LECTURE – 9

Phase – 2: System Design

More on UML Diagrams


Project Manager
How is it going, Tony?
Software Designer
I have this headache that comes and goes
Diagram-averse Client
Steve walks in
Hi, Tony
There it is again

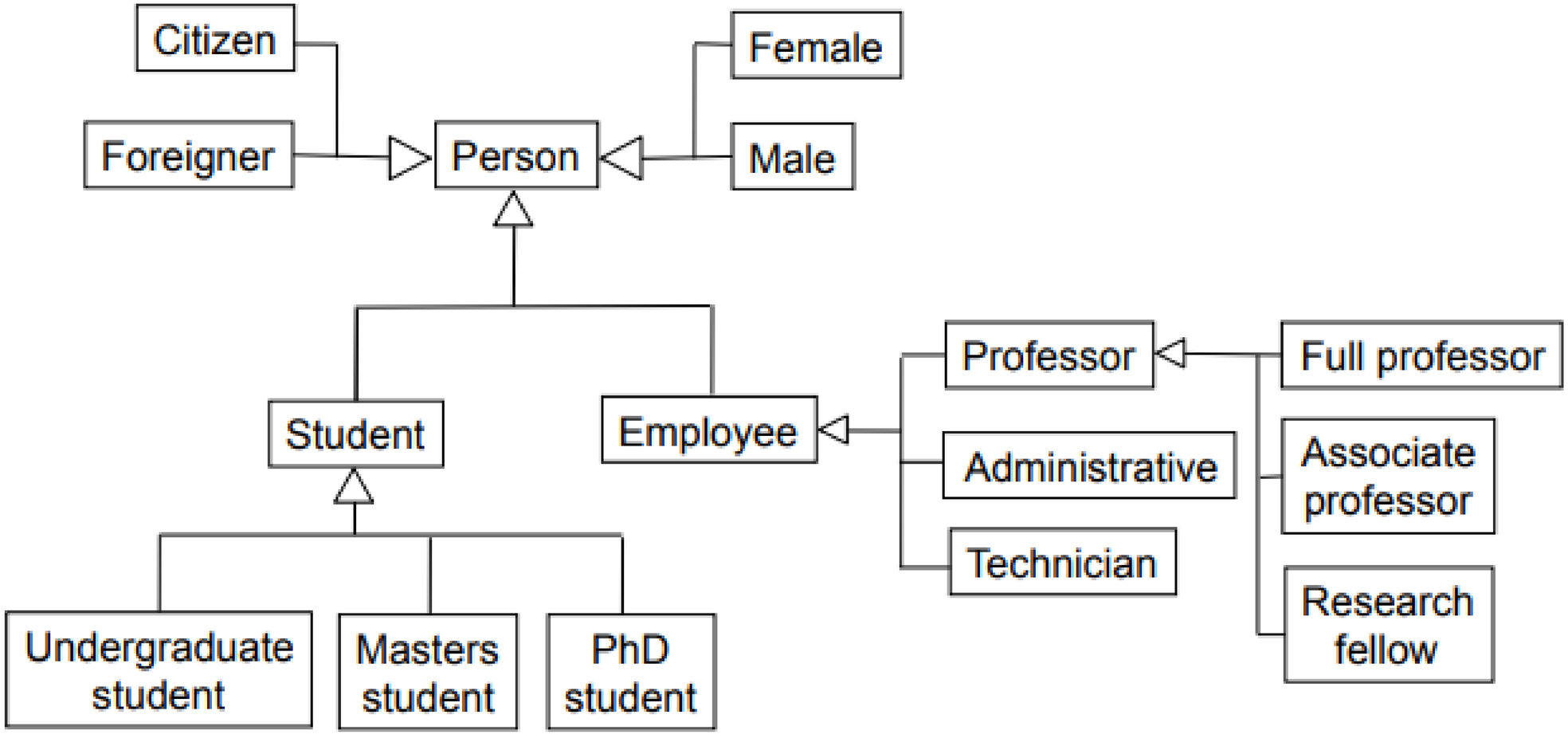
System Design

- Class Diagram
- State charts
- Activity Diagram
- Sequence Diagram

Class Diagram

Class Diagram

- Structural Diagram
  - A university is an organization where some persons work, some other study
  - There are several types of roles and grouping entities
    - Students
    - ….


Citizen
Foreigner
Person
Female
Male
Student
Employee
Undergraduate student
Masters student
PhD student
Professor
Administrative
Technician
Full professor
Associate professor
Research fellow

Class

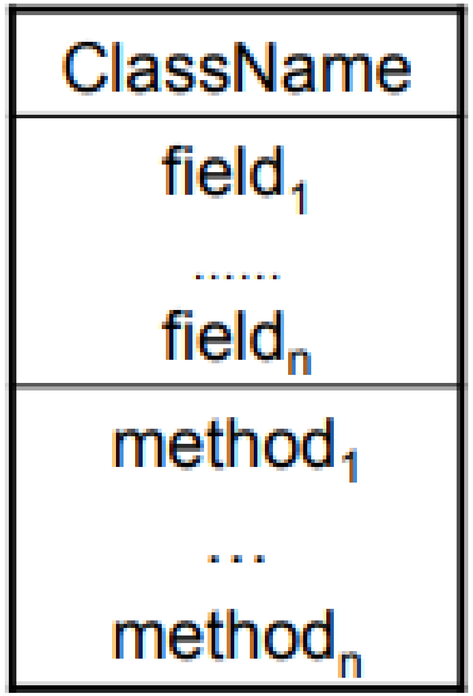
- Defines the structure of the system
  - Attributes and the behaviors

- Defines a template for creating instances
  - Names and types of all fields
  - Names, signatures, and implementations of all methods

Class Diagrams

- Nodes representing classes (types of objects)
- Links representing relationships among classes
  - Association
  - Aggregation or composition
  - Dependency
  - Inheritance
- Links can have multiplicities and/or names for roles played by participants

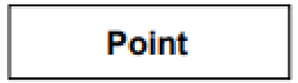
Example

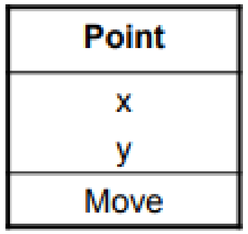


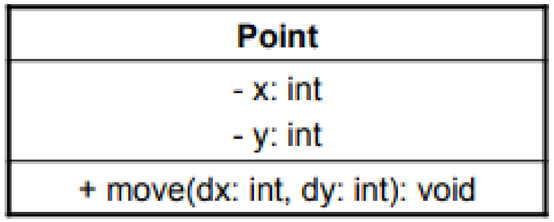
The top compartment shows the class **name**

The middle compartment contains the declarations of the fields, or *attributes*, of the class

The bottom compartment contains the declarations of the *methods* of the class







Class can be defined at different abstraction levels

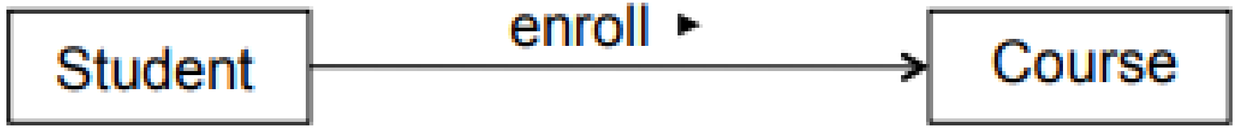
- private

+public

Association

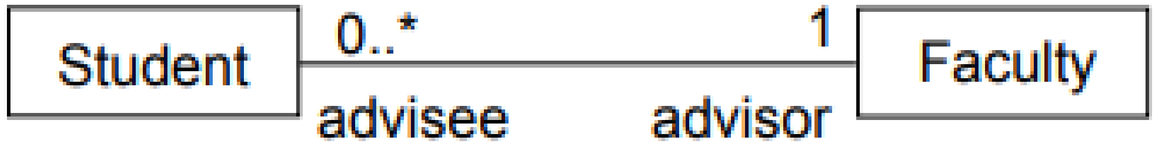
- An association is a structural relationship that connects two classes

- Represented as a line between boxes



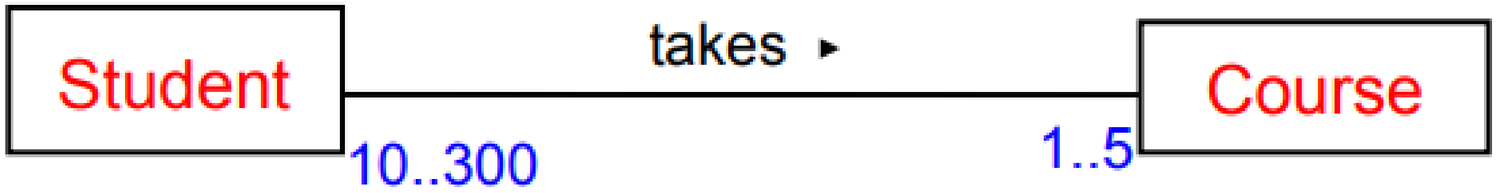
Roles and multiplicity

- An association line may have a role name and a multiplicity specification
- The multiplicity specifies an integer interval,



Deriving Association from Requirements

- Requirements
  - A Student can take up to five Courses
  - Every Student must be enrolled in at least one course
  - Up to 300 students can enroll in a course
  - A class should have at least 10 students

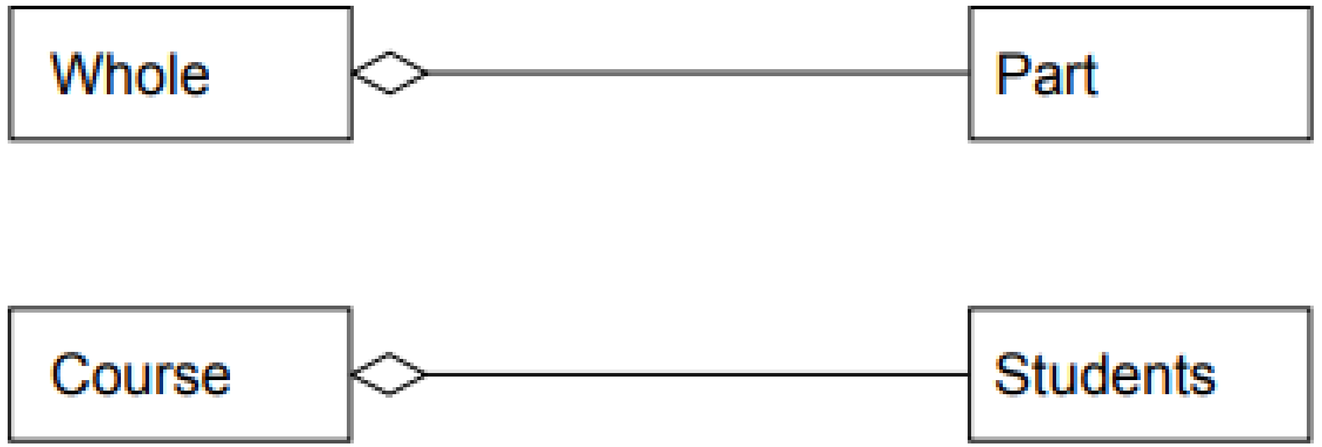


Example

- A teacher teaches 1 to 3 courses
- Each course is taught by only one teacher
- A student can take between 1 to 6 courses
- A course can have 20 to 300 students

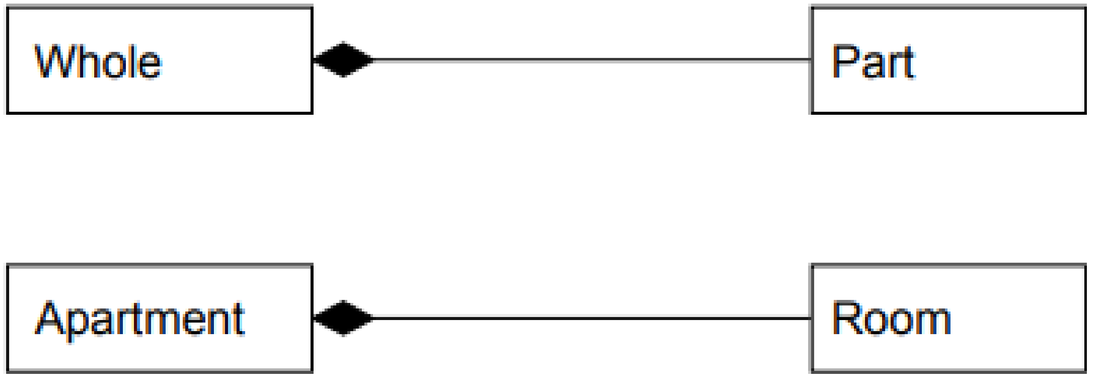
Aggregation

- An aggregation is a special form of association representing has-a or part-whole relationship
- It distinguishes the whole (aggregate class) from its parts (component class)

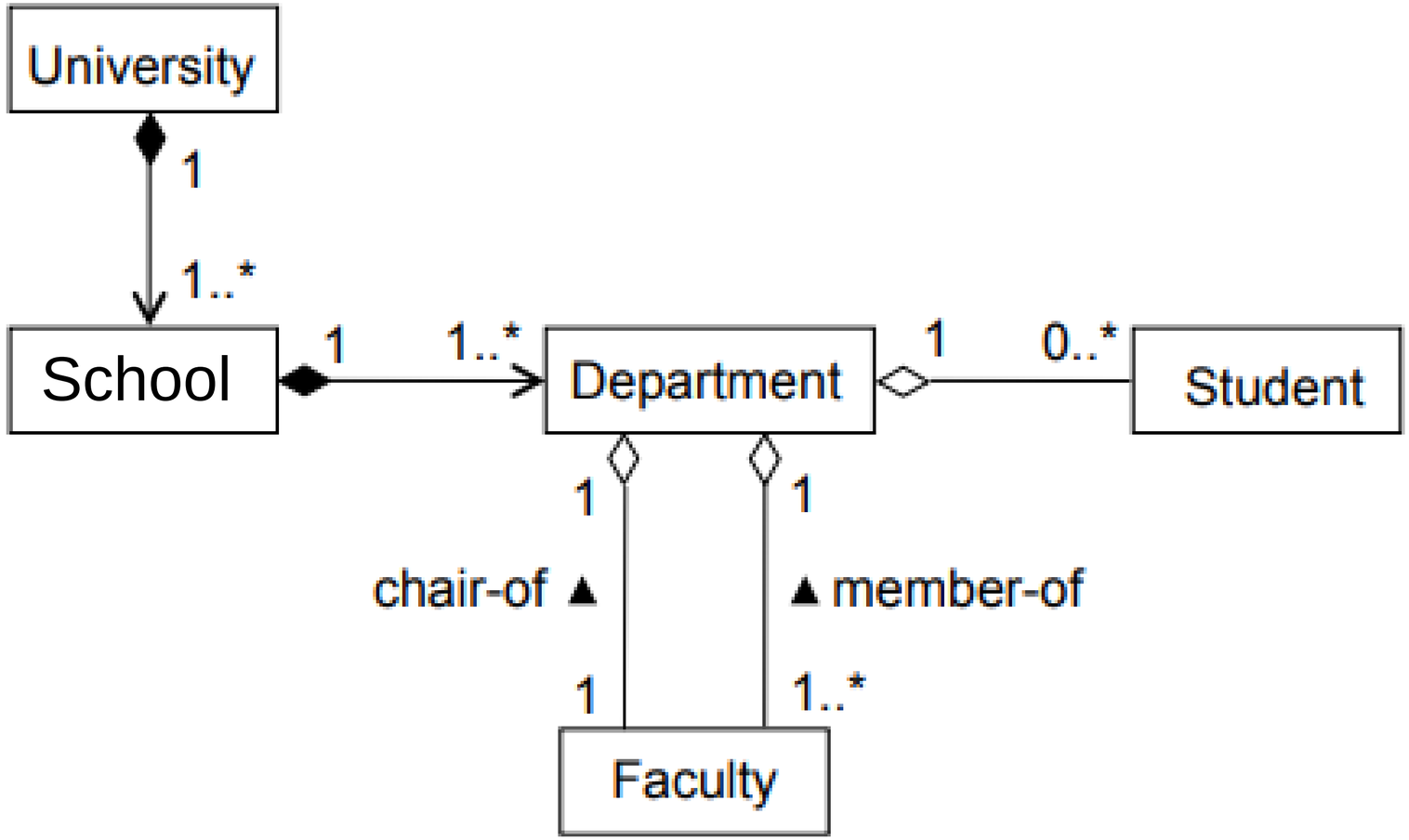


Composition

- A composition is a stronger form of aggregation
- It implies exclusive ownership of the component class by the aggregate class
- The lifetime of the parts is entirely included in the lifetime of the whole (a part can not exist without its whole)

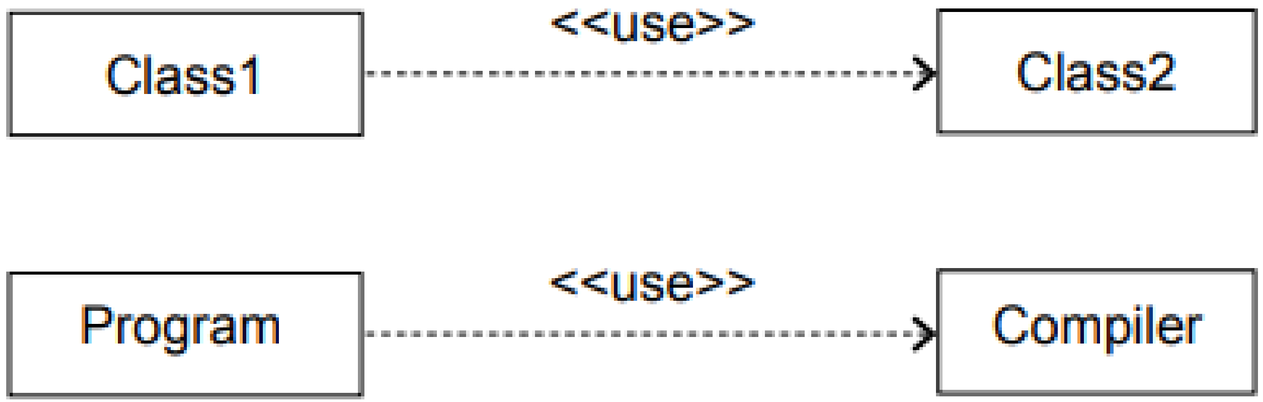


Example


University
1
1..*
School
1
1..*
Department
1
0..*
Student
1
1
chair-of
member-of
1
1..*
Faculty

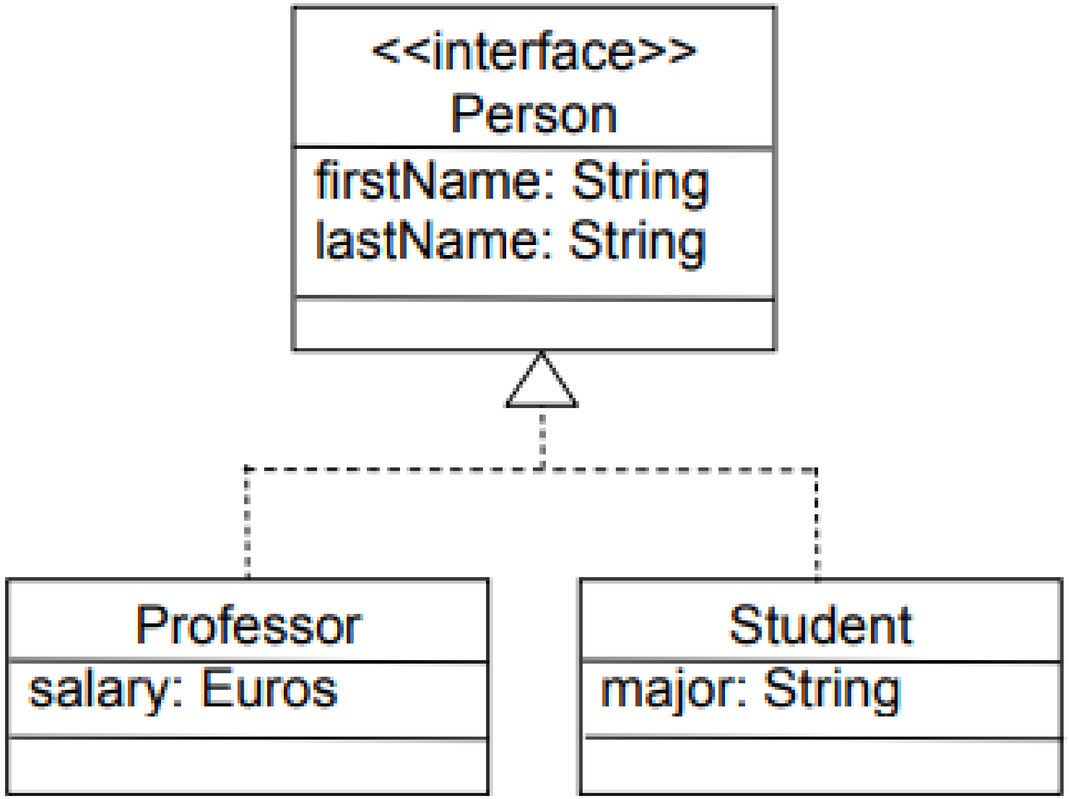
Dependency

- A dependency of A from B is a relationship between two entities A and B such that the proper operation of A depends on the presence of B, and changes in latter would affect the former
- Example: a common form of dependency is the use relation among classes

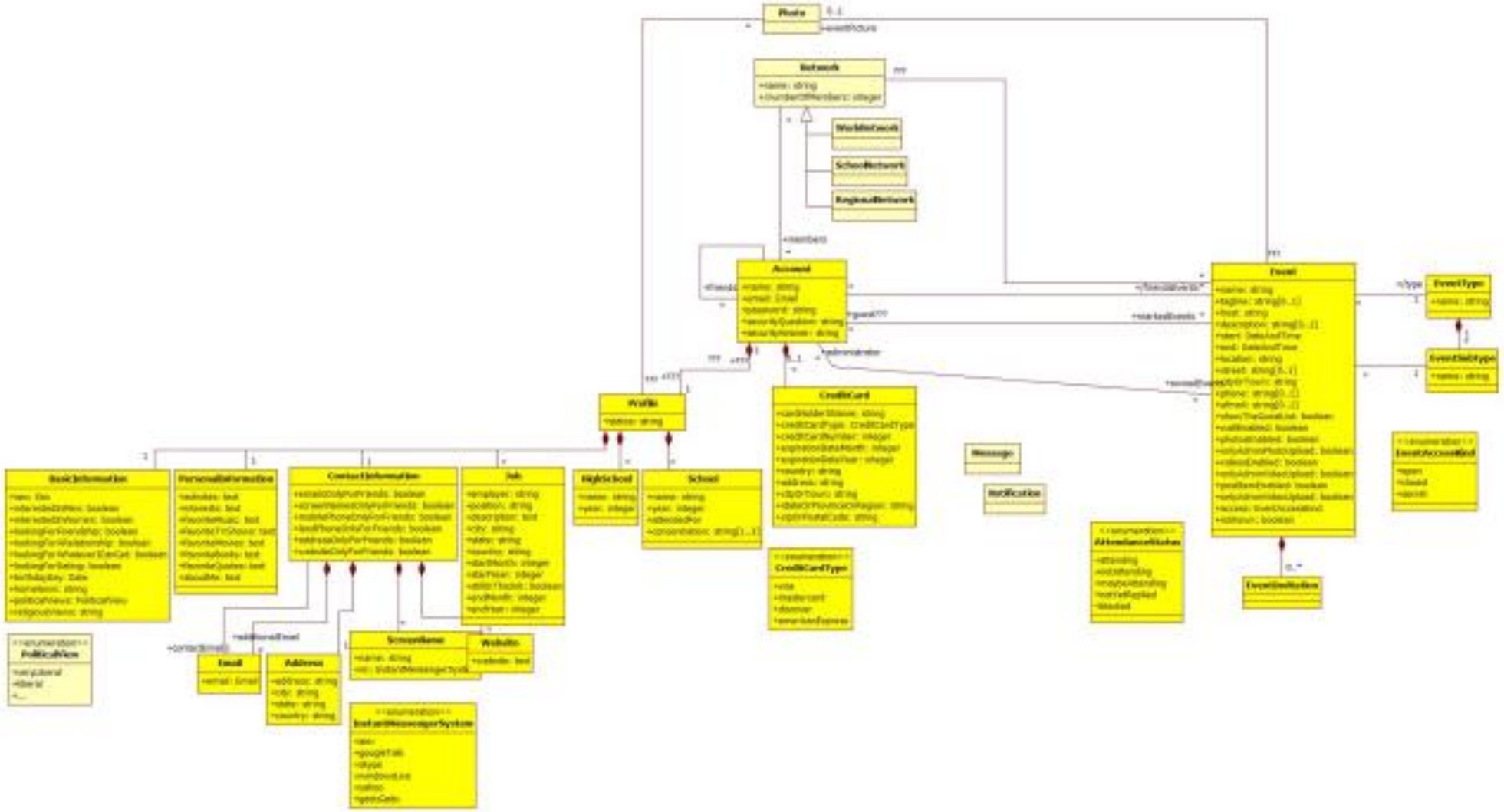


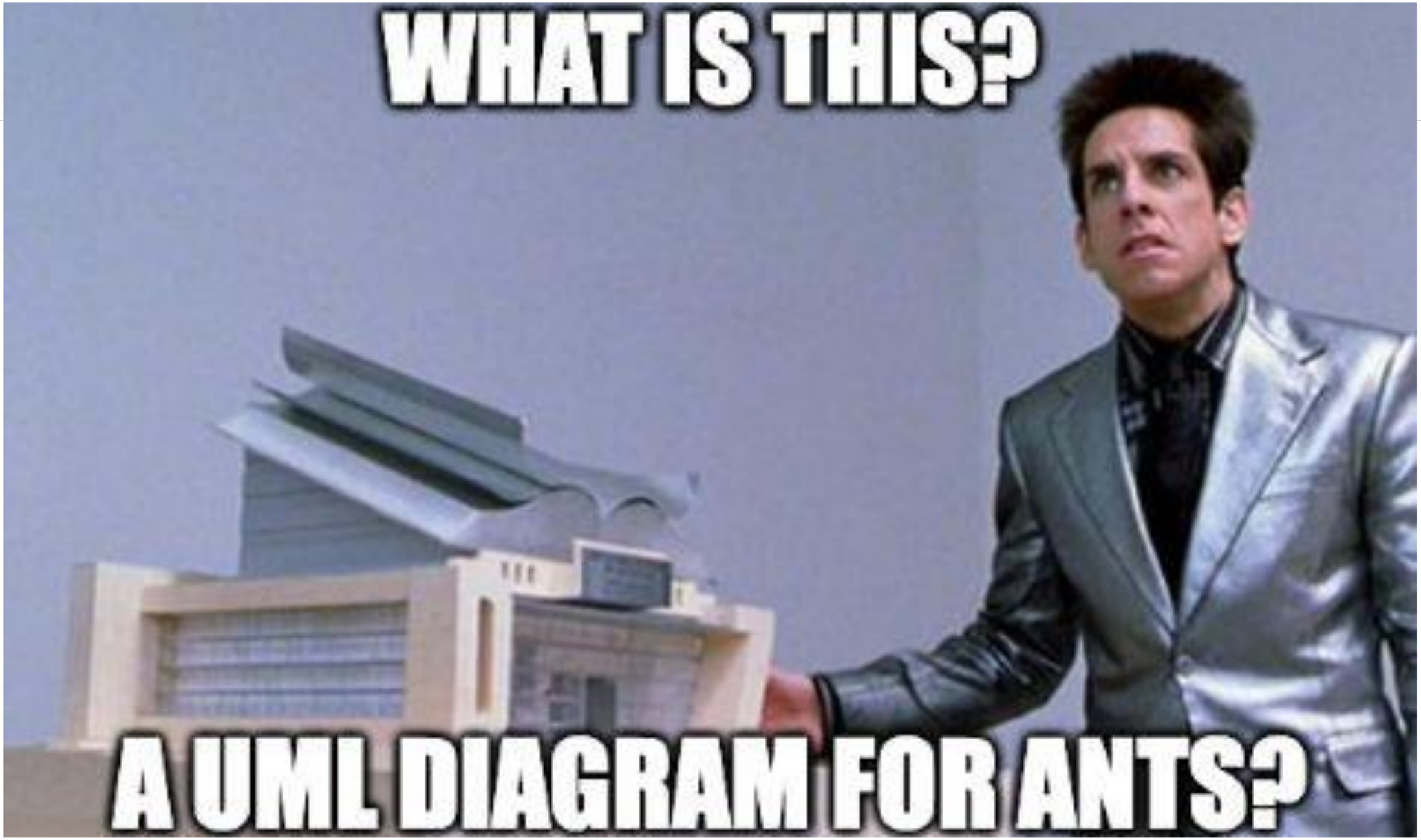
Inheritance

- Inheritance defines a relationship “IS_A” among classes



Real example: Facebook




WHAT IS THIS?
A UML DIAGRAM FOR ANTS?

Behavioral Diagrams

Behavioral Diagrams

- Verbs in the requirements
- Dynamic parts of UML models: "behavior over time"
- Usually connected to structural elements

- Types of Behavioral Diagrams
  - Interaction: a set of objects exchanging messages, to accomplish a specific purpose
  - State machines: specifies the sequence of states an object goes through during its lifetime in response to events (dynamic behavior)

State Diagram

- A state diagram represents the behavior of an object
- Graph representing a finite state machine
- Graph: net of states (nodes) and transitions (arrows)
- Useful for modeling a reactive (event-driven) system

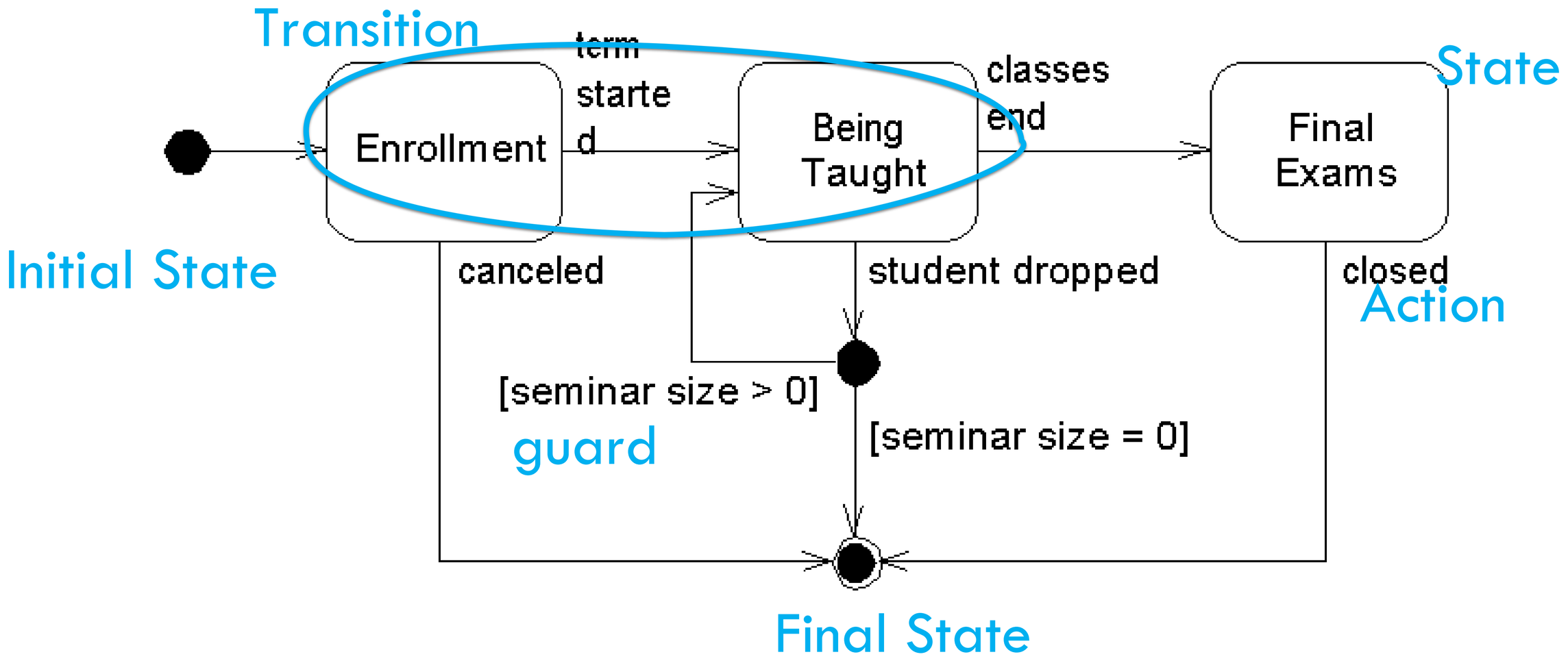
State Diagram

- A state is a stage in the behavior pattern of an entity. States are represented by the values of the attributes of an entity.

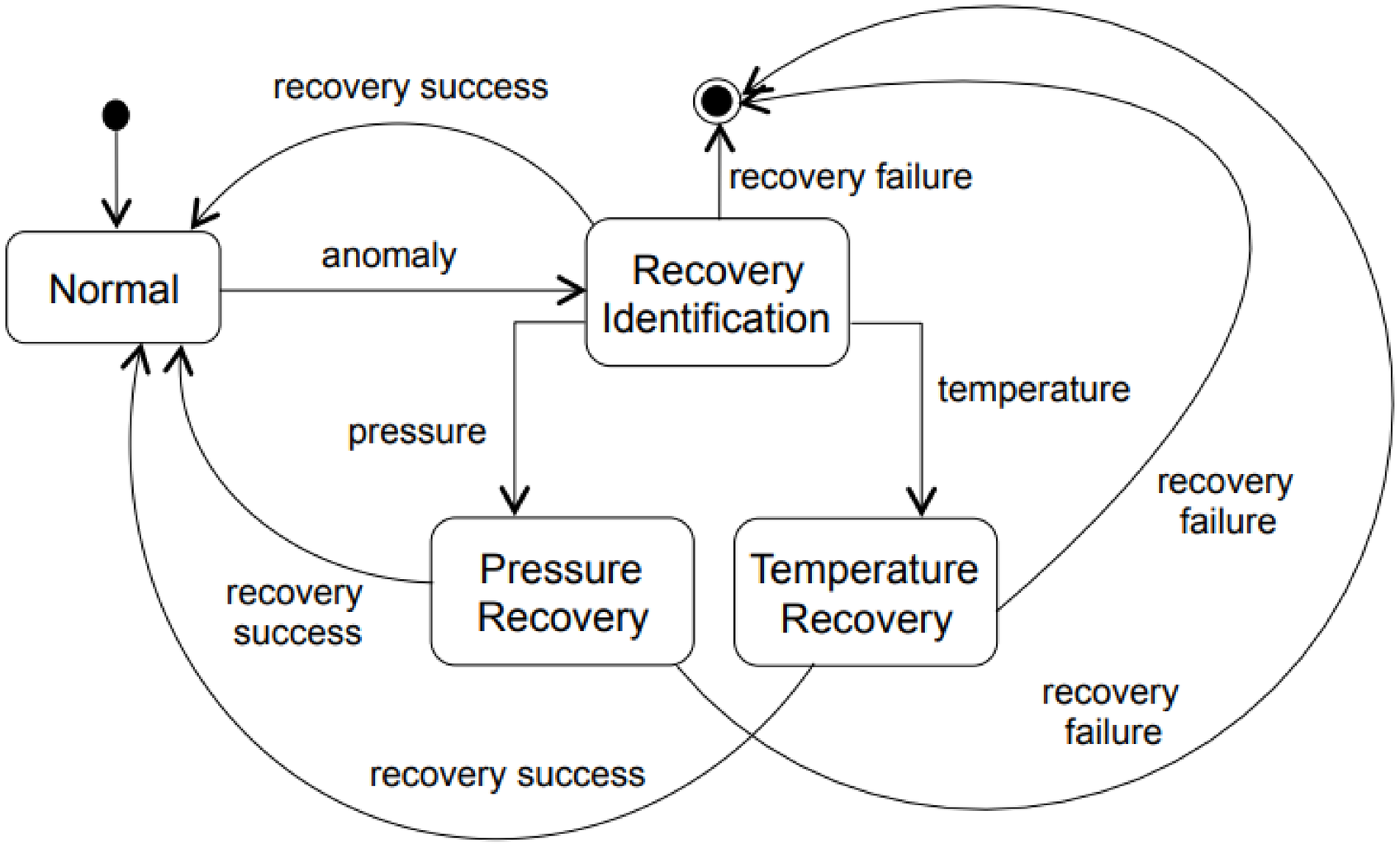
- A transition
  - progression from one state to another
  - triggered by an internal or external event

- A guard is a condition that must be true to traverse a transition.

State Chart: Terminologies

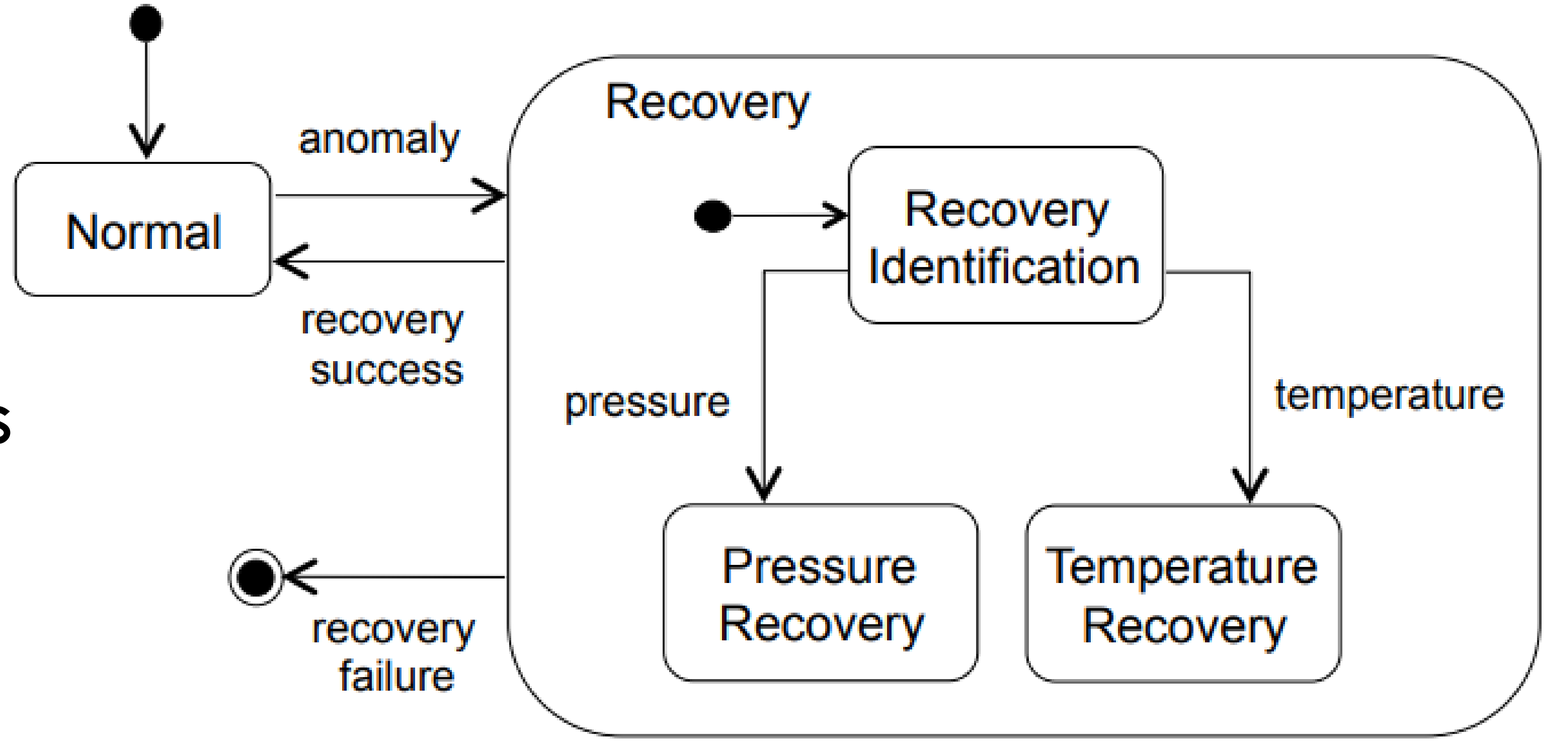


Example


recovery success
recovery failure
Normal
anomaly
Recovery Identification
temperature
pressure
recovery failure
Pressure Recovery
Temperature Recovery
recovery success
recovery failure
recovery success

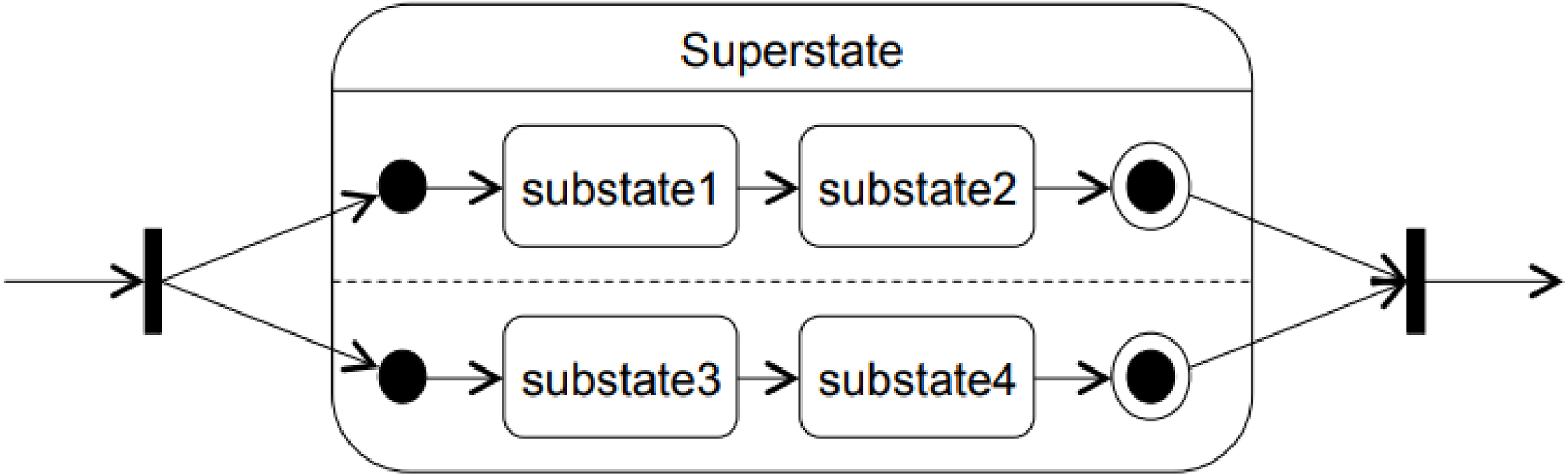
Composite States

- Used to simplify diagrams
- Inside, it looks like a state chart
- It may have transitions from substates
- It can be sequential or parallel



Parallel States

- Concurrency (multiple threads of control)



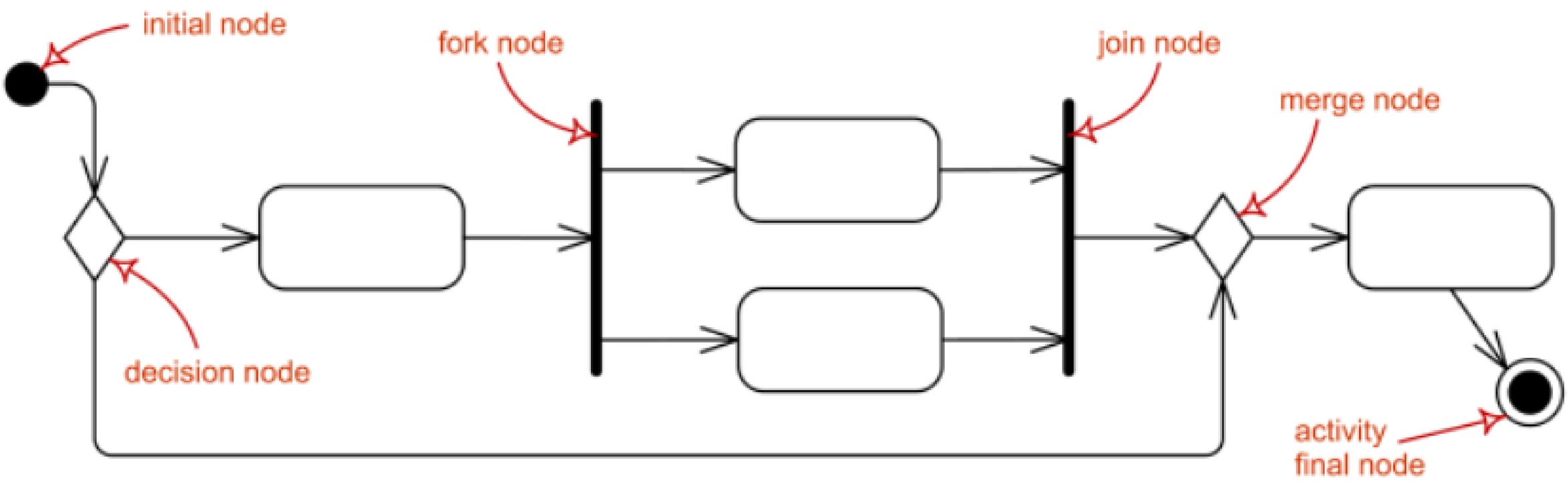
Exercise: State Charts

- Draw a state chart describing the operation of a cellular phone. Assume that the phone has keys for:
  - power on and off
  - keypad locking and unlocking
  - 0-9, #, and *
  - talk (or send) and end
- Model the following operations:
  - power on/off
  - keypad locking/unlocking
  - making calls (e.g., dialing, connecting, talking),
  - receiving calls (e.g., ringing, talking)

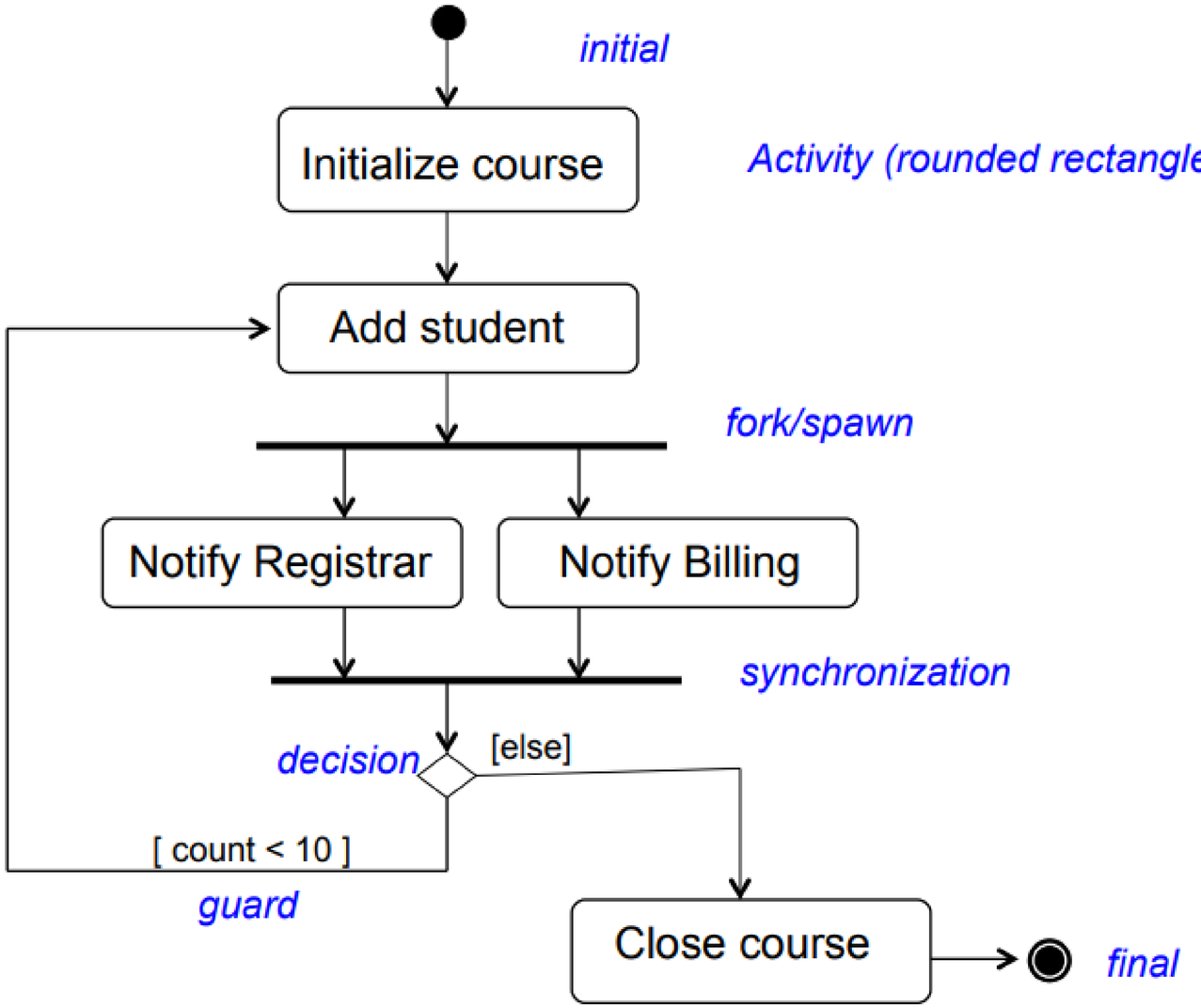
Activity Diagrams

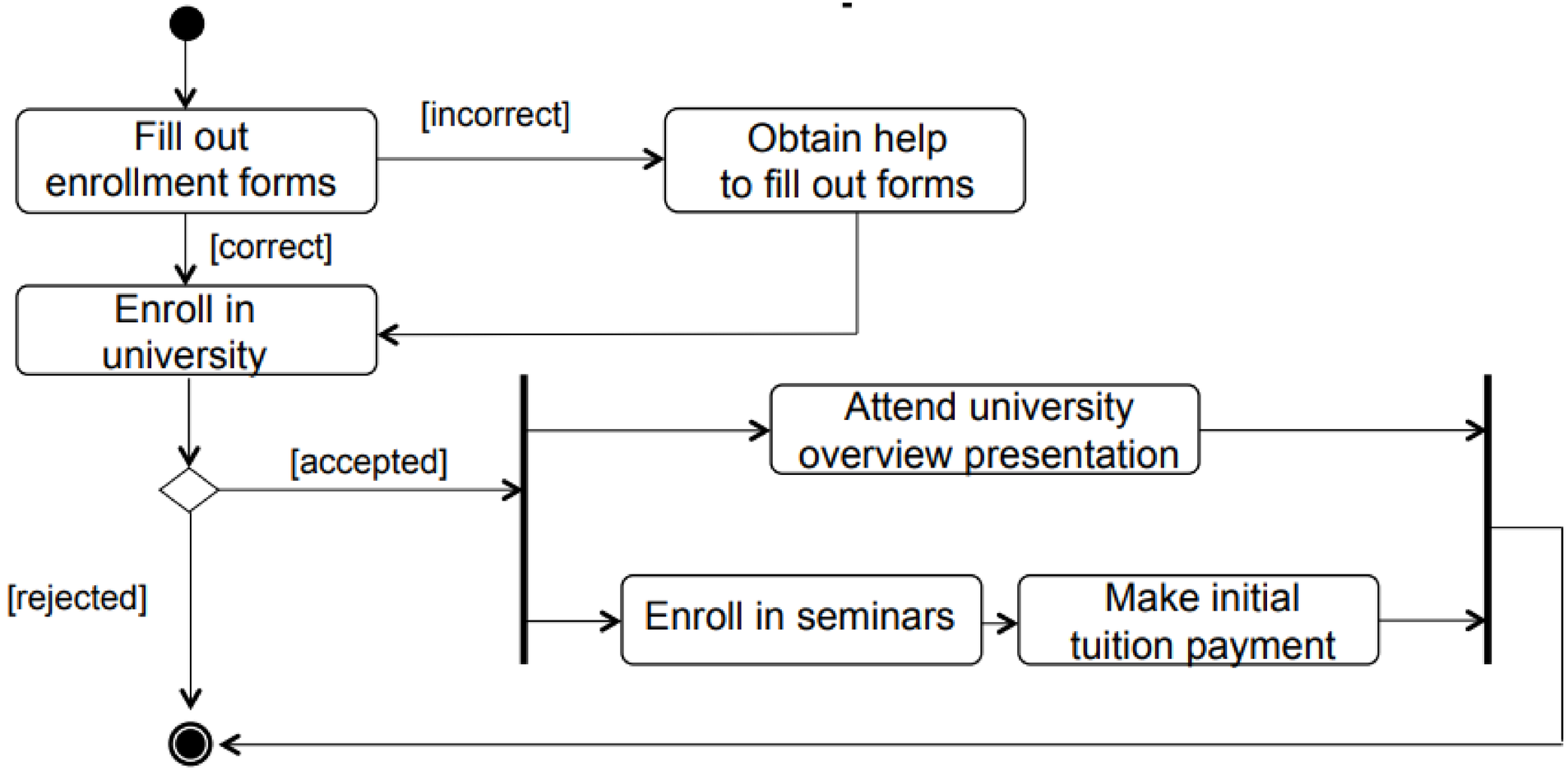
- Activity diagrams represent workflows of actions of several objects
- Activity diagram can be used to describe the activities of the components of a system
- Actions are composed by sequence, choice, iteration, and concurrency

Activity Diagram – Terminologies



Example

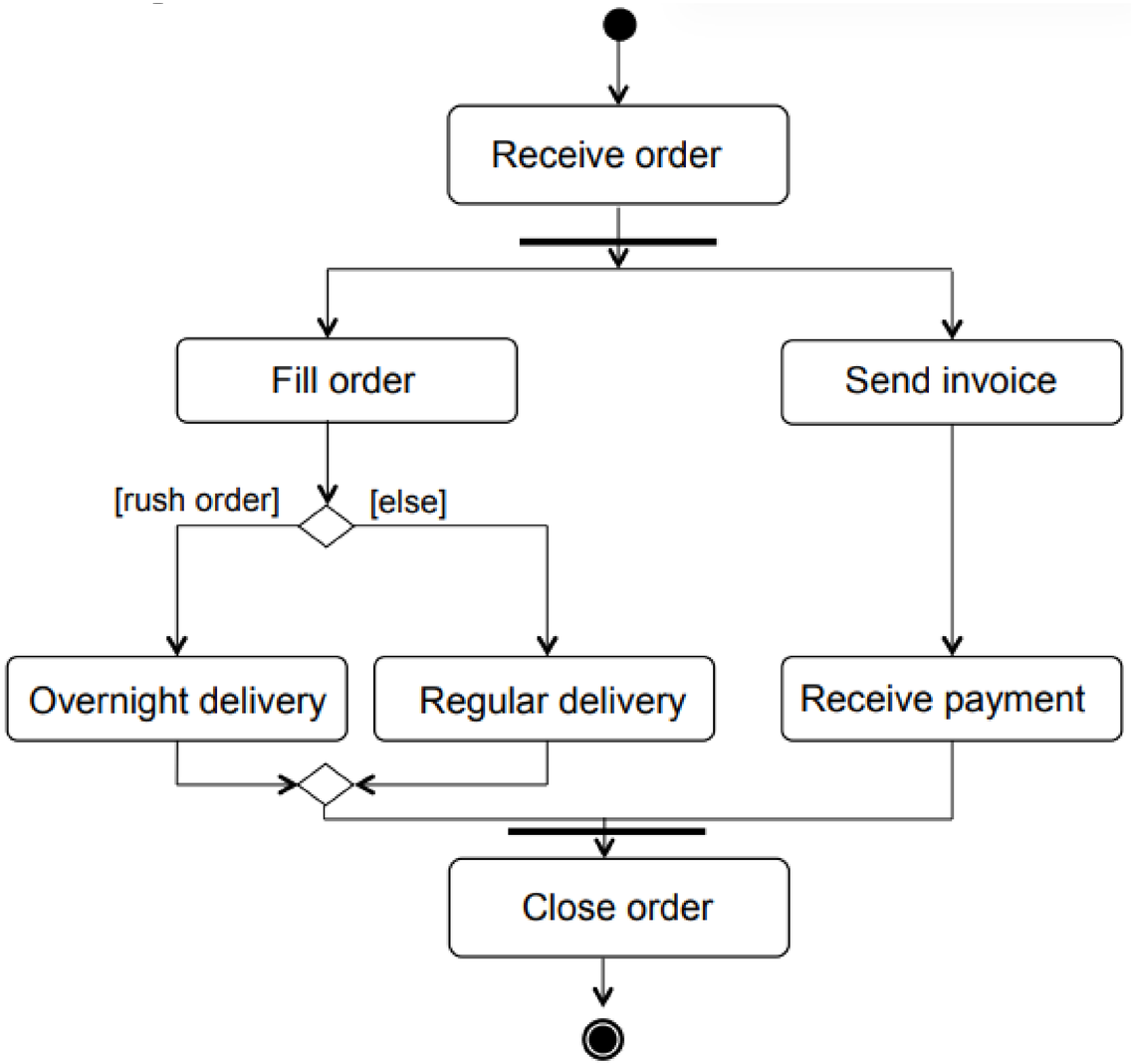

initial
Initialize course
Activity (rounded rectangle)
Add student
fork/spawn
Notify Registrar
Notify Billing
synchronization
decision
[else]
[count < 10]
guard
Close course
final


Fill out enrollment forms
[incorrect]
Obtain help to fill out forms
[correct]
Enroll in university
[accepted]
Attend university overview presentation
[rejected]
Enroll in seminars
Make initial tuition payment

Exercise

Draw an activity diagram for the following

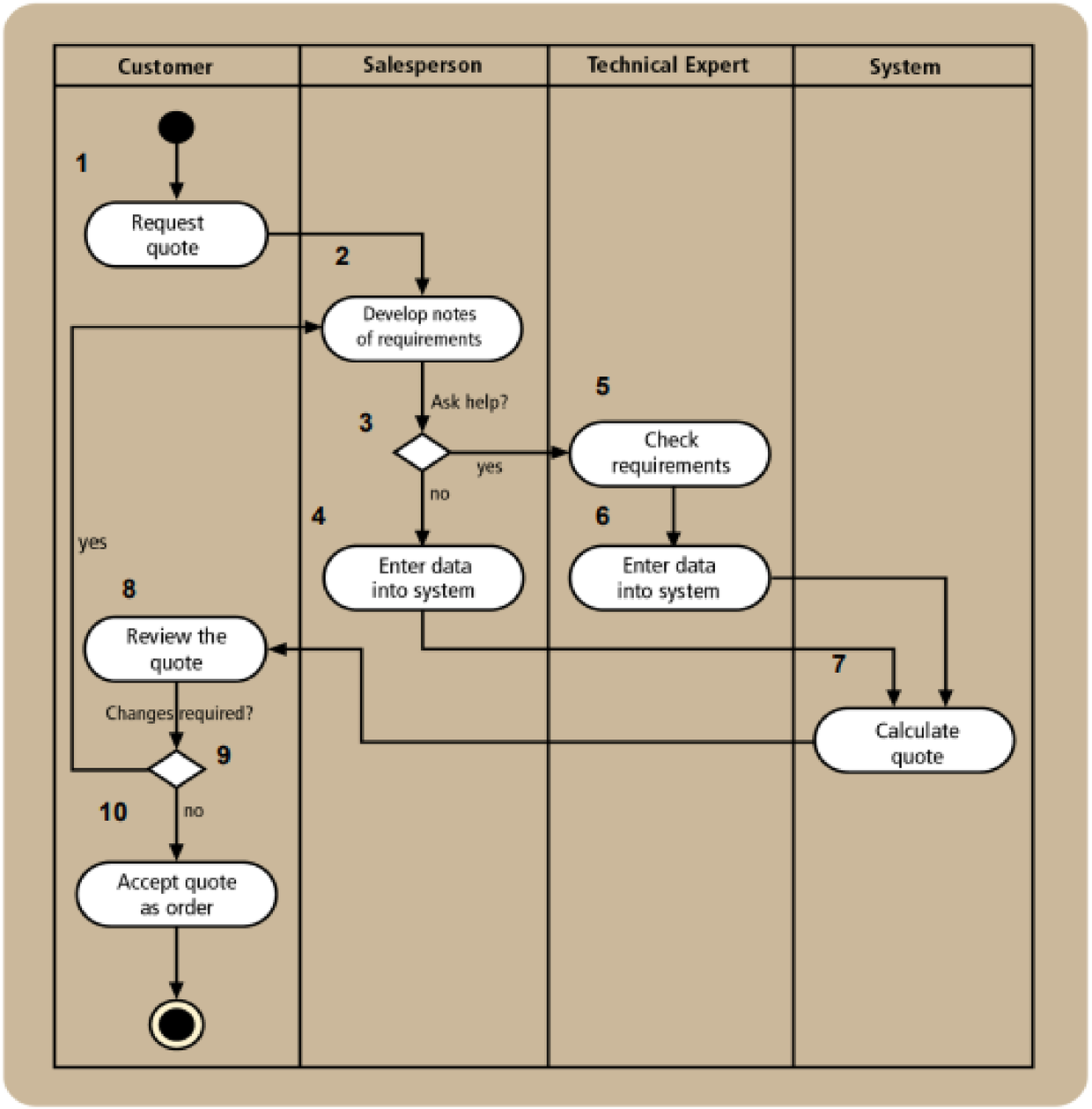
Upon receiving order, order is filled either for overnight(fast) or regular delivery and invoice is sent to the customer. Both operations can be done in parallel. Order is closed after receiving payment and successful delivery.


Receive order
Fill order
Send invoice
[rush order]
[else]
Overnight delivery
Regular delivery
Receive payment
Close order

Actor in Activity Diagram - Swimlanes

Exercise: Activity Diagram

Step ID	Process Step or Decision	Who/What Performs	Parallel Activity	Loop	Preceding Step
1	Request quote	Customer	No	No	-
2	Develop requirement notes	Salesperson	No	Yes	1
3	Decision: Help?	Salesperson	-	Yes	2
4	Salesperson enters data	Salesperson	No	Yes	3
5	Check requirements	Technical Expert	No	Yes	3
6	Tech. expert enters data	Technical Expert	No	Yes	5
7	Calculate quote	System	No	Yes	4, 6
8	Review quote	Customer	No	Yes	7
9	Decision: Changes?	Customer	No	Yes	8
10	Accept quote as order	Customer	No	No	9


Customer
Salesperson
Technical Expert
System
1
Request quote
2
Develop notes of requirements
Ask help?
3
yes
no
5
Check requirements
4
Enter data into system
6
Enter data into system
yes
8
Review the quote
7
Calculate quote
Changes required?
9
10
no
Accept quote as order

Modelling Interactions

Sequence Diagrams

Interaction Diagrams

- A use case diagram presents an outside view of the system

- The inside view of a system is shown by interaction diagrams

- Two types of interaction diagrams
  - Sequence diagrams
  - Collaboration (Communication) diagrams

Modeling Interaction

- Sequence diagram
  - Depicts objects' interaction by highlighting the time ordering of method invocations

- Communication (collaboration) diagram
  - Depicts the message flows among objects

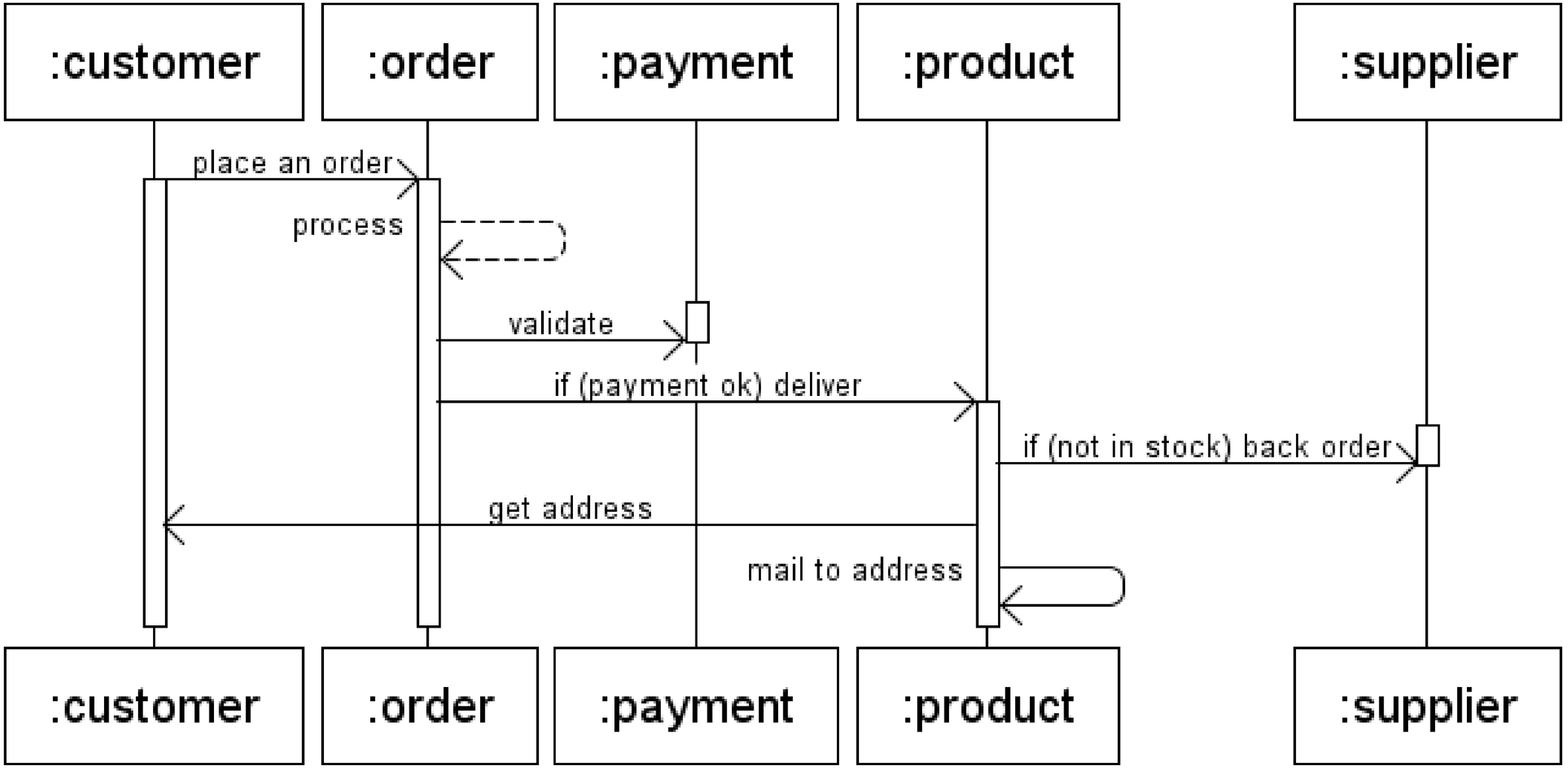
Sequence Diagram

- Participant
  - An object that acts in the sequence diagram

- Message
  - Communication between participant objects

- The axes in a sequence diagram
  - Horizontal shows the acting object/participant.
  - Vertical represent time. (down→ forward in time)

Sequence Diagram



Sequence diagram

Customer specifies an author on the search page.

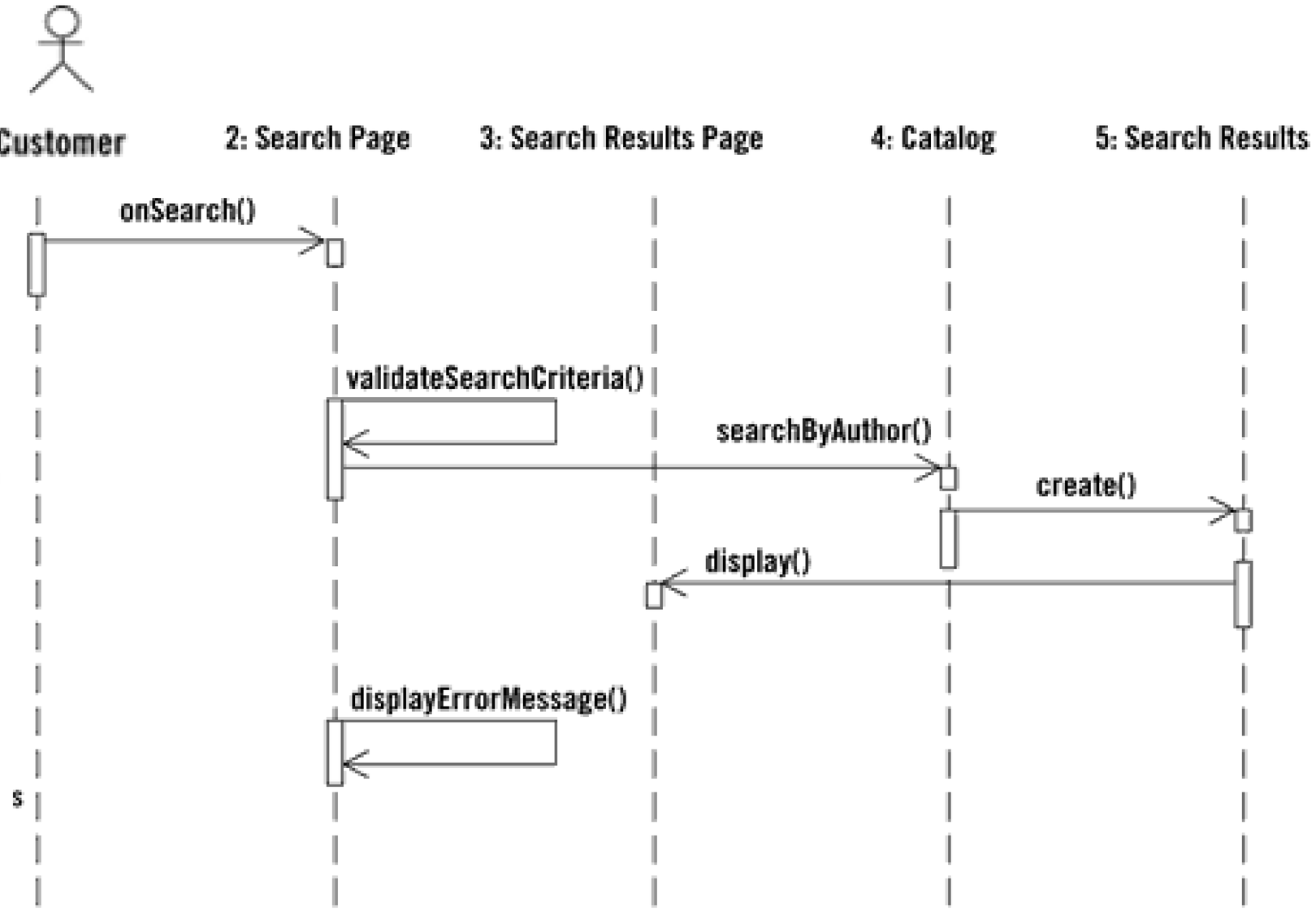
The system validates the search criteria.

The System searches the Catalog with specified author.

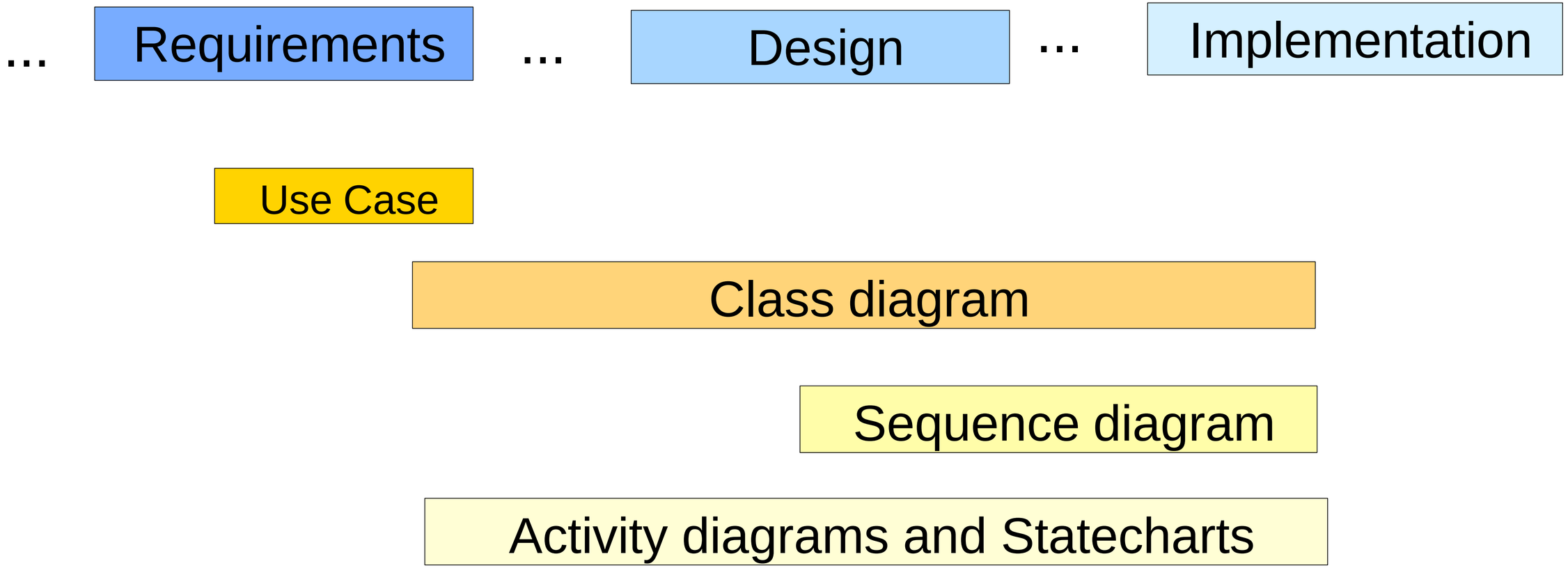
When search completes, system displays search results on page.

Alternative:

If customer didn't enter author before pressing search button, system displays an error message



Diagrams in SDLC



Readings

- On use cases
  www.ibm.com/developerworks/rational/library/5383.html
- On class diagrams
  www.ibm.com/developerworks/rational/library/content/RationalEdge/sep04/bell/index.html
- On activity diagrams
  www.ibm.com/developerworks/rational/library/2802.html
- On sequence diagrams
  www.ibm.com/developerworks/rational/library/3101.html

Tools

- Eclipse + several plugins, like Omondo
- argouml.tigris.org Argo or Poseidon
- www.genmymodel.com free online tool, sharable diagrams
- www.lucidchart.com web application, need license
- violet.sourceforge.net Open source editor for UML
- www.borland.com/us/products/together/index.html Borland Together
- www.visual-paradigm.com Visual Paradigm suite
- www.nomagic.com Magicdraw suite
- abstratt.com text UML
- www.umlgraph.org web application for class and seq diagrams
- www-01.ibm.com/software/rational/ Rational Rose
- jazz.net IBM platform
- smartuml.sourceforge.net UML on tablet PC
- metauml.sourceforge.net Beautiful UML diagrams in LaTeX
- softwarestencils.com/uml Images reusable in a graphic editor
- yuml.me Fast draw of UML diagrams for web pages

Thank you!