Escuela Técnica Superior de Ingeniería Informática
Departamento de Lenguajes y Sistemas Informáticos

Static Modeling
Class & Object Diagrams

Software Engineering and Databases Group
Department of Computer Languages and Systems
University of Seville
November 2015

La traducción de este material docente ha sido financiada mediante la convocatoria 1.10B - Ayudas de innovación y mejora docente, convocatoria 2013-2014, modalidad B del II Plan Propio de Docencia de la Universidad de Sevilla. No ha habido financiación alguna para este proyecto de otros soportes.

Escuela Técnica Superior de Ingeniería Informática
Departamento de Lenguajes y Sistemas Informáticos

Static Modeling: Class & Object Diagrams

- Learning objectives
  - Know the basic concepts of **static modeling** using **UML class & object diagrams**.
  - Be able to **develop** a static model of a software system from a requirements specification.

Escuela Técnica Superior de Ingeniería Informática
Departamento de Lenguajes y Sistemas Informáticos

Static Modeling: Class & Object Diagrams

- A model of a software system includes...
  - A **static model** (conceptual model)
    - Describes the structure and constraints of the information representing the system state.
  - A **behavior model**
    - Describes how the system interacts with actors and how its state evolves as a result of the interactions.

Model = Static model = Structure + Constraints + Behavior model = External interactions + Internal evolution

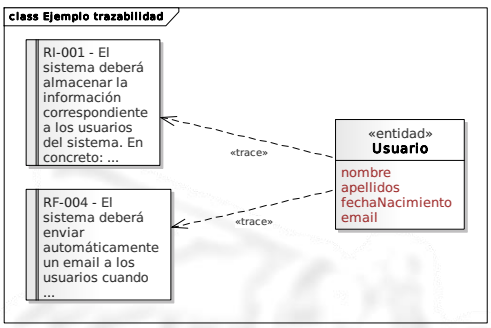
Escuela Técnica Superior de Ingeniería Informática
Departamento de Lenguajes y Sistemas Informáticos

Static Modeling: Class & Object Diagrams

- A model of a software system includes...
  - **Structure**
    - It describes the **information structure** representing the system state using...
      - UML class diagrams
      - UML object diagrams (scenarios)
  - **Constraints**
    - They describe which system states are **valid** and which are not using...
      - UML class diagrams (multiplicities)
      - Constraints associated to UML class diagrams elements, expressed either in natural language or in **OCL** (Object Constraint Language).

Static Modeling: Class & Object Diagrams

- Requirements traceability
  - All the elements in a static model must be **traced** back towards those requirements that justifies them, usually **information requirements** and **business rules**.

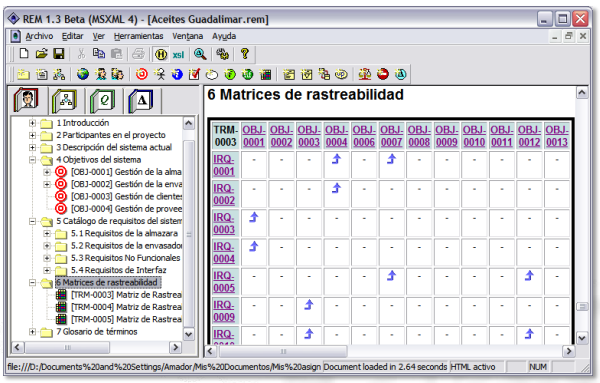


Modelado Estático: Diagramas de Clases y Objetos

Escuela Técnica Superior de Ingeniería Informática
Departamento de Lenguajes y Sistemas Informáticos

1. Static model of a softw. system
2. Basic concepts
   • Entity class
   • Association
   • Objects & links
3. Advanced concepts
   • Composition & aggregation
   • Generalization & especializat.
   • Association class
   • Constraints
   • Derived elements
   • Enumerations
4. Static model development

- Requirements traceability
  - All the elements of a static model must be **traced** back towards those requirements that justifies them, usually **information requirements** and **business rules**.



Escuela Técnica Superior de Ingeniería Informática
Departamento de Lenguajes y Sistemas Informáticos

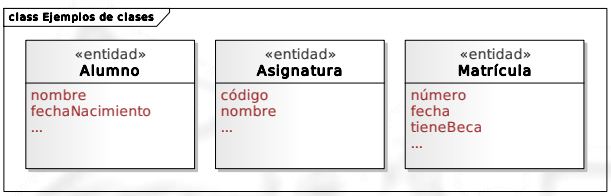
Static Modeling: Class & Object Diagrams

- Basic concepts of static modeling
  - Entity class
    - Attribute
  - Association
    - Role
    - Multiplicity
  - Association class
  - Object (instance of a class)
  - Link (instance of an association)
  - Generalization and specialization
  - Composition and aggregation

Escuela Técnica Superior de Ingeniería Informática
Departamento de Lenguajes y Sistemas Informáticos

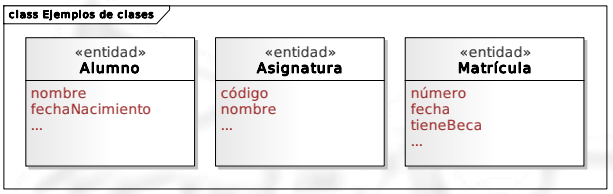
Static Modeling: Class & Object Diagrams

- Entity class
  - It represents a **relevant concept** from the problem domain for which the system must store information...
    - ...because it is specified in (or deduced from) some requirements.
  - Classes are named with a **noun in singular form**.



Escuela Técnica Superior de Ingeniería Informática
Departamento de Lenguajes y Sistemas Informáticos

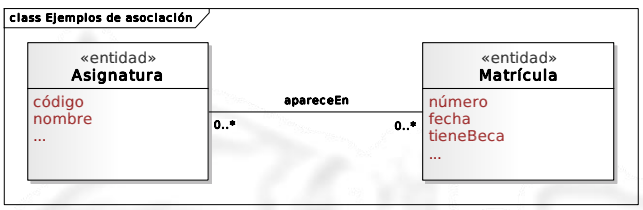
- Entity class attributes
  - They are **properties** of a relevant concept from the problem domain that the system must store...
    - ...because it is specified in (or deduced from) some requirements (traces*).
  - They are named with a **noun in singular form** and their values must be simple (not data structures).
  - They can be optional ([0..1]).



* Usually, traces are set at a class level, although they could also be set at an attribute level.

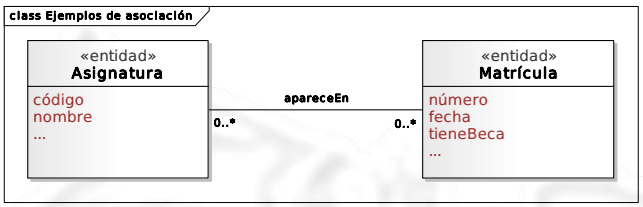
Escuela Técnica Superior de Ingeniería Informática
Departamento de Lenguajes y Sistemas Informáticos

- Association between entity classes
  - It represents some kind of **relationship** between two or more relevant concepts from the problem domain that the system must be aware of...
    - ...because it is specified in (or deduced from) some requirements.



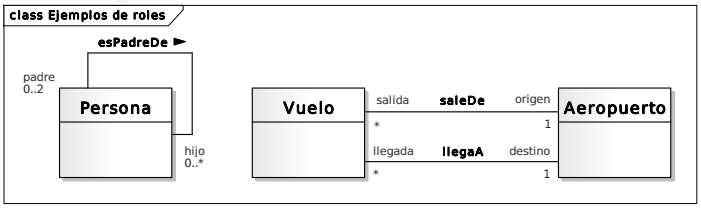
Escuela Técnica Superior de Ingeniería Informática
Departamento de Lenguajes y Sistemas Informáticos

- Association between entity classes
  - It is named using a **verb** in **singular third person** and prepositions if needed.
  - It must form a **comprehensible sentence** when reading it together with the roles.



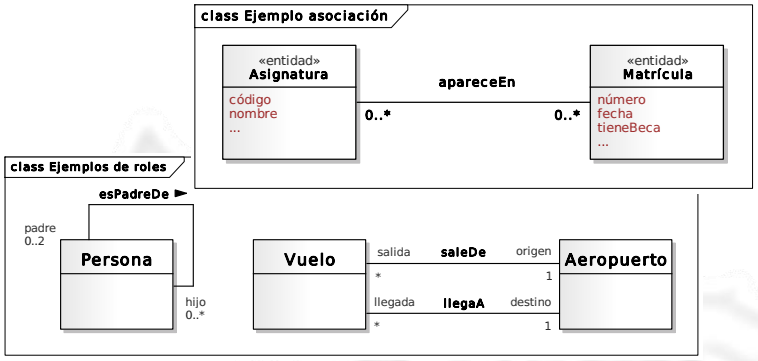
Escuela Técnica Superior de Ingeniería Informática
Departamento de Lenguajes y Sistemas Informáticos

- Role of an association end
  - **Role** played by each of the classes participating in an association.
    - By default, it is its own name starting in lower case.
  - They **must** be specified in:
    - Associations of a class with itself
    - When there exist more than one association between the same two classes.



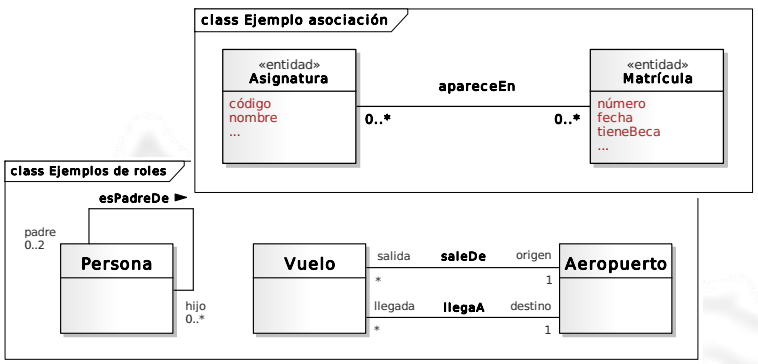
Escuela Técnica Superior de Ingeniería Informática
Departamento de Lenguajes y Sistemas Informáticos

- Multiplicity of an association end
  - Given an object of a class, it indicates the **minimum** and **maximum** number of objects of the other class to which it can be related to using association links.



Escuela Técnica Superior de Ingeniería Informática
Departamento de Lenguajes y Sistemas Informáticos

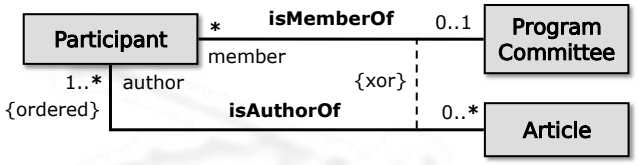
- Multiplicity of an association end
  - Usual multiplicity values
    - **0..1** : optional
    - **0..*** : multiple optional
    - ***** : equivalent to 0..*
    - **1..1** : mandatory
    - **1..*** : multiple mandatory
    - **1** : equivalent to 1..1



- Constraints at an association end
  - **{nonunique}**: duplicates are allowed
  - **{ordered}**: objects are ordered
  - By default, it assumed that duplicates are not allowed and that object are unordered.

	unordered	{ordered}
without duplicates	Set (default)	Ordered set
with duplicates {nonunique}	Bag	Sequence

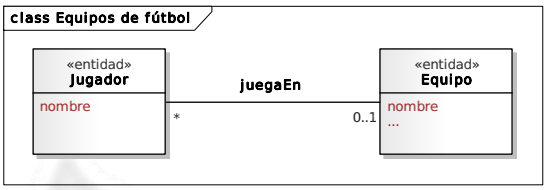
- Constraints between associations
  - **{xor}**: it indicates that the objects of the class common to the constrained associations cannot have links of more than one association.
  - Example: a participant in a conference cannot be an author of an article and, at the same time, member of the program committee.



Escuela Técnica Superior de Ingeniería Informática
Departamento de Lenguajes y Sistemas Informáticos

Static Modeling: Class & Object Diagrams

- Objects
  - Every **instance** or occurrence of a class.
- Links
  - Every **instance** or occurrence of an association.

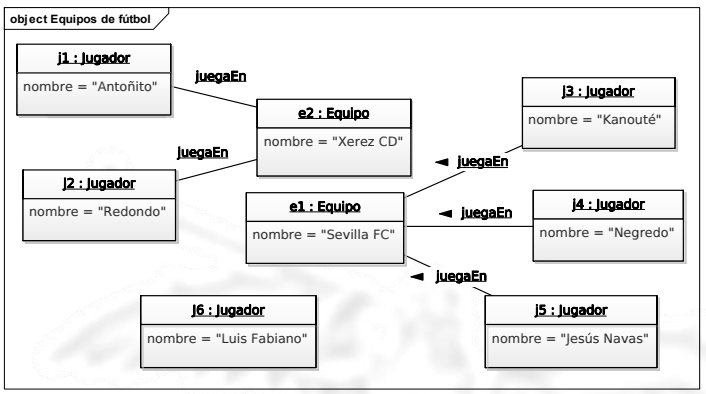


Escuela Técnica Superior de Ingeniería Informática
Departamento de Lenguajes y Sistemas Informáticos

1. Static model of a softw. system
2. Basic concepts
   • Entity class
   • Association
   • Objects & links
3. Advanced concepts
   • Composition & aggregation
   • Generalization & especializat.
   • Association class
   • Constraints
   • Derived elements
   • Enumerations
4. Static model development

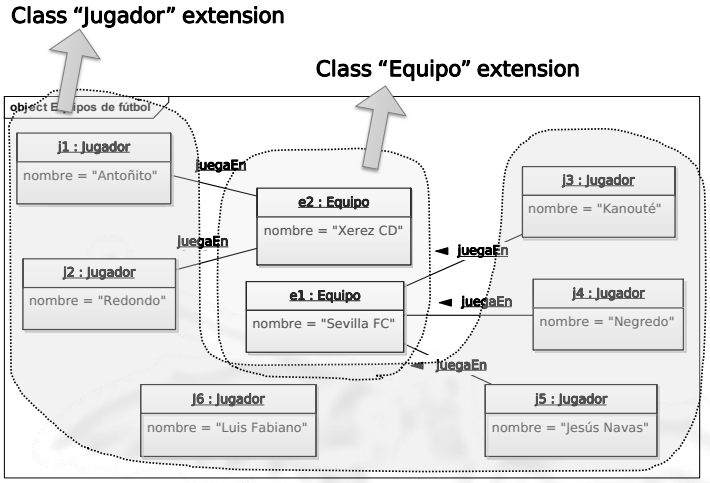
Static Modeling: Class & Object Diagrams

- Objects
  - Every instance or occurrence of a class.
- Links
  - Every instance or occurrence of an association.



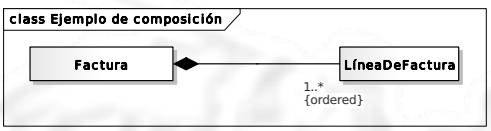
Escuela Técnica Superior de Ingeniería Informática
Departamento de Lenguajes y Sistemas Informáticos

- Extension of a class
  - All the instances of a given class in the system.



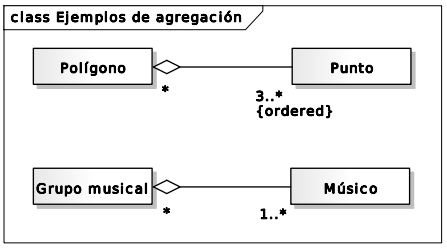
Escuela Técnica Superior de Ingeniería Informática
Departamento de Lenguajes y Sistemas Informáticos

- Composition
  - Special association that represents the concept of **being-part-of** or **being-composed-of**:
    - A *part* (component) only belongs to one *whole* (composite).
    - A *part* cannot exists without being part of one *whole*.
    - The elimination of the *whole* implies the elimination of all its *parts*.
    - It is a transitive and antisymmetric relation.
    - It can be recursive.



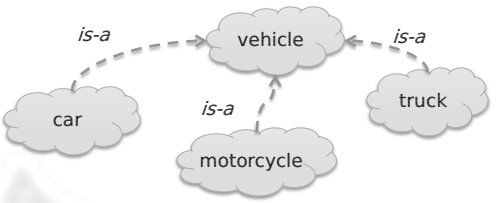
Escuela Técnica Superior de Ingeniería Informática
Departamento de Lenguajes y Sistemas Informáticos

- Aggregation
  - **Weak** composition in which:
    - A *part* (component) can belong to more than on *whole* (aggregate), i.e. its multiplicity is not constrained to be 1.
    - A *part* can exist without belonging to a *whole*.
    - The elimination of the *whole* does not necessarily implies the elimination of all its *parts*.



Escuela Técnica Superior de Ingeniería Informática
Departamento de Lenguajes y Sistemas Informáticos

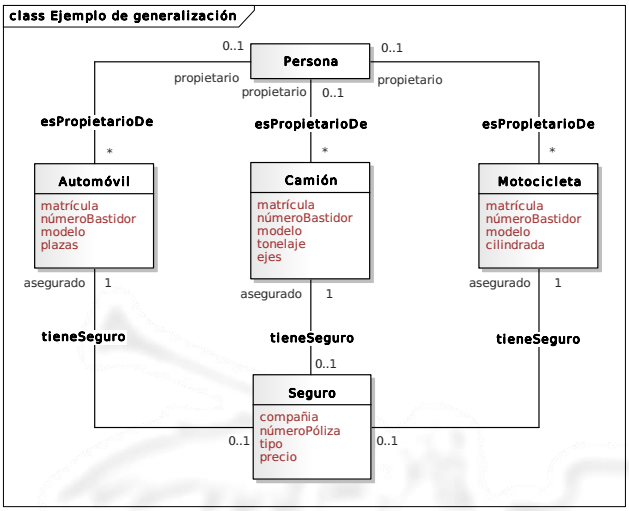
- Generalization/Specialization
  - Sometimes, some concepts from the problem domain have relationships of type **is-a**, for example:



  - These concepts usually have **common properties**, that appears as attributes or common associations when they are modeled.

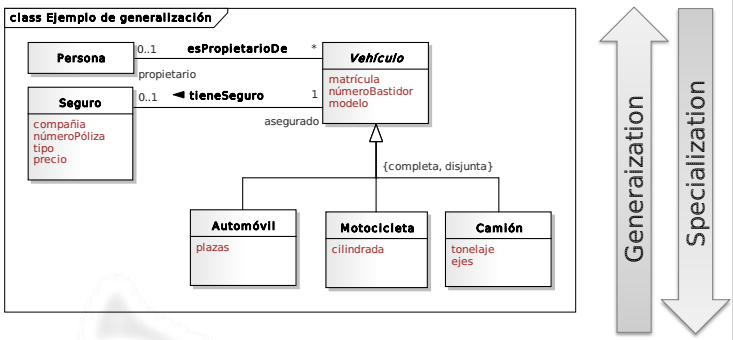
Escuela Técnica Superior de Ingeniería Informática
Departamento de Lenguajes y Sistemas Informáticos

- Generalization/Specialization



Escuela Técnica Superior de Ingeniería Informática
Departamento de Lenguajes y Sistemas Informáticos

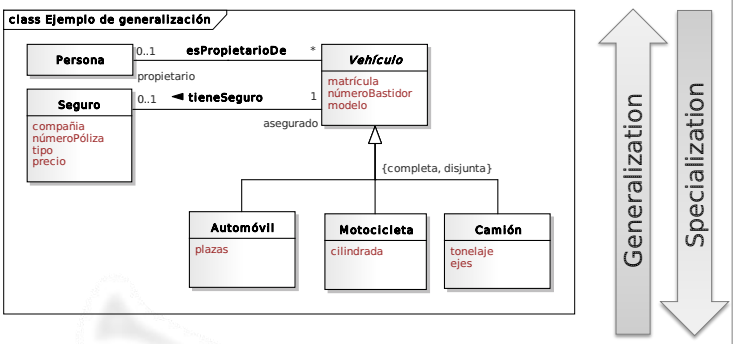
- Generalization/Specialization



  - The most general class (the **superclass**), contains all the common properties (attributes and associations), **inherited** by the specific classes (**subclasses**).

Escuela Técnica Superior de Ingeniería Informática
Departamento de Lenguajes y Sistemas Informáticos

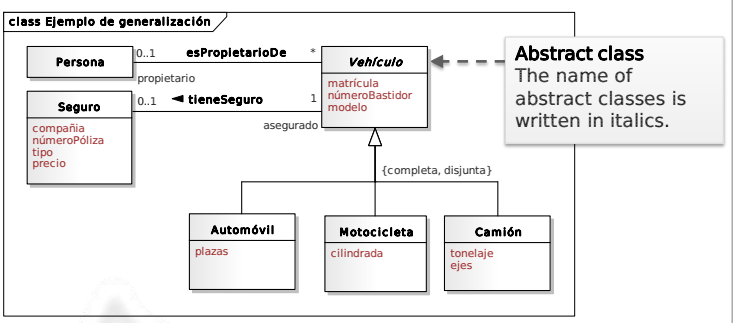
- Generalization/Specialization



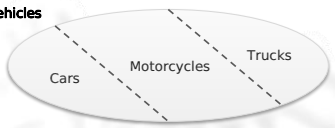
  - All the instances of the subclasses are also instances of the superclass.
  - Generalization is a transitive and antisymmetric relation.

Escuela Técnica Superior de Ingeniería Informática
Departamento de Lenguajes y Sistemas Informáticos

- Generalization/Specialization

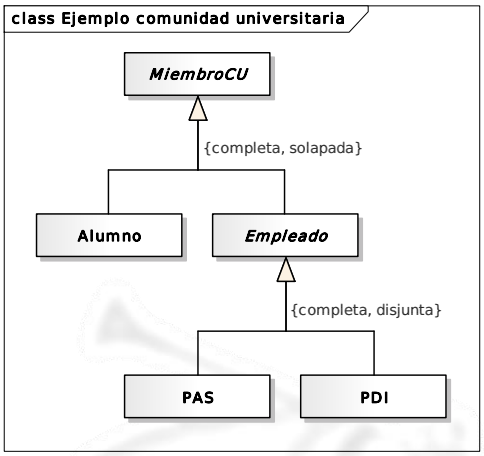


  - {complete, disjoint} implies a **partition** of the set of instances of the superclass.



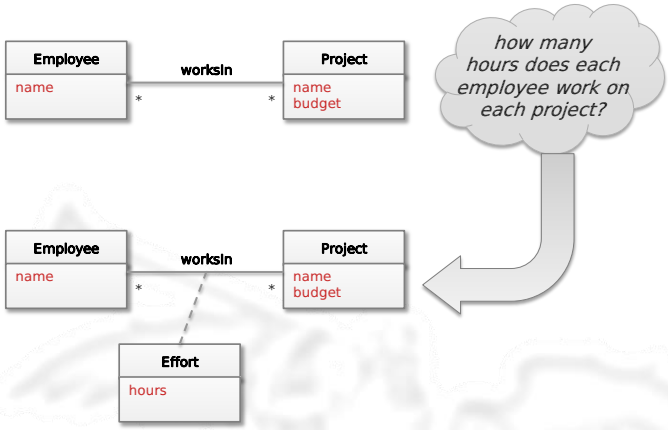
Escuela Técnica Superior de Ingeniería Informática
Departamento de Lenguajes y Sistemas Informáticos

- Generalization/Specialization

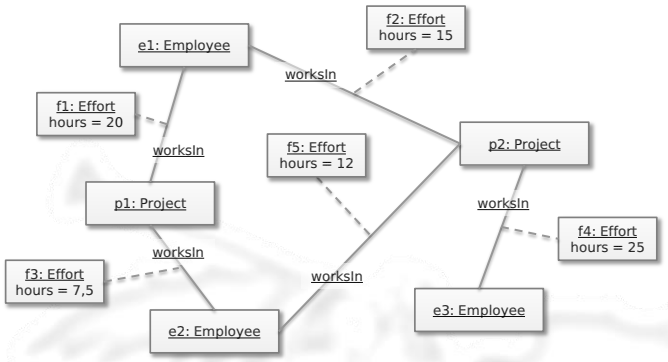


Escuela Técnica Superior de Ingeniería Informática
Departamento de Lenguajes y Sistemas Informáticos

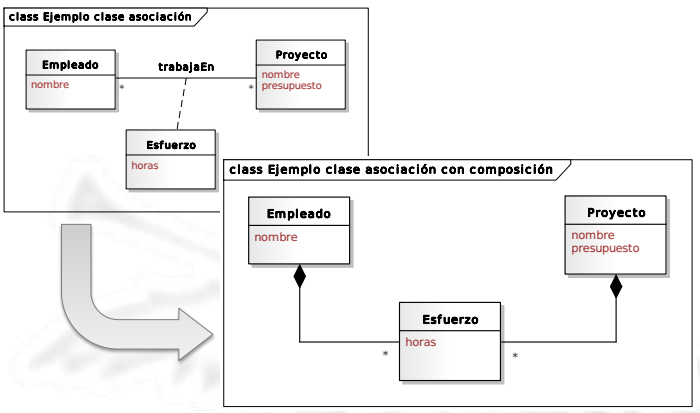
- Association class
  - Sometimes, some information must be added to associations, turning them into classes.



- Association class
  - Example: the hours that an employee works on a project are neither properties of the employee nor the project, but of the association between them.

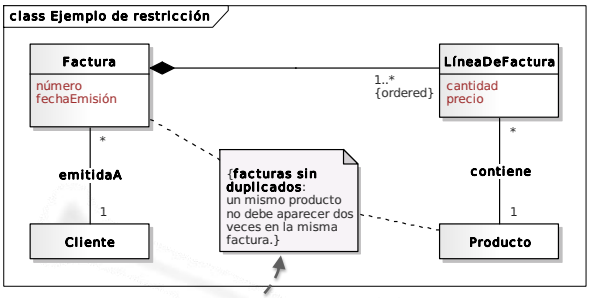


- Association class
  - It is also possible to model them as a class component of the participating classes.
  - Both models are equivalent.



Escuela Técnica Superior de Ingeniería Informática
Departamento de Lenguajes y Sistemas Informáticos

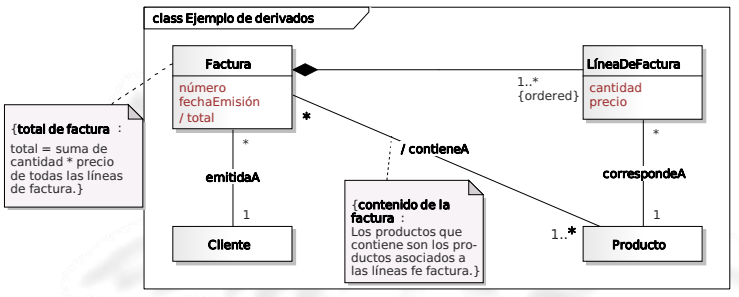
- Constraints
  - They allow additional information in the model which cannot be expressed in any other way.



Notation
Constraints are represented as **notes**. The text must be between curly braces, indicating the name of the constraint and its description. Optionally, it can be linked to the affected entities.

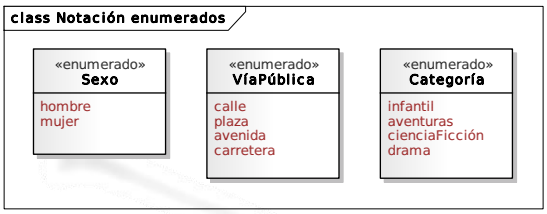
Escuela Técnica Superior de Ingeniería Informática
Departamento de Lenguajes y Sistemas Informáticos

- Derived elements
  - The slash (**/**) indicates that some attribute or association can be **derived** (computed).
  - **Constraints** describing derivations must be added to the model.



Escuela Técnica Superior de Ingeniería Informática
Departamento de Lenguajes y Sistemas Informáticos

- Notation for enumerations
  - They define a **type** that can be used in class attributes.
  - The attributes of the enumerations are the possible values.



Escuela Técnica Superior de Ingeniería Informática
Departamento de Lenguajes y Sistemas Informáticos

- Conceptual model development
  - Recommended steps:
    1. Analyze the information about the problem domain (glossary) and requirements.
    2. Identify possible entities and attributes.
    3. Identify possible associations.
    4. Incrementally build the conceptual model identifying the multiplicity of the associations.
    5. Identify classifications between entities with common properties (attributes and/or associations).
    6. Identify composition between entities.
    7. Add the constraints that cannot be graphically expressed.
    8. Validate with possible scenarios using object diagrams.
    9. Register all semantic problems that must be discussed with customers and users.

Escuela Técnica Superior de Ingeniería Informática
Departamento de Lenguajes y Sistemas Informáticos

Static Modeling: Class & Object Diagrams

- Bibliography
  - C. Larman, ***UML y Patrones***. Ed. Prentice-Hall, 1999.
    - Chapters 9 to 12
  - C. Larman, ***UML y Patrones*** (2ª edición). Ed. Prentice-Hall, 2003.
    - Chapters 10 to 12
  - M. Fowler, ***UML Distilled*** (3rd edition). Ed. Addison-Wesley, 2004.
    - Chapter 3

Escuela Técnica Superior de Ingeniería Informática
Departamento de Lenguajes y Sistemas Informáticos

Static Modeling: Class & Object Diagrams

- Comments, suggestions, ...

Amador Durán Toro
amador@us.es
Departamento de Lenguajes y Sistemas Informáticos
E.T.S. Ingeniería Informática, Universidad de Sevilla, España