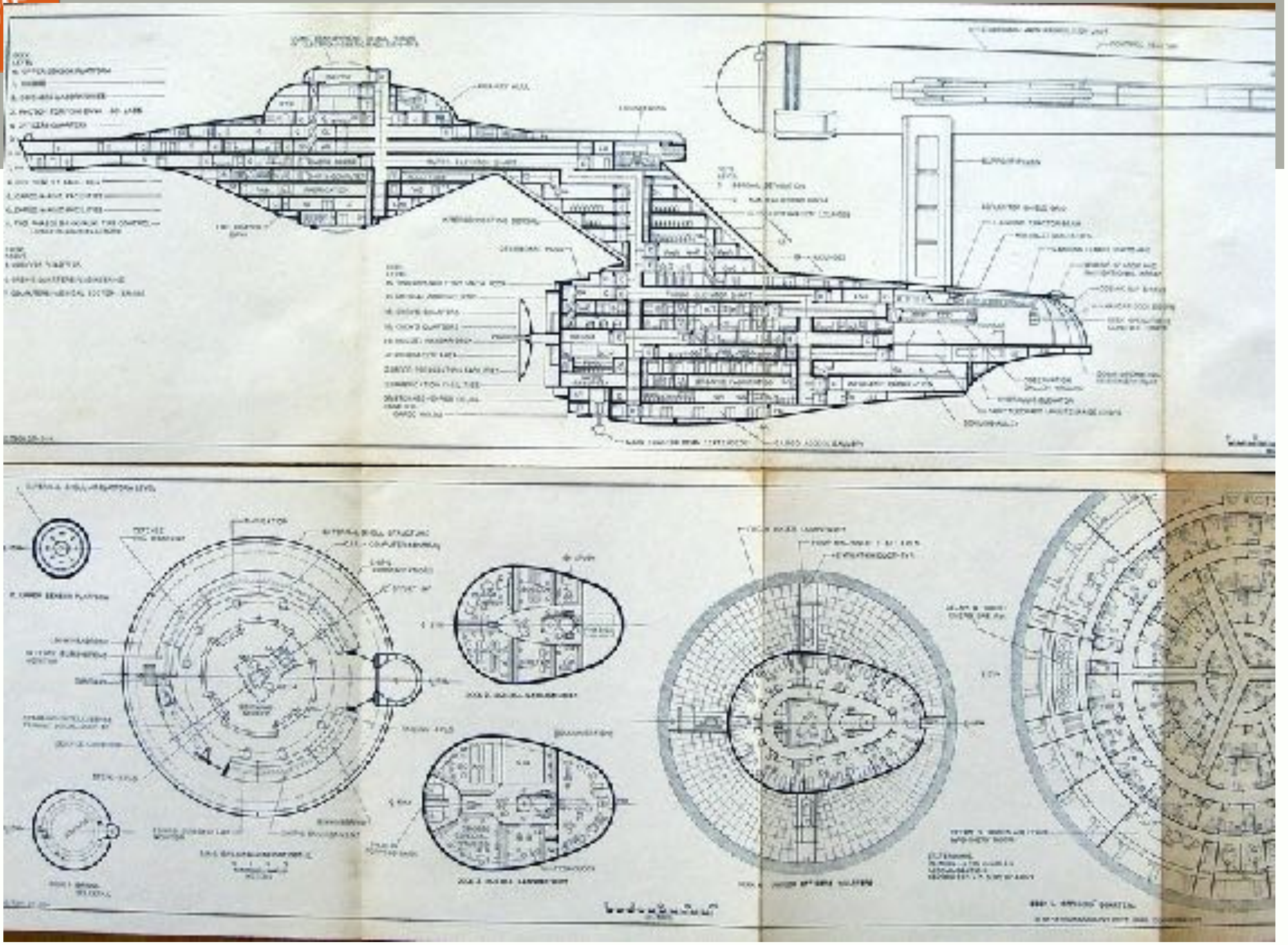
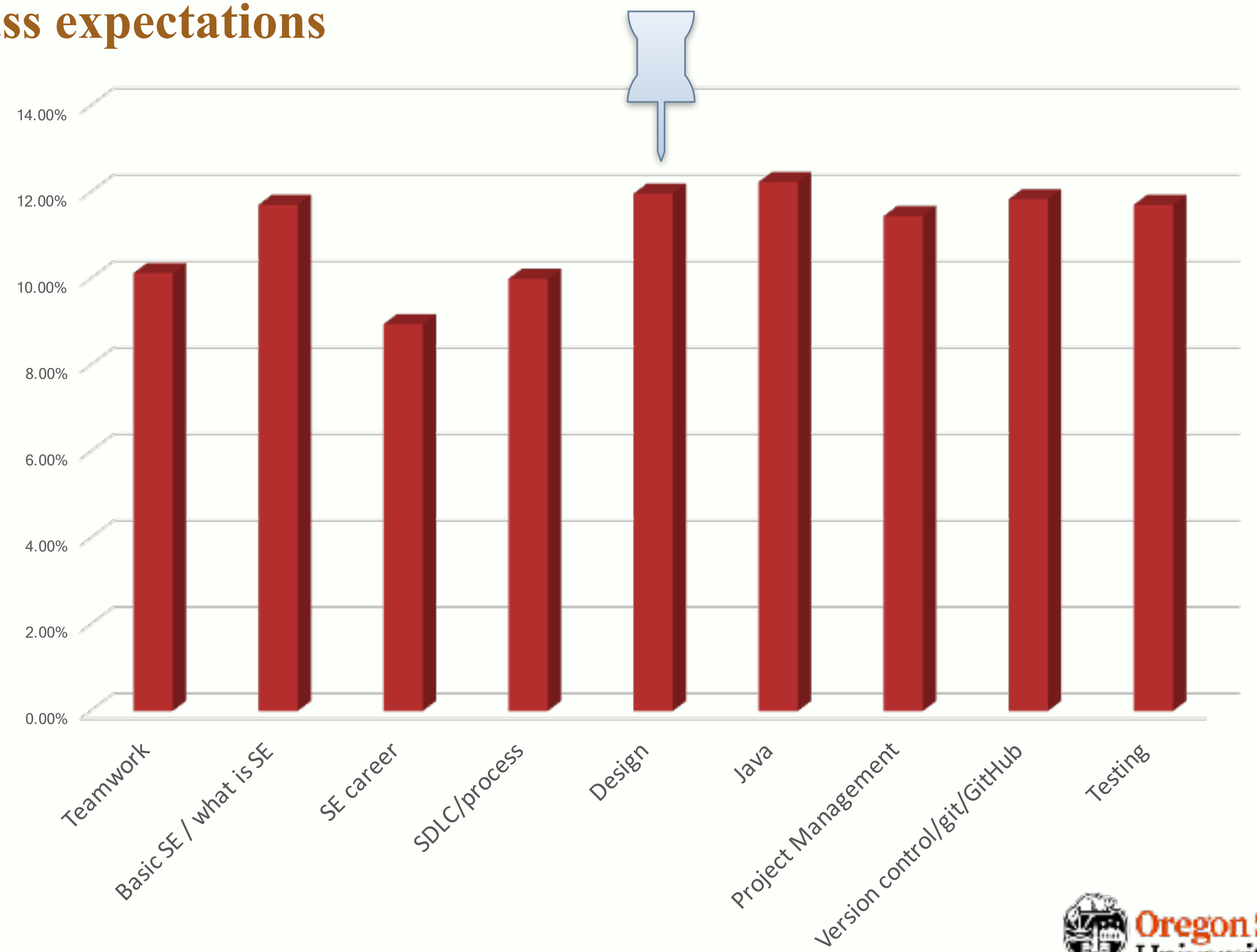
Diagram Notations - UML



Announcements

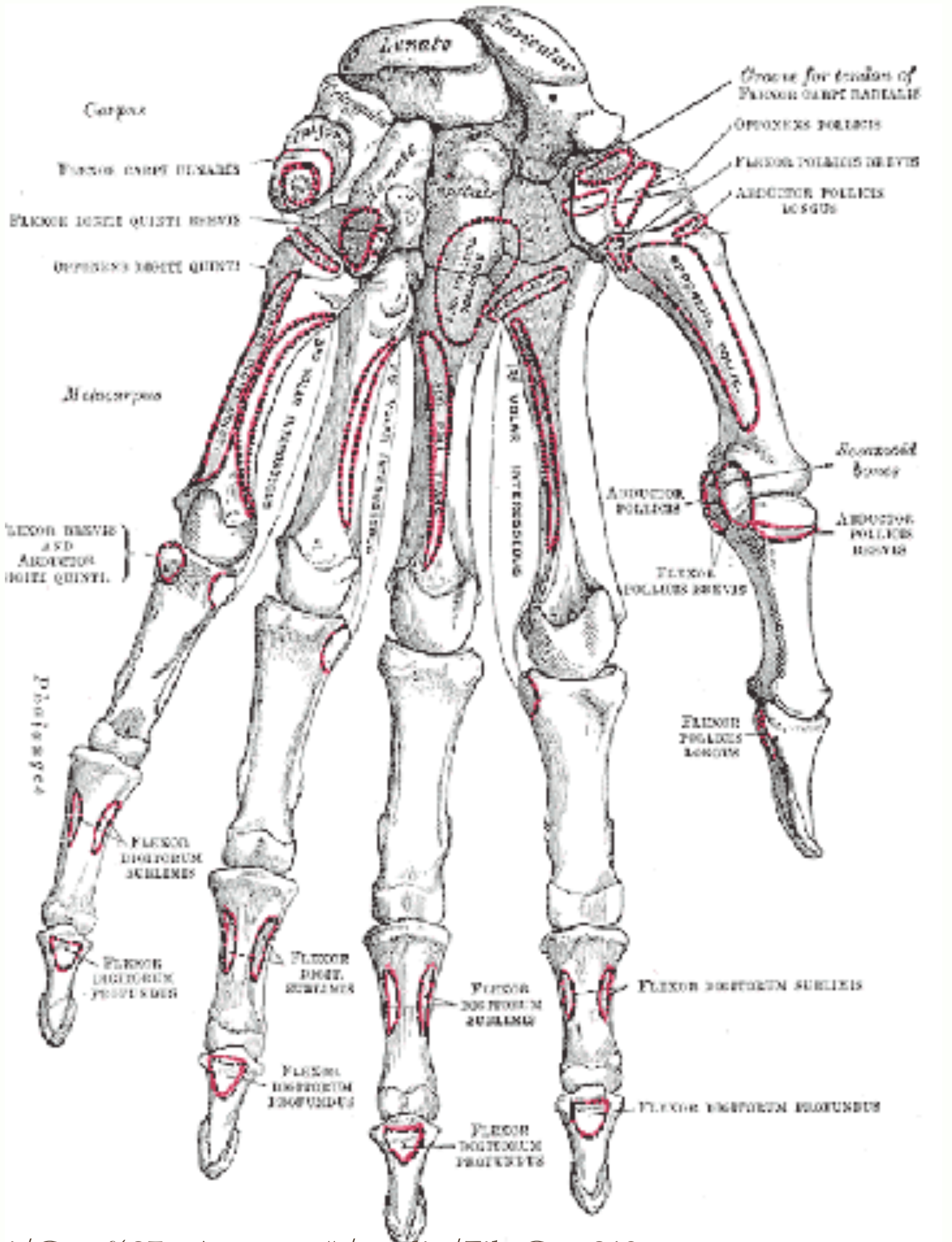
- Sprint 1 - constructive criticism
  - So that your comments can be directly imported to another project
- Thursday
  - Canceling office hours: Caius (Thu: 3-4)
  - Quiz
    - Open all day - starts at 1:00 and ends 23:59
    - 15 min long
    - On Canvas (already published)

Class expectations

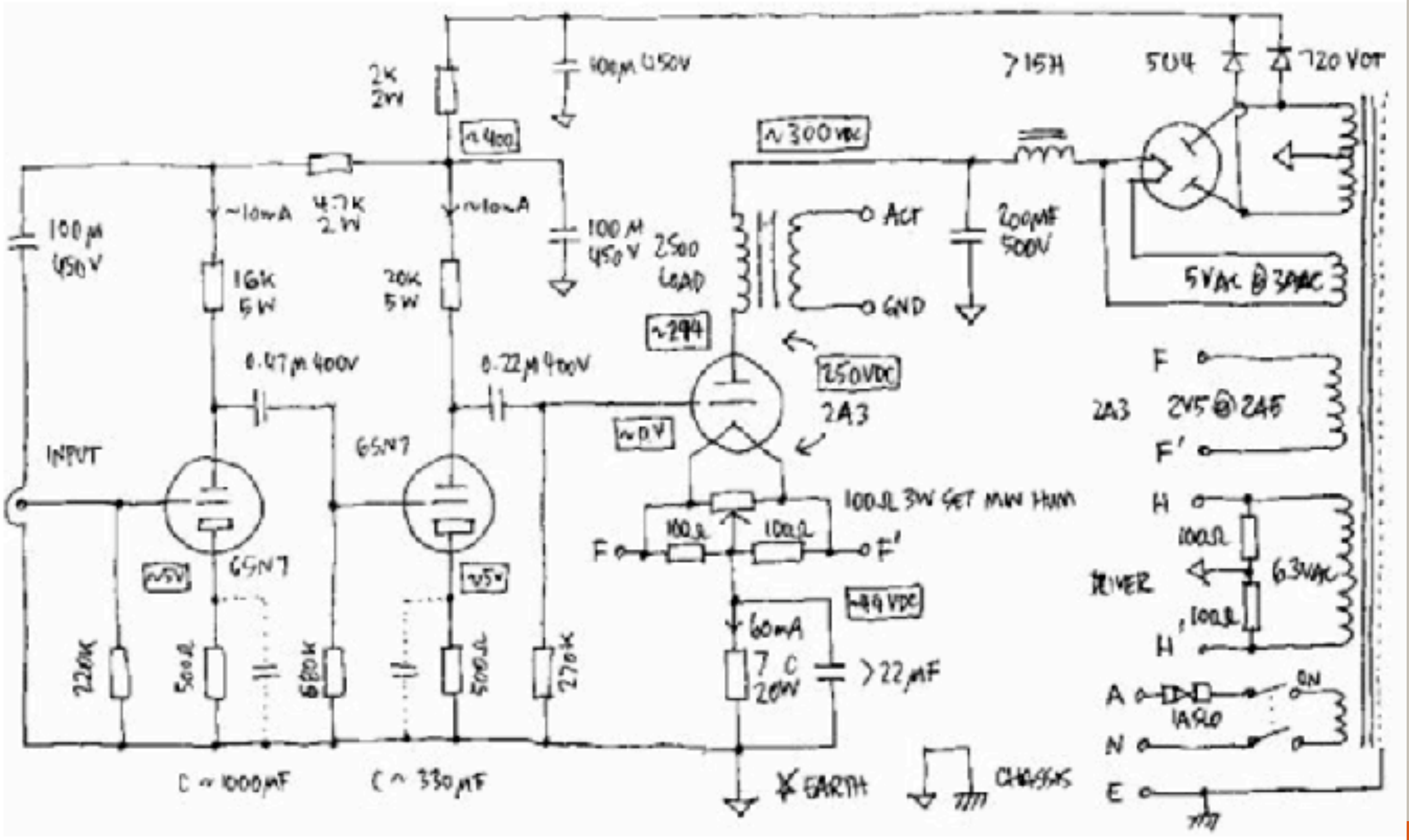


Software Diagramming

Useful when you need to:
Communicate,
Visualize, or
Analyze
Something, especially something with some structure

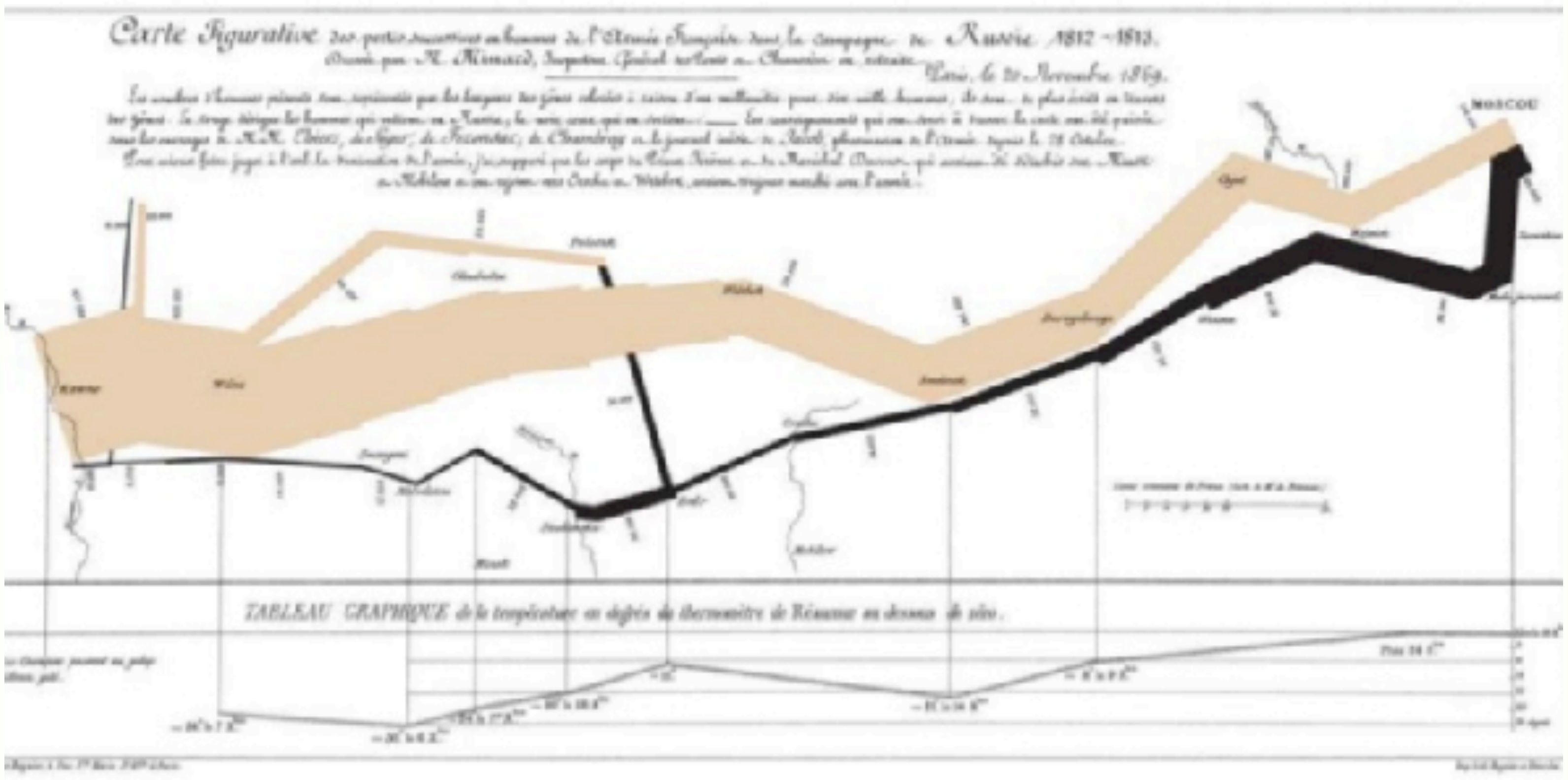

Lunate
Navicular
Carpus
Groove for tendon of Flexor carpi radialis
Opponens pollicis
Flexor pollicis brevis
Abductor pollicis longus
Flexor carpi ulnaris
Flexor digiti quinti brevis
Opponens digiti quinti
Metacarpus
Sesamoid bones
Adductor pollicis
Abductor pollicis brevis
Flexor brevis and Abductor digiti quinti
Flexor pollicis brevis
Phalanges
Flexor pollicis longus
Flexor digitorum sublimis
Flexor digitorum profundus
Flexor digit. sublimis
Flexor digitorum profundus
Flexor digitorum sublimis
Flexor digitorum profundus
Flexor digitorum sublimis
Flexor digitorum profundus

https://en.wikipedia.org/wiki/Gray%27s_Anatomy#/media/File:Gray219.png

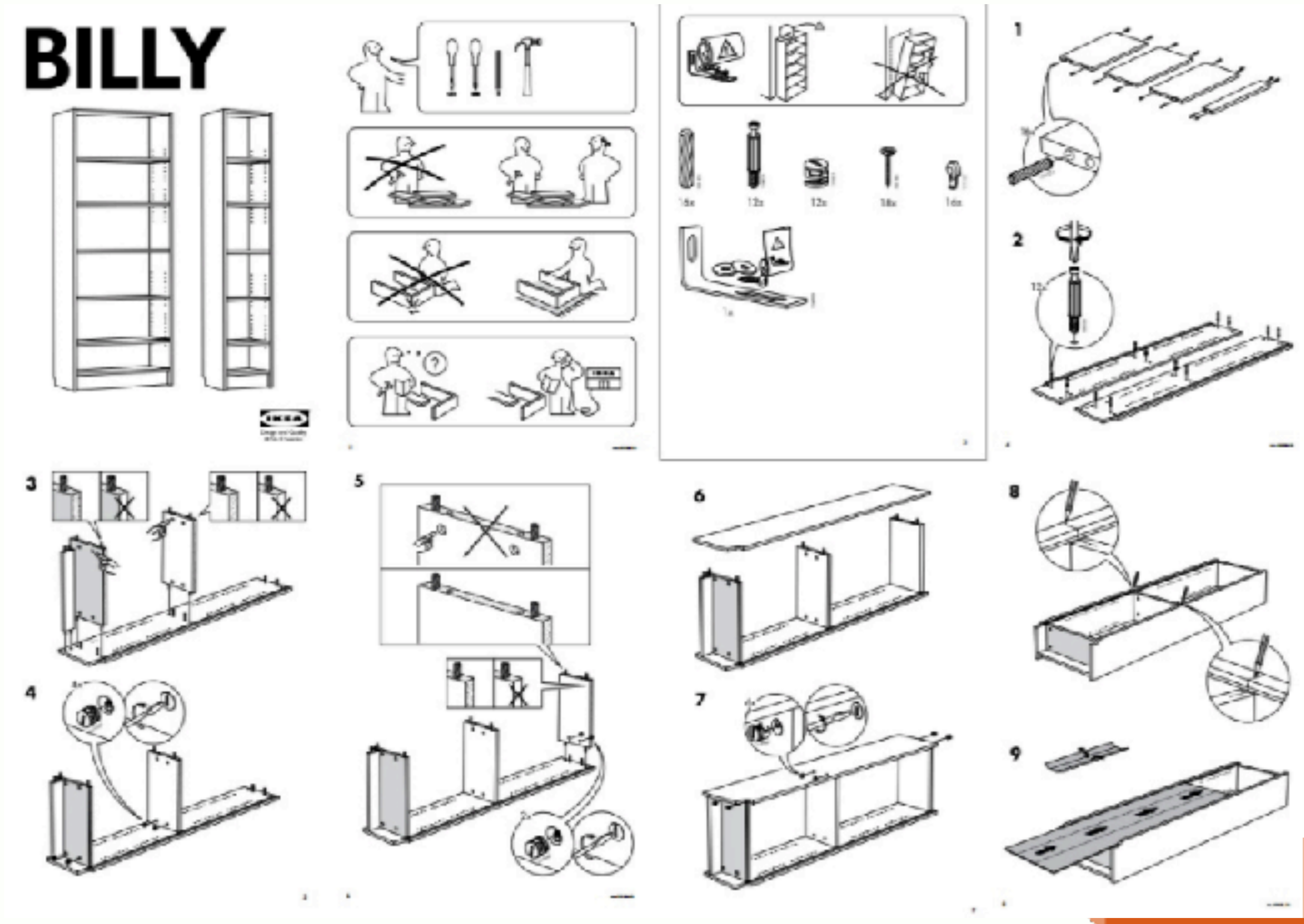


http://www.instructables.com/file/F7847TEFW4JU1UC/

Minard's Figurative Map of Napoleon's March on Russia



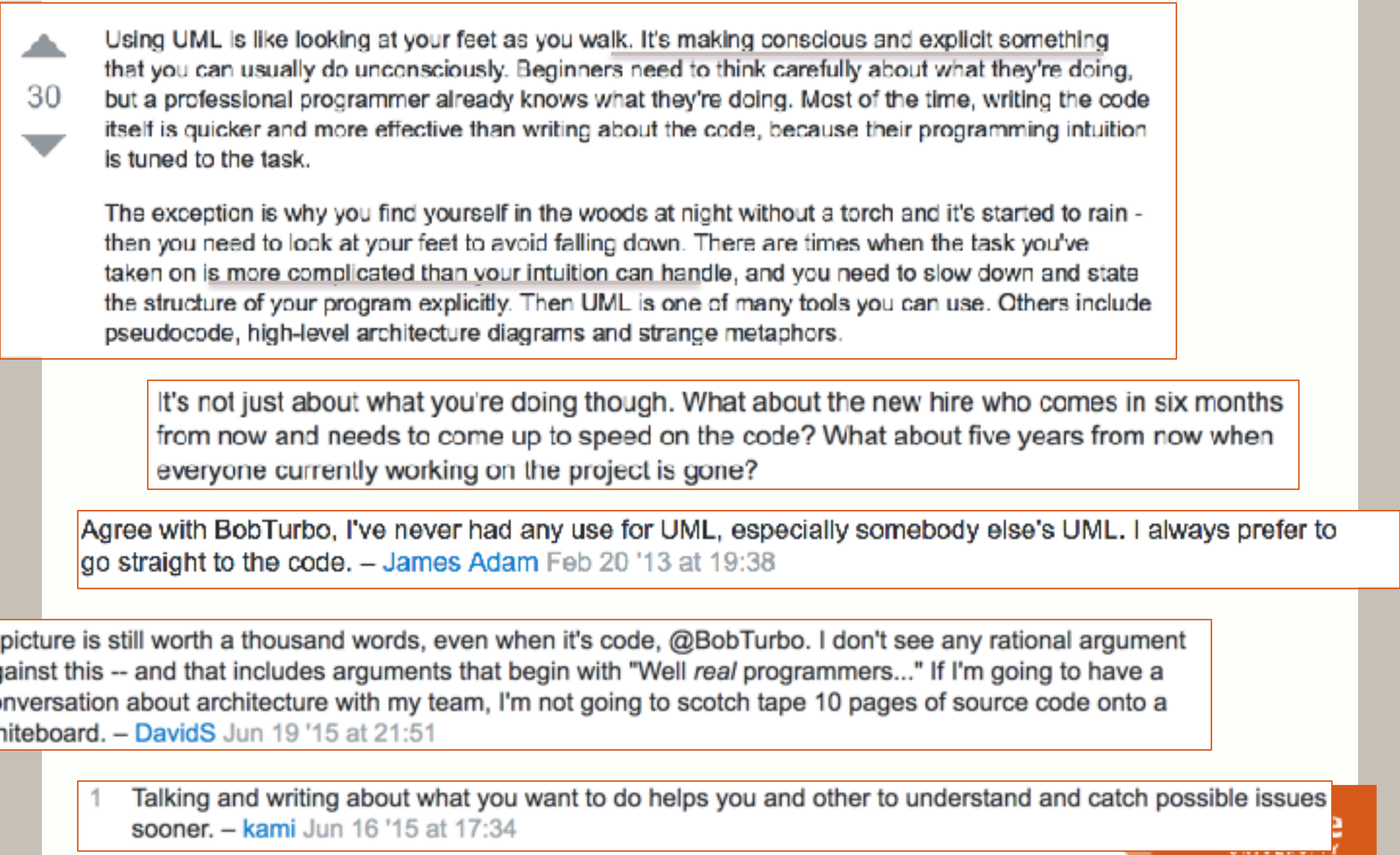
Carte Figurative des pertes successives en hommes de l'Armée Française dans la campagne de Russie 1812–1813.
Paris, le 20 Novembre 1869.
MOSCOU
TABLEAU GRAPHIQUE de la température en degrés du thermomètre de Réaumur au dessous de zéro.


BILLY
1
2
3
4
5
6
7
8
9

Unified Modeling Language - UML

A set of many visual modeling techniques …

Is UML practical? [closed]



30

Using UML is like looking at your feet as you walk. It's making conscious and explicit something that you can usually do unconsciously. Beginners need to think carefully about what they're doing, but a professional programmer already knows what they're doing. Most of the time, writing the code itself is quicker and more effective than writing about the code, because their programming intuition is tuned to the task.

The exception is why you find yourself in the woods at night without a torch and it's started to rain - then you need to look at your feet to avoid falling down. There are times when the task you've taken on is more complicated than your intuition can handle, and you need to slow down and state the structure of your program explicitly. Then UML is one of many tools you can use. Others include pseudocode, high-level architecture diagrams and strange metaphors.

It's not just about what you're doing though. What about the new hire who comes in six months from now and needs to come up to speed on the code? What about five years from now when everyone currently working on the project is gone?

Agree with BobTurbo, I've never had any use for UML, especially somebody else's UML. I always prefer to go straight to the code. – James Adam Feb 20 '13 at 19:38

picture is still worth a thousand words, even when it's code, @BobTurbo. I don't see any rational argument gainst this -- and that includes arguments that begin with "Well *real* programmers..." If I'm going to have a nversation about architecture with my team, I'm not going to scotch tape 10 pages of source code onto a hiteboard. – DavidS Jun 19 '15 at 21:51

1 Talking and writing about what you want to do helps you and other to understand and catch possible issues sooner. – kami Jun 16 '15 at 17:34

UML – what is it good for?

- Forces you to stop and think about design
- Get a high-level picture of the design
  - Better understand, find problems
- Communication tool
- Vocabulary

- Teaching tool

Marian Petre: ML in practice. ICSE 2013: 722-731

Some UML diagrams

- Activity Diagram
- Class Diagram
- Communication Diagram
- Component Diagram
- Composite Structure Diagram
- Deployment Diagram
- Interaction Overview Diagram
- Object Diagram
- Package Diagram
- Sequence Diagram
- State Machine Diagram
- Timing Diagram
- Use Case Diagram

http://www.agilemodeling.com/essays/umlDiagrams.htm

Oregon State UNIVERSITY

Some UML diagrams

- **Activity Diagram**
- **Class Diagram**
- Communication Diagram
- Component Diagram
- Composite Structure Diagram
- Deployment Diagram
- Interaction Overview Diagram
- Object Diagram
- Package Diagram
- **Sequence Diagram**
- **State Machine Diagram**
- Timing Diagram
- **Use Case Diagram**

Classes of UML diagrams

- Behavior
  - Depicts the behavioral features of the system or process
  - activity, sequence, state machine, use case
- Structure
  - Depicts the elements of a specification irrespective of time
  - Class diagram

Activity Diagram

Used to model business process, or a single usage scenario, or a business rule

Example:

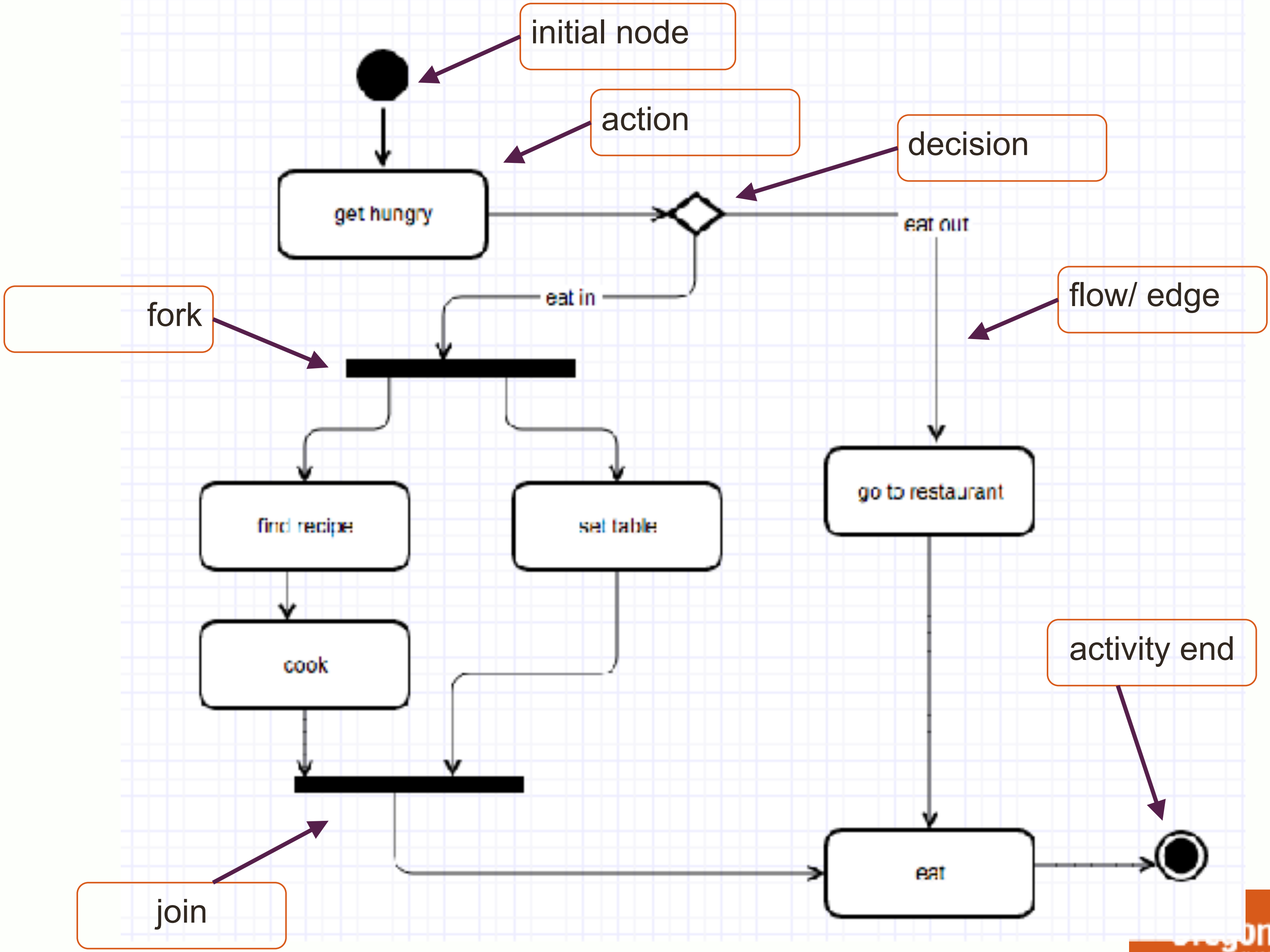
- Online Shopping
- Purchasing Ticket from vending machine
- Reserving a Flight

Activity Diagram

- Graphical representations of activities or workflow
- Different shapes have different meanings
- Flow goes from start to the end

Activity Diagram Parts

- Black circle represents the start
- Rounded rectangle represents actions
- Diamonds represent decisions
- Black Bars Represent concurrent activities
- Optional: Partition diagram with lines


initial node
action
decision
get hungry
eat out
flow/ edge
fork
eat in
go to restaurant
find recipe
set table
cook
activity end
eat
join

Exercise

- Get the SE recommended text book
  - From amazon.com

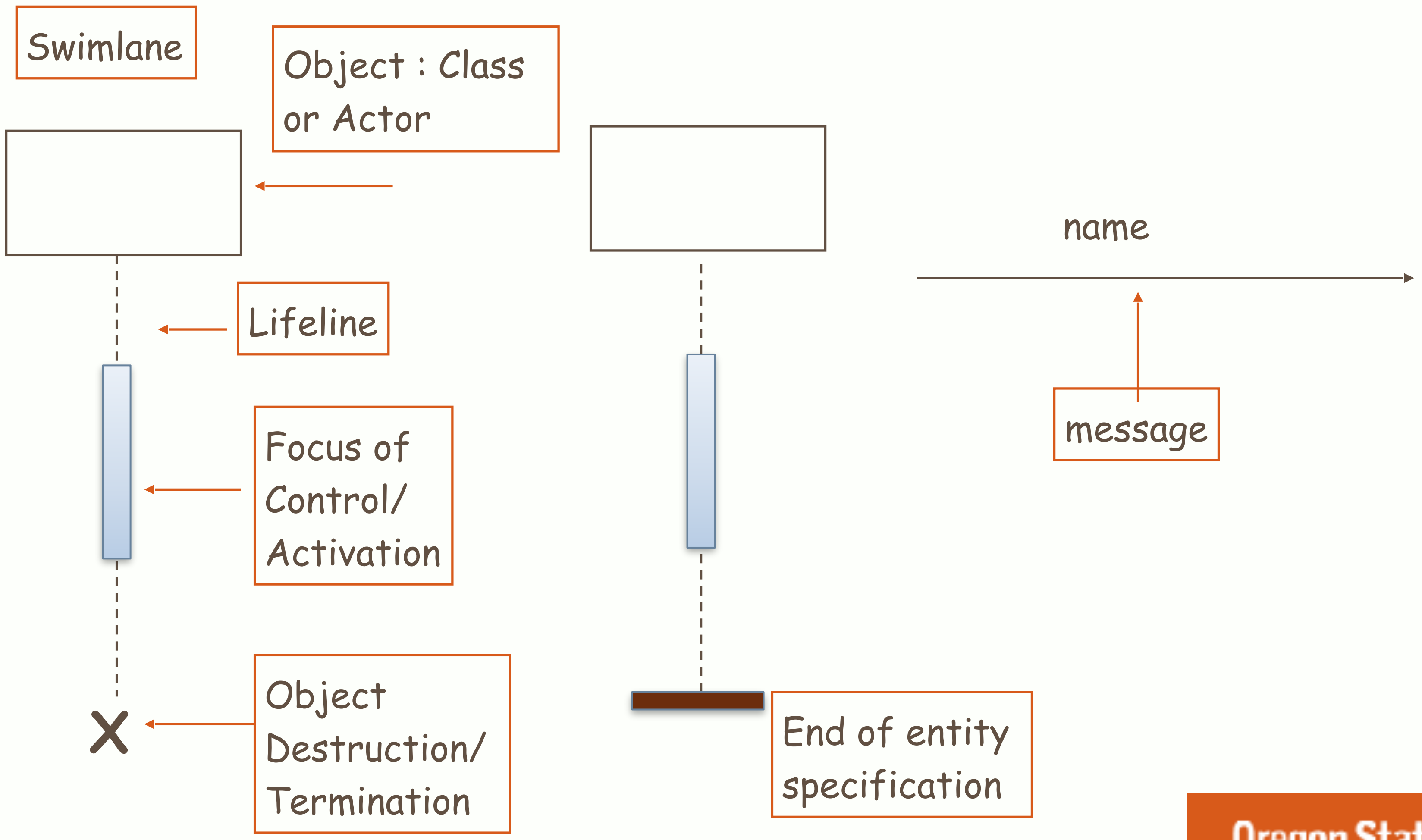
Sequence Diagram

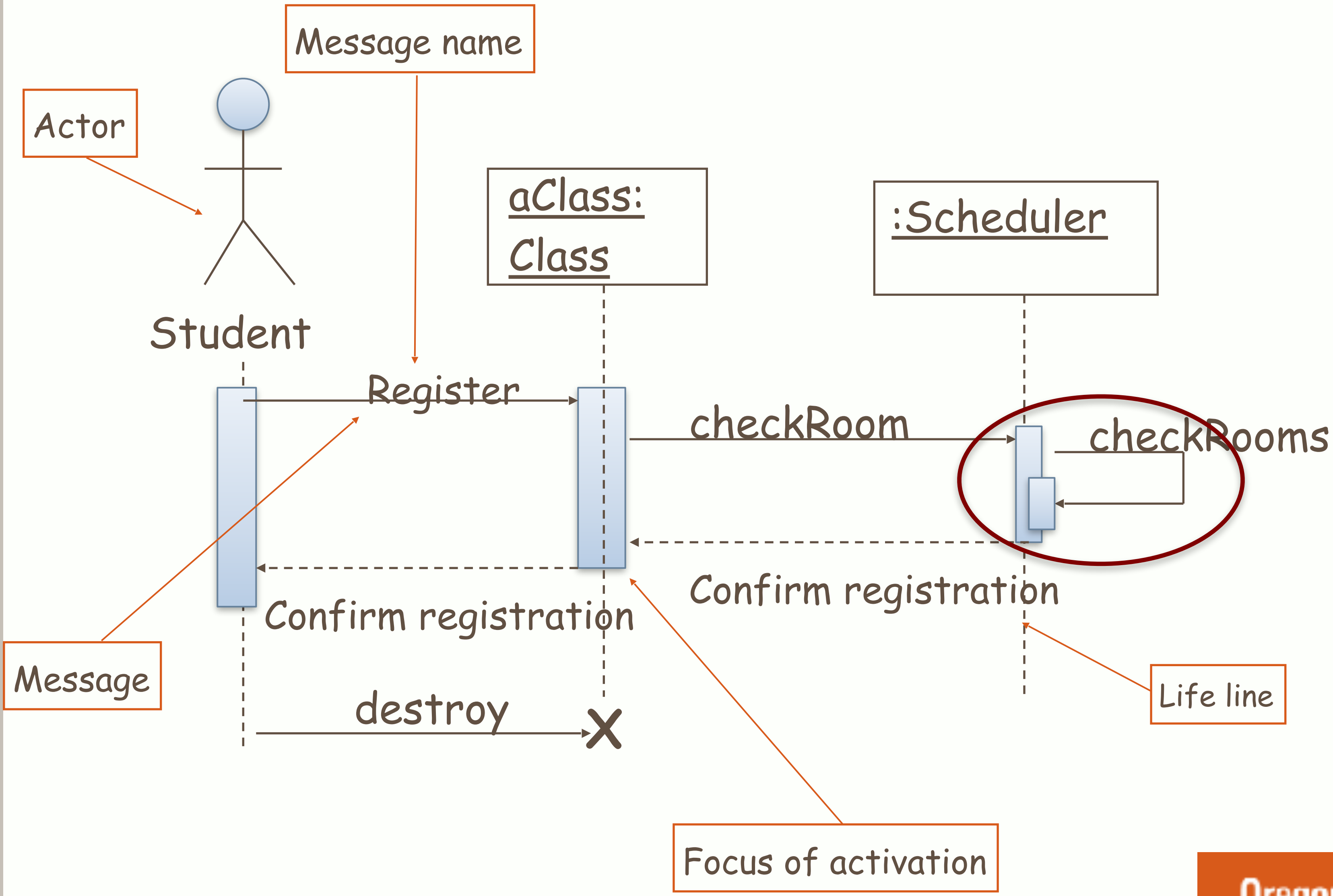
- A Sequence Diagram is an interaction diagram that shows how processes operate with one another and in what order
- Typically Model: Usage scenarios, Logic of methods, the logic of services
- Helpful for understanding asynchronous code
- Examples
  - Submitting comments on a website
  - Facebook user authentication

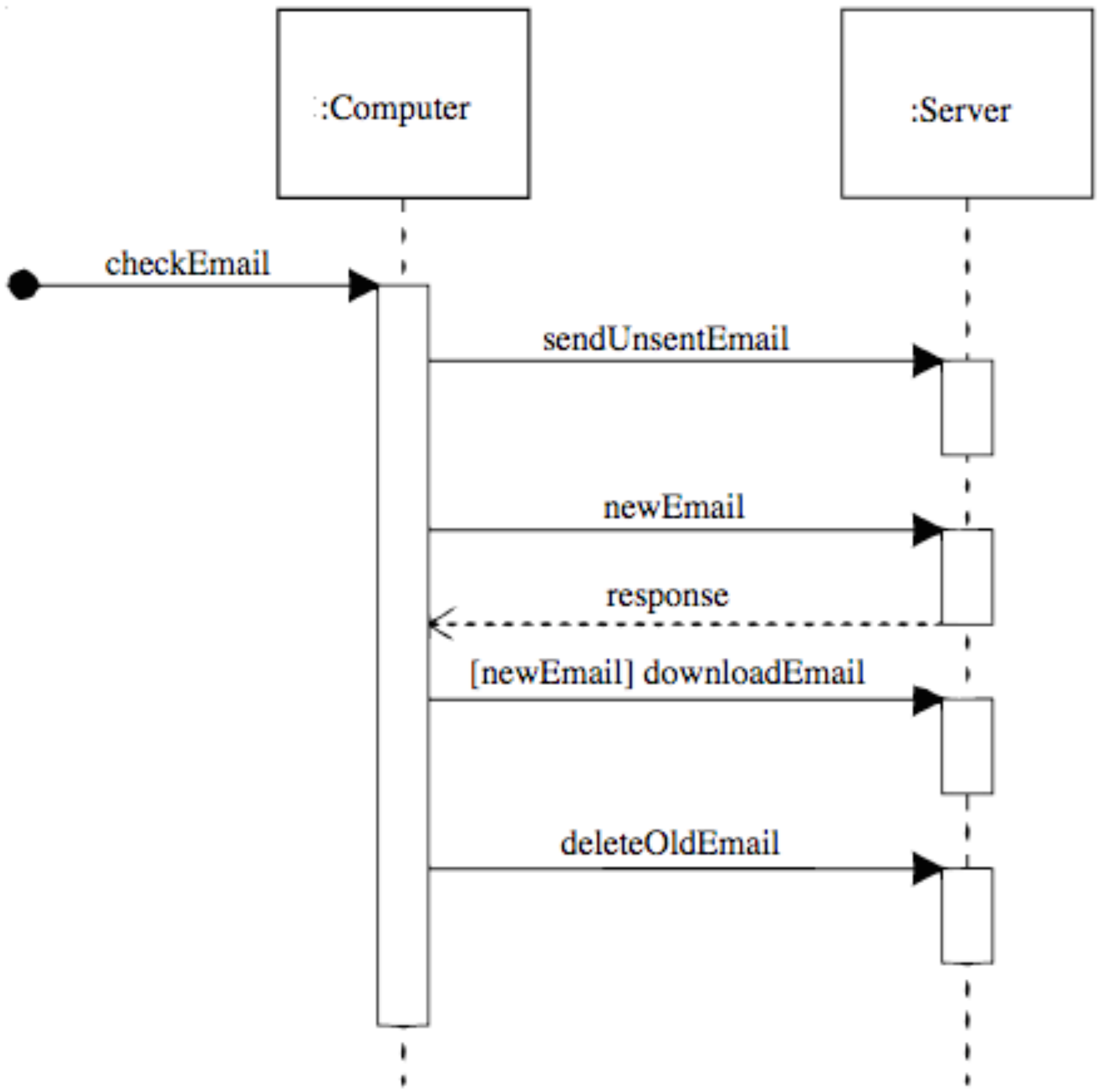
Sequence Diagram Parts

- Each actor is represented as a labeled vertical line
- Each message is a horizontal line, with message name written above line
- Open arrow heads represent async messages
- Dashed lines are responses

Sequence Diagrams components


Swimlane
Object : Class or Actor
name
Lifeline
message
Focus of Control/ Activation
Object Destruction/ Termination
End of entity specification


Message name
Actor
aClass: Class
:Scheduler
Student
Register
checkRoom
checkRooms
Confirm registration
Confirm registration
Message
destroy
Life line
Focus of activation


:Computer
:Server
checkEmail
sendUnsentEmail
newEmail
response
[newEmail] downloadEmail
deleteOldEmail

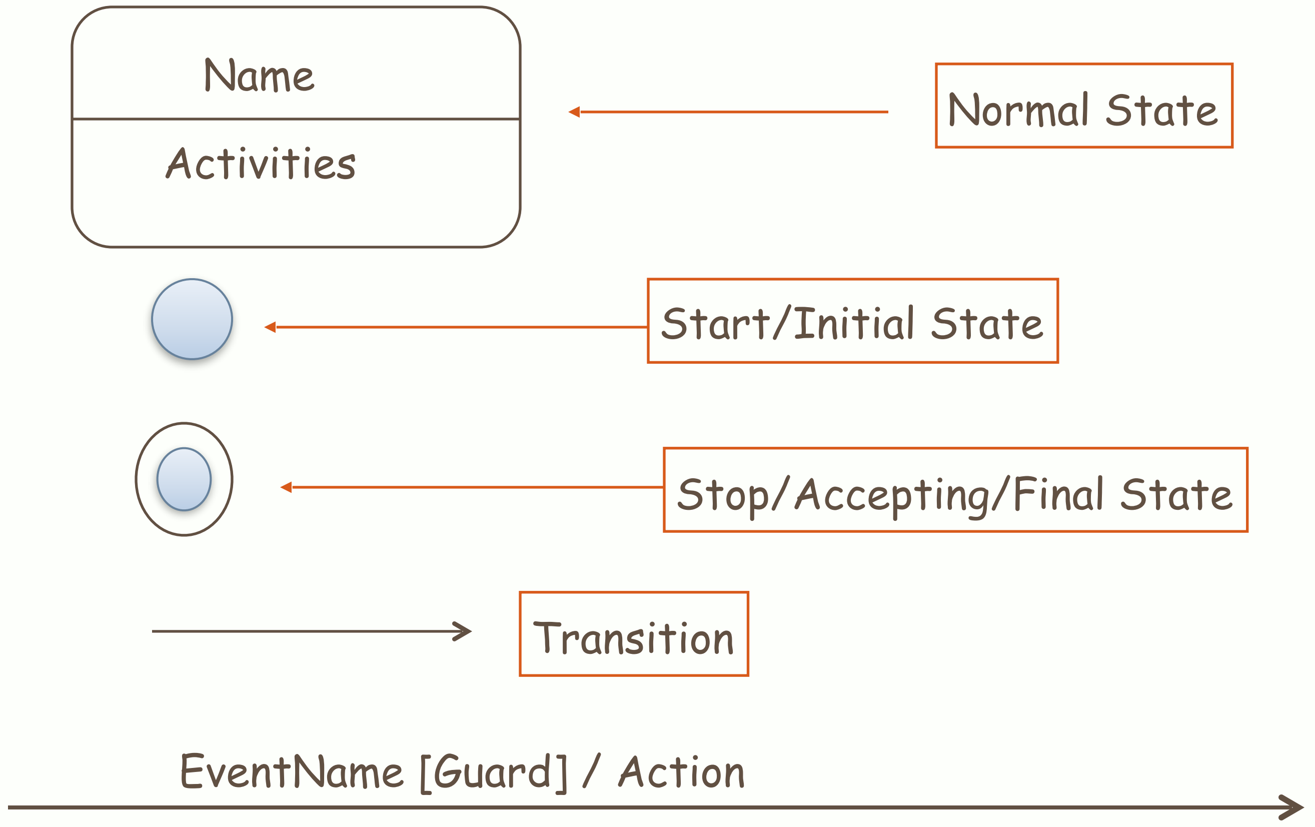
Exercise

- Buying a book in Amazon

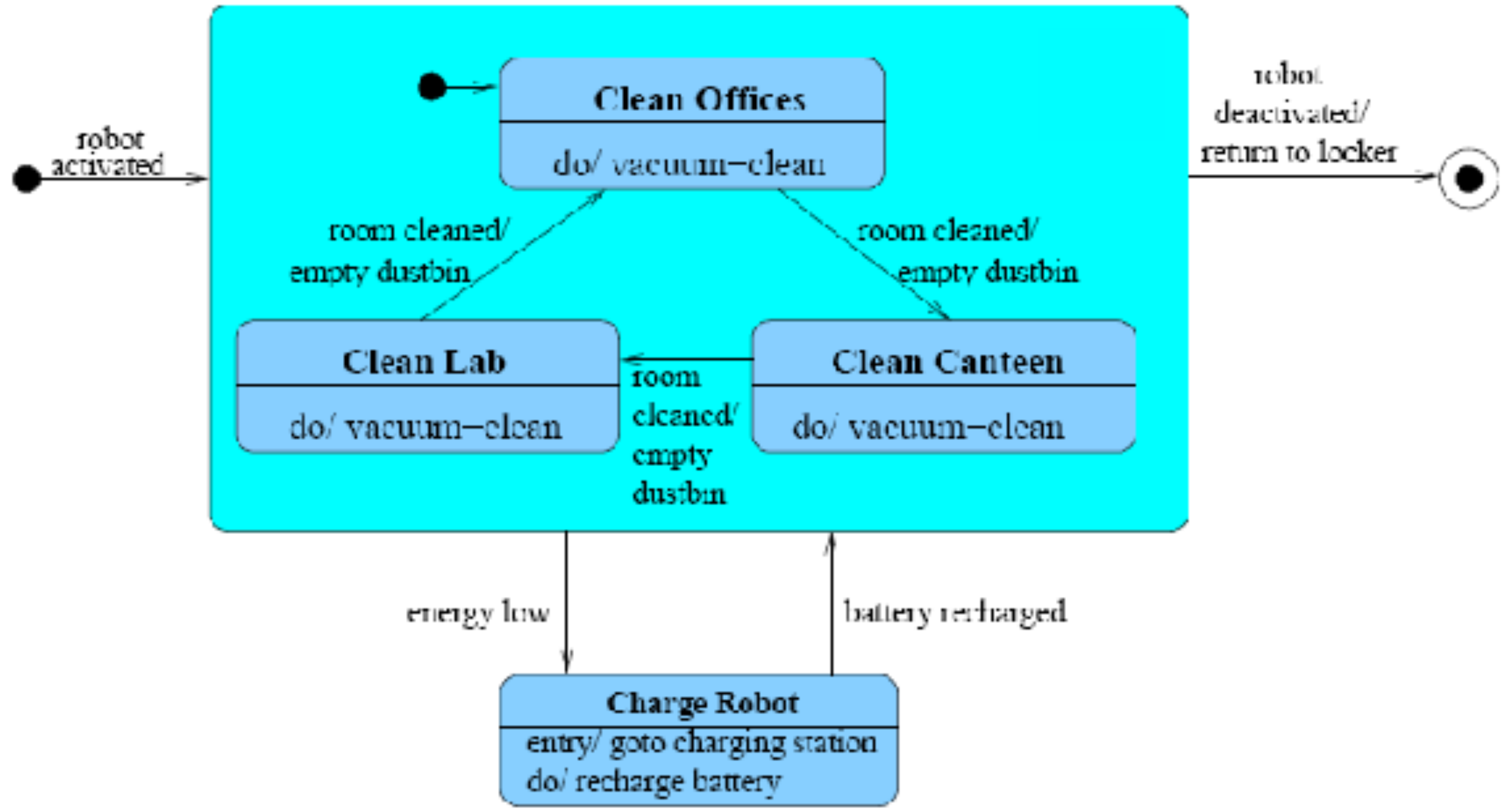
UML State Diagram

- A state diagram shows the states of an object. Similar to a other State Diagrams, e.g. State Machine
- Examples:
  - State of phone line
  - Elevator movement

State Diagram Parts



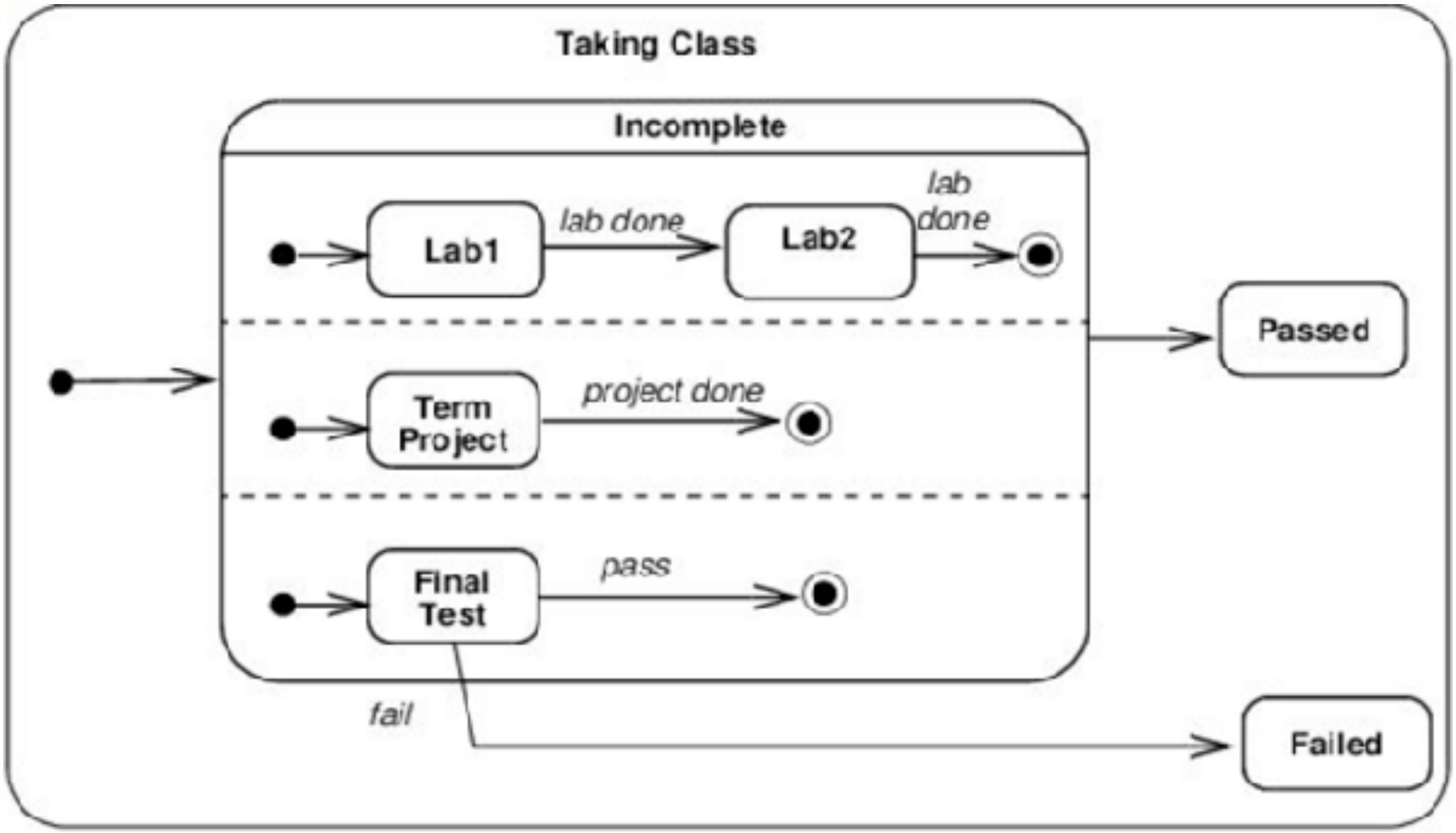
Cleaning Robot



Actions vs. Activities

- Actions are associated with transitions, are considered to be processes that occur quickly and are not interruptible
- Activities are associated with states, can take longer, and can be interrupted by events
  - "do" events can iterate
  - "entry" events happen only on entry to state

Example


Taking Class
Incomplete
Lab1
lab done
Lab2
lab done
Passed
Term Project
project done
Final Test
pass
fail
Failed

Exercise

- Buying book from Amazon (searching for books & reviews, process transaction)