

Génie logiciel pour la conception d'un Système  
d'Information

**CSC4521**

Voie d'Approfondissement  
Intégration et Déploiement de Systèmes d'Information  
(VAP DSI)

**Functional Architecture - Elevator/Lift**

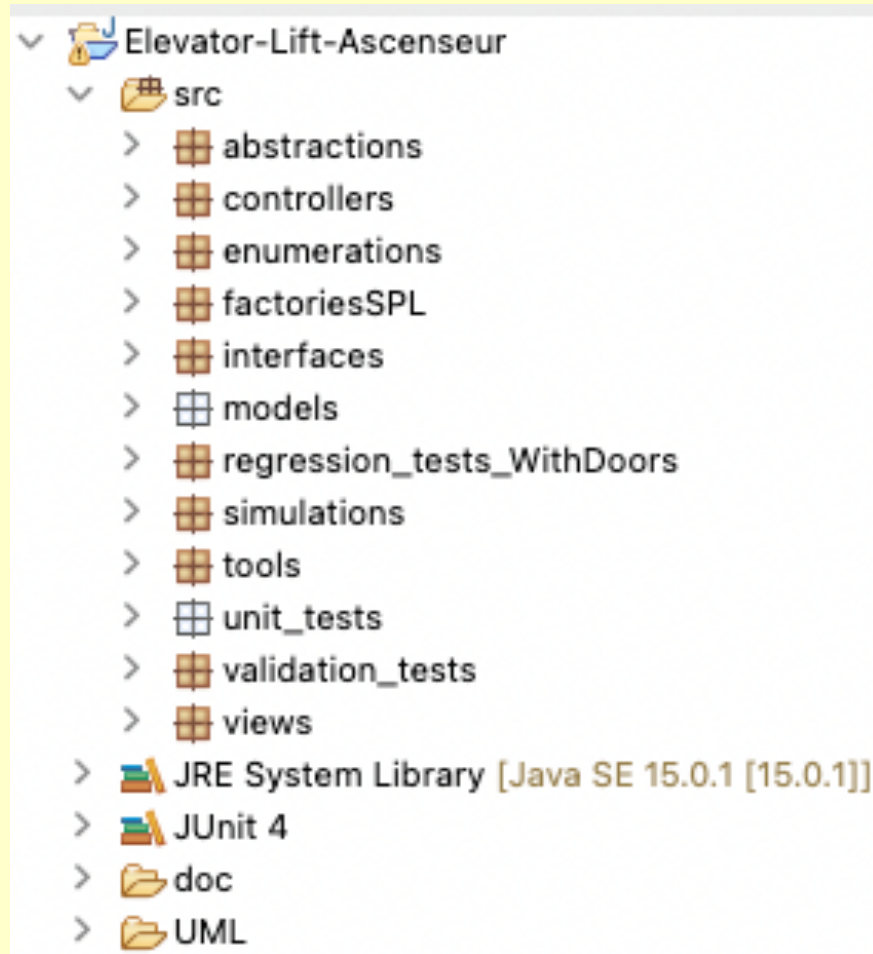
[paul.gibson@telecom-sudparis.eu](mailto:paul.gibson@telecom-sudparis.eu)

<http://jpaulgibson.synology.me/~jpaulgibson/TSP/Teaching/CSC4521/CSC4521-FunctionalArchitecture-ElevatorLift.pdf>

- In this session we will finalise the **functional architecture of your systems**. First we will analyse the functional architecture of the elevator/lift system.
- We will 'reverse engineer' the functional architecture from the Java implementation (provided):
  - What is the state of the system?
  - How much of the system environment is modelled in the state?
  - What are the state transitions (that require modelling as functions)?
  - How do interactions between the system and its environment lead to changes in the state?
  - What do you know about the frequency of these interactions?
  - What information/data is communicated during these interactions?
  - Can the state change independent of its environment? If so, how?
  - How is the state distributed amongst parts (components/subsystems) of the system?
  - Where/How are the state transitions implemented - how is this functionality distributed amongst parts (components/subsystems) of the system?
- Note that the Java is an OO design and so the functional architecture will group state and functions inside objects/classes (as attributes and methods)

Download the Java code from :

<http://jpaulgibson.synology.me/~jpaulgibson/TSP/Teaching/CSC4521/Code/Elevator-Lift-Ascenseur.zip>



How do you understand code written by another engineer?

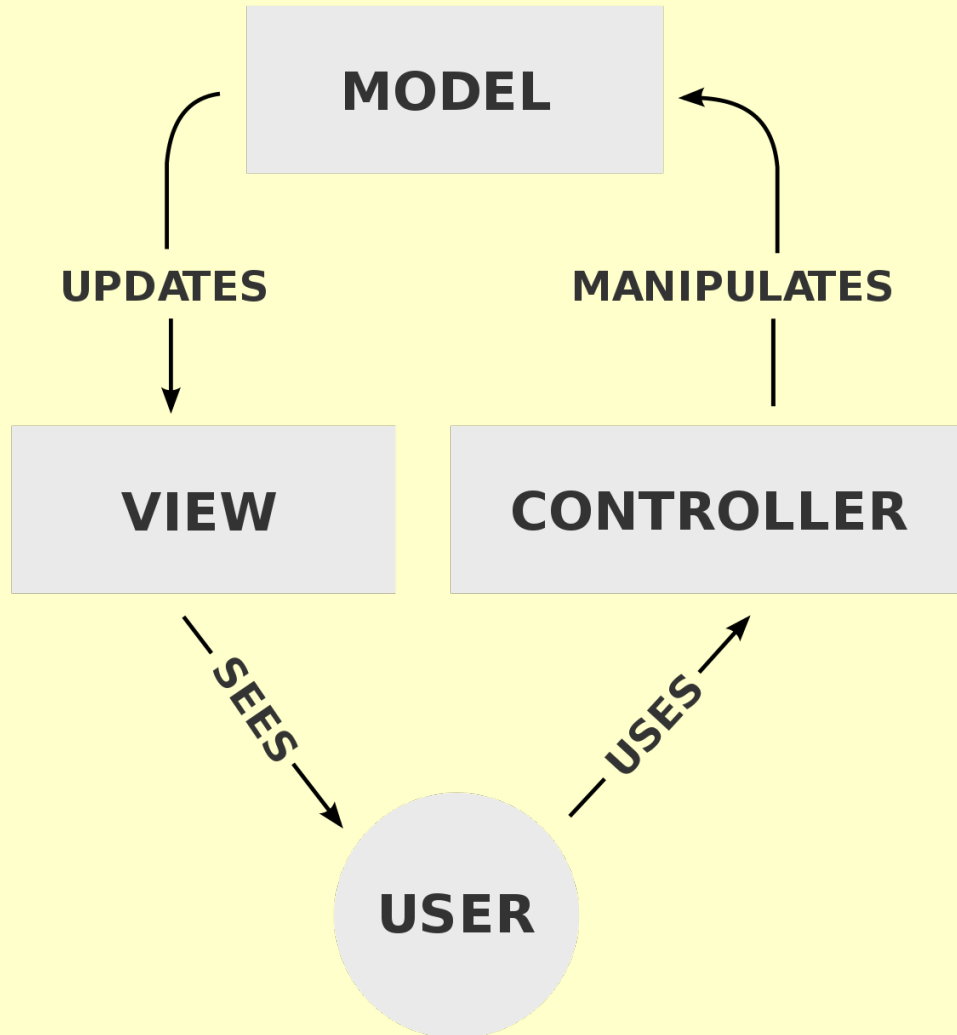
1. Look at documentation
2. Look at tests
3. Look at code
4. Run the code

Run the **simulation** on different elevator “products”



Does it function correctly?

# How is the functionality structured?



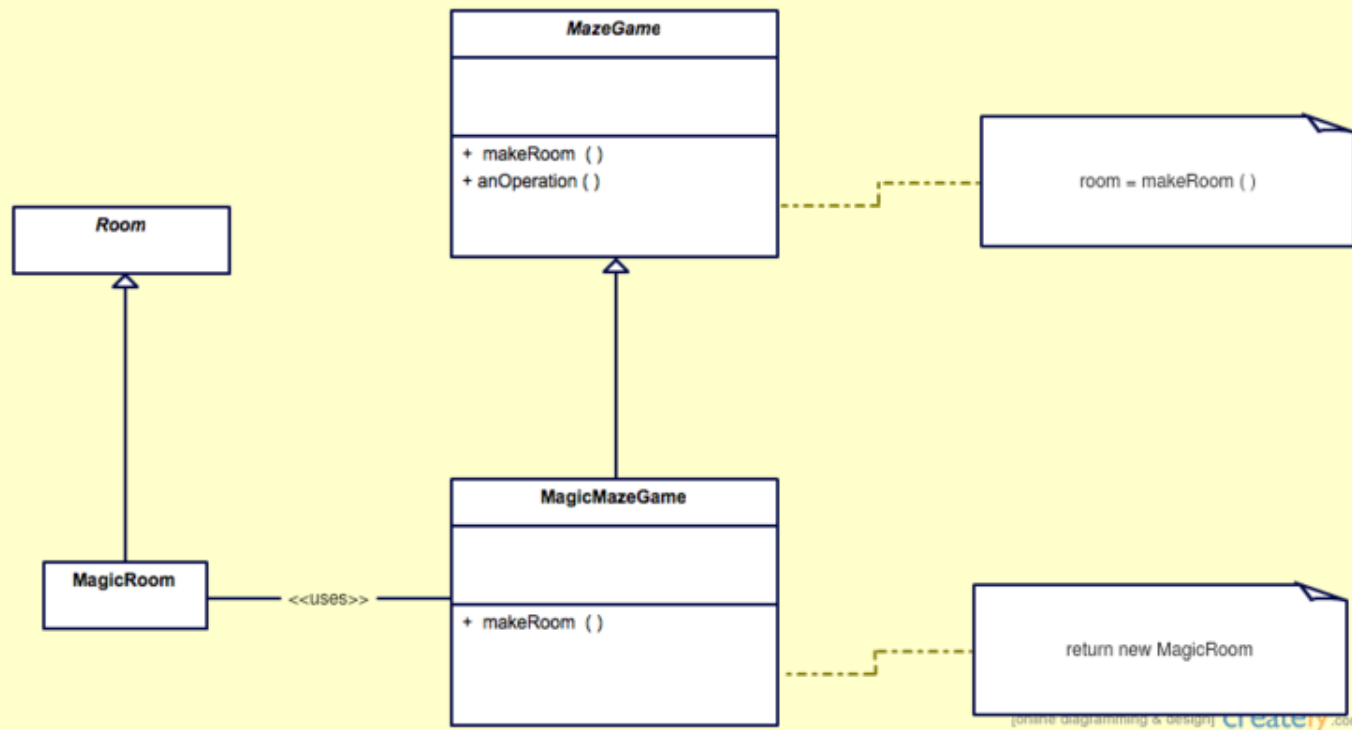
High level architecture is **MVC**

Should you use this in your system?

<https://en.wikipedia.org/wiki/Model-view-controller>

# How is the functionality structured?

There also seems to be a **Factory** for a software product line. Should you do the same for your system?

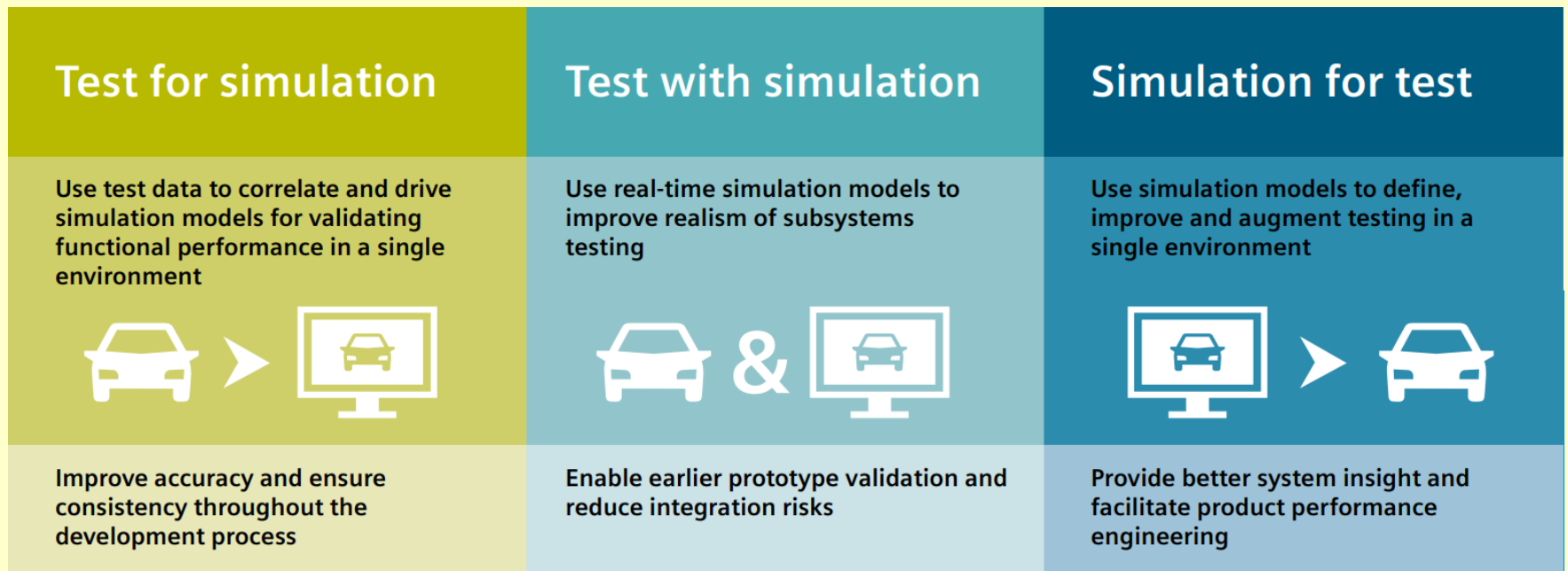


[https://en.wikipedia.org/wiki/Factory\\_method\\_pattern](https://en.wikipedia.org/wiki/Factory_method_pattern)

# How is the functionality validated?

There is a **simulation** that can be used to validate behaviour with the client before the system is deployed

Should you develop a simulation of your system?



<https://blogs.sw.siemens.com/simcenter/simcenter-testlab-2019-1-embedding-virtual-simulation-within-physical-testing/>

## Diagrams/ Models

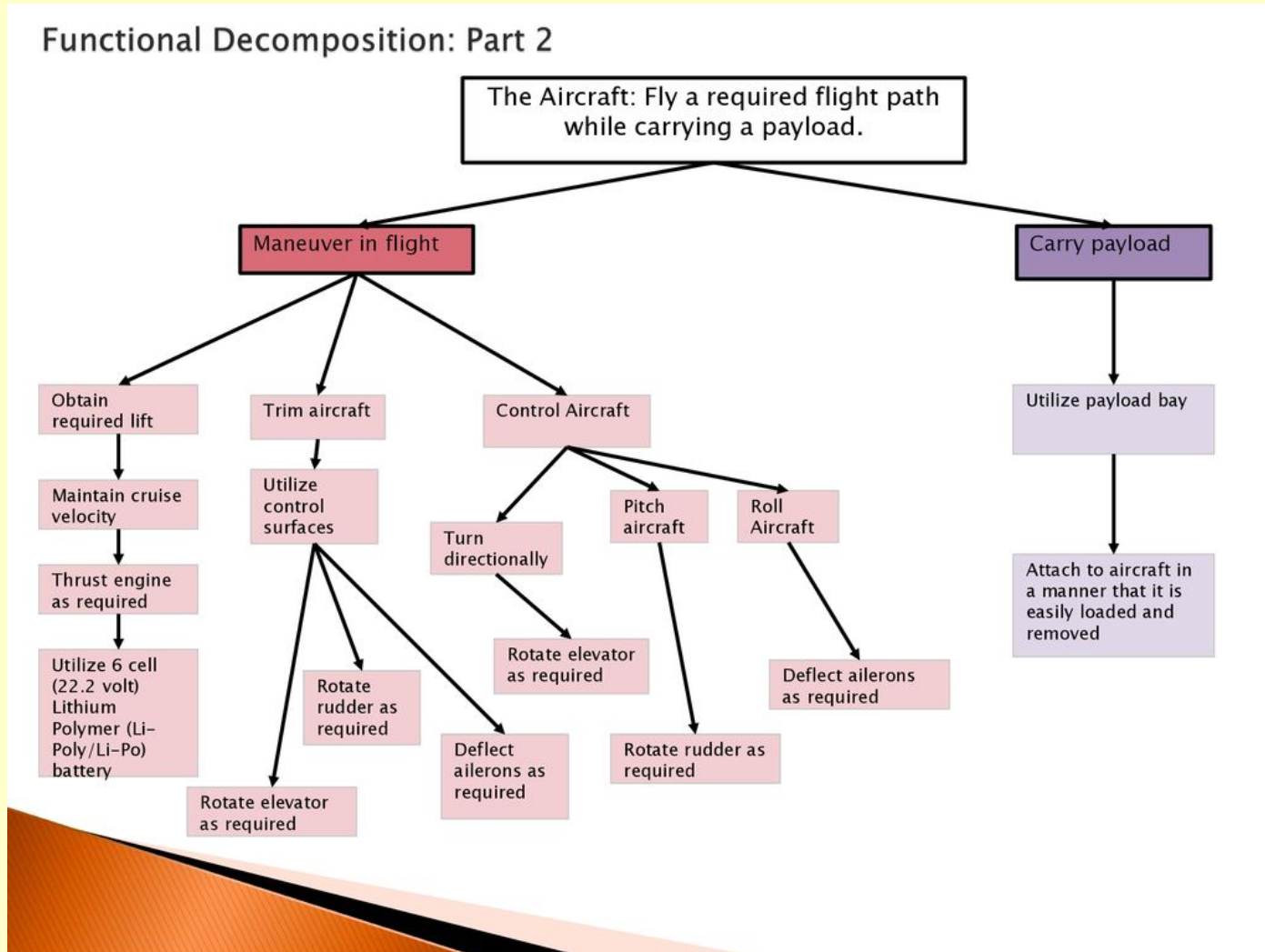
- Draw a UML class diagram for the elevator system/lift. A useful online tutorial can be found at: [JavaTpoint](#)
- After you have completed the class diagram for the provided lift system, and checked it with Paul, draw the functional architecture of your system as a UML class diagram. To guide you, answer the same questions as you did for the lift system.

You may choose to also (alternatively) draw:

- 1) A functional decomposition diagram
- 2) A Data Flow Diagram



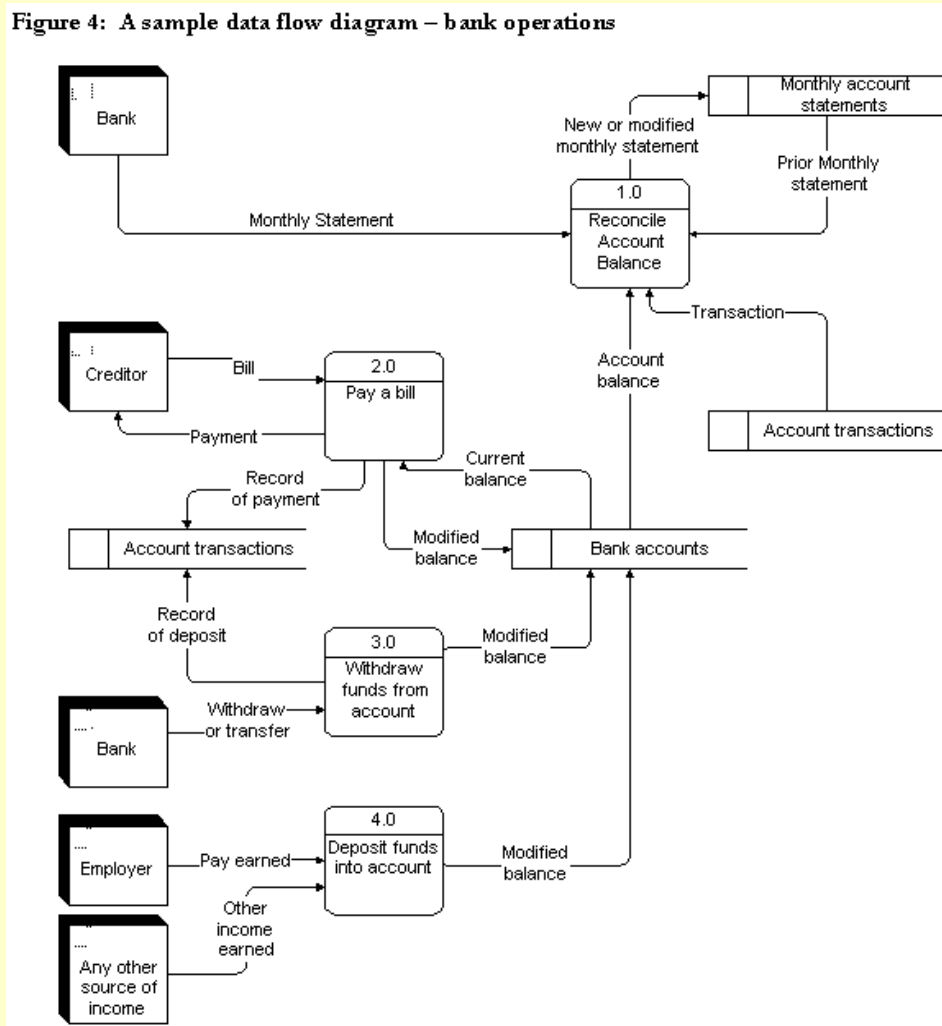
# A functional decomposition diagram (airplane example)



<https://slideplayer.com/slide/17162648/>

# A Data Flow Diagram (Bank Example)

Figure 4: A sample data flow diagram – bank operations



<https://faculty.babson.edu/dewire/readings/dfddiag.htm>