



Universidad de Oviedo



S O F T W A R E
A R C H I T E C T U R E

Software Architecture

Lab. 02

UML diagrams

PlantUML

Introduction to arc42 docs

2021-22

Jose Emilio Labra Gayo
Pablo González
Irene Cid
Hugo Lebreo

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

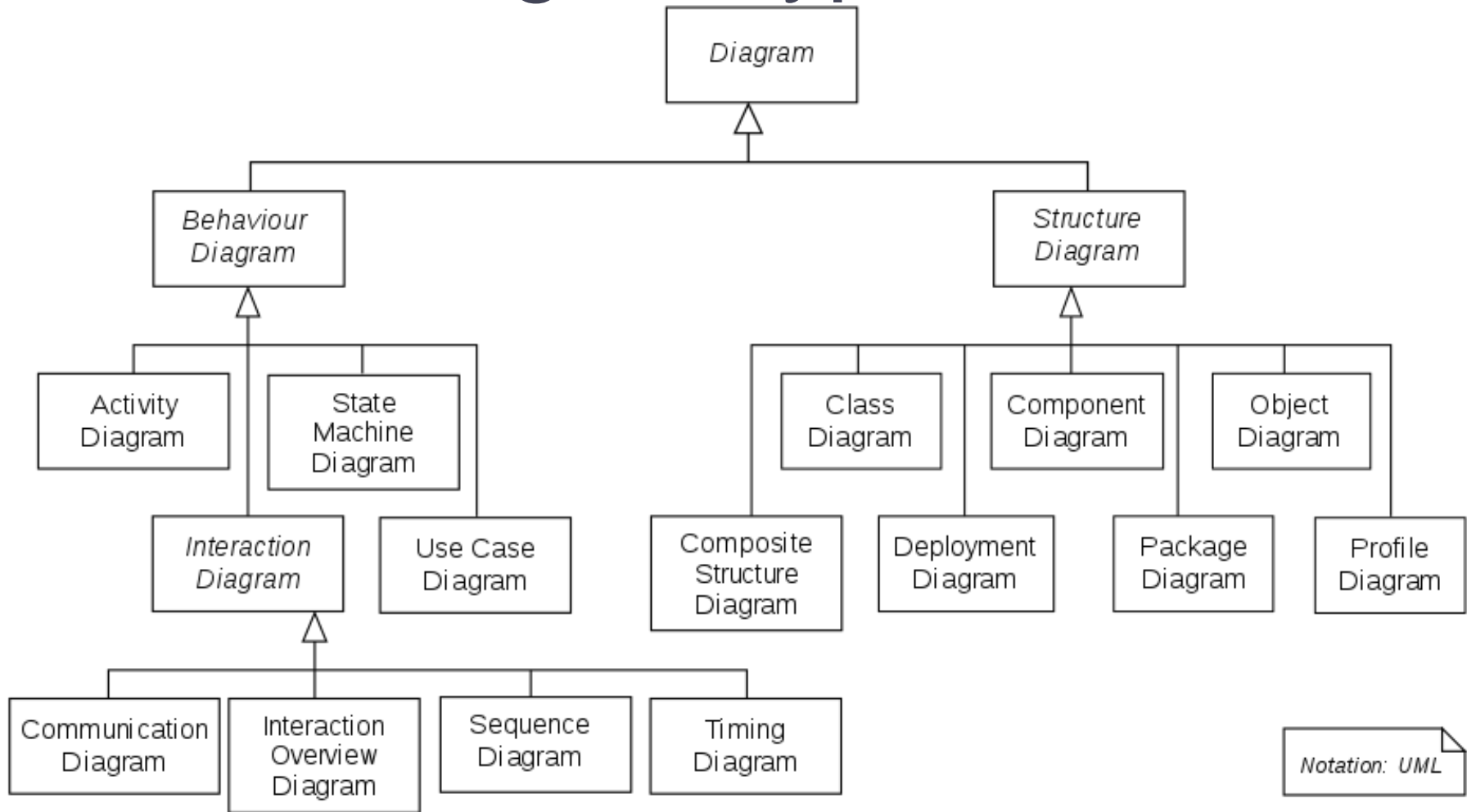
It can have different diagrams

Diagram = partial graphic representation of a model

OCL = Object Constraint Language

Constraints between objects using formal language

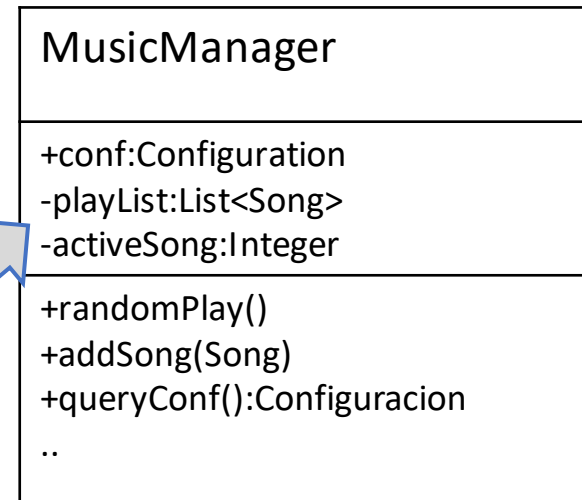
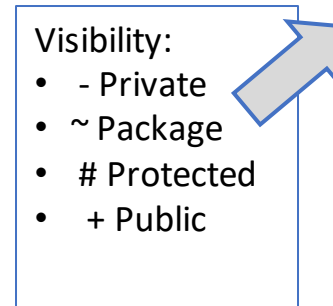
14 UML Diagram types



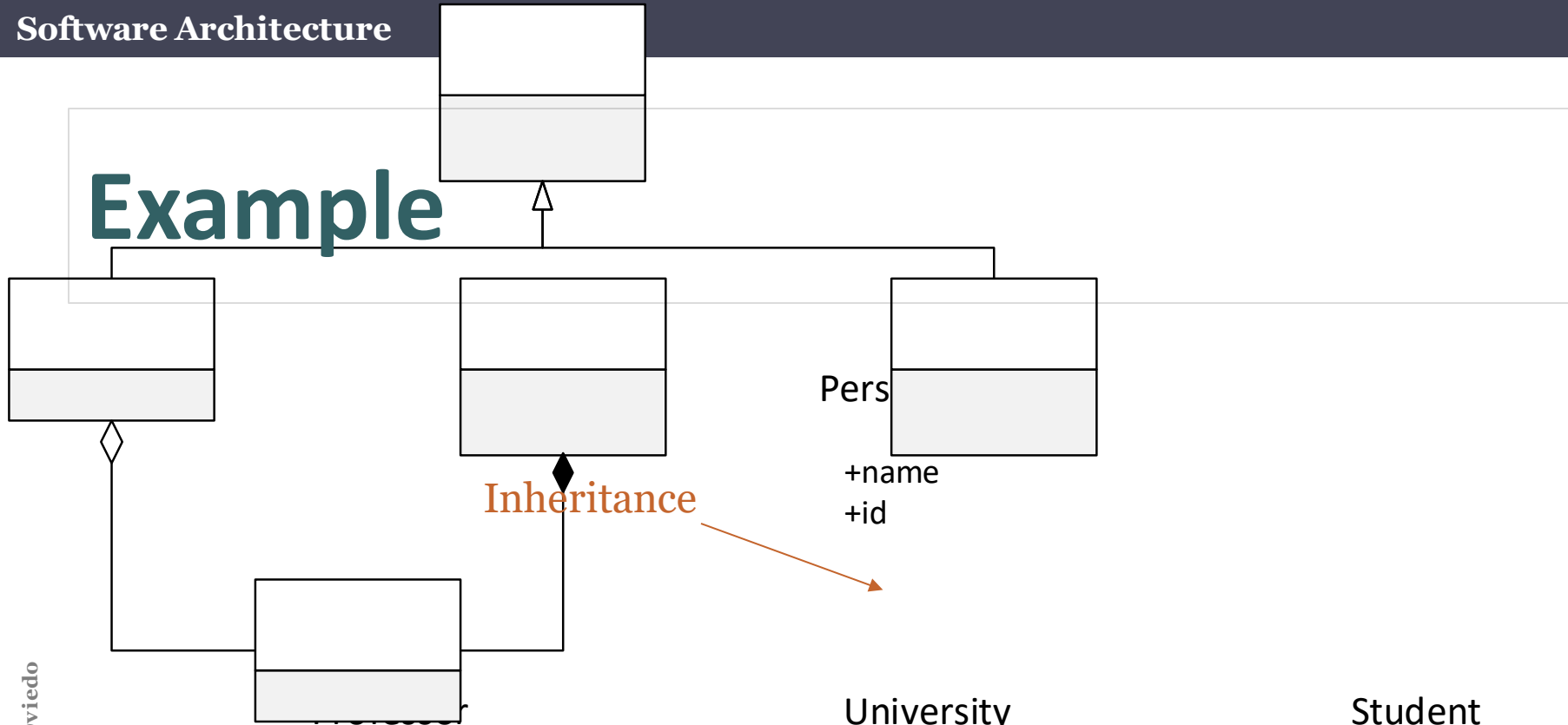
Notation: UML

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



Inheritance

Aggregation

Composition

(when the container is destroyed, the contents are destroyed)

+startDate

1

1..*

Course

+name

Pers

+name

+id

University

+name

+city

1

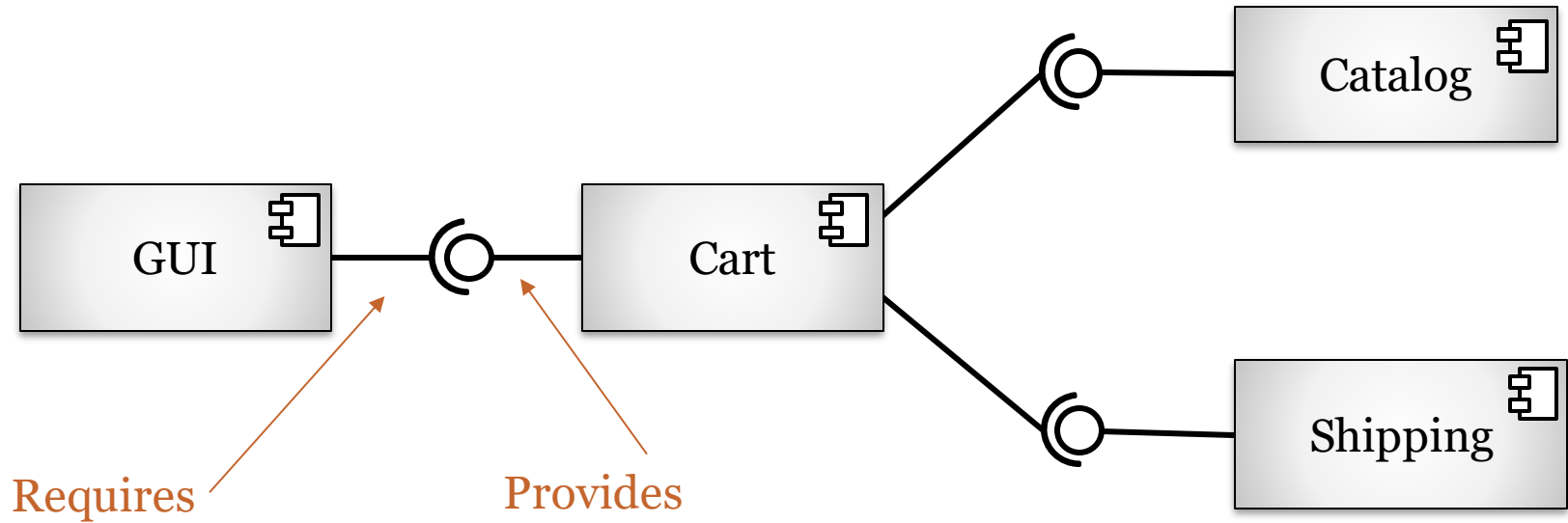
Student

+name

+id

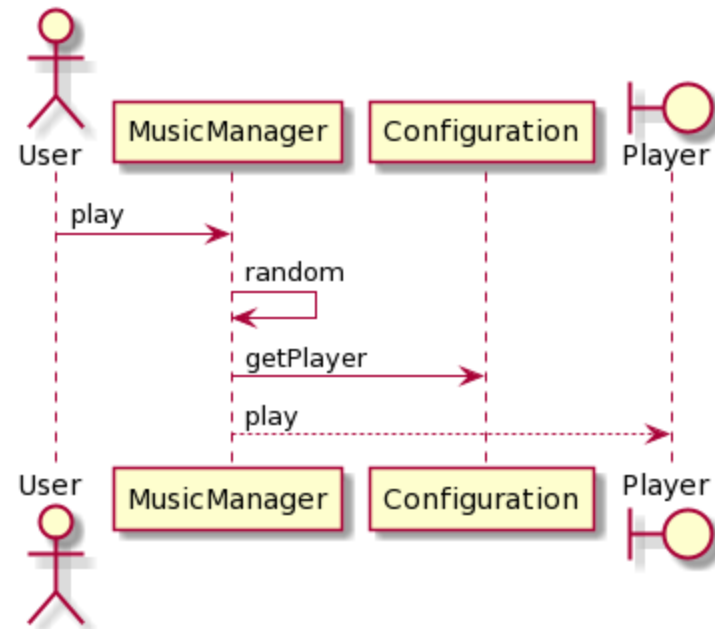
1..*

UML Component diagram



Sequence diagram

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



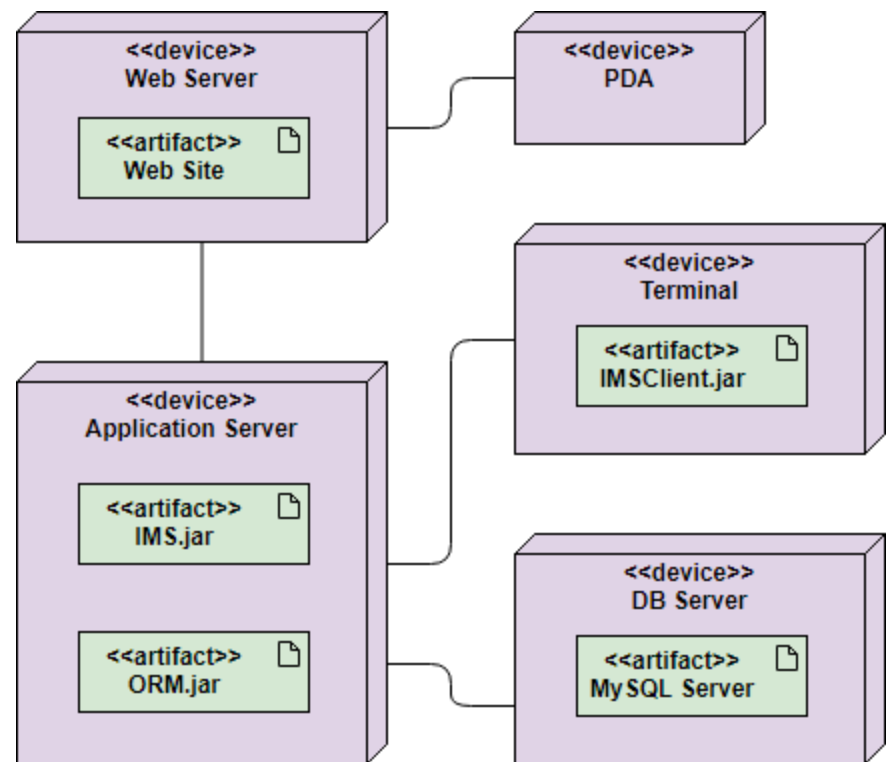
Deployment diagrams

Represents the final location of the components in an app

Elements:

Nodes , Components,
relationships

Arc42: 07.DeploymentView

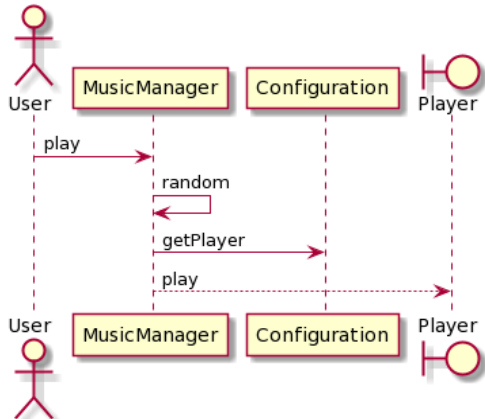


Text-based tools

PlantUML

```

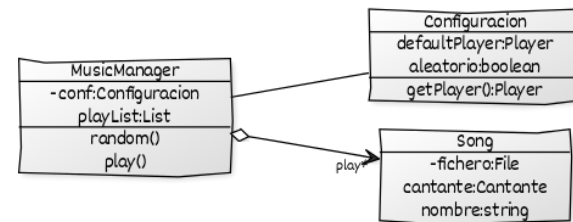
@startuml component
actor User
participant MusicManager
participant Configuration
boundary Player
User -> MusicManager: play
MusicManager -> MusicManager: random
MusicManager -> Configuration : getPlayer
MusicManager --> Player : play
@enduml
    
```



YUML

```

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



Drawing tools

Powerpoint

MsVisio

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

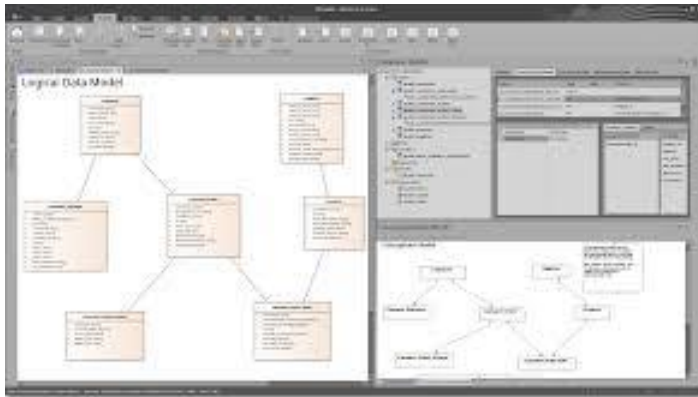
CASE tools

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

<https://arc42.org/>

DeDe already follows the template:

https://arquisoft.github.io/dede_0/

Generation of docs:

- > cd docs
- > npm run build

Documentation deployment

- Documentation is deployed using **GitHub Pages**
 - GitHub Pages allows users to publish a simple website directly on GitHub.
 - The docs website will be pushed to the branch **gh-pages**
 - The asciidoc files will be pushed to the develop branch of the repository (not automatically)
 - The npm package **gh-pages** takes care of pushing the doc website to the gh-pages branch
 - Everything is automatized with the following command:
 - > **npm run deploy**



End of presentation