

Object Oriented Methodology

Week – 6, Lecture – 1
Introduction to UML

SAURABH SRIVASTAVA
VISITING FACULTY
IIIT LUCKNOW

What if you were supposed to plan a trip?

Say a class trip for your batch – of course, provided that COVID recedes :-/

Assume that a *set of requirements* are provided to you already

- Say you are going to Goa (seems to be the cliché among class trip planners :D)
- ... and that there are some budgetary constraints to keep the cost per traveller to a maximum value

How will you go about your work?

- You can either do it the traditional way, i.e., chaotic and filled-up with last minute adjustments :D
- Or, you can do it the boring way, i.e., by chalking out the details systematically, evaluating all options !!
- Let us assume that you like the boring way ...

This will involve planning and documentation

- So what documents shall you prepare?

Structural vs Behavioural Details (1/2)

Probably the first set of information you will require will be about routes

- One aspect of this will be picking the mode of travel – Air, Train, Road or a combination of these
- For a given mode, there may be multiple options – probably with different costs

This information, in general, is *static* in nature ...

- ... i.e., it is highly unlikely that this information changes during your travel ...
- ... e.g., travel by air would probably still be costlier than travel by train
- Such information will capture the *structural* aspect of the trip

Structural aspect of a solution shows a sketch of its different elements and their connections

Structural vs Behavioural Details (2/2)

The other set of details you will have to figure out are dates and times

- For instance, what date and time will the journey start?
- Where will you pause for refreshments?
- If you face any issues during the trip, like a flat tyre or a delayed flight, how will you handle them?

This type of planning requires imagining the trip in real-time ...

- In essence, your plans capture the *dynamic* aspects of your trip

If you consider the trip as a mission, the previous set of information relates to its structure ...

- ... whereas the next set of information depicts its nature or *behaviour*

Behavioural aspect of a solution presents various aspects of the solution during its execution

Communicating Solution Details

Assume that you've planned the trip like a pro !!

- How will you communicate the details to all the travellers?

You can do it the way the government agencies work ...

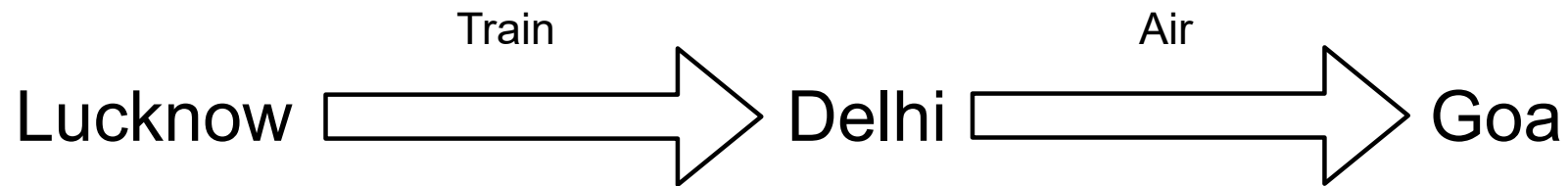
- ... i.e. sending out multiple-paged PDFs or long emails

Or, you can send well-crafted graphical content, making it easier to understand and remember

Textual vs Graphical Information

1. We will travel from Lucknow to Delhi via train.
2. We will travel from Delhi to Goa by air.

vs



Communicating Solution Details

Assume that you've planned the trip like a pro !!

- How will you communicate the details to all the travellers?

You can do it the way the government agencies work ...

- ... i.e. sending out multiple-paged PDFs or long emails

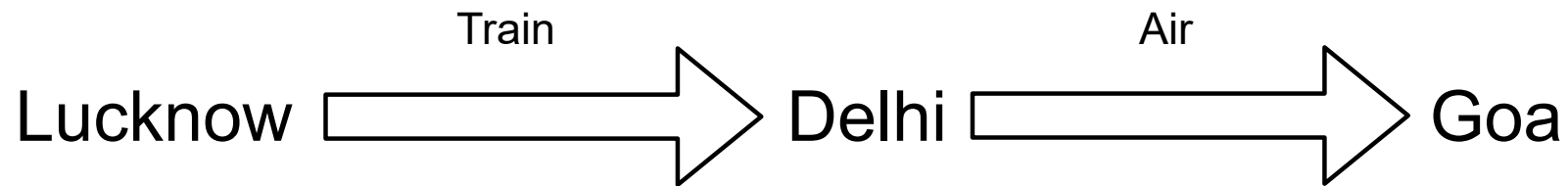
Or, you can send well-crafted graphical content, making it easier to understand and remember

- The problem with this mode is “universal understanding the graphical semantics”

Textual vs Graphical Information

1. We will travel from Lucknow to Delhi via train.
2. We will travel from Delhi to Goa by air.

vs



For example, the arrow and the text over the arrow here, has some *implied* meanings

Communicating Solution Details

Assume that you've planned the trip like a pro !!

- How will you communicate the details to all the travellers?

You can do it the way the government agencies work ...

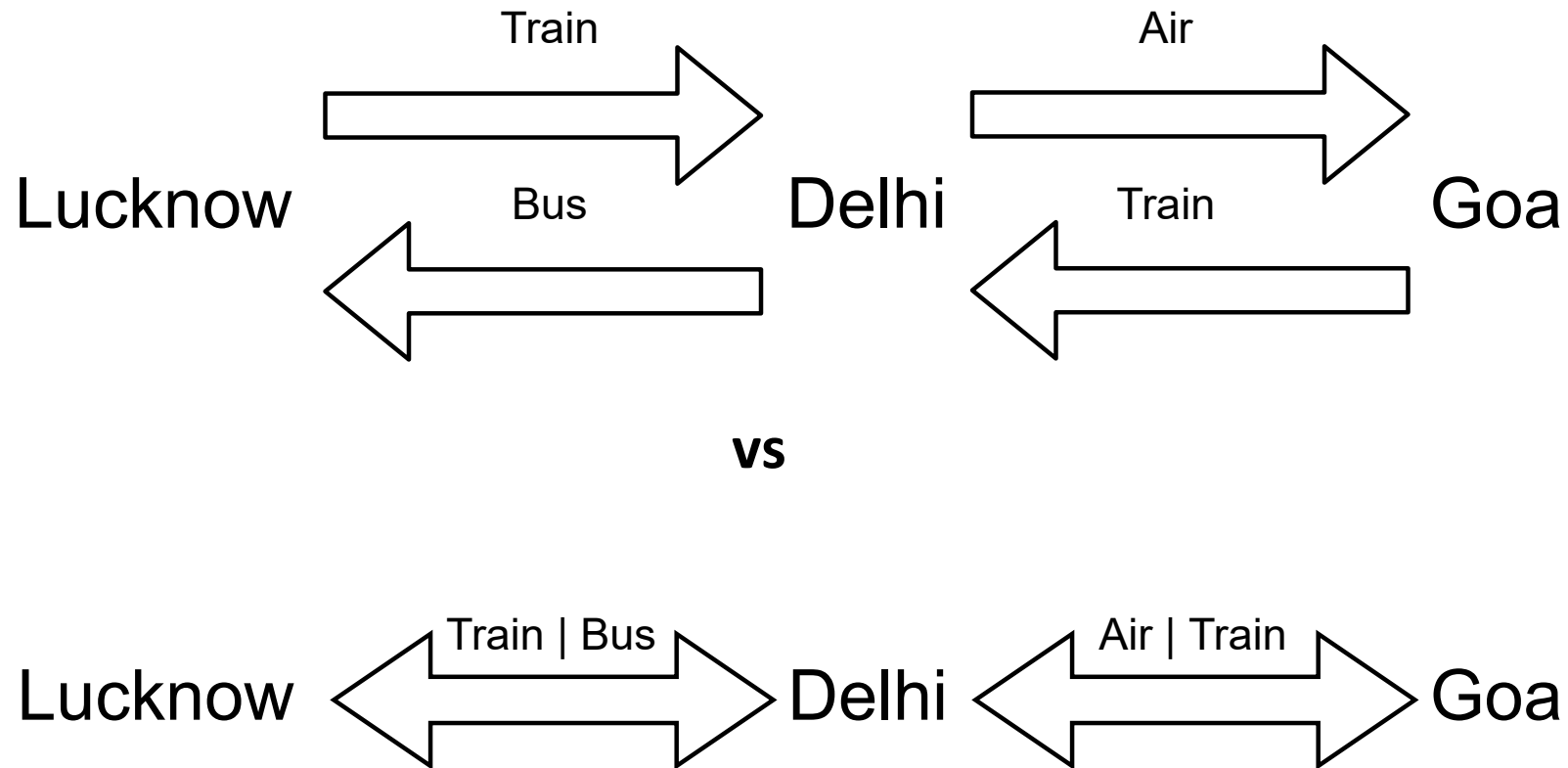
- ... i.e. sending out multiple-paged PDFs or long emails

Or, you can send well-crafted graphical content, making it easier to understand and remember

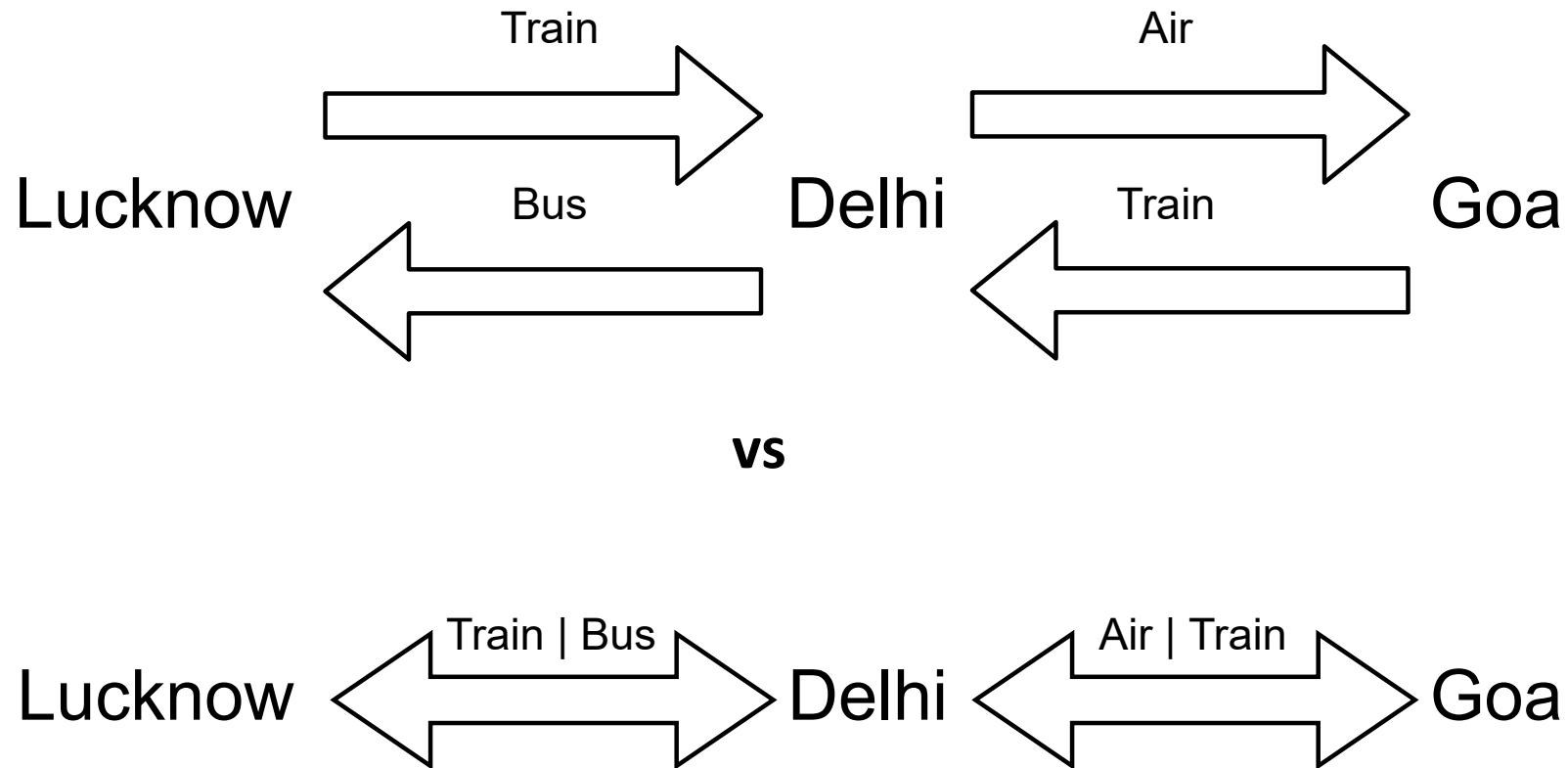
- The problem with this mode is “universal understanding the graphical semantics”

Without a well-understood set of conventions, the graphical mode may become a liability too

Textual vs Graphical Information



Textual vs Graphical Information



These two representations attempt to convey the same meaning, but are they *equivalent*?

Communicating Solution Details

Assume that you've planned the trip like a pro !!

- How will you communicate the details to all the travellers?

You can do it the way the government agencies work ...

- ... i.e. sending out multiple-paged PDFs or long emails

Or, you can send well-crafted graphical content, making it easier to understand and remember

- The problem with this mode is “universal understanding the graphical semantics”

Without a well-understood set of conventions, the graphical mode may become a liability too

Thus, a set of conventions, which provide explicit meanings to graphical elements is crucial

This is where UML comes in, and plays a pivotal role

The Universal Modelling Language

As you have started working on projects (at least I hope you have), you too will face this dilemma

- Whether to use textual descriptions or graphical conventions to convey aspects of your system?

If you choose the latter, what boxes and lines should your diagrams have?

- Do you need to supply a *key* with them, or is it that the meaning is implicitly implied?

The Universal Modelling Language or UML provides a set of conventions for such cases

UML consists of many different diagrams, with each one having a set of predefined conventions

Thus, if the person viewing your diagrams understands UML, your job becomes fairly easy ...

- ... all you need to do is create your diagrams following UML conventions, and you are good

Also, if the conventions are followed precisely, it may also be easier to convert diagrams to code

- This process, of converting UML models to code, is an active research topic !!

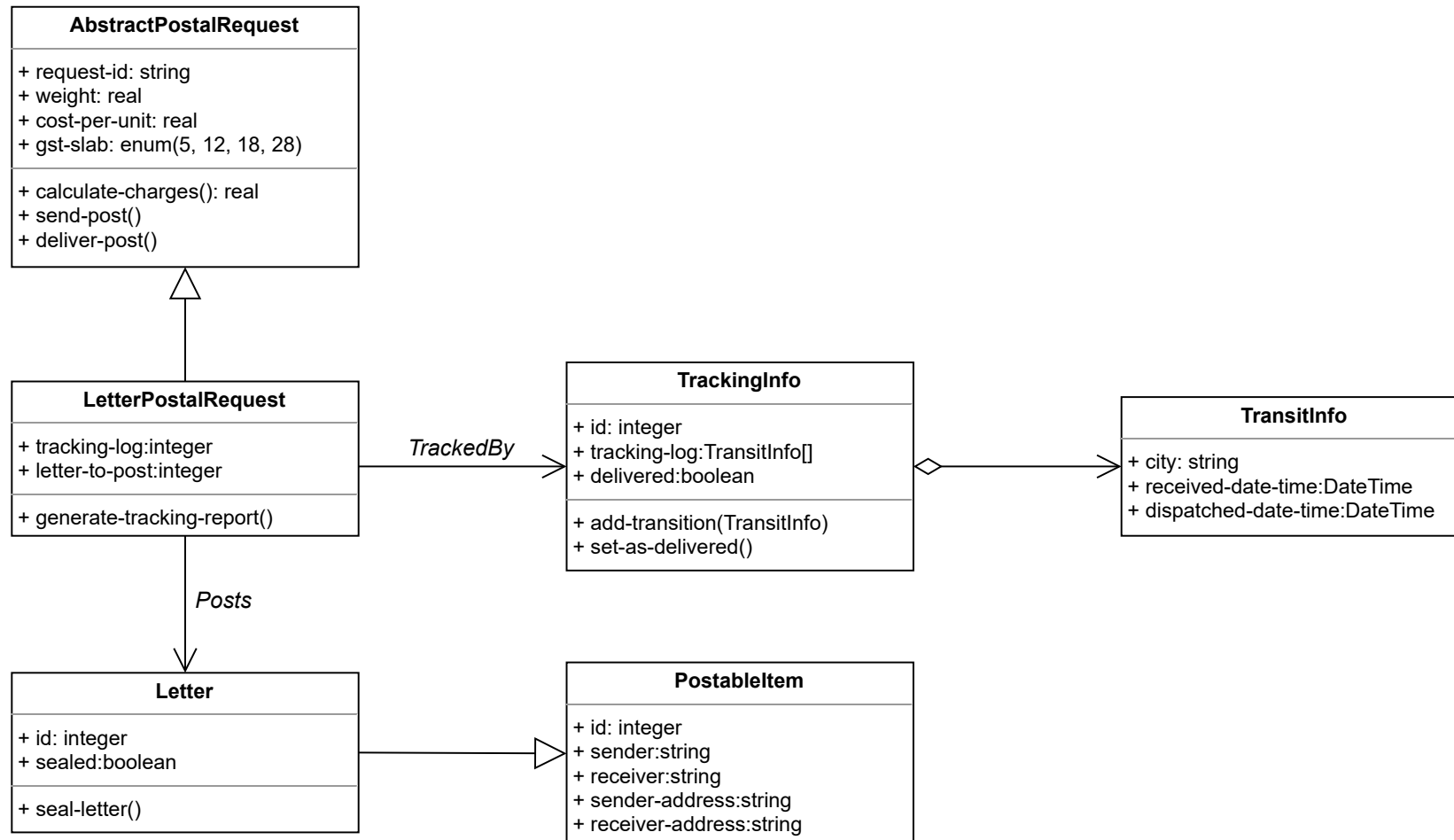
Types of UML Diagrams

The UML diagrams can be roughly classified into two categories – Structural and Behavioural

Structural Diagrams describe the entities of a system, as well as how they relate to each other

- Some examples of Structural Diagrams are provided below
- **Class Diagrams** describe the details of the classes in the system, along with their relationships
- **Package Diagrams** show the constituents of, and relationships between, sub-systems of the system
- **Deployment Diagrams** depict the deployment of the elements of a system over physical infrastructure

Example Class Diagram (from Lecture 2.1)



Types of UML Diagrams

The UML diagrams can be roughly classified into two categories – Structural and Behavioural

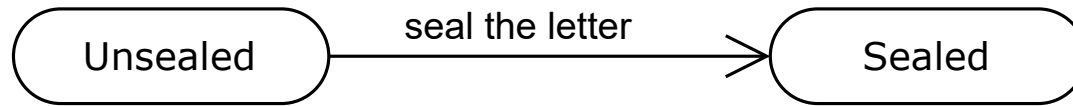
Structural Diagrams describe the entities of a system, as well as how they relate to each other

- Some examples of Structural Diagrams are provided below
- **Class Diagrams** describe the details of the classes in the system, along with their relationships
- **Package Diagrams** show the constituents of, and relationships between, sub-systems of the system
- **Deployment Diagrams** depict the deployment of the elements of a system over physical infrastructure

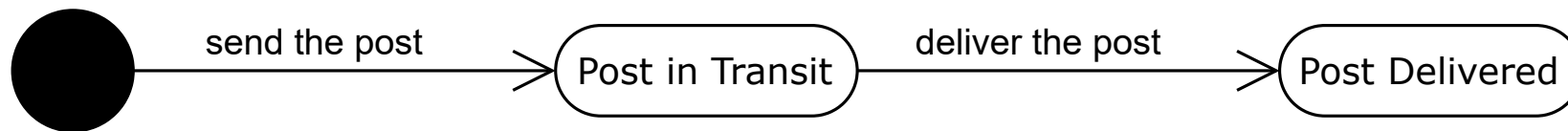
Behavioural Diagrams describe the interactions between the elements of a system at runtime

- Some examples of Behavioural Diagrams are provided below
- **Use case Diagrams** showcase the end-to-end expectations from a system for specific usage profiles
- **Sequence Diagrams** show a timeline of important events of the system along with the associated objects
- **State Machine Diagrams** present a view of the system or a sub-system, comprising of states and transitions

Example State Diagram (from Lecture 2.1)



State Diagram for Letter



State Diagram for AbstractPostalRequest

Types of UML Diagrams

The UML diagrams can be roughly classified into two categories – Structural and Behavioural

Structural Diagrams describe the entities of a system, as well as how they relate to each other

- Some examples of Structural Diagrams are provided below
- **Class Diagrams** describe the details of the classes in the system, along with their relationships
- **Package Diagrams** show the constituents of, and relationships between, sub-systems of the system
- **Deployment Diagrams** depict the deployment of the elements of a system over physical infrastructure

Behavioural Diagrams describe the interactions between the elements of a system at runtime

- Some examples of Behavioural Diagrams are provided below
- **Use case Diagrams** showcase the end-to-end expectations from a system for specific usage profiles
- **Sequence Diagrams** show a timeline of important events of the system along with the associated objects
- **State Machine Diagrams** present a view of the system or a sub-system, comprising of states and transitions

We will study some of these diagrams in detail over the next few weeks

Drawbacks of UML

Although UML certainly has its advantages, it does have its set of demerits

UML has a strong bias towards Object-Oriented Methodology

- If the system has elements that are not Object-Oriented in nature, it becomes difficult to represent them
- A typical example is the presence of global functions, which are not part of any class

UML maybe an overkill for relatively simpler projects

- For such projects, following UML conventions may mean unnecessarily longer design times
- You will realise it soon enough ;)

Most people do not follow UML conventions strictly

- Frankly, it may be too overwhelming to remember all the conventions anyway

The code generation capabilities of UML-based tools are still fairly limited

- The sheer number of design choices available with the designer may pose a daunting task

Homework !!

There are numerous resources associated with UML on the internet

- Pick up some resource and try to get an overview of UML
- Some nice starting points are given below:

<https://www.youtube.com/watch?v=WnMQ8HlmeXc>

<https://sparxsystems.com/resources/tutorials/uml2/>