UML Diagrams

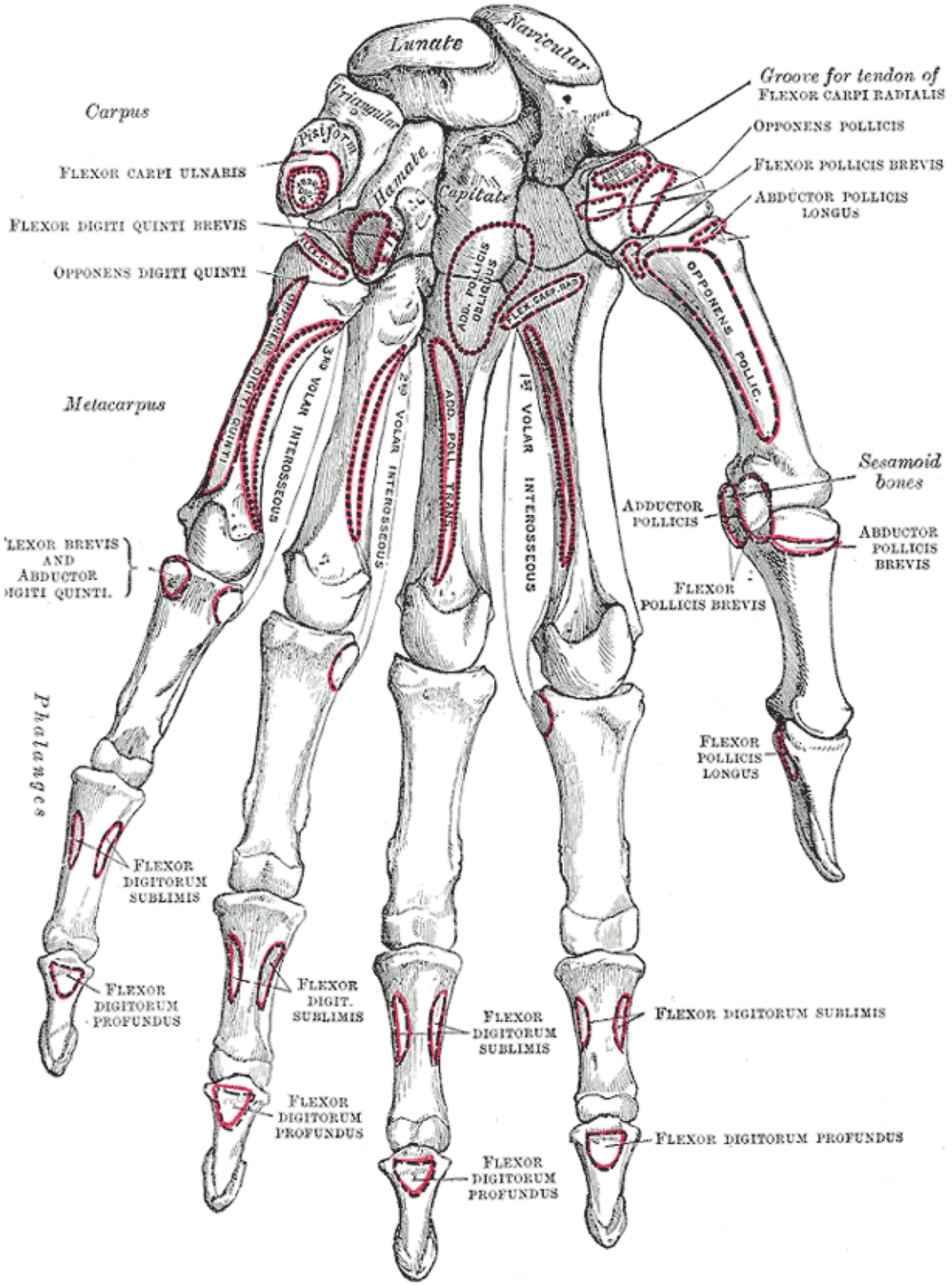
Tuesday, October 30

Announcements

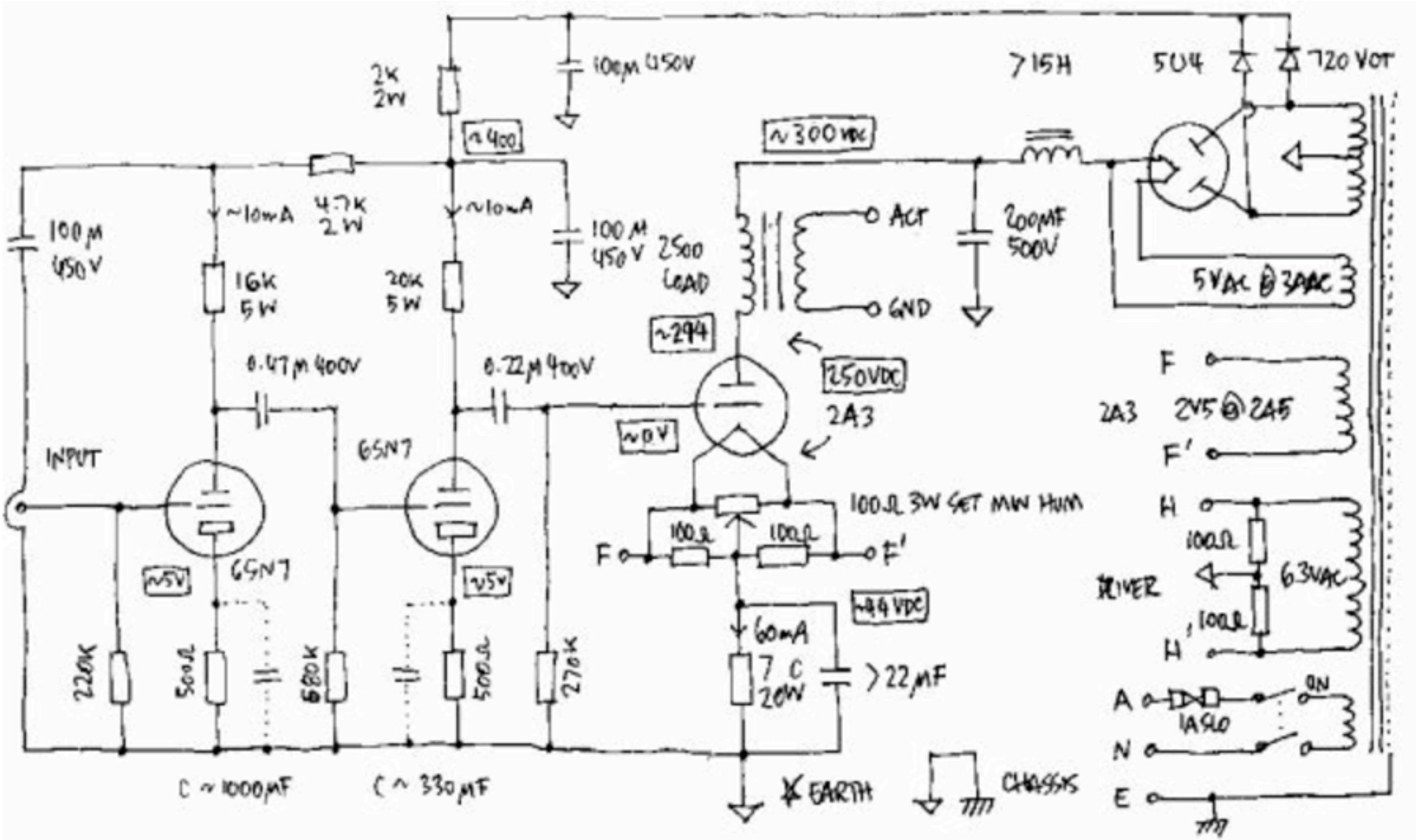
Sprint 3 overview

Software Diagramming

Useful when you need to **communicate, visualize, analyze** something, especially something with some structure



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

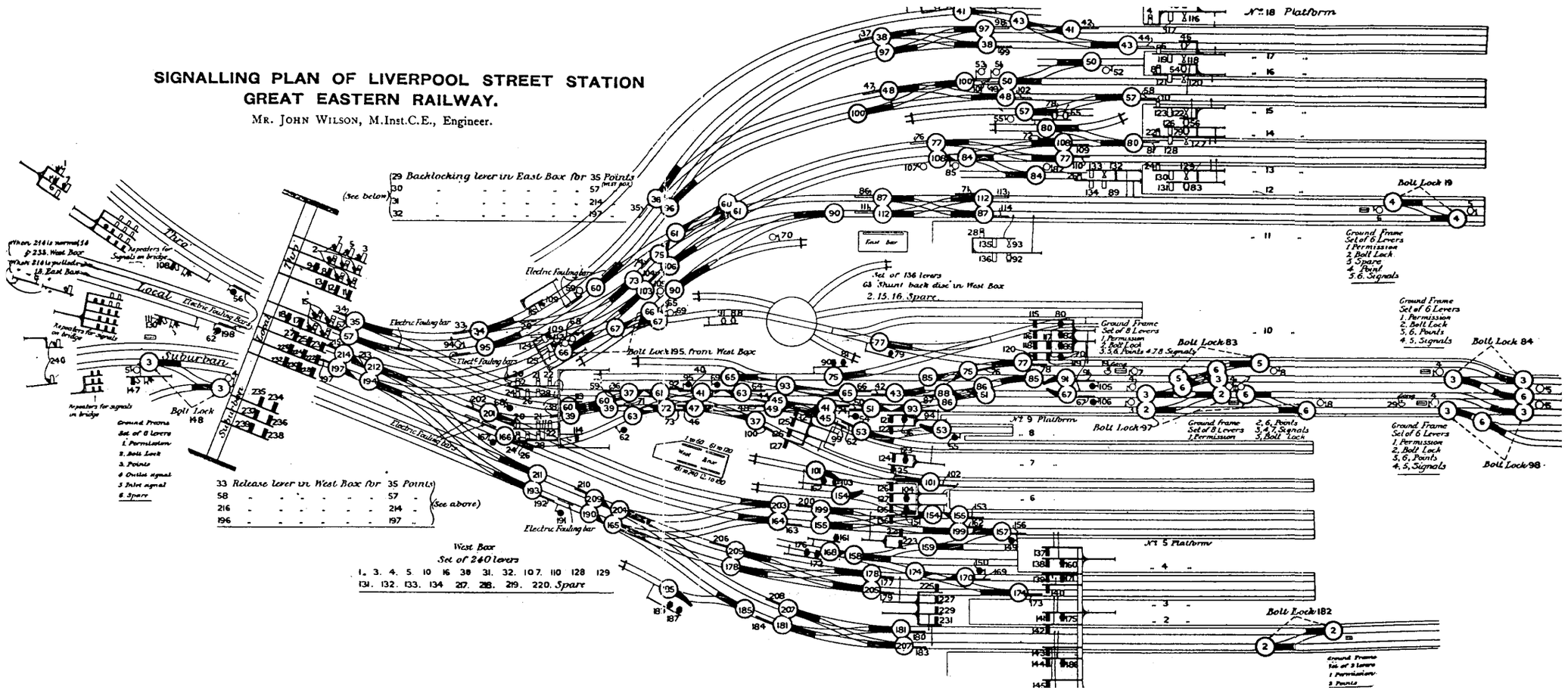


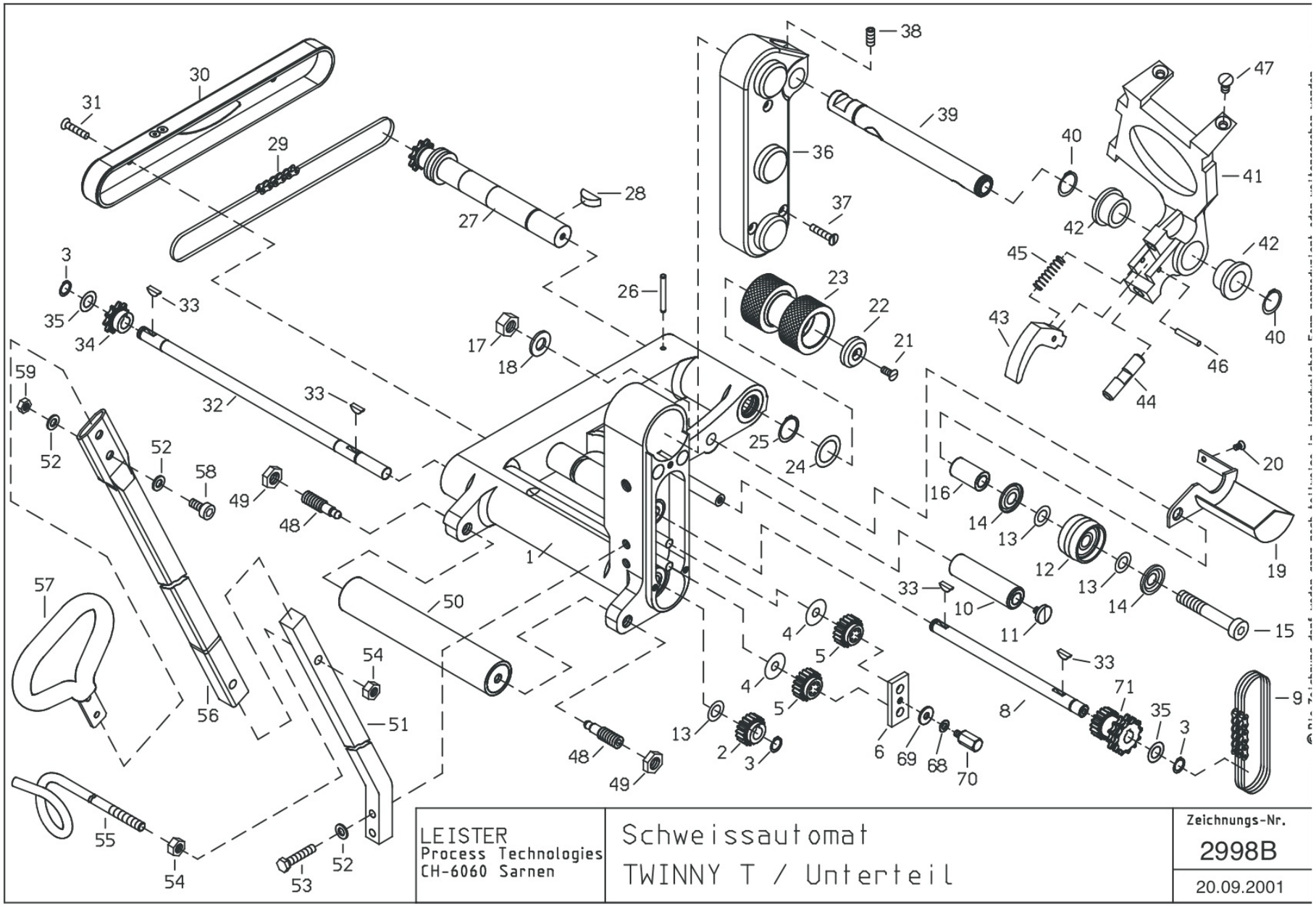
http://www.instructables.com/file/F7847TEFW4JU1UC/

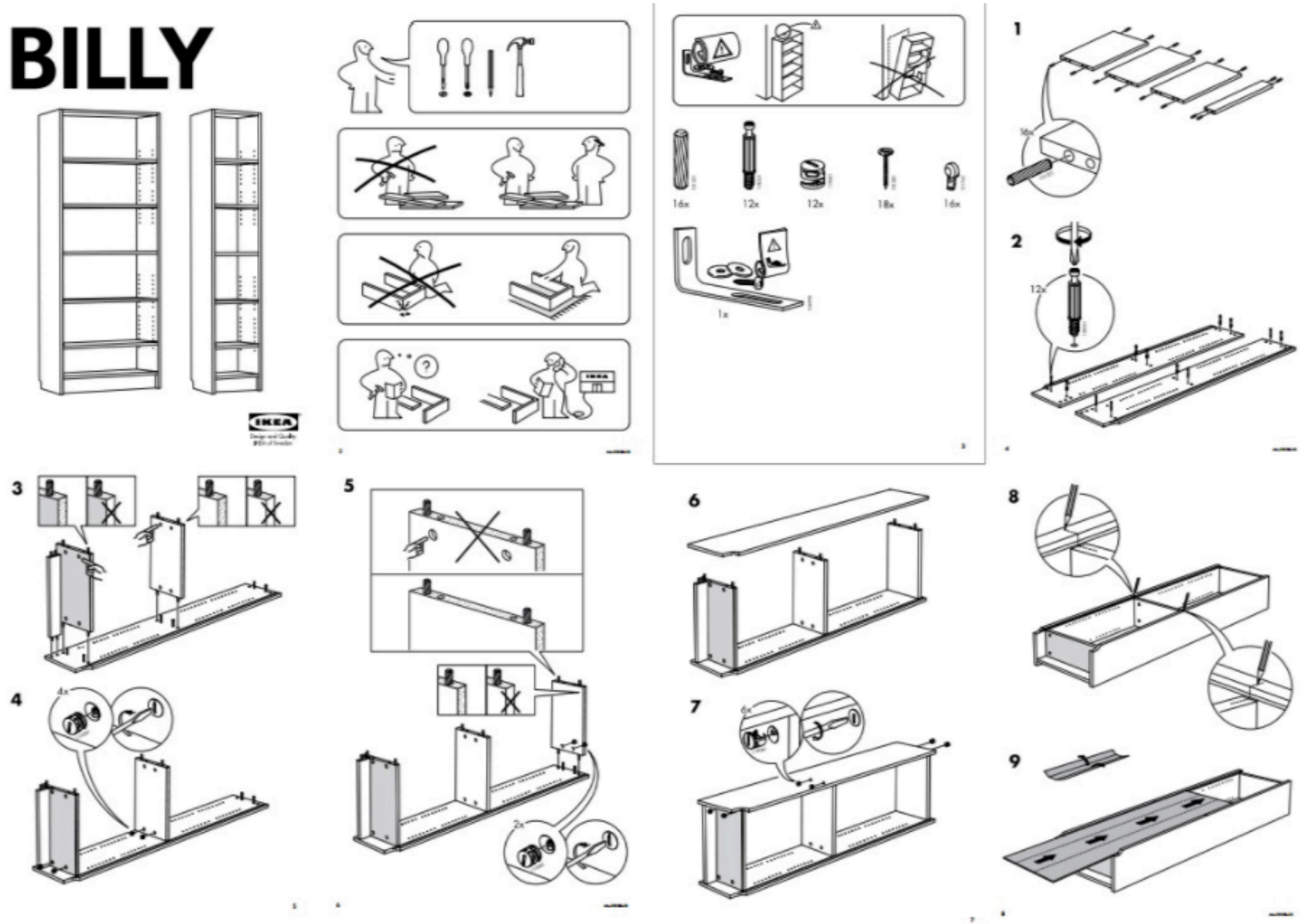
Oregon State University

SIGNALLING PLAN OF LIVERPOOL STREET STATION
GREAT EASTERN RAILWAY.

MR. JOHN WILSON, M.Inst.C.E., Engineer.







Oregon State University

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

> A picture is still worth a thousand words, even when it's code, @BobTurbo. I don't see any rational argument against this -- and that includes arguments that begin with "Well *real* programmers..." If I'm going to have a conversation about architecture with my team, I'm not going to scotch tape 10 pages of source code onto a whiteboard. – 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

Classes of UML diagrams

Behavior

Depicts the behavioral features of the system or process

Activity, sequence, state machine diagrams

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

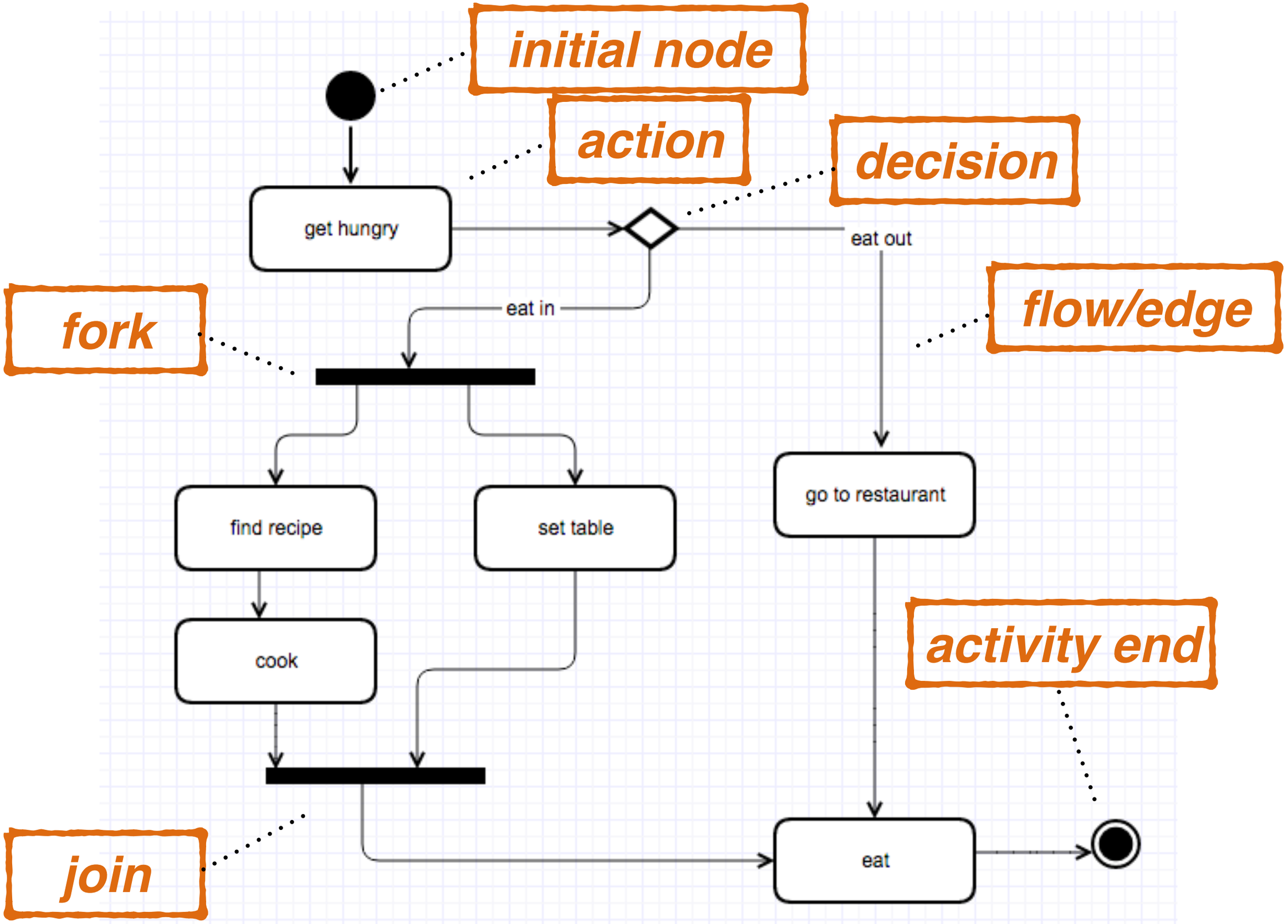
Black circle represents the start

Rounded rectangle represents actions

Diamonds represent decisions

Black Bars represent concurrent activities

Optional: Partition diagram with lines



Exercise

Buy the "UML Distilled" 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

They 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

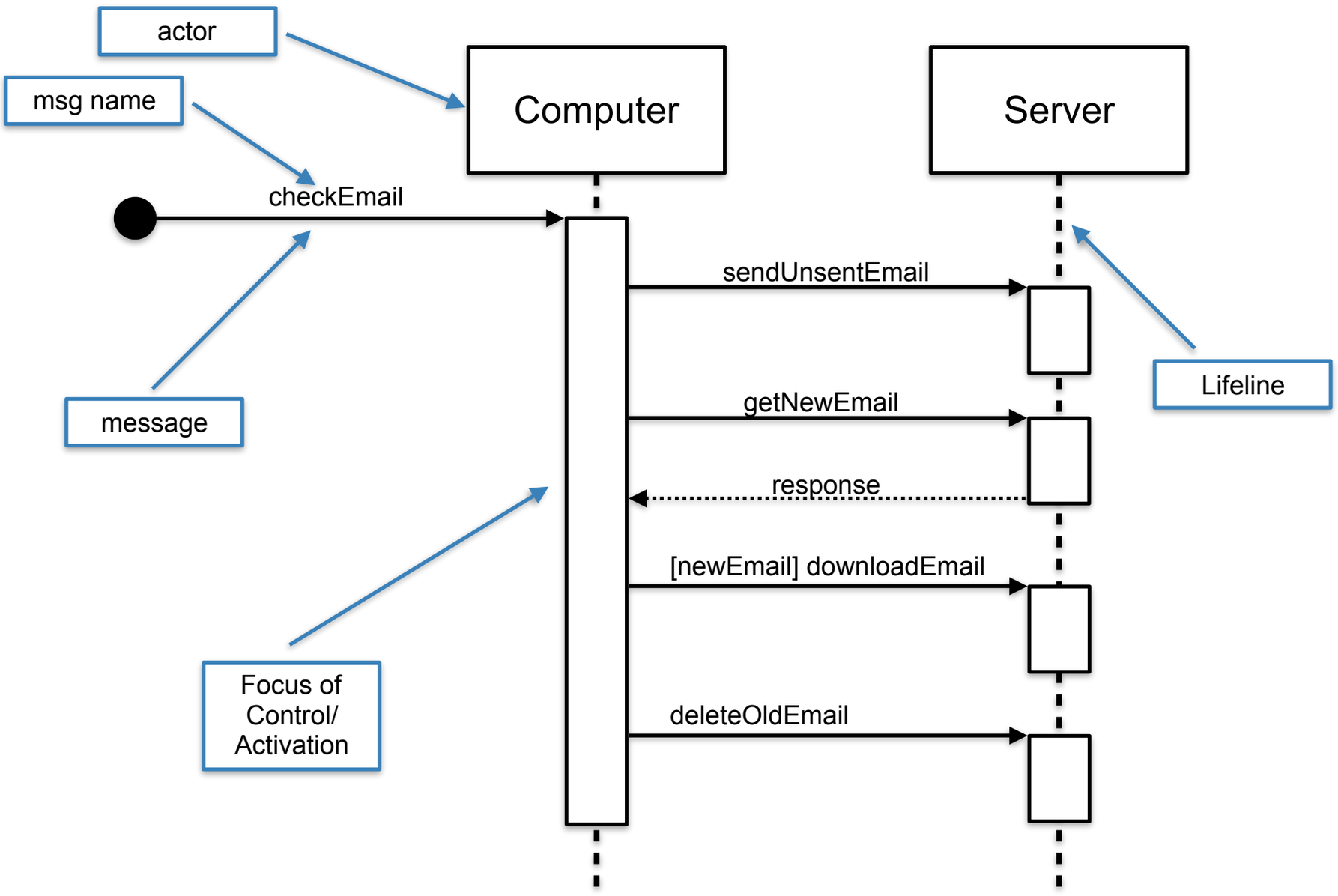
Each actor is represented as a **labeled vertical line**

Each message is a **horizontal line,** with message name written above line

Open arrow heads represent async messages

Dashed lines are responses

Sequence Diagrams



https://en.wikipedia.org/wiki/Sequence_diagram

Exercise

Buy the "UML Distilled" text book from amazon.com

UML State Diagram

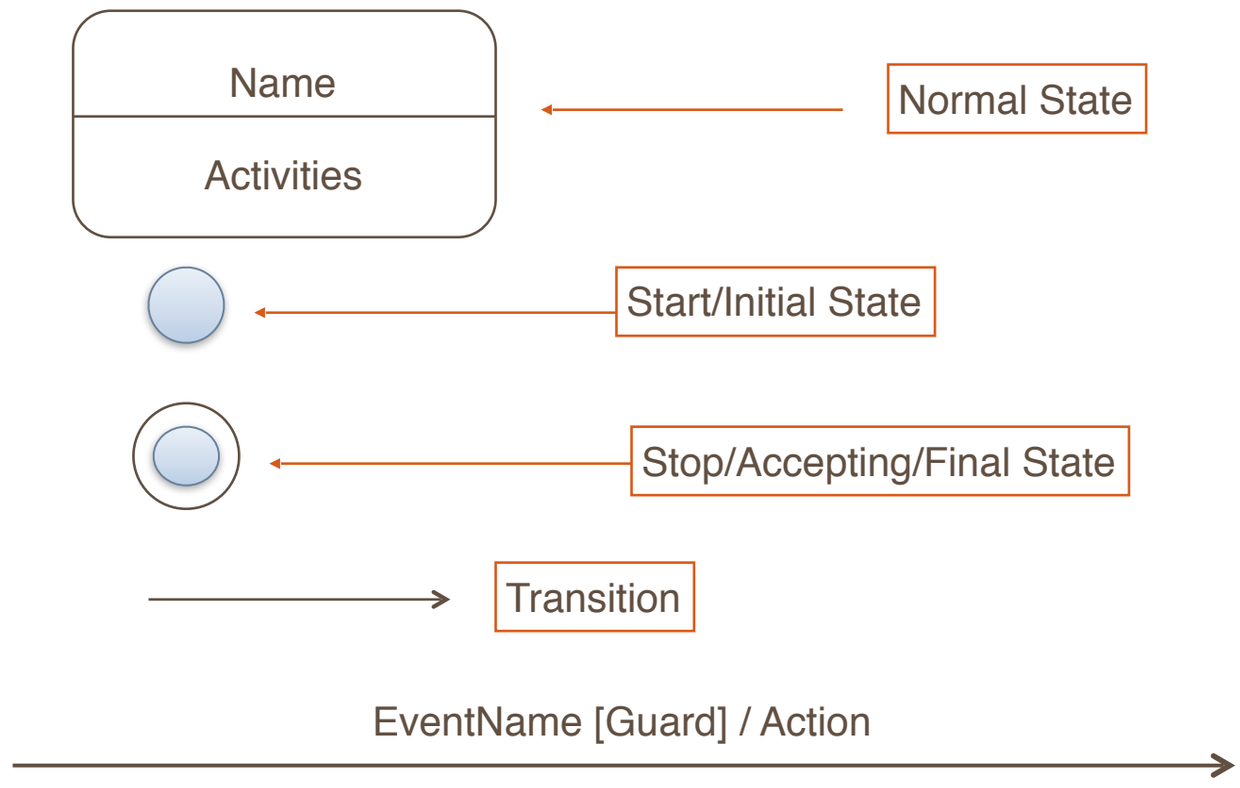
A state diagram shows the states of an object. Similar to a other State Diagrams, e.g. State Machine

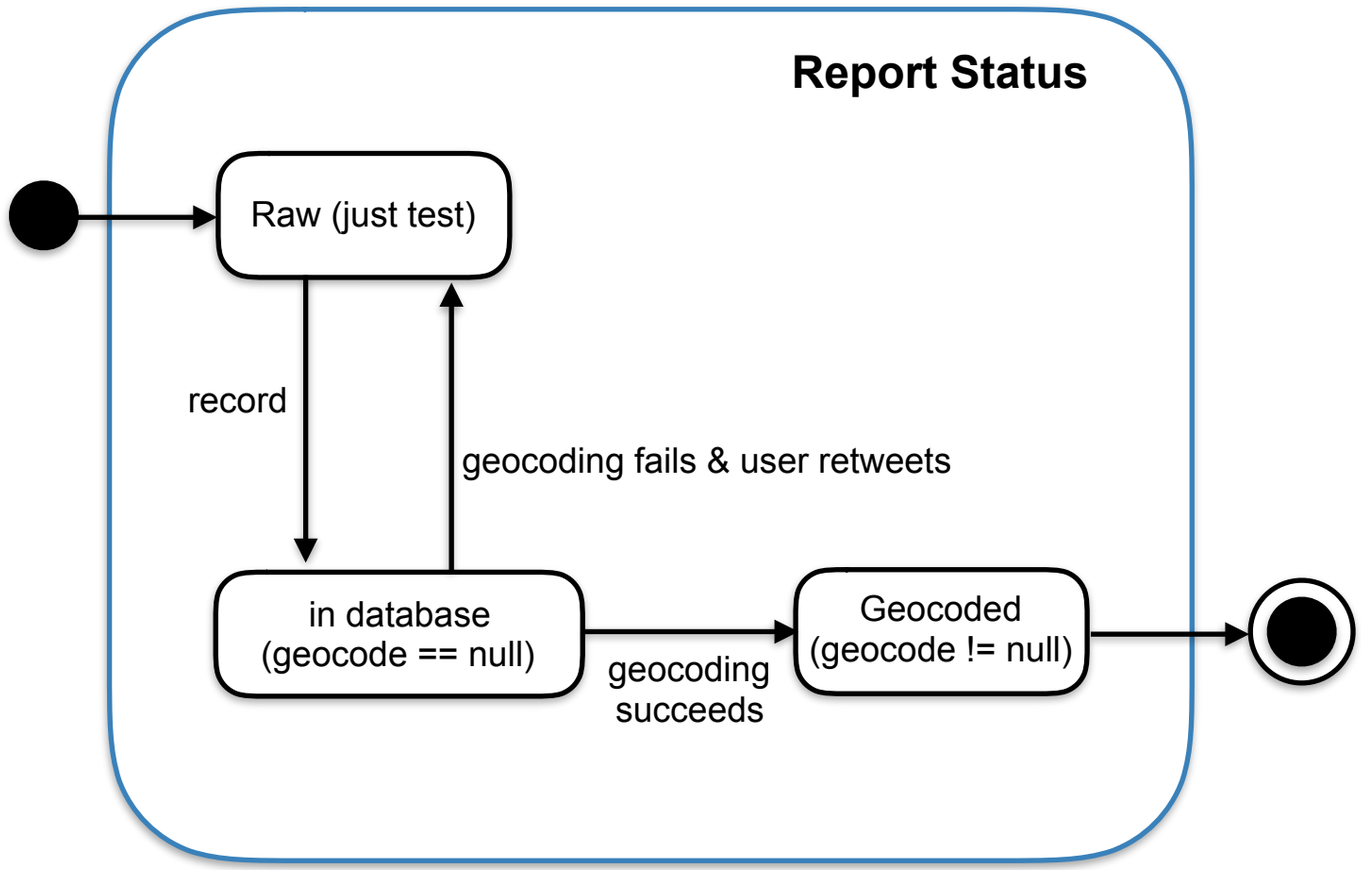
Examples:

State of phone line

Elevator movement

State Diagram Parts





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

Use Case Diagrams

Use Case Diagram at its simplest is a representation of a user's interaction with a system.

Use Cases similar to User Stories, but more formal and more complex

Use Case

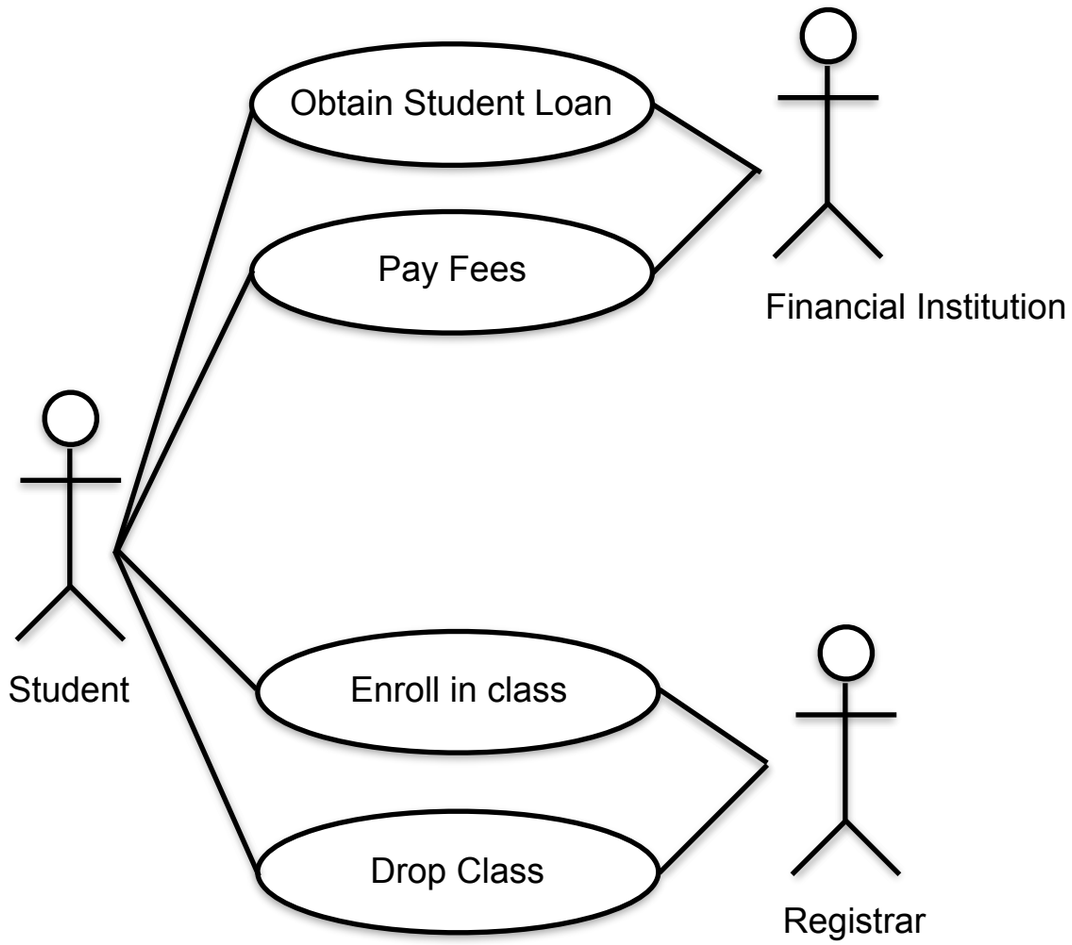
Include:

Summary of usage requirements, **from users point of view**

Basic Course of Events

Alternative Paths

Preconditions / Postconditions



http://agilemodeling.com/artifacts/useCaseDiagram.htm

Class Diagrams

Object Terminology

An object mirrors real world entity

Examples:

Person, student, book, card, game, etc.

Object Terminology

Objects Contain (class):

attributes (variables)

functionality (methods)

Objects can have properties or be acted upon

Encapsulation

Objects allow data and functionality to be **bundled together.**

Additionally, access to the data may be **restricted** to some of the objects components

Polymorphism

The ability to send the same message (call a method) to an Object, without knowing how the receiver (Object) will implement the message.

Building an Object Oriented Model

Our model should:

represent **entities**

show **connections** and **interactions**

show enough detail to **evaluate** designs

Example class:

employee:

has a name,

employee#,

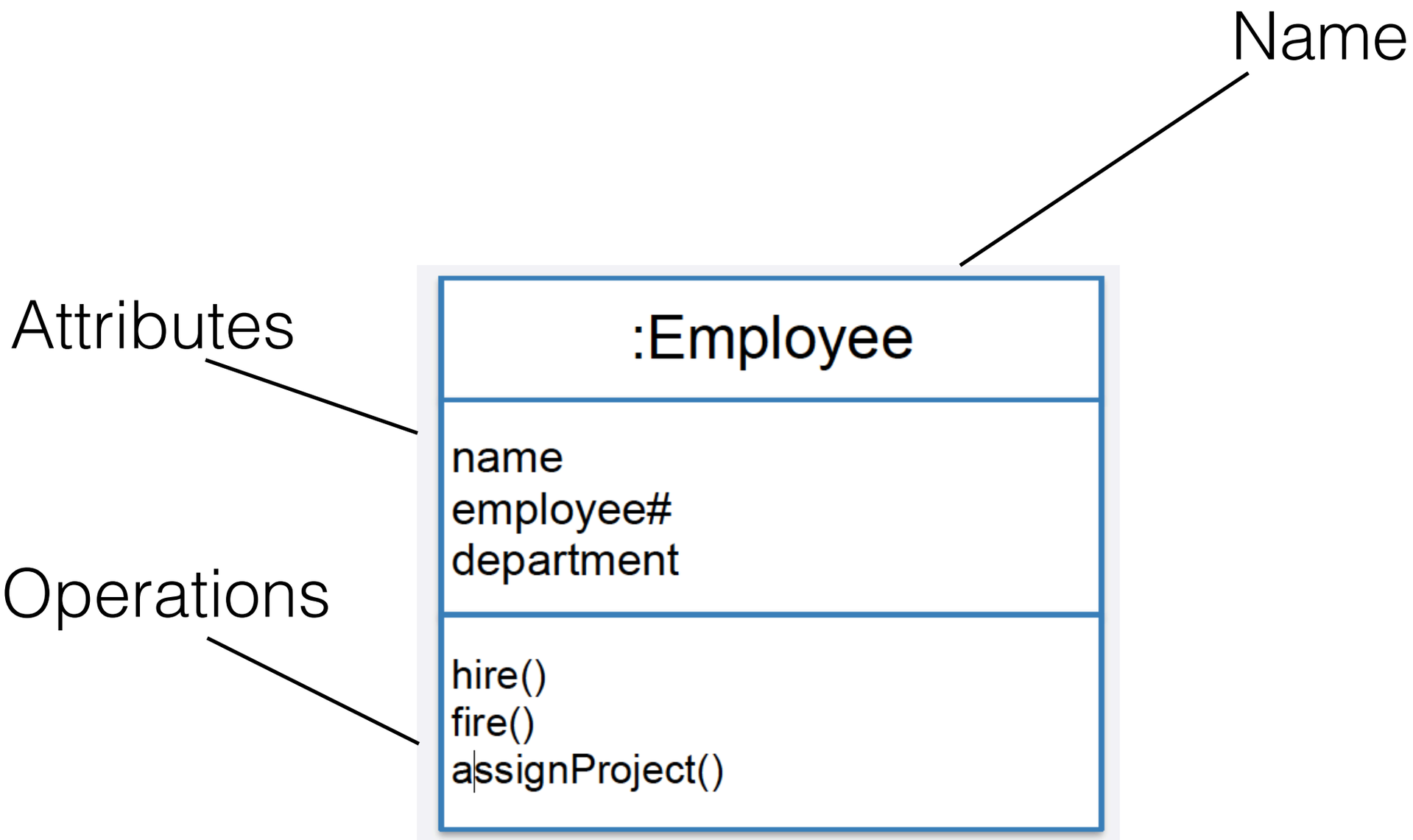
department

an employee is

hired,

fired;

an employee works in one or more projects



UML Class Diagram parts

Objects do not exist in isolation

UML supports:

Association

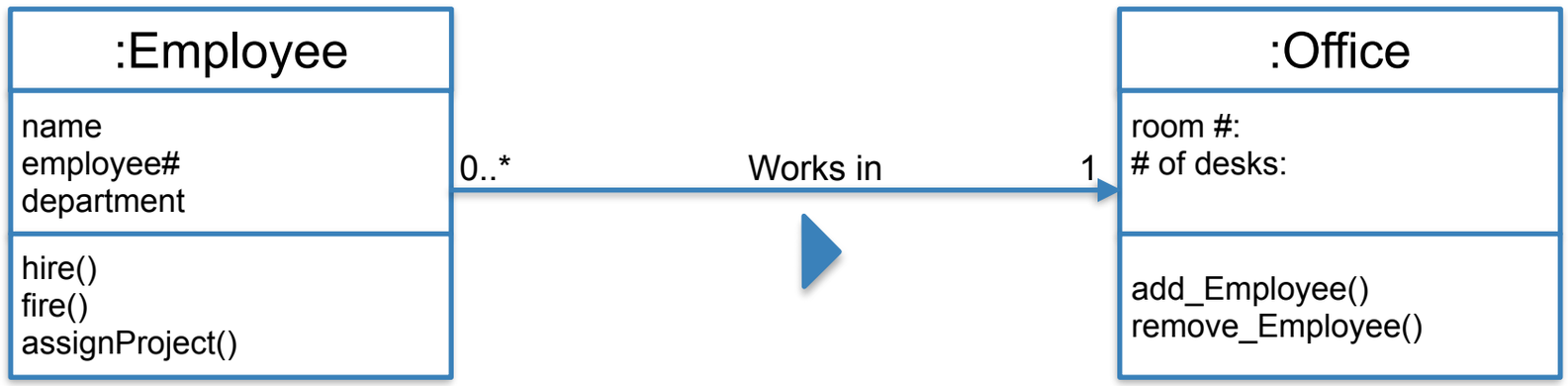
Aggregation and Composition

Generalization

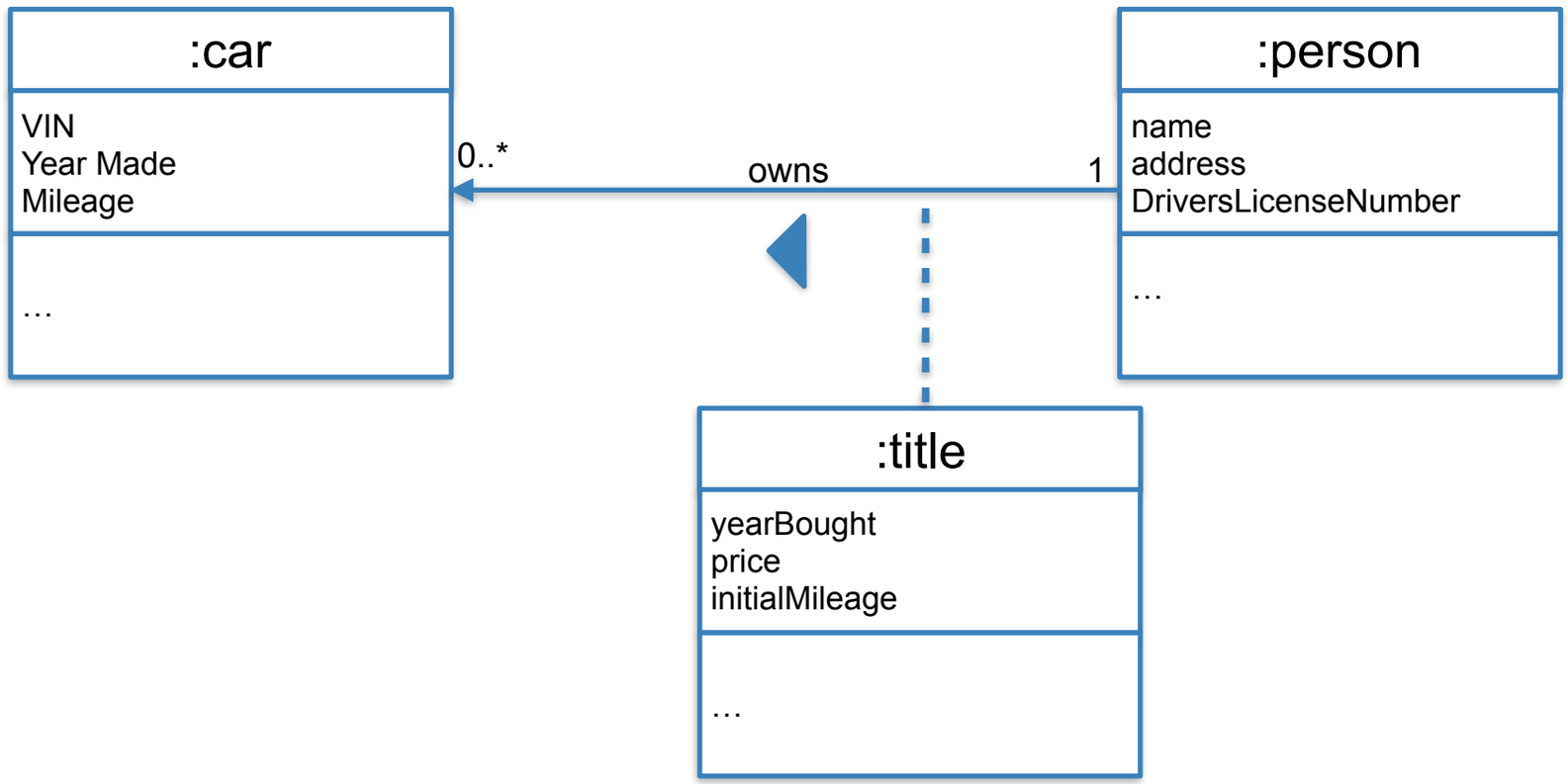
Dependency

Class Associations

Most generic kind of relationship



Example of an association class



Aggregation and Composition

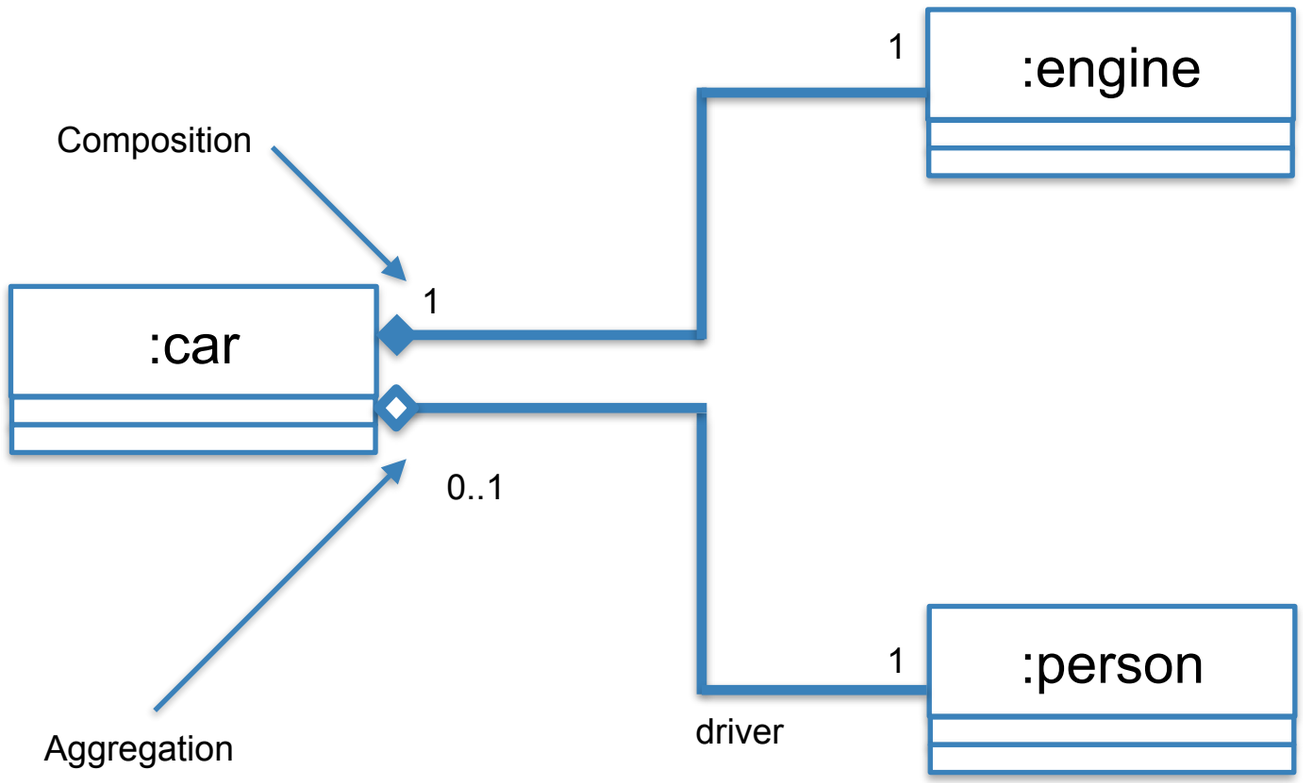
Aggregation:

This is the “has-a” or “whole/part” relationship

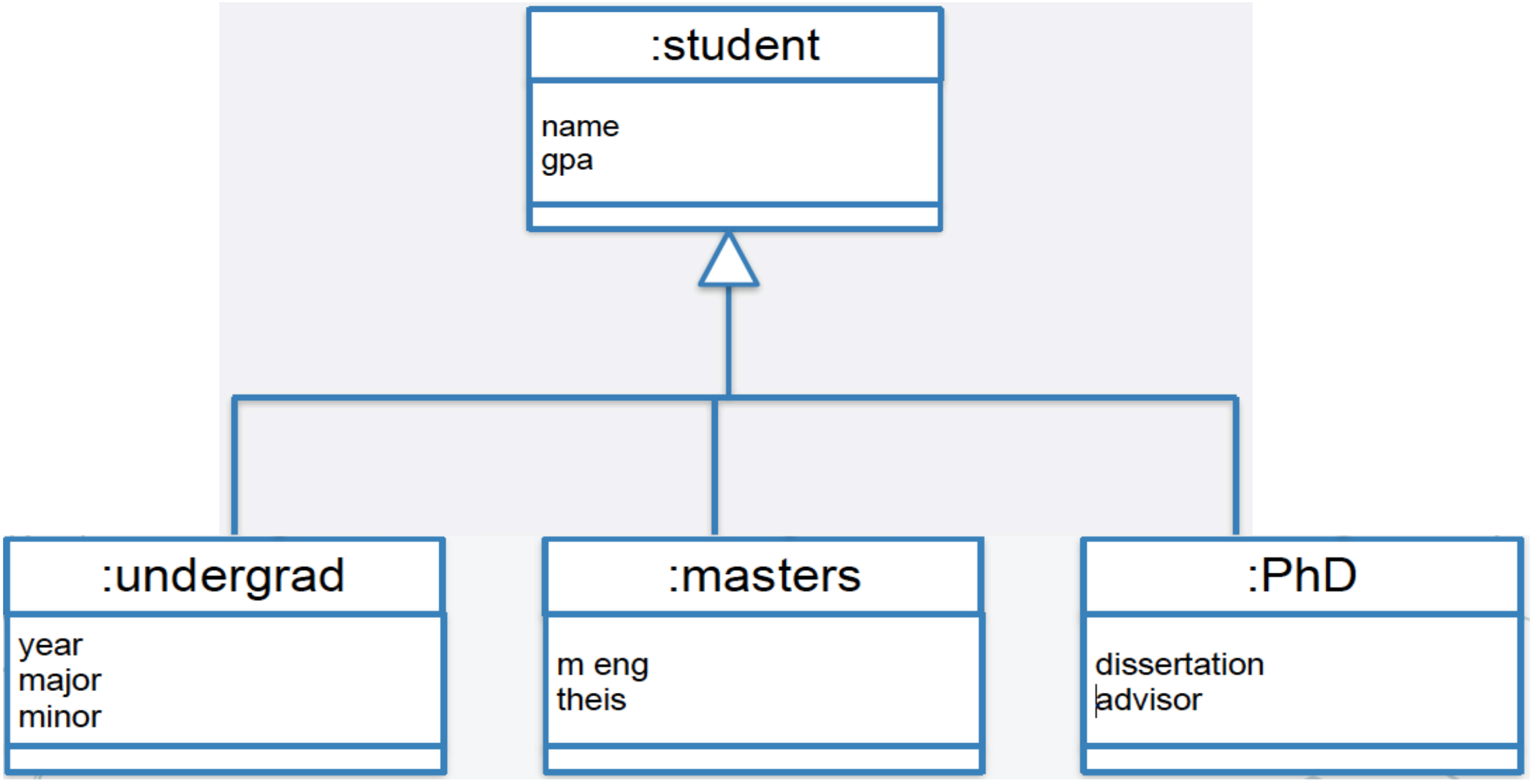
Composition:

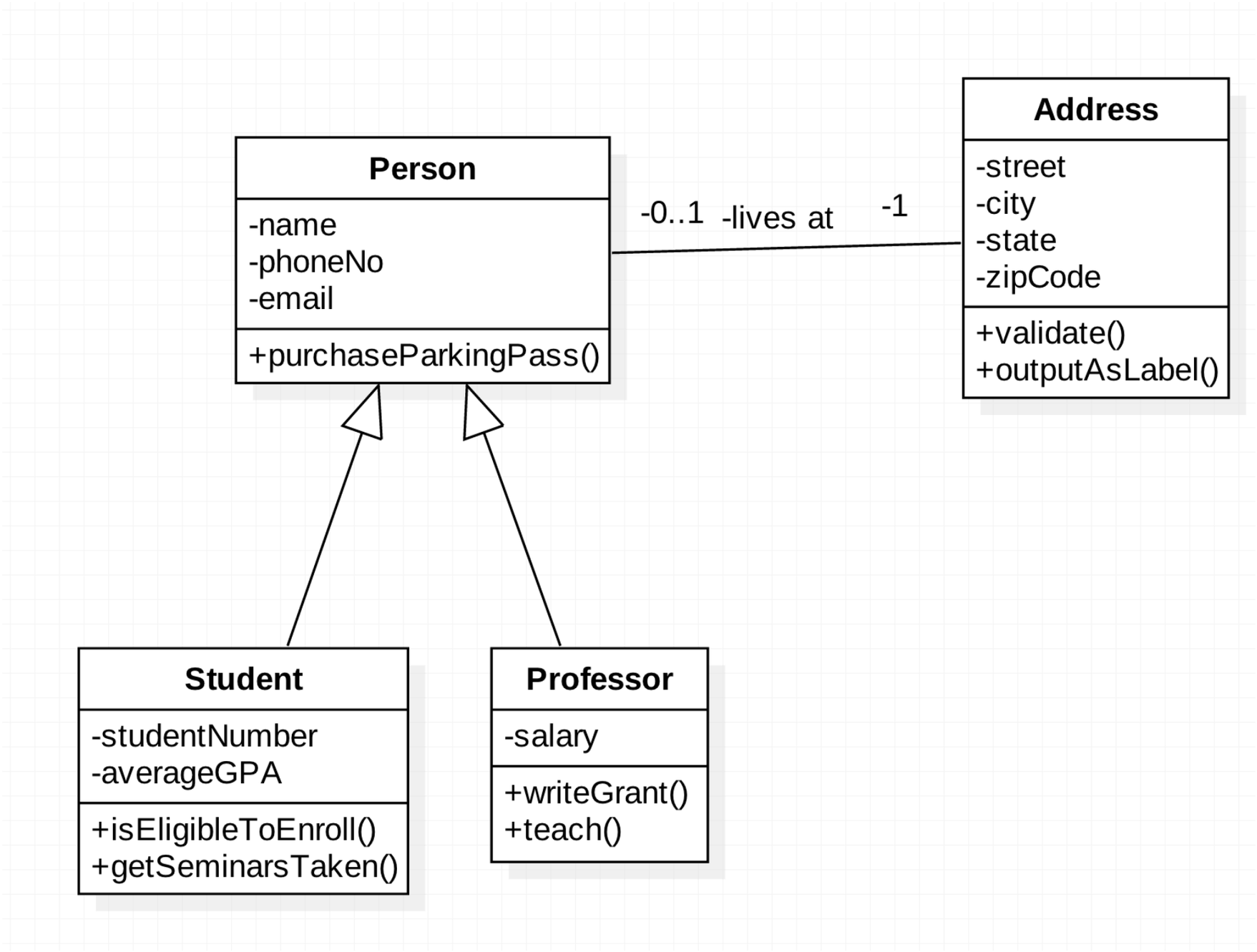
implies ownership

Aggregation and Composition example



Generalization example





Exercise

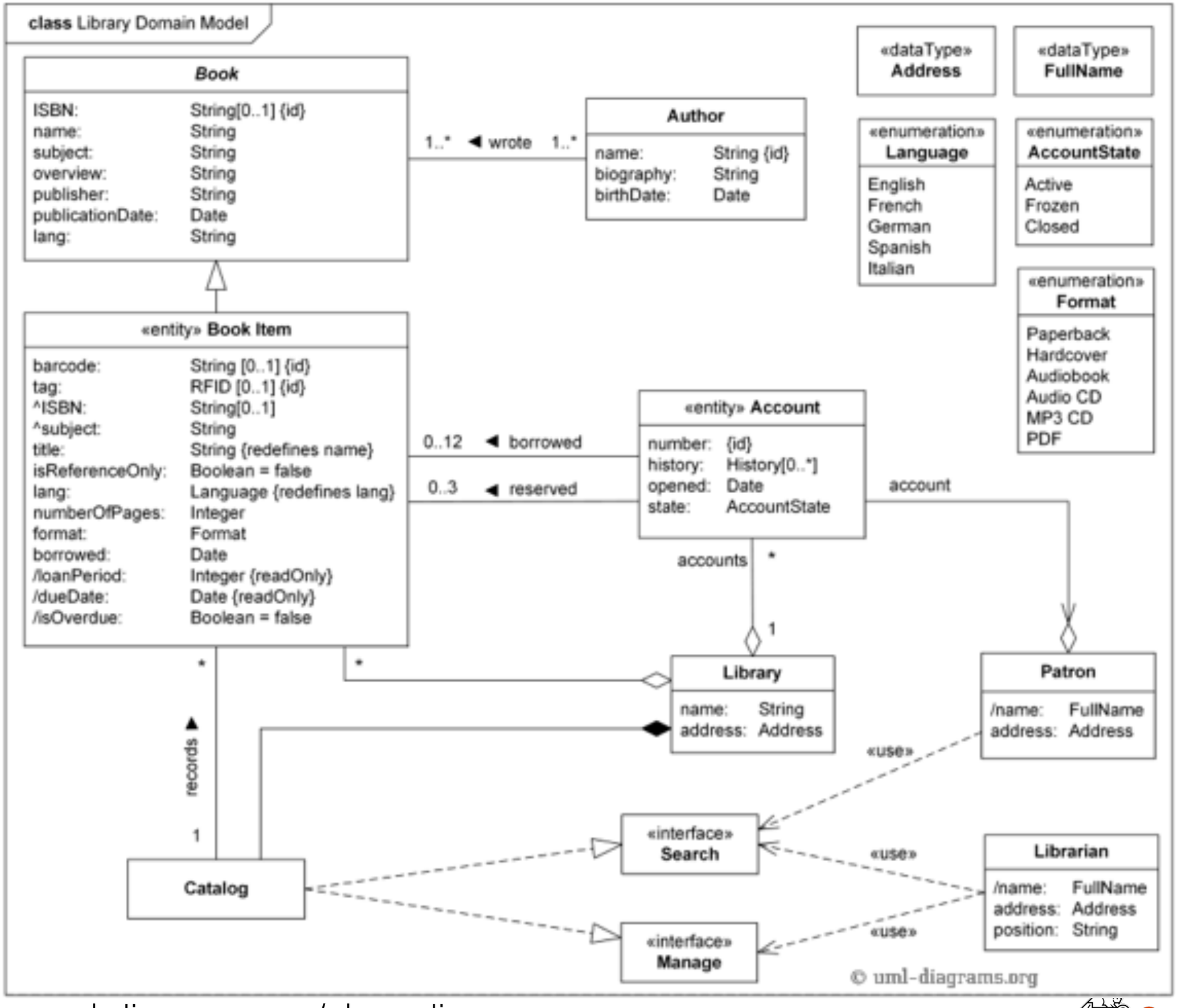
Book buying in Amazon

Specific book versions may have differences (online, hard copy, soft copy)

Books are written by authors

Books have reviews

A user (or their account)



http://www.uml-diagrams.org/class-diagrams-overview.html

Oregon State University