



**SOFTWARE
ARCHITECTURE**

2024-25

Jose E. Labra
Pablo González
Irence Cid
Diego Martín



Escuela de
Ingeniería
Informática



Universidad de Oviedo

Lab 2

Overview of UML
PlantUML
Introduction to Arc42

UML

Unified Modeling Language

Before UML there were several proposals

UML notation unifies them

Proposed by OMG (Object Management Group)

Current version UML 2.5.1 (2017)

Model = abstraction of a problem

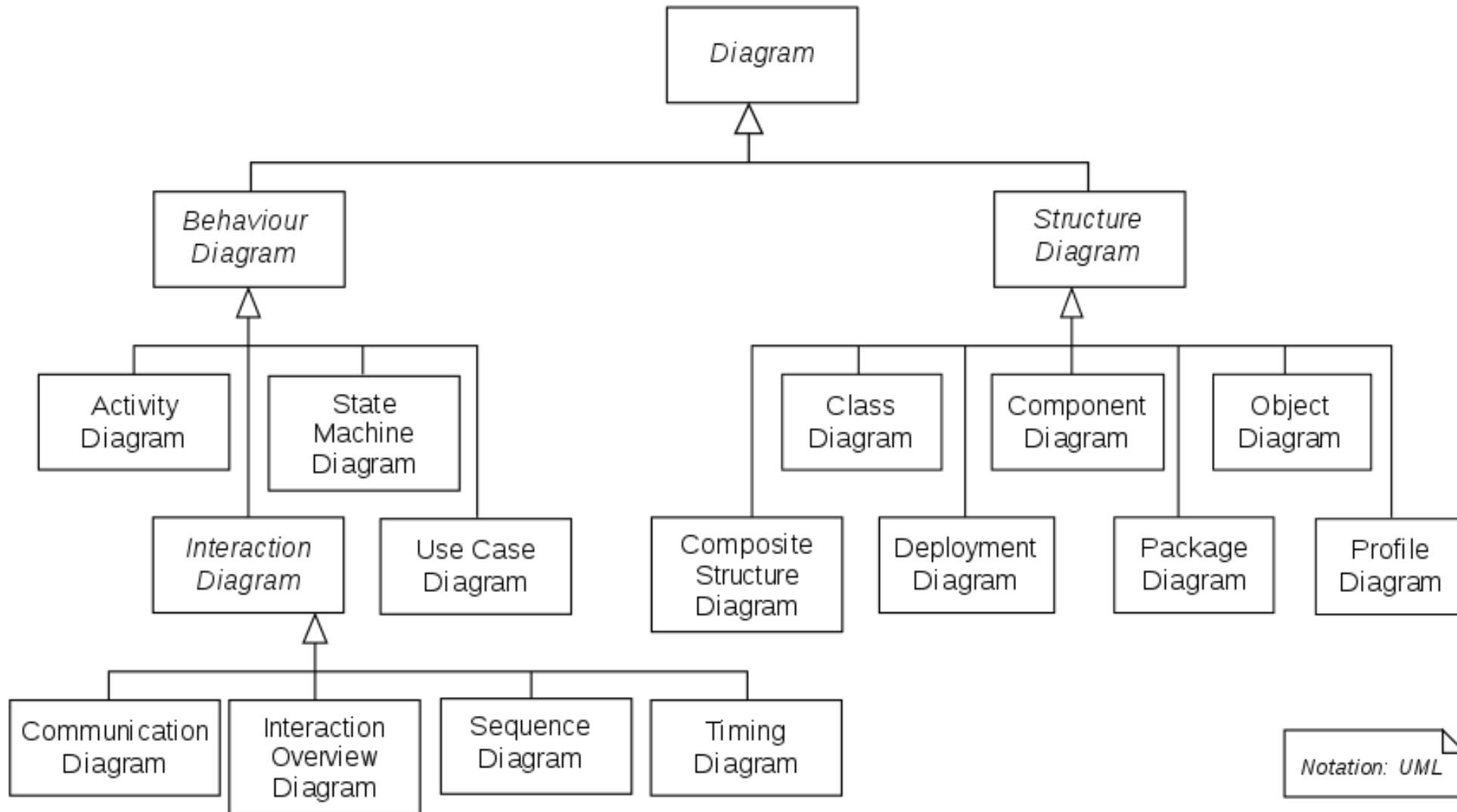
It can have different diagrams

Diagram = partial graphic representation of a model

OCL = Object Constraint Language

Constraints between objects using formal language

14 UML Diagram types



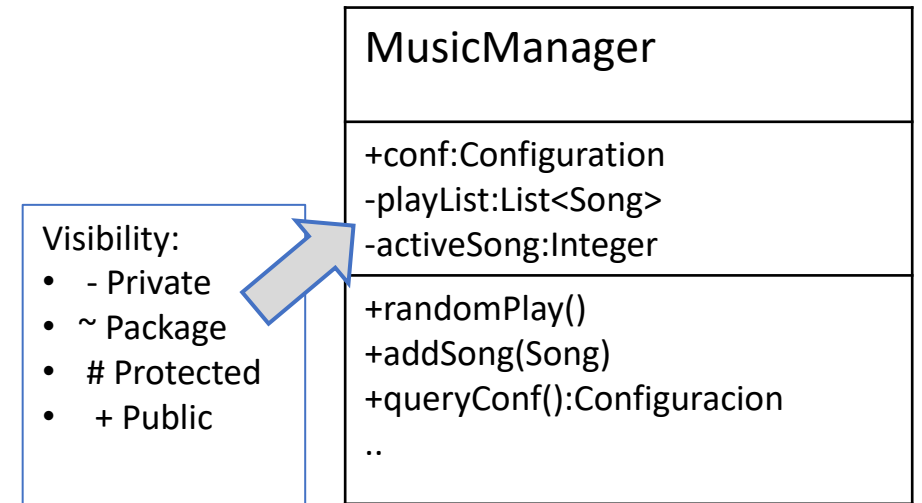
Notation: UML

Class diagrams

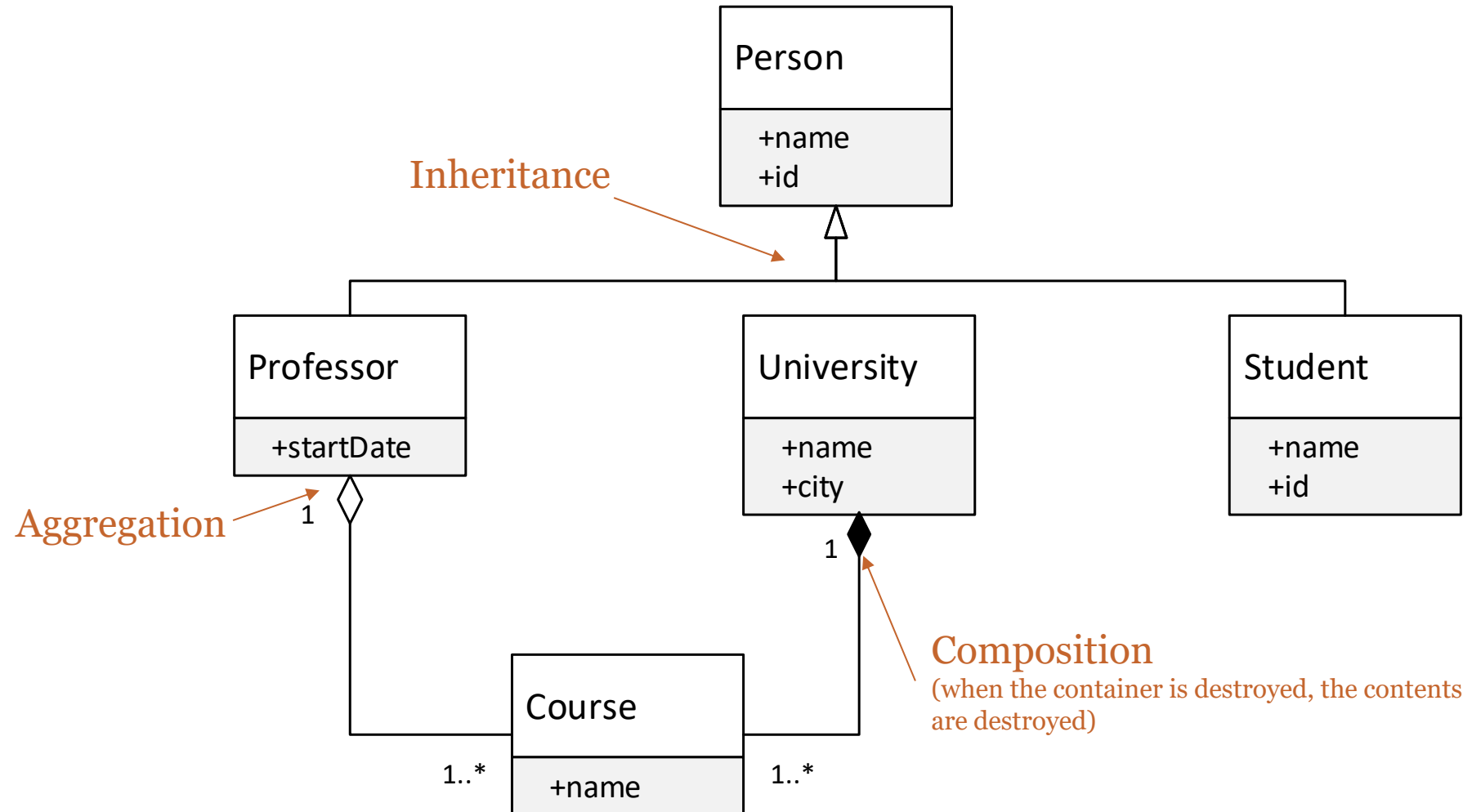
Models the static part of the project, without taking into account the time aspect

Explains the relationships between the different classes.

Arc42: 8-Concepts

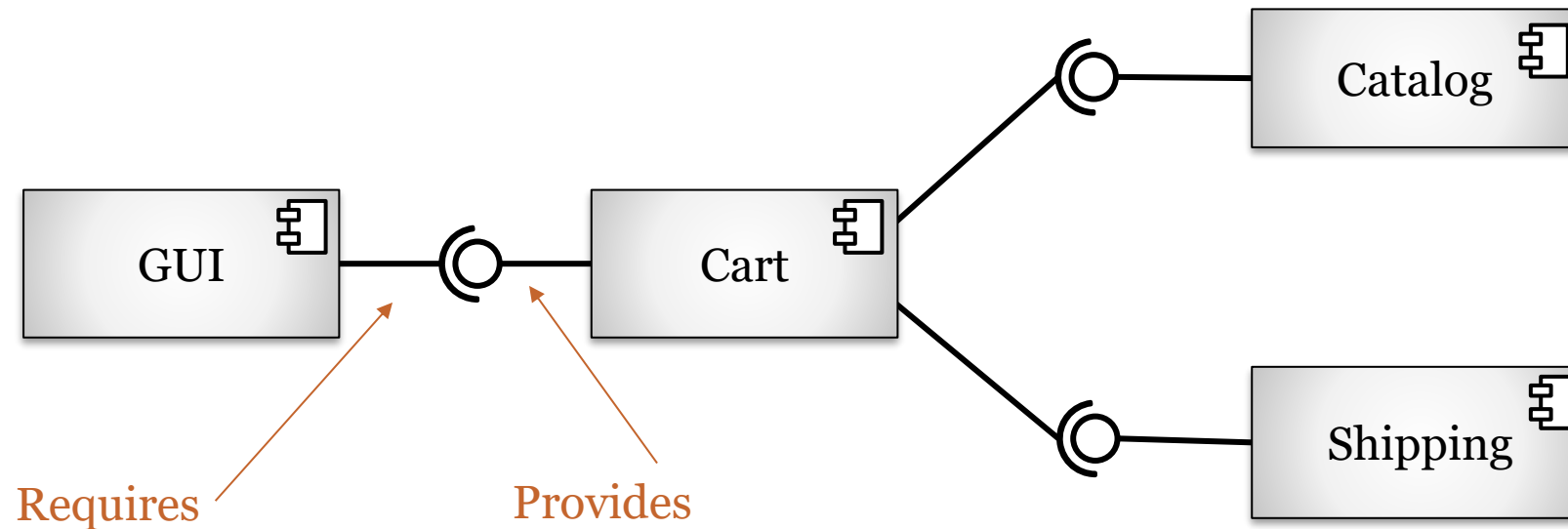


Example



UML Component diagram

Component diagram represents components relationships
Useful for Complex Systems with many components
Interface is usually represented with lollipop notation

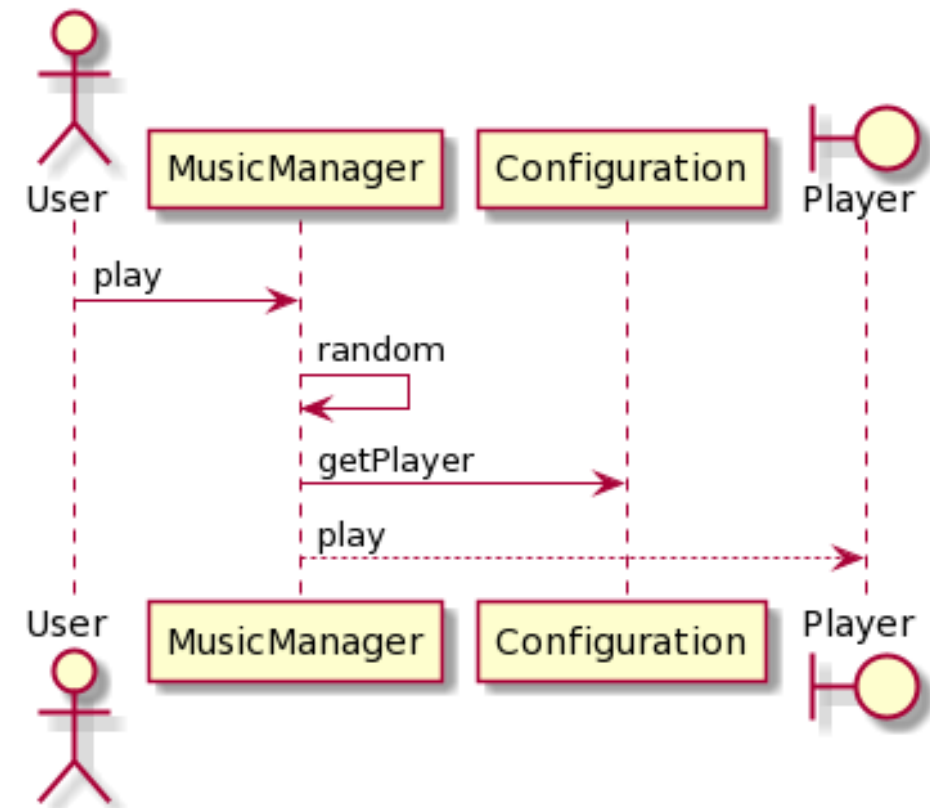


Sequence diagram

Models communication between some objects at a given time

Objects can send two types of messages: synchronous or asynchronous

Arc42:6-RuntimeView



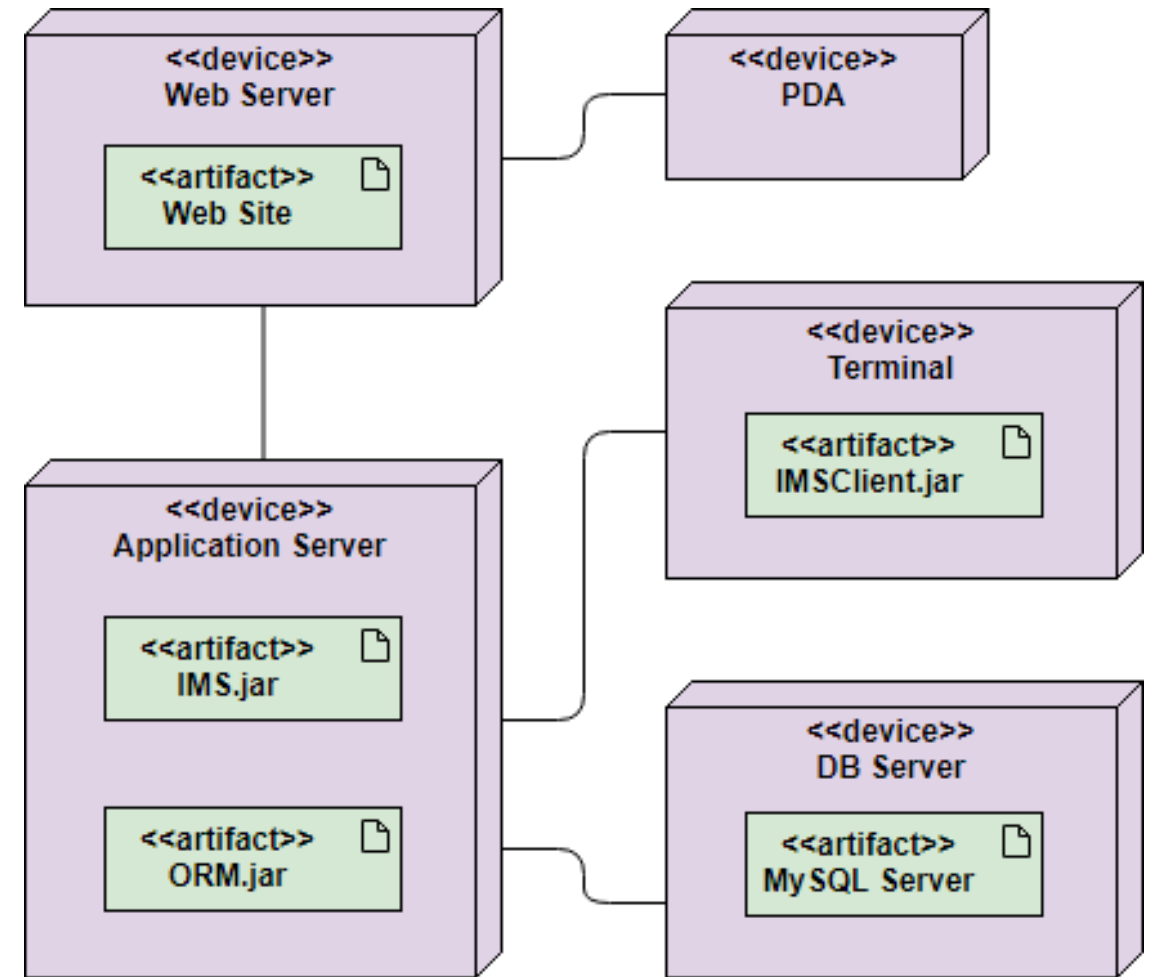
Deployment diagrams

Represents the final location of the components in an app

Elements:

Nodes , Components, relationships

Arc42: 07.DeploymentView



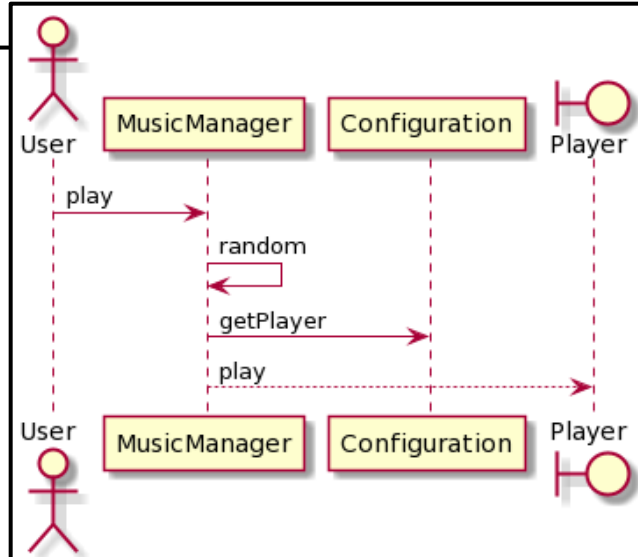
Text-based tools

PlantUML

```

@startuml component
actor User
participant MusicManager
participant Configuration
boundary Player
User -> MusicManager: play
MusicManager --> MusicManager: random
MusicManager --> Configuration : getPlayer
MusicManager -.-> Player : play
@enduml

```

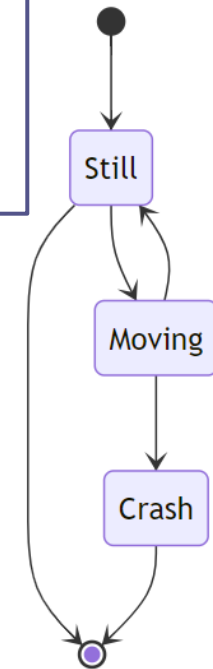


Mermaid

```

stateDiagram-v2
[*] --> Still
Still --> [*]
Still --> Moving
Moving --> Still
Moving --> Crash
Crash --> [*]

```



Drawing tools

Powerpoint

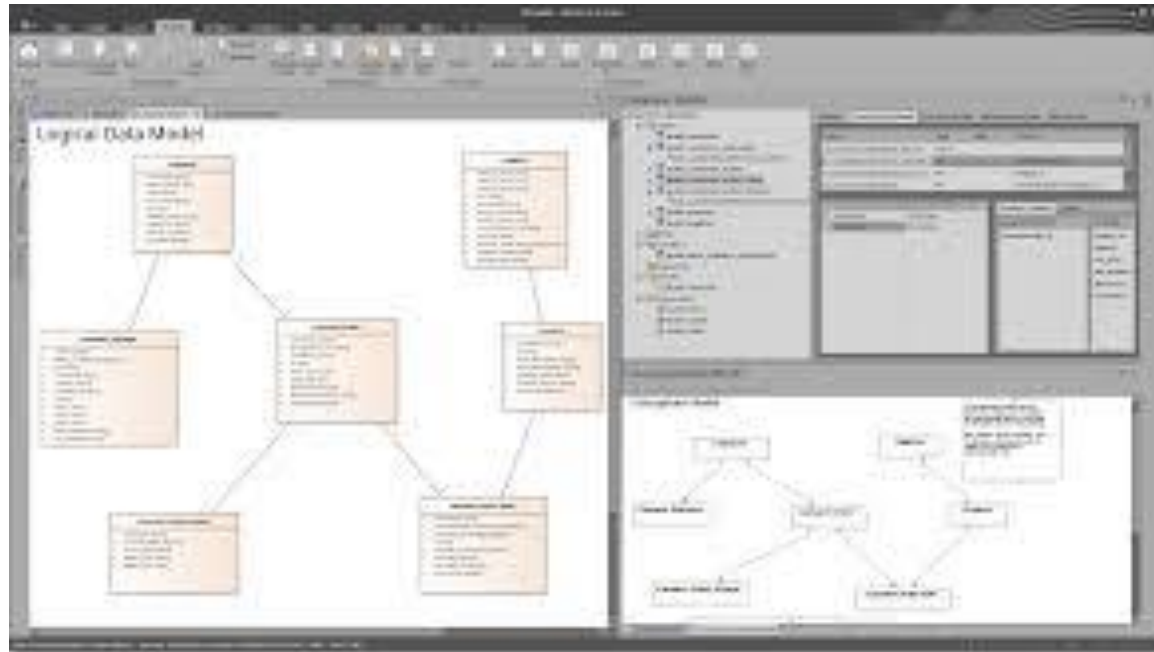
MsVisio

UMLet (<https://www.umlet.com/>)

CASE tools

EnterpriseArchitect

- Reverse Engineering with Java/C++
- Oracle connection for relational databases
- Word, HTML templates



MagicDraw

- Java based
- UML diagrams
- Reverse Engineering Java , C++

Visual Paradigm

- Commercial (student license)

Modelio

- Open source
- Java based
- Reverse Engineering Java , C++

Diagramming the architecture

Video:

<https://www.youtube.com/watch?v=wgpSdpny-0c>

Checklist:

<https://c4model.com/assets/software-architecture-diagram-review-checklist.pdf>

Arc42 templates

Arc42

<https://arc42.org/>

WIQ already follows the template:

https://arquisoft.github.io/wiq_0/

Generation of docs (locally):

```
$ cd docs  
$ npm install      (only first time)  
$ npm run build
```

GitHub Pages

GitHub supports creating websites

Useful for personal – project/repository

Branch **gh-pages**

GitHub Pages - examples

Organization level

Repository:

<https://github.com/Arquisoft/Arquisoft.github.io>

Deployed:

<https://arquisoft.github.io/>

It can be very useful for personal web pages

<http://pglez82.github.io>

Documentation deployment

Documentation is deployed using GitHub Pages

GitHub Pages allows users to publish a simple website directly on GitHub

Generated website will be pushed to the branch **gh-pages**

npm package **gh-pages** pushes doc website to gh-pages branch

Everything is automatized with the following command:

```
$ npm run deploy
```

