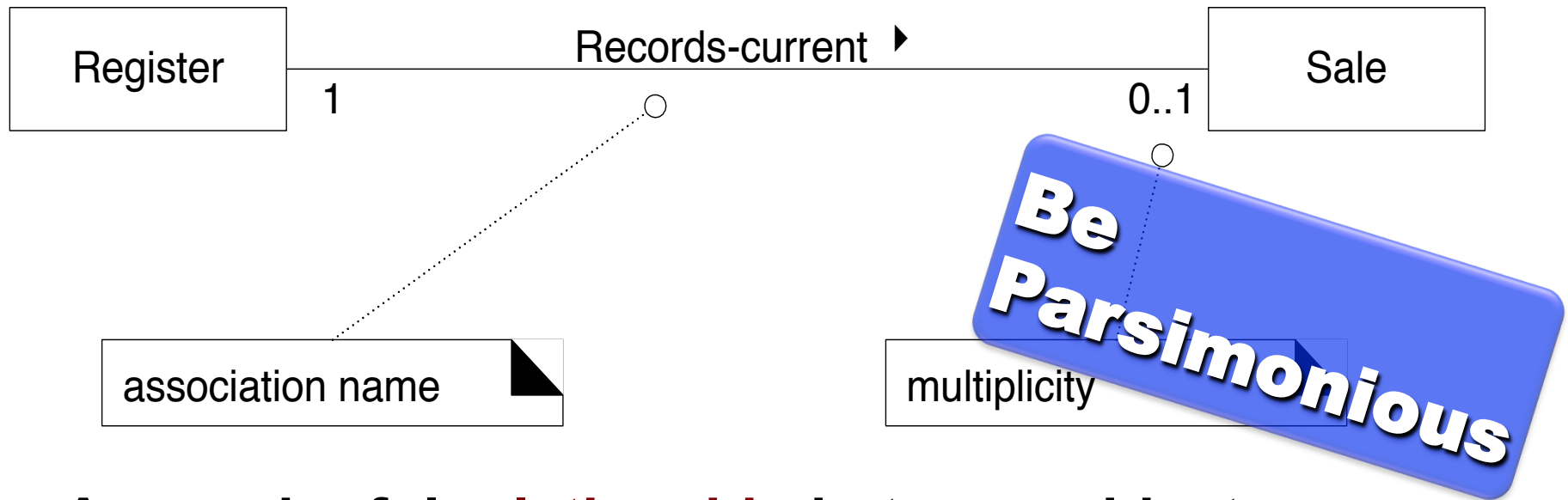
- **A meaningful relationship between objects**

- **Knowledge of a relationship must be preserved (some memory of a relationship)**

- **Associations are NOT**
  - ☐ A model of data flows
  - ☐ Instance variables
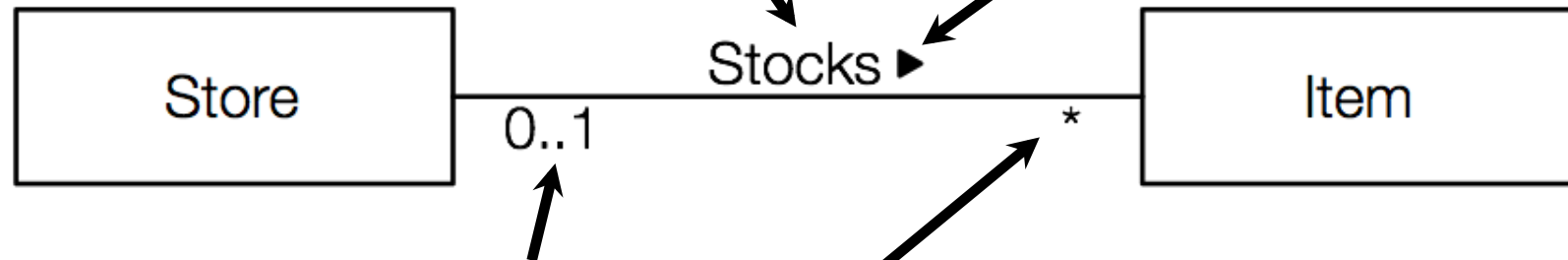 - ☐ DB foreign keys
  - ☐ SW object connections

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

Store	Stocks ▶	Item

0..1   *

Multiplicity (Cardinality):
- ✓ '*' means "many"
- ✓ x..y means from x to y inclusively

Cardinality (AKA Multiplicity)

——— *	[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

Let's Associate some more...

```
                    *                1
          +-------------------------------------+
          |          Flies-to                   |
+---------+                                     +---------+
|         +-------------------------------------+         |
|         |          Flies-from                 |         |
| Flight  |                                     | Airport |
|         +-------------------------------------+         |
+---------+    *                1               +---------+
          |                                     |
          +-------------------------------------+
```
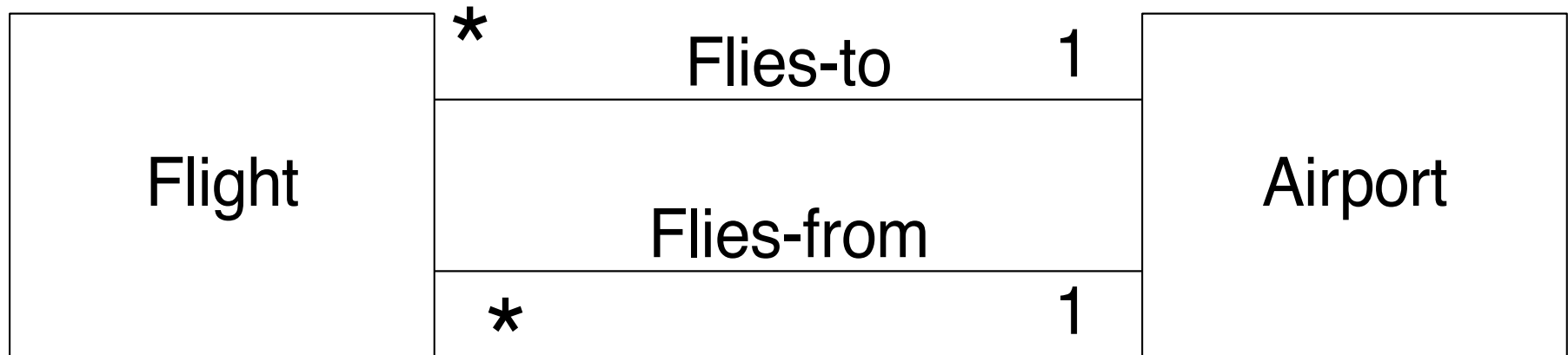
- **Classes can have multiple associations**

- **Classes can also self-associate!**
  - □ **Object creates itself**
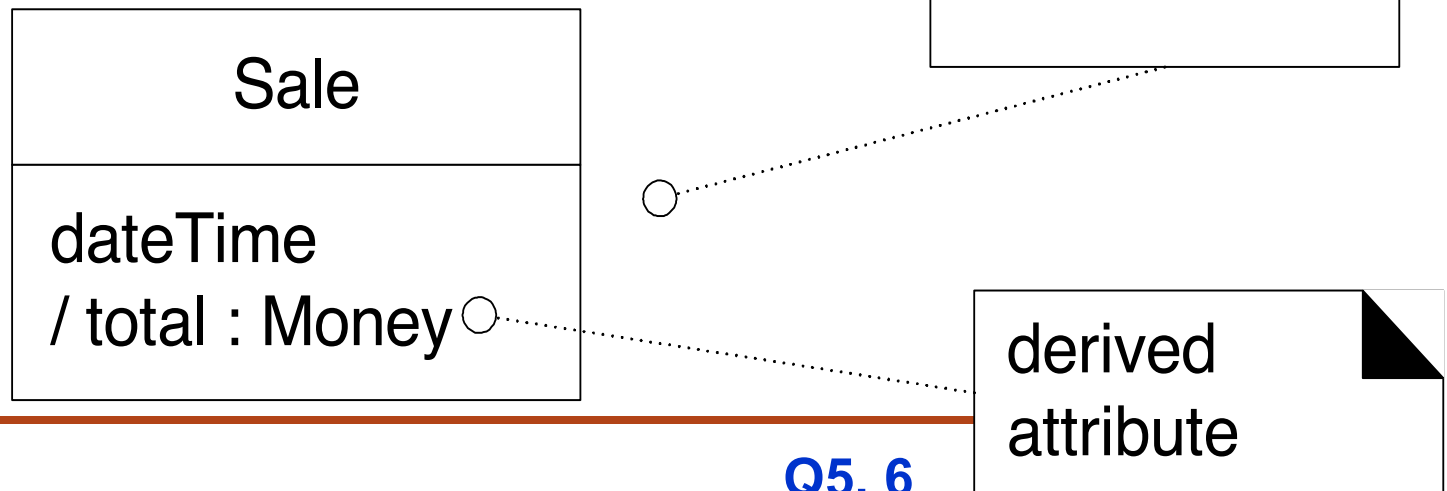 - □ **Object modifies itself**
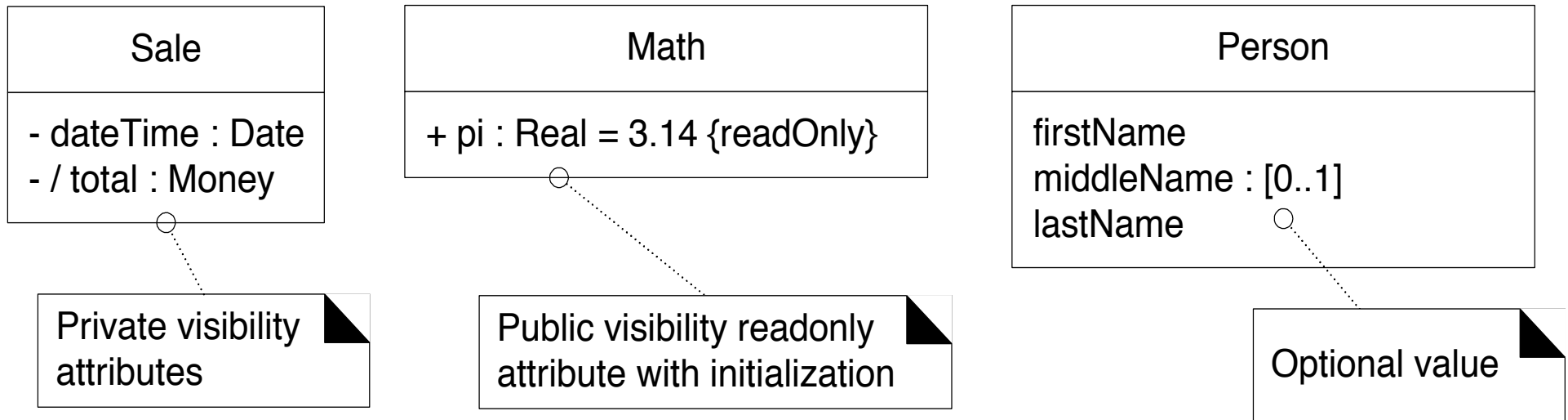 - □ **Object moves itself…**

Partial NexGen POS Domain Model

Records-sale-of

Ledger

Product Catalog

Product Description

Contains

1   1..*

1

Used-by
*

Records-accounts-for

0..1

Sales LineItem

1   1

Store

Describes
*

Stocks

Item

1   1   *   1..*

1..*

Contained-in

1

Logs-completed

1   Houses

1..*

Sale

*

Captured-on

Register

0..1   1

1

Paid-by   1   1   Is-for

1   1

Works-on   1

CashPayment

Customer

Cashier
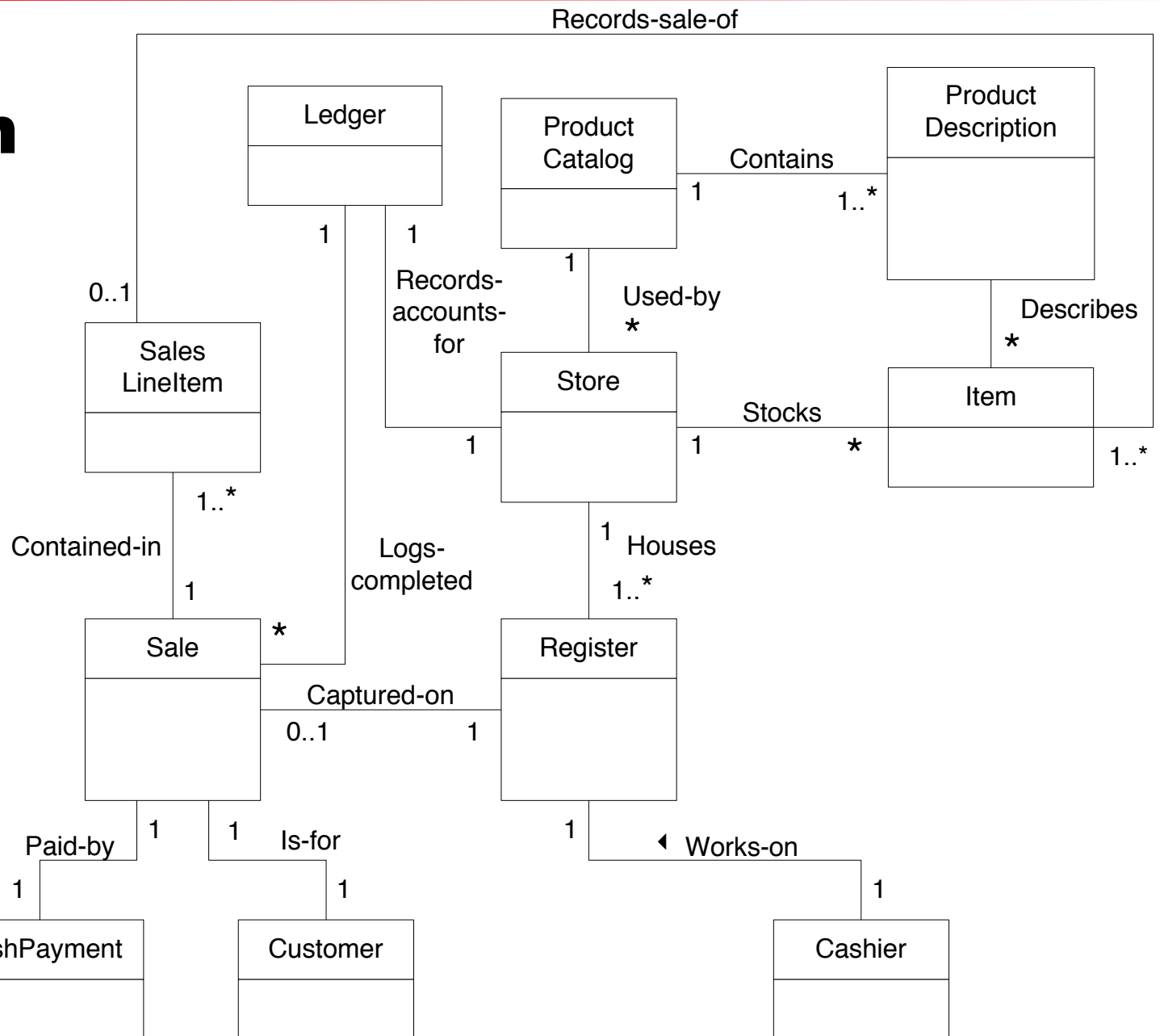
Domain Model: Attributes

- **An object's logical data value that must be remembered**
  - Some attributes are derived from other attributes

- **Usual primitive attributes (data types not shown in DM)**
  - Numbers, characters, booleans

- **Common compound attributes**
  - Date, time, address, SSN, phone number, bar codes, etc.
  - May even become full class objects in design…

Sale
dateTime / total : Money

attributes

derived attribute

Visibility in Domain Models

Sale
- dateTime : Date
- / total : Money

Private visibility attributes

Math
+ pi : Real = 3.14 {readOnly}

Public visibility readonly attribute with initialization

Person
firstName
middleName : [0..1]
lastName

Optional value

While visibility may be possible in Domain Models, showing visibility here may be overkill!

Recording Quantity of Items

SalesLineItem			Item
	0..1 Records-sale-of 1		

Each line item records a
separate item sale.
For example, 1 tofu package.

SalesLineItem			Item
	0..1 Records-sale-of 1..*		

Each line item can record a
group of the same kind of items.
For example, 6 tofu packages.

SalesLineItem			Item
/quantity ○...	0..1 Records-sale-of 1..*		

derived attribute from
the multiplicity value
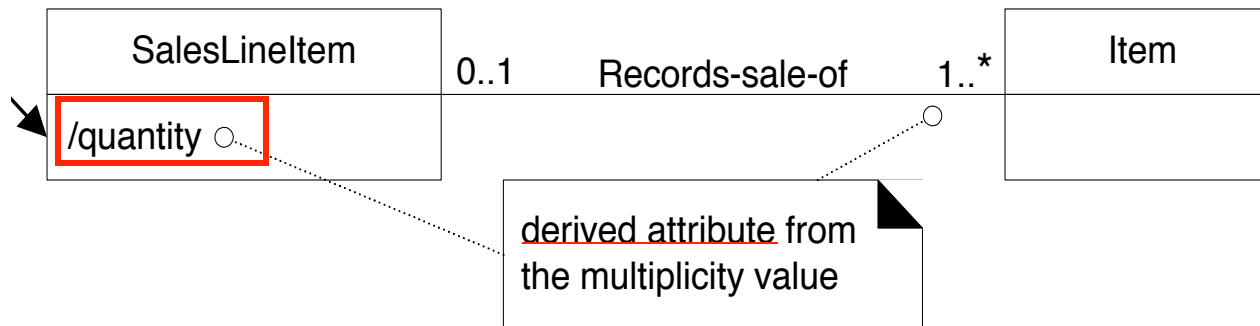
xkcd on Extraterrestrials

... reporters are just dying to experience the latest on arsenic-based DNA?

Class Exercise on Domain Modeling

- **Break up into your project teams**

- **From Use Cases identify Conceptual Classes**

- **Draw Conceptual Classes & requisite Associations between them**

- **Add the Multiplicities to the Associations**

- **Add the Attributes to Conceptual Classes**

Note that the classes here are not all of the ones needed for Homework 1 – just a good start.

Homework 1: Basic Use Case 1/2

- **UC1: Customer rents videos**
  - ☐ **Preconditions**: Customer has a membership, has selected videos they want, and made system aware of their choices.
  - ☐ **Actor:** Customer (self-service/remote), or store associate (in store)
- **Main flow:**
  1. Actor indicates to rent first item (e.g., clicking "rent" on a networked device, or scanning it physically in a store)
  2. System verifies immediate availability, and waits to make next option
  3. Actor indicates they are done selecting
  4. System shows total, prompts for payment
  5. Actor selects method of payment, entering additional data if needed (e.g., credit card number)
  6. System verifies the payment has gone through, schedules the goods for rental (e.g., sets up a window to click on to view the video remotely, or tells the store clerk where to find the DVD)
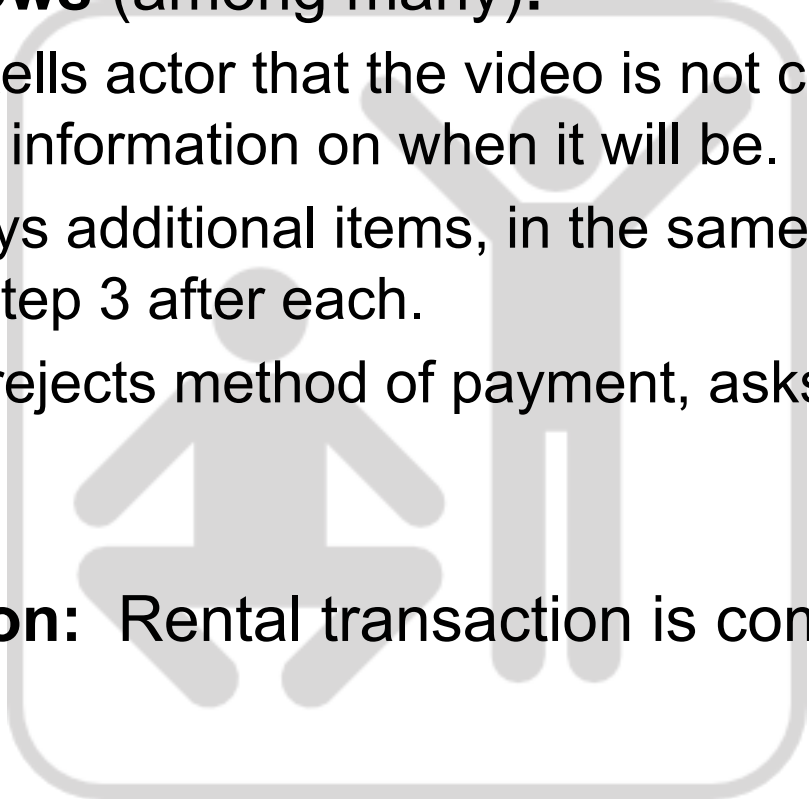
ROSE-HULMAN
INSTITUTE OF TECHNOLOGY

- **Alternate flows** (among many)**:**

  2a. System tells actor that the video is not currently available, and provides information on when it will be.

  3a. Actor buys additional items, in the same way, if desired, returning to step 3 after each.

  6a. System rejects method of payment, asks actor for alternative.

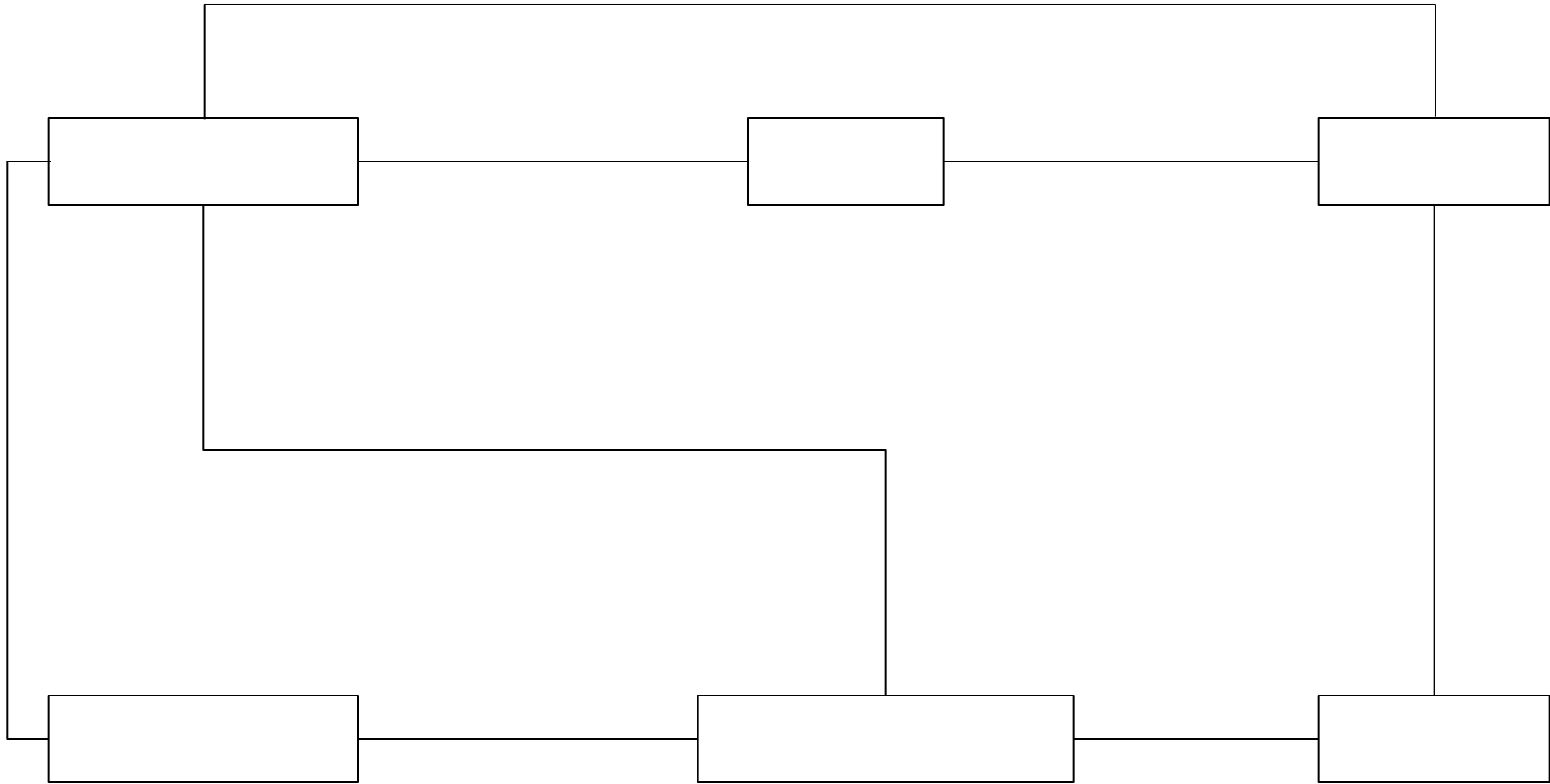- **Postcondition:** Rental transaction is complete.
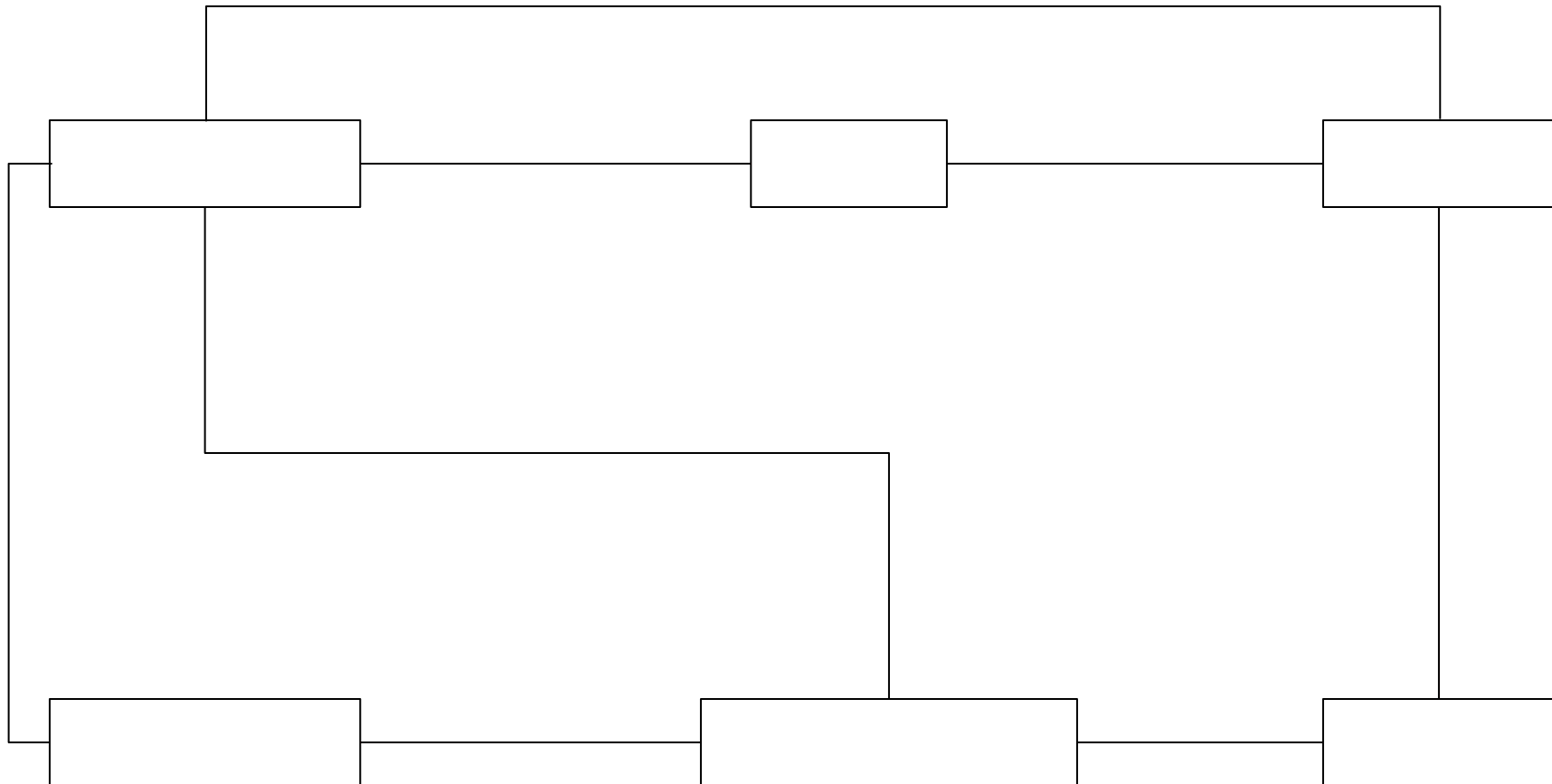
What are the Conceptual Classes?
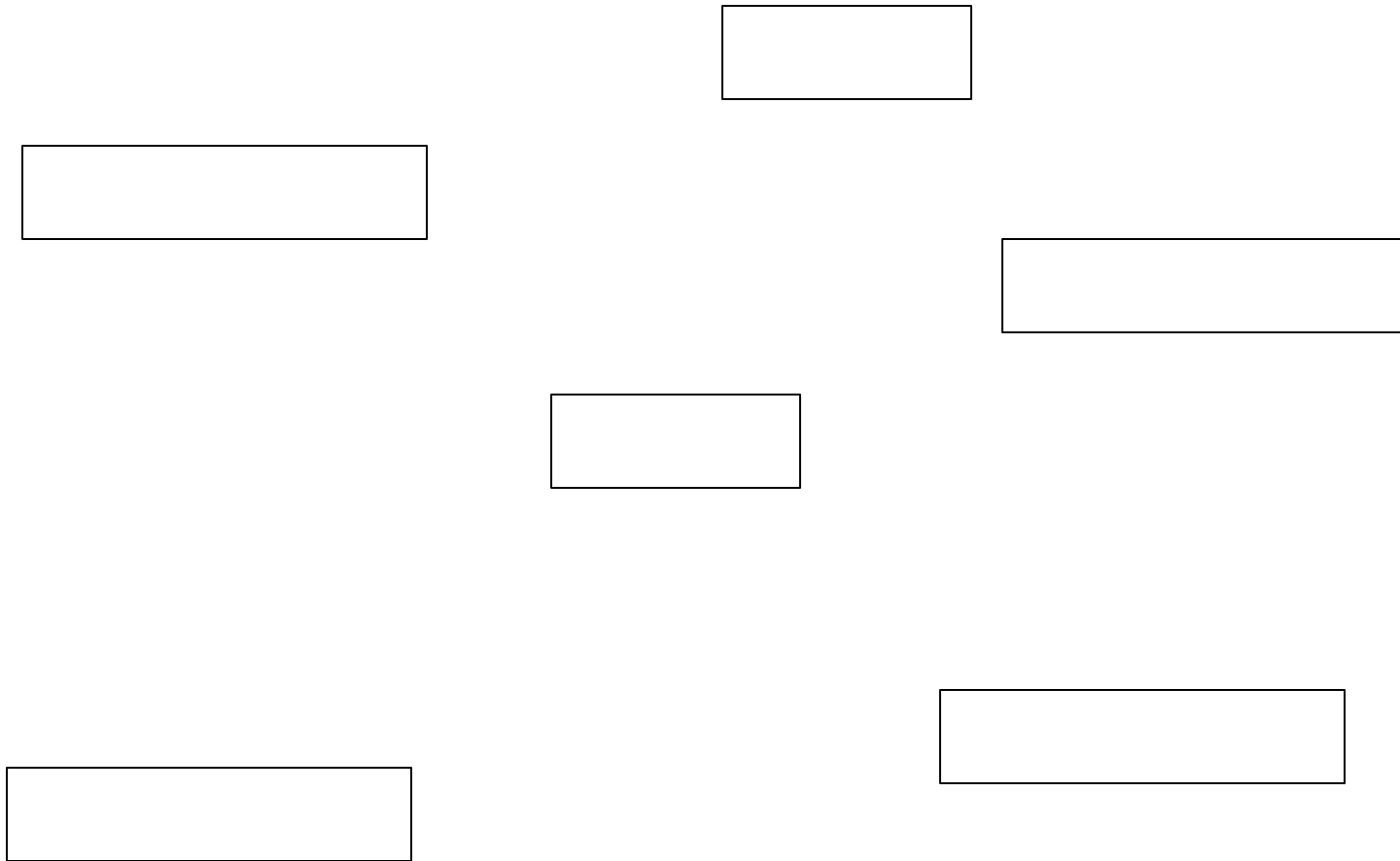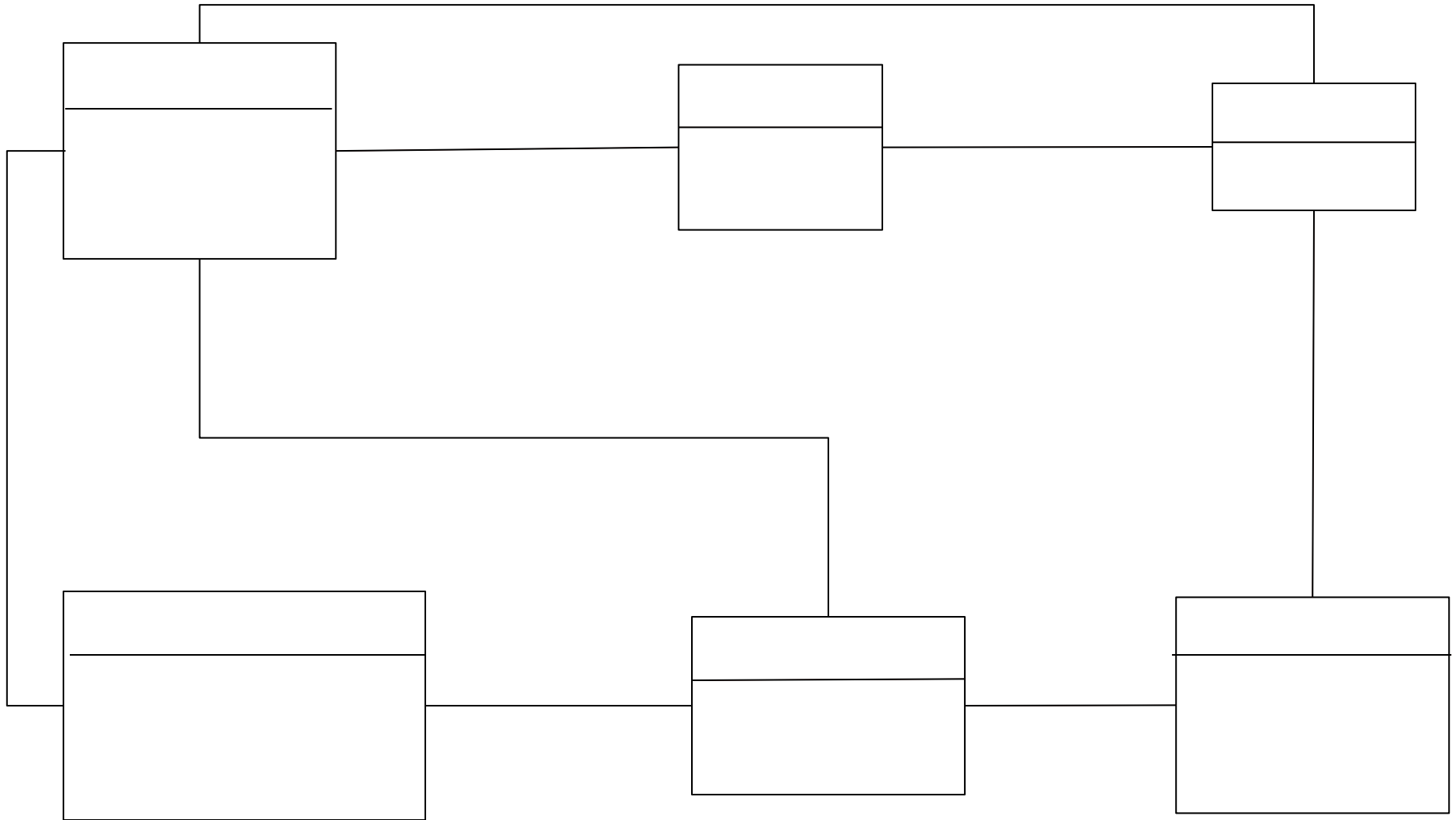
How do we relate these Classes?

What are the Multiplicities on the Associations?

Incorporate Attributes in your DM

Homework and Milestone Reminders

- **Read Chapter 10**

- **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**