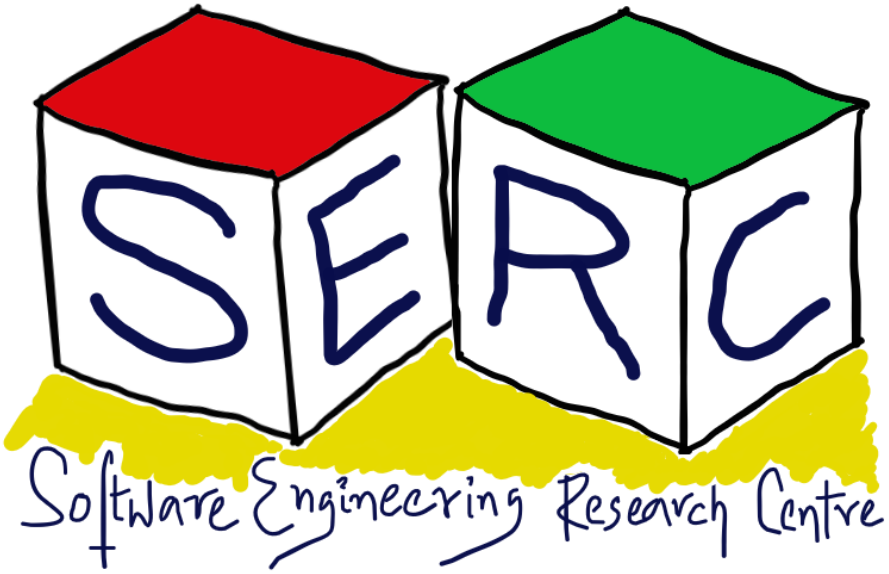
Software Modeling: An Overview

CS6.401 Software Engineering



Dr. Karthik Vaidhyanthan

karthik.vaidhyanathan@iiit.ac.in

https://karthikvaidhyanathan.com

INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY
HYDERABAD

Acknowledgements

The materials used in this presentation have been gathered/adapted/generate from various sources as well as based on my own experiences and knowledge

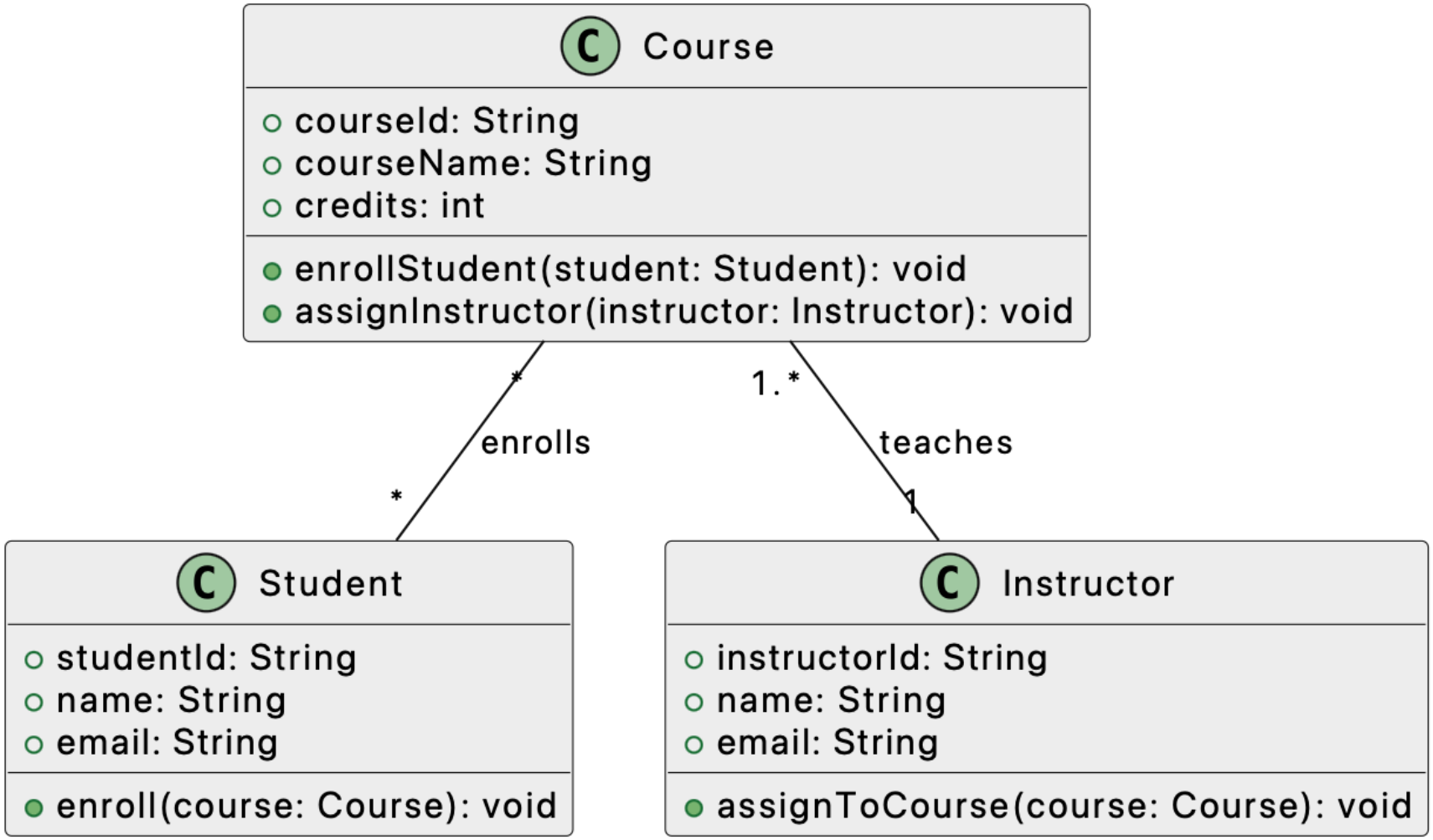
-- Karthik Vaidhyanathan

Sources:

1. Introduction to MDE, Ludovico Iovino, GSSI, Italy
2. UML@Classroom, An Introduction to Object-Oriented Modeling by Martina Seidl, Marion Scholz, Christian Huemer and Gerti Kappel
3. UML Modelling lecture, Dr. Raghu, IIIT Hyderabad

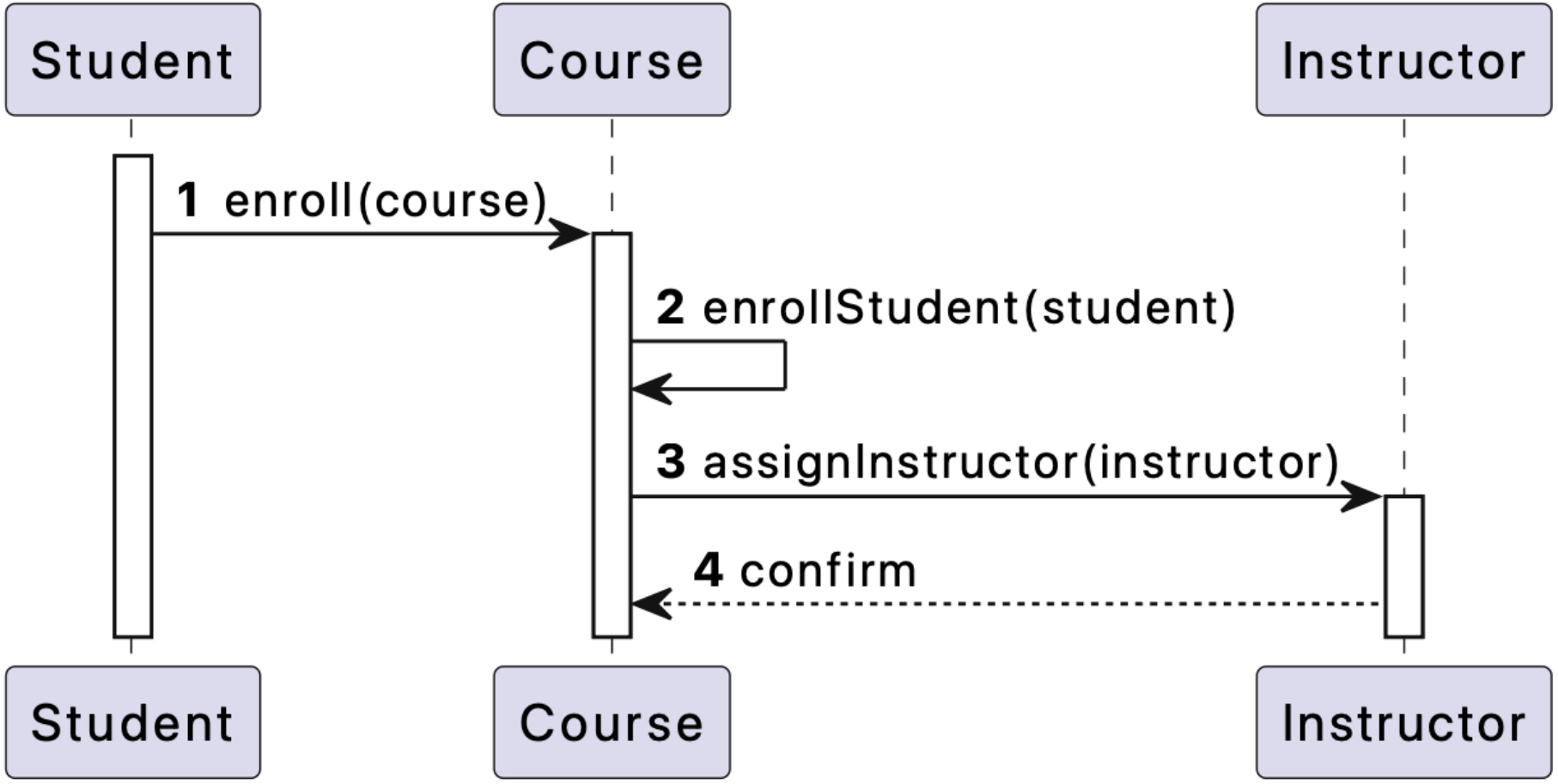
Source: Google images

A Simple Class Diagram



Modeling the Dynamic Aspects: Sequence Diagram [Interaction Diagram]

What about the interaction?



Sequence Diagram

- Captures the dynamic behavior
- Two dimensional-diagram
  - Horizontal: Involved interaction
  - Vertical: Chronological order of the interaction
- Interaction => sequence of event specifications

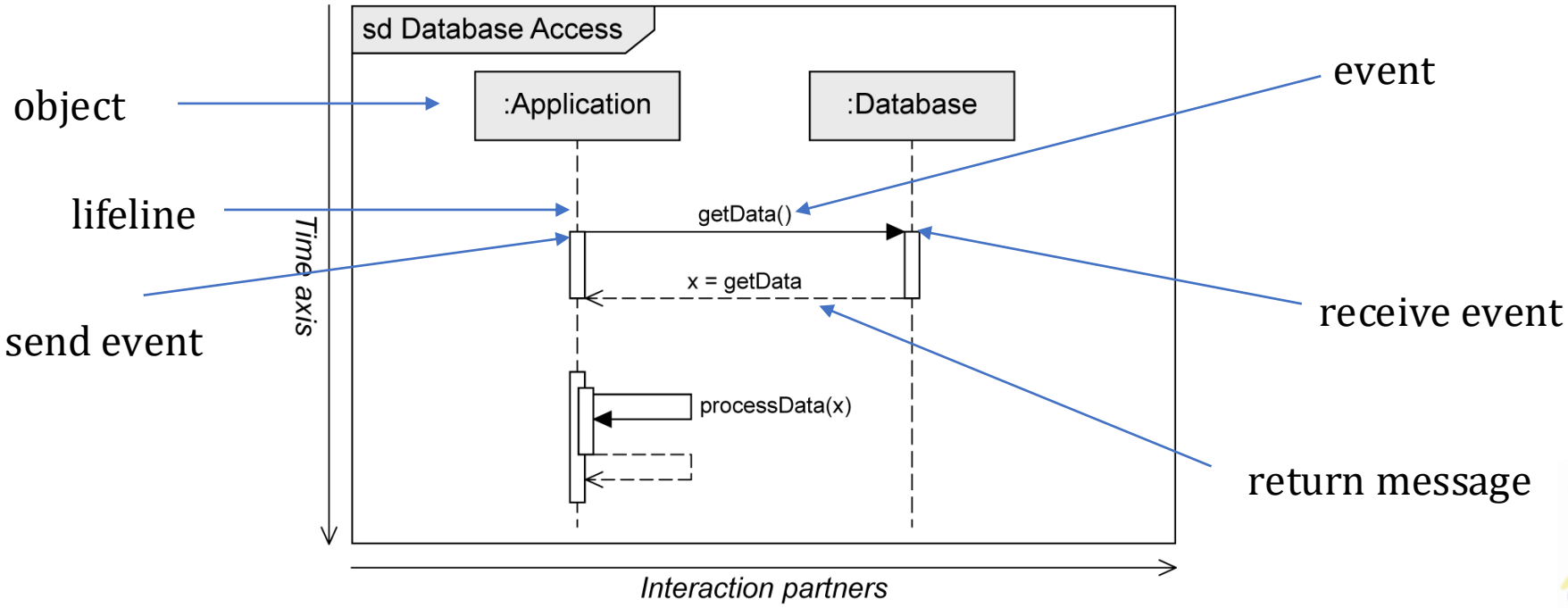
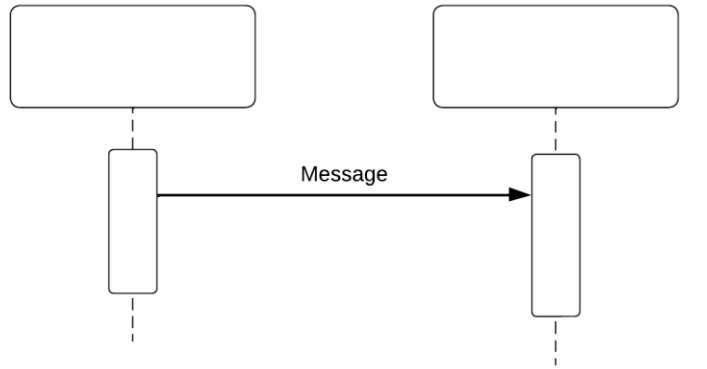


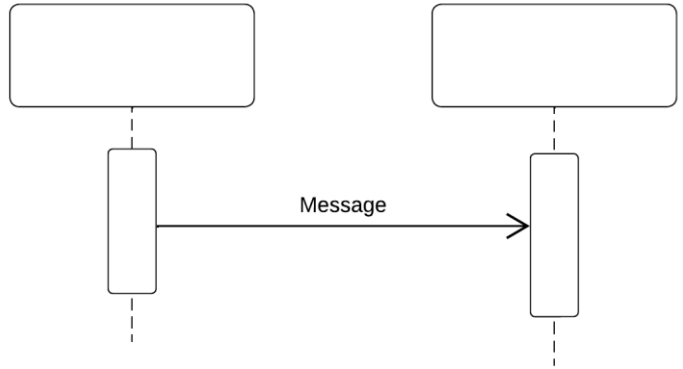
Image source: uml@classroom

Sequence Diagram – Main Message types

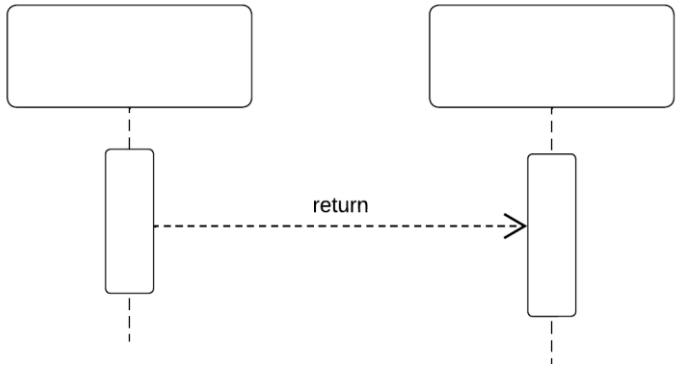
- Synchronous Message
  - Sender waits till the return message is received before next



- Asynchronous Messages
  - Sender does not wait for response message

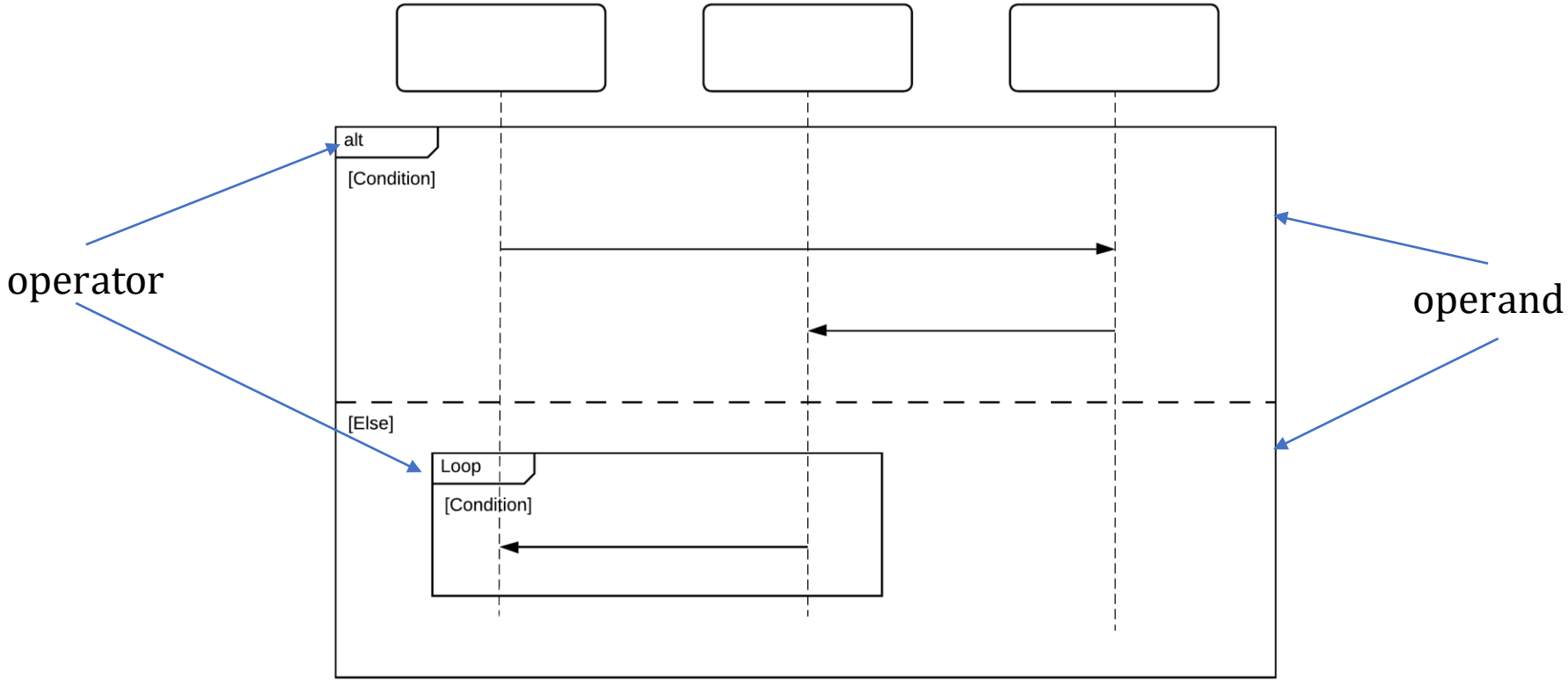


- Response message
  - Not mandatory in obvious situations



Sequence Diagram – Combined Fragments

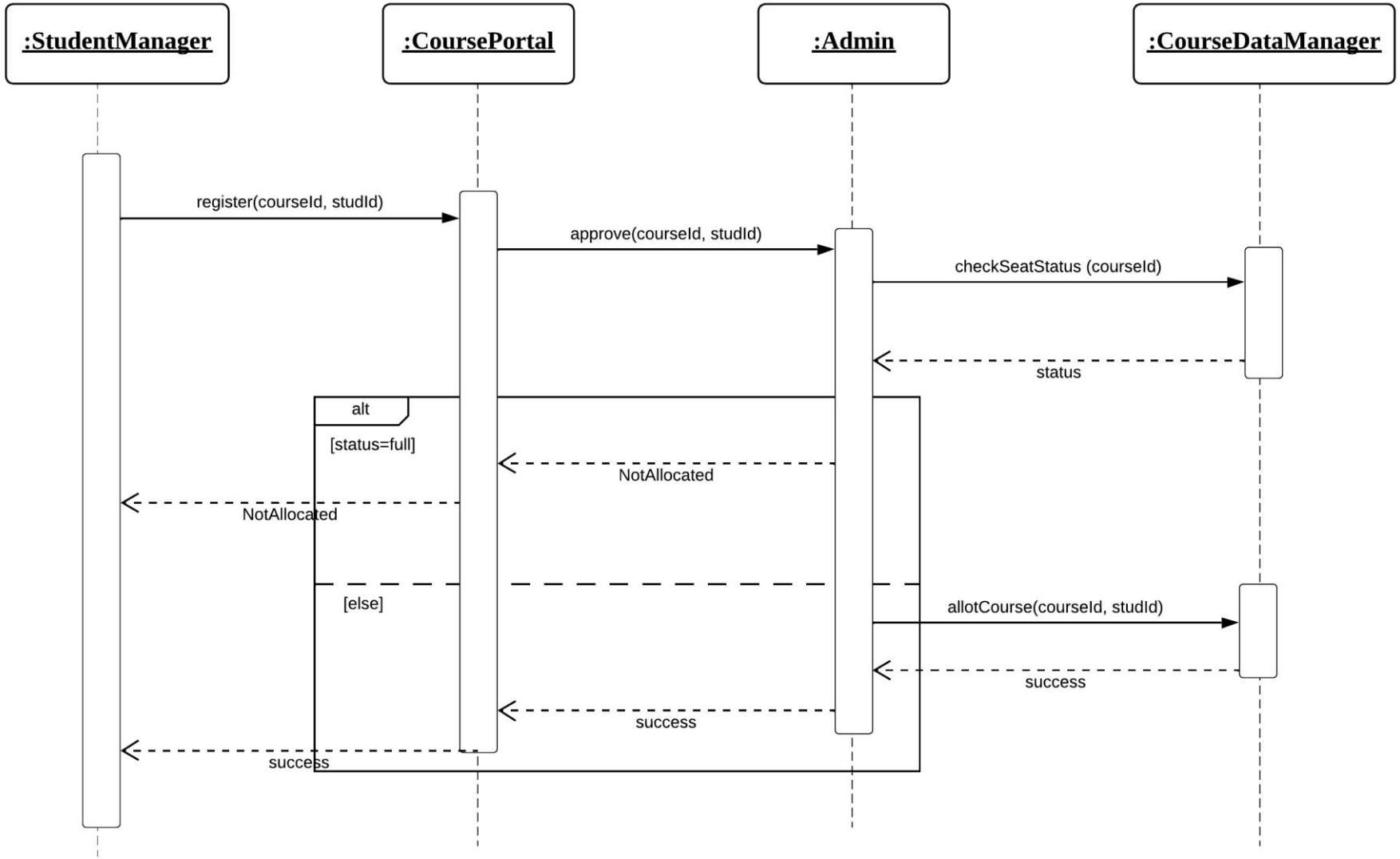
- Model control structures explicitly
- UML sequence diagram supports 12 operators. Three groups
  - Branches and loops, Concurrency and order, Filters and Assertions



Different Operators

Name and Operator	Use
Alternative (alt)	Express alternative execution (if-else)
Optional (opt)	Fragment executes based on guard condition (if)
Break (break)	Execution of a fragment when break condition is met
Loop (loop)	Repeated execution of a fragment
Sequential (seq)	Weak ordering (default model)
Strict (strict)	Interaction with strict order
Parallel (par)	Concurrent execution of sub-scenarios
Critical (critical)	Atomic interactions (no other interactions can affect)
Ignore (ignore)	Irrelevant messages (insignificant messages at runtime)
Consider (consider)	Important messages of the interactions
Negate (neg)	Model invalid interactions, undesirable situations
Assert (assert)	To assert certain interactions (mandatory)

Sequence Diagram – Example



Time for an activity

Lets create the interaction diagram for the course management system. Expand your modeling with the interaction aspects. You are free to revisit/modify your class diagram

Thank You

Course website: karthikv1392.github.io/cs6401_se

Email: karthik.vaidhyanathan@iiit.ac.in
Web: https://karthikvaidhyanathan.com