

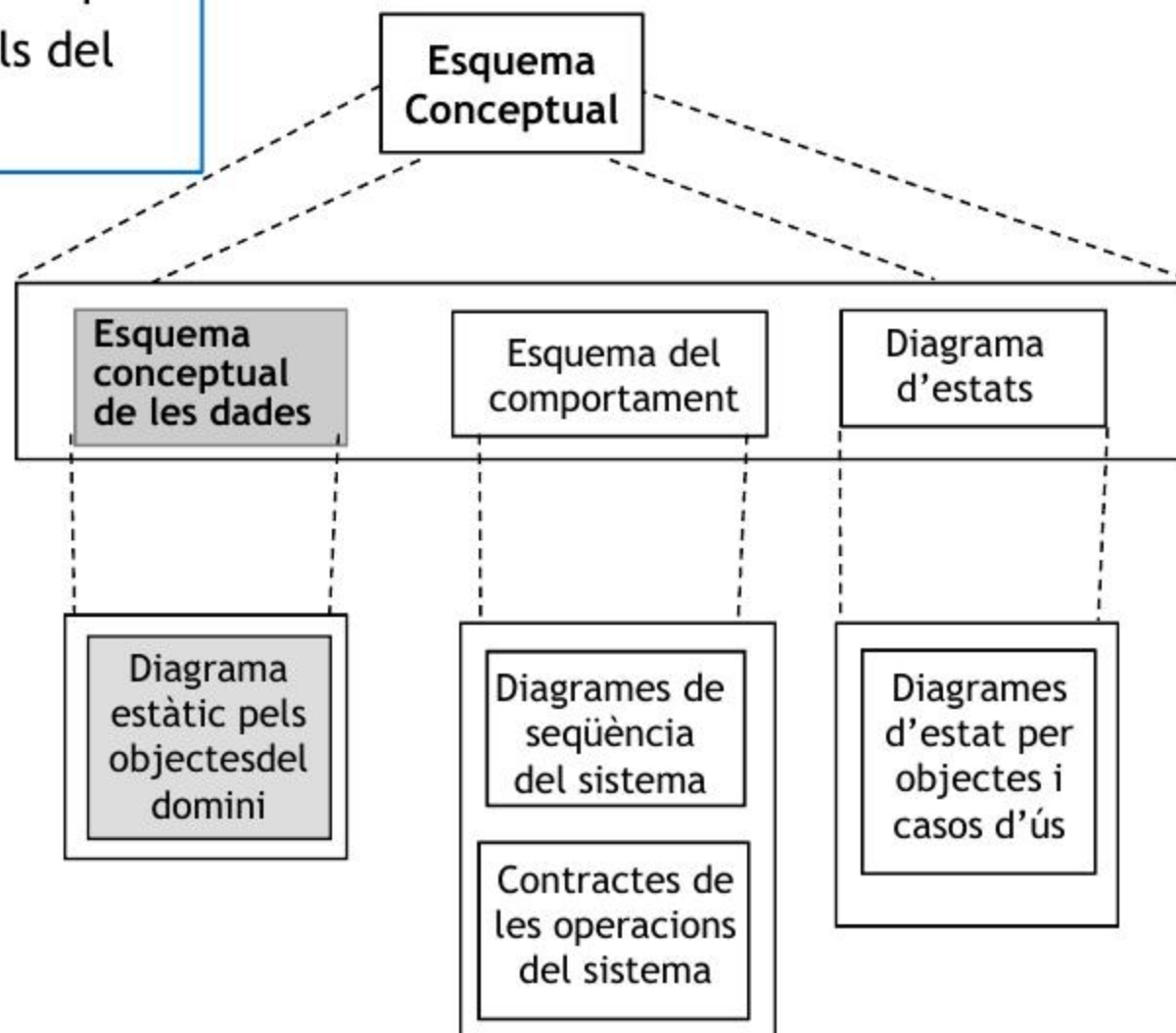
Especificació en UML: Esquema conceptual de les dades

Esquema conceptual de les dades en UML

- Objectes i classes d'objectes
- Especificació i disseny orientats a objectes
- Atributs
- Associacions
- Classe associativa
- Generalització/Especialització
- Agregació i composició
- Ampliacions
- Exemples

Esquema conceptual (especificació)

Esquema conceptual: descripció dels requisits funcionals del sistema software



Esquema conceptual de les dades

És la representació dels conceptes (objectes)
significatius en el domini del problema

Mostra, principalment:

- Classes d'objectes.

- Associacions entre classes d'objectes.

- Atributs de les classes d'objectes.

- Restriccions d'integritat, gràfiques i textuais

Objectes

Objecte:

Entitat que existeix al món real

Té identitat pròpia i és distingible dels altres objectes



l'avió amb
matrícula 327



una poma



un semàfor



la factura 3443



l'avió amb
matrícula 999

Classe d'objectes

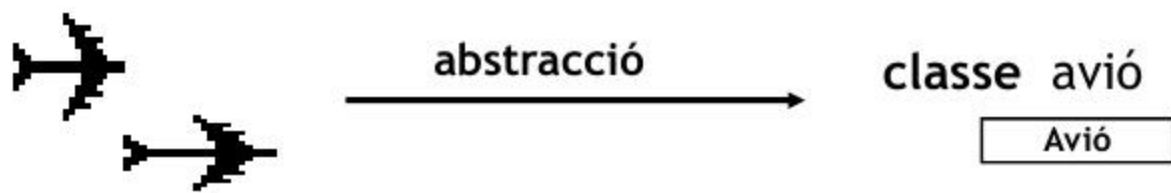
Classe d'objectes. Descriu un conjunt d'objectes amb:

Semàntica comuna

Les mateixes propietats

Idèntica relació amb altres objectes

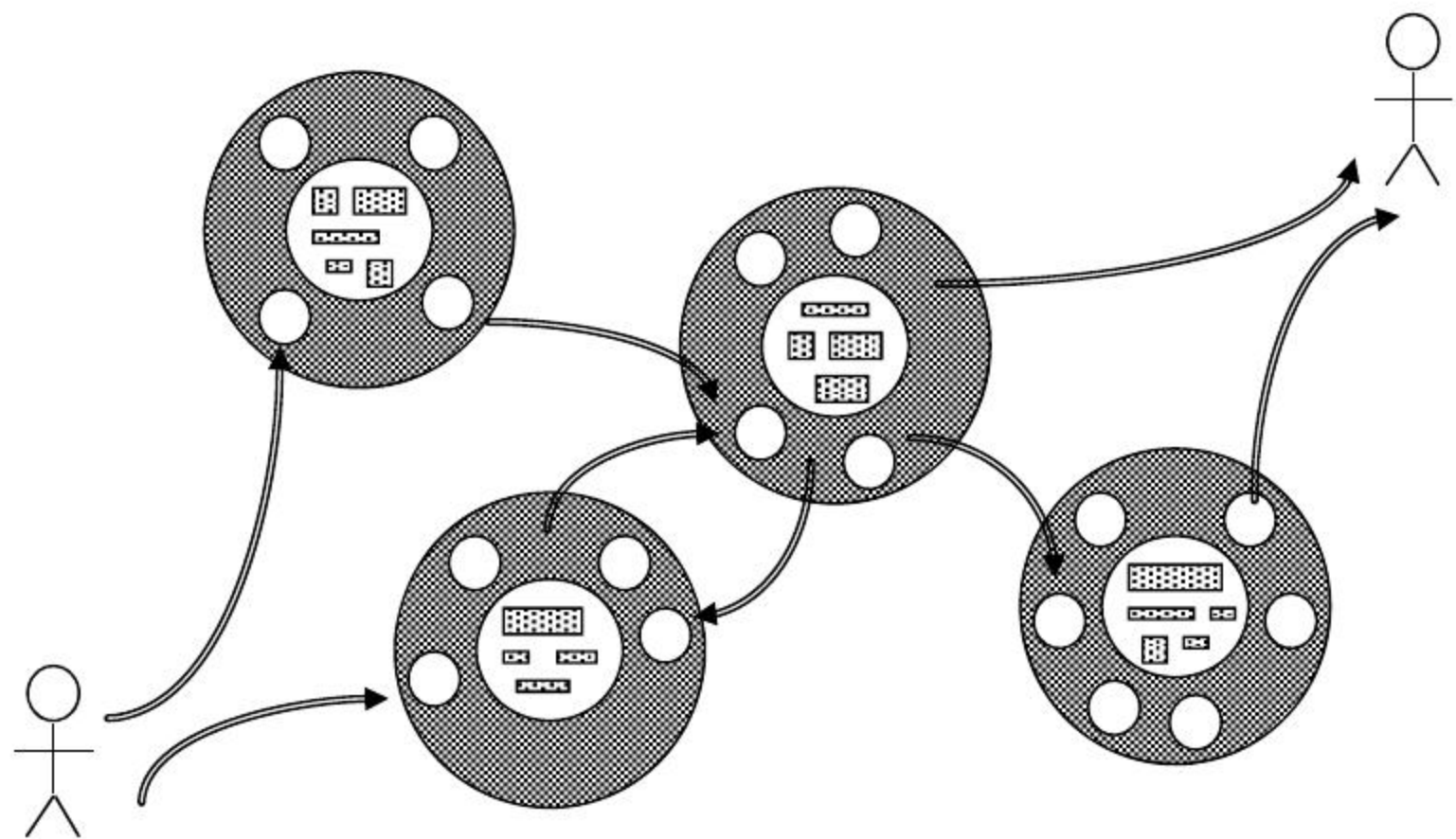
Comportament comú



Abstracció: permet eliminar distincions entre objectes per poder-hi observar aspectes comuns

Els objectes d'una classe tenen les mateixes propietats i els mateixos patrons de comportament

Visió orientada a objectes d'un sistema software



Especificació i disseny orientats a objectes

Especificació:

- Descripció dels requisits funcionals del sistema software
- **Què** ha de fer el sistema software

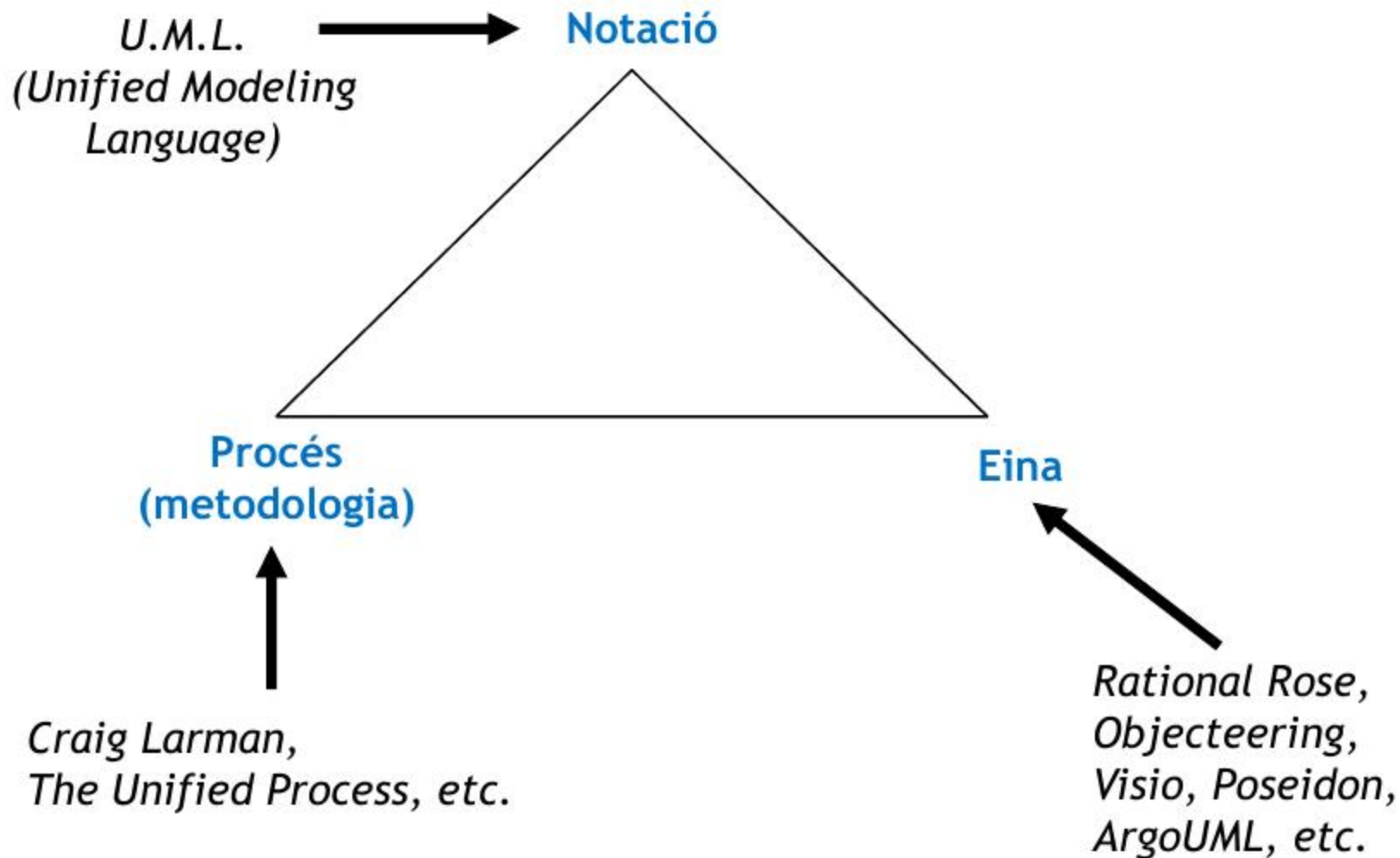
Disseny:

- Definició d'una solució software que satisfaci els requisits
- **Com** ho farà el sistema software

... orientats a objectes

- S'usen els mateixos conceptes a l'especificació i al disseny
- És difícil determinar on acaba l'especificació orientada a objectes i on comença el disseny:
 - estratègia de desenvolupament iterativa
 - diferències de criteris segons els autors

El triangle de l'èxit



Exemple: classes i atributs d'un supermercat

Un atribut és una propietat compartida pels objectes d'una classe

TerminalPuntVenda
num-pv: Integer

Línia de venda
quantitat: Integer

Pagament
import: Integer

Supermercat
adreça: String nom: String

Client
nom: String telfs: Integer [1..*] tipcli: TipusClient

Venda
data: Date hora: Time

Producte
upc: Integer descripció: Text [0..1] preu: Integer

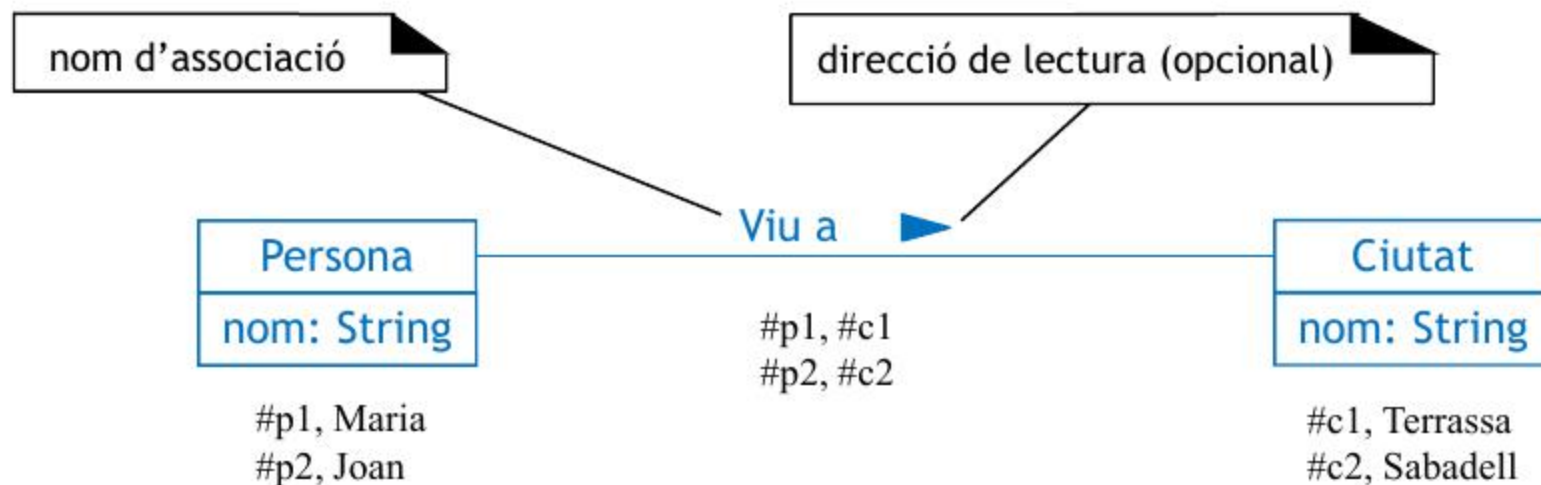
Els atributs:

- Poden ser **clau externa**, per identificar l'objecte dins la classe (upc de Producte)
- Poden prendre valors **nuls** (descripció de Producte)
- Poden ser **multivaluats** (telfs de Client)
- Poden ser **definitos per l'usuari** mitjancant **enumeracions**
 - Per exemple, TipusClient.

<<enumeration>> TipusClient
normal preferent

Associacions

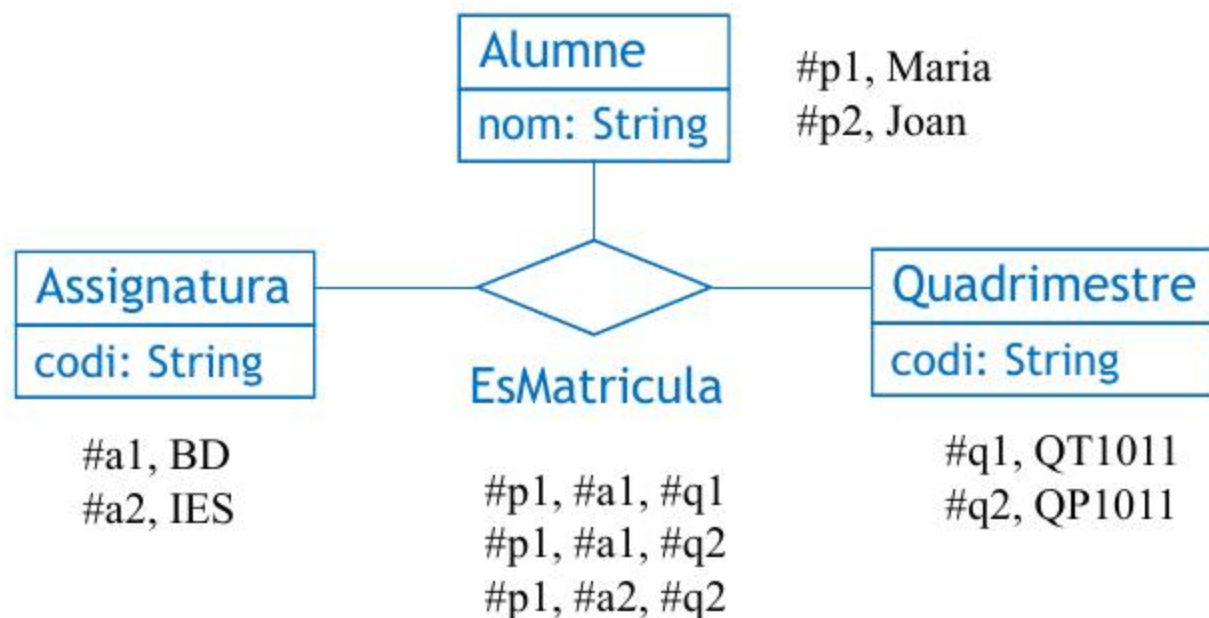
Es la representació de relacions entre dos o més objectes



Restriccions implícites de les associacions:

- Tota instància de l'associació conté exactament tants objectes com classes la defineixen
- Una associació no pot tenir instàncies repetides

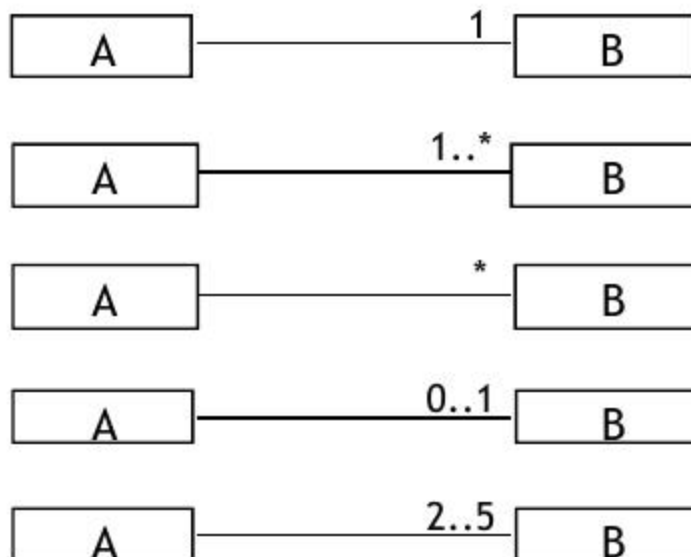
Associacions d'ordre superior a dos



Les restriccions implícites de les associacions també s'apliquen en aquest cas

Multiplicitats a les associacions binàries

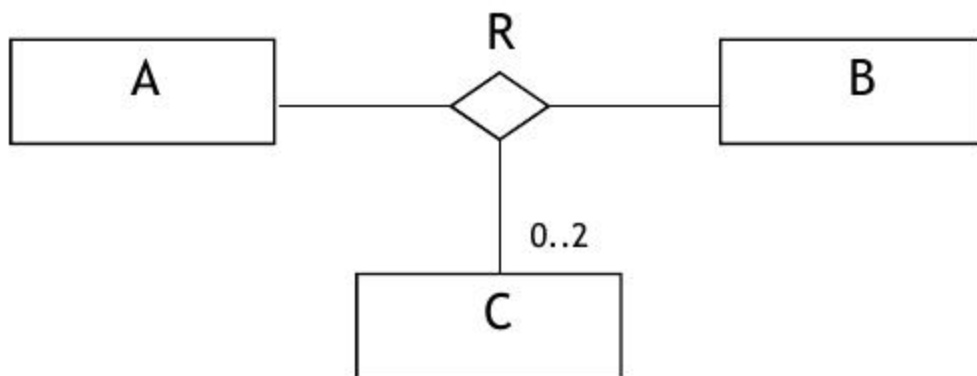
Per tota instància a de la classe A, la multiplicitat del costat B defineix el nombre mínim i màxim d'instàncies de B associades amb a .



Una multiplicitat és una restricció d'integritat gràfica que restringeix el nombre d'instàncies admeses per una associació

Multiplicitats a les associacions ternàries

Per qualsevol parella d'instàncies (a,b) , on a és instància de la classe A i b és instància de B , la multiplicitat al costat C defineix el nombre mínim i màxim d'instàncies de C associades amb la parella (a,b) via l'associació

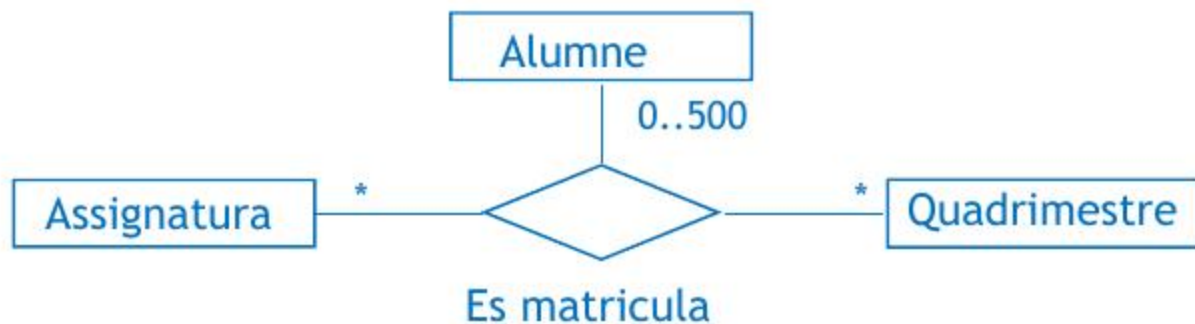


A una parella d'instàncies $(\#a1, \#b1)$ qualsevol, com a màxim li corresponen dues instàncies de C via R

Multiplicitats a les associacions ternàries: exemples



Segons aquest esquema, per tota parella d'assignatura i quadrimestre, hi ha d'haver com a mínim un professor reponsable.

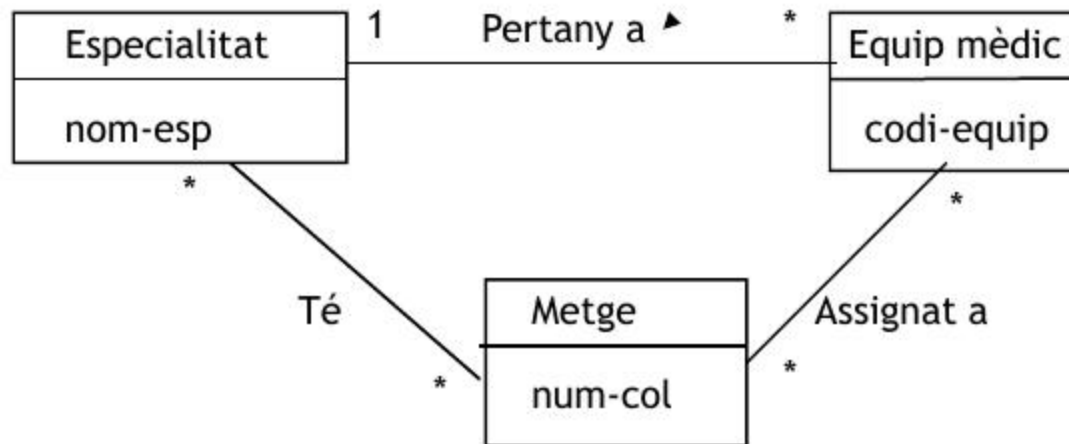


Aquest esquema permet que hi hagi alguna parella d'assignatura i quadrimestre, per la qual no hi ha cap alumne que s'hagi matriculat de l'assignatura en el quadrimestre.

Restriccions textuais

Les restriccions que no es poden especificar gràficament en UML s'especifiquen de forma textual amb llenguatge natural, OCL, etc.

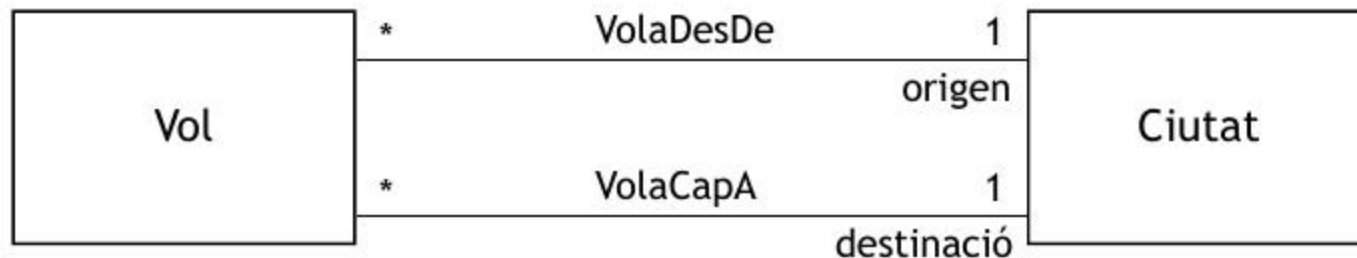
Com tota restricció, serveixen per limitar les instàncies admeses per l'esquema conceptual.



1. Claus externes: (Especialitat, nom-esp); (EquipMèdic, codi-equip); (Metge, num-col)
2. Un metge no pot estar assignat a un equip mèdic que pertany a una especialitat que el metge no té

Nom de rol a les associacions

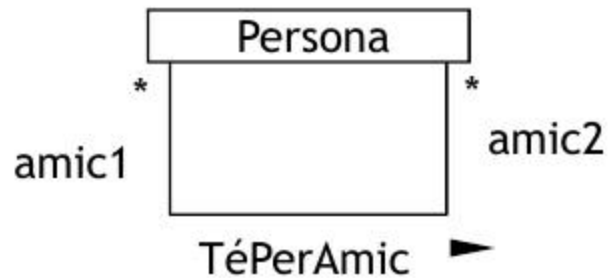
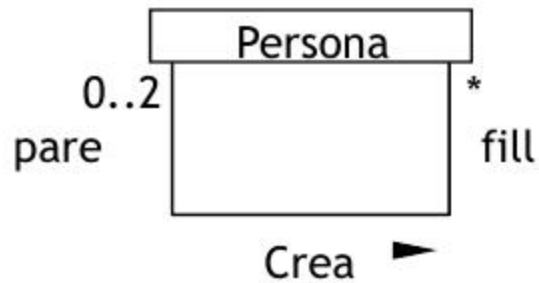
Cada extrem d'una associació és un rol, amb un nom i una multiplicitat.
El nom de rol identifica un cap de l'associació i descriu el paper jugat pels objectes d'aquest cap en l'associació.
És imprescindible posar-lo quan cal evitar ambigüitats.



nom de rol:
descriu el rol d'una ciutat en
l'associació VolaCapA

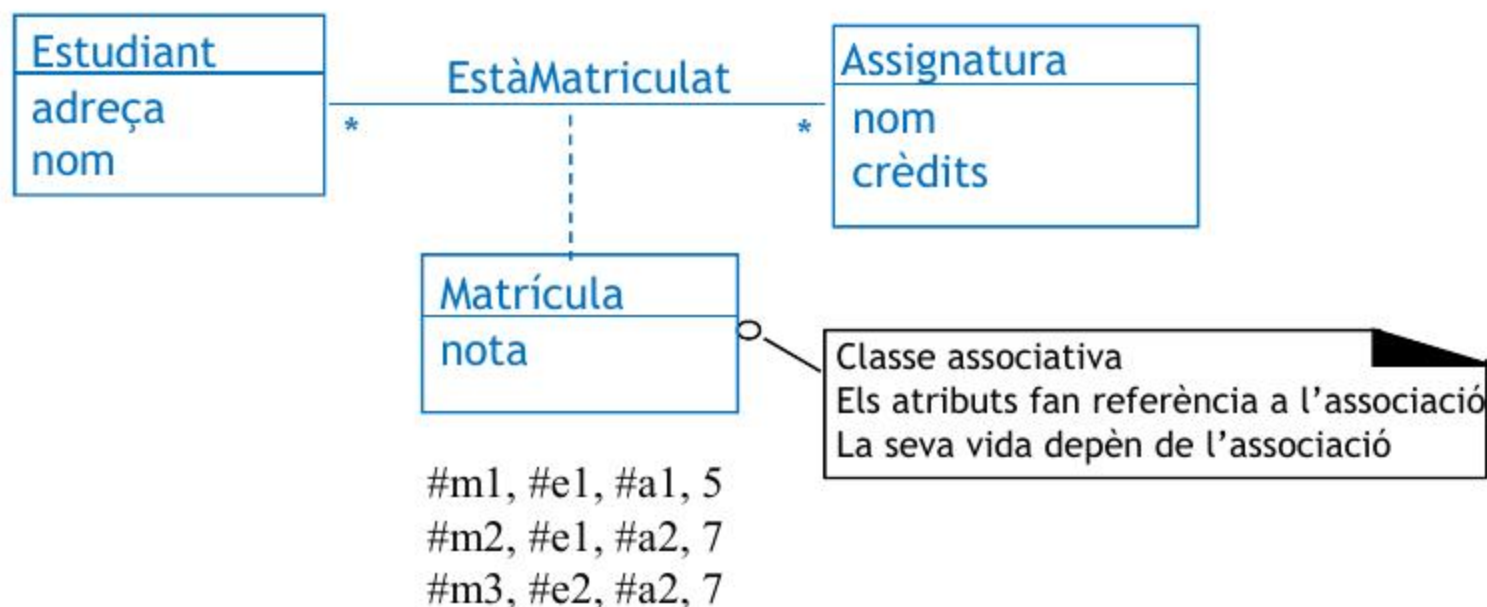
Associacions recursives

Associacions en les que una mateixa classe d'objectes hi participa més d'una vegada (amb papers diferents o no)



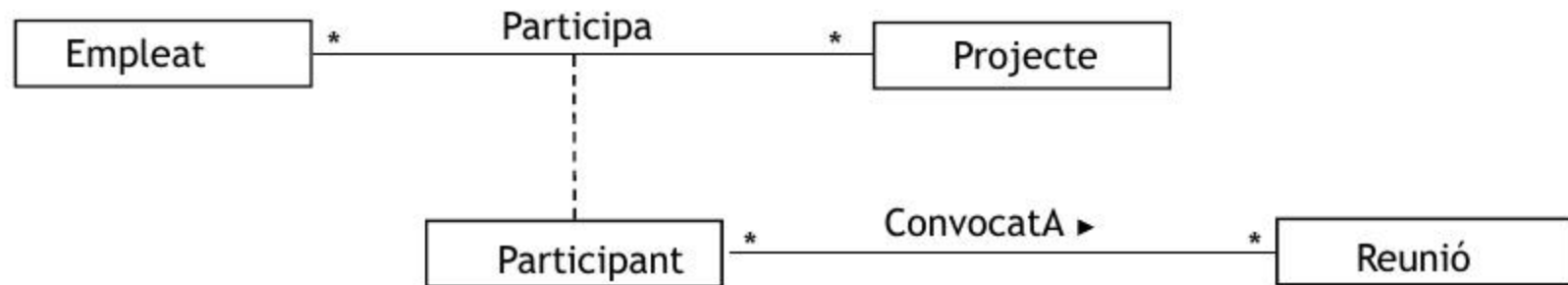
Classe associativa

Consisteix a veure una associació com una classe d'objectes: reïficació

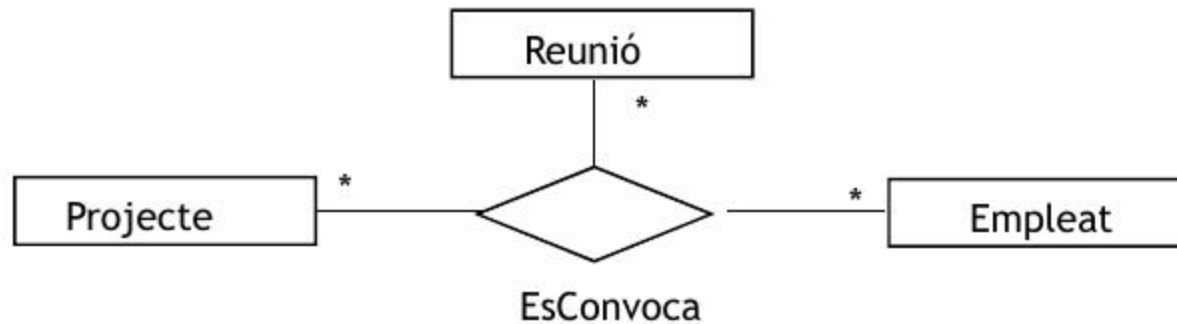


- Hi ha una correspondència 1:1 entre les instàncies de l'associació i les de la classe associativa
- Es mantenen les restriccions implícites de l'associació reïficada

Exemple de classe associativa



No és equivalent a:



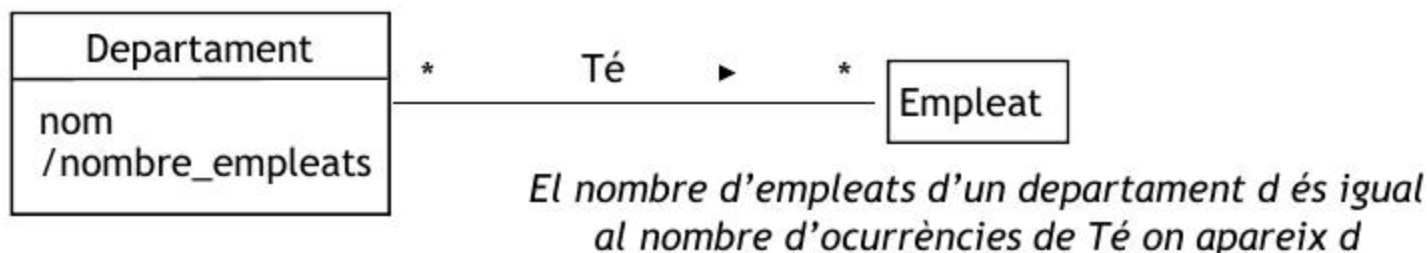
Informació derivada

Un atribut o una associació són **derivats** si es poden calcular a partir d'altres elements

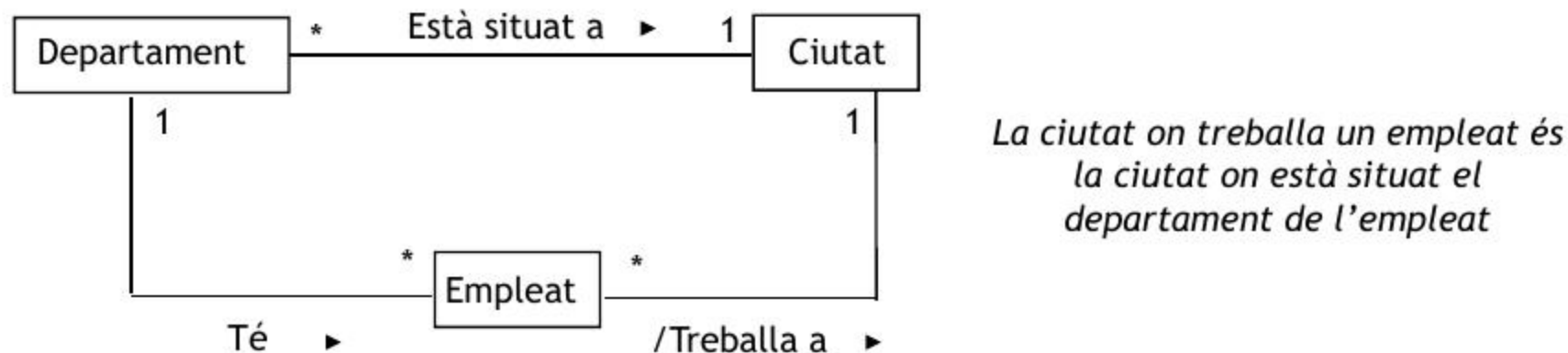
S'inclou quan millora la **claredat** del model conceptual

Una 'constraint' (regla de derivació) ha d'especificar com es deriva

Atribut derivat

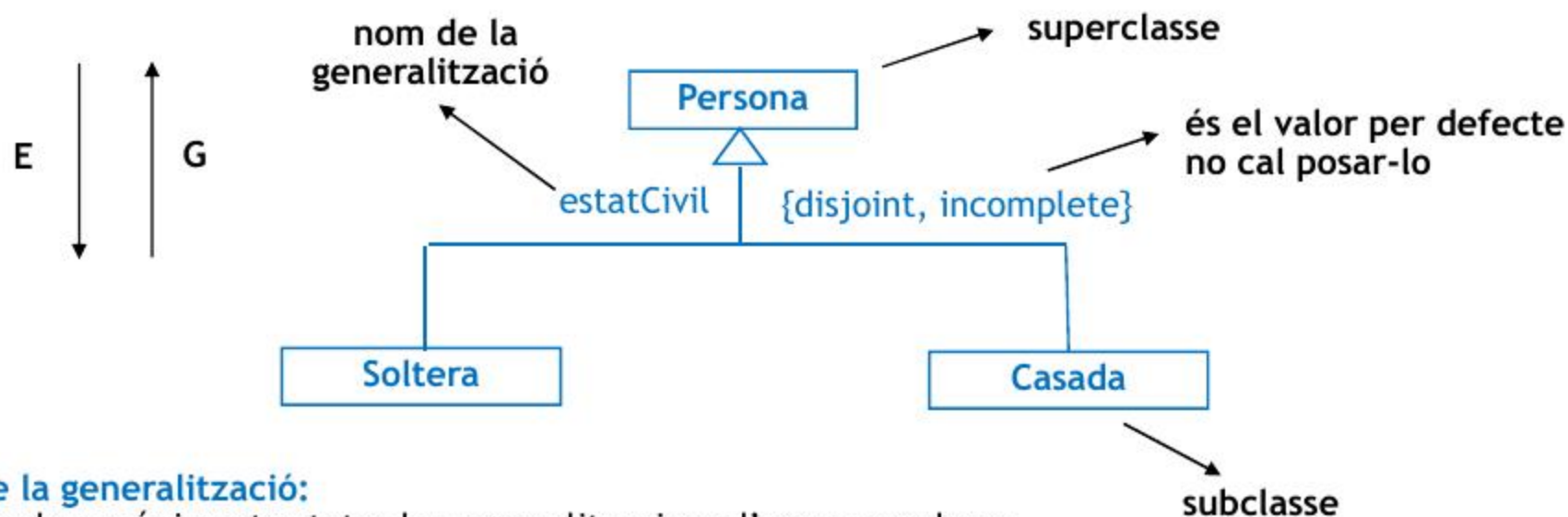


Associació derivada



Generalització / Especialització (1)

Permet identificar elements comuns entre els conceptes definint relacions de superclasse (objecte general) i subclasse (objecte especialitzat).



Nom de la generalització:

ha de ser únic entre totes les generalitzacions d'una superclasse.

Restriccions gràfiques:

disjoint - Un objecte de la superclasse no pot pertànyer a més d'una subclasse

overlapping - Un objecte de la superclasse pot pertànyer a més d'una subclasse

complete - Tota instància de la superclasse ho és també d'alguna de les seves subclasses

incomplete - Alguna instància de la superclasse no ho és de cap de les seves subclasses

La classificació pot ser **estàtica** o **dinàmica**

Generalització / Especialització (2)

Utilitat:

- Permet entendre els conceptes en termes més generals, refinats i abstractes
- Fa els diagrames més expressius

Restriccions implícites:

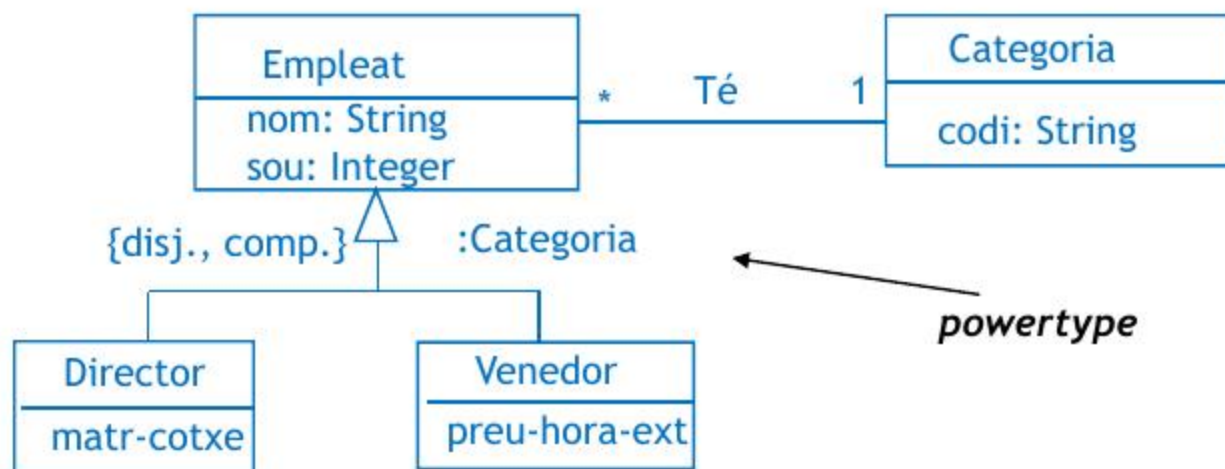
- Tots els objectes de la subclasse ho són també de la superclasse
- La definició de la superclasse és aplicable a la subclasse (atributs, associacions, restriccions d'integritat, etc.)

Criteris per definir una generalització:

- La subclasse té atributs o associacions addicionals
- Les subclasses tenen atributs o associacions que poden ser factoritzats i expressats a les superclasses
- Les subclasses potencials representen variacions d'un mateix concepte

Generalització / Especialització (3)

De vegades les subclasses depenen d'una altra classe del model

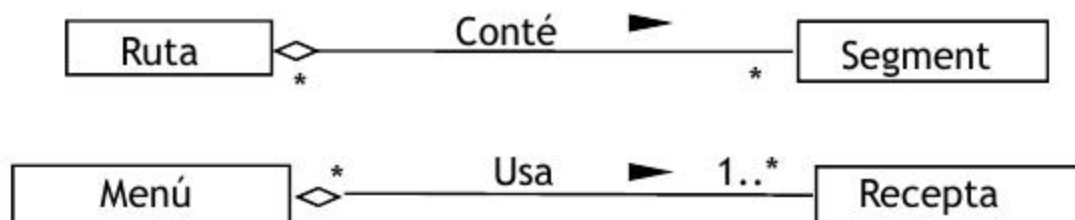


Powertype:

- Una superclasse C amb un *powertype* P té com a màxim tantes subclasses com instàncies té P.
- Si no es vol fer servir el *powertype*, cal afegir una restricció textual per cada subclasse.
p.ex. Tot empleat director ha de tenir la categoria de director

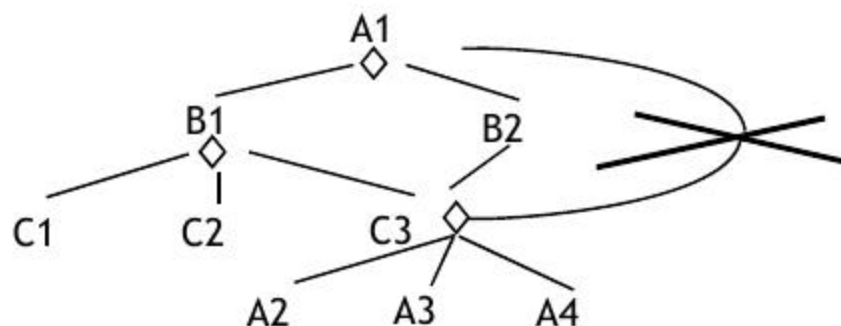
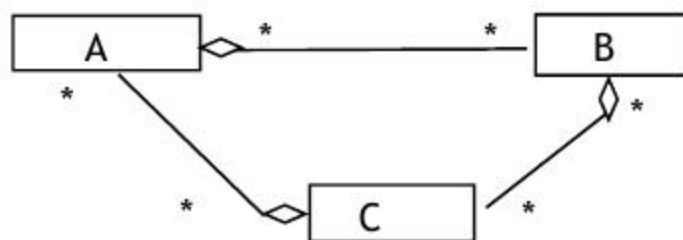
Agregació

És un tipus d'associació usat per modelar relacions “part-tot” entre objectes
El “tot” s'anomena composat i les “parts” components



La distinció entre associació i agregació és sovint subjectiva.

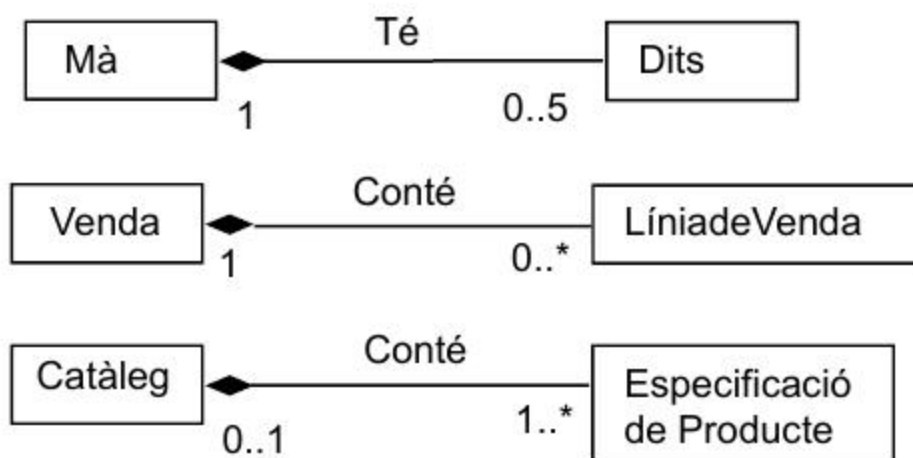
L'única restricció que afegeix l'agregació respecte l'associació és que les cadenes d'agregacions entre instàncies d'objectes no poden formar cicles.



Composició

La composició és un tipus d'agregació per al qual:

- La multiplicitat del cap compost pot ser com a màxim 1 (com a màxim un compostat posseeix un component)
- Si un “component” està associat a un “composat” i el “composat” s'esborra aleshores el “component” també s'ha d'esborrar

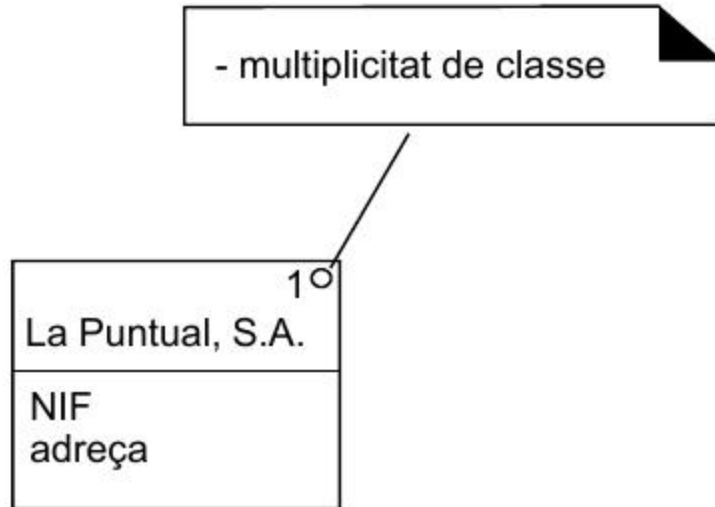


Multiplicitat de classe

La **multiplicitat de classe** estableix el rang de possibles cardinalitats per les instàncies d'una classe

Per defecte, és **indefinida**

En alguns casos, però, és útil establir una multiplicitat finita, **especialment** en casos de classes que poden tenir **una sola instància** (i que s'anomenen “singleton”)

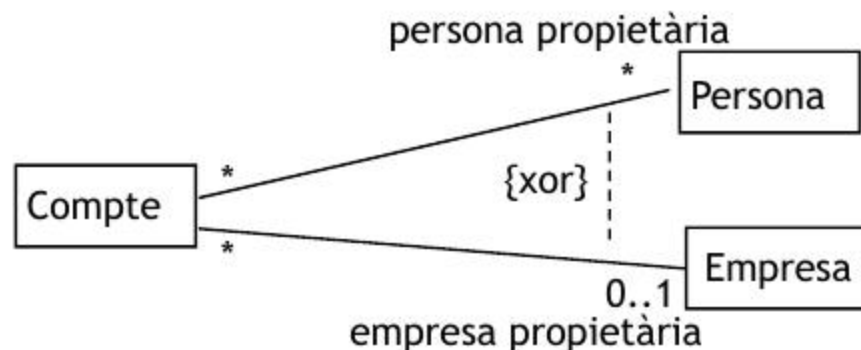


Altres restriccions gràfiques

Xor

Uneix diverses associacions lligades a una mateixa classe bàsica

Una instància de la classe bàsica pot participar com a màxim en una de les associacions unides per “xor”.



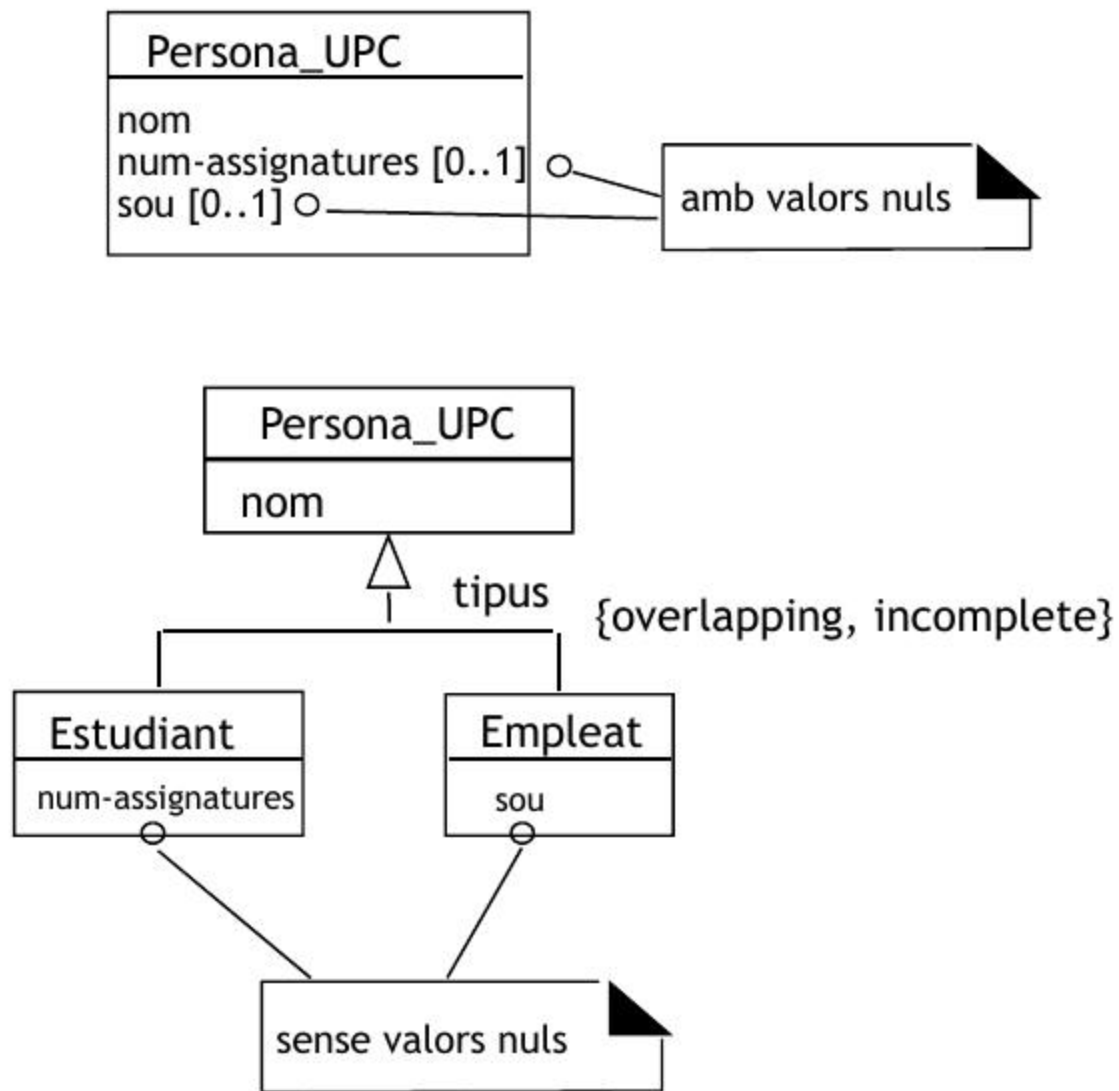
Subset

Indica que les instàncies d'una associació són un subconjunt de les d'una altra associació

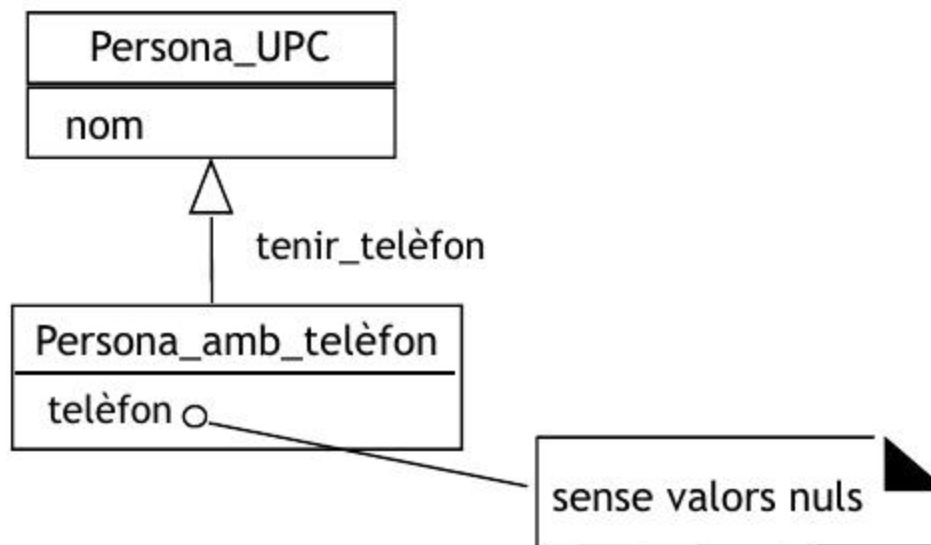
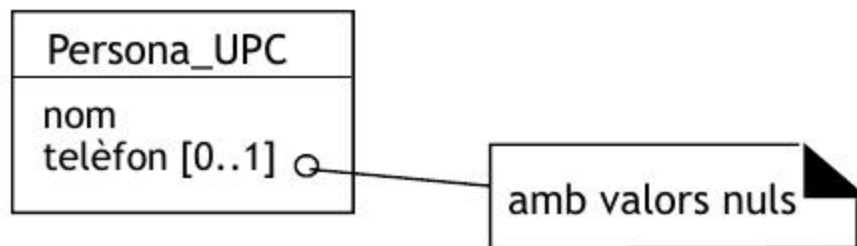


També es podria haver posat: usuari {subsets usuariAutoritzat}

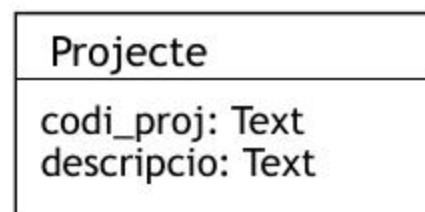
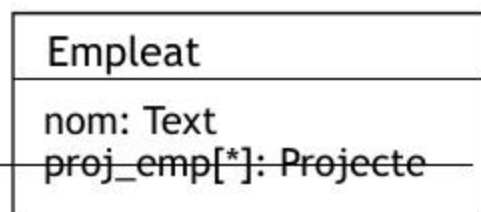
Exemples (I): quina solució és millor?



Exemples (II): quina solució és millor?



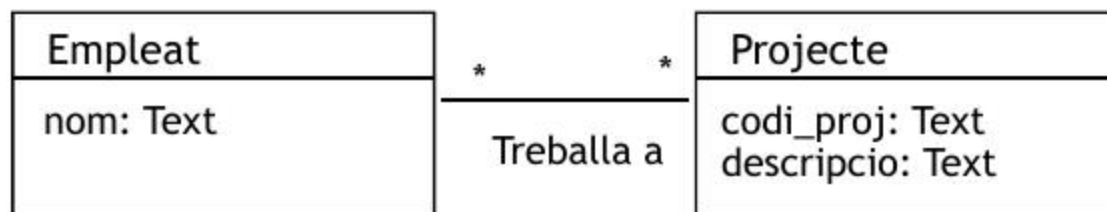
Exemples (III)



Un **atribut** no pot prendre **valors d'una de les classes** de l'esquema conceptual



Aquest cas és una **associació**



Excepció: Classe d'objectes de tipus temporal, com ara, Data, Any, etc.

Bibliografia

- Larman, C. *“Applying UML and Patterns. An Introduction to Object-oriented Analysis and Design”*, Prentice Hall, 2005, (3^a edició)
- Rumbaugh, J.; Jacobson, I.; Booch, G. *“The Unified Modeling Language Reference Manual”*, 2^a edició, Addison-Wesley, 2004.
- Olivé, A. *“Conceptual Modeling of Information Systems”*, Springer, 2007.