Use Case

informal definition:

A **use case** describes one "case" of how a user can use the system.

– *i.e., a 'system service' that the user can request from the system.*

example:

Register Club Member

Order Vehicle Model

UML definition:

A **use case** is a **sequence of actions** performed by a system that yields an **observable result of value** to a particular actor.

Use Case documentation ~ the initial elements

writing up the Use Case...

For example:

Use Case name:

Order Vehicle Model

Use Case summary, in terms of a brief description of:

the nature of what the actor needs (the system) to do

the observable result of value to the actor

For example, In "Order Vehicle Model" the actor wishes to ...

add more vehicles of a particular model into EU-Rent's vehicle inventory,

increasing the model's vehicle-count of the model's.

More detail will be added (later).

Use Cases

the 6 most important things to know about a Use Case model:

What a Use Case is *NOT*

Where does ‘Use Case’ fit in the development process?

What are the basic elements of a Use Case?

How do I draw a Use Case as a diagram?

How do I discover Use Cases?

After the basics, how do I refine a Use Case?

How do I discover Use Cases?

Business Workflow

Decide the activities in the Business Workflow that call for system support.

These will become our initial “use cases.”

Business Workflow

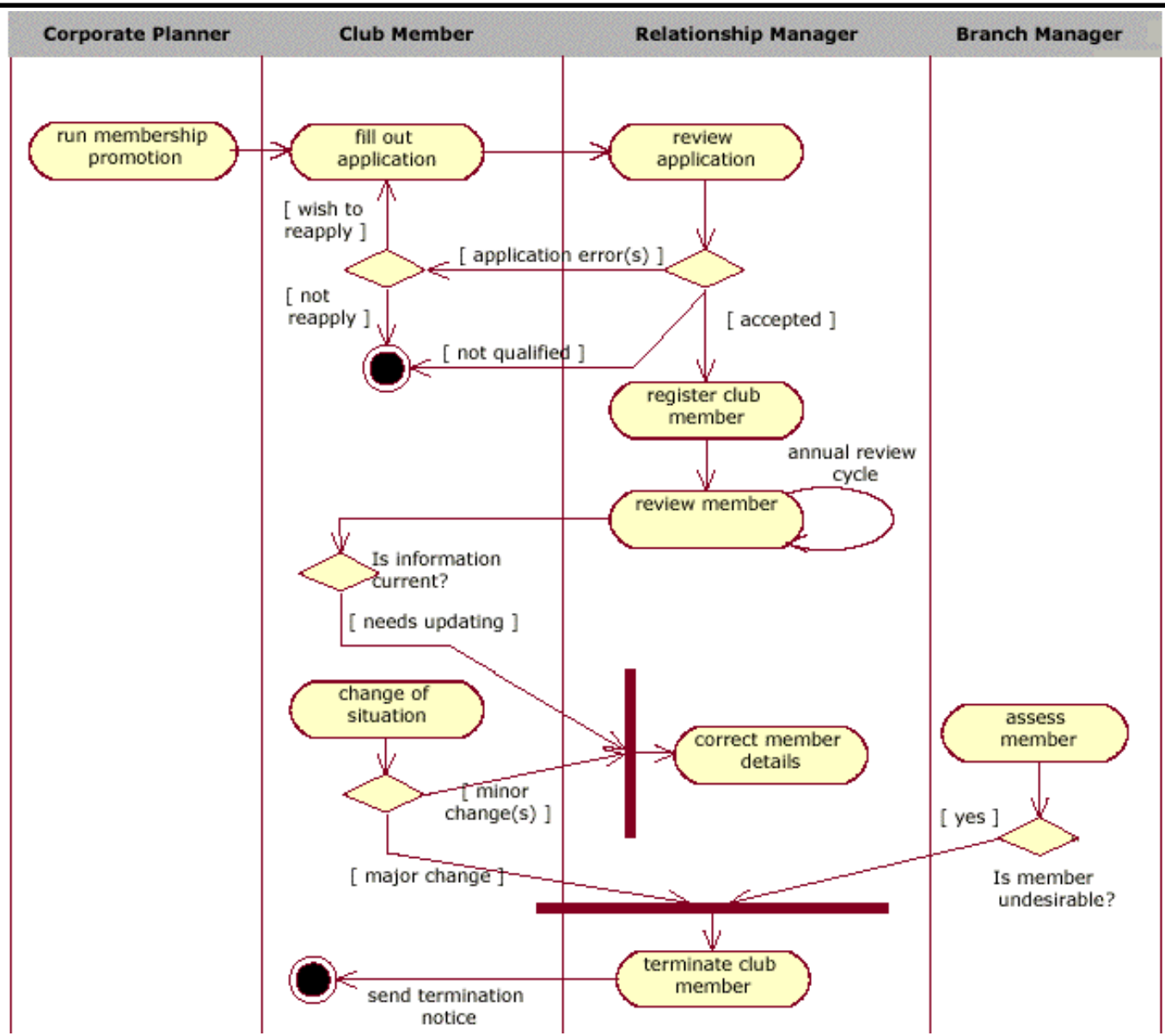
Describes business activities and workers

Drawn as Swimlanes

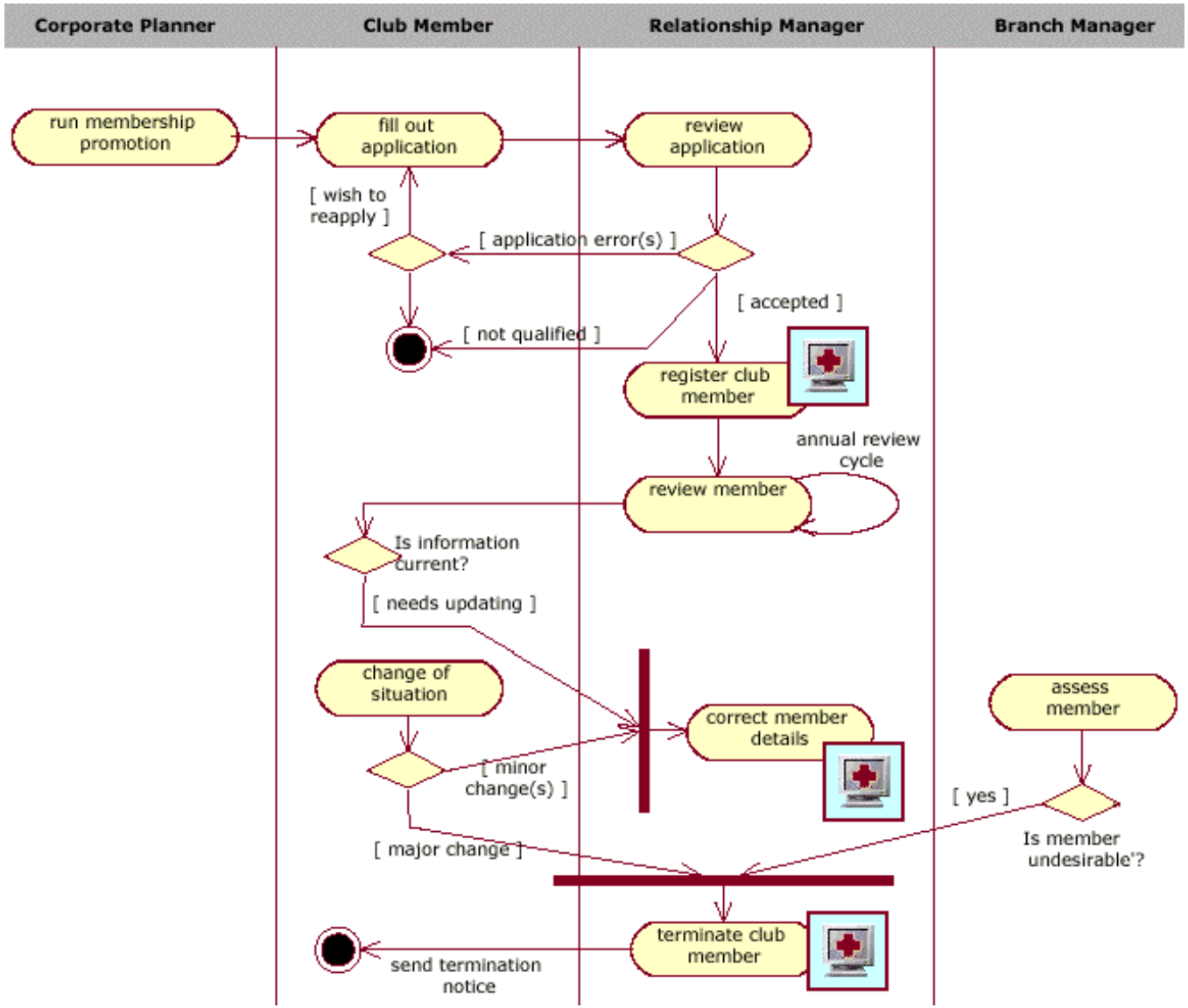
Swimlane - UML definition

A partion on an activity diagram for organizing the responsibilities for actions. Swimlanes typically correspond to organizational units in a business model.

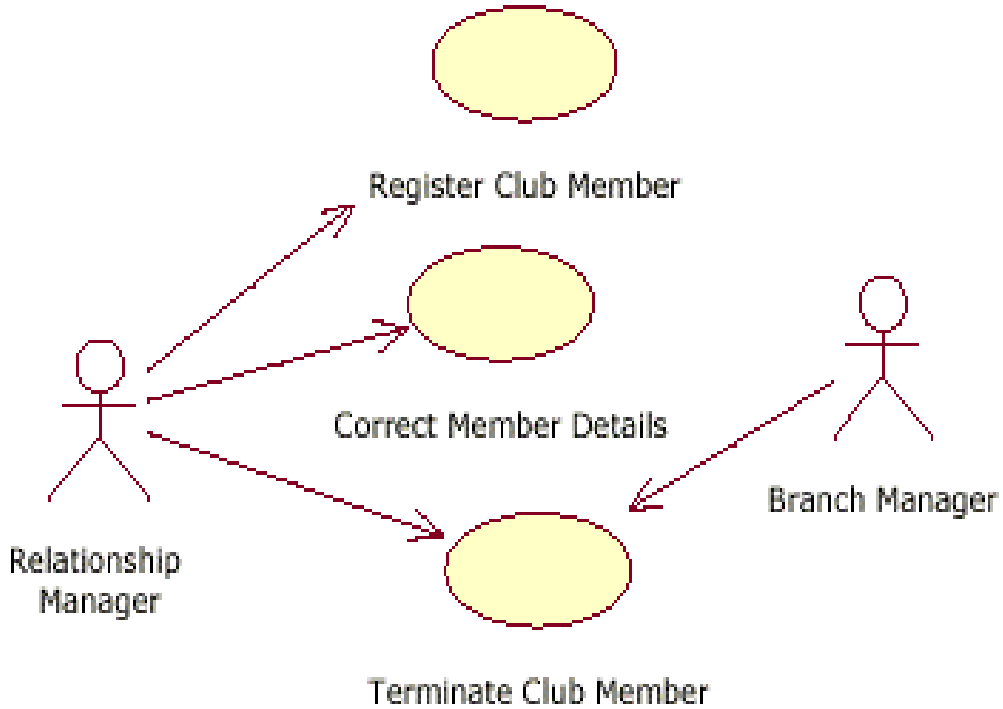
Loyalty Program business workflow as swimlanes diagram



Loyalty Program business workflow, annotated with system support points



Loyalty Program Use Cases



After the basics, how do I refine a Use Case?

In addition to the associations between actors and use-cases, you can link use-cases:

using stereotype extensions:

the <<include>> stereotype

the <<extend>> stereotype

using use-case generalization / specialization

used to describe a specific form of a more general use-case.

In our work, we will only use the two stereotype extensions.

Refinement via stereotype

UML has standard features to customize / extend the language.

The *Guillemet* character denotes a stereotype extension.

<< >>

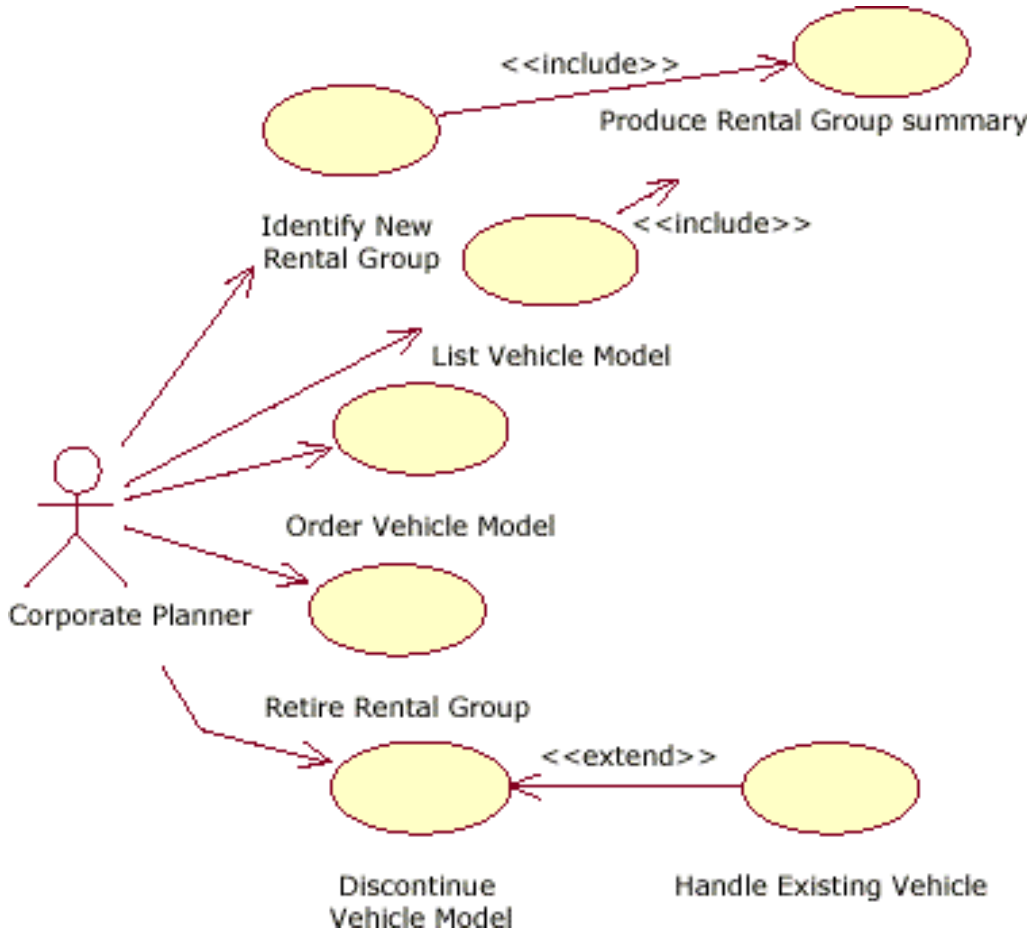
2 stereotype extensions are used to define relationships between Use Cases:

<<include>>

<<extend>>

Note: "UML trivia" ~ a Guillemet *is a single symbol, not two 'less-than' or two 'greater-than' characters.*

Inventory Management



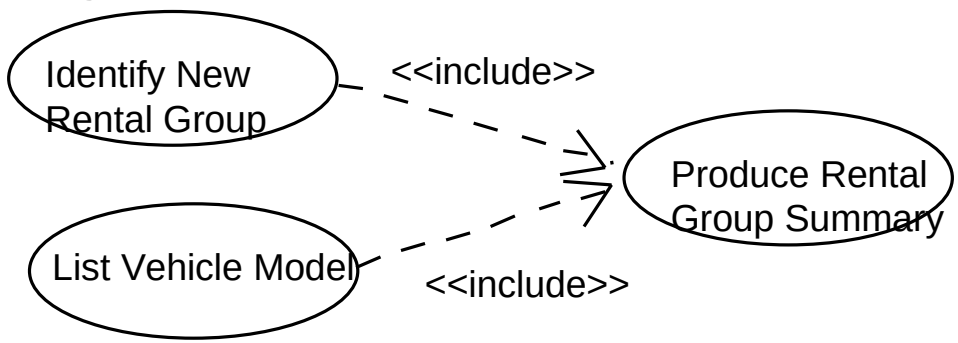
The Use Case <<include>>

- *use <<include>> if you find you are repeating actions in two or more separate use-cases and you want to "factor out" the common actions into one use-case that can be used in many places.*

For example,

You may find that the "List Vehicle Model" use-case in the Inventory Management system also includes the step of listing the groups and models.

This can be factored out into a "Produce Rental Group Summary " use-case which is related to each of the including use-cases.



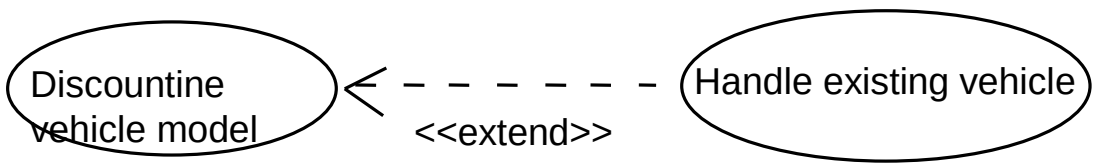
The Use Case <<extend>>

- *use <<extend>> when you need to describe a conditional variation of a base case.*

  The "extension points" will be defined in that base case.

For example,

Discounting a model sometimes involves handling any vehicles that remain on the rental lots.



This is used primarily to represent optional behavior, to handle exceptions, or to simplify complex event flows.

Where do we go from here?

- *Once identified, it can still be hard to decide whether a set of actor-system interactions is a single use-case or several use-cases.*

How do we know one big use case or two little ones?

> "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 from the system

(2) the Information Updates
keeping the system data used in (1) current and correct

- *Next Lessons...*