INFO213: Lecture 2

OO Analysis and Design

From Requirements to Specifications

Recap: Object Orientation

Class

"A logical grouping of data and the methods that manipulate that data"

Properties (I have a …)

- Attributes (primitive e.g., String)
- Reference (relationship with another class)

Methods (I can do …)

Recap: Object Orientation

Object

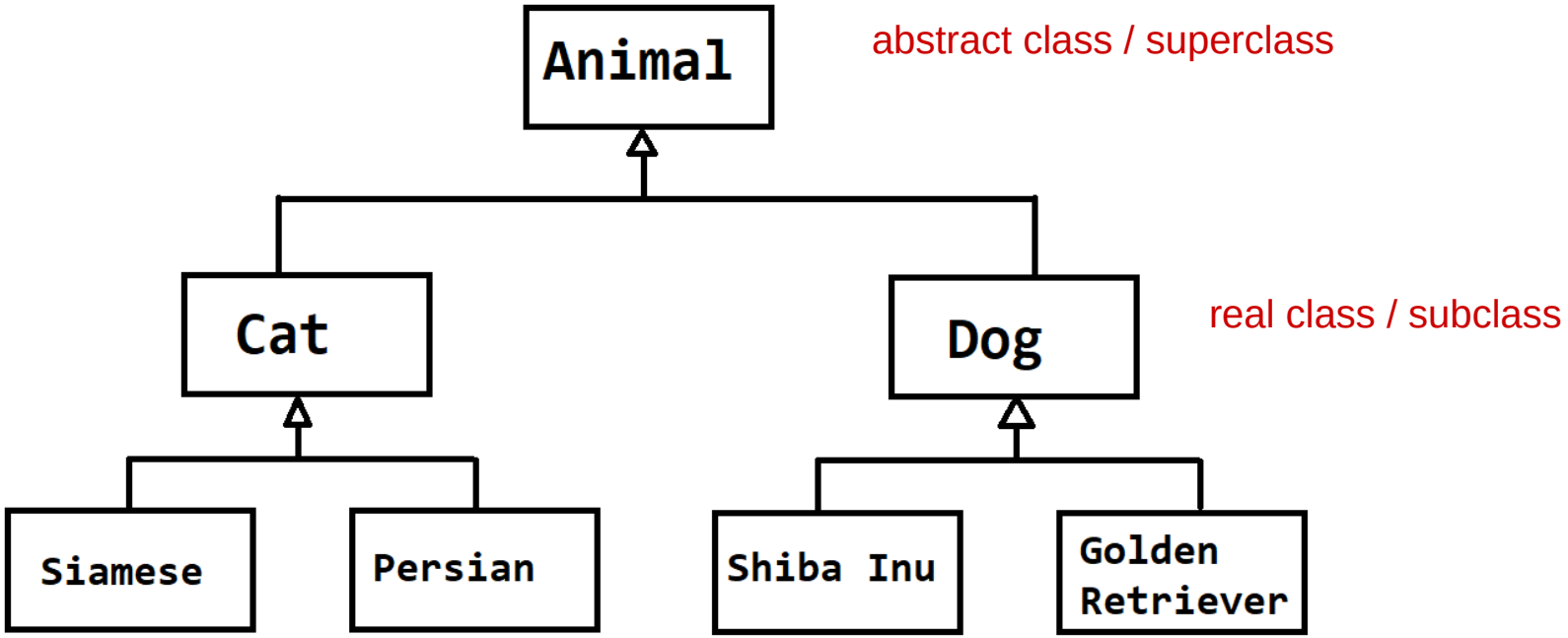
"A particular instance of a Class"

If the Class is "Human", we're all "Human" objects.

Recap: Object Orientation

Inheritance

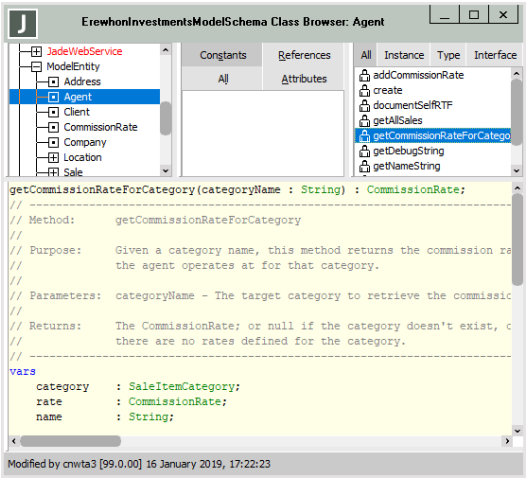
"Is kind of a"



Today: UML

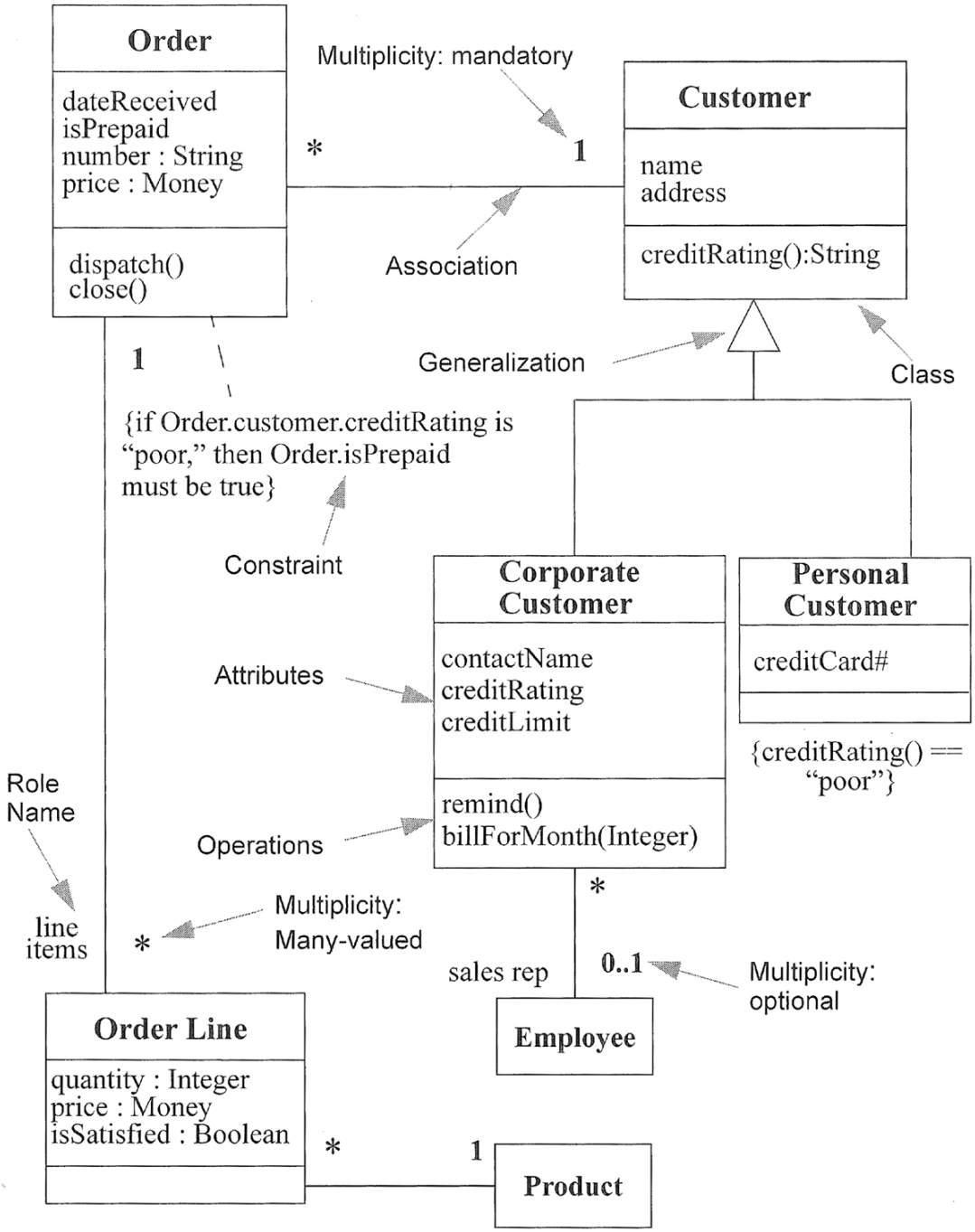
Today: UML

UNIFIED MODELING LANGUAGE™



What problem are we trying to solve? → How are we going to solve the problem? → A software solution! The problem is solved!

class diagram



UML Notation for Class Diagrams

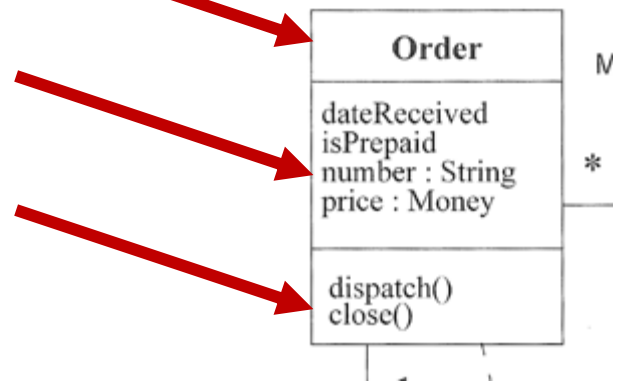
- What do boxes represent?
- What are the three sections in those boxes?
- What do lines represent?
- What are the numbers on those lines?
- How do you show inheritance?

UML: Class definition

Name of the class

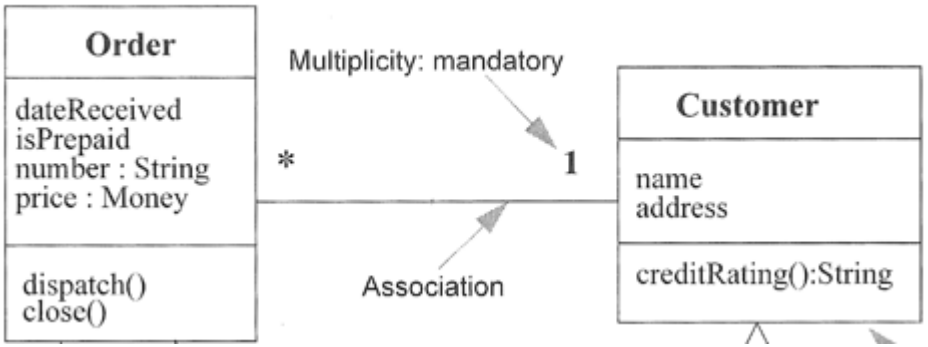
(Primitive) **Attributes of the class**

Methods of the class

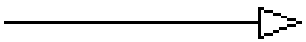
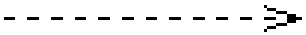
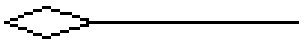
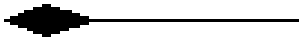
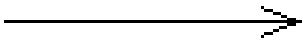
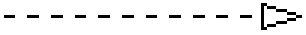


UML: Relationships between classes

any one customer can have any numbers of order

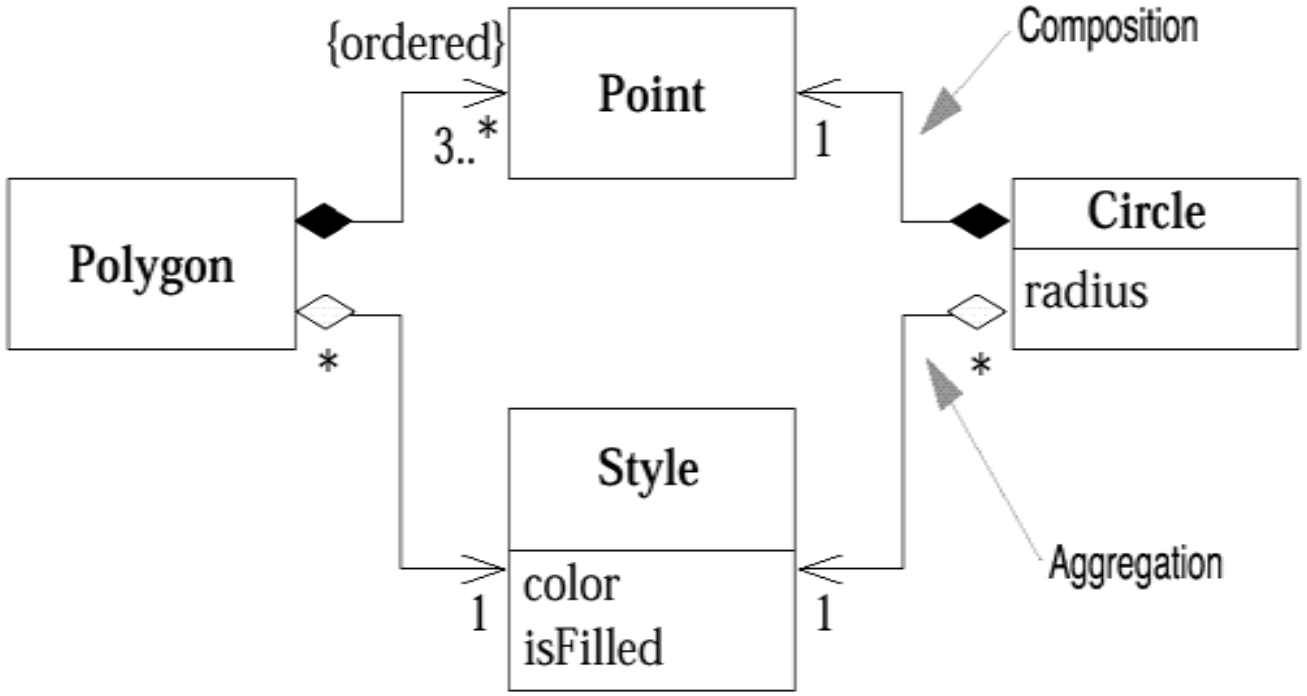


UML: Types of Relationship

Inheritance		"I am a kind of…"
Dependency		"I need you…" class2 needs class1 to function
Aggregation		"unpluggable component" chair is part of the room but you can take the chair away and put it somewhere else which cause no harm
Containment		"This is literally part of me…" (AKA Composition) part of the thing and can't be taken away
Association		"We know each other…" class knows each each
Directed Association		"I know you, but you don't know me…" one way
Realization		"I pinkie-promise I can fulfil this contract of behaviour…" interface

Aggregation vs. Composition

- Inseparable part vs. component



Multiplicity

Symbol	Meaning
1	Exactly 1
3	Exactly 3
0..1	Either 0 or 1
0..3	0, 1, 2 or 3
*****	Any number (including 0)
1..*	Any number except 0

UML Notation for Class Diagrams

- What do boxes represent?
  - Classes
- What are the three sections in those boxes?
  - Name, Attributes, Methods
- What do lines represent?
  - Relationships (References)
- What are the numbers on those lines?
  - Multiplicity
- How do you show inheritance?
  - Arrow with a triangle Page 4

So how do I come up with one of those?!

- Start with a requirements document
  - *(Will usually be sourced through customer interviews)*
- Perform a "Noun Analysis"
  - Look for the nouns
  - Collect them all!
  - Then sort out what's what
- Decide the classes first
  - Leftover nouns become properties…
  - Or discarded if they are irrelevant!

Noun Analysis Exercise (Part 1)

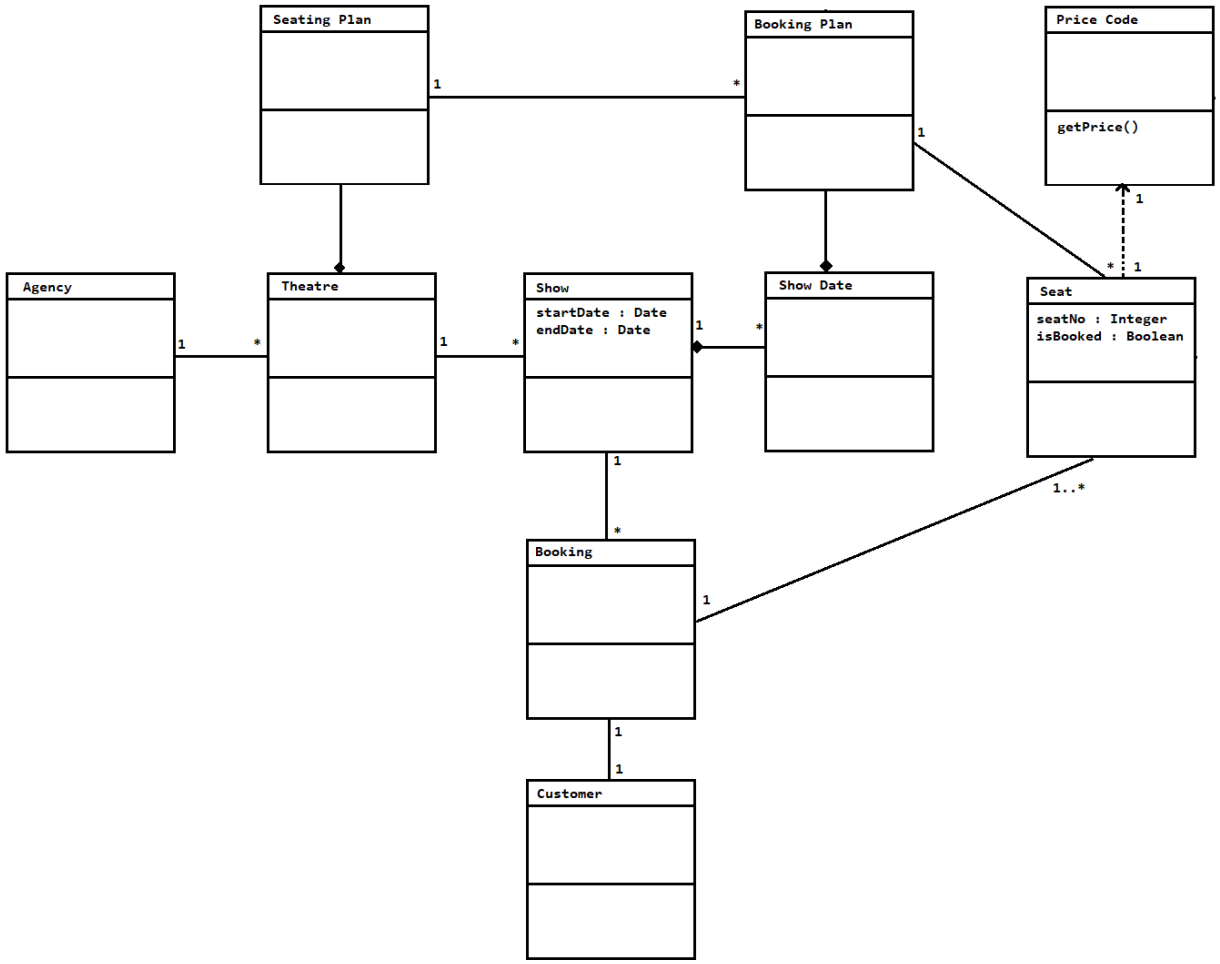
- From the Theatre Agency Requirements: *(additional handout provided)*
  - Identify all Nouns
  - If the Noun has an adjective, consider including it with the Noun
  - If the Noun is plural, make it single

The Nouns?

- Show
- Theatre
- Agency
- Booking
- Show date
- Start date
- Finish date
- Seat
- Layout
- Seating Plan
- Booking Plan
- Customer
- Price
- Price Code

Now let's try making some classes!

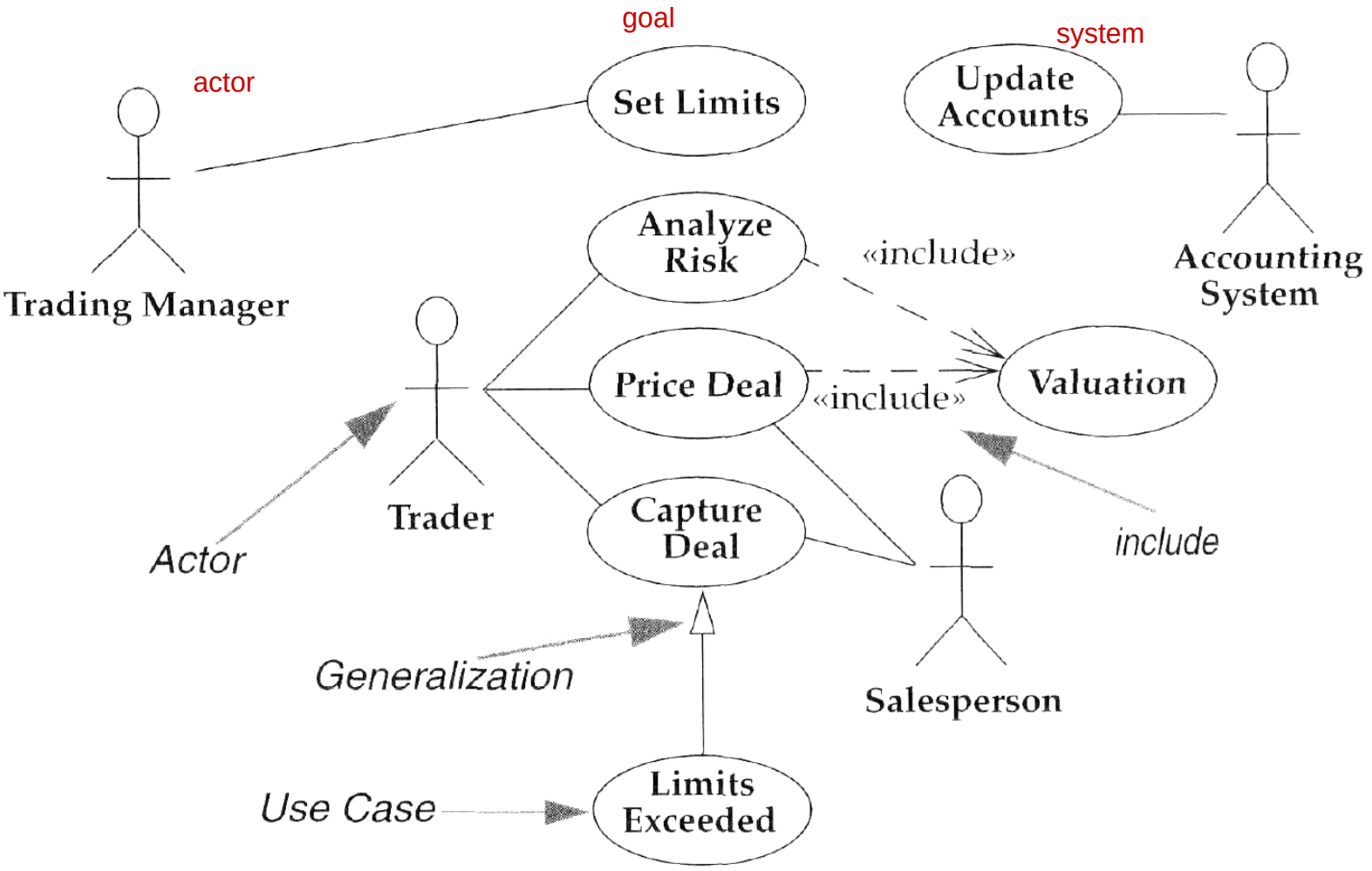
One possible solution (incomplete)??



UML Use Case Diagrams Purpose: External Interaction with the System

- Essential at design stage
  - Focus on interaction
    - System input
    - System output
- Actors/roles
  - Different type of actors
- Use cases are later mapped to user interface functionality

Use Case Diagram Notation



Use case Exercise (Part 2)

- From the Theatre Agency Requirements: *(additional handout provided)*
  - Identify all Actors
  - Identify all Systems
  - Identify all Goals
  - Use case diagram should be light on detail
  - Can incorporate some detail into UML Class Diagram instead!

Action Points, Readings for Next Week

- Take the UML Class Diagram we created in class, create JADE Classes, Attributes, References and Methods to model it
- You ***Don't*** need to actually implement any of the methods, just leave the method body empty, perhaps with a comment about what it should do.