SOFTWARE ARCHITECTURE

Lab 2

Ovierview of UML
PlantUML
Introduction to Arc42

2023-24

Jose Emilio Labra Gayo
Pablo González
Cristian Augusto Alonso
Jorge Álvarez Fidalgo

Escuela de Ingeniería Informática

Universidad de Oviedo

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)

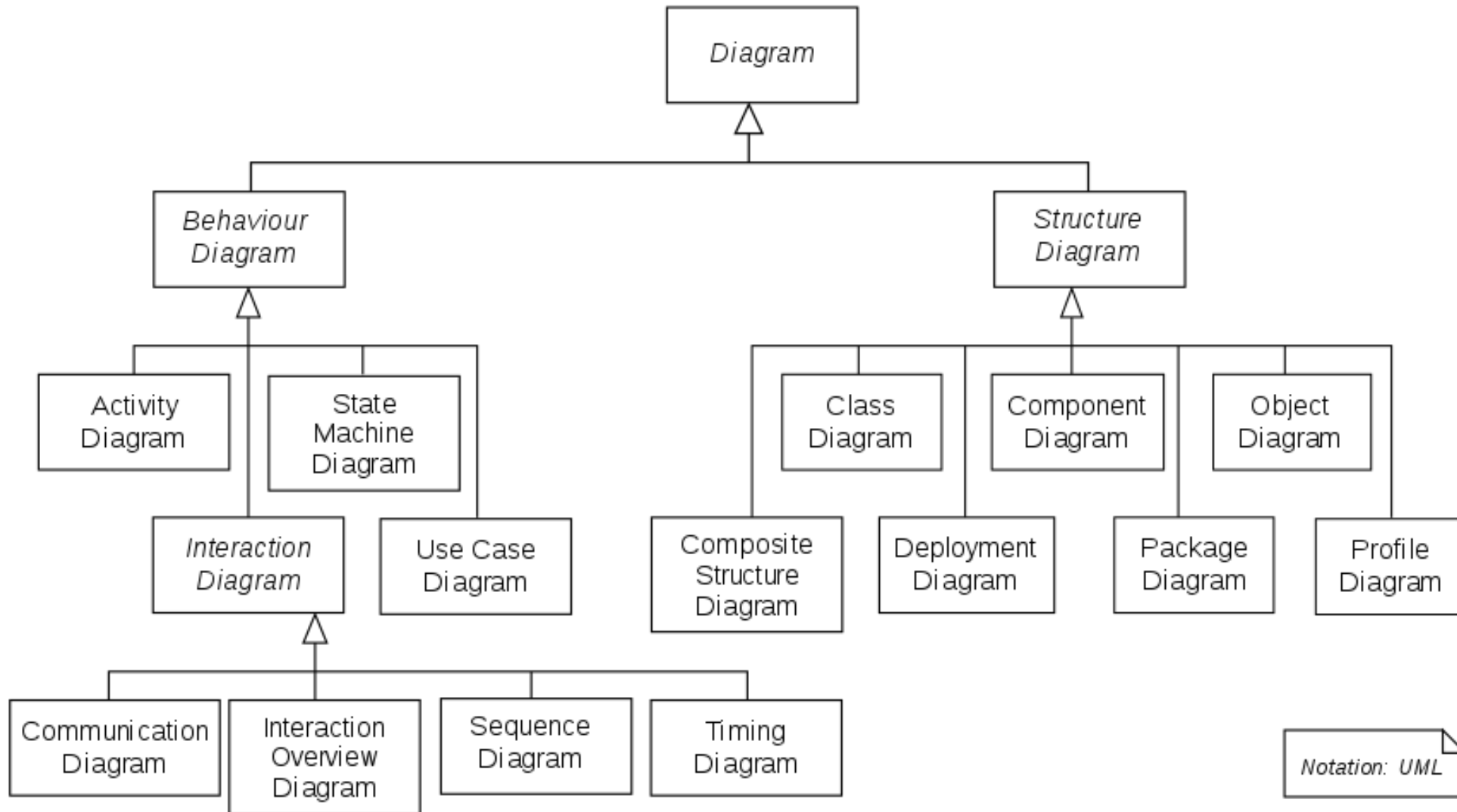## Model = abstraction of a problem

It can have different diagrams

Diagram = partial graphic representation of a model

OCL = Object Constraint Language

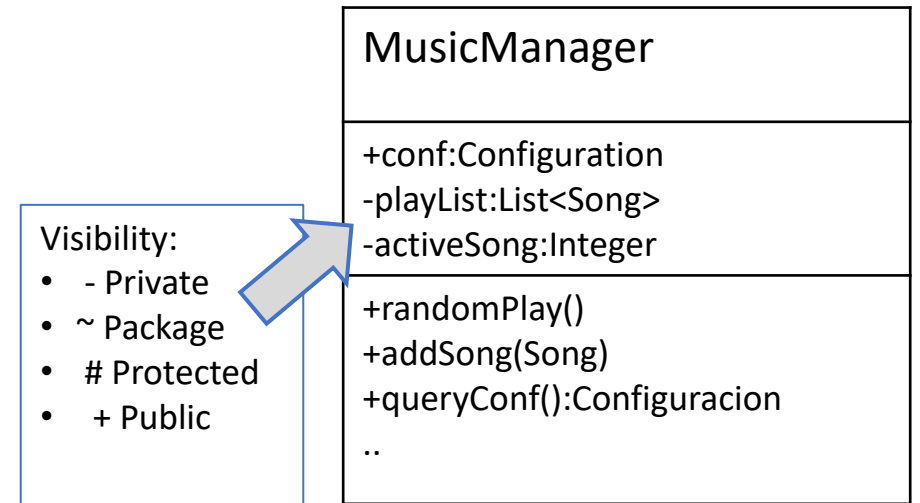Constraints between objects using formal language

14 UML Diagram types

Diagram

Behaviour Diagram

Structure Diagram

Activity Diagram

State Machine Diagram

Interaction Diagram

Use Case Diagram

Class Diagram

Component Diagram

Object Diagram

Composite Structure Diagram

Deployment Diagram

Package Diagram

Profile Diagram

Communication Diagram

Interaction Overview Diagram

Sequence Diagram

Timing Diagram

Notation: UML
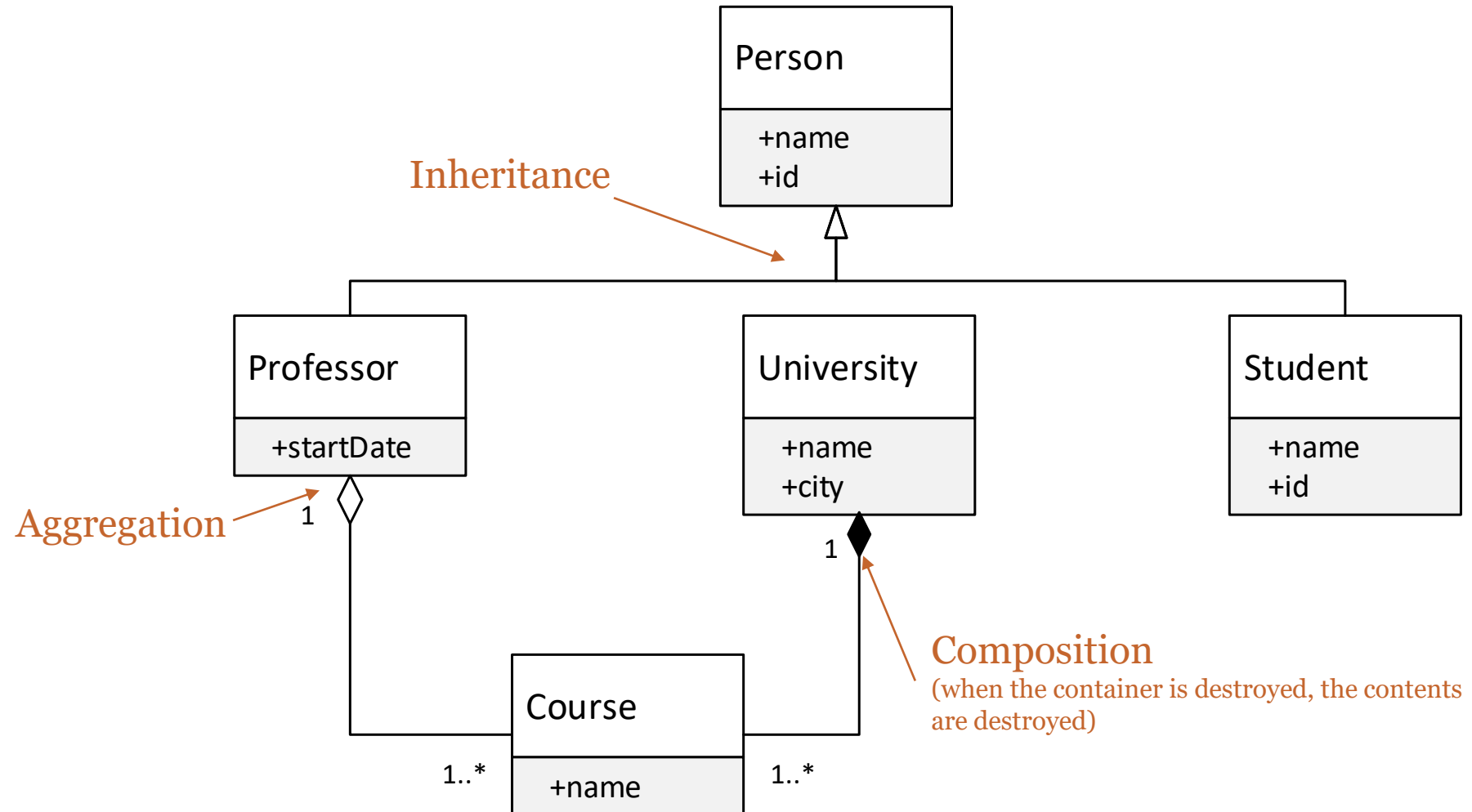
Fuente: Wikipedia

Class diagrams

Models the static part of the project, without taking into account the time aspect
Explains the relationships between the different classes.
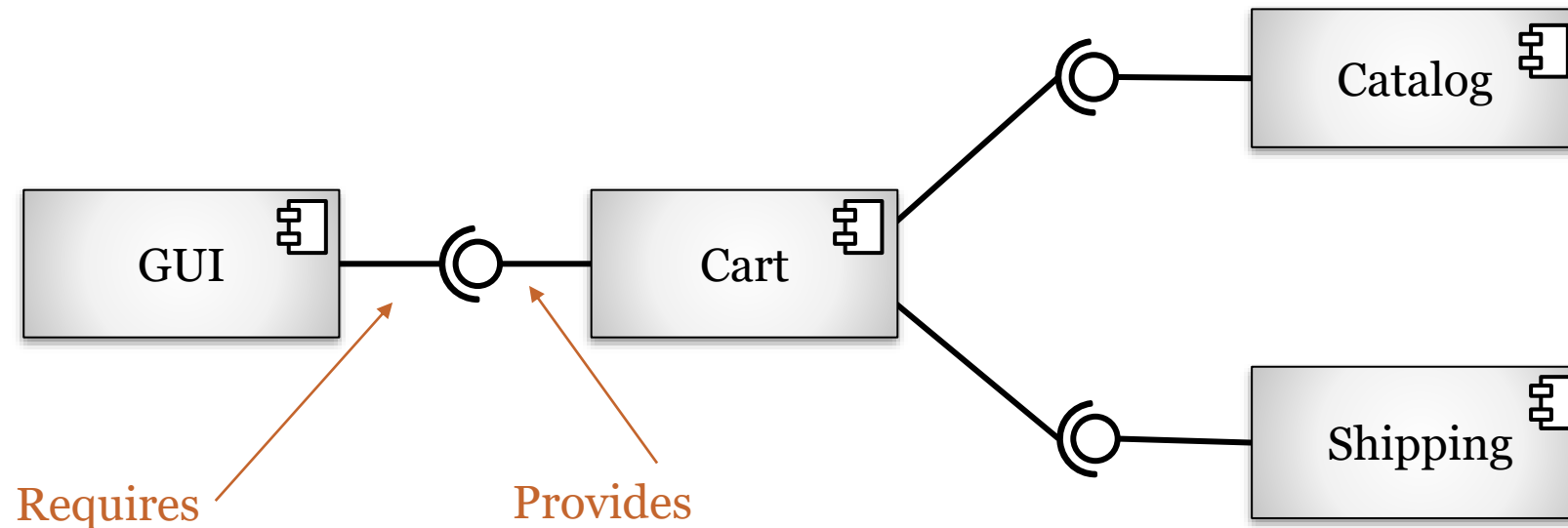Arc42: 8-Concepts

Visibility:
- - Private
- ~ Package
- # Protected
- + Public

MusicManager

+conf:Configuration
-playList:List<Song>
-activeSong:Integer

+randomPlay()
+addSong(Song)
+queryConf():Configuracion
..

Example

UML Component diagram

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

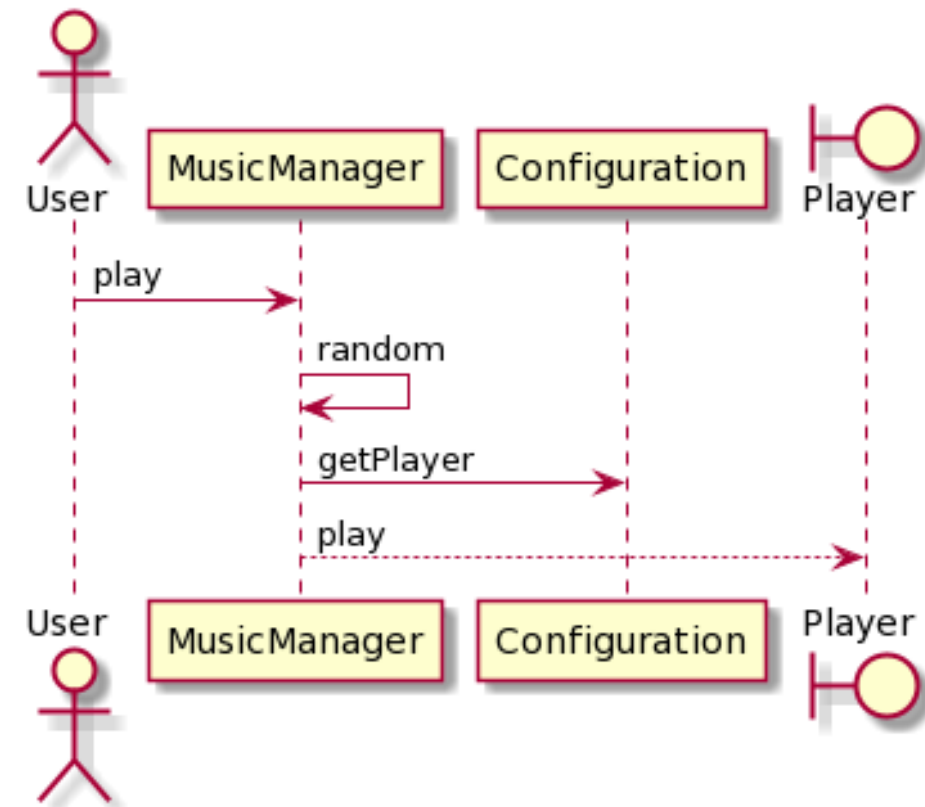School of Computer Science, University of Oviedo

Sequence diagram

Models communication between some objects at a given time
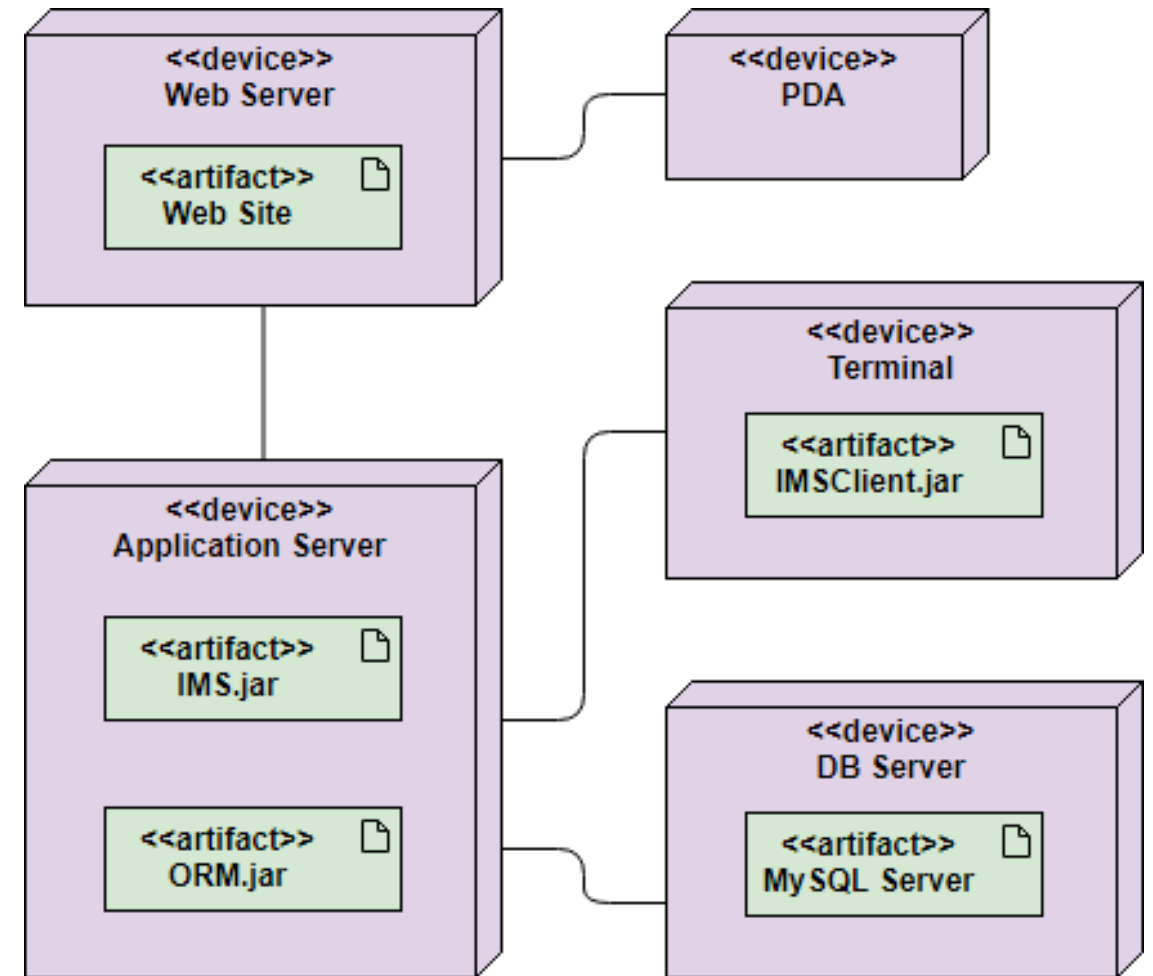Objects can send two types of messages: synchronous or asynchronous
Arc42:6-RuntimeView

Deployment diagrams

Represents the final location of the
components in an app

Elements:

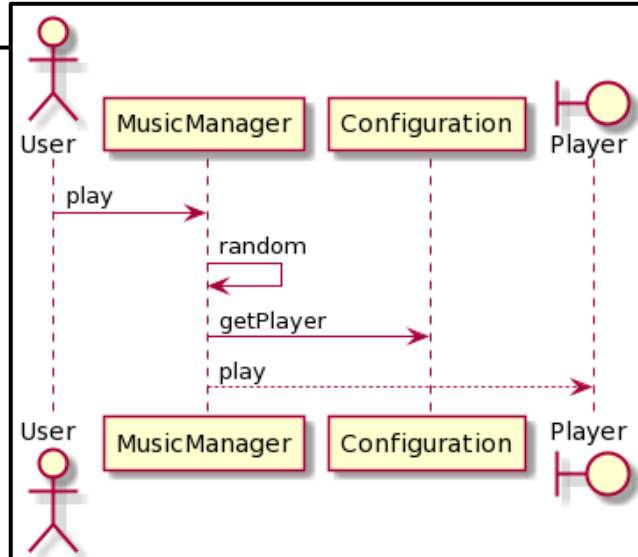Nodes , Components, relationships

Arc42: 07.DeploymentView

Source:VisualParadigm
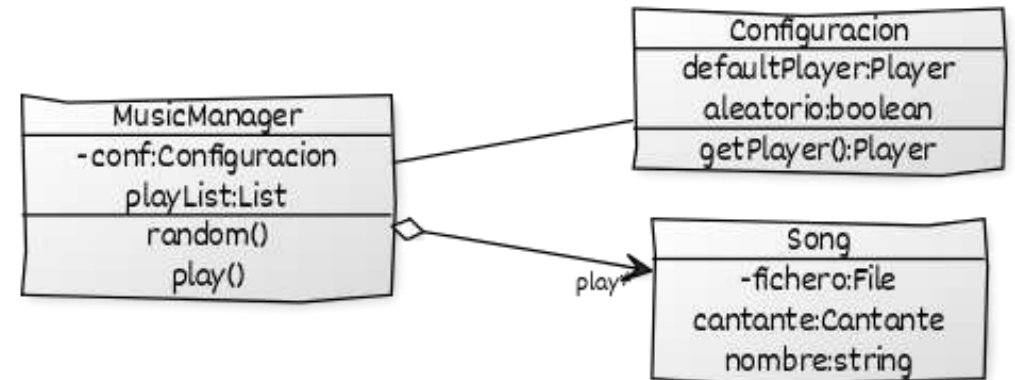
Text-based tools

PlantUML

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

YUML

```
// Cool Class Diagram
[MusicManager|-conf:Configuracion;
playList:List |random();play()]
[MusicManager]<>-play*>[Song|-
fichero:File;cantante:Cantante;nombre
:string]
[MusicManager]-
[Configuracion|defaultPlayer:Player;a
leatorio:boolean|getPlayer():Player]
]
```
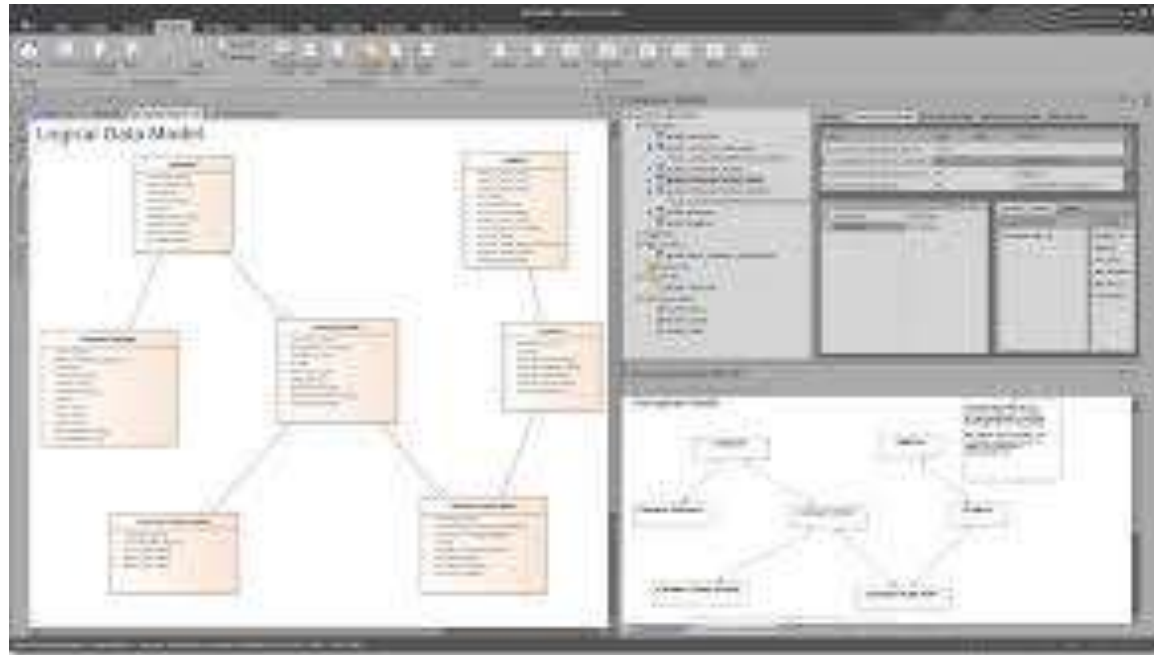
Drawing tools

Powerpoint

MsVisio

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

CASE tools

EnterpriseArchitect

Reverse Enginering 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
```

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
```