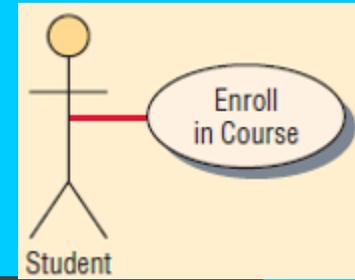


geek & poke



Use Cases

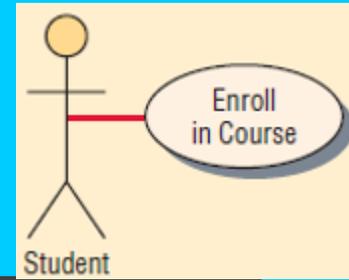
Needed for your champion document



Dilbert.com DilbertCartoonist@gmail.com

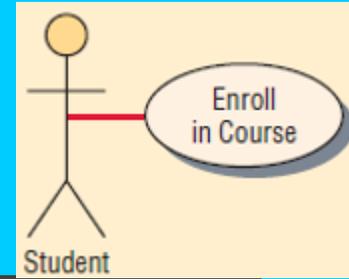
© 2013 Scott Adams, Inc., Dist. by Universal Uclick

Learning Objectives



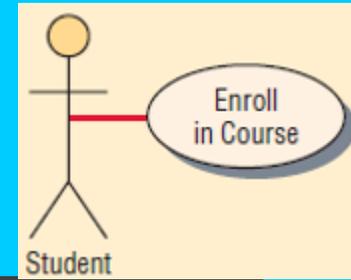
- Use Cases

Depicting Systems Graphically



- ✧ Context-level data flow diagrams
- ✧ Entity-relationship model
- ✧ Use Case Modeling

Analyst hat: Where are we in the SDLC?



✧ Specification

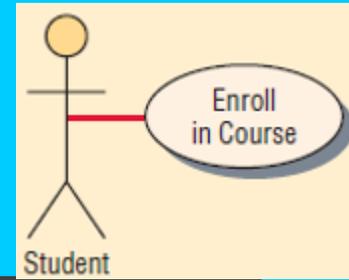
- Identify Problem
- **Gather requirements**
- Analyze & Model requirements
- Select architecture/COTS

✧ Design and implementation

✧ Validation

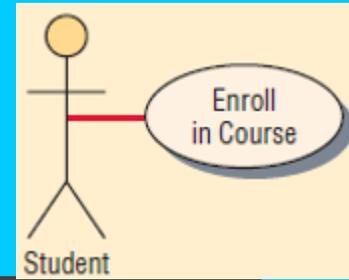
✧ Evolution

Use Case Benefits



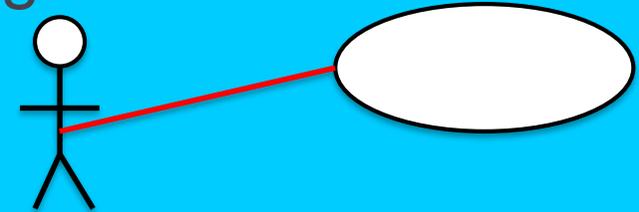
- ✧ A use case provides developers with a view of **what the users want**.
- ✧ It is **free of technical or implementation details**.
 - Non-Technical people can understand them.
- ✧ Use cases are used to document a **single** transaction or event.
 - **An event** is an input to the system that happens at a specific time and place and **causes the system to do something**.

What is a Use Case?



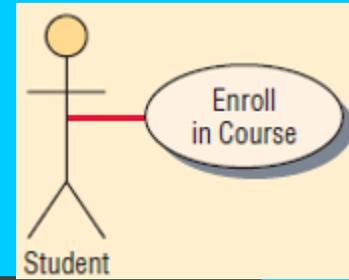
✧ A use case always describes three things:

- an **actor** that initiates an event;
- the **event** that triggers a use case;
- and the **use case** that performs the actions triggered by the event.



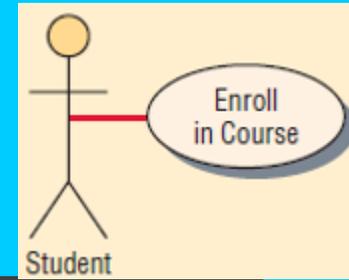
✧ In a use case, an actor using the system initiates an event that begins a related series of interactions in the system.

What is a Use Case?

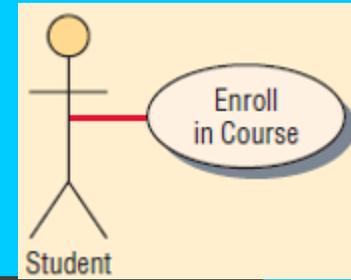


- ✧ Use cases may also be nested, if needed.
- ✧ Some use cases use the verb **manage** to group use cases for adding, deleting, and changing into another, lower-level, use case diagram.
- ✧ You can include a use case on several diagrams, but **the actual use case is defined only once** in the repository. - **DRY: Don't repeat yourself**
- ✧ A use case is **named with a verb and a noun**.

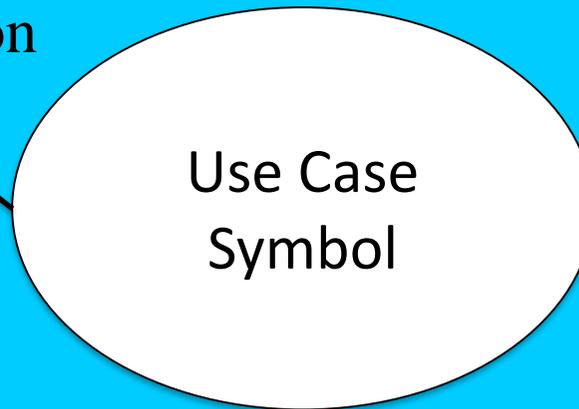
Business Notes



- ✧ For this class you will each create at least one use case for your feature. – **Normally that would be way to many!**
- ✧ It is better to create fewer use cases rather than more. **20 use cases (and no more than 40 or 50) are sufficient for a large system.**



Association



Use Case
Symbol

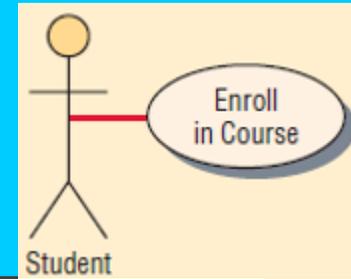
← Scope of
your system

Actor:

A particular role
of a user of the
system.

ALWAYS outside
the scope.

Actors



✧ The term **actor** refers to a particular **role of a user** of the system.

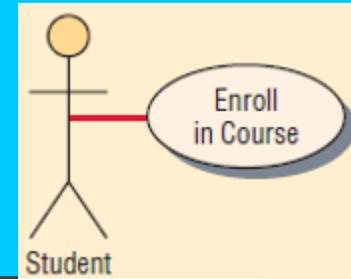
- For example, an actor may be an **employee**, but also may be a **customer** at the company store.
- Even though it is the same person in the real world, it is represented as two different symbols on a use case diagram, because the person interacts with the system in different roles.



✧ The actor exists outside of the system and interacts with the system in a specific way.

✧ **An actor can be a human, another system, or a device such as a keyboard or Web connection.**

Actors



✧ Actors can initiate an instance of a use case.

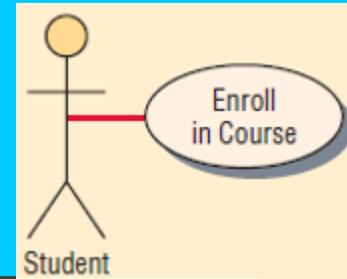
- An actor may interact with one or more use cases, and
- a use case may involve one or more actors.

✧ Actors may be divided into two groups.

- **Primary actors supply data or receive information from the system.**
 - Some users directly interact with the system (system actors), but primary actors may also be businesspeople who do not directly interact with the system but have a stake in it.
- **Supporting actors** (also called secondary actors) **help to keep the system running or provide other services.** (i.e help desk, the analysts, programmers, etc.)



Use Case Associations



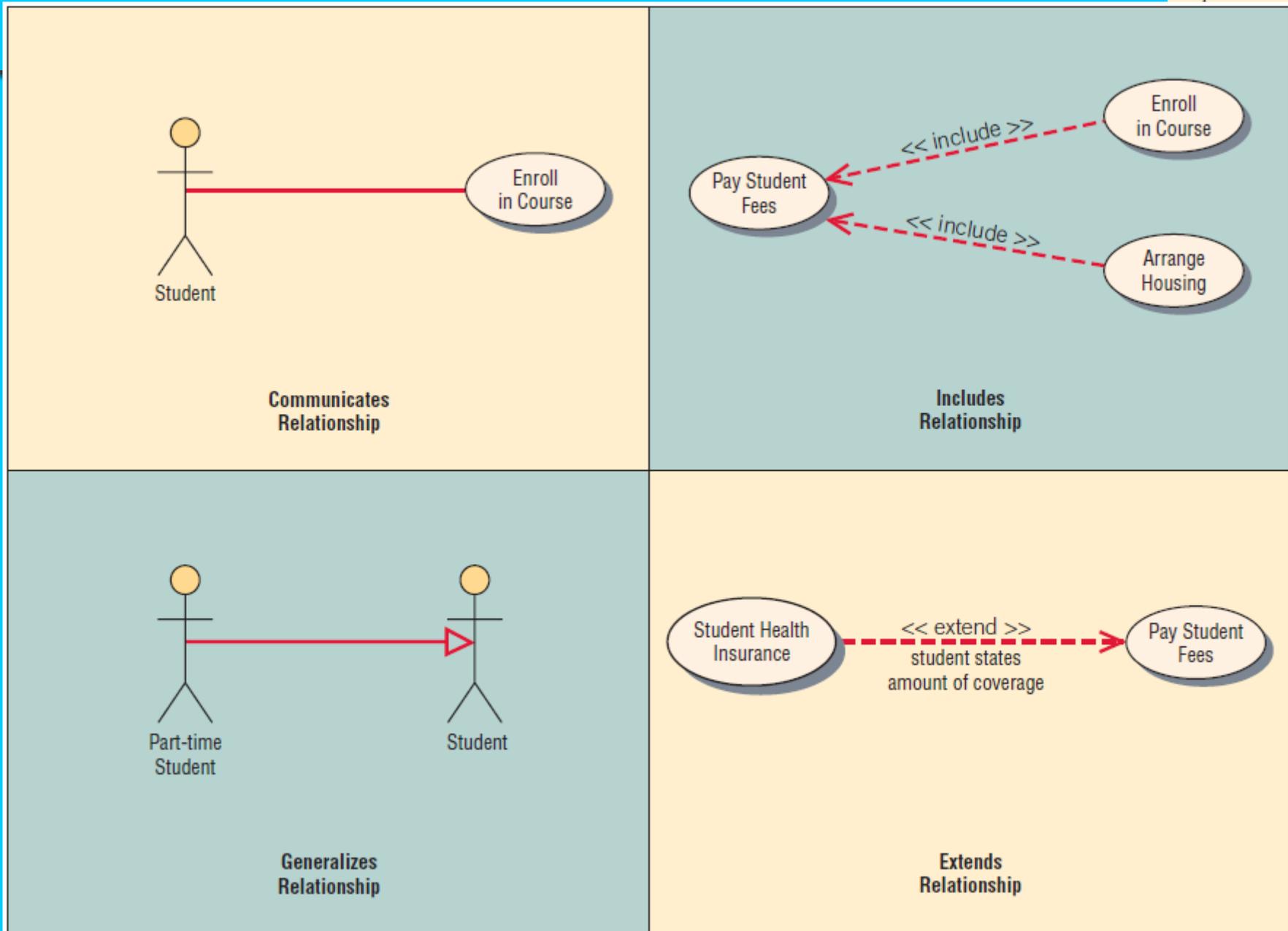
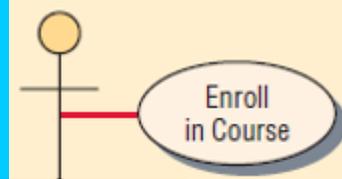
| Relationship | Symbol | Meaning |
|---------------------|--------|--|
| Communicates | | An actor is connected to a use case using a line with no arrowheads. |
| Includes | | A use case contains a behavior that is common to more than one other use case. The arrow points to the common use case. |
| Extends | | A different use case handles exceptions from the basic use case. The arrow points from the extended to the basic use case. |
| Generalizes | | One UML "thing" is more general than another "thing." The arrow points to the general "thing." |

Copyright ©2011 Pearson Education, Inc. publishing as Prentice Hall

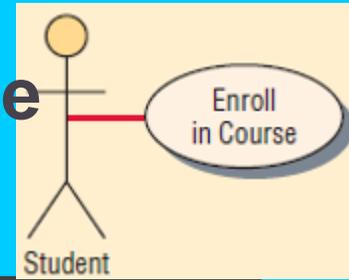
Extend is different from include because it is used for exception cases

Old Mid-term Question: Name, draw the symbol and define the four basic behavioral relationships found in Use Case Diagrams. Give an example of each in use in a diagram

Example of Relationships in use



Quick Quiz: Which of these is not an appropriate Generalization?

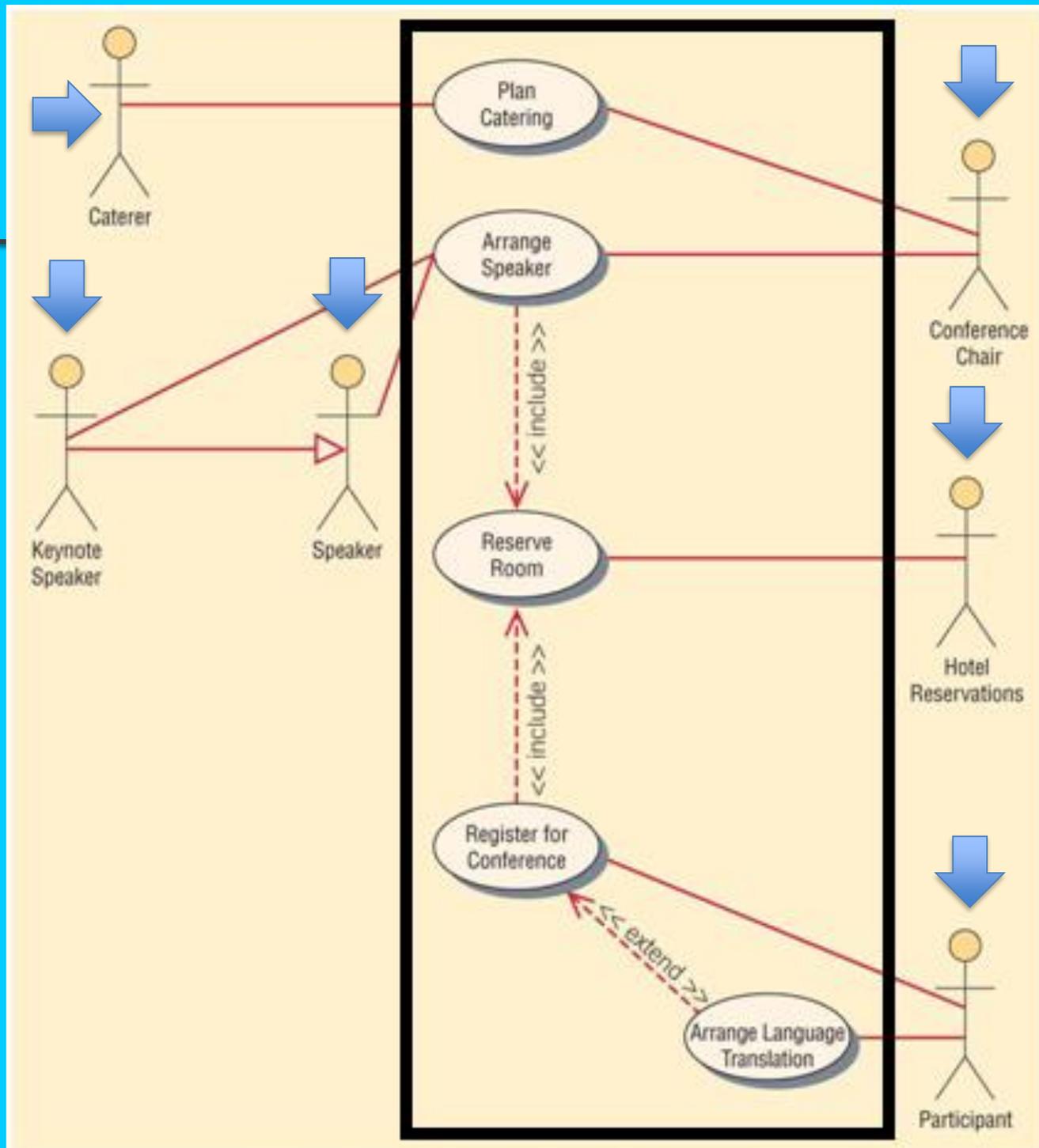


- ❖ ~~Between actor and use case~~
- ❖ Between use cases
- ❖ Generalisation of actors
 - Types of users
- ❖ Generalisation of use cases
 - Shows inheritance and specialization
 - One use case is simply a special kind of another

A system used to plan a conference

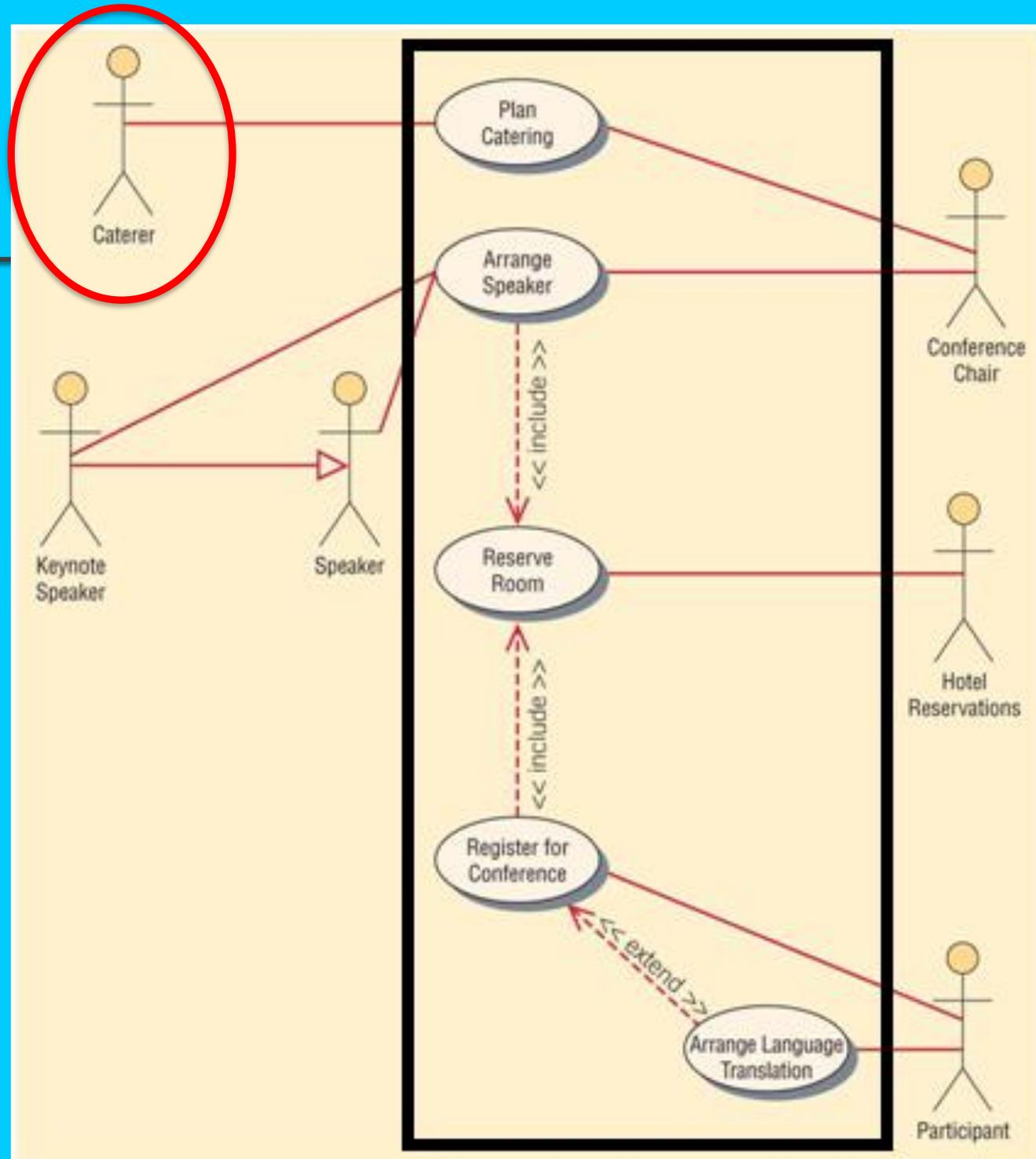
❖ The actors are

- the **Conference Chair**, responsible for planning and managing the conference,
- the conference **Participant**,
- **Speakers**,
- a **Keynote Speaker**,
- **Hotel Reservations**, and
- a **Caterer**.



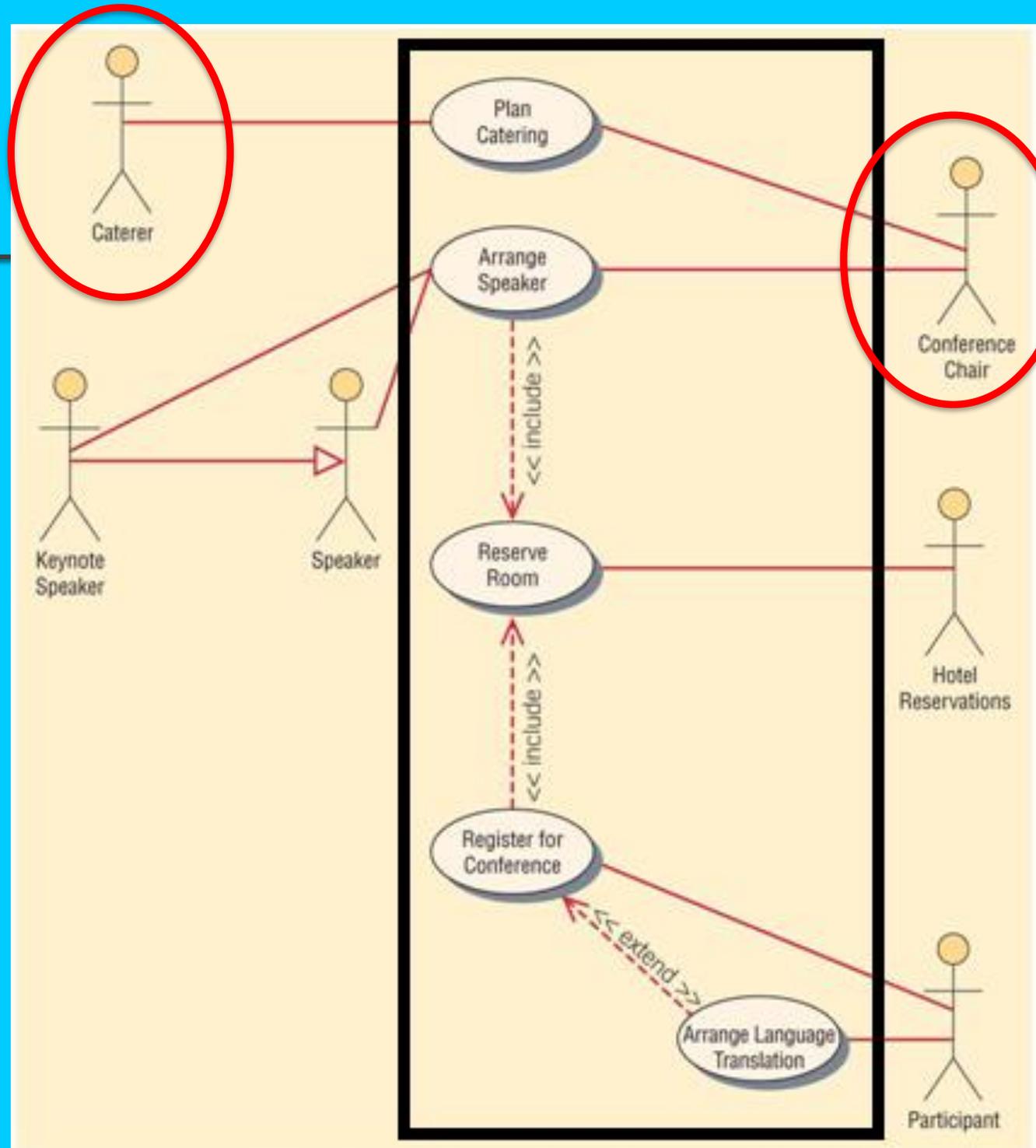
A system used to plan a conference

- ✧ Actors represent the *role* the user plays,
 - the **Caterer** may be either a hotel employee or an external catering service.



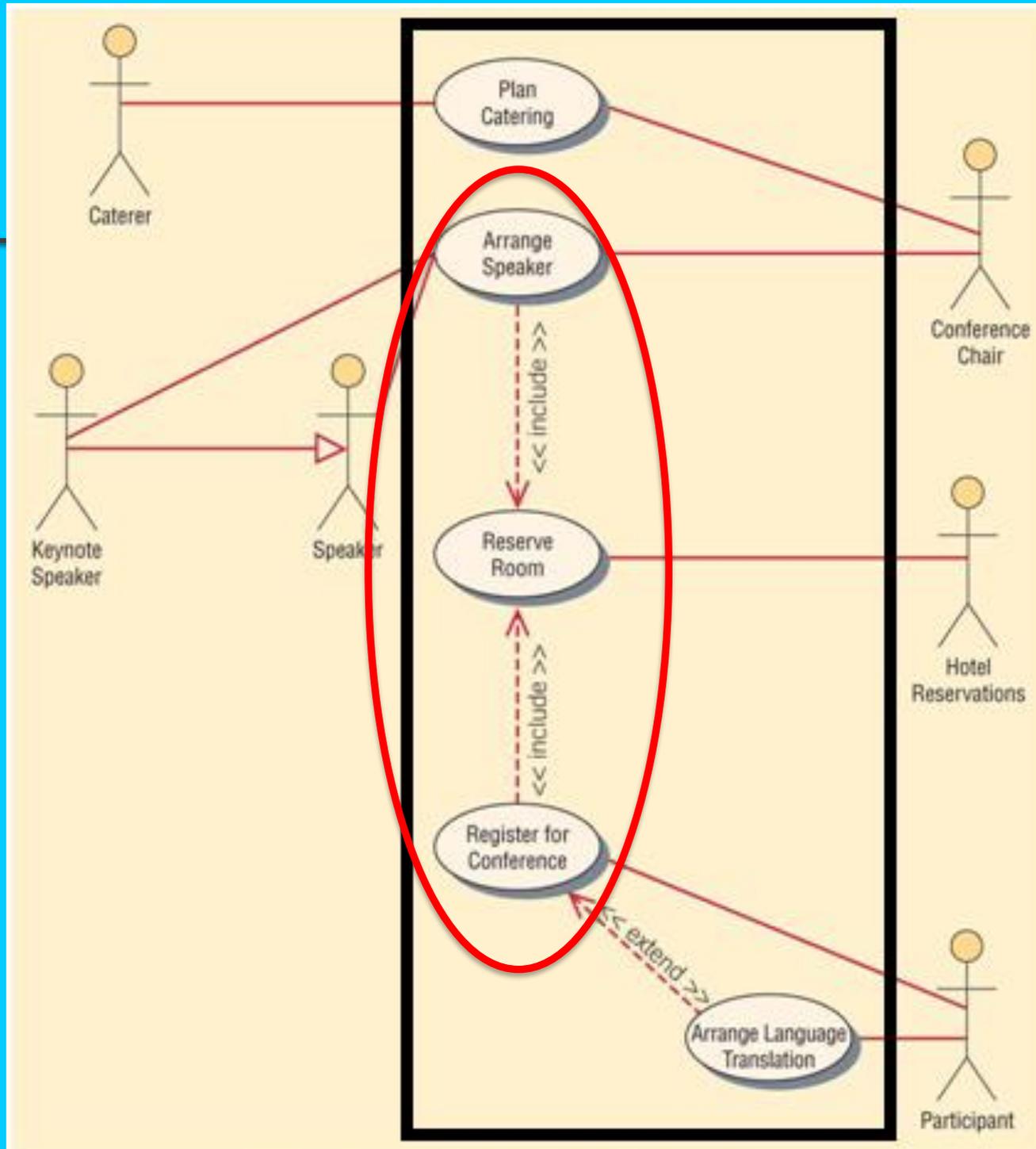
A system used to plan a conference

- Both the **Conference Chair** and the **Caterer** are involved in planning meals and banquets.
- The **Conference Chair** is also responsible for arranging speakers.



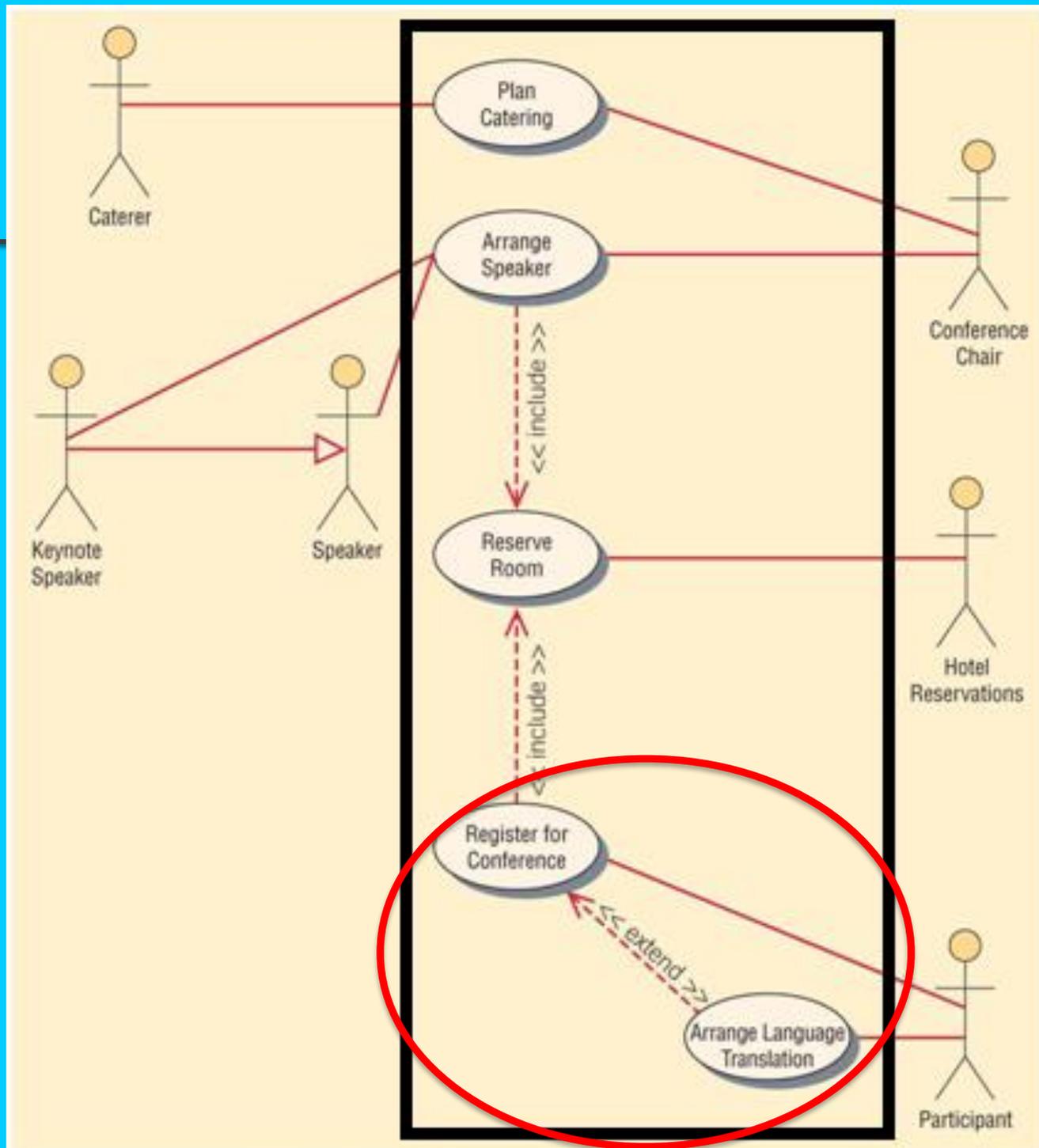
A system used to plan a conference

- Notice that the **Reserve Room** use case is involved in an *includes* relationship with the **Arrange Speaker** and **Register for Conference** use cases, since both speakers and participants will need lodging.



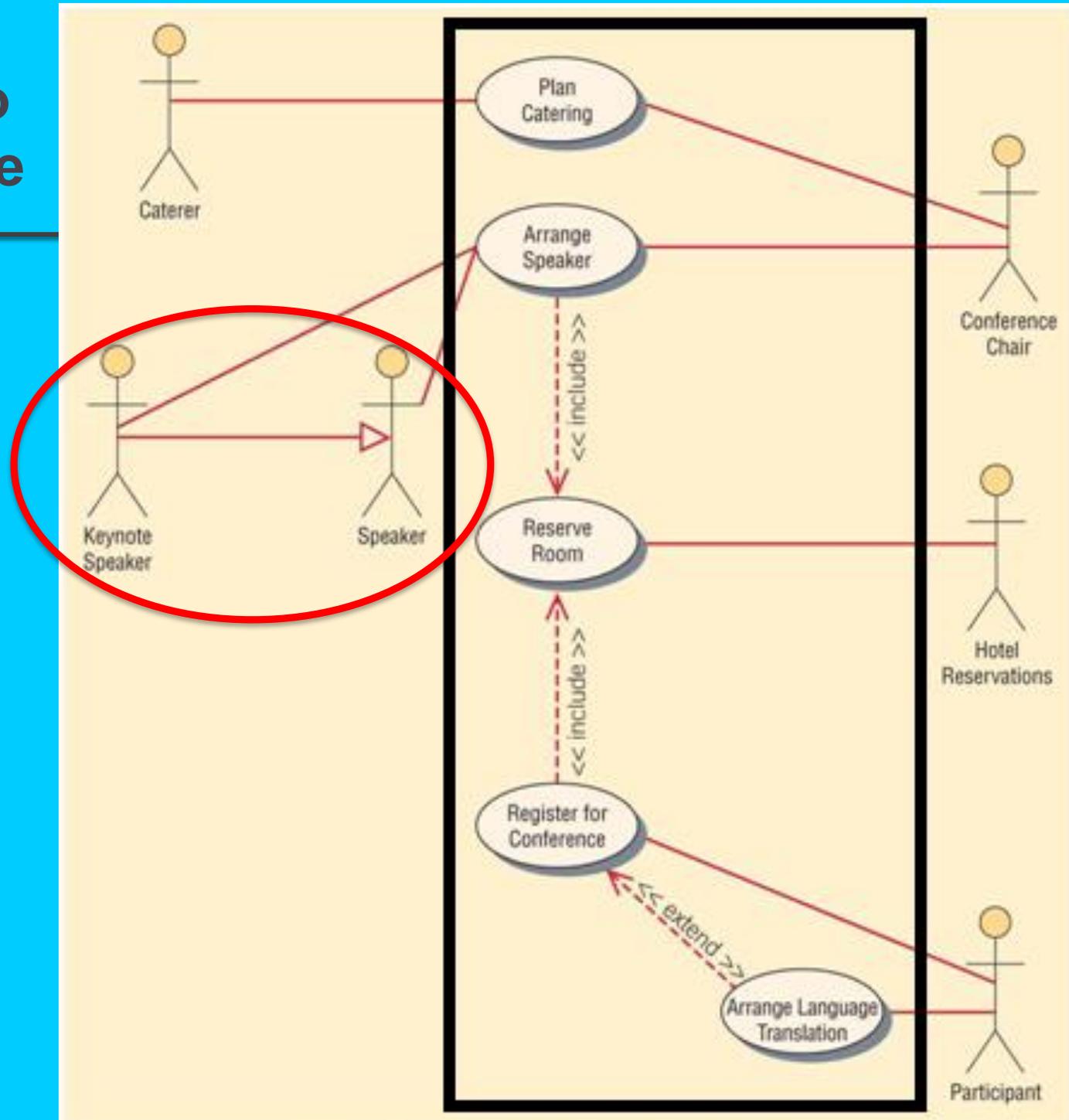
A system used to plan a conference

- ✧ The **Arrange Language Translation** use case extends the **Register for Conference** use case because not all participants will require language translation services.

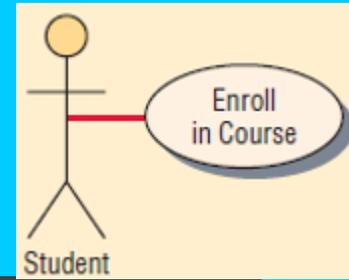


A system used to plan a conference

✧ The **Speaker** actor is a generalization of **Keynote Speaker**.

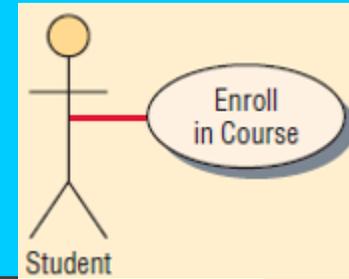


Why Use Case Diagrams Are Helpful



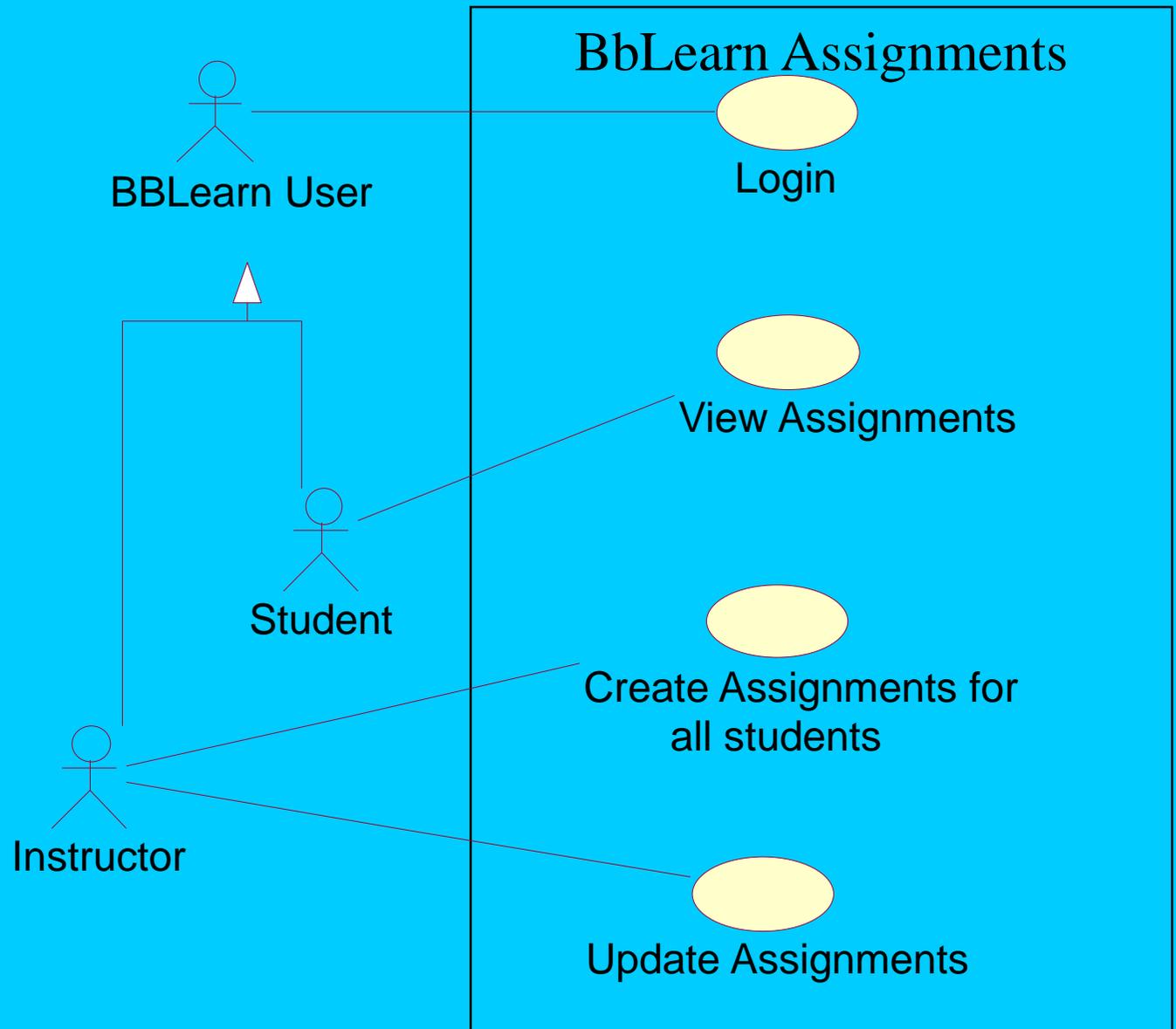
- ✧ Identify all the actors in the problem domain
- ✧ Actions that need to be completed are also clearly shown on the use case diagram
- ✧ The use case scenario is also worthwhile
- ✧ Simplicity and lack of technical detail

Generalisation of Actors

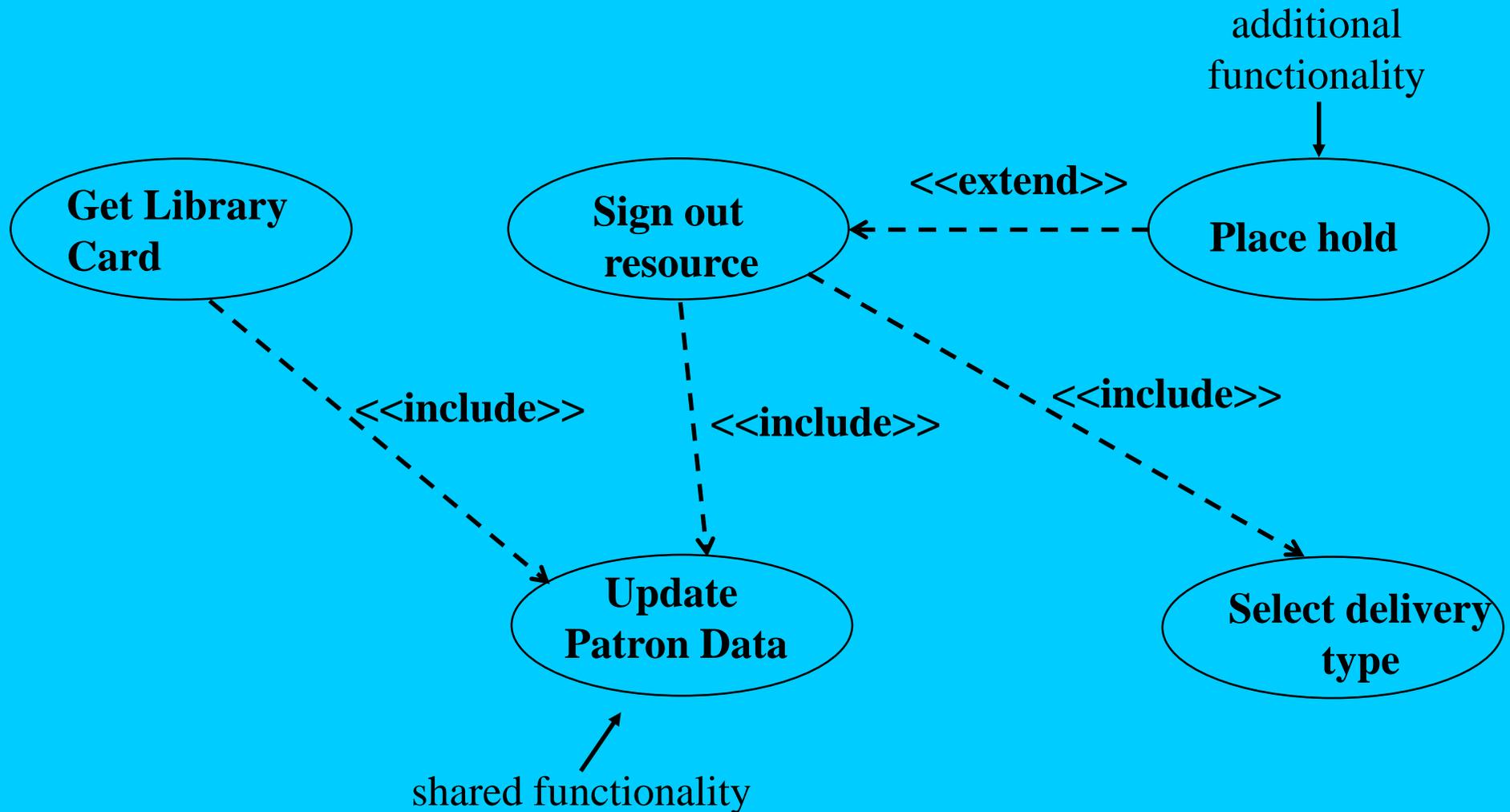
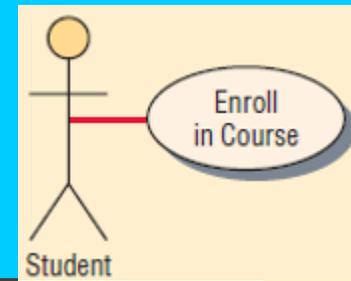


Question:
Should Login
be a use case?

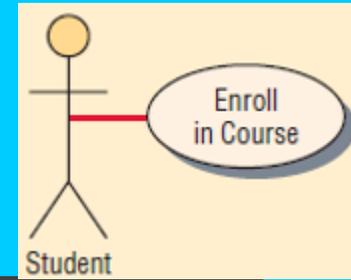
Question:
Could Create
Assignments
have another
type of
relationship?



Example of Use Case Relationships



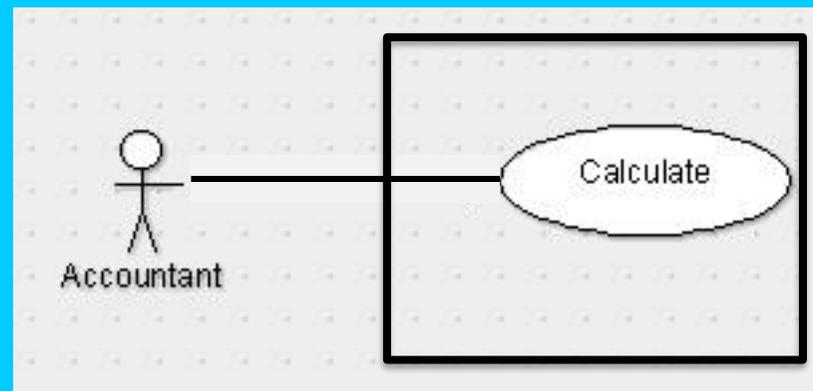
Example: the computer, circa 1850



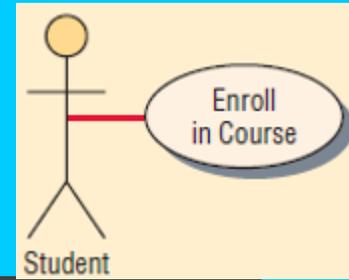
Problem: Accountants need a simple but reliable way to add, multiply, divide and subtract large quantities of numbers

Existing systems work but are cumbersome or not accurate enough:

- counting on fingers and toes
- Abacus
- Pascal's Adder
- Leibniz's Calculating Machine

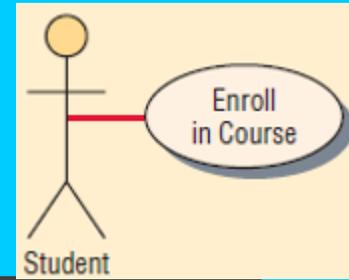


Functional Requirements: What the system shall do

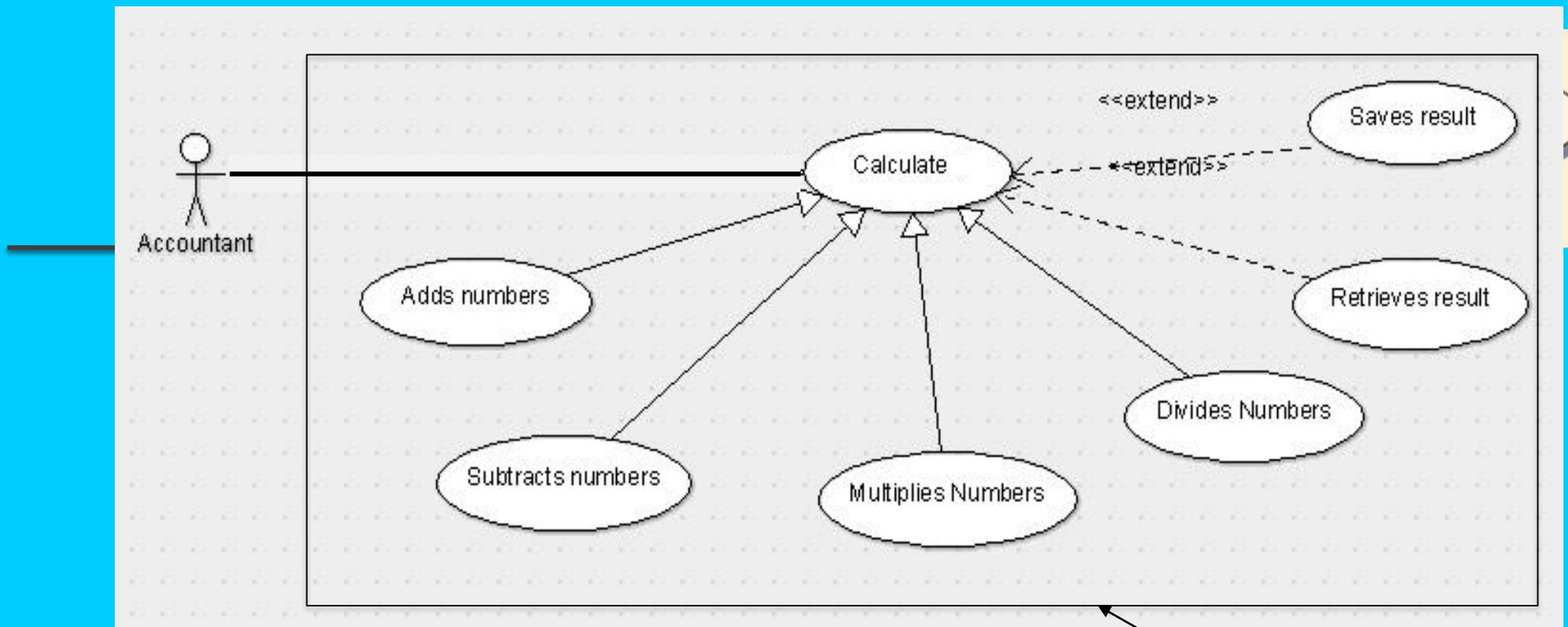


- Shall allow addition, subtraction, multiplication and division of two or more numbers
 - Shall allow repeated operations
 - Ideally should tabulate a running total
 - Must support signed and unsigned decimal #s
 - Nice to support percentages
 - Nice to allow saving and retrieval of results
 - ...
- Note: "Nice to have"s do not belong in your RFP

Non-functional Requirements: What the system shall be



- Shall be small enough to fit into accounting office
- Shall not require more than 2 people to operate
- Shall take no more than 1 minute to calculate a result
- ...



Use Case diagram

Use Case specification: **Add numbers**

Pre-conditions: Calculator has been initialized

End-state (success): Calculated value is displayed

Steps:

Accept input of first number

Accept input of second number

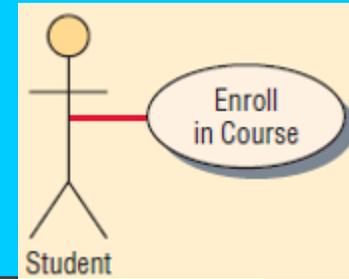
...

Accept input of nth number

Accept calculate command

Calculate and show result

Use Case Exercise

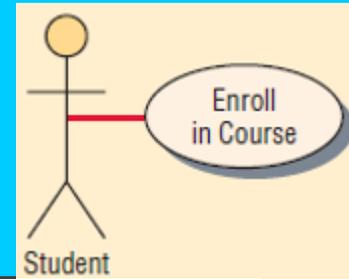


✧ Consider Mario Brothers.

1. Give an example of a **functional requirement** for Mario Brothers? What the system **shall do**.
2. Give an example of a **non-functional requirement** for Mario Brothers. What the system **shall be**.



Use Case Exercise



✧ Consider Mario Brothers.

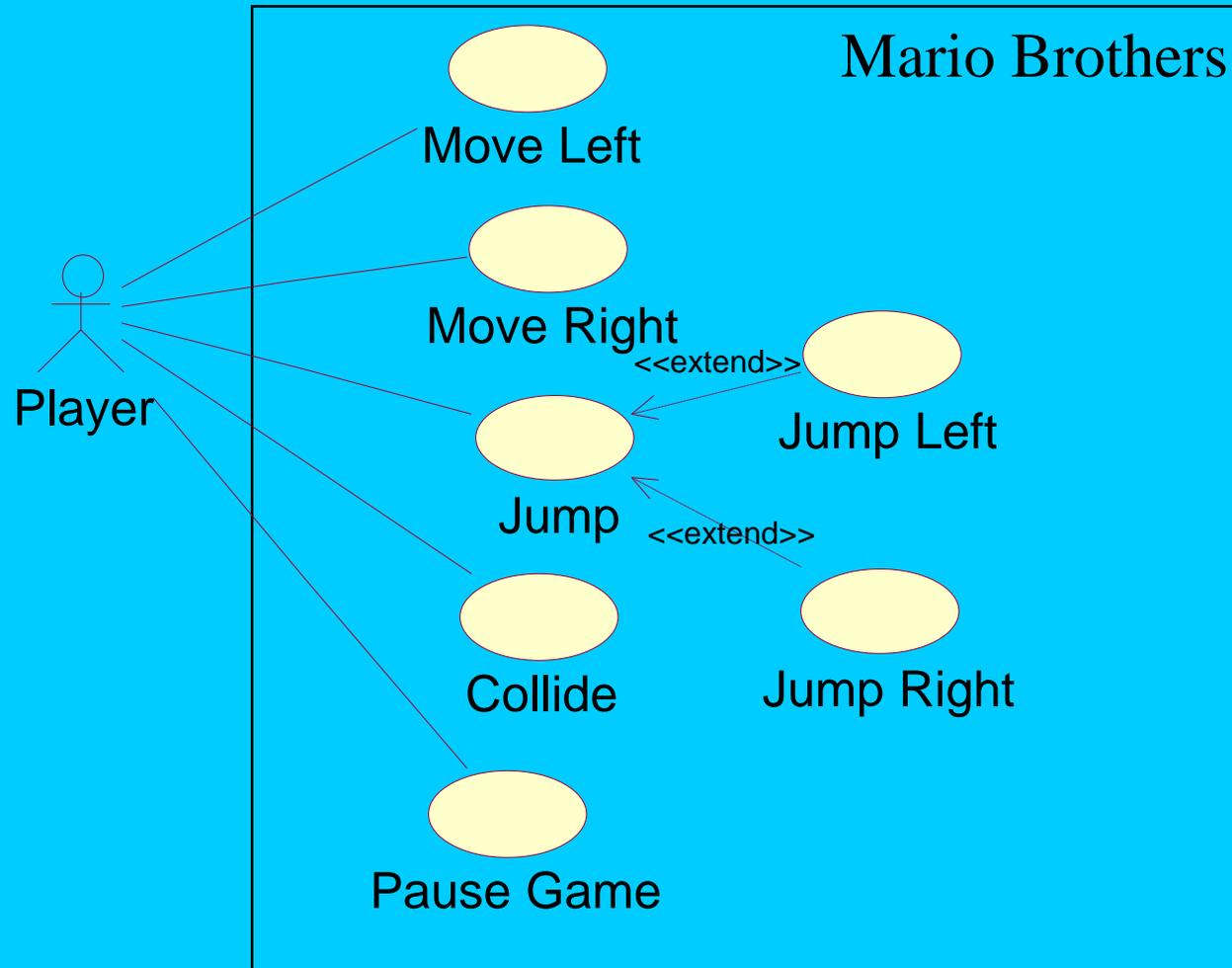
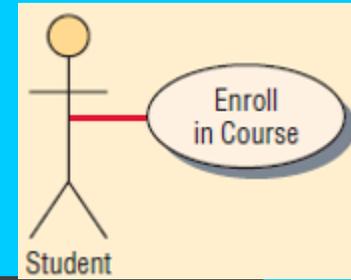
1. What can Mario do?
2. What are the use cases?
3. Are there any relationships
4. Draw the use case diagram

1. Move left
2. Move right
3. Jump
4. Jump left
5. Jump right
6. Collide
7. Pause

Probably should break
collide down. (Manage)



Possible Solution



Use Case Scenario

✧ Each Use case diagram will have a corresponding Use Case Scenario.

| | | | |
|------------------------------------|---|--|-------------|
| Use case name: | Register for Conference | UniqueID: | Conf RG 003 |
| Area: | Conference Planning | | |
| Actor(s): | Participant | | |
| Stakeholder: | Conference Sponsor, Conference Speakers | | |
| Level: | Blue | | |
| Description: | Allow conference participant to register online for the conference using a secure Web site. | | |
| Triggering Event: | Participant uses Conference Registration Web site, enters userID and password, and clicks the logon button. | | |
| Trigger type: | <input type="checkbox"/> External <input checked="" type="checkbox"/> Temporal | | |
| Steps Performed (Main Path) | | | |
| | Steps Performed (Main Path) | Information for Steps | |
| 1. | Participant logs in using the secure Web server. | userID, Password | |
| 2. | Participant record is read and password is verified. | Participant Record, userID, Password | |
| 3. | Participant and session information is displayed on the Registration Web page. | Participant Record, Session Record | |
| 4. | Participant enters information on the Registration Web form and clicks Submit button. | Registration Web Form | |
| 5. | Registration information is validated on the Web server. | Registration Web Form | |
| 6. | Registration Confirmation page is displayed to confirm registration information. | Confirmation Web Page | |
| 7. | Credit card is charged for registration fees. | Secure Credit Card Web Page | |
| 8. | Add Registration Journal record is written. | Confirmation Web Page | |
| 9. | Registration record is updated on the Registration Master. | Confirmation Web Page, Registration Record | |
| 10. | Session record is updated for each selected session on the Session Master. | Confirmation Web Page, Session Record | |
| 11. | Participant record is updated for the participant on the Participant Master. | Confirmation Web Page, Participant Record | |
| 12. | Successful Registration Confirmation Web page is sent to the participant. | Registration Record Confirmation Number | |
| Preconditions: | Participant has already registered and has created a user account. | | |
| Postconditions: | Participant has successfully registered for the conference. | | |
| Assumptions: | Participant has a browser and a valid userID and password. | | |
| Success Guarantee: | Participant has registered for the conference and is enrolled in all selected sessions. | | |
| Minimum Guarantee: | Participant was able to logon. | | |
| Requirements Met: | Allow conference participants to be able to register for the conference using a secure Web site. | | |
| Outstanding Issues: | How should a rejected credit card be handled? | | |
| Priority: | High | | |
| Risk: | Medium | | |

Player scenario

Name: Create character

Summary: The player connects the game website and makes character for him/her.

Actors: The player of the game.

Preconditions: The creator of the game has created a game and submitted it into the system.

Basic sequence:

Step 1: The player connects the game website.

Step 2: The player chooses Create Character option.

Step 3: The player selects the game to which he/she wants to create a character for.

Step 4: The player fills in the needed information (list of information is provided by the game creator).

Step 5: The player enters a username.

Step 6: The player enters and verifies a password.

Exceptions:

Step 4: Some of the filled in information is not acceptable. The player is asked to correct false information.

Step 5: The username could be already in use. The player is asked to choose a new username.

Step 6: The password is not acceptable. The player is asked to choose a new password.

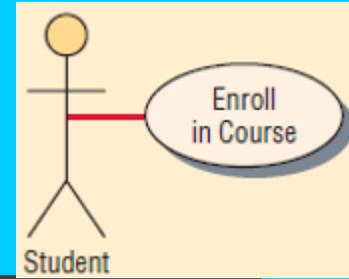
Post conditions: The player has now a ready character for this game.

Priority: 2*

ID: P01

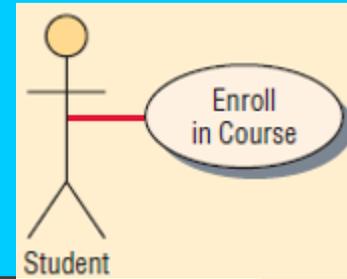
*The priorities are 1 = must have, 2 = essential, 3 = nice to have.

Use Case Levels (defined by Alistair Cockburn)



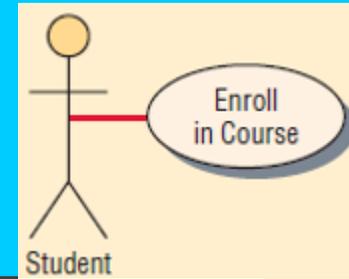
- ✧ **White** is the highest level, like clouds. This is the enterprise level, and there may only be four to five for the entire organization. Examples might be to advertise goods, sell goods to customers, manage inventory, manage the supply chain, and optimize shipping.
- ✧ **Kite** is lower than white but still a high level, providing an overview. The kite use case may be at the business unit or department level and is a summary of goals. Examples would be to register students, or if working with a travel company: make an airline, hotel, car, or cruise reservation.

Use Case Levels (defined by Alistair Cockburn)



- ✧ **Blue** is at sea level, and is usually created for user goals. This often has the greatest interest for users and is easiest for a business to understand. It is usually written for a business activity and each person should be able to do one blue level activity in anywhere from 2 to 20 minutes. Examples are register a continuing student, add a new customer, place an item in a shopping cart, and order checkout.
- ✧ **Indigo** or fish is a use case that shows lots of detail, often at a functional or subfunctional level. Examples are choose a class, pay academic fees, look up the airport code for a given city, and produce a list of customers after entering a name.

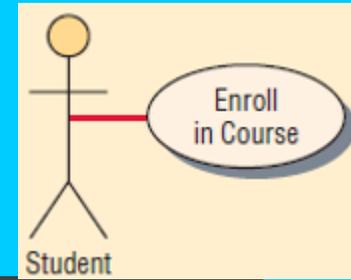
Use Case Levels (defined by Alistair Cockburn)



- ✧ **Black** or clam, like the bottom of the ocean, are the most detailed use cases, at a subfunction level. Examples might be a secure logon validation, adding a new field using dynamic HTML, or using Ajax to update a Web page in a small way.



To Do



- ✧ Decide with your group who will champion which features.
- ✧ For your feature sketch out a use case diagram and scenario.