

UML For PBL

Vasile Drumea

November 12, 2023

Introduction

History

Problems

Ambiguity

Chaos

Diagrams

Important OOP Stuff

Structural

Use Case

Class

Component

Deployment

Behavioral

Sequence

Activity

State Machine

Strengths

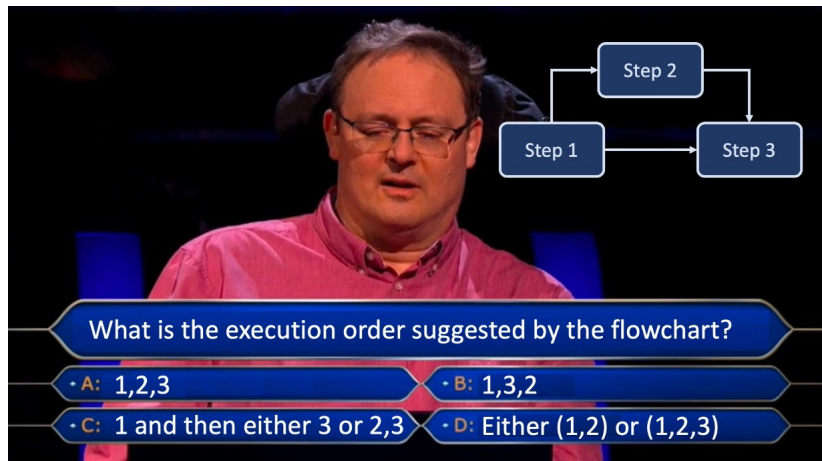
Introduction

- ▶ Started growing from OOP Concepts (1980s - 1990s)
- ▶ Object Management Group (OMG) (1997)
- ▶ UML 2, major revision (2005)

Main Problem

What problems can we encounter in modeling?

Ambiguous Diagrams

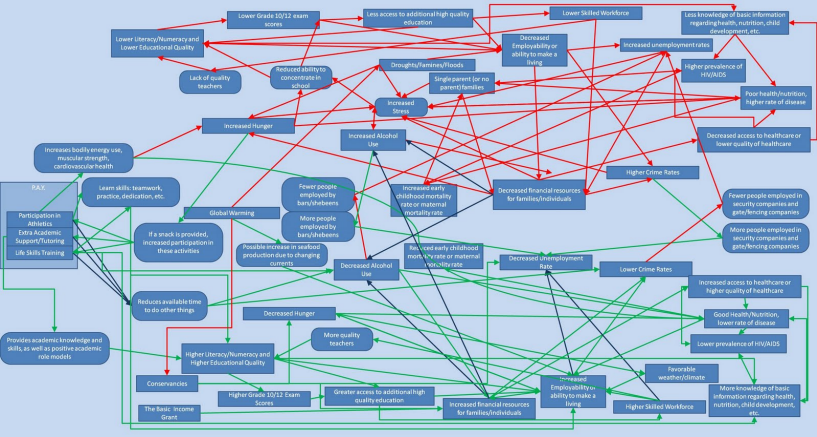


The flowchart consists of three rectangular boxes: 'Step 1' at the bottom left, 'Step 2' at the top, and 'Step 3' at the bottom right. An arrow points from 'Step 1' to 'Step 2'. Another arrow points from 'Step 1' to 'Step 3'. A third arrow points from 'Step 2' to 'Step 3'.

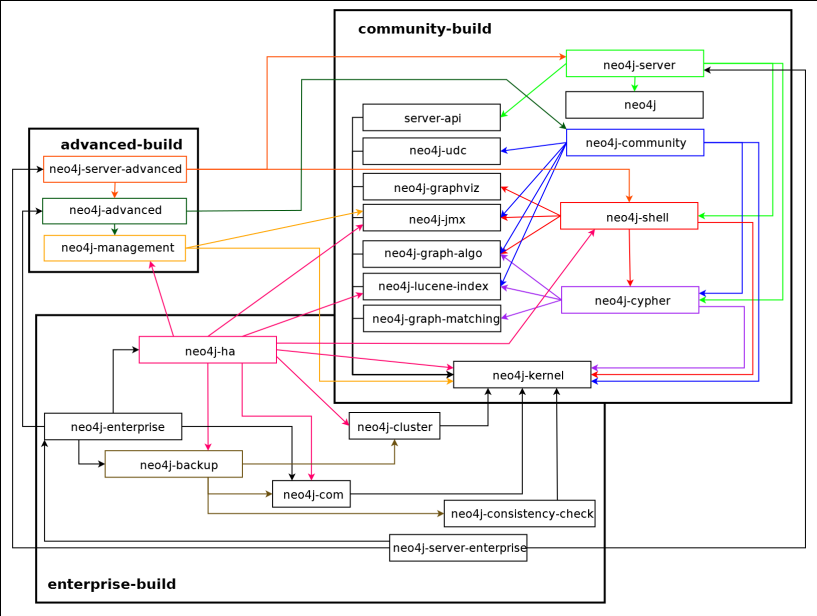
What is the execution order suggested by the flowchart?

- A: 1,2,3
- B: 1,3,2
- C: 1 and then either 3 or 2,3
- D: Either (1,2) or (1,2,3)

Chaos #1



Chaos #2



OOP Pillars

Composition:

- ▶ The HAS-A relationship;
- ▶ Code reuse through delegation;
- ▶ It is very change friendly.

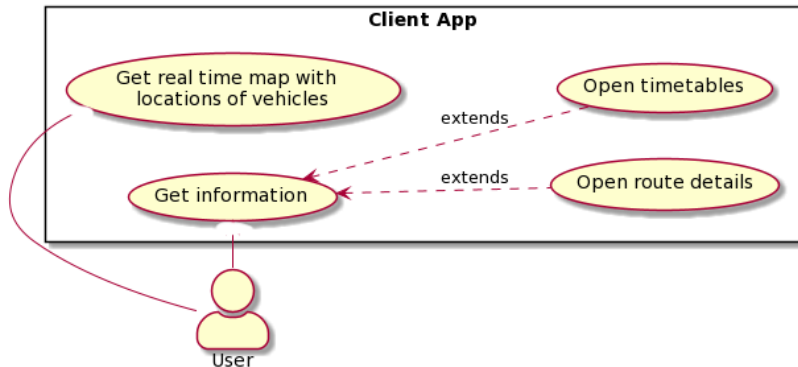
Inheritance:

- ▶ The IS-A relationship;
- ▶ Code reuse through inheritance;
- ▶ It can become very rigid.

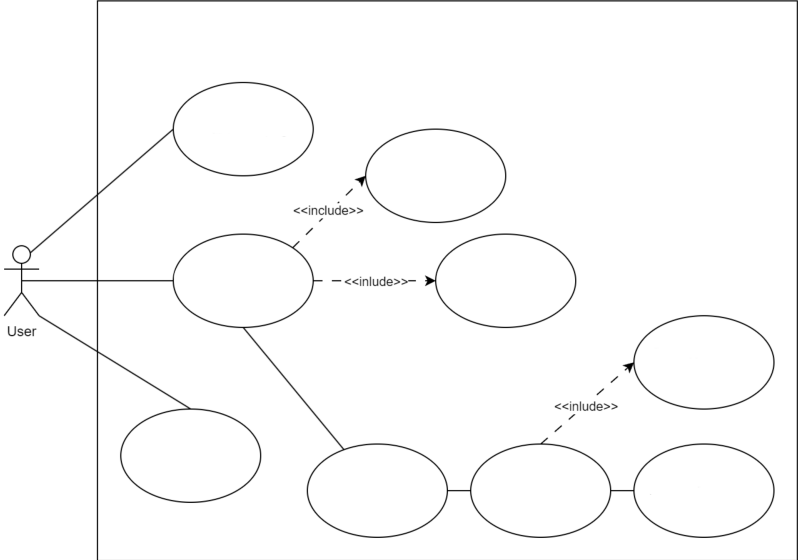
Use Case D. Tips

- ▶ Include the most relevant use case (domain specific ones).
- ▶ Keep them simple.
- ▶ Different diagrams for different types of actors, subsystems or perspectives.

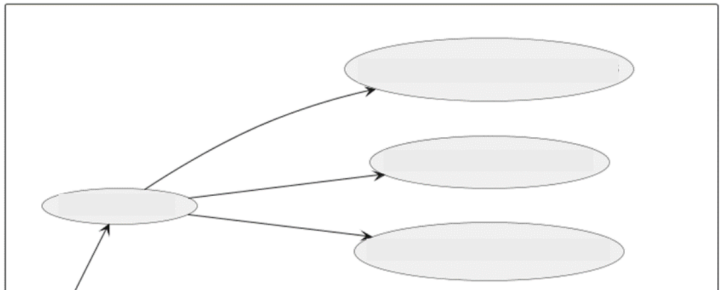
Use Case D. Example



Use Case D. Mistakes



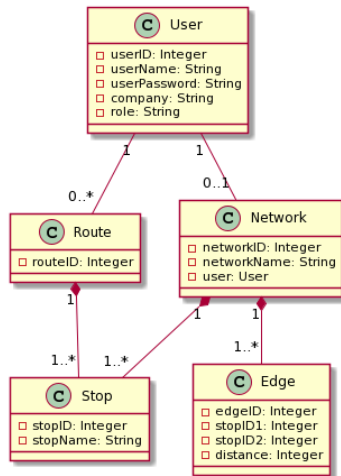
Use Case D. Mistakes



Class D. Tips

- ▶ You can use them to represent the data models or entities from a RDBMS.
- ▶ Keep them small (Generally diagrams should be readable if you put them in a slide).
- ▶ Use multiplicities.

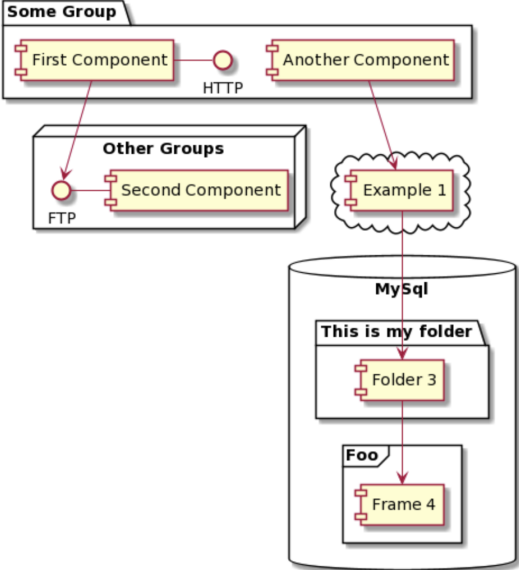
Class D. Example



Component D. Tips

- ▶ It is usually more abstract than deployment diagram.
- ▶ You can use it in order to represent non-concrete things.
- ▶ The client side plugs into the interface, the provider exposes the interface.

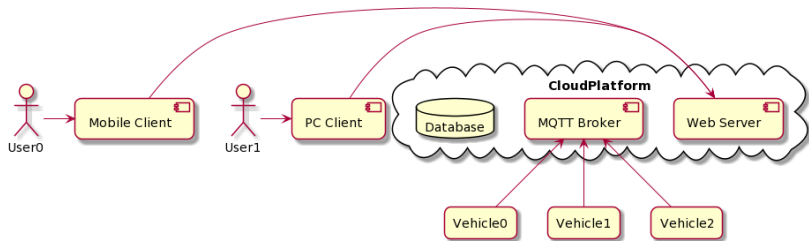
Component D. Example



Deployment D. Tips

- ▶ If you'll be using components into deployment diagrams maybe you don't need the component ones.
- ▶ Represent a deployment model as close as a real world one.
- ▶ Be careful with arrows to not get them tangled.

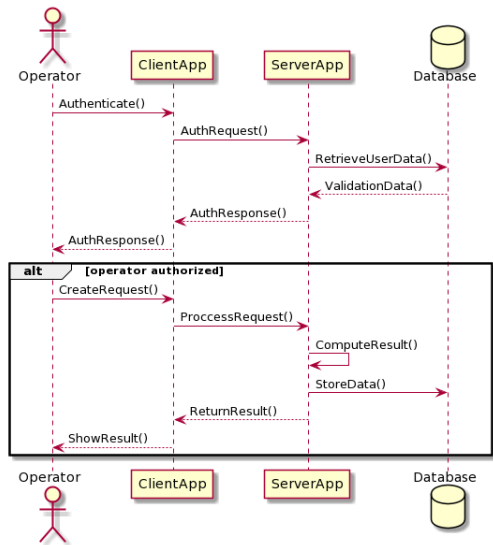
Deployment D. Example



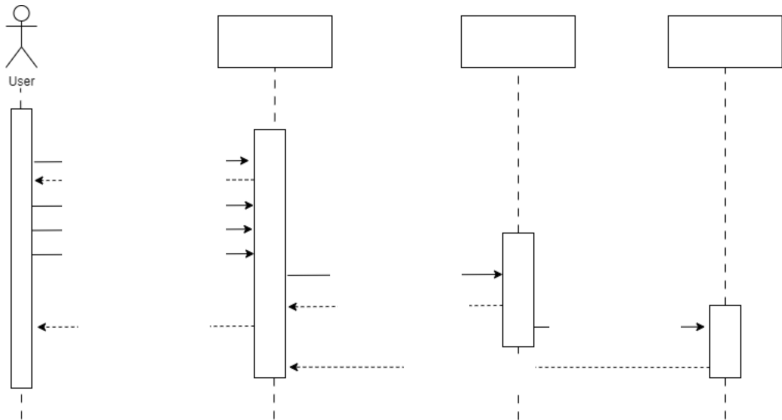
Sequence D. Tips

- ▶ All the requests should be ended by responses or destruction occurrences.
- ▶ Synchronous requests should be followed by a response or an asynchronous request.
- ▶ Divide the sequences using fragments.

Sequence D. Example



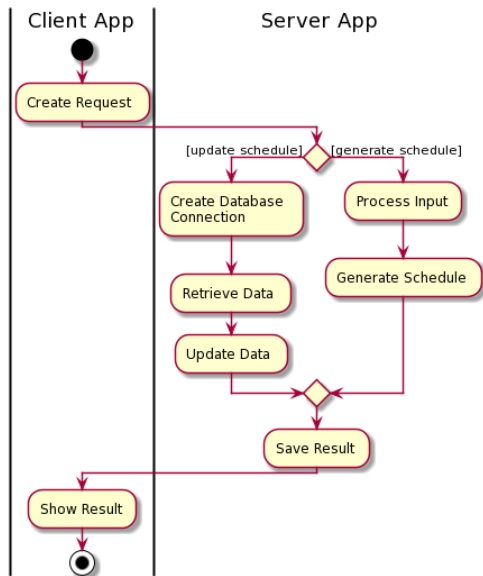
Sequence D. Mistakes



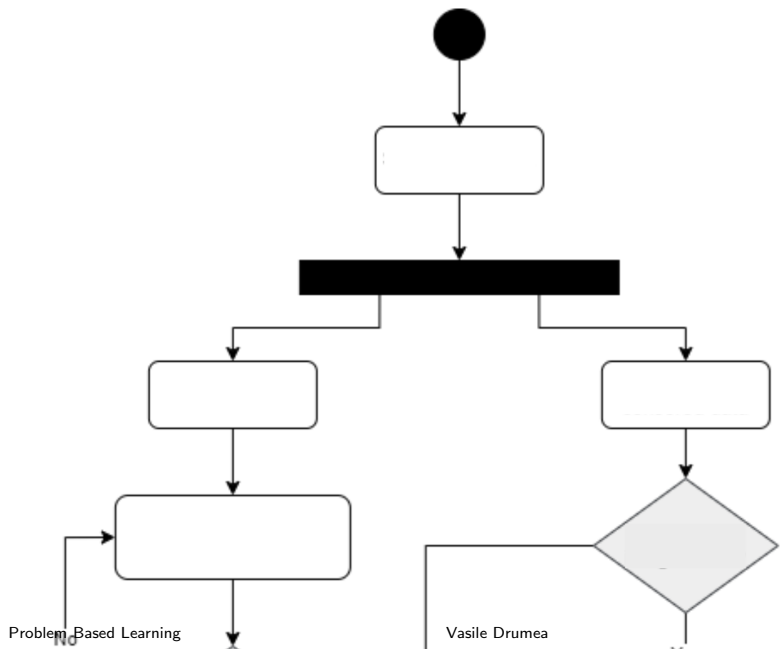
Activity D. Tips

- ▶ Use one initial node and one final node.
- ▶ You can split vertically/horizontally based on the subsystem.

Activity D. Example



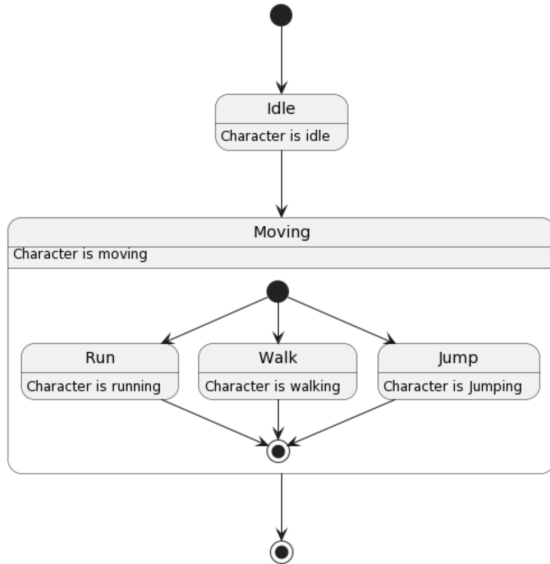
Activity D. Mistakes



State Machine D. Tips

- ▶ You can group state transitions into states.
- ▶ In case you do that provide start and end states.

State Machine D. Example



Strengths

- ▶ Multiple perspectives of a system
- ▶ Non Ambiguous (if used correctly)
- ▶ Varies from simple/intuitive to complex/technical

Summing up

- ▶ Make your diagrams viewer friendly:
 - ▶ Readable;
 - ▶ Intuitive;
 - ▶ On point;
- ▶ A diagram, if put in a slide should be readable.
- ▶ Like in life, when creating diagrams:
 - ▶ Don't overthink;
 - ▶ Make your point visible;
 - ▶ Put some love into it;

Thanks for your attendance/attention!
Questions?