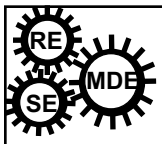
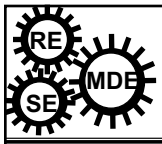


Use Cases and Scenarios

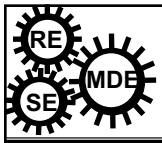
1



We Will Cover

- What is a use-case
  - Use-case versus user interaction
- Use-Case diagrams
  - The constructs in the use-case diagrams
- Capturing the use-case
  - High-level use-case
  - Extended use-case
  - Difference between use case and scenario

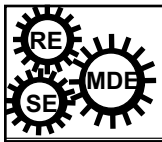
2



What is a Use-Case

- A use-case captures some user visible function
- This may be a large or small function
  - Depends on the level of detail in your modeling effort
- A use-case achieves a discrete goal for the user
- Examples
  - Format a document
  - Request an elevator
- How are the use cases found (captured or elicited)?

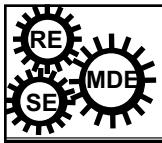
3



User Goals versus User Interactions

- Consider the following when formatting a document
  - Define a style
  - Change a style
  - Copy a style from one document to the next
  - versus
  - Format a document
  - Ensure consistent formatting of two documents
- The latter is a user goal
  - Something the user wants to achieve
- The former are user interactions
  - Things the user does to the system to achieve the goal

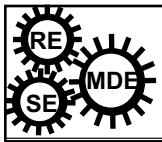
4



Goals and Interactions

- There is a place for both goals and interactions
- Understand what the system shall do
  - Capture the ***user goals***
- Understand how the user will achieve the goals
  - Capture ***user interactions***
  - Sequences of user interactions
- Thus, start with the user goals and then refine the user goals into several (many) user interactions

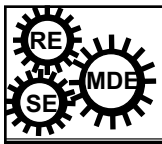
5



Point of Sale Terminal (POST)

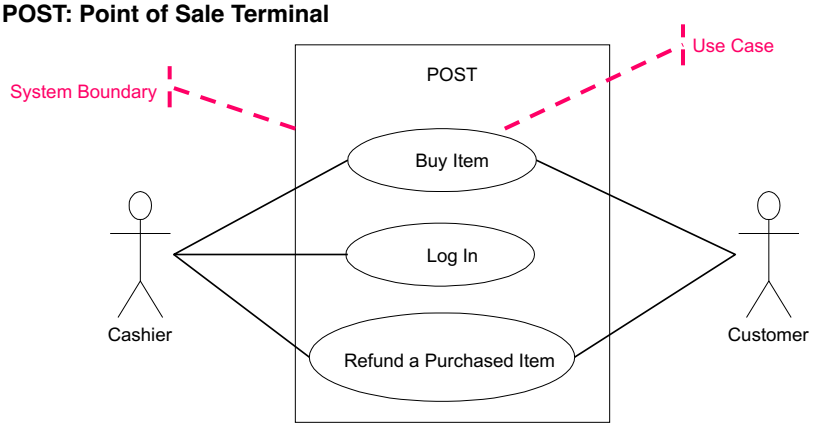
- Computerized system to record sales and handle payments
- Typically used in a retail store
- Includes HW components, such as computer and bar code scanner
- Software to run the system
- Goals of system:
  - Increase checkout automation
  - fast and accurate sales analysis
  - automatic inventory control

6

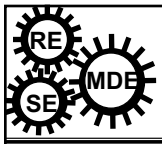


Use-Case Diagrams (POST)

POST: Point of Sale Terminal

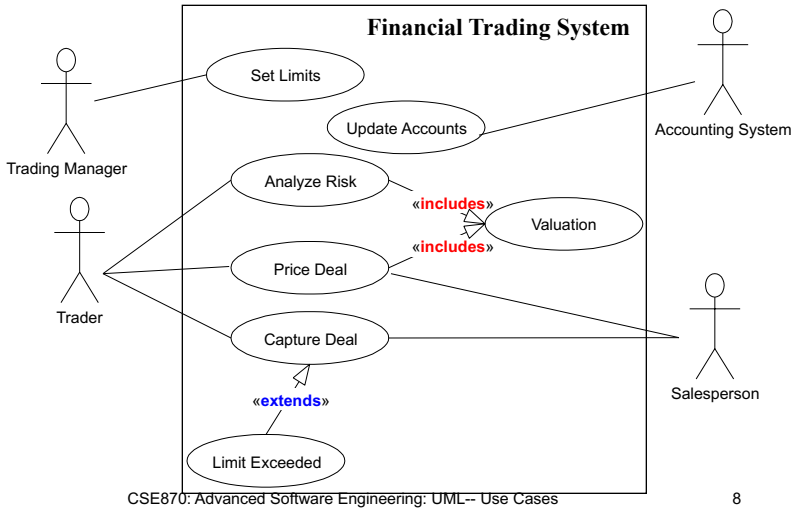


7

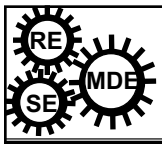


Another Example

Financial Trading System

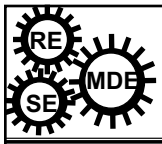


8



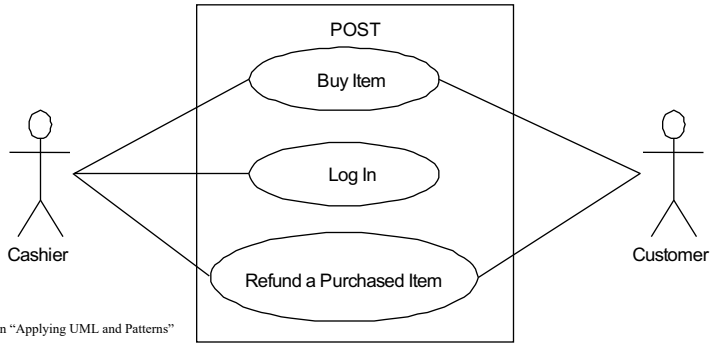
Includes and Extends

- Includes/Uses
  - You have a piece of behavior that is similar across many use cases
  - Break this out as a separate use-case and let the other ones "include" it
  - Examples include
    - Valuation
    - Validate user interaction
    - Sanity check on sensor inputs
    - Check for proper authorization
- Extends
  - A use-case is similar to another one but does a little bit more
  - Put the normal behavior in one use-case and the exceptional behavior somewhere else
    - Capture the normal behavior
    - Try to figure out what can go wrong in each step
    - Capture the exceptional cases in separate use-cases
  - Makes it much easier to understand

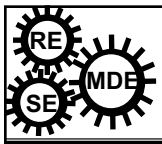


Setting the System Boundary

- The system boundary will affect your actors and use-cases

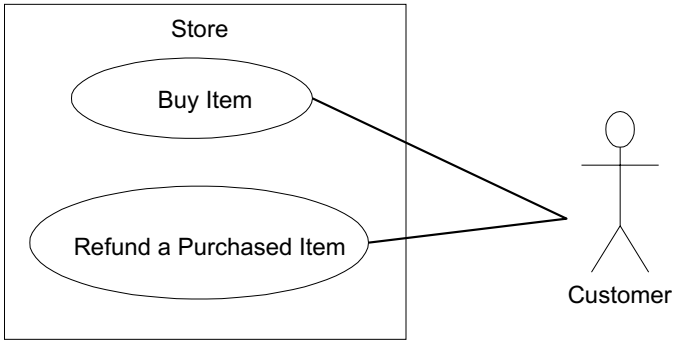


Adapted from Larman "Applying UML and Patterns"



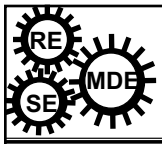
A Different Boundary

- Let us view the whole store as our system

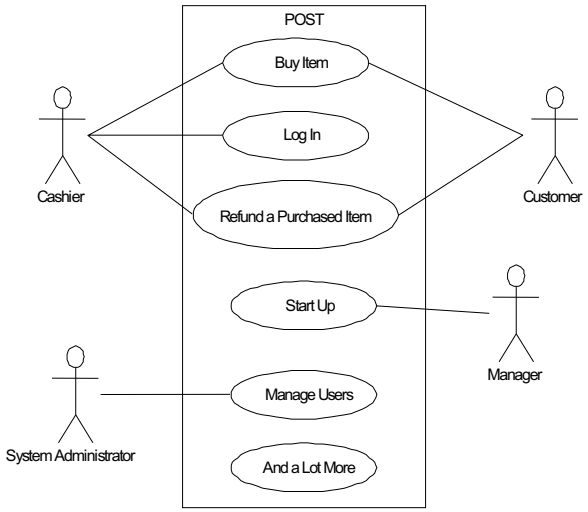


Adapted from Larman "Applying UML and Patterns"

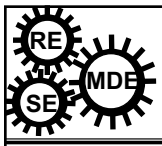
13



Partial POST



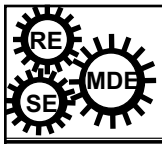
14



POST Use-Case

Use case: **Buy Item**
Actors: Customer (initiator), Cashier
Type: Primary
Description: The Customer arrives at the checkout with items to purchase. The Cashier records the purchase items and collects a payment. On completion the Customer leaves with the items

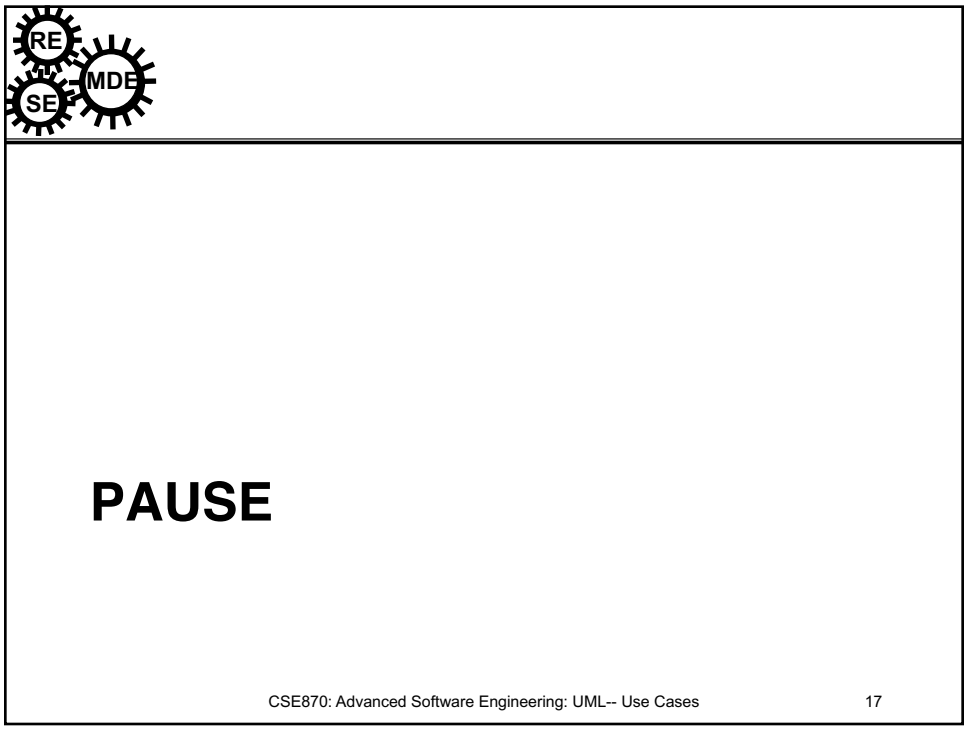
15



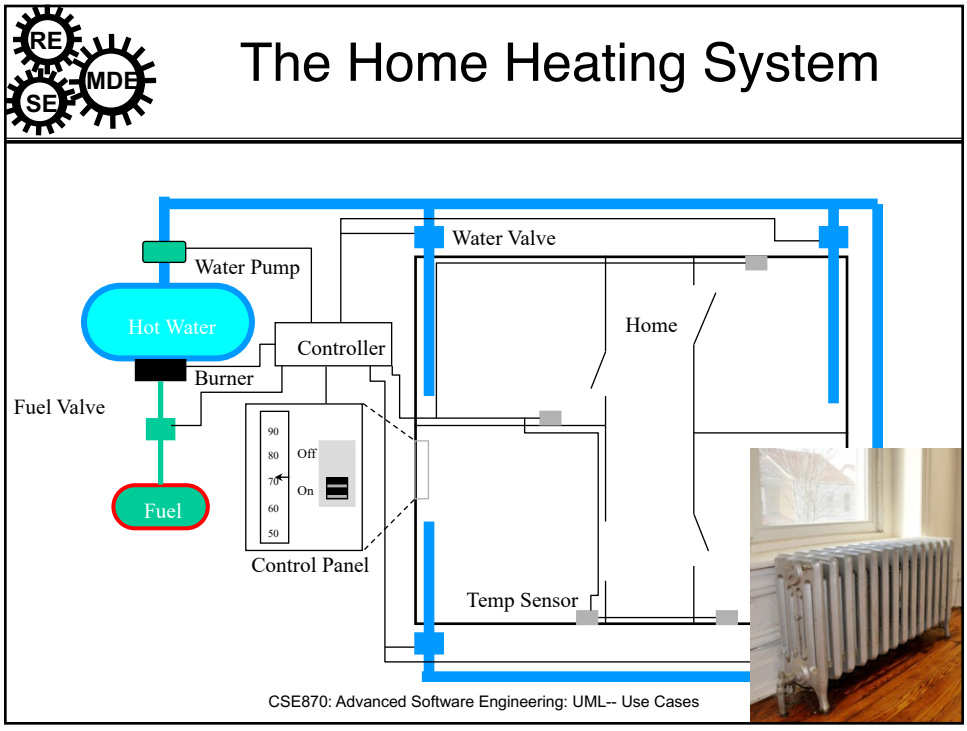
POST Expanded Use-Case

Use case: **Buy Item**
Actors: Customer (initiator), Cashier
Type: Primary and essential
Description: The Customer arrives at the checkout with items to purchase. The Cashier records the purchase items and collects a payment. On completion the Customer leaves with the items.
Cross Ref.: Requirements XX, YY, and ZZ
Use-Cases: Cashier must have completed the *Log In* use-case

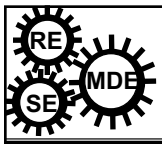
16


RE
MDE
SE
PAUSE
CSE870: Advanced Software Engineering: UML-- Use Cases
17

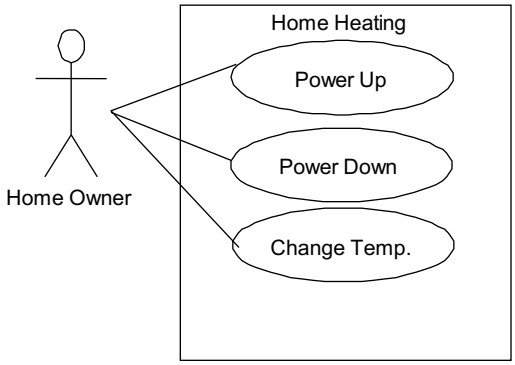
17


RE
MDE
SE
The Home Heating System
Water Valve
Water Pump
Hot Water
Controller
Home
Burner
Fuel Valve
90
80
70
60
50
Off
On
Fuel
Control Panel
Temp Sensor
CSE870: Advanced Software Engineering: UML-- Use Cases

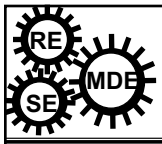
18



Home Heating Use-Case Diagram



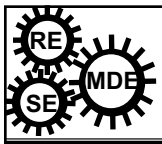
19



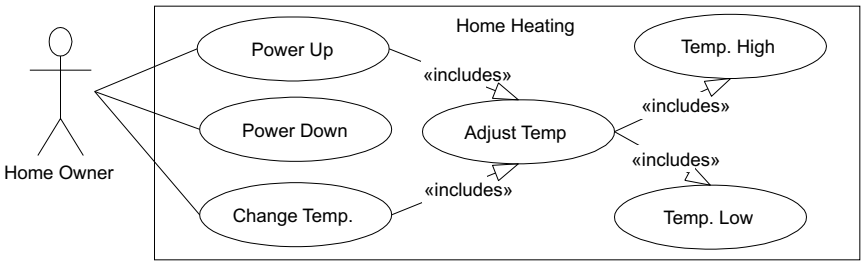
Home Heating Use-Cases

Use case: **Power Up**
Actors: Home Owner (initiator)
Type: Primary and essential
Description: The Home Owner turns the power on. Each room is temperature checked. If a room is below the the desired temperature the valve for the room is opened, the water pump started, the fuel valve opened, and the burner ignited.
If the temperature in all rooms is above the desired temperature, no actions are taken.
Cross Ref.: Requirements XX, YY, and ZZ
Use-Cases: None

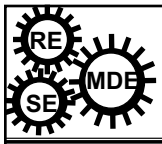
20



Modified Home Heating



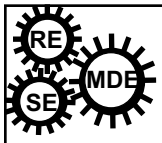
21



HACS

- Homework assignment and collection are an integral part of any educational system. Today, this task is performed manually. What we want the homework assignment distribution and collection system (HACS for short) to do is to automate this process.
- HACS will be used by the instructor to distribute the homework assignments, review the students' solutions, distribute suggested solution, and distribute student grades on each assignment.
- HACS shall also help the students by automatically distributing the assignments to the students, provide a facility where the students can submit their solutions, remind the students when an assignment is almost due, remind the students when an assignment is overdue.

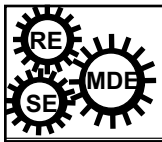
22



If Time Permits

In-Class activity

23

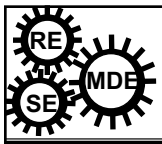


In-class exercise

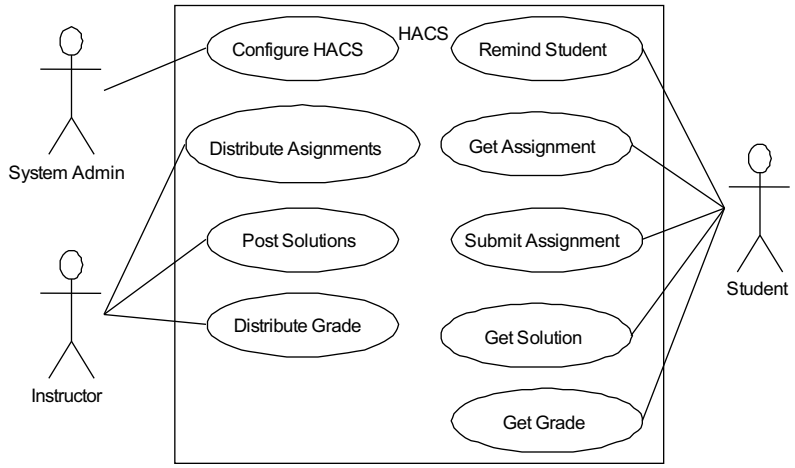
Work in Pairs

- Create use case diagram
- Have an example of <<includes>> relationship
- Bonus: have an example of <<extends>> relationship
- Remember to include key elements of UC diagram.

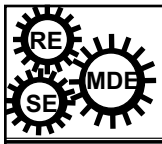
24



HACS Use-Case Diagram



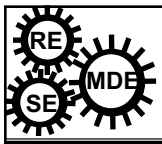
25



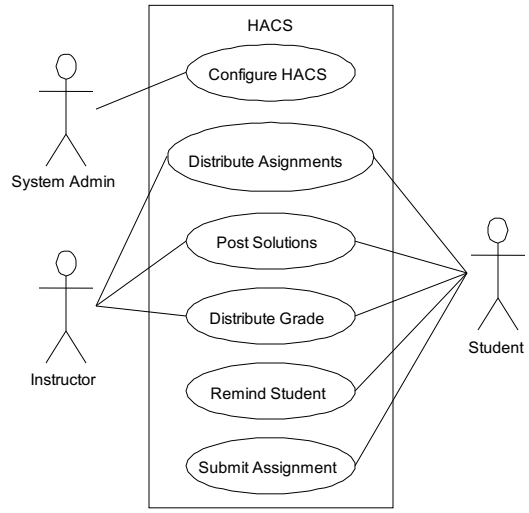
HACS Use-Cases

Use case:	**Distribute Assignments**
Actors:	Instructor (initiator)
Type:	Primary and essential
Description:	The Instructor completes an assignment and submits it to the system. The instructor will also submit the due date and the class the assignment is assigned for.
Cross Ref.:	Requirements XX, YY, and ZZ
Use-Cases:	*Configure HACS* must be done before any user (Instructor or Student) can use HACS

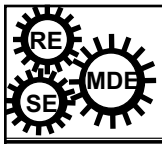
26



Alternate HACS



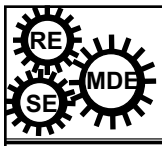
27



Alternate HACS Use-Cases

Use case: **Distribute Assignments**
Actors: Instructor (initiator), Student
Type: Primary and essential
Description: The Instructor completes an assignment and submits it to the system. The instructor will also submit the delivery date, due date, and the class the assignment is assigned for. The system will at the delivery date mail the assignment to the student.
Cross Ref.: Requirements XX, YY, and ZZ
Use-Cases: *Configure HACS* must be done before any user (Instructor or Student) can use HACS

28



When to use Use-Cases

- In short, always!!!
- Requirements is the toughest part of software development
  - Use-Cases is a powerful tool to understand
    - Who your users are (including interacting systems)
    - What functions the system shall provide
    - How these functions work at a high level
- Spend adequate time on requirements and in the elaboration phase