

# Machine Learning SWE Diagrams

**Mostafa S. Ibrahim**

*Teaching, Training and Coaching for more than a decade!*

*Artificial Intelligence & Computer Vision Researcher*

*PhD from Simon Fraser University - Canada*

*Bachelor / MSc from Cairo University - Egypt*

*Ex-(Software Engineer / ICPC World Finalist)*

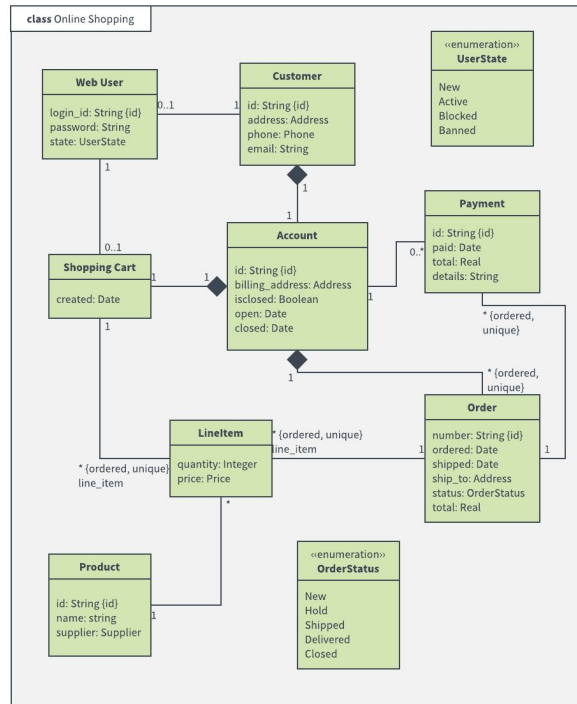


© 2023 All rights reserved.

Please do not reproduce or redistribute this work without permission from the author

# How can we Elegantly Represent a System or Task?

- We want to use natural text (Functional Requirement) to describe it!
- Just code the system (Like in the AskMe Project)
- Draw a **diagram** for it!

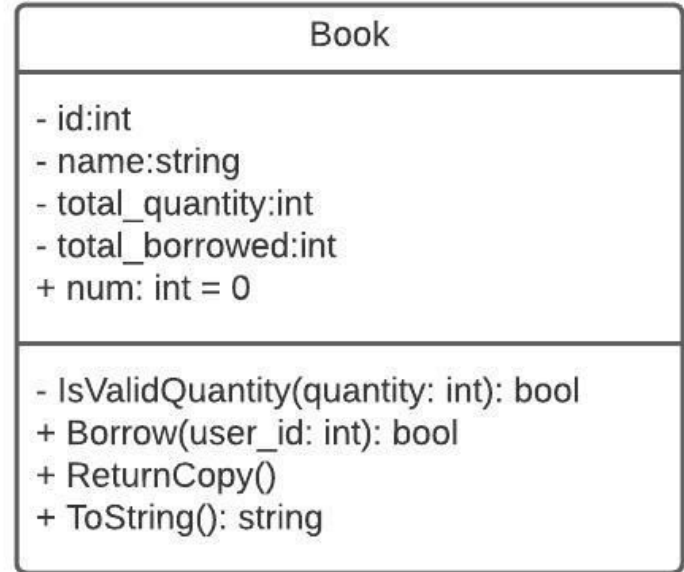


# Unified Modeling Language (UML)

- A **standardized** modeling language for the purpose of software **visualization**
- Imagine a system that consists of:
  - 3 subsystems
  - each sub-system contains ~10 components,
  - each component has ~100 classes
- We need different ways to communicate how this system interacts with others
  - Written documents are one aspect
  - Another is the use of diagrams to visualize different aspects of the system
- Class diagrams are one of the most important UML diagrams

# Class Diagram

```
6- class Book {  
7   private:  
8     int id;  
9     string name;  
10    int total_quantity;  
11    int total_borrowed;  
12  
13+   bool IsValidQuantity(int quantity) {  
16  
17   public:  
18     int num = 0;  
19  
20+   bool Borrow(int user_id) {  
26  
27+   void ReturnCopy() {  
31  
32+   string ToString() {  
35 };  
36
```

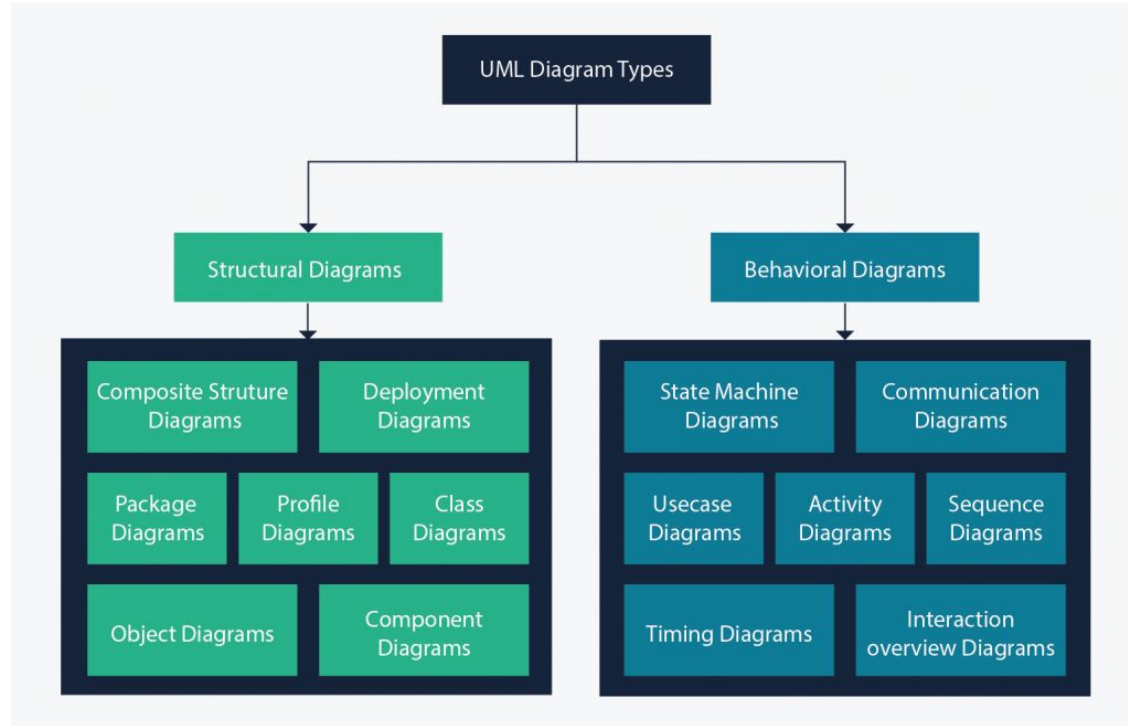


# Other UML Diagrams

- Tools

- [Lucidchart](#) (online)
- [Diagrams](#) (online)
- ArgoUML
- Visual Paradigm
- StarUML
- Microsoft Visio

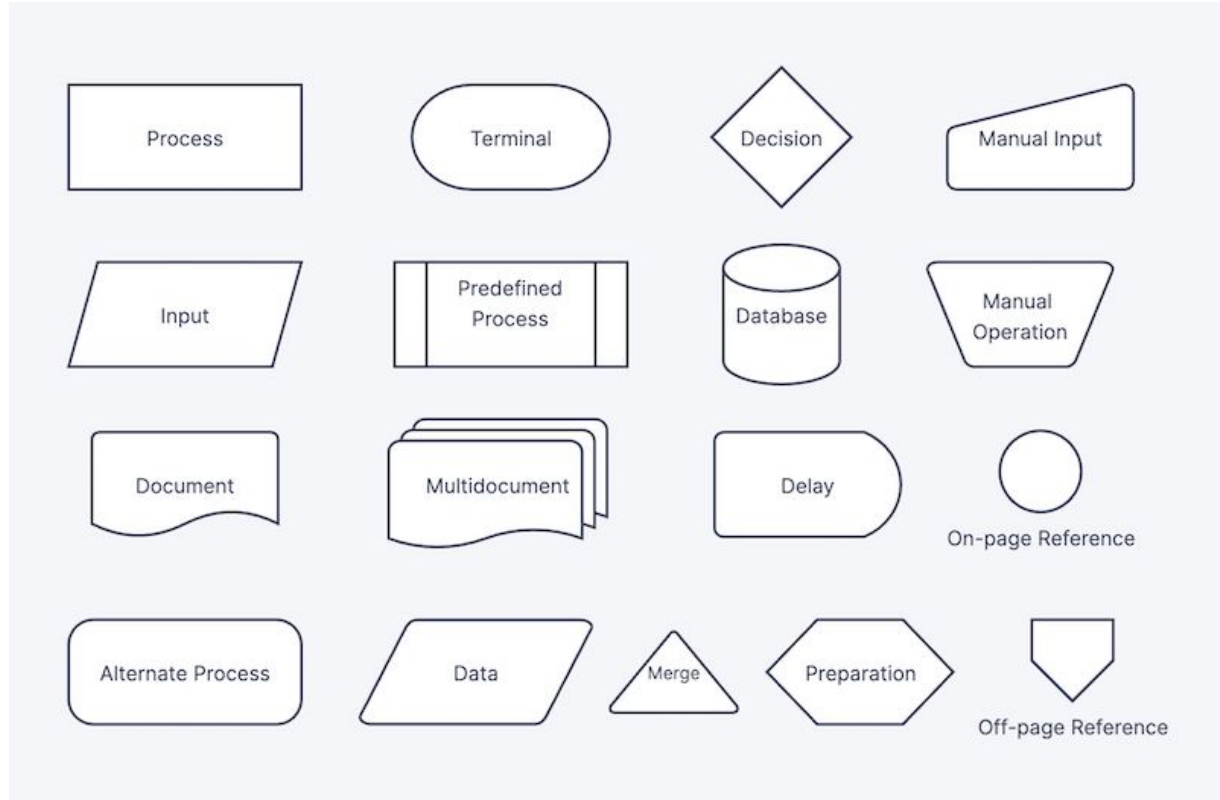
Img src



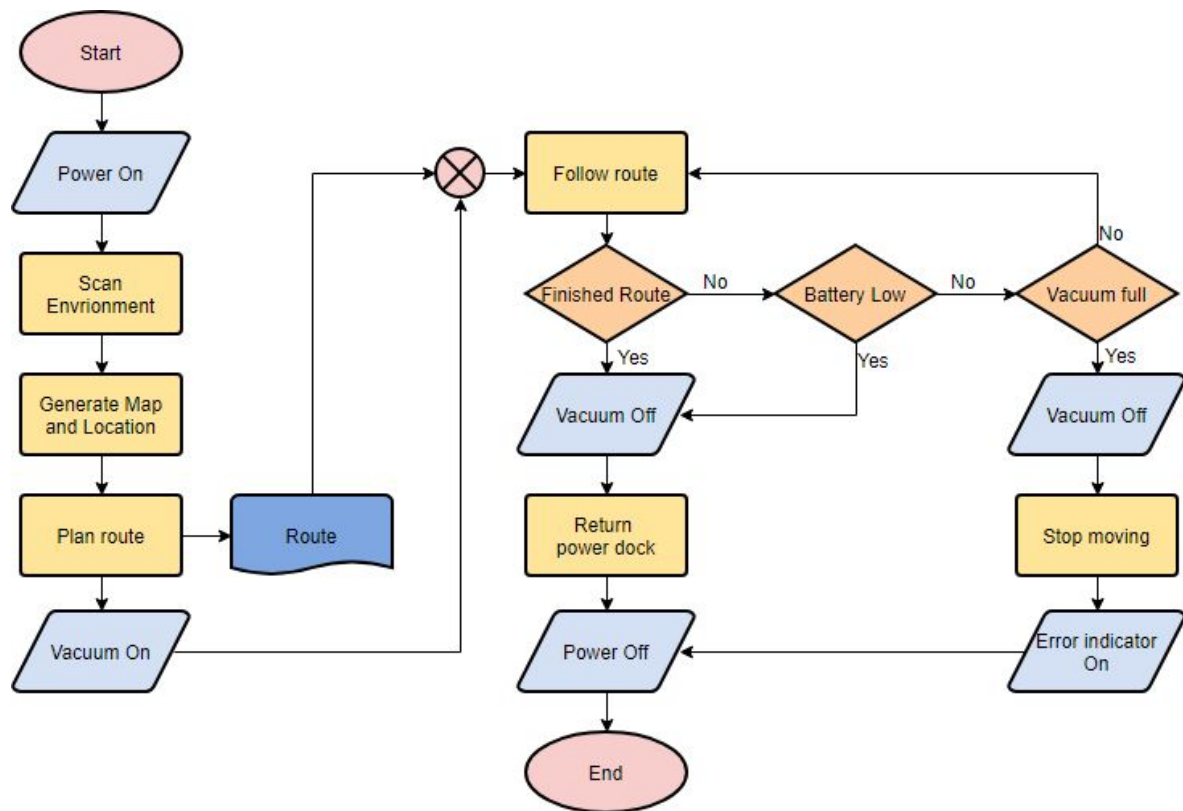
# Flowchart

- Flowcharts are used for representing algorithms or workflows in a graphical manner.
  - For example, the pipeline of trained models (input/output/process)
  - It can also model the data flow with the algorithms

# Flowchart symbols



# Flowchart



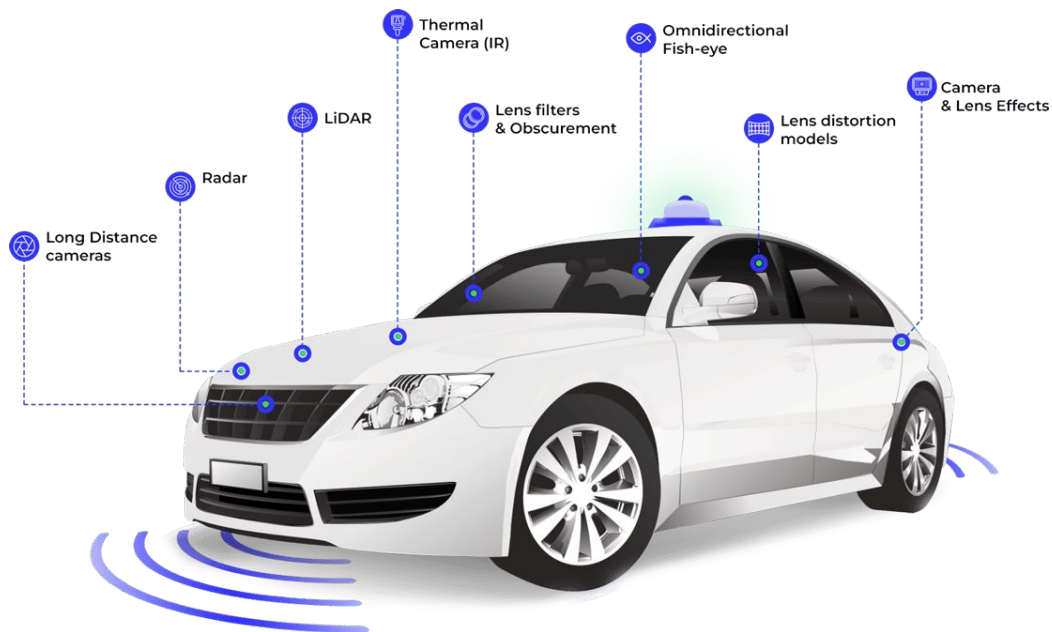


# Data Flow Diagram (DFD)

- DFD is used to represent the **flow of data** within the system.
  - It shows how data **enters** and **exits** the system, where/how it gets **modified/stored**.
  - Example: the DFD might show the flow from **image acquisition**, **pre-processing**, **feature extraction**, **feature-storage** and **result storage/display**

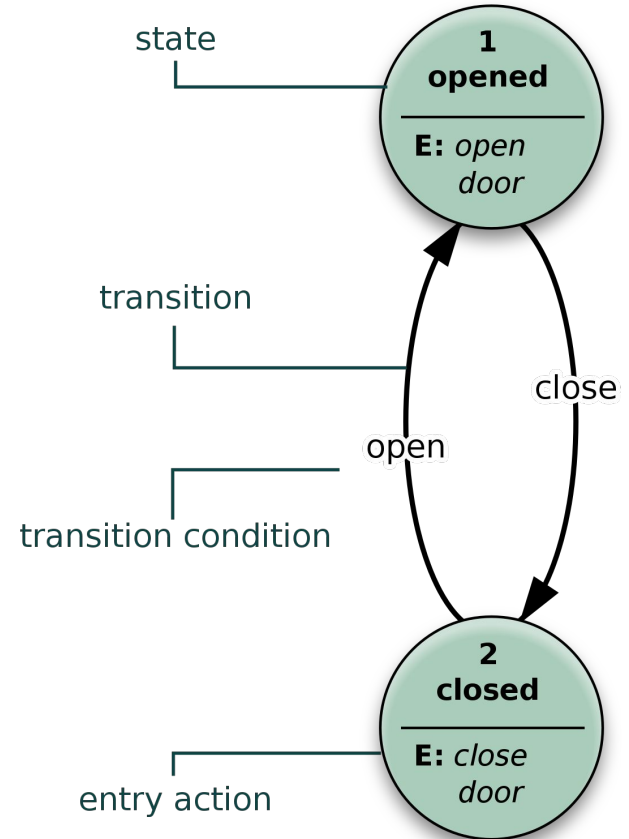
# Data Flow Diagram

- For example, our car has
  - 8 cameras
  - Radar
  - Lidar
  - 3 [Kinects](#) (RGB, IR, depth)
- We must know
  - What we read
  - Where processed
  - Where stored
  - Etc



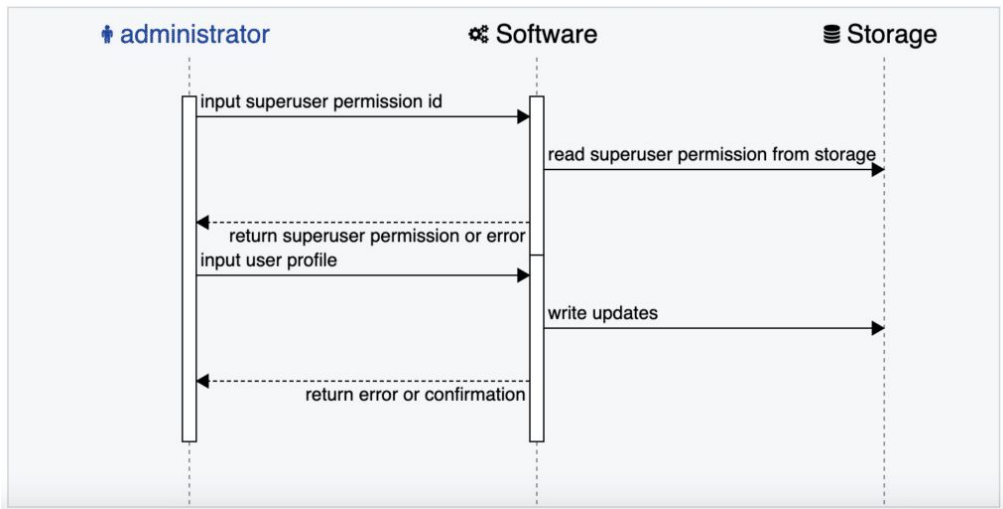
# State Diagram

- This diagram describes the behavior of a system by showing its states and the transitions between those states



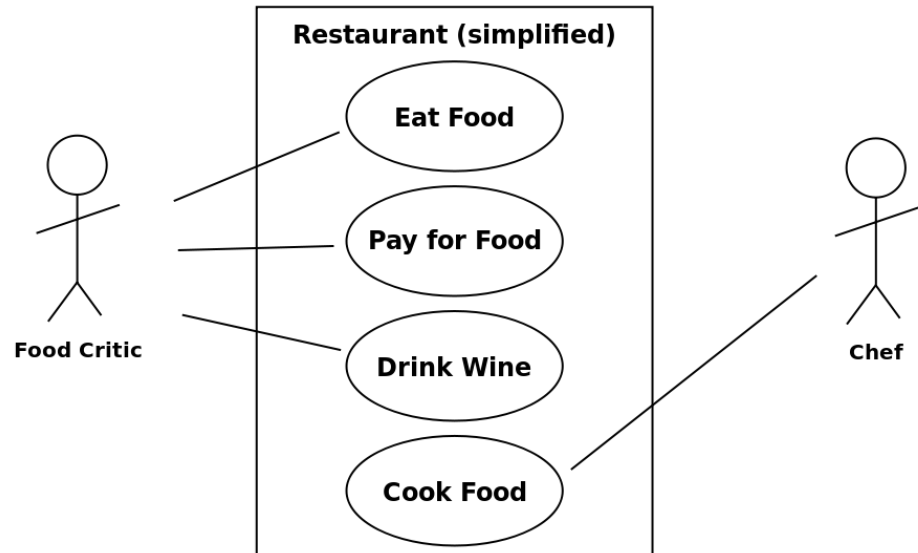
# Sequence Diagram

- A sequence diagram is commonly used to **show interactions** between objects in the sequential order that they occur.
  - Example: In a recommendation system, the sequence diagram may start from a user logging in, the system fetching user preferences, calculating recommendations, and finally displaying them.



# Use case diagram

- Models the different types of users (actors) that interact and with a system



# More diagrams

- Class diagram: structure of the classes and their relationships
- Architecture Diagram: a high-level view of the architecture / components
- ER Diagram: how different entities like databases related to each other.
- Deployment Diagram
  - This diagram shows the deployment of software components on hardware resources.
  - Example: shows distribution of components: how your training data is stored in a data lake, while your ML models are deployed in containers on a Kubernetes cluster
- We typically don't use diagrams for trivial things. They are used when we want to visually stress something or clarify a complex idea

*“Acquire knowledge and impart it to the people.”*

*“Seek knowledge from the Cradle to the Grave.”*

