OOAD vs SASD

3조	
200312468	김완수
200511363	한상현
200960122	사인빌릭 체렝밤바

Contents

1. OOAD – Object Oriented Analysis and design
2. SASD – Structured Analysis and Structured Design
3. OOAD vs SASD (Similarities)
4. OOAD vs SASD (Differences)
5. Summary

1.OOAD
– Object Oriented Analysis and design

- What is OOAD?

  - **Object-oriented analysis and design** (OOAD) is a softw are engineering approach that models a system as a gro up of interacting objects. Each object represents some e ntity of interest in the system being modeled, and is cha racterised by its class, its state (data elements), and its b ehavior. Various models can be created to show the stati c structure, dynamic behavior, and run-time deployment of these collaborating objects. There are a number of dif ferent notations for representing these models, such as t he Unified Modeling Language (UML). – said **Wikipedia**

1.OOAD
– Object Oriented Analysis and design

- We used OSP!
  - OSP is one of the common methods used by **Software Development Process** based on RUP
  - OSP has 3 stages
    - Stage 1000 : Plan and Elaboration
      - – Planning, defining requirements, building prototyping, etc
      - – Corresponding to Inception/Elaboration phases in the RUP
    - Stage 2000 : Build
      - Construction of the system
      - Corresponding to Construct phase in the RUP
    - Stage 3000 : Deployment
      - Implementation of the system into use
      - Corresponding to Transition phase in the RUP

2.SASD
– Structured Analysis and Structured Design

- What is SASD?

  - **Structured Analysis** (SA) in software engineering and it s allied technique, Structured Design (SD), are methods f or analyzing and converting business requirements into specifications and ultimately, computer programs, hard ware configurations and related manual procedures. Str uctured analysis and design techniques are fundamental tools of systems analysis, and developed from classical s ystems analysis of the 1960s and 1970s.

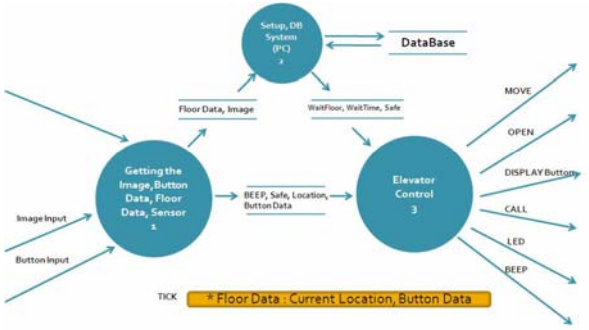
    – said **Wikipedia too**

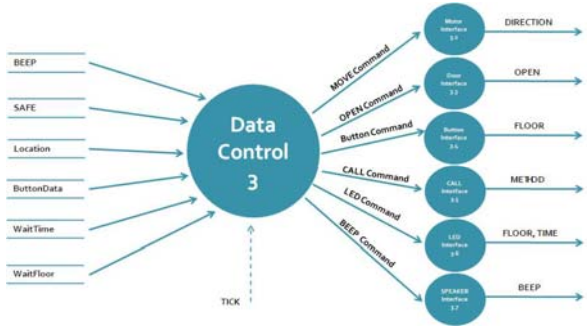
3.OOAD vs SASD (Similarities)

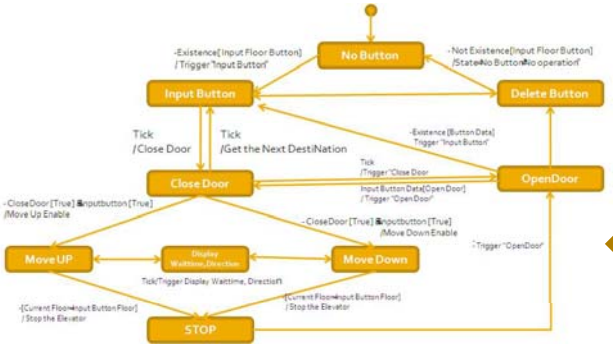
- Similarities
  - The both have started off from programming techniques.
  - The both use graphical design and tools to analyze and model **requirements**.
  - The both provide a systematic step-by-step process for developers.
  - The both focus on the documentation of requirements

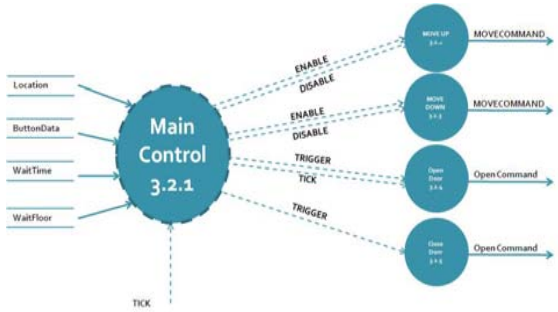
3.OOAD vs SASD - step – by – step

- Ex) SASD - DFD



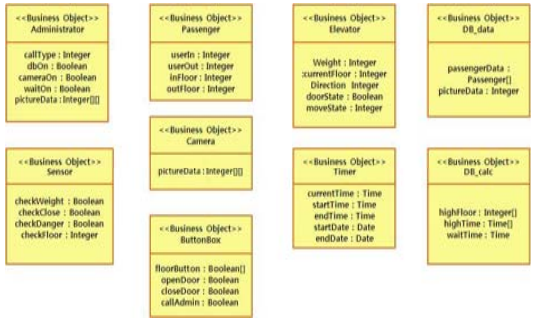


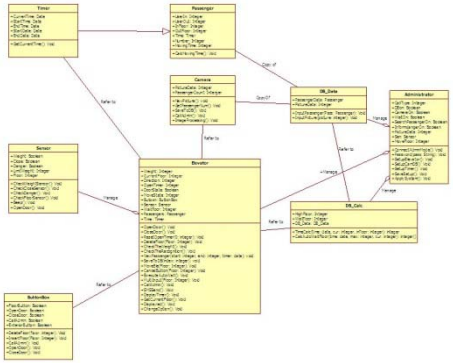


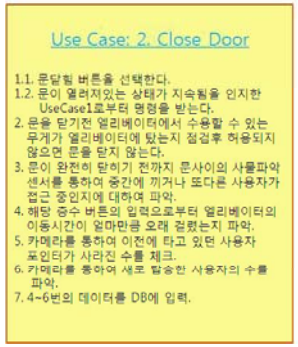


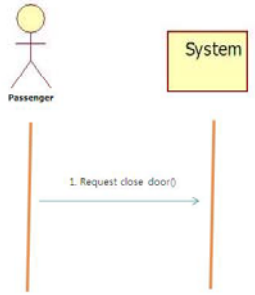
3.OOAD vs SASD - step – by – step

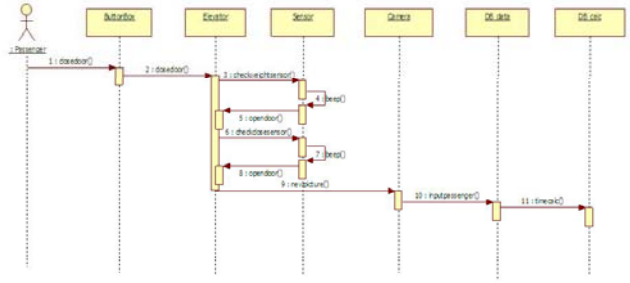
- Ex)OOAD











3.OOAD vs SASD - Data

- Both of OOAD and SASD are defining about the data used in program. But because what they are focusing is different, they are used in different ways.

-Refine Terms

Term	Category	remarks
Elevator.weight	Attribute	무게허용한도를 가지고 있는 변수
Elevator.currentFloor	Attribute	현재층의 데이터를 가지고 있는 변수
Elevator.direction	Attribute	엘리베이터의 이동방향을 가지고 있는 변수
Elevator.doorState	Attribute	문의 상태값을 가지고 있는 변수
Elevator.moveState	Attribute	이동상태를 가지고 있는 변수
Elevator.button	Attribute	입력된 버튼의 정보를 가지고있는 구조체형 변수
Elevator.sensor	Attribute	sensor클래스의 정보를 받아오는 변수
Elevator.waitFloor	Attribute	자동대기기능을 실행할 층의 정보를 가지고 있는 변수
ButtonBox.floorButton	Attribute	버튼박스의 층버튼 입력 정보를 가진 배열형 변수
ButtonBox.openDoor	Attribute	버튼박스의 문열기 기능 버튼 입력 정보를 가진 변수
ButtonBox.closeDoor	Attribute	버튼박스의 문닫기 기능 버튼 입력 정보를 가진 변수
ButtonBox.callAdmin	Attribute	버튼박스의 관리자 호출 기능 버튼 입력 정보를 가진 변수
DB_data.passengerData	Attribute	승객 이용정보를 가지고있는 구조체 변수
DB_data.pictureData	Attribute	사진정보를 가지고 있는 변수
DB_calc.highFloor	Attribute	자동대기층 우선순위를 계산해서 가지고있는 배열형 변수
DB_calc.passengerData	Attribute	DB_data에서 통계 계산을 위해 가져오는 구조체 변수
DB_calc.waitFloor	Attribute	현재 자동대기층값을 가지고 있는 변수
Sensor.checkWeight	Attribute	중량초과의 유무를 가지고 있는 변수
Sensor.checkClose	Attribute	문닫힘 상태 유무를 가지고 있는 변수
Sensor.checkDanger	Attribute	위험상태 유무를 가지고 있는 변수
Sensor.checkFloor	Attribute	엘리베이터의 위치를 체크하여 가지고 있는 변수

-DATA DICTIONARY

Input / Output Event	Description	Format / type
Weight Sensor	Check the total weight at a elevator	Figure(int), Interrupt
Location Sensor	Check the Elevator's Current Floor	Figure(int), Periodic
SafeDoor Sensor	Detected Obstacle in Between the two sides Door	True / False (Bool), Interrupt
Image Input	Get the Current Image in the Elevator	Image(int[][]) , Periodic
Button Input	Get the Passenger's Input Button Data	Figure(int[]), Interrupt
Move	Move Command to the Elevator	Figure(int[])
Open	Open Command to the Door	Open/Close(Bool)
Display Button	Display the Inputed Button Data at other Button Box	Figure(int[])
CALL	Call the admin and	ADMIN,119
Display LED	Display the Current Floor and Direction and WaitTime	Figure(int[][])
BELL	Output the BeepSound when total weight exceed	ON/OFF(Bool)

4.OOAD vs SASD (Differences)

- SA/SD Objects
  - Requirement Specification
  - Entities
  - E-R Diagrams
  - Data Flow Diagrams
  - State Transition Diagrams
  - Data Dictionaries
  - Structure Charts

4.OOAD vs SASD (Differences)

- SA/SD Objects
  - Requirement Specification
  - Entities
  - E-R Diagrams
  - Data Flow Diagrams
  - State Transition Diagrams
  - Data Dictionaries
  - Structure Charts

4.OOAD vs SASD (Differences)

- OOAD Objects
  - Use Case Diagrams
  - Objects
  - Collabration Diagrams
  - Sequence Diagrams
  - State Transition Diagrams
  - Class Diagrams & Packages
  - Component Diagram
  - Unified Modelling Language
  - Deployment Diagrams

4.OOAD vs SASD (Differences)

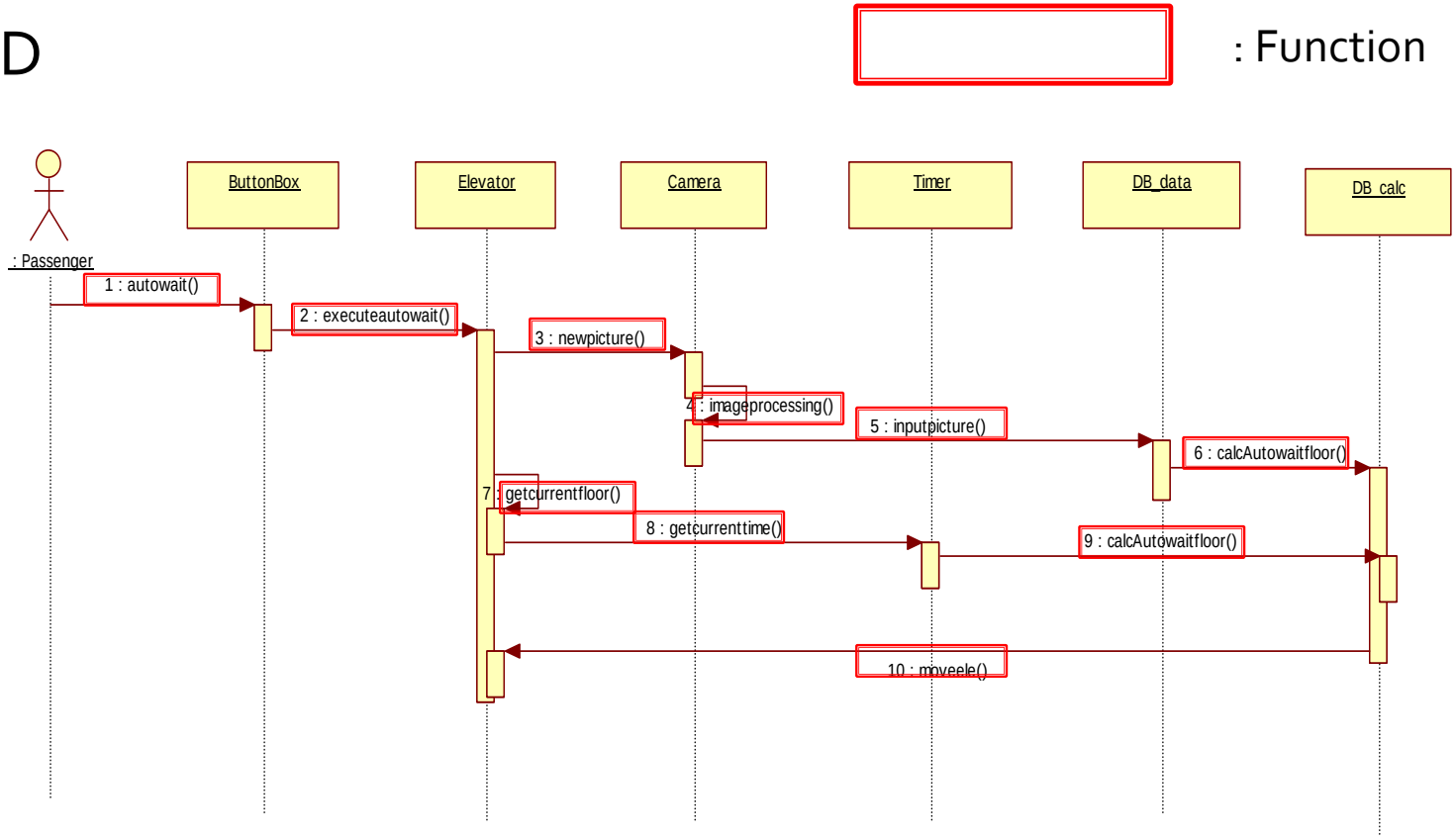
- The difference between OOAD and SASD

1. Flow Point
2. Revision Procedure
3. Incorrect Data Process
4. Design Complexity
5. Development System

4.OOAD vs SASD - Flow

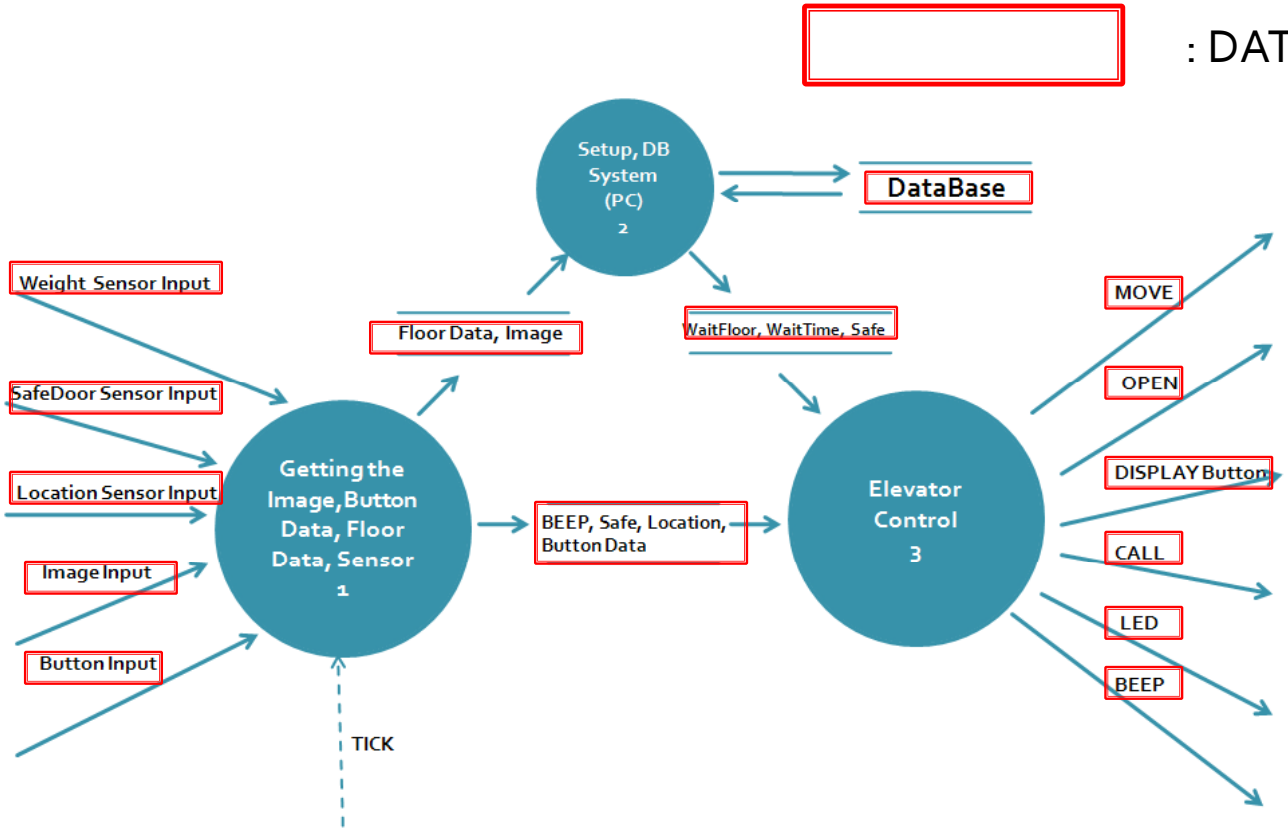
- OOAD be focus on the relationship between the function and Class(Objec t). But, SASD be Focus on the relationship between the datas.

- OOAD



4.OOAD vs SASD - Flow

- SASD



-DATA(Variables such as) I / O of the SASD Go and make sure you can see the most.

-How should we design to a lot of DATA (100.1000 Data)

4.OOAD vs SASD - Revision

- If OOAD or SASD need to modify or delete or insert any part?

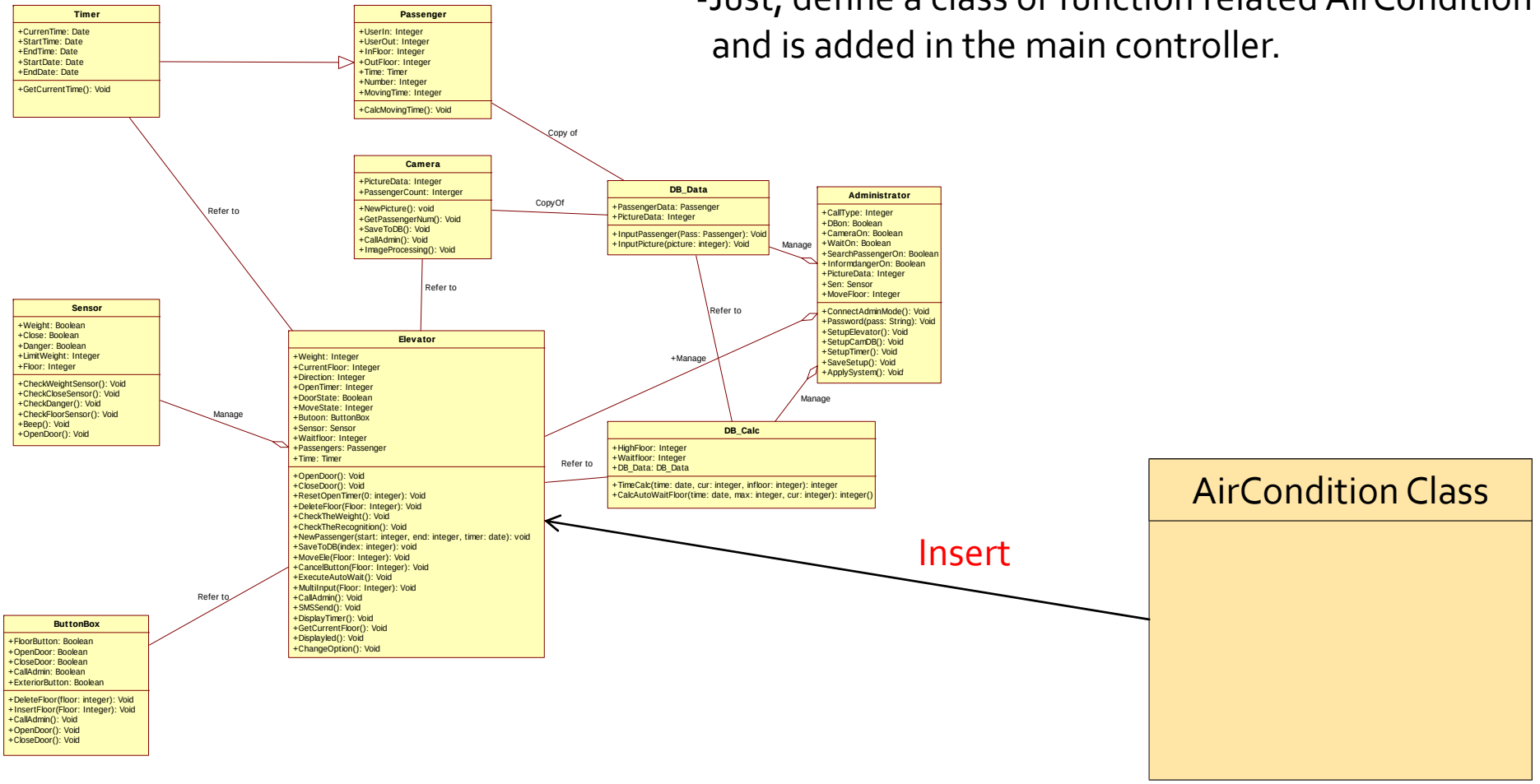
Example - insert

- If we can be created depending on the person number in the elevator controller to control the internal air-condition ing and feature-century If you would like to add?

4.OOAD vs SASD - Revision

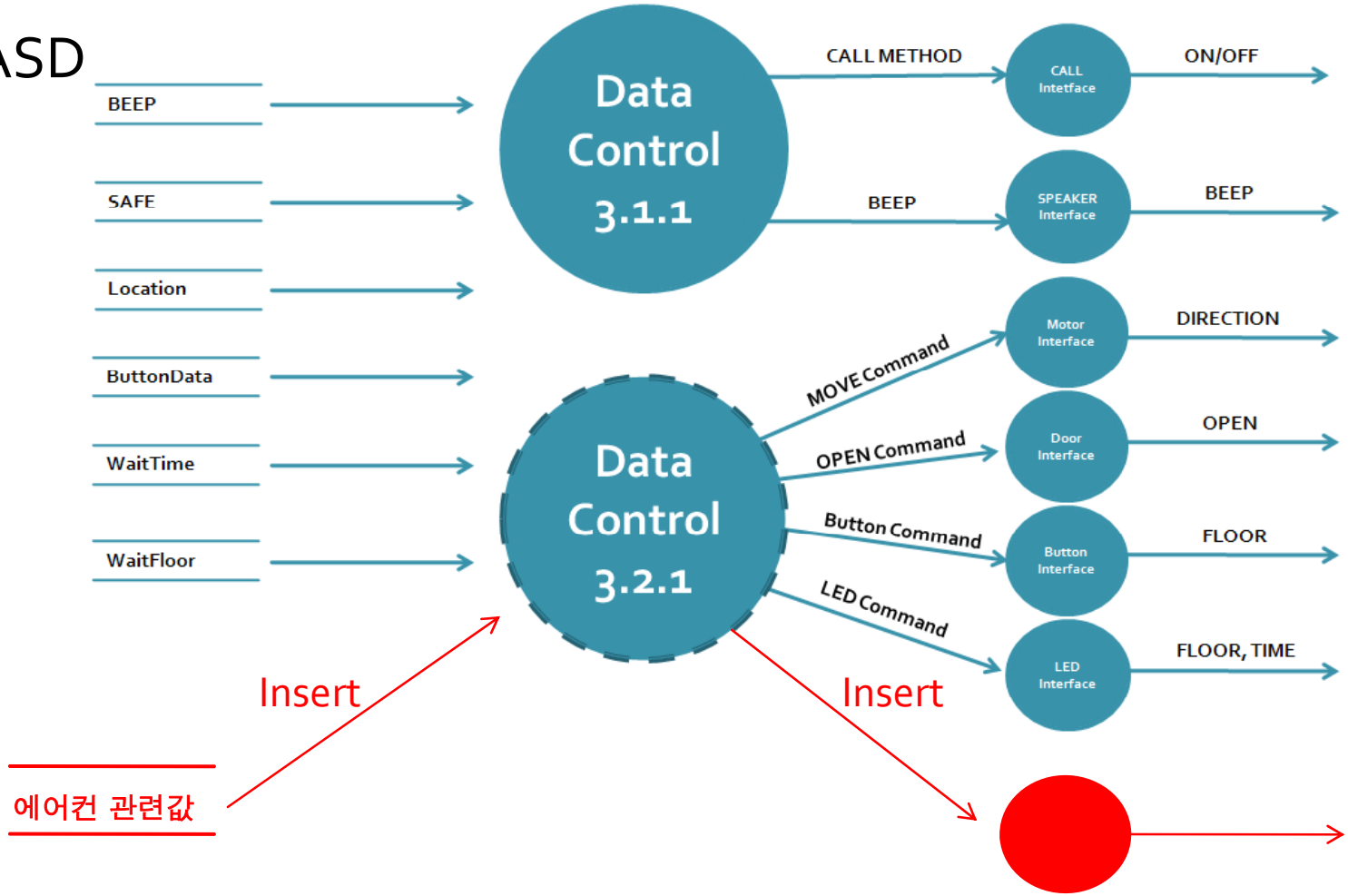
- OOAD

-Just, define a class or function related AirCondition and is added in the main controller.



4.OOAD vs SASD - Revision

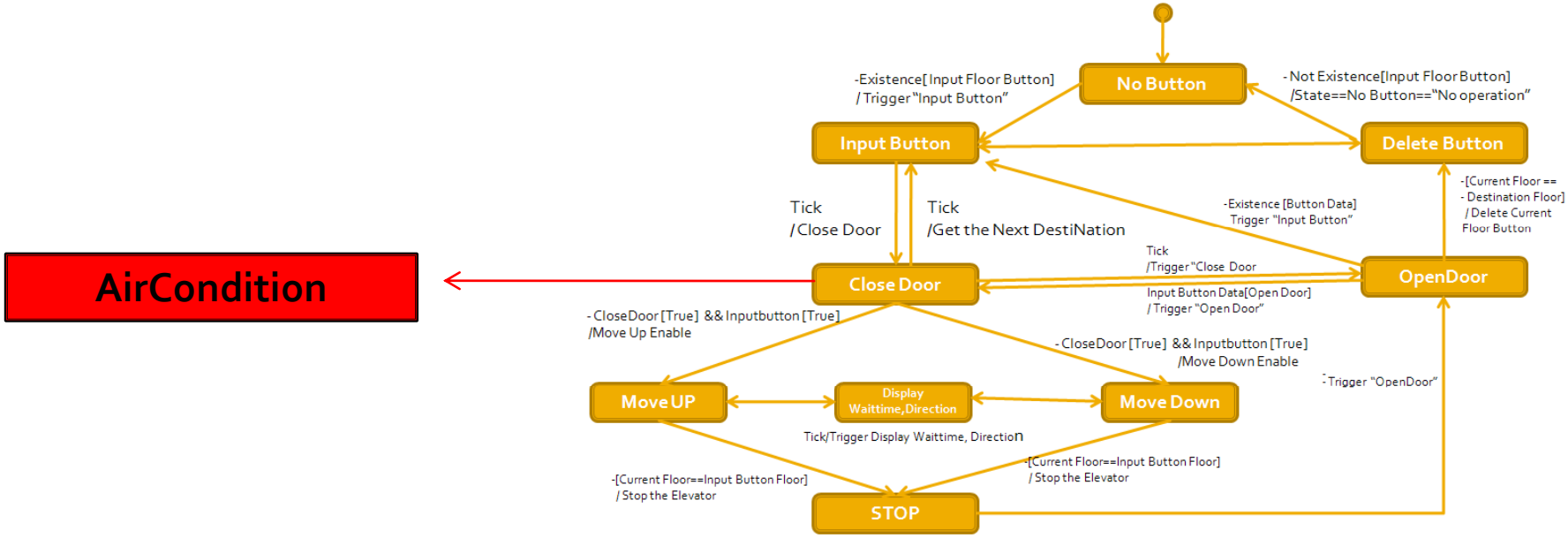
- SASD



- Air conditioning all the information you need to be rewritten from beginning to end.

4.OOAD vs SASD - Revision

- SASD – State Diagram



- State Diagram of state for air-conditioning-related, even if this requires modifying

4.OOAD vs SASD - Revision

- If OOAD or SASD need to modify any part?

- OOAD : In this case, it can modify any part by making a new cont rol class of air-conditioner and constructing a method to control a ir conditioner class. This process rarely require any other specific step.

- SASD : Due to the initially inserted data which is controlling an ai r-conditioner ,it is required to modify all of the SASD processes b efore the air-conditioner is actually controlled. Depending on a n ecessity, it is required to modify not only DFD Level 1-4 but also
- all parts like State Diagram or Structured Chard, etc. If it happens not in the planning step but in the actual developing process or even after the development, the cost for modifying process will be more expensive than that of OOAD.

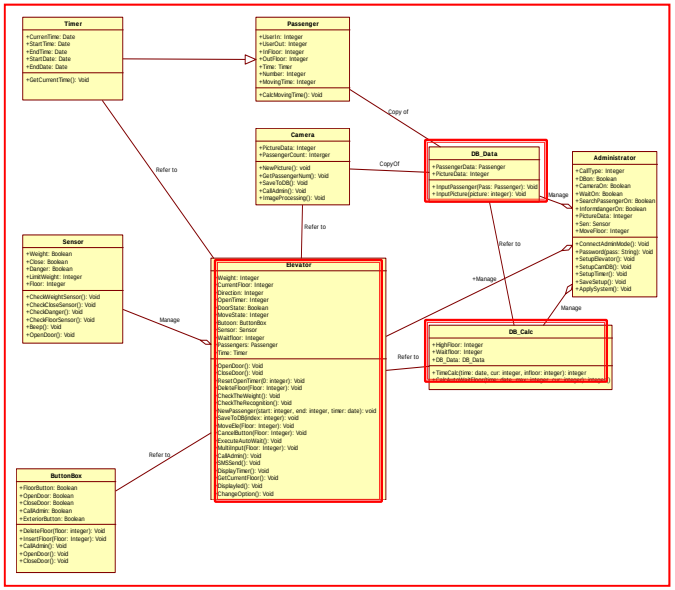
4.OOAD vs SASD - Incorrect DATA

- Results Was different from expected results during the development process. if you want to it traced?

- For Example, We created the elevator controller and the automatic standby feature, the behavior is wrong. Where did it wrong ?????

4.OOAD vs SASD - Incorrect DATA

- OOAD

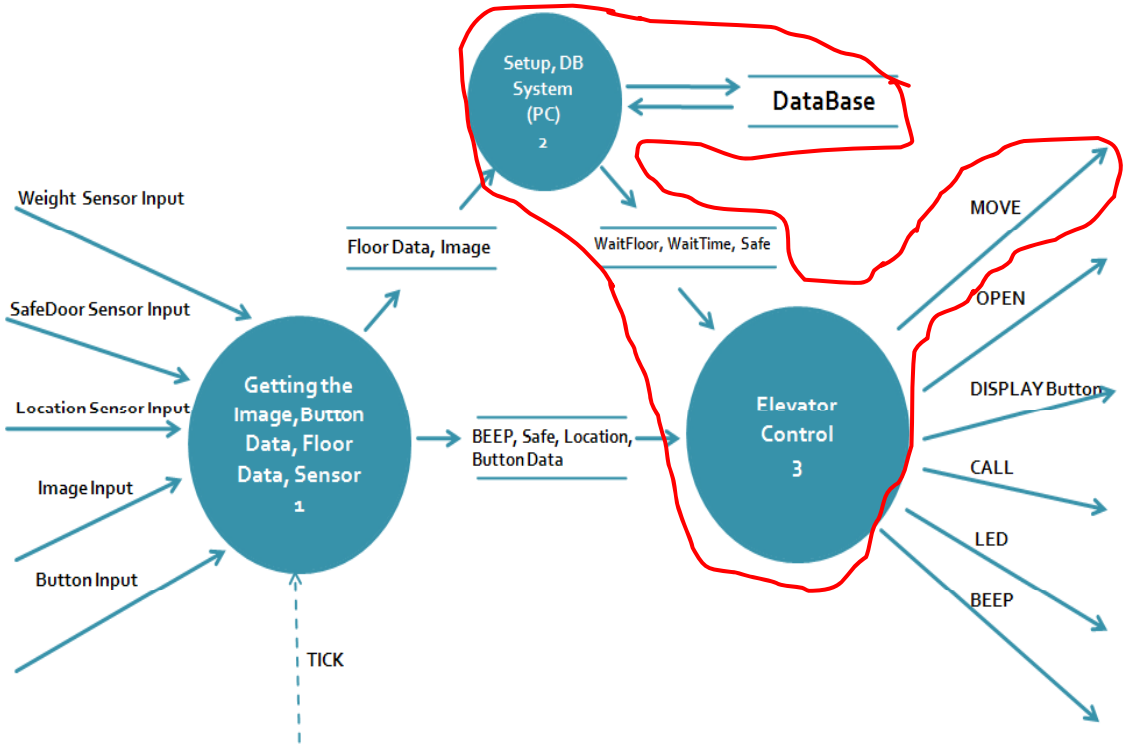


-It's hard to distinguish for which method of which class to access Variable of Auto wait. It seems like that how and where the data is defined and used is not much cared in OOAD

-If it's not your own project, it will be hard to analyze the referred class.

4.OOAD vs SASD - Incorrect DATA

- SASD



-Through out the SASD, all of the data flow from the starting point to the output point can be seen clearly.

4.OOAD vs SASD - Incorrect DATA

- In the OOAD, it is hard to detect where the date is used and which part makes the error. But in the SASD, it's more easier to find out where was missing data and the error by tracking the pathway data has followed , because following the data flow is possible.

4.OOAD vs SASD - Complexity

- Which one is better to programming between OOAS and SASD in the aspect of simplicity and understandability?

- OOAD: To make actual elevator controller, it will be required to make more classes(objects) and to analyze the data depending on their function by their each class. It seems like inefficient in the simple programming like elevator controller.

- SASD: It just requires to develop the process to send out the output according to the input by checking the each data's flow. So it would be more efficient way to develop the elevator controller.

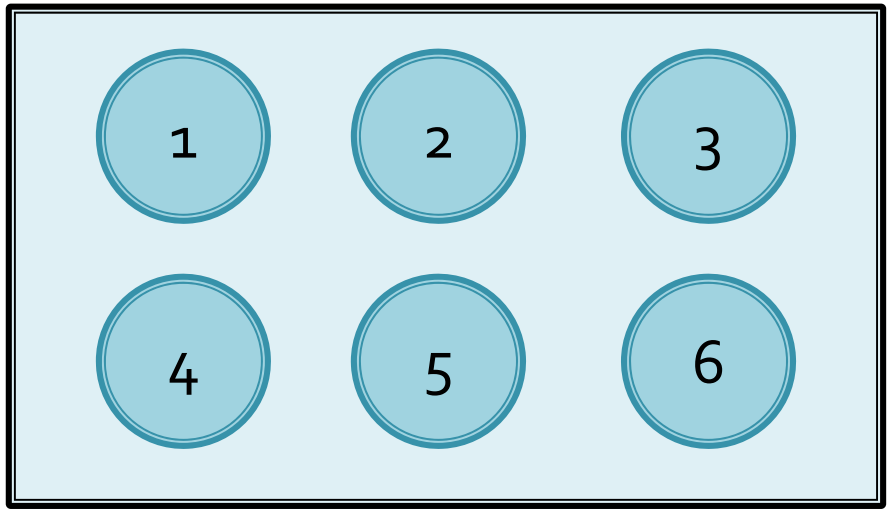
4.OOAD vs SASD - Complexity

- But! Let's think about the so complicated case like shopping mall which has uncountable flowing data.
  In this case, which one will be better?

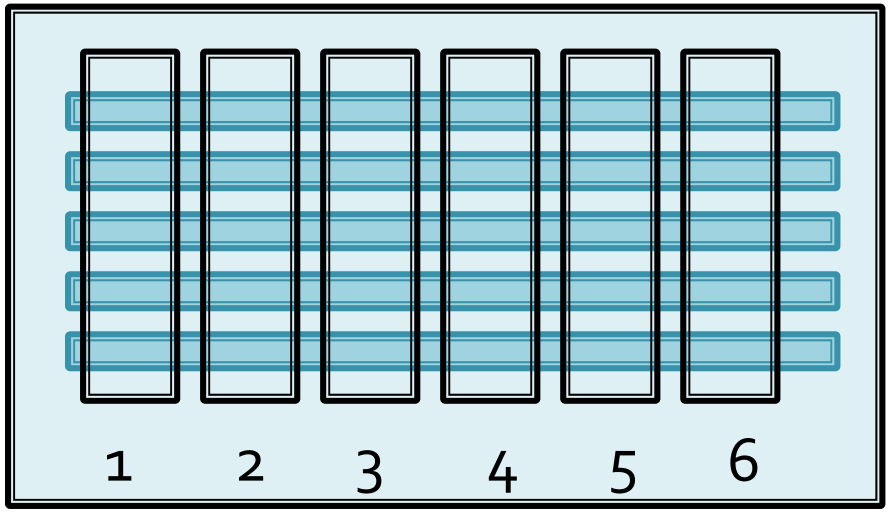
- It's true that SASD is easy to check clearly the data flow . But if there is too much flowing data, the rate that SASD actually can track the flow will be considerably decreased and it will be hard to decide which input makes the following output because of the amount of data. Otherwise, in the OOAD case, it can have steady clarity by grouping the date which have similar characteristics .

4.OOAD vs SASD - Development

What is the difference between OOAD and SASD when you develop actual plan?
OOAD is like assembling the parts developed by each class.
SASD is like gradually developing all parts at the same time.



[OOAD]



[SASD]

*Each number means development stop

-This development set the order of the Classes of OOAD division easily can a while ago,
-but the division of labor that it will SASD.

4.OOAD vs SASD - Conclusion

- 엘리베이터 컨트롤러를 위한 OOAD 기법과 SASD 기법 모두를 통해 설계를 해보았다. 엘리베이터 컨트롤러의 설계는 체감상 내가 개발자라고 했을때 SASD의 개발 방법을 따르는 것이 좀 더 효율 적이라고 생각 했다. 입출력 되는 데이터 및 state다이어그램의 복잡성이 심하지 않고 데이터의 오류 및 검증이 민감한 엘리베이터에 좀더 효율 적이라고 생각 된다.

- 사실 OOAD와 SASD 중 무엇이 더 좋은 설계 방법인가는 양 설계 방법 모두 장,단점이 있으므로 섣불리 얘기 할 수 없다. 하지만 중요한 것은 프로젝트의 목표 및 규모 특징 등을 잘 파악 해서 OOAD와 SASD의 설계 방법을 선택 해야 할 것이다.

4.OOAD vs SASD - Conclusion

- 데이터의 입출력이 많고 재사용성을 필요로 하거나 규모가 상당히 커서 분업화가 필요한 프로젝트의 경우에는 OOAD를 쓰는 것이 효율적일 것이고 데이터의 흐름 및 정교한 프로그래밍을 위해서는 SASD가 더 효율적일 것이다.

- SASD : 임베디드 시스템, 실시간 데이터 처리 시스템 등에 유용하게 쓰일수 있다.

- OOAD : 웹 시스템, 대규모 프로젝트, 데이터가 많은 시스템에 유용 하게 쓰일수 있다.

- 혹시 이런것은 어떨까???

4.OOAD vs SASD - Idea!!!!!????

- If OOAD + SASD ????

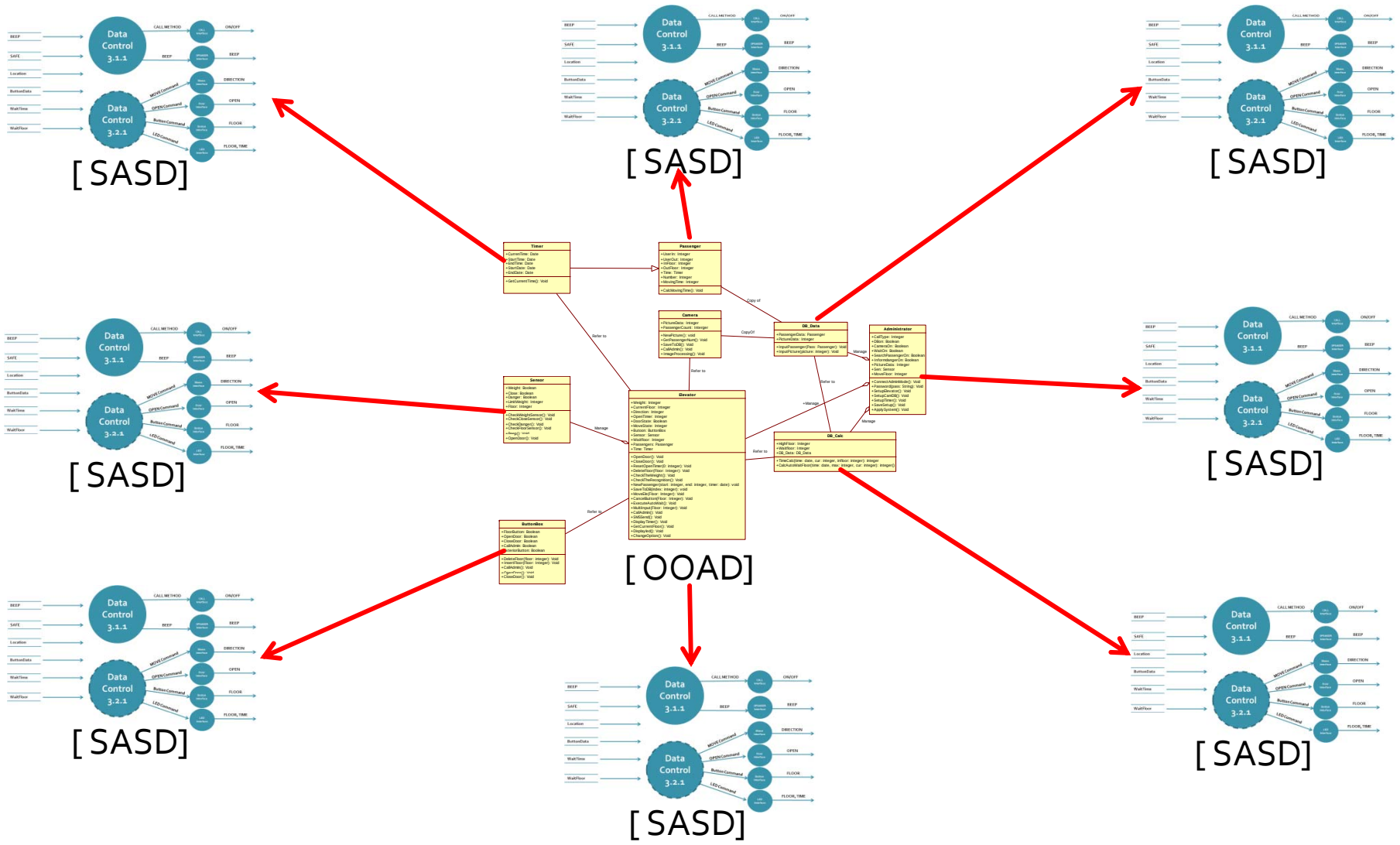
  - Combining the advantages of the benefits of OOAD and SASD do not can the make the new design?

  - For example, If there is a large-scale project design, OOAD is defined the relationship between the classes. And The data inside the class, SASD is designed input and output.
    How about use the double method of OOAD and SASD?

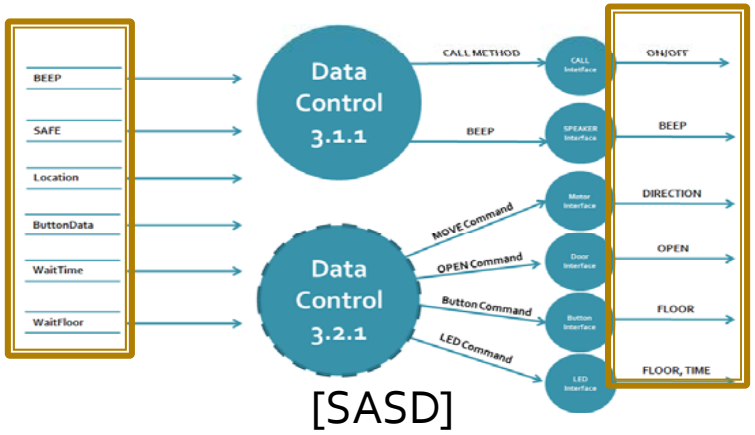
  - Of data tracking, development, maintenance, division of labor, and might be able to satisfy all that?

  - It will not be easy.....

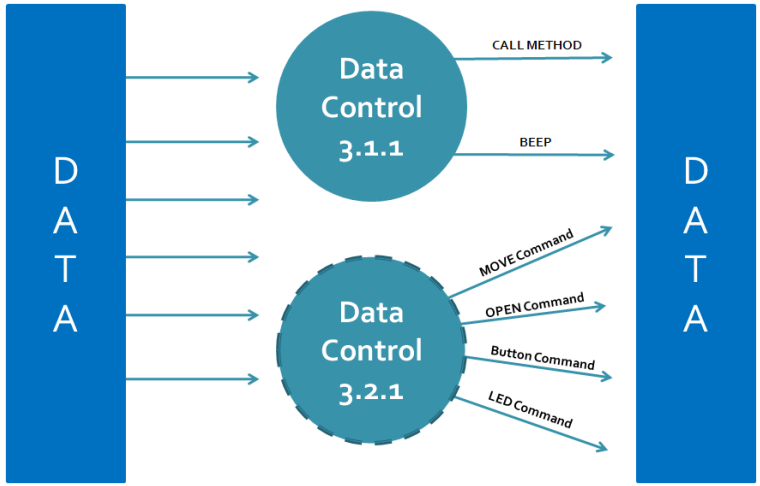
4.OOAD vs SASD - Idea!!!!!????



4.OOAD vs SASD - Idea!!!!!????



[SASD]



-The place where the output and input is not hardware. this place is function or variable.

5.Summary

- 입맛대로 골라써라
- 엘리베이터 컨트롤러 같이 간단한 시스템은 SASD가 더 좋은거 같다
- 과연 소프트웨어 검증까지 들어야 하는가

The end
Really end
Thank you for Listening