

---

---

## **WEEK 4 Material**

### *Lecture 4b (Mon.)*

*Intro to Information Requirements*

*& “Storyboarding the UCs”*

*Monday “Paper Prototyping” Workshop by Bonnie*

# Information Requirements

---

---

*In week-3, we quoted Kruchten:*

*For example,...*

"Use cases emerge when you focus on the things of value that a system provides to an actor." ~ Kruchten

**We focus on these "valued outputs" by analyzing the "Information Requirements" of the system, in two flavors:**

(1) the Information Queries

*providing critical outputs needed from the system*

(2) the Information Updates

*keeping the persistent data used in the queries current and correct*

# Information Requirements ~ (1) the Information Queries

---

*Our Use Case list should include the important query requests the actors need to make of the system*

**As a part of understanding of the business workflow and the initial set of use cases, we need to identify each actor's most critical information needs that the system will support.**

*Then, for each identified:*

**With the actor, sketch out what it might look like.**

first, the layout only

next, develop some example values

**Discuss what the output means.**

why is it important

how it will be used

## Information Requirements ~ (2) the Information Updates

---

*Update requests supply the system with new values from the 'real-world' ~ i.e., what is needed to keep the system's persistent data up-to-date. They may not have a "visual" aspect.*

### **Identify the update events**

*i.e., the requests that trigger the updating of the system's data.*

*Doing an update is never an end in itself. It always supports some business need for correct information.*

For example, think:

*"I need to change the customer's address because, if I don't, our mailing campaign won't reach them."*

vs.

~~*"I need to update the mailing-address variable with a new value."*~~

## **Information Requirements ~ document any system support problems**

---

---

*Document any current problems with system support, such as:*

**Is any of the information not available at all today? Is data missing?**

**Is the data available but wrong? (in whole or in part? ...which parts?)**

**Does it get to you too late? (when do you receive it? when do you need it?)**

**Is the information available to others (other places) but not in your area?**

**Are you getting too much data? (data on an existing output that you do not refer to)**

**Is there data external to the business that would be helpful if it could be incorporated into the output? (say, for comparisons)**

## Information Requirements ~ gather processing profile details

---

*begin gathering any processing profile details that are known about the output of the use case, such as the output's:*

### **frequency**

*is this produced: ... daily? ... monthly? ... on-demand?*

### **time-criticality**

*for those 'on-demand,' how long can the user wait to see an answer:*

*... a week? ... overnight? ... 3 seconds?*

### **volumes**

*does this reflect: ... one object? ... the entire file?*

### **number of users**

*how many users need to refer to the output's data?*

*And in how many geographic locations?*

---

---

. *Review of OOAD so far.....*

Entity class

Class Generalization

Domain Model

Multiplicity

System Boundary and Actors

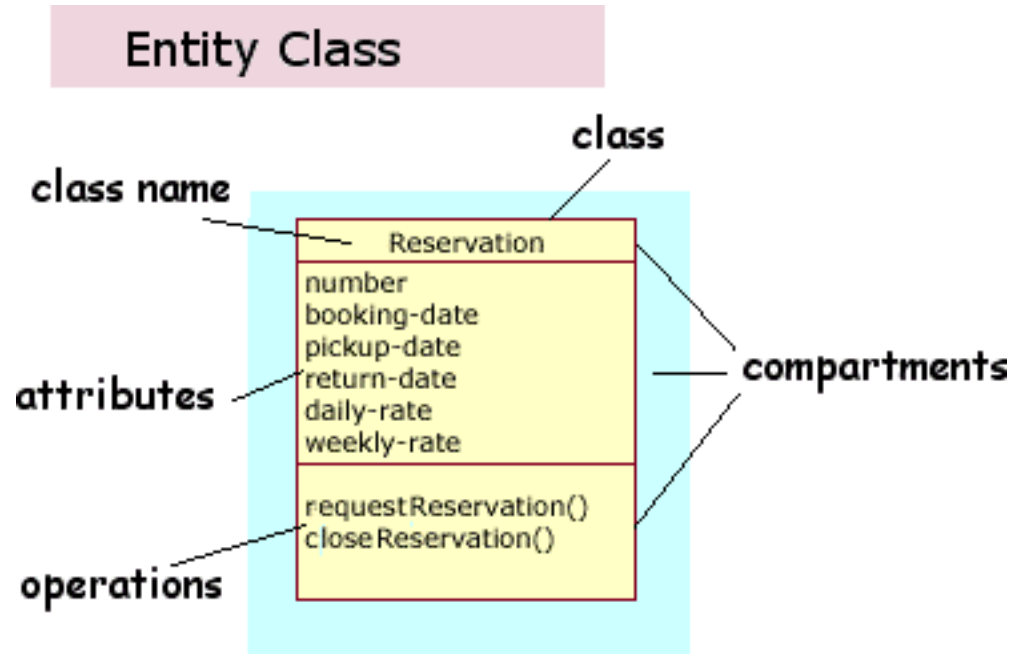
Packages

Use Case Diagram

Business Workflow as “Swimlanes” Diagram

# the Domain Model ~ the user's vocabulary as a graphic

---

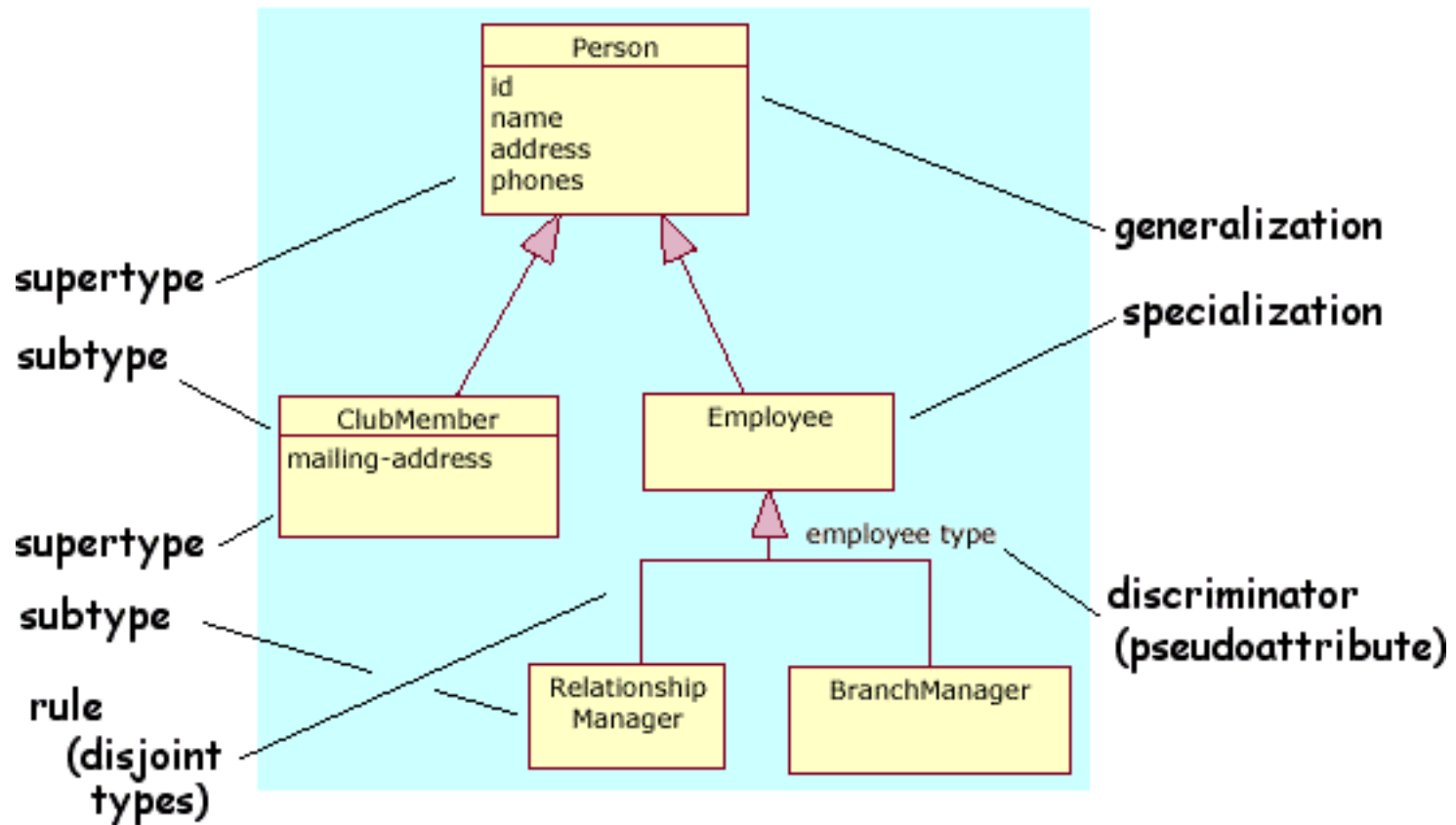


*A companion glossary defines domain terms....*

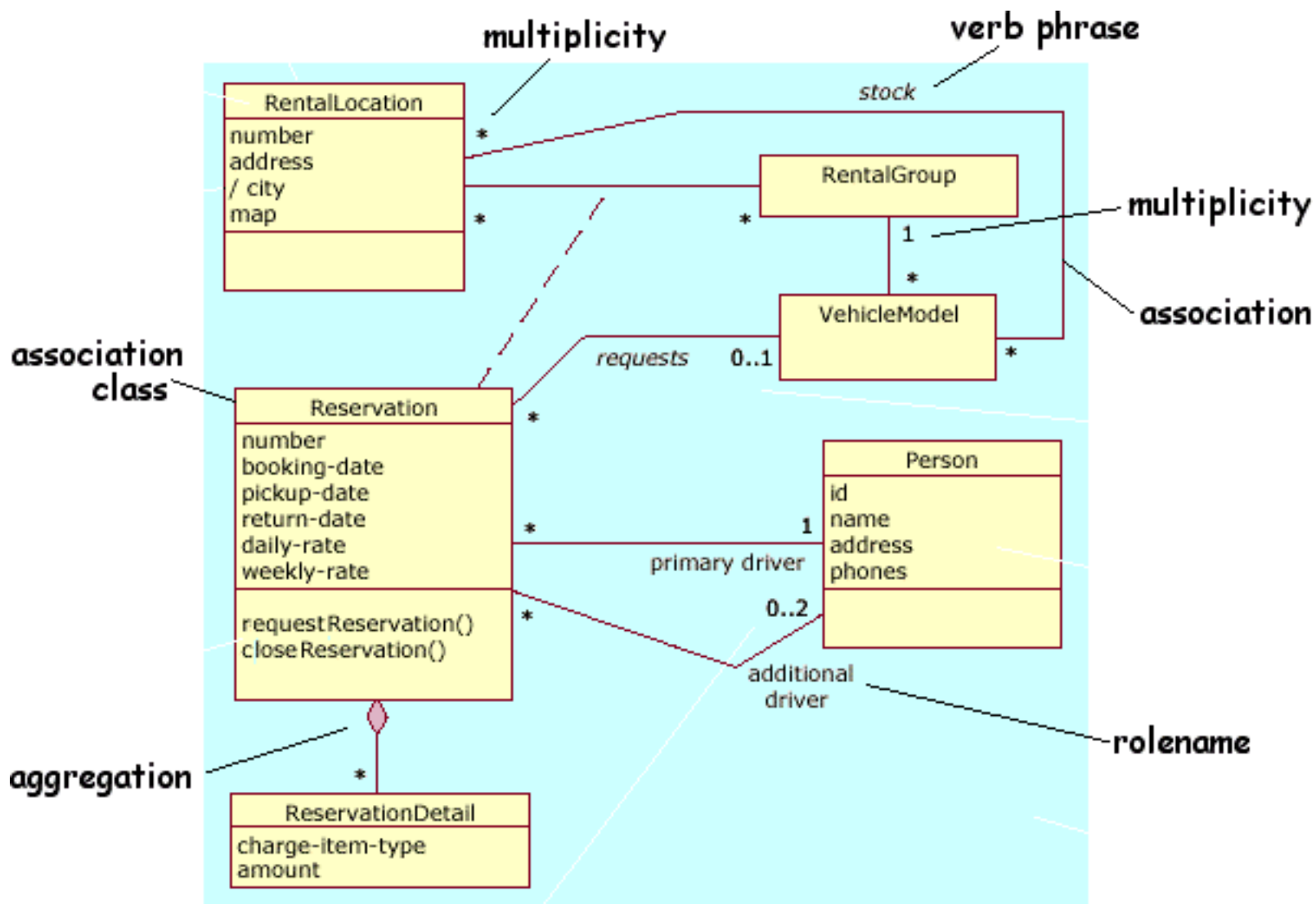


# the Domain Model ~ users' concepts in hierarchies

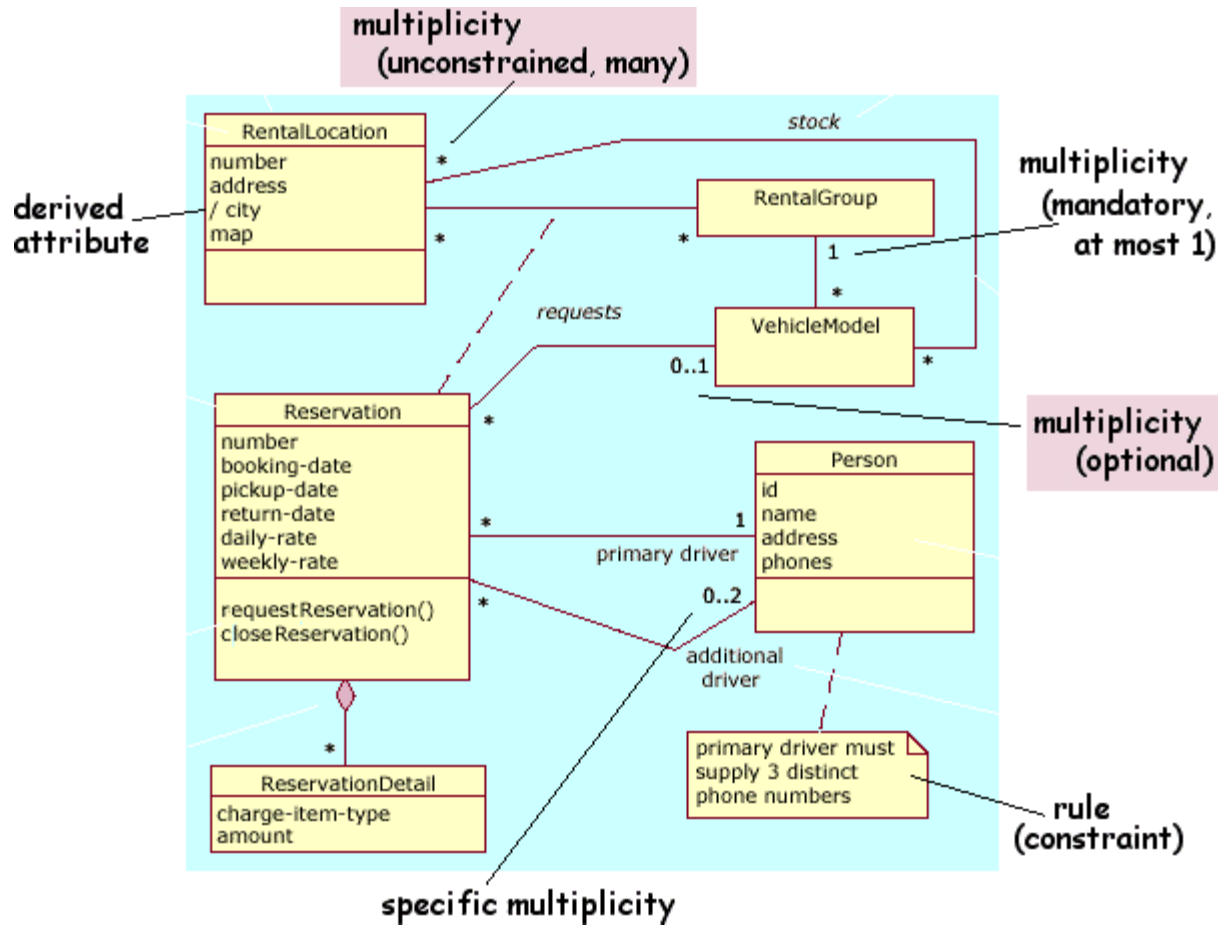
## Class Generalization / Specialization



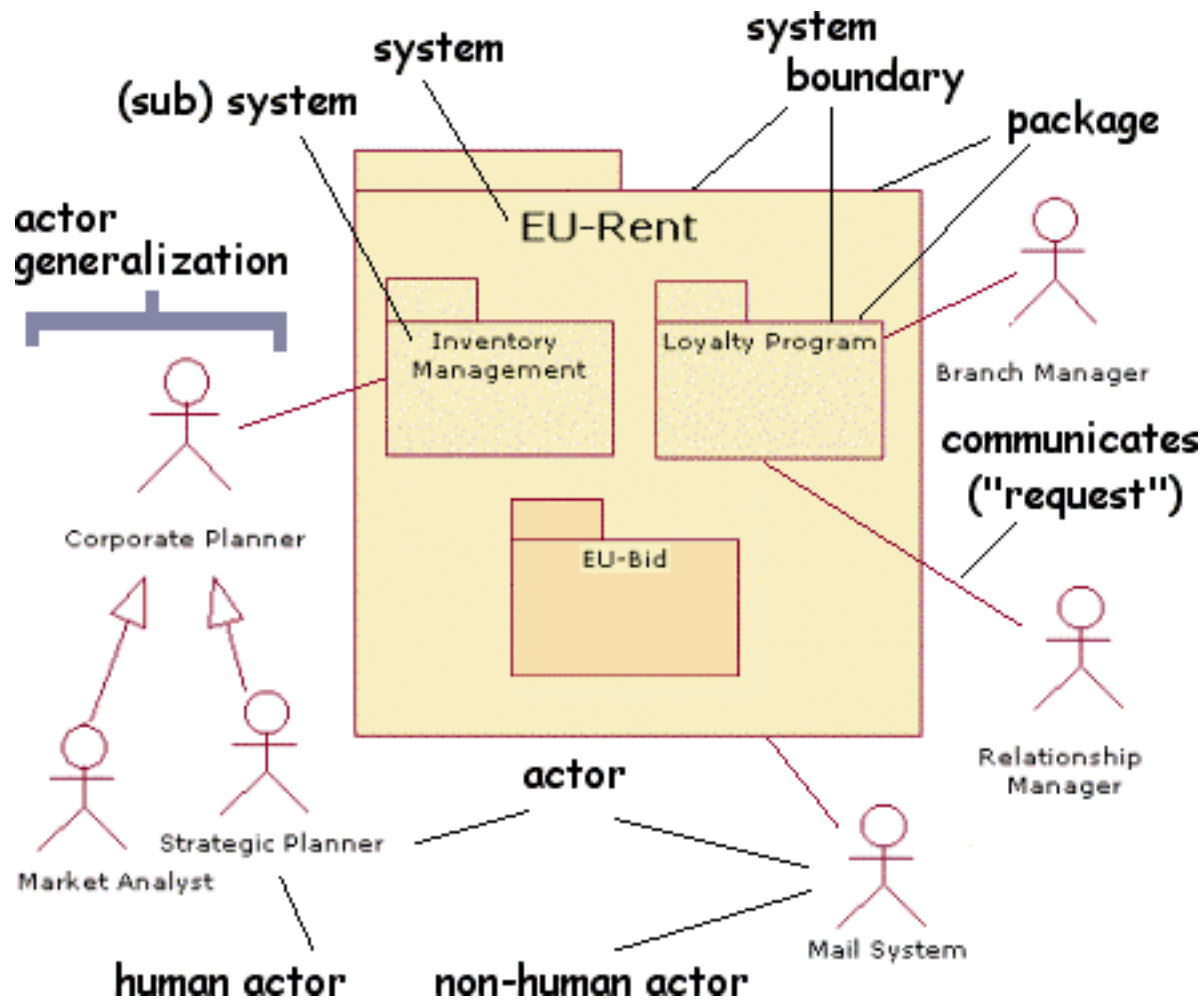
# the Domain Model ~ “facts” the users speak



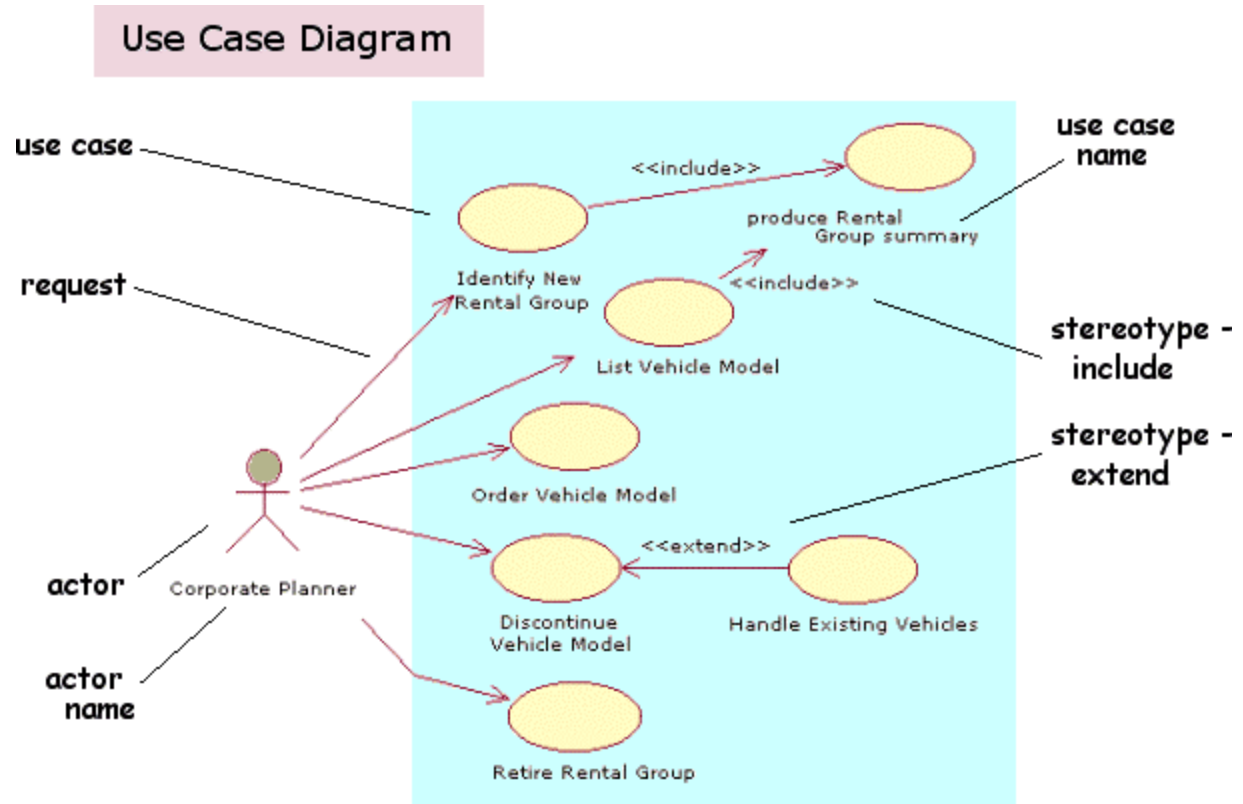
# the Domain Model ~ reflects some of the users' "rules"



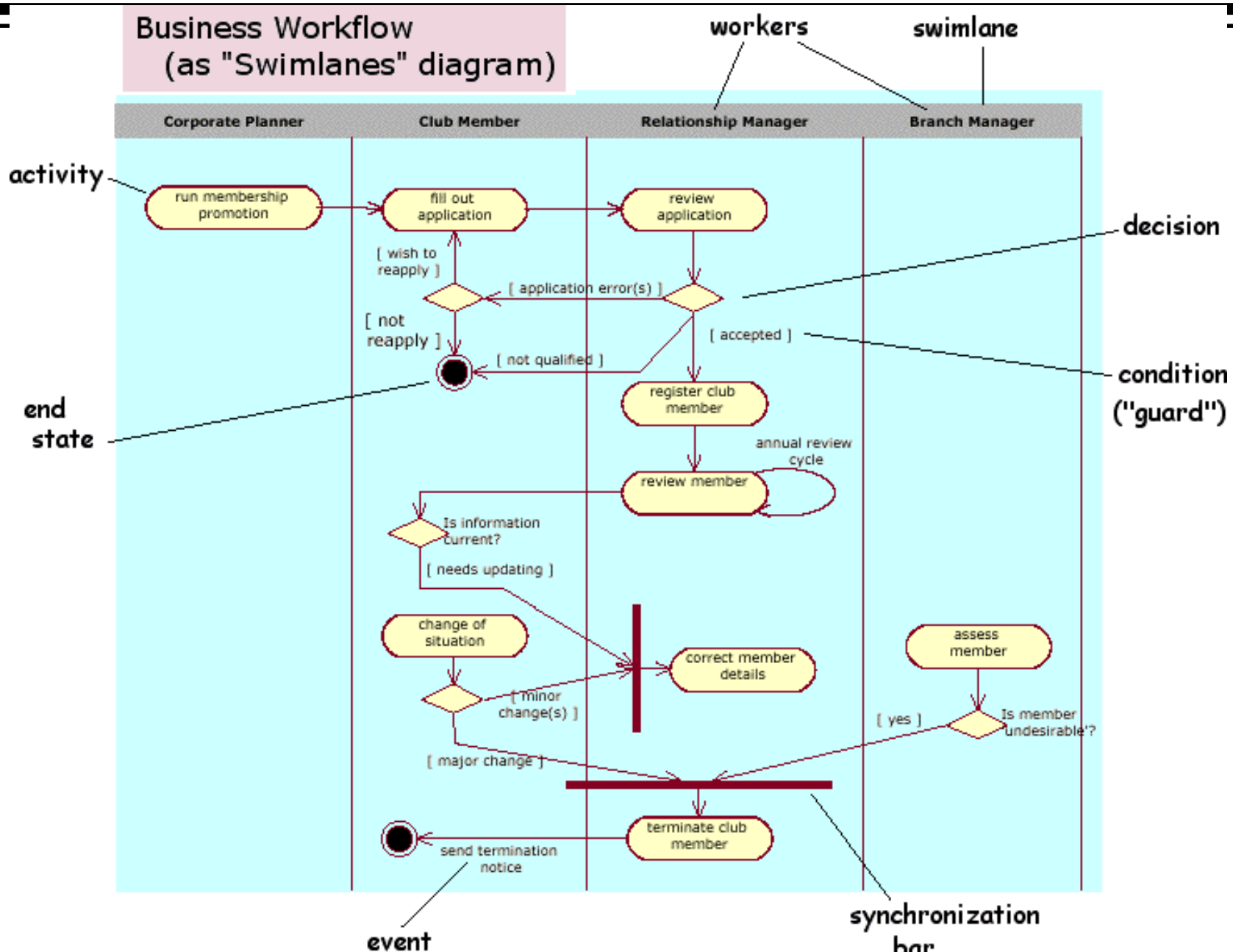
# the Use Case model ~ specifies the functionality (“dynamics”)



# for example: use cases for the Inventory Management subsystem



# How to "discover" the use cases?



... and the workflow, annotated with system support points

