

# **Section 1.3**

## **UML Notation**

1. UML overview
2. Use case diagrams
3. Class diagrams
4. State machine diagrams
5. Activity diagrams
6. Sequence diagrams
7. Packages


# 1.3.1 UML Overview

- Unified Modelling Language
  - what is it?
    - a tool for expressing system models
      - functional model, dynamic model, object model
  - what is it used for?
    - to facilitate communication between:
      - client and development team
      - members of development team
  - characteristics of good system models
    - clear
    - unambiguous
    - complete

# The UML Family

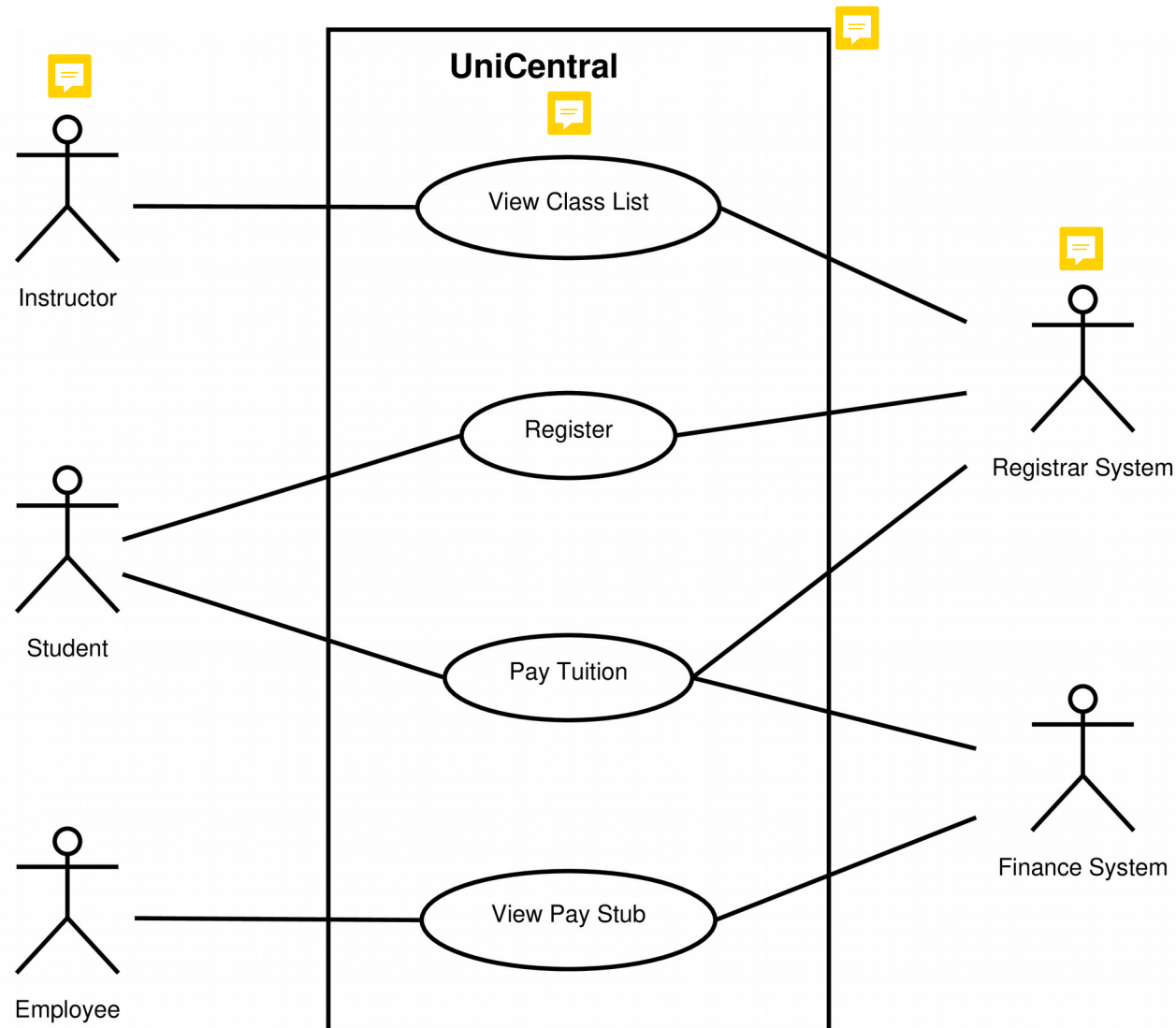
- UML is a family of notations
  - each notation is used to represent a specific model
- Models and notations
  - functional model
    - use case diagrams
  - dynamic model
    - state machine diagrams
    - sequence diagrams
    - activity diagrams
  - object model
    - class diagrams

## 1.3.2 Use Case Diagrams

- What is a use case? 
  - describes system behaviour, as observed by external entities
  - external entities are called *actors*
    - end users (people); different actor “roles”
    - external systems (existing systems that our system will interact with)
  - use cases can also be represented textually (table-based)
- What are use case diagrams?
  - graphical representation of use cases
- Purpose
  - to establish system boundaries

# Example of Detailed Use Case Diagram

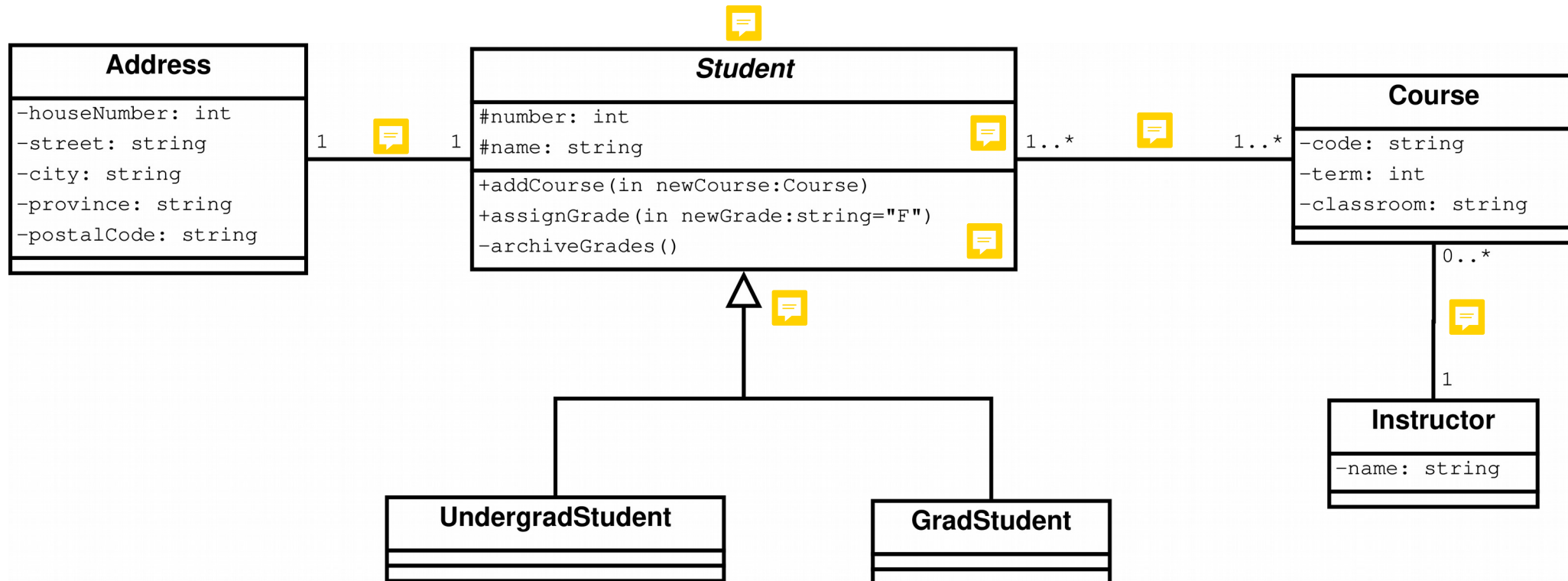
- Simplistic example (doesn't show high-level use cases)



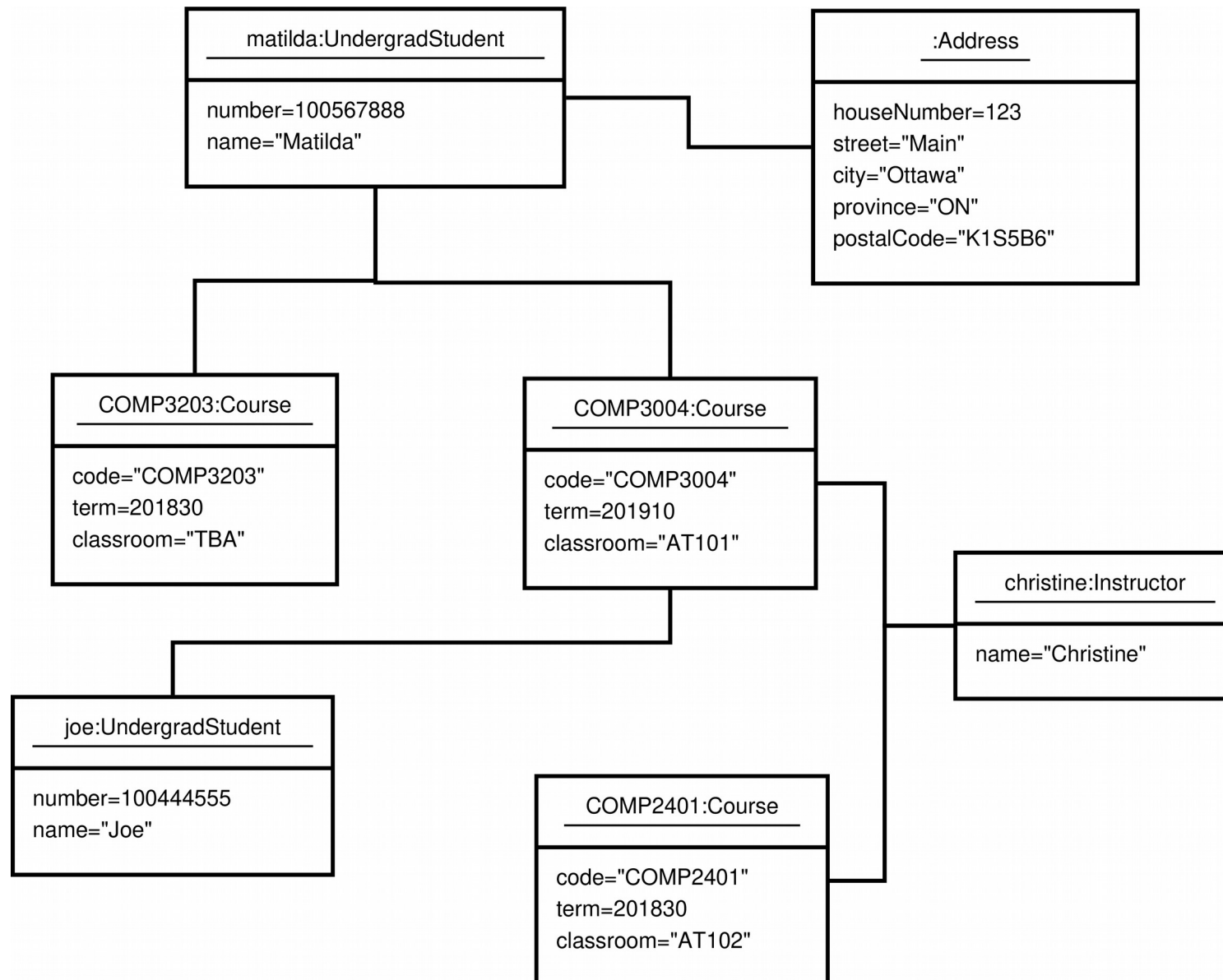
## 1.3.3 Class Diagrams

- What are class diagrams?
  - a graphical representation of classes and objects
    - instance names are underlined
- Purpose
  - to describe the system in terms of classes
    - includes attributes, operations, associations

# Example of Class Diagram



# Example of Object Diagram



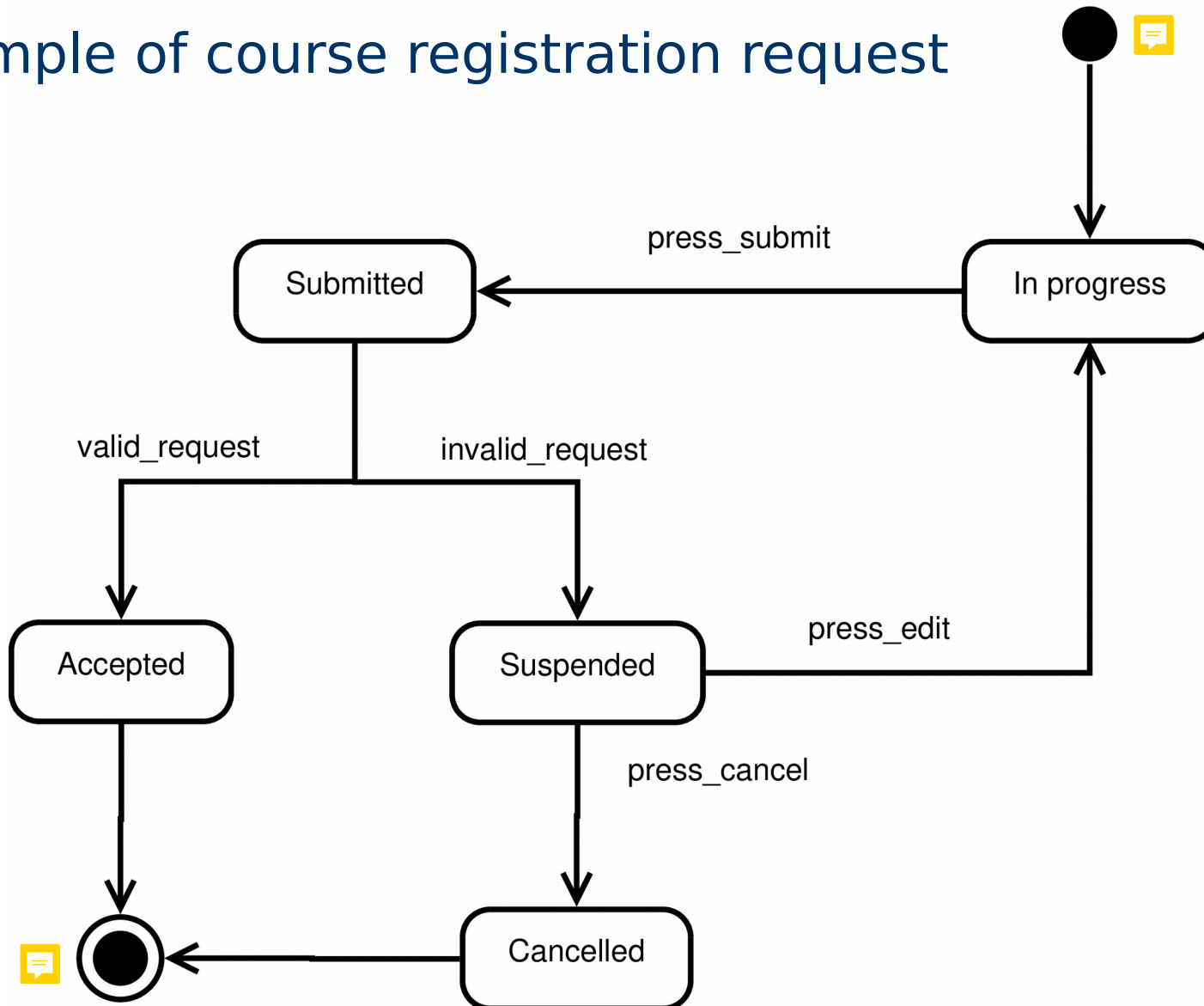


## 1.3.4 State Machine Diagrams

- What are state machine diagrams?
  - graphical representation of object state behaviour
- Purpose
  - to describe object behaviour as a set of states and transitions
    - state: particular set of attribute values for the object
    - transition: condition(s) under which the object changes state
  - every state machine describes behaviour of **one** specific object

# Example of State Machine Diagram

- Example of course registration request

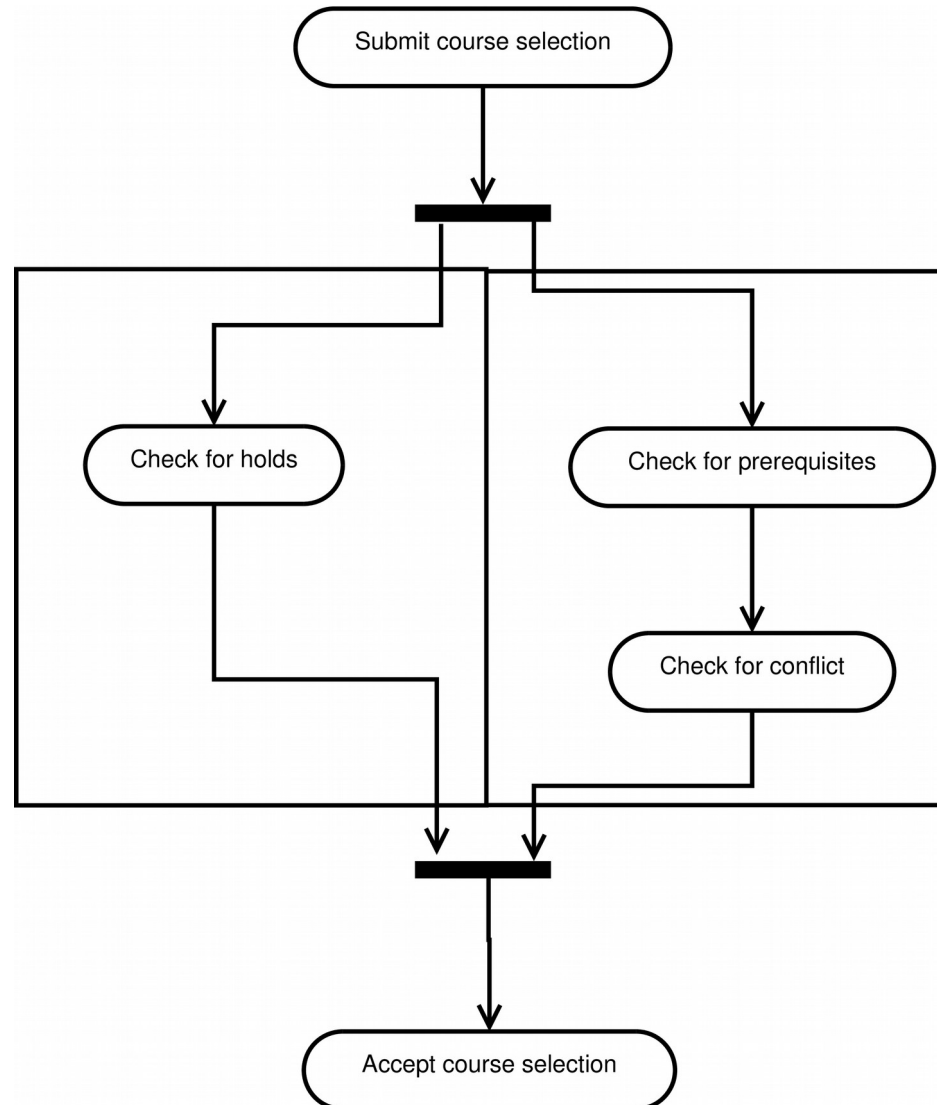


## 1.3.5 Activity Diagrams


- What are activity diagrams?
  - graphical representation of system behaviour
    - sequencing
    - coordination
- Purpose
  - to describe sequential steps in system processing
    - control flow
    - concurrency

# Example of Activity Diagram

- Example of course registration validation

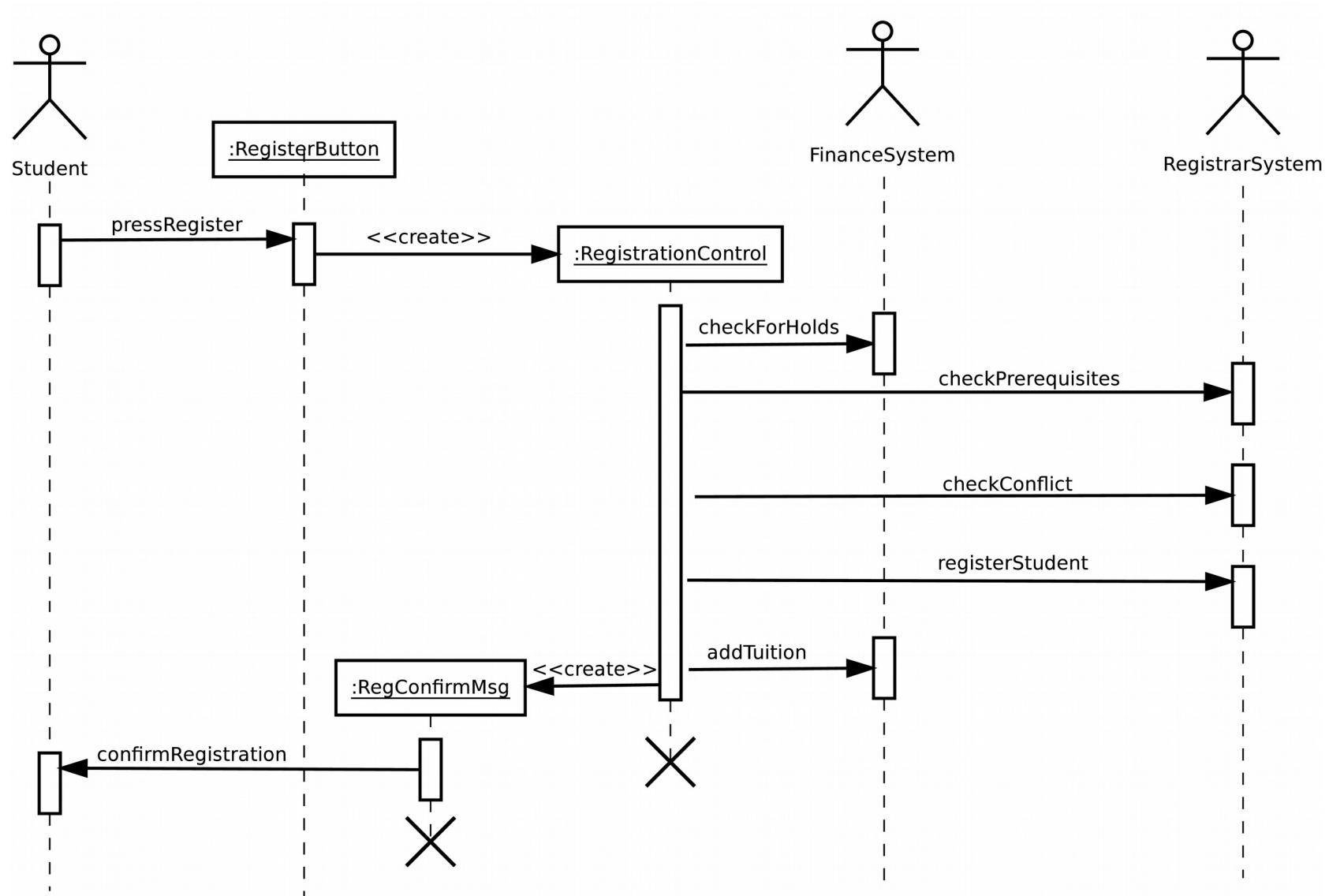


## 1.3.6 Sequence Diagrams

- What are sequence diagrams?
  - graphical representation of messages between objects 
    - actors
    - internal objects
- Purpose
  - to capture system behaviour, from the *user's point of view*
  - to show how use case behaviour is distributed across objects
  - every sequence diagram describes behaviour of **one** use case
  - **note:** we are still very far from the code here

# Example of Sequence Diagram

- Example of course registration



# 1.3.7 Packages

- What are UML packages?
  - groupings of related UML diagrams
    - use case diagrams
    - class diagrams
    - sequence diagrams
    - state machine diagrams
- Purpose
  - to organize the diagrams
  - to reduce the complexity of diagrams

# Example of Use Case Packages

