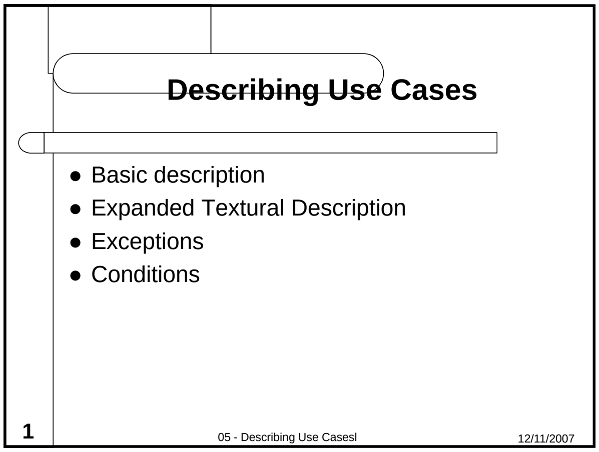
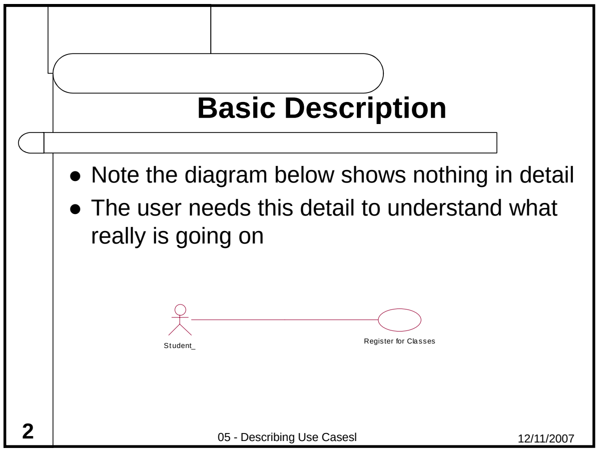


Describing Use Cases

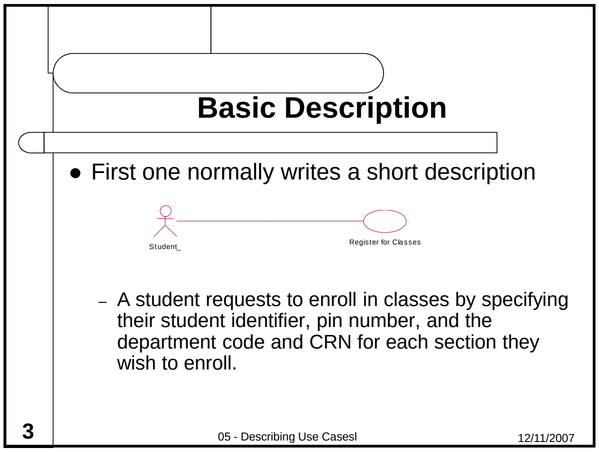
- Basic description
- Expanded Textural Description
- Exceptions
- Conditions



Basic Description

- Note the diagram below shows nothing in detail
- The user needs this detail to understand what really is going on

Student_
Register for Classes

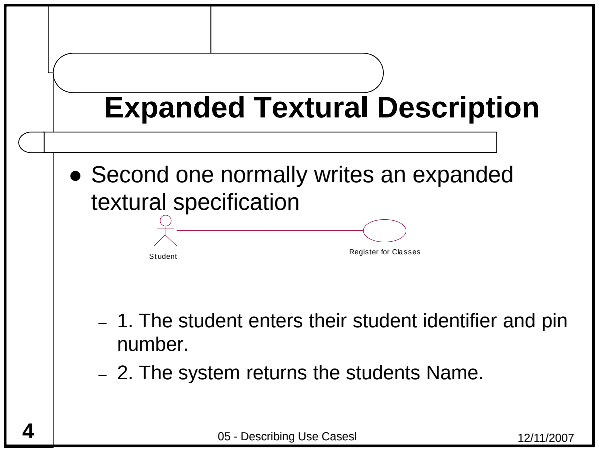


Basic Description

- First one normally writes a short description

Student_
Register for Classes

  - A student requests to enroll in classes by specifying their student identifier, pin number, and the department code and CRN for each section they wish to enroll.

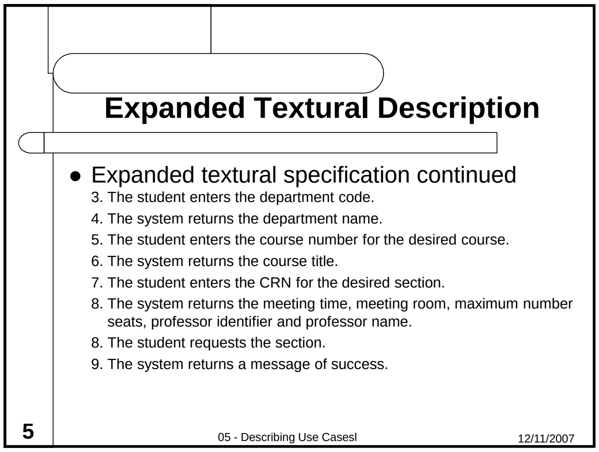


Expanded Textural Description

- Second one normally writes an expanded textural specification

Student_
Register for Classes

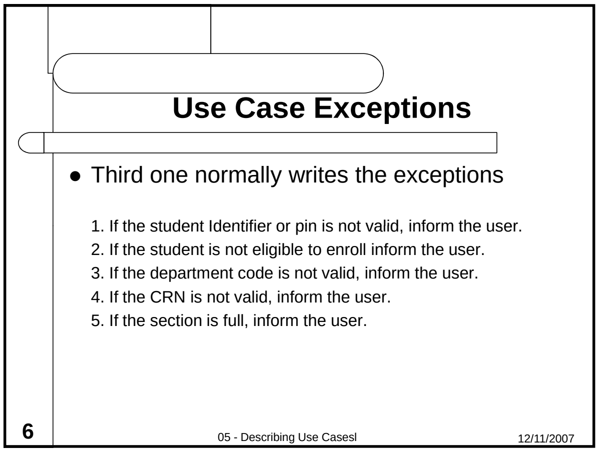
  - 1. The student enters their student identifier and pin number.
  - 2. The system returns the students Name.



Expanded Textural Description

- Expanded textural specification continued

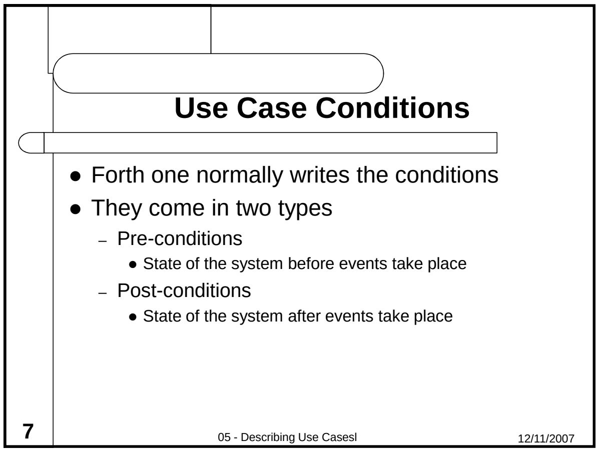
3. The student enters the department code.
4. The system returns the department name.
5. The student enters the course number for the desired course.
6. The system returns the course title.
7. The student enters the CRN for the desired section.
8. The system returns the meeting time, meeting room, maximum number seats, professor identifier and professor name.
8. The student requests the section.
9. The system returns a message of success.



Use Case Exceptions

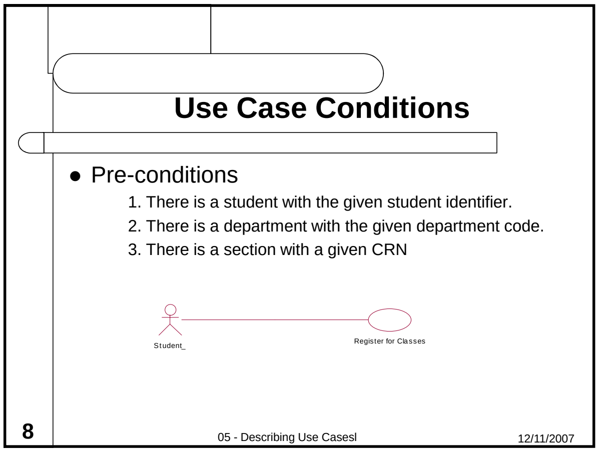
- Third one normally writes the exceptions

1. If the student Identifier or pin is not valid, inform the user.
2. If the student is not eligible to enroll inform the user.
3. If the department code is not valid, inform the user.
4. If the CRN is not valid, inform the user.
5. If the section is full, inform the user.



Use Case Conditions

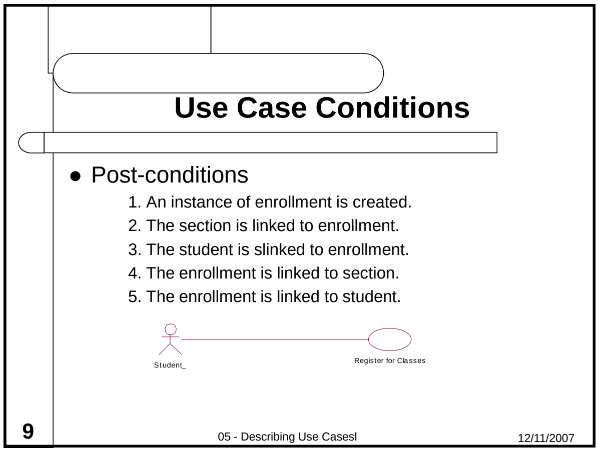
- Forth one normally writes the conditions
- They come in two types
  - Pre-conditions
    - State of the system before events take place
  - Post-conditions
    - State of the system after events take place



Use Case Conditions

- Pre-conditions
  1. There is a student with the given student identifier.
  2. There is a department with the given department code.
  3. There is a section with a given CRN

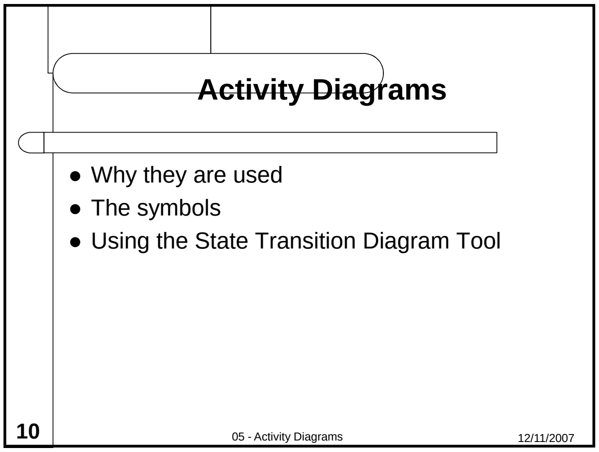
Student_
Register for Classes



Use Case Conditions

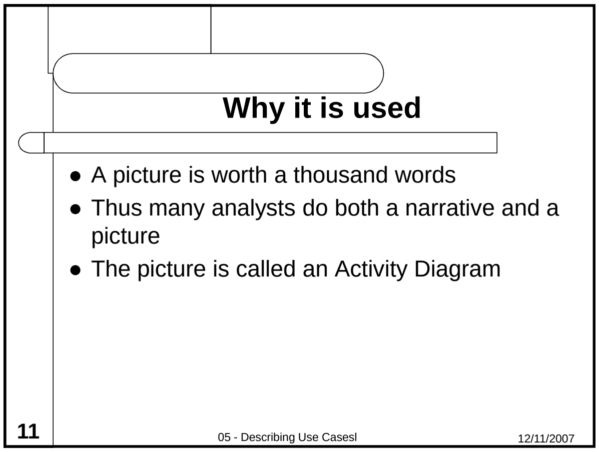
- Post-conditions
  1. An instance of enrollment is created.
  2. The section is linked to enrollment.
  3. The student is slinked to enrollment.
  4. The enrollment is linked to section.
  5. The enrollment is linked to student.

Student_
Register for Classes



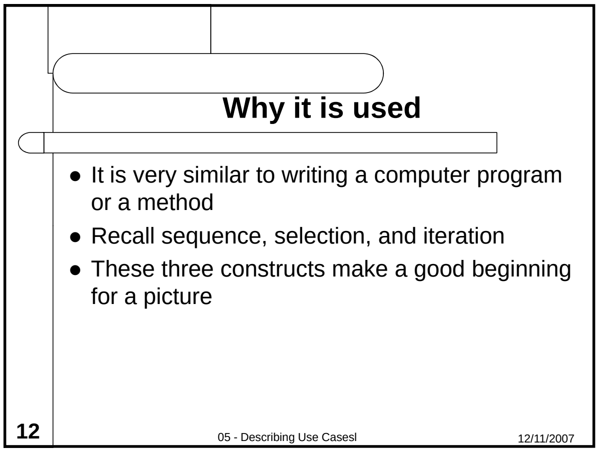
Activity Diagrams

- Why they are used
- The symbols
- Using the State Transition Diagram Tool



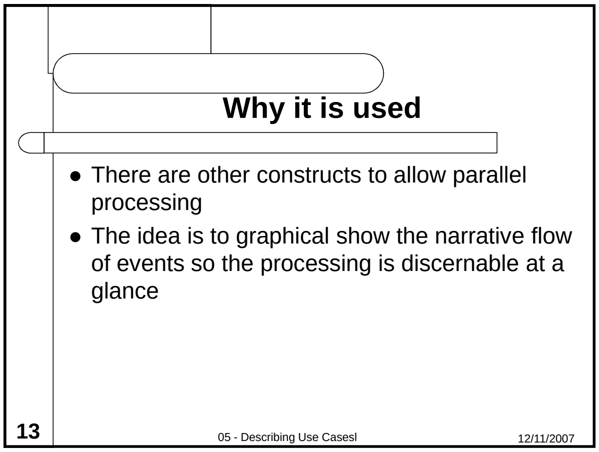
Why it is used

- A picture is worth a thousand words
- Thus many analysts do both a narrative and a picture
- The picture is called an Activity Diagram



Why it is used

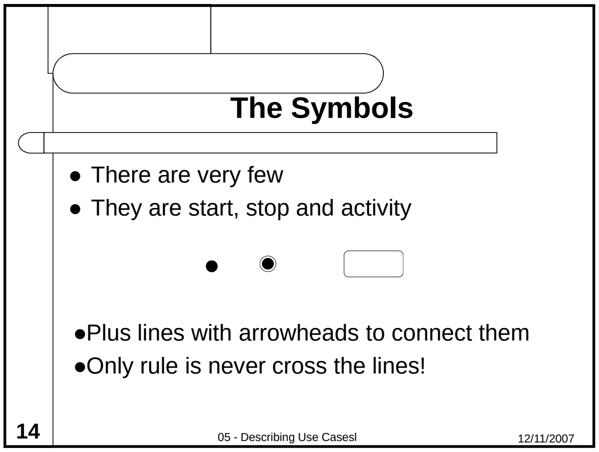
- It is very similar to writing a computer program or a method
- Recall sequence, selection, and iteration
- These three constructs make a good beginning for a picture



Why it is used

- There are other constructs to allow parallel processing
- The idea is to graphical show the narrative flow of events so the processing is discernable at a glance

13 05 - Describing Use Casesl 12/11/2007

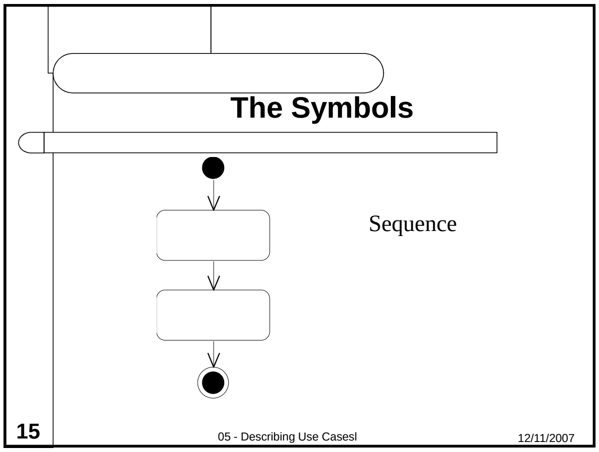


The Symbols

- There are very few
- They are start, stop and activity

●Plus lines with arrowheads to connect them
●Only rule is never cross the lines!

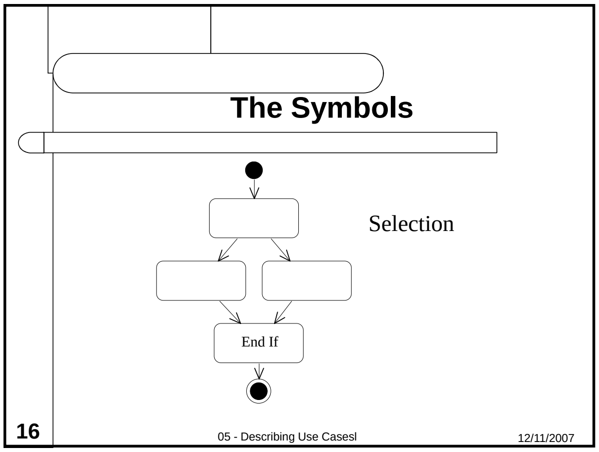
14 05 - Describing Use Casesl 12/11/2007



The Symbols

Sequence

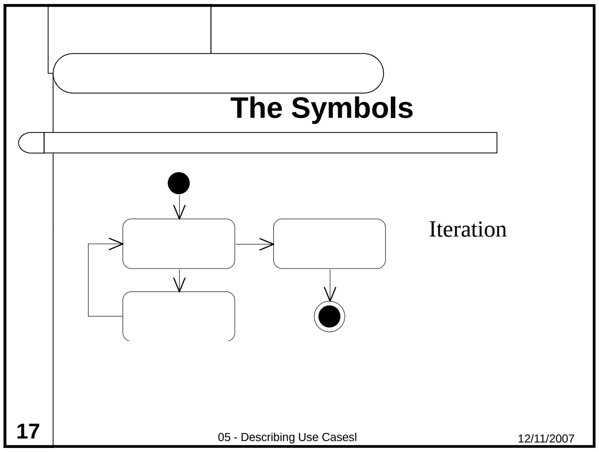
15 05 - Describing Use Casesl 12/11/2007



The Symbols

Selection

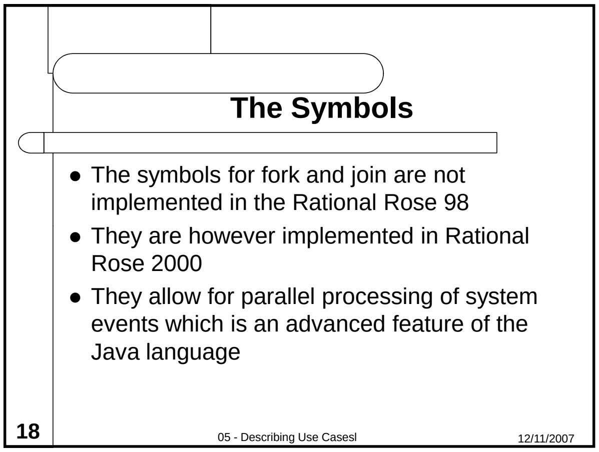
16 05 - Describing Use Casesl 12/11/2007



The Symbols

Iteration

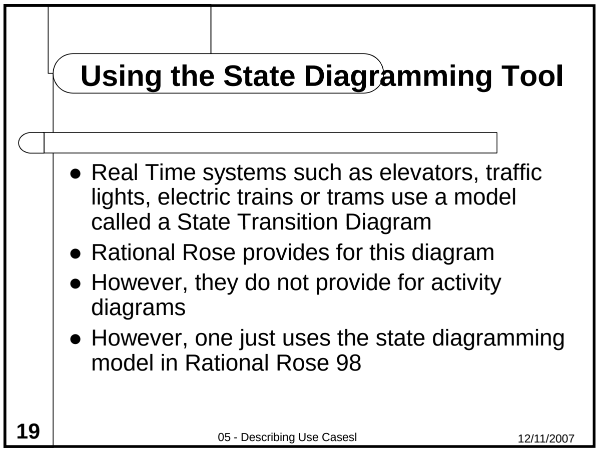
17 05 - Describing Use Casesl 12/11/2007



The Symbols

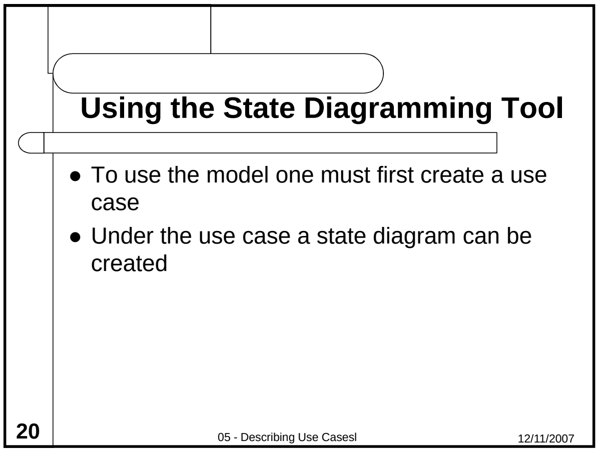
- The symbols for fork and join are not implemented in the Rational Rose 98
- They are however implemented in Rational Rose 2000
- They allow for parallel processing of system events which is an advanced feature of the Java language

18 05 - Describing Use Casesl 12/11/2007



Using the State Diagramming Tool

- Real Time systems such as elevators, traffic lights, electric trains or trams use a model called a State Transition Diagram
- Rational Rose provides for this diagram
- However, they do not provide for activity diagrams
- However, one just uses the state diagramming model in Rational Rose 98



Using the State Diagramming Tool

- To use the model one must first create a use case
- Under the use case a state diagram can be created