

Dolors Costal - M. Ribera Sancho
Ernest Teniente

Enginyeria del Software

Especificació

Especificació de sistemes orientats a objectes
amb la notació UML

Primera edició: febrer del 2000

Aquesta publicació s'acull a la política de normalització lingüística i ha comptat amb la col·laboració del Departament de Cultura i de la Direcció General d'Universitats, de la Generalitat de Catalunya.

En col·laboració amb el Servei de Llengües i Terminologia de la UPC

Disseny de la coberta: Manuel Andreu

© Els autors, 2000

© Edicions UPC, 2000
Edicions de la Universitat Politècnica de Catalunya, SL
Jordi Girona Salgado 31, 08034 Barcelona
Tel.: 934 016 883 Fax: 934 015 885
Edicions Virtuals: www.edicionsupc.es
A/e: edupc@sg.upc.es

Producció: CPET (Centre de Publicacions del Campus Nord)
La Cup. Gran Capità s/n, 08034 Barcelona

Dipòsit legal:
ISBN: 84-8301-366-5

Són rigorosament prohibides, sense l'autorització escrita dels titulars del copyright, sota les sancions establertes a la llei, la reproducció total o parcial d'aquesta obra per qualsevol procediment, inclosos la reprografia i el tractament informàtic, i la distribució d'exemplars mitjançant lloguer o préstec públics.

Índex

- Transparències

1. Introducció a l'Enginyeria del Software	11
2. Requeriments i Especificació	19
3. Introducció a l'orientació a objectes	28
3.1 - Motivació i orígens	28
3.2 - Aspecte Estàtic	33
3.3 - Aspecte Dinàmic.....	34
3.4 - El llenguatge UML (Unified Modeling Language)	37
4. El llenguatge UML.....	39
4.1 - Model Conceptual en UML.....	39
4.2 - El llenguatge OCL (Object Constraint Language)	56
4.3 - Model de Casos d'Ús en UML	65
4.4 - Model del Comportament en UML	75
4.5 - Model dels Estats en UML.....	91
5. El Procés Unificat de Desenvolupament de Software.....	97

- Recull d'exercicis

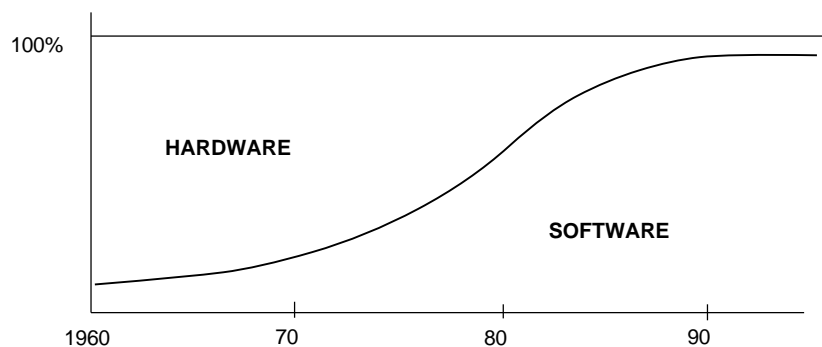
1. Introducció.....	105
2. Model Conceptual en UML.....	106
3. El llenguatge OCL	127
4. Altres models d'UML	128

Transparències

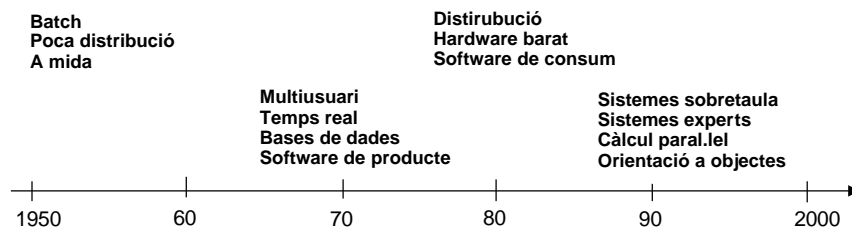
Introducció a l'Enginyeria del Software

- Software
 - Importància
 - Evolució
 - Característiques
 - Crisi del software
- Enginyeria del Software
 - Definicions
 - Característiques
 - Visió genèrica
- Paradigmes
 - Cicle de vida clàssic
 - Prototipatge
 - Model en espiral
- Bibliografia

Evolució de Costos del Hardware i del Software



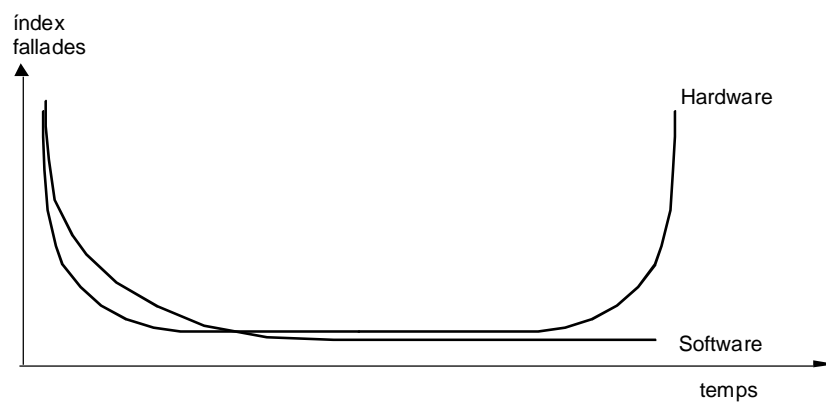
Evolució del software



Característiques del software

- Es desenvolupa, no es fabrica
- No s'espalla
- Manteniment més difícil que el hardware
- Es construeix a mida, no es reusa

Fallades H/S



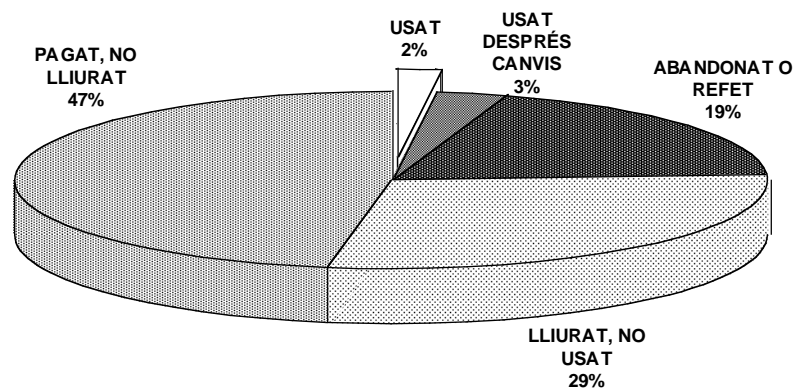
Aplicacions del Software

- Sistemes
- Temps real
- Gestió
- D'enginyeria
- Científic
- Empotrat
- De PC
- D'Intel·ligència Artificial

Crisi del software

- Crisi?
 - Aflicció crònica !!!
- Problemes:
 - Evolució contínua
 - Insatisfacció dels usuaris
 - Poca qualitat
 - Manteniment difícil
- Causes:
 - Naturalesa del software
 - Complexitat
 - Gestió
- Mites:
 - De gestió
 - Dels clients
 - Dels dissenyadors

Resultat de projectes de software



Enginyeria del software

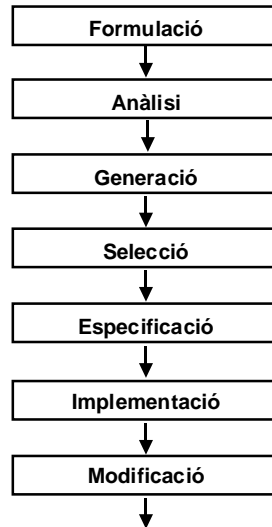
Establiment i ús de principis de l'enginyeria orientats a obtenir software:

- Econòmic
- Fiable
- Que funcioni eficientment
- Que satisfaci les necessitats dels usuaris

Un enginyer ...

- Disposa d'un ventall de tècniques provades que donen resultats precisos.
- Es preocupa de la fiabilitat i del rendiment.
- Tracta de reduir costos i complexitat.
- Basa els seus models en teories matemàtiques sòlides.
- Construeix prototipus dels nous dissenys.
- Utilitza diagrames formals.

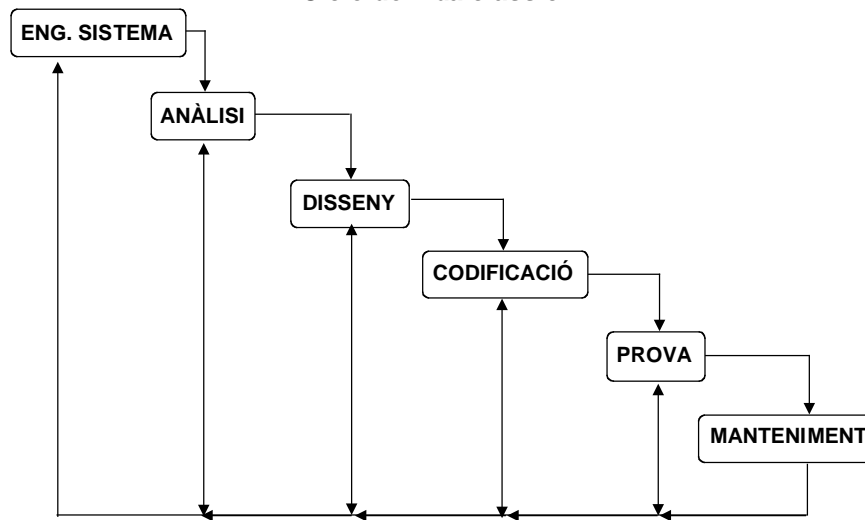
El procés de l'enginyeria



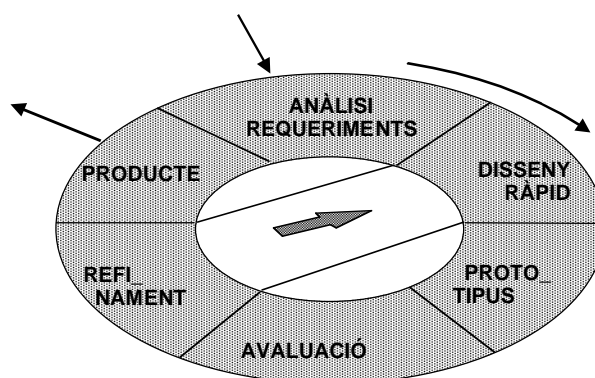
Visió genèrica de l'Enginyeria del Software

- Definició:
 - Anàlisi del sistema
 - Planificació del projecte
 - Anàlisi de requeriments del software
- Desenvolupament:
 - Disseny del software
 - Codificació
 - Prova
- Manteniment:
 - Correcció
 - Adaptació
 - Millora

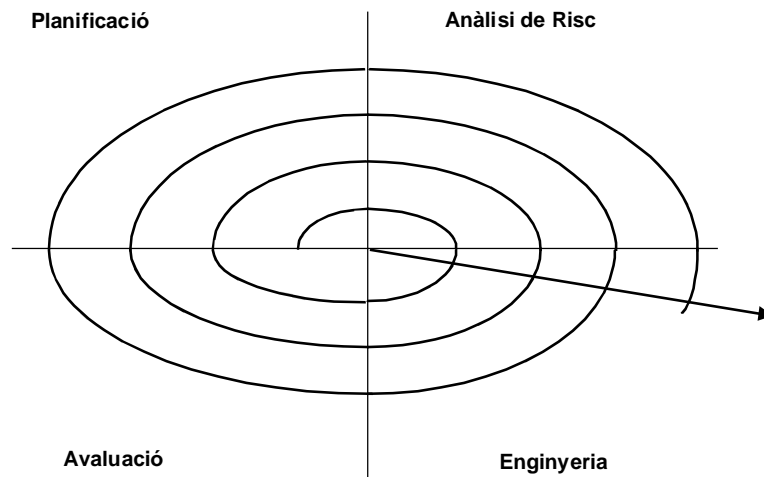
Cicle de vida clàssic



Prototipatge



Model en Espiral



Bibliografia

- R.S. Pressman
Ingeniería del Software: Un enfoque práctico.
4a edició.
McGraw Hill, 1997. (Cap. 1)

Requeriments i Especificacions

- **Determinació dels requeriments del software**
 - Requeriments del sistema versus requeriments del software
 - Etapes
 - Estratègies
- **Especificacions de sistemes software**
 - Requeriments funcionals i no funcionals
 - Propietats desitjables de les especificacions
 - Estàndards de documentació
- **Bibliografia**

Requeriments del sistema vs. requeriments del software

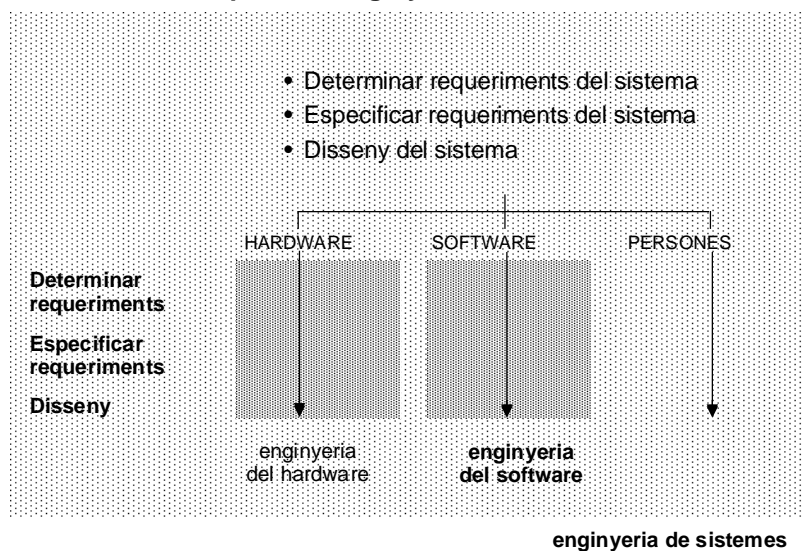
Requeriment: Condició o capacitat necessitada per un usuari per tal de solucionar un problema o aconseguir un objectiu

- La solució al problema es pot realitzar amb software, hardware, manualment, o amb una combinació de tots tres.
- Si la solució és composta, abans de dissenyar els detalls d'un component software concret, cal dissenyar el sistema global.

Exemple de sistema compost: refinaria automatitzada

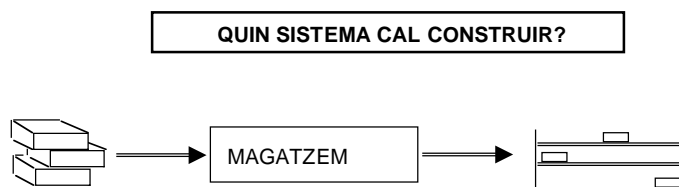
Exemple de sistema només software: control d'estocs

Etapas de l'enginyeria de sistemes



Determinar requeriments del sistema global

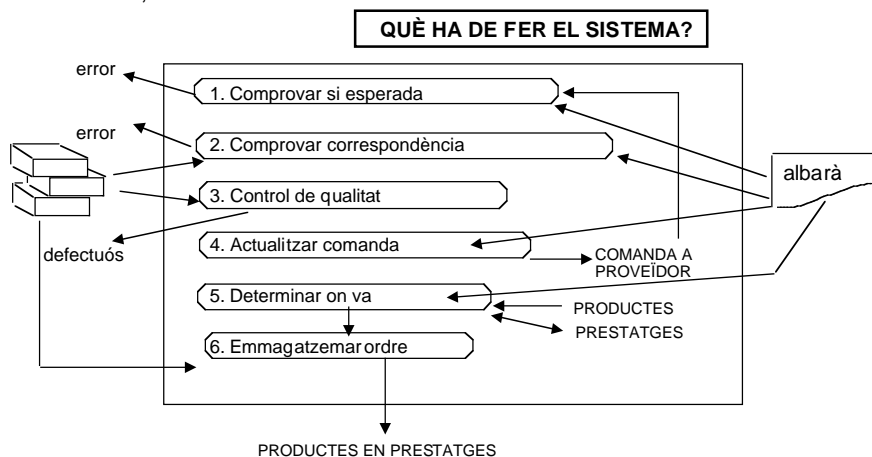
- Comprensió dels objectius i necessitats de l'usuari
- Definir el conjunt de sistemes que podrien satisfer les necessitats o objectius i avaluar-los
- Triar el sistema més adient



Sistema que rep i verifica els productes demanats als proveïdors, i els emmagatzema als prestatges

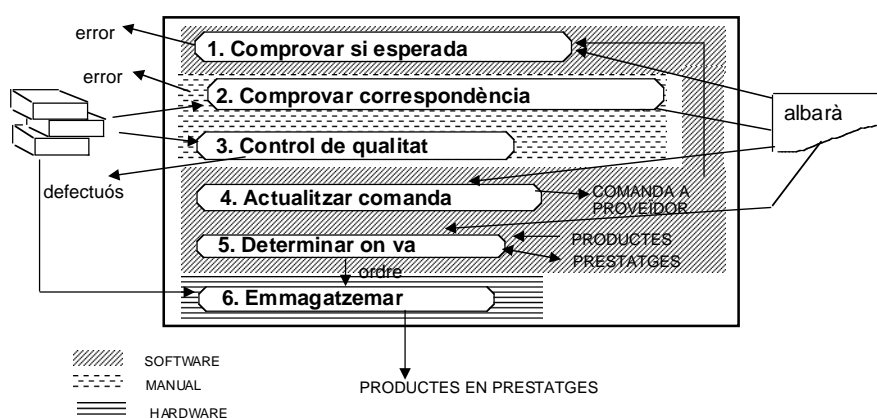
Especificar els requeriments del sistema global

- Descriure el comportament extern del sistema, des del punt de vista de l'usuari, o de l'entorn

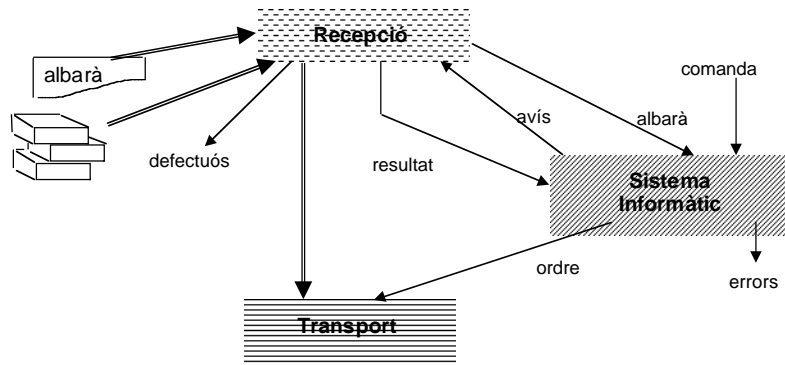


Disseny del sistema

- Determinar l'arquitectura general del sistema que millor satisfà els requeriments, en termes de:
 - components físics: hardware, software, persones
 - comunicació entre ells

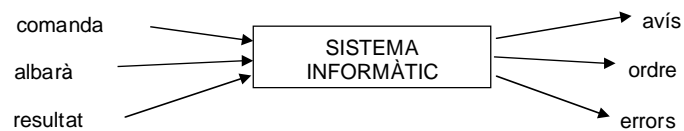


Disseny del sistema



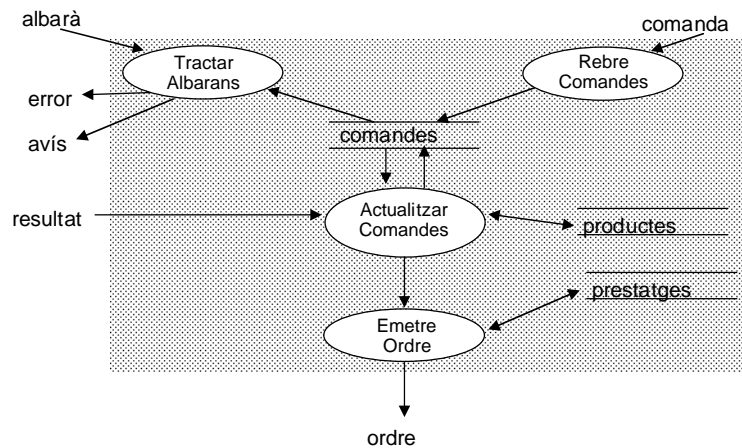
Determinar requeriments del software

- És el subconjunt dels requeriments del sistema global que han estat assignats al component software concret



Especificar els requeriments del software

- Descriure amb detall el comportament extern del component software concret



Determinació dels requeriments del software: Estratègies

- Demanar-ho a l'usuari
- Treure-ho d'un sistema software existent
- Sintetitzar-ho a partir del sistema global
- Descobrir-ho mitjançant experimentació

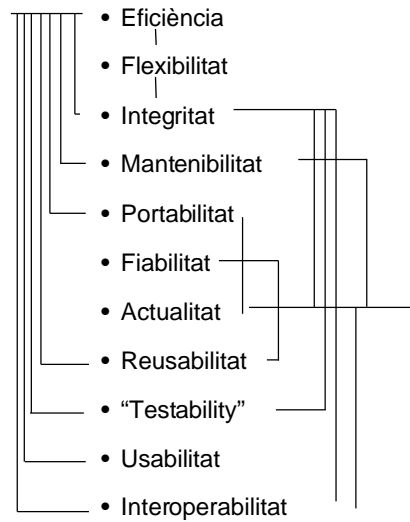
Requeriments del software

- Funcionals
 - Descriuen totes les entrades i sortides, i la relació entre ambdues
 - de dades
 - de procés
- No funcionals
 - Defineixen les qualitats generals que ha de tenir el sistema en realitzar la seva funció
 - Qualitat
 - Interfícies
 - Rendiment extern
 - Econòmics
 - Estructurals
 - Polítics

Factors de qualitat del software

- Eficiència
- Flexibilitat
- Integritat
- Mantenibilitat
- Portabilitat
- Fiabilitat
- Actualitat
- Reusabilitat
- "Testability"
- Usabilitat
- Interoperabilitat

Factors de qualitat del software: conflictes



Propietats desitjables de les especificacions

- No ambigües
- Completes
- Verificables
- Consistents
- Modificables
- "Traçables"
- Usables durant l'operació i el manteniment

Com organitzar una especificació de requeriments? estàndard de documentació ANSI/IEEE

1. Introducció
 - 1.1 Propòsit de l'especificació
 - 1.2 Abast del producte
 - 1.3 Definicions i abreviatures
 - 1.4 Referències
 - 1.5 Visió general de l'especificació
2. Descripció General
 - 2.1 Perspectiva del producte
 - 2.2 Funcions del producte
 - 2.3 Característiques dels usuaris
 - 2.4 Restriccions generals
 - 2.5 Supòsits i dependències
3. Requeriments específics
4. Apèndixos
5. Índex

Com organitzar una especificació de requeriments? estàndard de documentació ANSI/IEEE

3. Requeriments específics
 - 3.1 Requeriments funcionals
 - 3.1.1 Requeriment funcional 1
 - 3.1.1.1 Introducció
 - 3.1.1.2 Entrades
 - 3.1.1.3 Tractament
 - 3.1.1.4 Sortides
 - 3.1.2 Requeriment funcional 2
 - 3.2 Requeriments d'interfícies externes
 - 3.2.1 D'usuaris
 - 3.2.2 Hardware
 - 3.2.3 Software
 - 3.2.4 Comunicacions
 - 3.3 Requeriments de rendiment
 - 3.4 Restriccions de disseny
 - 3.5 Atributs
 - 3.6 Altres requeriments

Bibliografia

- K.Shumate, M.Keller
Software Specification and Design - A Disciplined Approach for Real-Time Systems.
John Wiley, 1992. (Cap. 3)
- A. M. Davis
Software Requirements - Objects, Functions and States.
Prentice-Hall, 1993. (Cap. 1-5)

Introducció a l'orientació a objectes

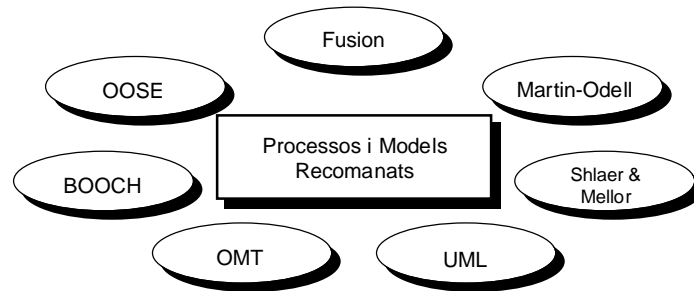
- Motivació i orígens
- Visió d'un sistema software
 - Aspecte estàtic
 - Aspecte dinàmic
- La notació UML

Motivació

- Aparició dels llenguatges de programació orientats a objectes.
 - SIMULA: finals dels 1960's
 - SMALLTALK: principis dels 1970's
- L'ús d'aquests llenguatges requereix un nou enfocament d'anàlisi i de disseny.
- Altres factors:
 - Desenvolupament de noves aplicacions
 - Èmfasi principal en l'estructura de dades

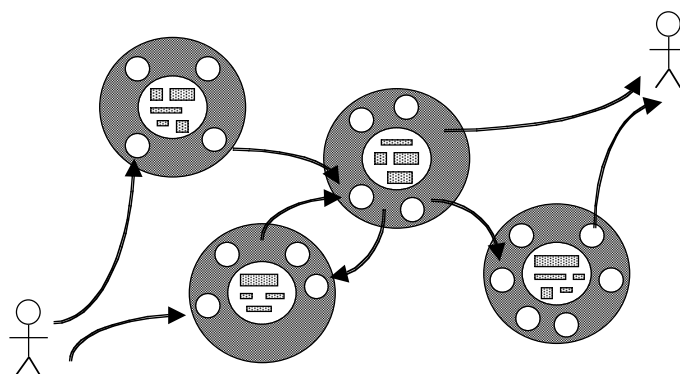
**Aparició dels primers mètodes d'anàlisi i de disseny
orientats a objectes**

Orígens



Rumbaugh, Blaha, Premerlani (OMT) - 1991
Coad, Yourdon - 1991
Shlaer, Mellor - 1992
Booch - 1992
Odell, Martin - 1992
Jacobson (OOSE) - 1993
Fusion - 1994
Booch, Rumbaugh, Jacobson (UML) - 1997

Visió d'un sistema software



Objectes

Objecte:

Entitat que existeix al món real

Tenen identitat i són distingibles entre ells



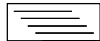
l'avió amb
matrícula 327



una poma



un semàfor



la factura 3443

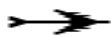


l'avió amb
matrícula 999

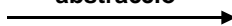
Classes d'objectes

Classe d'objectes: descriu un conjunt d'objectes amb:

- les mateixes propietats
- comportament comú
- idèntica relació amb altres objectes
- semàntica comuna



abstracció



classe avió

Avió

Abstracció: eliminar distincions entre objectes per a poder observar aspectes comuns

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

Atributs

Atribut: és una propietat compartida pels objectes d'una classe.

Exemples:

Persona => nom, adreça, telèfon, ...

Avió => model, capacitat, color, ...

Persona
Nom
Adreça
Telèfon
Data_naixement
Estat

Avió
Model
Capacitat
Color

- Cada atribut té un valor (probablement diferent) per cada instància d'objecte
- Els atributs poden ser bàsics o derivats

Operacions (I)

Operació: és una funció o transformació que es pot aplicar als objectes d'una classe.

Persona
Nom
Adreça
Telèfon
Canvi_adreça
Canvi_feina

Avió
Model
Capacitat
Colors
Reparar
Moure

- Les operacions d'un objecte són invocades pels altres objectes

Mètode: especificació procedural (implementació) d'una operació dins d'una classe.

Encapsulament: consisteix a separar els aspectes externs d'un objecte dels detalls d'implementació.

Operacions (II)

- En les operacions, cal indicar també el tipus dels arguments i del resultat.

Triangle
Color
Posició
Girar (angle)
seleccionar (p:Punt):Booleà

Quadrat
Color
Posició
Girar (angle)

Polimorfisme:

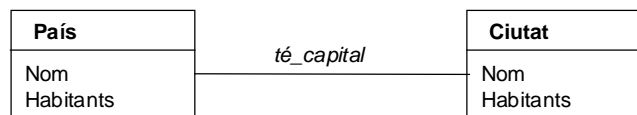
- una mateixa operació es pot aplicar a diferents classes
- la seva implementació depèn de cada classe

Associacions

Associació:

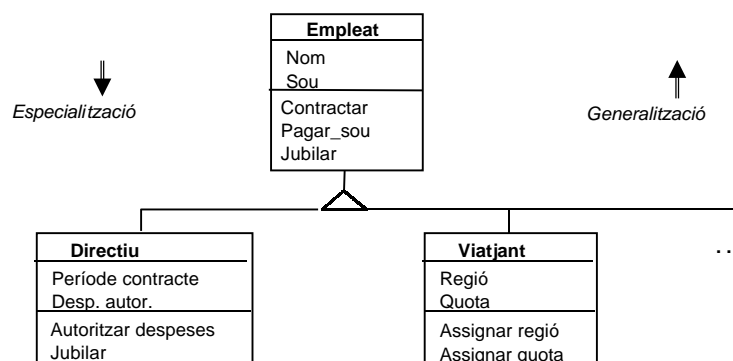
defineix la manera d'enllaçar o connectar objectes de diferents classes

Exemple: Un país té una única capital.



Generalització

Generalització: és l'acte o resultat de distingir un concepte que és més general que un altre



Herència: permet que l'estructura de dades i les operacions d'una classe siguin accessibles a una subclasse

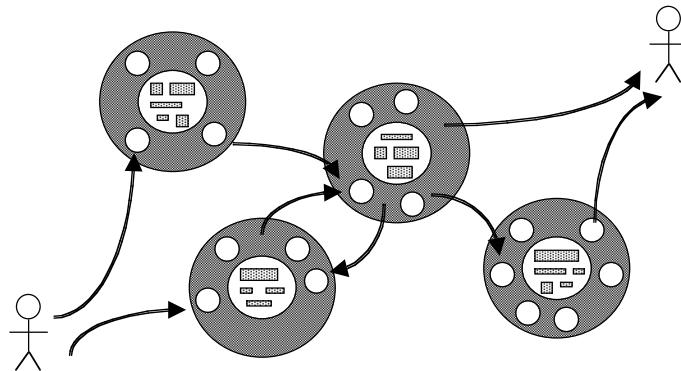
Orientació a objectes (I)

Aspecte estàtic: descriu l'estructura estàtica dels objectes del sistema i les seves interrelacions

	Intra - objecte	Inter - objectes
Aspecte estàtic	Classes d'objectes Atributs Operacions	Associació Generalització ...

Descripció del comportament (I)

Els objectes es comuniquen mitjançant la invocació d'operacions d'altres objectes



Descripció del comportament (II)

Aspecte dinàmic (de comportament): descriu els aspectes d'un sistema que canvien amb el temps

L'**aspecte dinàmic** d'un sistema descriu:

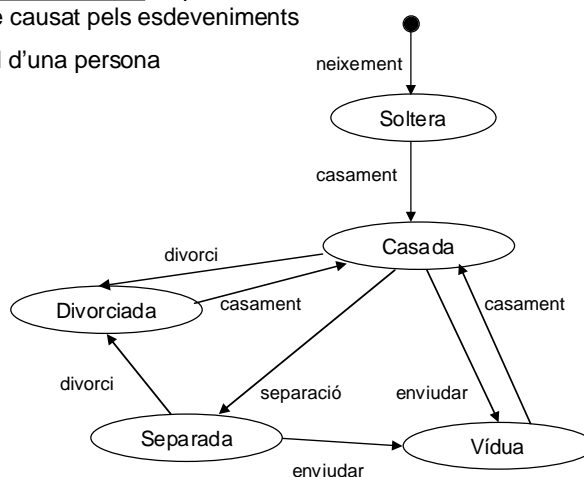
- Interaccions entre els objectes
- Possibles estats d'un objecte
- Transicions entre estats
- Quins esdeveniments es produeixen
- Quines operacions s'executen

Hi ha molta divergència entre els mètodes actuals a l'hora de tractar l'aspecte dinàmic

Descripció del Comportament (III)

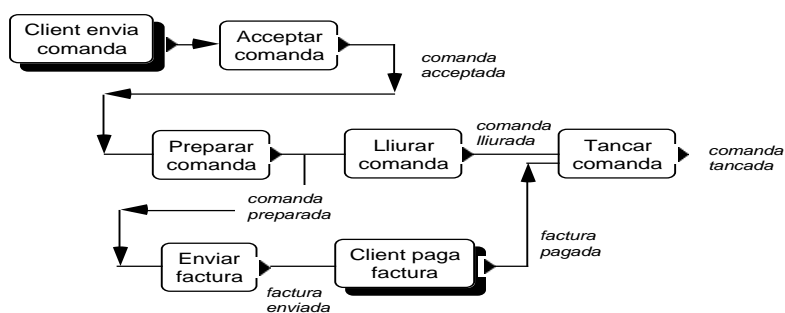
Diagrama de transició d'estats: especifica el canvi d'estat d'un objecte causat pels esdeveniments

Exemple: estat civil d'una persona



Descripció del comportament (IV)

Esquema d'esdeveniments: permet especificar la interacció entre diferents objectes (usat pel mètode de Martin i Odell [MO92])



Orientació a objectes (II)

Aspecte estàtic: descriu l'estructura estàtica dels objectes del sistema software i les seves interrelacions

Aspecte dinàmic (de comportament): descriu els aspectes d'un sistema software que canvien amb el temps

	Intra - objecte	Inter - objectes
Aspecte estàtic	Classes d'objectes Atributs Operacions	Associació Generalització ...
Aspecte dinàmic	Diagrama de transició d'estats	Esquema d'esdeveniments

Anàlisi i disseny orientats a objectes

Anàlisi:

- Creació d'una especificació del problema i dels requeriments
- **Què** ha de fer el sistema software

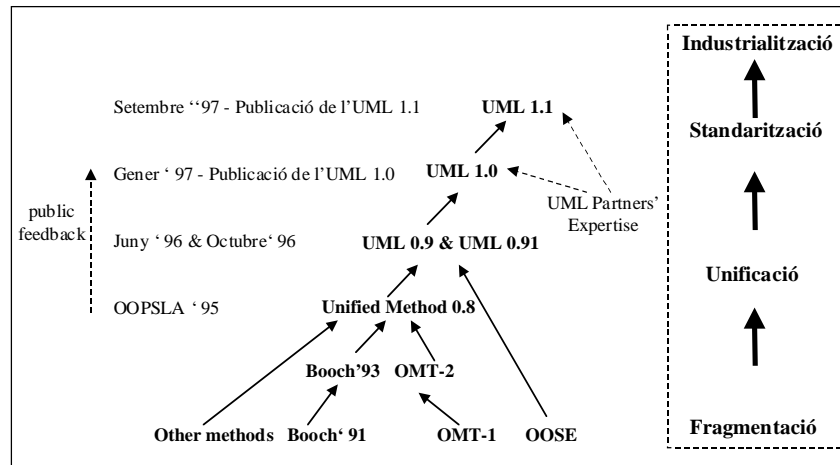
Disseny:

- definició d'una solució software que satisfaci els requeriments
- **Com** ho farà el sistema

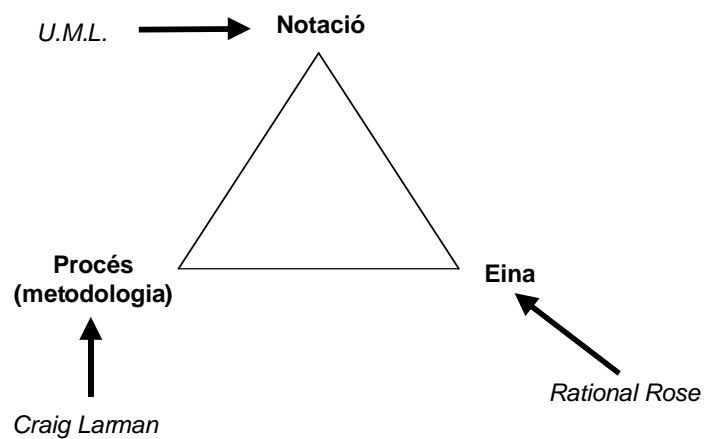
... orientats a objectes

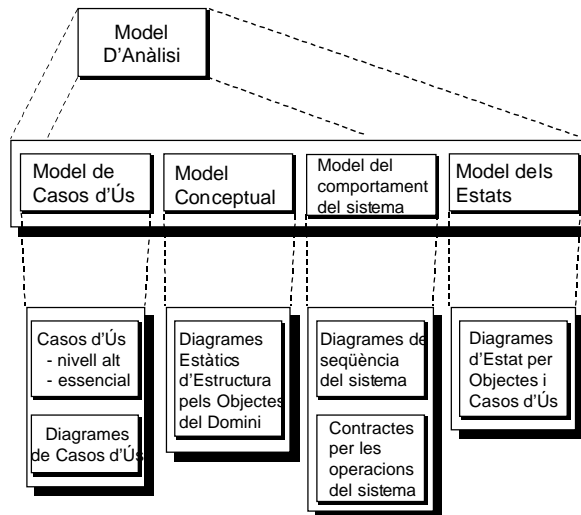
- S'usen els mateixos conceptes a l'anàlisi i al disseny
- És difícil determinar on acaba l'anàlisi orientada a objectes i on comença el disseny:
 - estratègia de desenvolupament iterativa
 - diferències de criteris segons els autors

El llenguatge UML (Unified Modeling Language)



El triangle de l'èxit

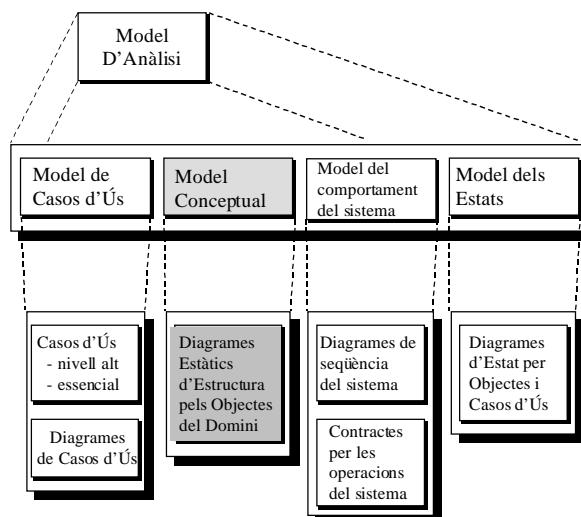


Model d'Anàlisi (especificació)

Model Conceptual en UML

- Introducció
- Objectes i classes d'objectes
- Atributs
- Associacions
- Classe associativa
- Generalització/Especialització
- Agregació i composició
- Ampliacions
- Exemples

Model d'Anàlisi (especificació)



Model Conceptual

Es 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.

Objectes

Objecte:

Entitat que existeix al món real

Tenen identitat i són distingibles entre ells



l'avió amb
matrícula 327



una poma



un semàfor



la factura 3443

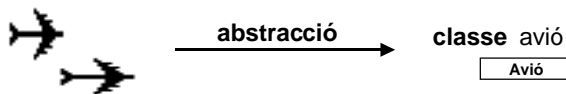


l'avió amb
matrícula 999

Classe d'objectes

Classe d'objectes: descriu un conjunt d'objectes amb:

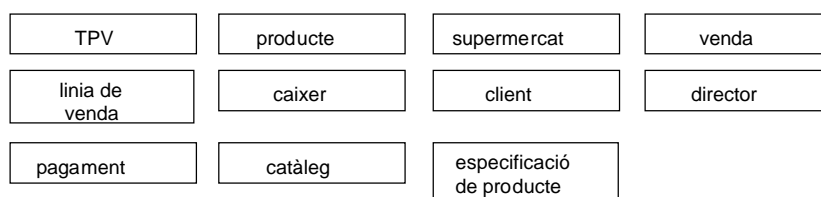
- les mateixes propietats
- comportament comú
- idèntica relació amb altres objectes
- semàntica comuna



Abstracció: eliminar distincions entre objectes per a poder observar aspectes comuns

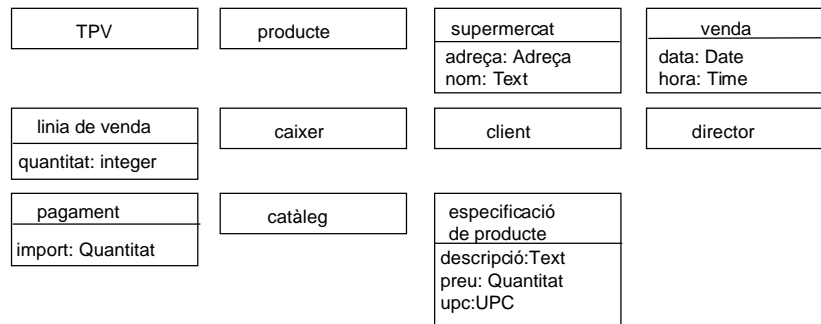
Els objectes d'una classe tenen les mateixes propietats i els mateixos patrons de comportament

Diagrama de classes



Atributs

Es una propietat compartida pels objectes d'una classe

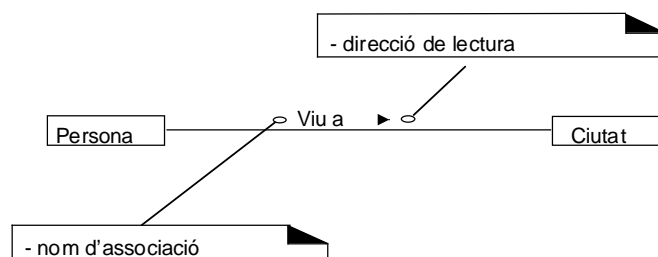


Els atributs:

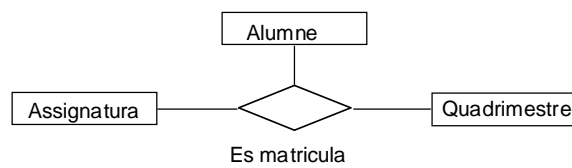
- Poden prendre valors **nuls**
- Poden ser **multivaluats**

Associacions

Es la representació de relacions entre dos o més objectes

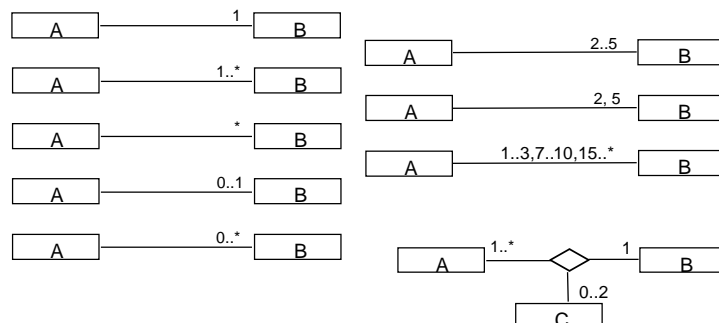


Associacions d'ordre superior a dos



Associacions - Multiplicitat

Defineix quantes instàncies d'una classe A es poden associar amb una instància de la classe B en un moment de temps determinat

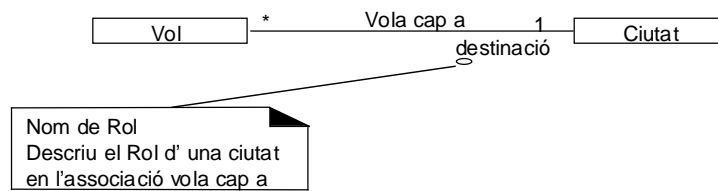


Nom de rol a les associacions

Cada extrem d'una associació es un rol, que té diverses propietats com:

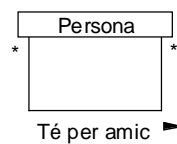
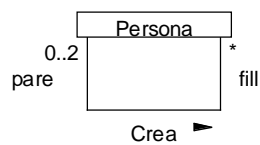
- nom
- multiplicitat

El nom de rol identifica un cap de l'associació i descriu el paper jugat pels objectes en l'associació.



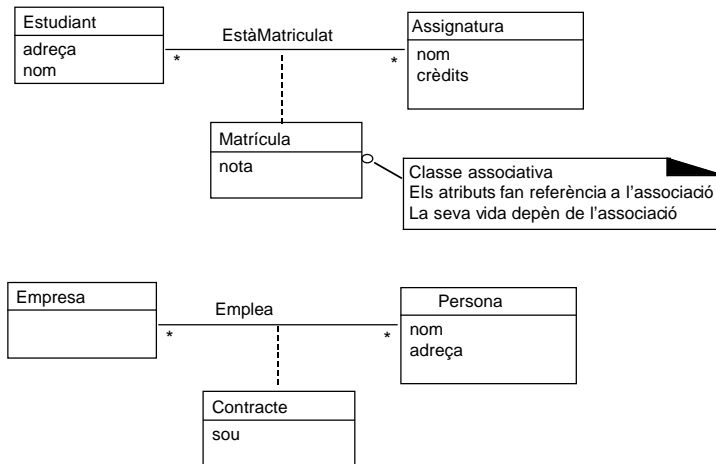
Associacions recursives

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

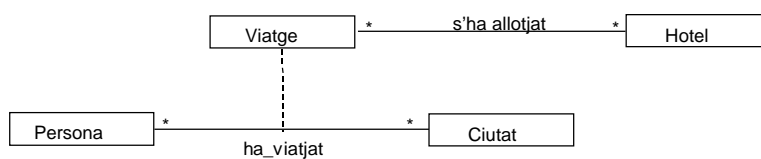


Classe associativa

Representa una associació que es pot veure com una classe

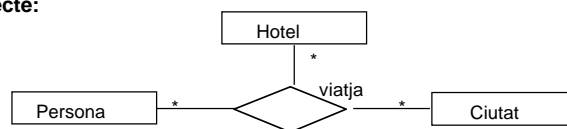


Exemple classe associativa



(Si una persona P ha viatjat algun cop a una ciutat C, existeix una ocurrència de l'associació ha_viatjat)

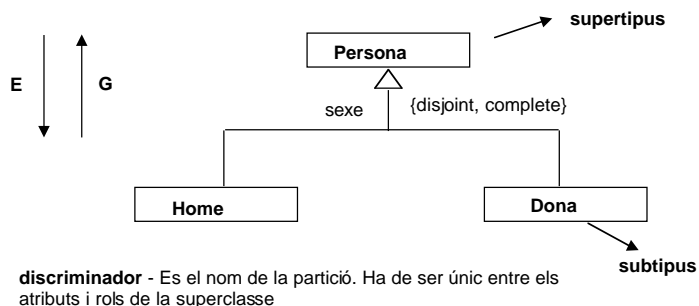
Incorrecte:



(Hi ha viatges sense allotjament en hotel)

Generalització/Especialització

Identificar elements comuns entre els objectes definint relacions de supertipus (objecte general) i subtipus (objecte especialitzat).



discriminador - Es el nom de la partició. Ha de ser únic entre els atributs i rols de la superclasse

disjoint - Un descendent no ho pot ser en més d'una subclasse

overlapping - Un descendent pot ser de més d'una subclasse

complete - S'han especificat totes les subclasses

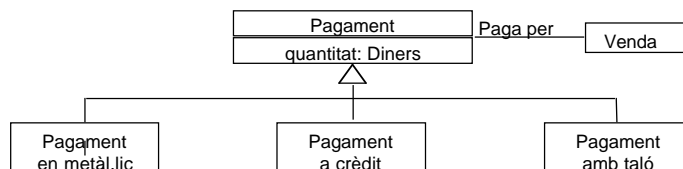
incomplete - La llista de subclasses és incompleta

Generalització/Especialització

Permet entendre els conceptes en termes més generals, refinats i abstractes.

Fa els diagrames més expressius.

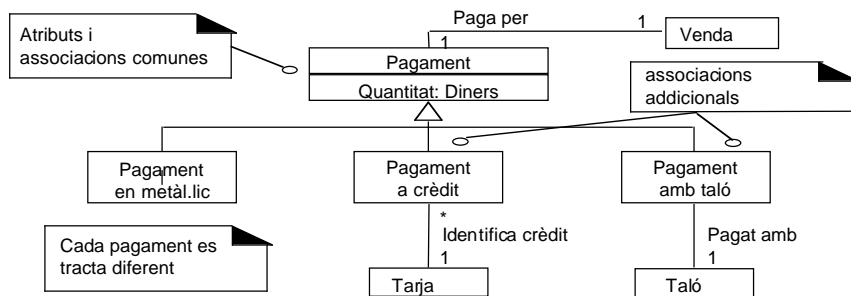
- Tots els membres del subtipus són membres del supertipus.
- El 100% de la definició del supertipus ha de ser aplicable al subtipus.
 - atributs
 - associacions
 - (- operacions)



Generalització/Especialització

Motivacions per particionar un tipus en subtipus

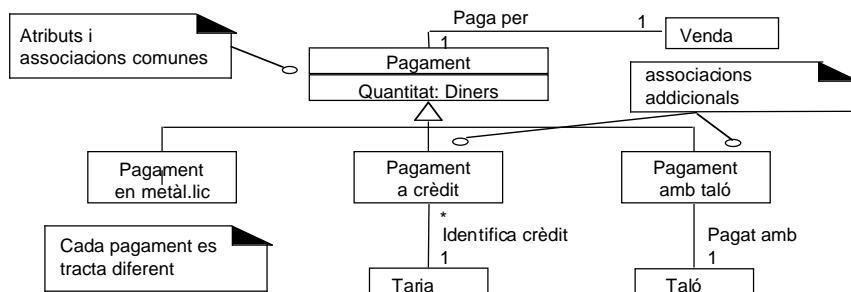
- El subtipus té atributs addicionals.
- El subtipus té associacions addicionals.
- El subtipus és tractat o manipulat de forma diferent d'altres subtipus.
- El subtipus es comporta de manera diferent del supertipus o dels altres subtipus.



Generalització/Especialització

Motivacions per definir un supertipus:

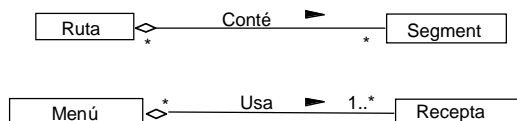
- Els subtipus potencials representen variacions d'un mateix concepte.
- Els subtipus tenen atributs que poden ser factoritzats i expressats als supertipus.
- Els subtipus tenen associacions que poden ser factoritzades i relacionades amb el supertipus.



Agregació

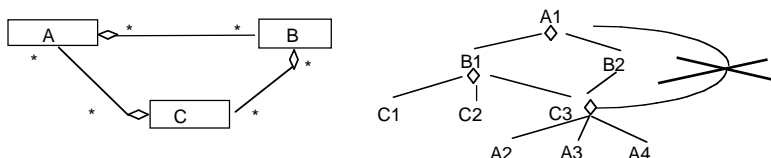
L'agregació és un tipus d'associació usada per modelar relacions "part-tot" entre objectes.

El "tot" s'anomena compostat 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.

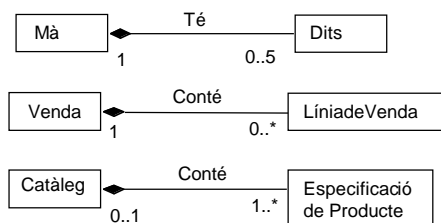


Composició

La composició és un tipus d'agregació per la 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 (no el pot sobreviure)



Agregació i composició - quan mostrar-la

Agregació/Composició

- Existeix un assemblatge tot-part físic o lògic.
- Algunes propietats del tot es propaguen a les parts (com destrucció, moviment, ...)

Composició

- La vida de la part està inclosa en la vida del compostat.
- Existeix una dependència crear-esborrar de la part respecte del compostat.

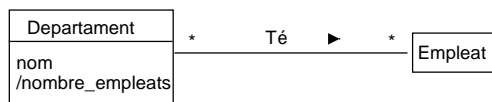
Informació derivada

Un element (atribut o associació) és **derivat** si es pot calcular a partir d'altres elements.

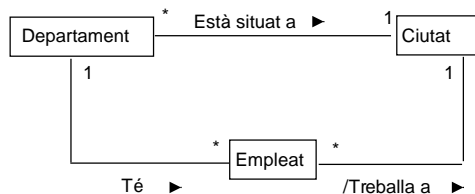
S'inclou quan millora la **claredat** del model conceptual

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

Atribut derivat



Associació derivada

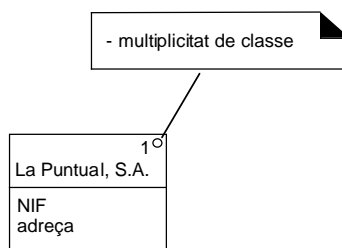


Multiplicitat de classe

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 sobre associacions

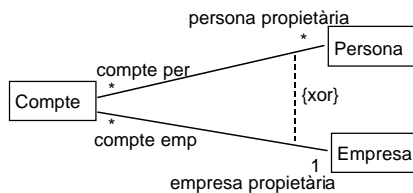
A part de la multiplicitat, és possible expressar altres restriccions sobre les associacions:

- Xor
- Subset

Xor

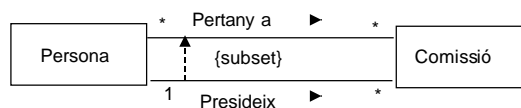
Uneix diverses associacions lligades a una mateixa classe bàsica

Una instància de la classe bàsica participa exactament en una de les associacions unides per "xor".



Subset

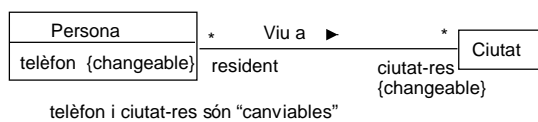
Indica que una associació és un subconjunt d'una altra associació



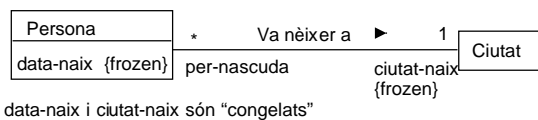
Canviabilitat

La canviabilitat indica si els valors d'un atribut o l'extrem d'una associació poden canviar o no

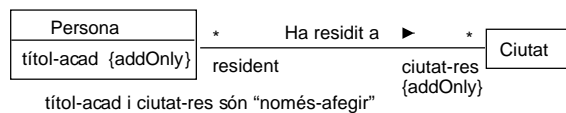
Canviable (changeable) - opció per defecte -



Congelat (frozen)

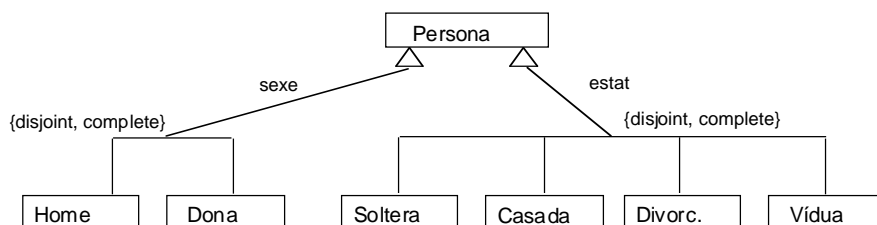


Només-afegir (addOnly)



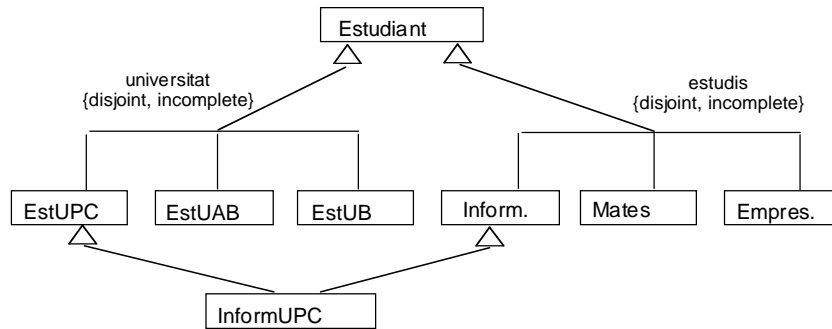
Classificació múltiple

Variant de generalització/especialització en la qual una superclasse pot tenir diverses jerarquies d'especialització en funció de diferents discriminadors



Herència múltiple

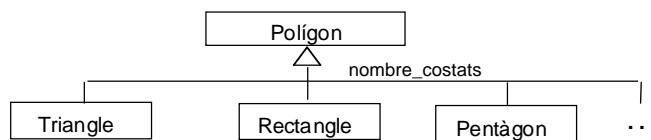
Variant de generalització/especialització en la qual una subclasse té més d'una superclasse



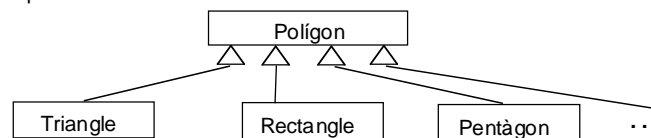
Només es pot utilitzar si no hi ha conflictes d'herència

Equivalències notacionals (1)

En UML:



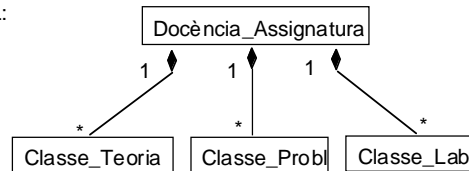
és equivalent a:



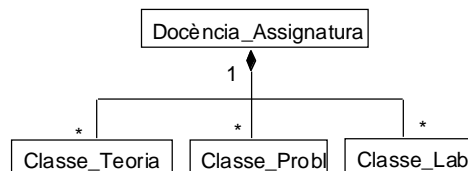
Problema?

Equivalències notacionals (2)

En UML:

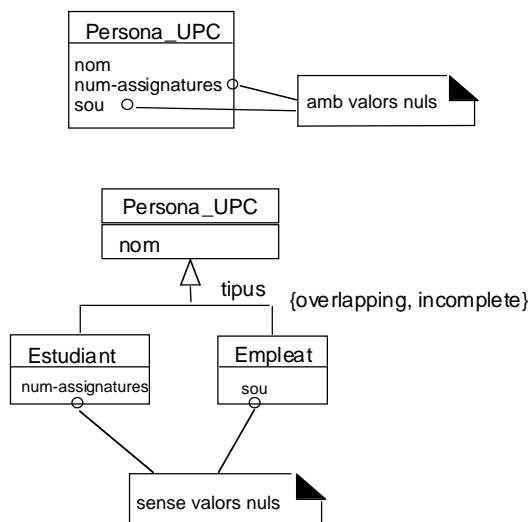


és equivalent a:



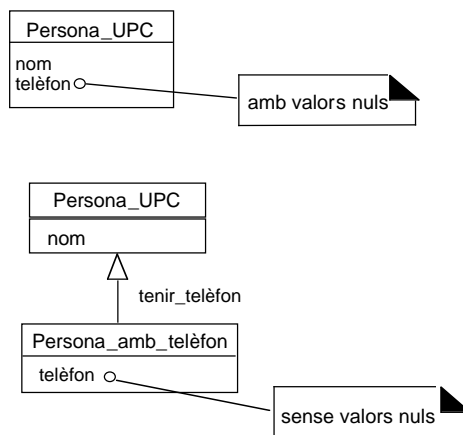
Exemples

Quina solució és millor?

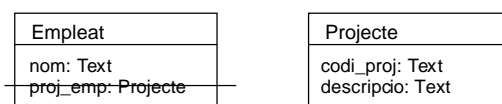


Exemples

Quina solució és millor?



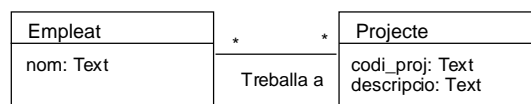
Exemples



Un **atribut** no pot prendre **valors d'una de les classes** del model conceptual



Aquest cas és una **associació**



Bibliografia

- *The Unified Modeling Language User Guide*
G. Booch, J. Rumbaugh, I. Jacobson
Addison-Wesley, 1999

En castellà:

El Lenguaje Unificado de Modelado
G. Booch, J. Rumbaugh, I. Jacobson
Addison-Wesley, 1999

- *The Unified Modeling Language Reference Manual*
J. Rumbaugh, I. Jacobson, G.Booch
Addison-Wesley, 1999
- *Applying UML and Patterns*
An Introduction to Object-Oriented Analysis and Design
C. Larman
Prentice-Hall 1998
Cap. 9, 10, 11, 28, 30

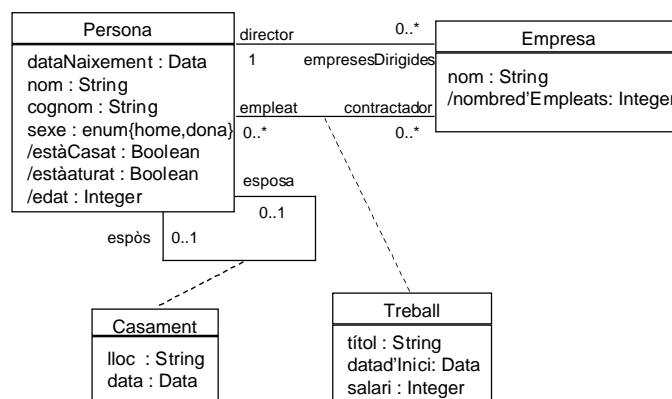
Object Constraint Language (OCL)

- Per què serveix?
- Propietats del Model Conceptual
- Col·leccions: conjunts, bosses i seqüències
- Navegació per classes associatives
- Generalització / Especialització
- Com especificar en OCL

Per què serveix OCL?

- Els models gràfics no són suficients per a una especificació precisa i no ambigua
- L'OCL:
 - és un llenguatge formal
 - permet definir expressions (no té efectes laterals)
 - no és un llenguatge de programació
- S'usa per:
 - especificar invariants (restriccions) del Model Conceptual
 - com a llenguatge de navegació
 - (especificar restriccions en les operacions)

Exemple

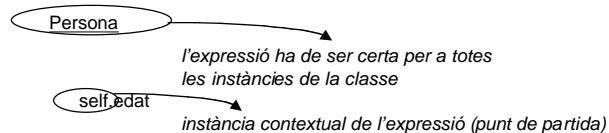


Propietats dels objectes

- Una expressió OCL fa referència a les propietats dels objectes especificats al Model Conceptual
- Una propietat pot fer referència a:
 - atributs d'una classe d'objectes*
 - navegació a través de les associacions*

Propietats dels atributs d'una classe d'objectes:

- Propietat: `edat` d'una persona -- `enter`



- Restricció de la propietat: "l'edat de les persones ha de ser superior o igual a zero"

`Persona`
`self.edat >= 0`

o, alternativament: `p:Persona`
`p.edat >= 0`

Propietats dels objectes (II)

Propietats d'una navegació a través d'associacions:

Partint d'un objecte concret, podem navegar a través de les associacions del Model Conceptual per referir-nos a d'altres objectes i a les seves propietats

- Com es navega? **Objecte.nom-de-rol**



Resultat: conjunt d'objectes de l'altre extrem de nom-de-rol

- si no hi ha nom de rol especificat, es pot usar el nom de la classe d'objectes de l'altre extrem de l'associació (amb minúscules)

- Exemples de navegació:

```

Empresa
self.director      -- director de l'empresa -- Persona
self.director.nom  -- nom del director -- String
self.empleat      -- empleats de l'empresa -- set(Persona)
self.empleat.espòs -- esposos dels empleats -- set(Persona)
  
```

Col·leccions: conjunts, bosses i seqüències

Una *col·lecció d'elements* pot ser del tipus:

- **Conjunt:** cada element ocorre una única vegada a la col·lecció
- **Bossa (multiconjunt):** la col·lecció pot contenir elements repetits
- **Seqüència:** bossa on els elements estan ordenats

Exemple:

- “nombre de treballadors que treballen per a una persona”

```

Persona
num-treb = self.empresesDirigides.empleat -> size
  
```

Incorrecte: el resultat és una bossa i pot contenir repetits.

```

Persona
num-treb = self.empresesDirigides.empleat -> asSet -> size
  
```

Regles de navegació:

- si la multiplicitat de l'associació és 1 el resultat és un objecte o un conjunt d'un únic objecte.
- si la multiplicitat de l'associació és >1 el resultat és una bossa, encara que, de vegades, pot ser un conjunt

Operacions bàsiques sobre colleccions (I)

Select: especifica un subconjunt de la col·lecció

- “persones majors de 50 anys que treballen a una empresa”

Empresa

self.empleat -> select(edat>50)

Empresa

self.empleat -> select(p | p.edat>50)

Empresa

self.empleat -> select(p:Persona | p.edat>50)

Collect: especifica una col·lecció que es deriva d'una altra, però que conté objectes diferents

- “edats (amb repetits) dels empleats d'una empresa”

Empresa

self.empleat -> collect(dataNaixement)

versió simplificada: *self.empleat.dataNaixement*

Operacions bàsiques sobre colleccions (II)

forAll: expressió que han de satisfer tots els elements

- “tots els empleats de l'empresa s'anomenen Jack”

Empresa

self.empleat -> forAll(nom='Jack')

Exists: condició que satisfà almenys un element

- “com a mínim un empleat de l'empresa s'ha de dir Jack”

Empresa

self.empleat -> exists(nom='Jack')

Combinació de propietats

- Les propietats es poden combinar per formar expressions més complexes

- “Les persones casades han de ser majors d’edat”

Persona

self.esposa -> notEmpty implies self.esposa.edat >= 18

and self.espòs -> notEmpty implies self.espòs.edat >= 18

- “Una empresa té com a màxim 50 empleats”

Empresa

self.empleat -> size <= 50

- “Una persona no pot tenir alhora un espòs i una esposa”

Persona

not ((self.esposa -> size=1) and (self.espòs -> size=1))

- “definició de l’atribut derivat nombred’Empleats”

Empresa

nombred’Empleats = (self.empleat -> size)

- “definició de l’atribut derivat estàaturat”

Persona

estàaturat = if self.contractador-> isEmpty then true else false

Navegació per classes associatives

Navegació a una classe associativa:

Es fa servir el nom de la classe associativa (amb minúscula)

- “els sous de les persones que treballen a la UPC han de ser més alts que 1000”

Persona

(self.contractador -> select(nom='UPC')).treball)

-> forAll (t | t.salari > 1000)

Navegació des d’una classe associativa:

S’usa el nom de rol de l’extrem cap on es vol navegar

Si no s’ha especificat nom de rol, s’usa el nom de la classe.

- “les persones que treballen no poden estar aturades”

Treball

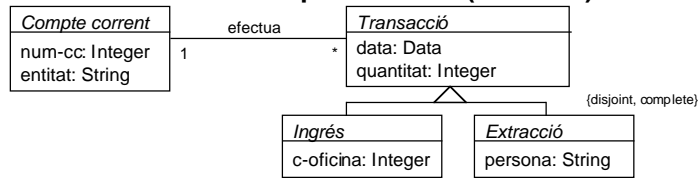
self.empleat.estàaturat = false

- “un casament ha de ser entre una dona i un home”

Casament

self.esposa.sexe = #dona and self.espòs.sexe = #home

Generalització - Especialització (herència)



Avantatges:

- Les propietats de les subclasse es poden ignorar
Compte corrent
`self.transacció -> select(data.isafter(5-4-1998))` -- transaccions d'un compte posteriors al 5-4-1998
- Accés directe a les propietats de la superclasse
Ingrés
`self.compte-corrent.num-cc` -- número de compte d'un ingrés

Aspectes addicionals:

- Selecció d'objectes que pertanyen a la subclasse
Compte corrent
`self.transacció -> select(oclType=Ingrés)` -- ingressos d'un compte
- Accés a les propietats definides al nivell de subclasse
Compte corrent
`self.transacció.oclType(Extracció).persona -> asSet`
-- persones que han tret diners d'un compte corrent

Com especificar en O.C.L.

- Una expressió O.C.L s'especifica sempre començant en una classe d'objectes determinada: *instància contextual*
- Una *expressió* es pot especificar de diverses maneres, segons la instància contextual de partida:

Persona

`self.esposa -> notEmpty implies self.esposa.edat >= 18 and`
`self.espòs -> notEmpty implies self.espòs.edat >= 18`

Casament

`self.esposa.edat >= 18 and self.espòs.edat >= 18`

- Indicacions per escollir la instància contextual
 - si la restricció restringeix el valor de l'atribut d'una classe, aquesta és la classe candidata
 - si la restricció restringeix el valor dels atributs de més d'una classe, qualsevol d'aquestes n'és la candidata
 - qualsevol restricció hauria de navegar a través del menor nombre possible d'associacions

Com especificar en OCL: exemples

- “Totes les persones han de ser majors d'edat”

Persona

self.edat >= 18 -- és preferible a

Empresa

self.empleat -> forAll (age >= 18)

- “Els dos membres d'un matrimoni no poden treballar a la mateixa empresa”

Empresa

self.empleat.esposa -> intersection(self.empleat) -> isEmpty -- és preferible a

Persona

self.espos.contractador -> intersection(self.contractador) -> isEmpty

and

self.esposa.contractador -> intersection(self.contractador) -> isEmpty

- “Ningú no pot ser director i empleat d'una empresa”

Empresa

not(self.empleat -> includes(self.director))

Persona

not(self.empresesDirigides.empleat -> includes(self))

-- quina és preferible en aquest cas?

Operacions estàndard de tipus booleà

Operació	Notació	Resultat
or	a or b	booleà
and	a and b	booleà
or exclusiu	a xor b	booleà
negació	not a	booleà
igualtat	a = b	booleà
desigualtat	a <> b	booleà
implicació	a implies b	booleà
if-then-else	if a then b else b'	tipus de b o b'

Operacions estàndard de tipus string

Operació	Notació	Resultat
concatenació	<code>string.concat(string)</code>	string
tamany	<code>string.size</code>	integer
substring	<code>string.substring(int, int)</code>	string
igualtat	<code>string 1 = string2</code>	booleà
desigualtat	<code>string 1 <> string2</code>	booleà

Operacions estàndard de tipus col.lecció

Operació	Resultat
<code>size</code>	nombre d'elements de la col.lecció
<code>count(object)</code>	nombre d'ocurrències de l'objecte
<code>includes(object)</code>	cert si l'objecte pertany a la col.lecció
<code>includesAll(collection)</code>	cert si els elements del paràmetre <i>collection</i> són a la col.lecció actual
<code>isEmpty</code>	cert si la col.lecció és buida
<code>notEmpty</code>	cert si la col.lecció no és buida
<code>sum()</code>	suma de tots els elements (els elements s'han de poder sumar)
<code>exists(expression)</code>	<i>expression</i> és cert per algun element?
<code>forAll(expression)</code>	<i>expression</i> és cert per tots els elements?
<code>select(expression)</code>	selecciona el subconjunt d'elements per als quals <i>expression</i> és cert
<code>reject(expression)</code>	elimina el subconjunt d'elements per als quals <i>expression</i> és cert
<code>union(collection)</code>	resultat d'unir les dues col.leccions
<code>intersection(collection)</code>	resultat de la intersecció

Bibliografia

- Unified Modeling Language
Object Constraint Language Specification, v. 1.1.
Setembre 1997.
- J. Warmer; A. Kleppe
The Object Constraint Language: precise modeling with UML
Addison-Wesley, 1999.

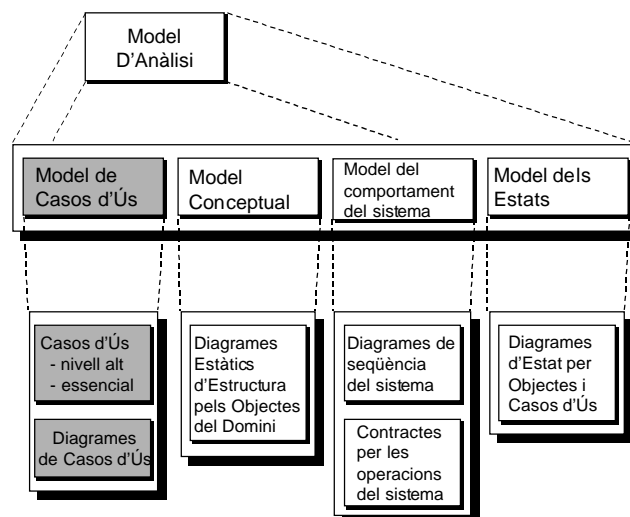
Pàgines web amb informació d'OCL:

- <http://www.software.ibm.com/ad/ocl>
- <http://www.rational.com>
- <http://www.omg.org>

Model de Casos d'Ús en UML

- Propòsit
- Casos d'ús
- Diagrama de Casos d'ús
- Especificació de Casos d'ús
- Estructuració de Casos d'ús
- Identificació de Casos d'ús

Model d'Anàlisi (especificació)



Determinació de requeriments d'un sistema software:

- Identificar i categoritzar les funcions del sistema (requeriments funcionals).
- Identificar i categoritzar els atributs del sistema (requeriments no funcionals).
- Relacionar els requeriments no funcionals amb els funcionals.

Especificació dels requeriments d'un sistema software

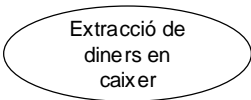
- Comprensió dels requeriments.
- Organitzar els requeriments segons les funcions del sistema.
- Delimitar la frontera del sistema.



Model de Casos d'Ús: Quines son les funcions del sistema **PER CADA ACTOR?**
Èmfasi: USOS del sistema, valor afegit per cada actor

Casos d'Ús

Cas d'Ús: Document que descriu una seqüència d'esdeveniments que realitza un actor (agent extern) que usa el sistema per dur a terme un procés que té algun valor per a ell [Jacobson 92].



Extracció de
diners en
caixer

Actor: Entitat externa al sistema que participa en la història del Cas d'Ús.
Pot ser una persona, un conjunt de persones, un sistema hardware, un sistema software o un rellotge.

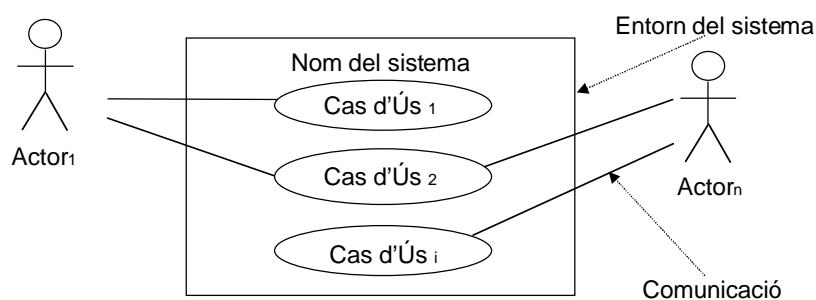
Iniciador: Genera l'estímul que provoca l'execució del procés (únic).

Participant: Intervé en el procés.



Diagrama de Casos d'Ús

Mostra conjuntament els diferents casos d'ús d'un sistema software, els actors i les relacions entre actors i casos d'ús.



Especificació de Casos d'Ús

D'alt nivell: Descripció breu de les accions del cas d'ús.

Cas d'ús: Nom del cas d'ús.

Actors: Llista d'actors, iniciador.

Propòsit: Objectiu del cas d'ús.

Resum: Descripció breu de les activitats que s'han de dur a terme.

Tipus: 1 - primari, secundari, opcional.

2 - real o essencial.

Expandida: Descripció detallada de les accions i els requeriments.

Afegeix a l'especificació d'alt nivell:

Referències creuades: Requeriments a què fa referència.

Curs típic d'esdeveniments: Descripció detallada de la interacció (conversa) entre els actors i el sistema.

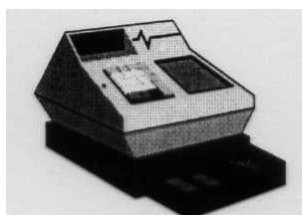
Descripció dels esdeveniments pas a pas.

Cursos alternatius: Descriu excepcions al curs típic.

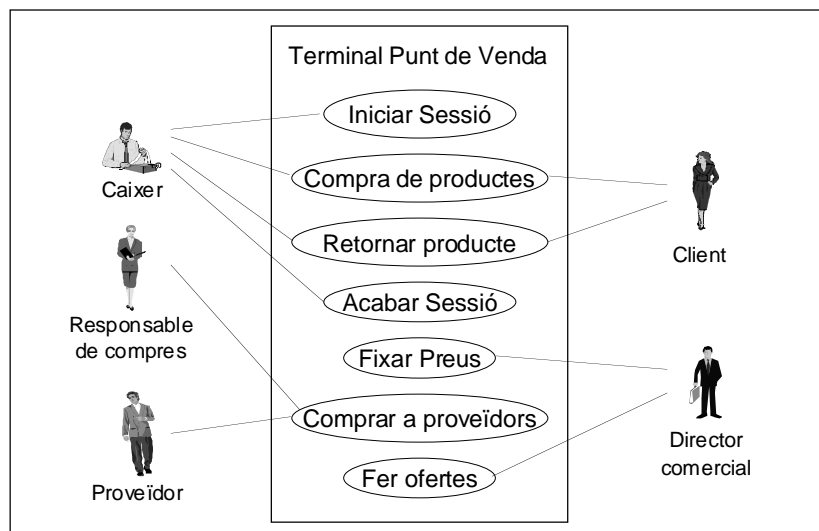
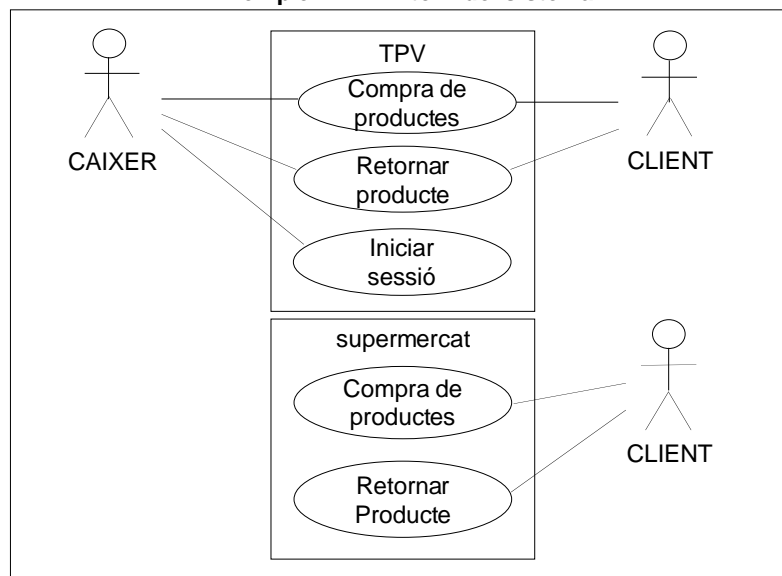
Exemple: Terminal de Punt de Venda

Un terminal de punt de venda (TPV) és un sistema computeritzat usat per enregistrar les vendes i gestionar pagaments. S'usa, principalment en supermercats i grans magatzems. Inclou components hardware (com l'ordinador i l'scanner del codi de barres) i software per executar el sistema.

Se'ns demana que especifiquem el software d'aquest sistema.



Ref #	Funció	Exemple TPV: Funcions bàsiques	Categoria
R1.1	Enregistrar la venda actual - els productes comprats.		evident
R1.2	Calcular el total de la renda actual, incloent impostos i càlcul de "punts de client".		evident
R1.3	Capturar la informació dels productes comprats d'un codi de barres, usant un scanner o bé a partir de l'entrada manual del codi de barres (Universal Product Code).		evident
R1.4	Descomptar les quantitats venudes de l'estoc, quan la venda es confirmi.		hidden
R1.5	Guardar informació sobre les vendes realitzades.		hidden
R1.6	El caixer ha d'identificar-se en iniciar una sessió amb un identificador i una clau d'accés.		evident
R1.7	Mostrar la descripció i el preu de cada producte comprat.		hidden
R2.1	Tractar els pagaments en efectiu capturant la quantitat entregada pel client i calculant el canvi.		evident
R2.2	Tractar els pagaments amb tarja de crèdit capturant el número de la tarja des d'un lector de targes o manualment, demanar confirmació del pagament al servei d'autorització de crèdit (extern) amb una connexió via modem.		evident
R2.3	Enregistrar els pagaments amb tarja per tal que puguin ser facturats.		hidden

Example TPV: Diagrama de Casos d'Ús**Exemple TPV: Entorn del sistema**

Exemple TPV: especificació del cas d'ús "compra de productes en efectiu"

Cas d'Ús: Compra de productes en efectiu.

Actors: Client (iniciador), Caixer.

Propòsit: Capturar una venda i el seu pagament en efectiu.

Resum: Un client arriba a la caixa amb productes per comprar.

El caixer enregistra els productes i recull el pagament.

En acabar, el client se'n va amb els productes.

Tipus: Primari i essencial.

Referències creuades: R1.1, R1.2, R1.3, R1.7, R2.1

Curs típic d'esdeveniments**Accions dels Actors**

1. El cas d'Ús comença quan un client arriba a la caixa amb els productes per comprar.
2. El Caixer indica que comença una nova venda
4. El Caixer enregistra l'identificador de cada producte.
Si hi ha més d'una unitat del producte el Caixer pot entrar la quantitat.
6. En acabar l'entrada de productes el caixer ho indica.

Resposta del sistema

3. Dóna la venda per iniciada
5. Determina el preu del producte i afegeix la seva informació al compte.
7. Calcula i mostra el total del compte.

(continua)

Accions dels Actors

8. El Caixer li diu el total al client.
9. El Client entrega una quantitat de diners possiblement més gran que el total del compte.
10. El Caixer indica els diners que ha rebut.
13. El Caixer diposita els diners rebuts a la caixa i extreu el canvi.
El Caixer dóna el canvi i el rebut al client.
14. El Client se'n va amb els productes comprats.

Resposta del sistema

11. Calcula i mostra el canvi al Client.
Imprimeix un rebut.
12. Enregistra la venda que s'acaba de fer.

Cursos Alternatius

- Línia 4: S'entra un identificador de producte invàlid. Indicar error.
- Línia 9: El Client no té prou diners. Cancel·la la venda.

Estructuració de Casos d'ús: seccions

Cas d'Ús: Compra de productes

Curs típic d'esdeveniments

Accions dels Actors

Resposta del sistema

- | | |
|---|--|
| <ol style="list-style-type: none"> 1. El cas d'Ús comença quan un client arriba a la caixa amb els productes per comprar. 2. (passos intermedis exclosos)... 9. El Client escull la forma de pagament: <ol style="list-style-type: none"> a. If en efectiu, see section pagar en efectiu b. If amb tarja see section pagar amb tarja 12. El Caixer dóna el rebut al client. 13. El Client se'n va amb els productes comprats. | <ol style="list-style-type: none"> 10. Enregistra la venda que s'acaba de fer. 11. Imprimeix un rebut. |
|---|--|

Estructuració de Casos d'ús: seccions

Section: Pagar en efectiu.

Curs típic d'esdeveniments

Accions dels actors

Resposta del sistema

- | | |
|--|---|
| <ol style="list-style-type: none"> 1. El Client entrega una quantitat de diners possiblement més gran que el total del compte. 2. El Caixer indica els diners que ha rebut. 4. El Caixer diposita els diners rebuts i extreu el canvi.
El Caixer dóna el canvi al client. | <ol style="list-style-type: none"> 3. Calcula i mostra el canvi al Client. |
|--|---|

Cursos Alternatius

- Línia 4: Efectiu insuficient per tomar el canvi. Demanar canvi al supervisor.

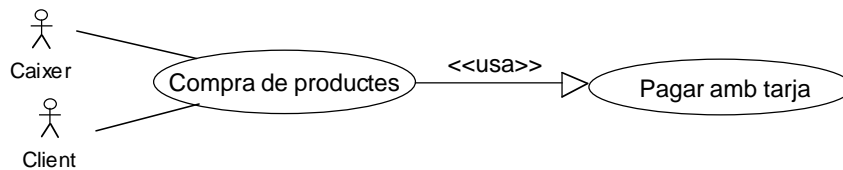
Section: Pagar amb tarja.

Cursos típics i alternatius per l'història del pagament amb tarja.

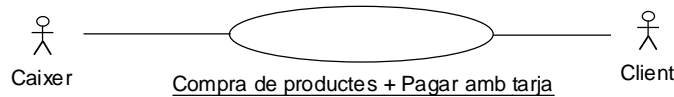
Estructuració de Casos d'ús: Relació Usa

Usa: Relació d'un cas d'ús concret amb un d'abstracte, en la qual la conducta definida pel cas concret usa (empra) la conducta definida en l'abstracte.

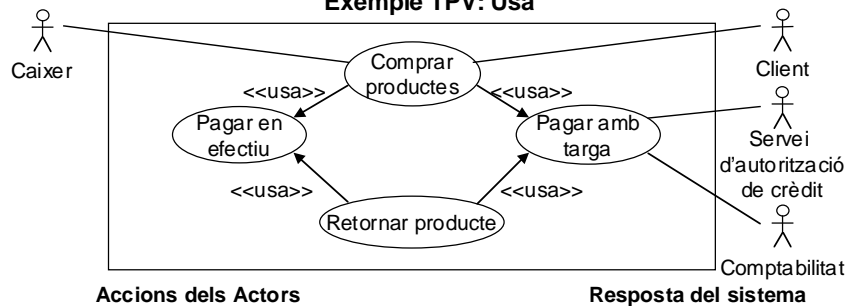
Permet reduir la redundància quan una seqüència d'accions és compartida per diversos casos d'ús



Cas d'ús "Real"



Exemple TPV: Usa



Accions dels Actors

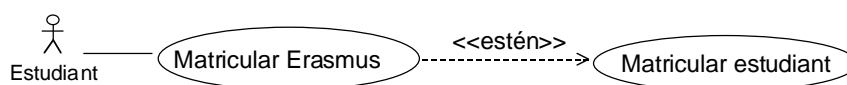
Resposta del sistema

1. El cas d'Ús comença quan un client arriba a la caixa amb els productes per comprar.
2. (Passos intermedis exclosos)...
9. El Client escull el tipus de pagament:
 - a. **If** és en efectiu **initiate**
Pagar en efectiu
 - b. **If** és amb targa **initiate**
Pagar amb targa
10. Enregistra la venda que s'acaba de fer.
11. Imprimeix un rebut.
12. El Caixer dona el rebut al client.
13. El client se'n va amb els productes comprats

Estructuració de Casos d'ús: Relació Estén

Estén: Relació d'un cas d'ús a un altre especificant com la conducta definida pel primer pot ésser inserida en la conducta definida pel segon. També es pot dir que el primer és semblant al segon, però fa una mica més.

Una extensió es comporta com si fóssin accions que s'afegeixen en un punt concret de l'especificació original si es dóna una certa condició



Identificació de casos d'Ús

Mètode basat en els actors

1. Identificar els actors relatius al sistema.
2. Per cada actor, identificar els processos que inicia o en els quals participa.

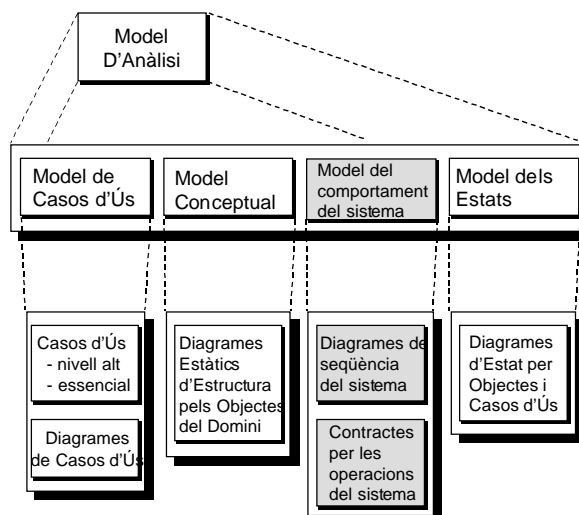
Mètode basat en els esdeveniments

1. Identificar els esdeveniments externs als que el sistema ha de respondre.
2. Relacionar els esdeveniments amb els actors i casos d'ús.

Model del comportament en UML

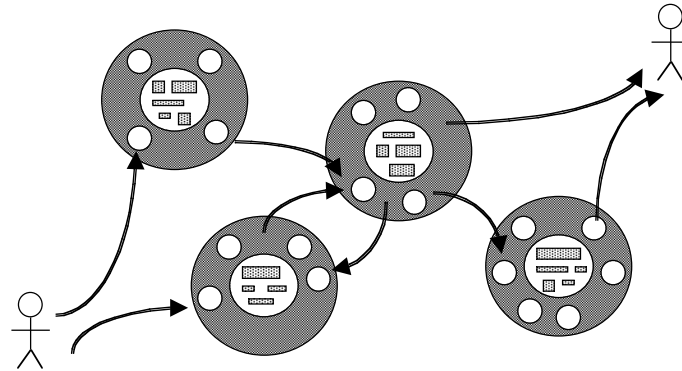
- Introducció
- Diagrames de seqüència del sistema
- Contractes de les operacions del sistema
- Altres consideracions
 - Compartició d'informació entre les operacions d'un diagrama
 - Informació elemental vs informació composta
 - Nombre d'esdeveniments del diagrama de seqüència
 - Redundància entre els models
- Bibliografia

Model d'Anàlisi (Especificació)

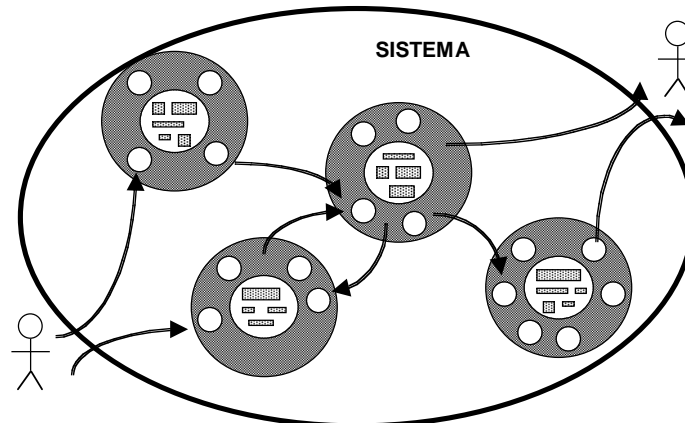


Descripció del comportament en OO

Els objectes es comuniquen mitjançant la invocació d'operacions d'altres objectes



“Especificació” del comportament en OO



- Considerem un tipus especial “**sistema**” que engloba tots els objectes
- L'**especificació del comportament** es fa amb el **model del comportament del “sistema”**

Model del comportament del sistema

- **Diagrames de seqüència del sistema:**
 - Mostren la seqüència d'esdeveniments entre els actors i el sistema.
 - Permeten identificar les operacions del sistema
- **Contractes per les operacions del sistema:**
 - Descriuen l'efecte de les operacions del sistema

Diagrames de seqüència del sistema

- **Objectius:**
 - identificar els esdeveniments i les operacions del sistema
- **Punt de partida:**
 - casos d'ús
 - la descripció dels diagrames de seqüència del sistema és posterior a la descripció dels casos d'ús
- **Casos d'ús:**
 - descriuen com els actors interaccionen amb el sistema software
 - l'actor genera esdeveniments cap al sistema que exigeixen l'execució d'alguna operació com a resposta (durant la interacció)
 - a partir dels casos d'ús podem identificar quins són els esdeveniments que van dels actors cap al sistema

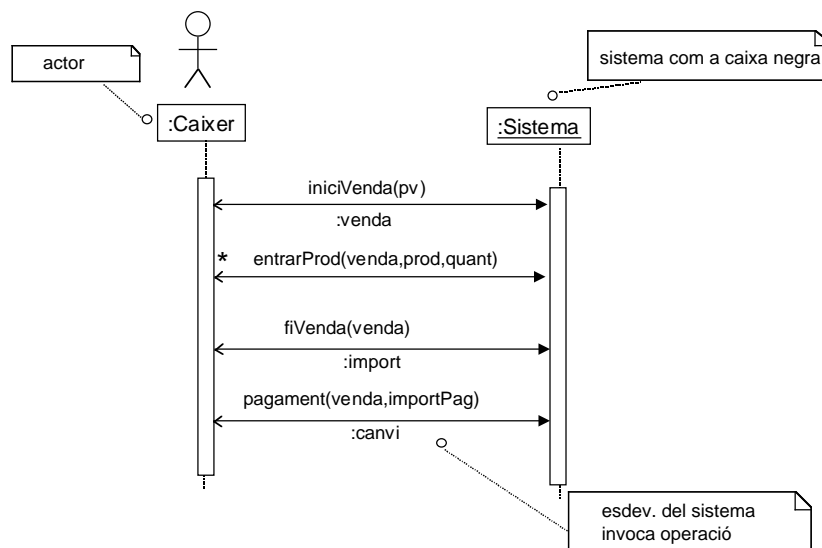


diagrames de seqüència del sistema

Diagrames de seqüència del sistema

- **Mostra**, per a un escenari particular d'un cas d'ús :
 - els esdeveniments generats pels actors externs
 - el seu ordre
 - els esdeveniments interns al sistema (operacions) que resulten de la invocació
- Definirem un diagrama de seqüència per cada **curs rellevant** d'esdeveniments d'un cas d'ús

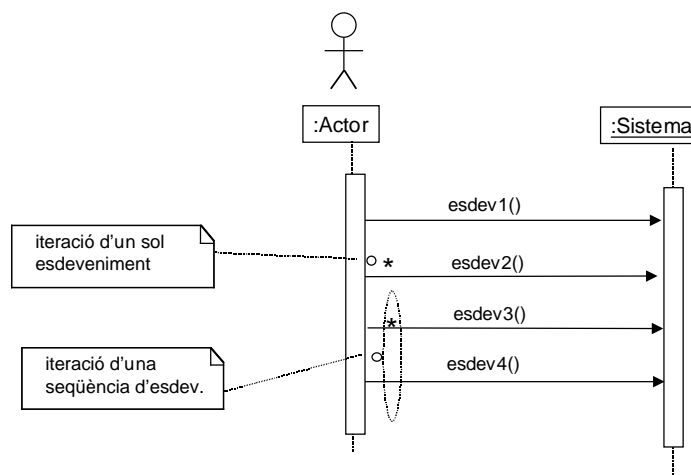
Exemple: comprar producte



Construcció d'un diagrama de seqüència

1. Dibuixar una línia vertical que representa el **sistema**
2. Dibuixar una línia per cada **actor** que interacciona **directament** amb el sistema
3. Del curs d'esdev. del cas d'ús, **identificar** els **esdeveniments** externs generats pels actors. Mostrar-los al diagrama

Representació d'iteracions d'esdeveniments



Esdeveniments i operacions

- **Esdeveniment del sistema:** Esdeveniment extern generat per un actor
- **Operació del sistema:** Operació interna que s'executa com a resposta a la comunicació de l'esdeveniment



La comunicació d'un esdeveniment del sistema provoca l'execució d'una operació del sistema amb el **mateix nom** i els **mateixos paràmetres**

Operacions del sistema

- Les operacions del sistema **s'agrupen** com a operacions del tipus especial "sistema"
- En canvi, les operacions **no s'assignen a objectes** concrets durant l'etapa d'especificació

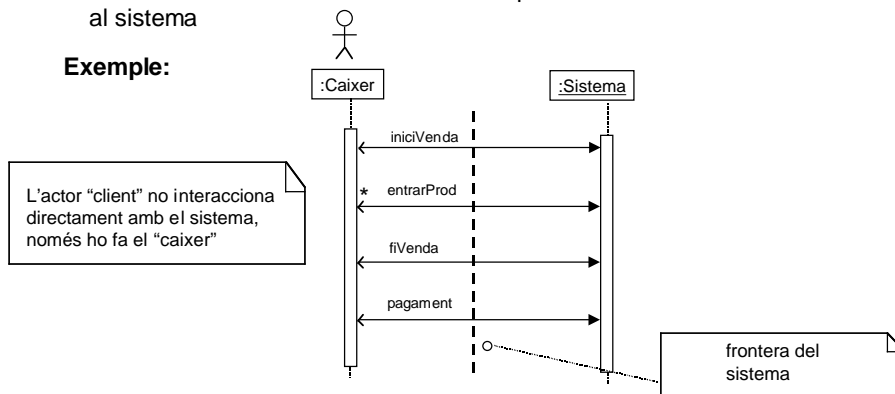
Exemple:

SISTEMA
iniciVenda(pv) :venda entrarProd(venda,prod,quant) fiVenda(venda) :import pagament(venda,importPag): canvi

Esdeveniments i el límit del sistema

- Per identificar els esdeveniments del sistema és necessari haver delimitat clarament la **frontera** del sistema
- Els esdeveniments del sistema són els que estimulen **directament** al sistema

Exemple:



Contractes de les operacions

- **Contracte d'una operació**
 Descriu el comportament del sistema en termes de:
 - quins són els **canvis d'estat** de la base d'informació
 - quines són les **sortides** que el sistema proporciona quan s'invoca l'operació
- El tipus de descripció és **declaratiu**:
 - l'èmfasi es posa en el **què** farà l'operació més que en el **com** ho farà
- Els contractes de les operacions **inclouen** primordialment:
 - **precondicions** i **postcondicions** que descriuen els canvis d'estat
 - **sortides**

Contractes de les operacions: components

Name: nom i arguments de l'operació (*signatura de l'operació*)

Responsibilities:

Descripció informal del propòsit de l'operació

Exceptions:

Descripció de la reacció del sistema a situacions excepcionals

Preconditions:

Assumpcions sobre l'estat del sistema abans de la invocació de l'operació

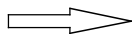
Postconditions:

Canvis d'estat que s'han produït:

- altes/baixes d'instàncies de classes d'objectes
- altes/baixes d'instàncies d'associacions
- modificació d'atributs
- generalització d'un objecte
- especialització d'un objecte
- canvi de subclasse d'un objecte

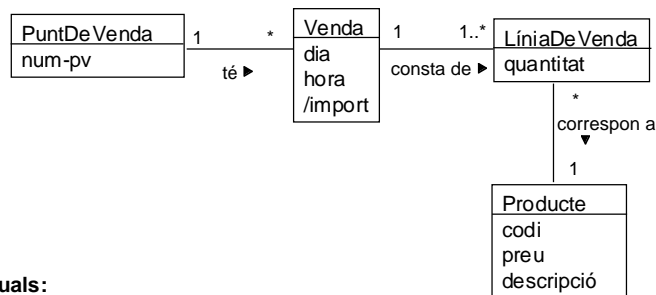
Sortida:

Descripció de la sortida que proporciona l'operació en pseudo-OCL



Observeu el lligam que existeix entre els **contractes** de les operacions i l'**esquema conceptual**

Exemple: esquema conceptual de partida



RI textuais:

- 1- La clau externa de PuntDeVenda és num-pv
- 2- La clau externa de Producte és codi
- 3- Un punt de venda no pot tenir més d'una venda amb el mateix dia i hora

Exemple: operació iniciVenda

Name: iniciVenda(pv) :venda

Responsibilities:

Iniciar l'enregistrament d'una venda

Exceptions:

Si no existeix cap PuntDeVenda amb num-pv=pv, indicar error

Preconditions:

Existeix un PuntDeVenda amb num-pv=pv

Postconditions:

- alta d'una instància V de Venda amb el dia i l'hora actuals
- alta d'una instància de l'associació 'té' que associa la venda V i la instància de PuntDeVenda amb num-pv=pv

Sortida: V

Exemple: operació entrarProd

Name: entrarProd(venda,prod,quant)

Responsibilities:

Enregistrar un línia d'una venda

Exceptions:

Si no existeix cap Producte amb codi=prod, indicar error

Preconditions:

Existeix un Producte amb codi=prod

Postconditions:

- alta d'una instància de LíniaDeVenda L amb quantitat=quant
- alta d'una instància de l'associació 'consta de' que associa L i venda
- alta d'una instància de l'associació 'correspon a' que associa L i el producte amb codi=prod

Sortida:

Exemple: operació fiVenda

Name: fiVenda(venda) :import

Responsibilities:

Finalitzar l'enregistrament d'una venda i mostrar l'import a pagar

Exceptions:

Preconditions:

Postconditions:

Sortida: import = venda.import

Exemple: operació pagament

Name: pagament(venda,importPag): canvi

Responsibilities:

Mostrar el canvi a retornar

Exceptions:

Si importPag < venda.import indicar error

Preconditions:

importPag ≥ venda.import

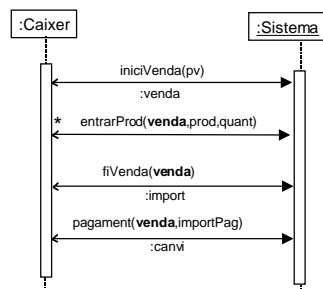
Postconditions:

Sortida: canvi = venda.import - importPag

Compartició d'informació entre les operacions d'un diagrama

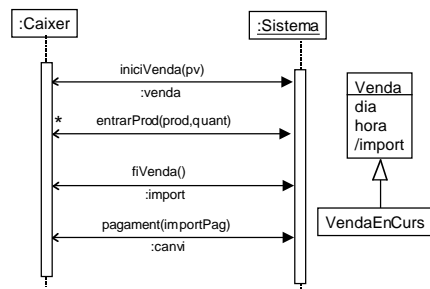
- UML no precisa de quina manera les operacions d'un diagrama de seqüència poden compartir informació
- Dues possibles solucions són:

Mitjançant arguments addicionals de les operacions



Restricció implícita: El valor de l'argument 'venda' és el mateix a tots els esdeveniments del diagrama

Mitjançant informació addicional a l'esquema conceptual



Informació Elemental vs Informació Composta

- La informació tractada per una operació s'expressa tant a nivell dels paràmetres com de la sortida que apareixen a la signatura de l'operació
- Hi ha dos tipus d'informació:
 - Informació Elemental:** conté un únic element d'informació indivisible
 - Propietat
 - Classe d'objectes predefinida: Integer, Real, ...
 - Informació Composta:** és una composició d'informacions elementals (i, per tant, cal especificar com es defineix la composició)

Exemple: ResumVendes(num-pv) que retorna un llistat de vendes amb la informació:

- Per cada Producte p venut en aquell PuntDeVenda num-pv mostrar
- el codi del producte
 - la quantitat total venuda de p a num-pv

ResumVendes (num-pv:Integer): ???

Informació Composta - Definició

Problema: com s'especifica el contingut d'una informació composta?

- En l'actualitat, UML no proposa cap solució
- Utilitzarem una adaptació de la definició de fluxes de dades que es fa en l'anàlisi estructurada (Yourdon, 1993)

- Mecanismes bàsics de definició d'informació composta

- Inclusió: $i1 = i2 + i3 + i4$
- Selecció: $i1 = [i2 \mid i3 \mid i4]$
- Repetició: $i1 = \{i2 + i3 + i4\}$
- Opcionalitat: $i1 = i2 + (i3 + i4)$

Exemple:

LlistatVendes = num-pv + {codi-prod + quantitat}

num-pv = Integer; codi-prod = Integer; quantitat = Integer

ResumVendes(num-pv:Integer) : LlistatVendes

Informació composta - Contractes de les operacions

- Les operacions necessitaran mecanismes per manipular (accedir, construir, etc.) la informació composta que apareix a la seva signatura
- Es necessita estendre OCL per poder manipular informació composta

Exemple: contracte de l'operació ResumVendes (num-pv): LlistatVendes

Nom: ResumVendes (num-pv): LlistatVendes

Responsabilitats: emetre el resum de vendes demanat

Tipus: sistema

Excepcions:-

Preconditions: -

Postconditions: -

Sortida:

Mostrar (num-pv)

Per cada producte p resultant de

(Producte.allInstances ->

select (p | p.LiniaDeVenda.Venda.PuntDeVenda.num-pv->includes(num-pv))

Per

Qt = (p.LíniesdeVenda ->

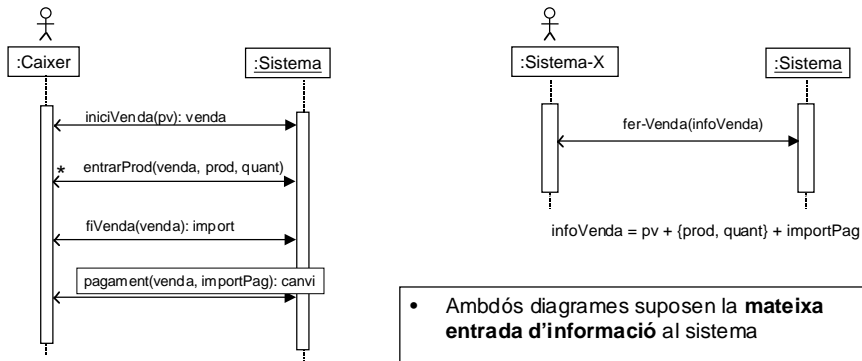
(select (lv | lv.Venda.PuntdeVenda.num-pv = num-pv).quantitat) -> Sum)

Mostrar (p.codi-prod)

Mostrar (Qt)

Diagrama de seqüència: quants esdeveniments?

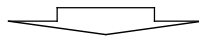
- El nombre d'esdeveniments d'un diagrama de seqüència depèn de **com es produeix la interacció** entre els actors i el sistema software.



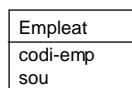
- Ambdós diagrames suposen la **mateixa entrada d'informació** al sistema
- Els dos diagrames de seqüència poden ser **correctes, segons les circumstàncies**

Redundància - Exemple

- L'esquema Conceptual conté **restriccions d'integritat** (gràfiques i textuais)
- Els contractes de les operacions tenen **precondicions**, que són requeriments del contingut de l'esquema conceptual per poder executar una transacció



Cal que les precondicions incloguin la comprovació de les restriccions del M.Conceptual?



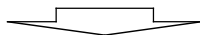
R.I. textual: dos empleats
no poden tenir el mateix codi

Nom: AltaEmpleat (codi-emp, sou)
Responsabilitats: donar d'alta l'empleat
Excepcions: -
Preconditions: no existeix Empleat amb codi-emp
Postconditions:
 creació d'un nou objecte Empleat amb
 codi-emp i sou
Sortida: -

Cal posar aquesta precondició ???

Redundància - Definició

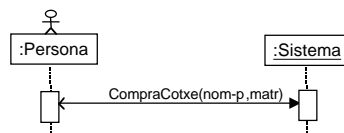
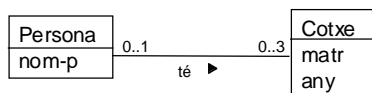
- Una especificació és redundant si un **mateix aspecte** del sistema software **està especificat diverses vegades**.
- La redundància **dificulta la modificabilitat** de l'especificació perquè si varia aquell aspecte cal modificar tots els models que hi fan referència.



L'especificació no hauria de ser redundant !!!

- Redundàncies possibles:
 - Entre l'Esquema Conceptual i els Contractes
 - Entre els Diagrames de Seqüència i els Contractes
 - ...

Redundància - Esq. Conceptual i Contractes (I)



- Significat habitual:

Nom: CompraCotxe (nom-p, matr)

Responsabilitats: compra d'un cotxe

Excepcions:

... (les que es dedueixen de les prec.)

Preconditions:

- existeix Persona p amb nom-p

- existeix Cotxe c amb matr

~~p no té c~~

~~p no té ja 3 cotxes~~

Postconditions:

- creació d'una associació té entre p i c

Sortida: -

- Significat alternatiu:

Nom: CompraCotxe (nom-p, matr)

Responsabilitats: compra d'un cotxe

Excepcions:

... (les que es dedueixen de les prec.)

Preconditions:

- existeix Persona p amb nom-p

- existeix Cotxe c amb matr

~~p no té c~~

Postconditions:

- Si p té ja 3 cotxes llavors
eliminar l'associació entre p i el
cotxe c' de més antiguitat

- creació d'una associació té entre p i c

Sortida: -

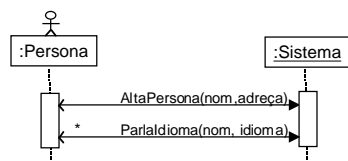
Redundància - Esq. Conceptual i Contractes (II)

- *Com s'han d'interpretar els contractes en relació al Model Conceptual?*
 - Precondicions:
 - **què** ha de contenir el Model Conceptual per **intentar executar una operació**
 - Restriccions del Model Conceptual:
 - Estan garantides després de l'execució de totes les operacions que participen en un cas d'ús
 - Es rebutgen totes les operacions d'un cas d'ús si la seva execució viola (globalment) alguna restricció d'integritat del Model Conceptual

Redundància - Esq. Conceptual i Contractes (III)



Cas d'ús: Afegir-Persona-1



Nom: AltaPersona(nom,adreça)

...

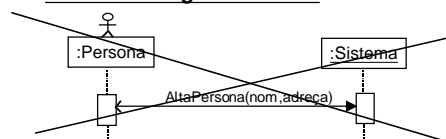
Preconditions:

Postconditions:

- creació d'un objecte Persona amb el nom i l'adreça especificats

Sortida:

Cas d'ús: Afegir-Persona-2



No permet afegir mai una persona !

Nom: ParlaIdioma(nom,idioma)

...

Preconditions:

- existeix idioma amb nom=idioma

Postconditions:

- creació d'una associació parla entre la persona amb nom=nom i l'idioma

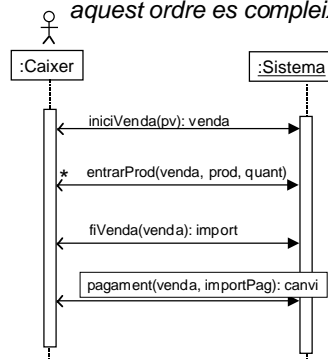
Sortida:

Redundància - Diagrames de Seqüència i Contractes

Els diagrames de seqüència defineixen un **ordre d'invocació** de les operacions



Les operacions **no han d'incorporar informació** per garantir que aquest ordre es compleixi



Nom: Pagament (venda, importPag): canvi

...

Preconditions:

- existeix venda "venda"
- la venda "venda" ha finalitzat

Postconditions:

...

Sortida:

són redundants

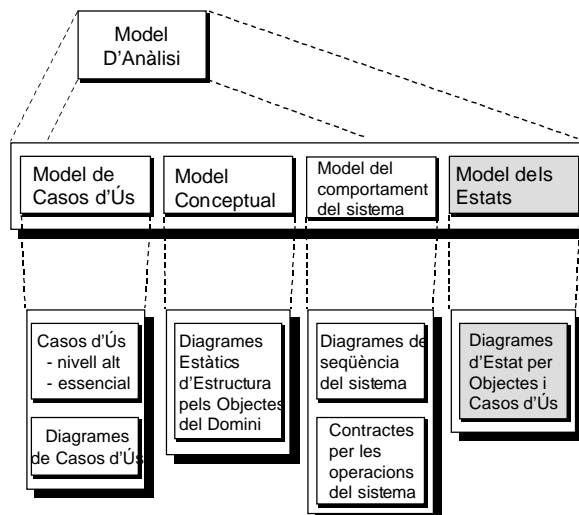
Bibliografia

- C. Larman
Applying UML and Patterns.
An Introduction to Object-Oriented Analysis and Design.
Prentice Hall, 1998. (Cap. 13, 14)
- J. Rumbaugh, I. Jacobson, G. Booch
The Unified Modeling Language Reference Manual
Addison-Wesley, 1999.
- G. Booch, J. Rumbaugh, I. Jacobson
The Unified Modeling Language User Guide
Addison-Wesley, 1999.

Model dels Estats en UML

- Introducció
- Ús dels diagrames d'estat
- Exemples
- Accions i condicions d'una transició
- Estats imbricats
- Bibliografia

Model d'Anàlisi (especificació)

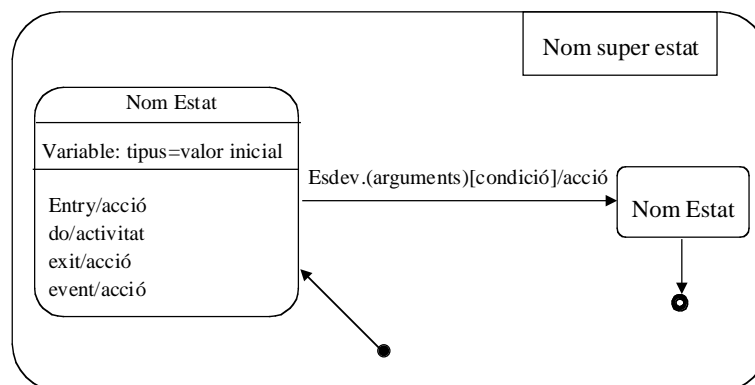


Model dels Estats

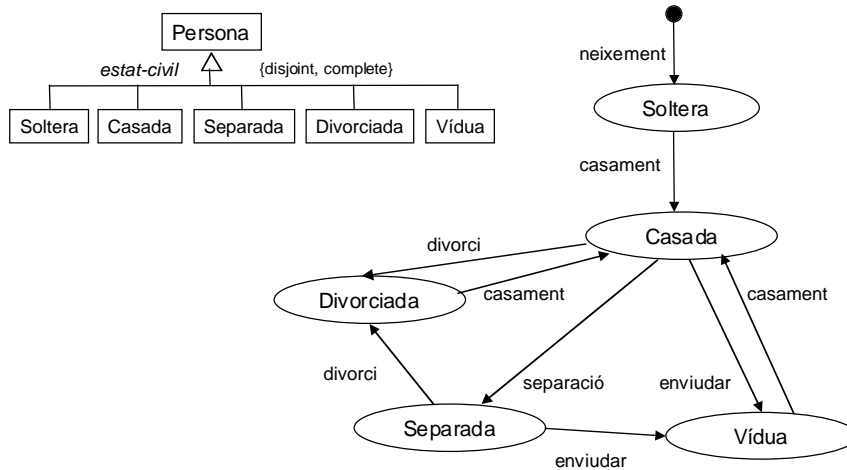
- **Objectius:**
 - crear diagrames d'estat per objectes i casos d'ús
- **Esdeveniments, estats i transicions:**
 - Esdeveniment:
 - tot allò que requereix una resposta del sistema software
 - Estat:
 - condició d'un objecte o d'un cas d'ús en un moment del temps
 - Transició:
 - canvi d'estat com a conseqüència d'un esdeveniment

Diagrama d'estats

Un diagrama d'estats mostra la seqüència d'estats que passa un objecte (o una interacció) durant la seva vida en resposta als estímuls rebuts, juntament amb les seves respostes.



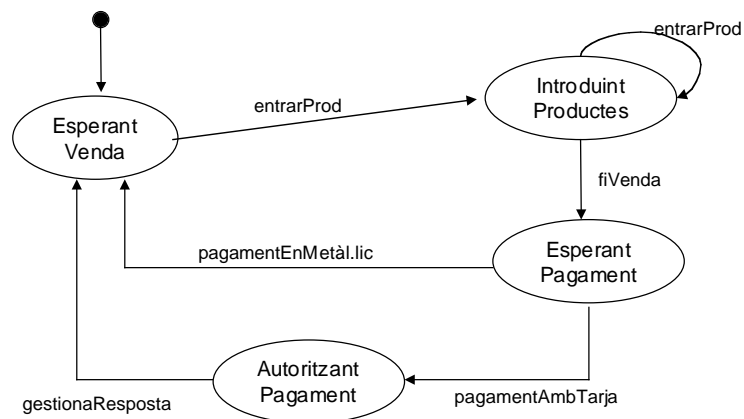
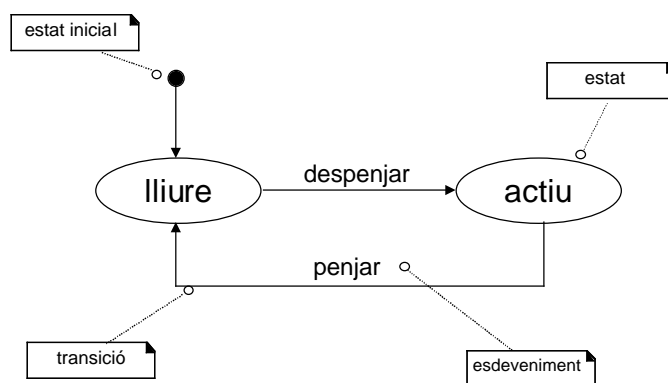
Canvi d'estat civil d'una persona



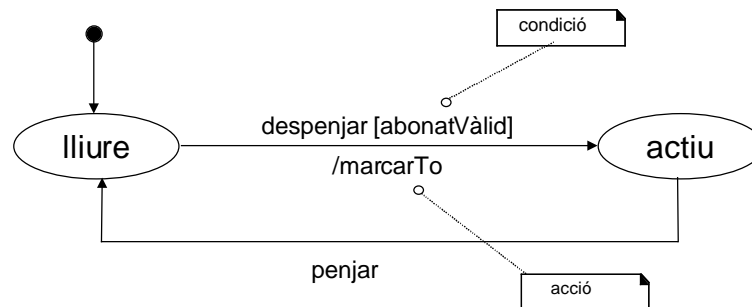
Ús dels diagrames d'estat

- Un diagrama d'estats es pot especificar per a una:
 - **Classe d'objectes:**
 - per descriure perquè els objectes canvien de subclasse
 - les subclasses d'un diagrama d'estats no tenen perquè aparèixer explícitament a l'esquema conceptual
 - per descriure classes d'objectes amb important "comportament dinàmic"
 - **Cas d'ús:**
 - per descriure la seqüència legal en la que els esdeveniments es poden produir al món real

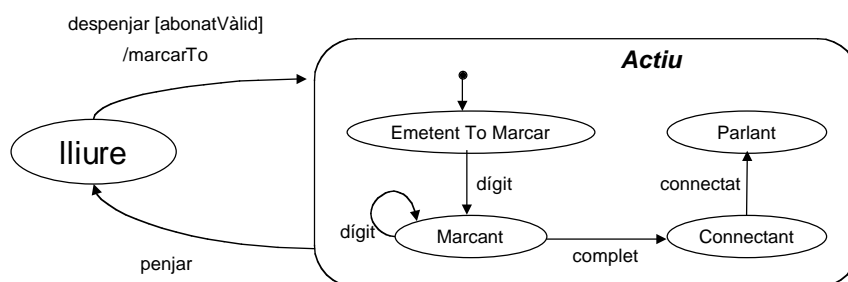
p.ex. en una compra de producte no es pot fer el pagament fins que no s'hagi tancat la venda.

Diagrama d'estats del cas d'us "comprar productes"**Diagrama d'estats d'un telèfon**

Accions i condicions d'una transició



Estats imbricats



Bibliografia

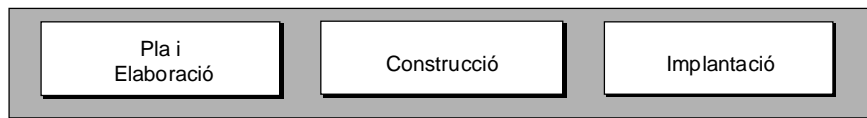
- C. Larman
Applying UML and Patterns.
An Introduction to Object-Oriented Analysis and Design.
Prentice Hall, 1998. (Cap. 33)
- G. Booch, J. Rumbaugh, I. Jacobson
The Unified Modeling Language User Guide
Addison-Wesley, 1999
- B. Powell Douglass
Real-Time UML.
Addison-Wesley, 1998.

El Procés Unificat de Desenvolupament de Software

- Etapes del procés iteratiu de desenvolupament del software
- Cicles de desenvolupament
- Exemple: compra de productes
- Bibliografia

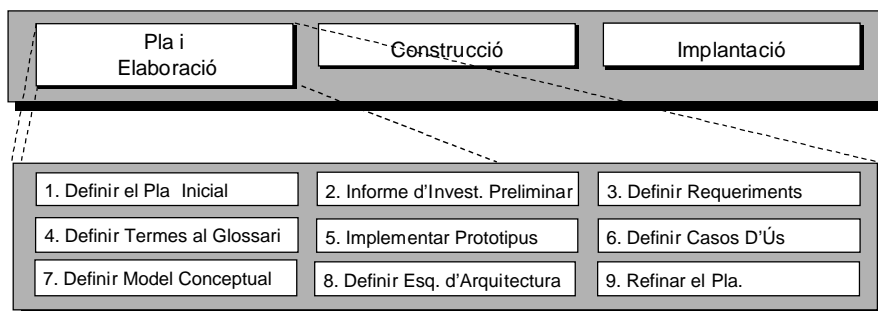
Procés de desenvolupament del software - Macro Nivell

1. **Planificar i Elaborar** - Planificar, definir requeriments, construir prototipus, ...
2. **Construir** - Desenvolupament del sistema (especificació, disseny, etc.)
3. **Implantar** - Implantar el sistema pel seu ús.



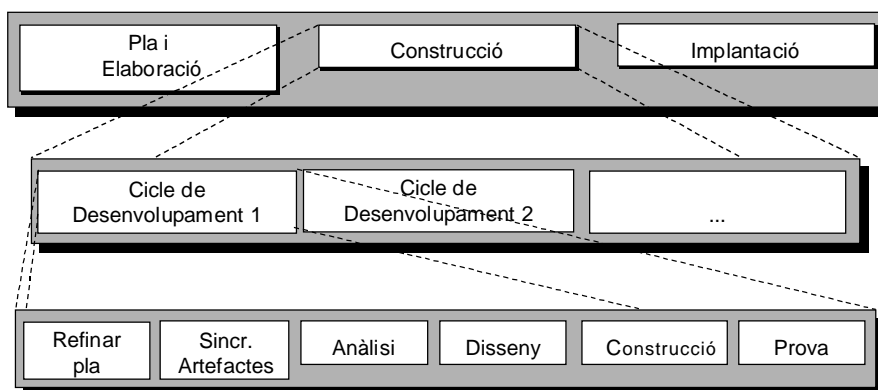
Etapa de planificació i elaboració

Inclou la concepció inicial del projecte, detecció de problemes, determinació d'objectius, investigació d'alternatives de canvi, planificació i especificació dels requeriments.



- **Pla:** calendari, pressupost, etc.
- **Informe d'investigació preliminar:** motivació, alternatives, necessitats de negoci.
- **Especificació de requeriments:** descripció declarativa dels requeriments
- **Glossari:** diccionari de termes (conceptes, noms) i qualsevol informació associada

Etapa de construcció

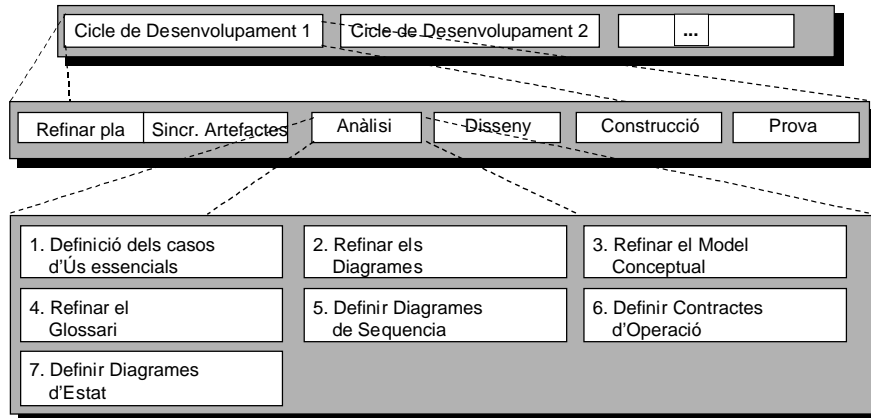


Avantatges:

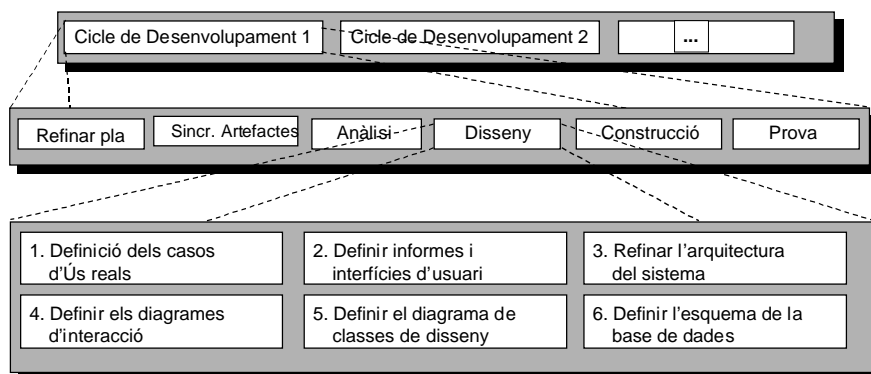
- La complexitat mai no sobrepassa al disenyador
- Primeres impressions molt ràpidament, perquè la implementació d'una petita part del sistema es fa molt ràpidament.

Etapa de construcció: cicles de desenvolupament (I)

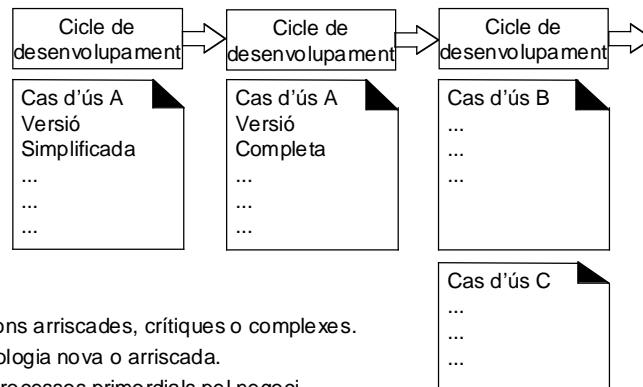
L'etapa de construcció inclou diversos cicles de desenvolupament en els quals el sistema es va estenent de forma progressiva.



Etapa de construcció: cicles de desenvolupament (II)



Ordenar i prioritzar casos d'ús

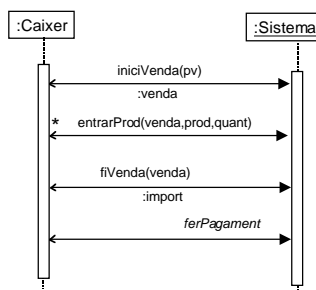


Com prioritzar?

- a. Inclouen funcions arriscades, crítiques o complexes.
- b. Involucra tecnologia nova o arriscada.
- c. Representen processos primordials pel negoci.
- d. Impacte significatiu en el disseny (afegeix moltes classes al domini o requereix molts serveis).
- e. Permet obtenir informació significativa respecte al disseny amb poc esforç.

Exemple: compra de productes

- En un primer cicle de desenvolupament pot no interessar-nos distingir entre les diverses formes possibles de pagament

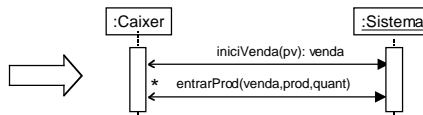


- Interessa desenvolupar el subsistema necessari per poder efectuar una compra
- El contracte de **pagament** hauria de ser prou genèric per no entrar en detalls de la seva forma de pagament

Compra de productes - 2on cicle

- Tenim tants diagrames de seqüència com formes de pagament possibles
- Tots aquests diagrames parteixen del diagrama que s'ha fet al primer cicle de desenvolupament.

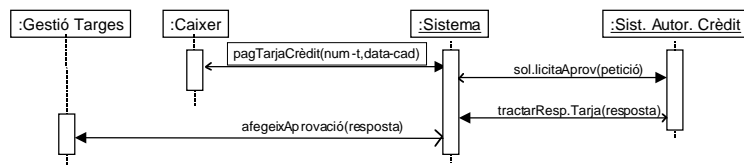
Interacció comuna a tot tipus de pagament



Interacció específica al pagament en metàl·lic



Interacció específica del pagament amb tarja



Nom: pagTarjaCrèdit (num-tarja, data-cad)

Responsabilitats: pagar amb la tarja de crèdit

Excepcions:

Preconditions:

Postconditions:

- creació d'un nou PagamentAmbTarja pmt
- nova associació entre pmt i la venda actual
- ...

Sortida:

- s'envia una sol.licitud d'aprovació al servei d'autorització de crèdit

Nom: tractarResp.Tarja (resposta)

Responsabilitats:

- respondre a la resposta d'aprovació rebuda

Excepcions:

Preconditions:

Postconditions:

- ...

Sortida:

- s'envia una aprovació al sistema de Gestió de Targes perquè l'enregistri

Bibliografia

- C. Larman
Applying UML and Patterns.
An Introduction to Object-Oriented Analysis and Design.
Prentice Hall, 1998. (Cap. 13, 32)
- I.Jacobson, G.Booch, J.Rumbaugh
The Unified Software Development Process.
Addison-Wesley, 1999.

Recull d'exercicis

Part I: Introducció

1. Dóna, i justifica, una definició de l'enginyeria del software.
2. El cicle de vida anomenat "model espiral" està basat en quatre activitats: Planificació, Anàlisi de risc, Enginyeria i Avaluació. Comenta breument l'objectiu de cada una d'aquestes activitats i com s'encadenen.
3. Dóna una definició de requeriment d'un sistema i d'especificació de requeriments d'un sistema.
4. Indica la diferència entre requeriments d'un sistema i requeriments del software.
5. Resumeix les quatre estratègies per a determinar els requeriments.
6. Defineix, i descriu breument, tres requeriments no funcionals, de diferent tipus, relatius al projecte fet durant el curs.
7. Una de les propietats desitjables de les especificacions és que siguin "traçables" (traceability). Defineix aquesta propietat.
8. Una de les propietats desitjables de les especificacions és que siguin verificables. Defineix quan es pot dir que una especificació determinada compleix aquesta propietat. Posa un exemple d'un fragment qualsevol d'una especificació que sigui verificable i un altre que no ho sigui.
9. Molta gent argumenta que el terme "manteniment" és incorrecte aplicat al software - que les activitats associades amb el manteniment del software no són del tot "manteniment". Què en penses?. (Exercici 1.11 de Pressman 93).
10. Indica quina relació hi ha (si és que n'hi ha) entre la construcció de prototipus i el model de desenvolupament de software en espiral.
11. Indica els tres tipus de manteniment de software que hi ha, i explica'ls breument.
12. Quins són els avantatges i els inconvenients principals, si n'hi ha, del cicle de vida basat en el "desenvolupament incremental i iteratiu" d'UML en relació al cicle de vida clàssic, en cascada?
13. Explica breument la diferència entre requeriments funcionals i requeriments no funcionals d'un sistema software. Defineix, i descriu breument, dos requeriments funcionals i dos requeriments no funcionals, de diferent tipus, relatius al projecte fet durant el curs.

Part II : Model Conceptual en UML

1. Feu el Model Conceptual amb la notació UML d'un sistema que conté l'horari i les assignatures de la Facultat, d'una sola enginyeria.

Una assignatura té un codi, un nom i un cert nombre de crèdits (no distingirem entre teoria, problemes i laboratoris), i està assignada a un departament. Les assignatures poden ser obligatòries o opcionals. Les assignatures poden estar relacionades per prerequisits, pre/corequisits i corequisits.

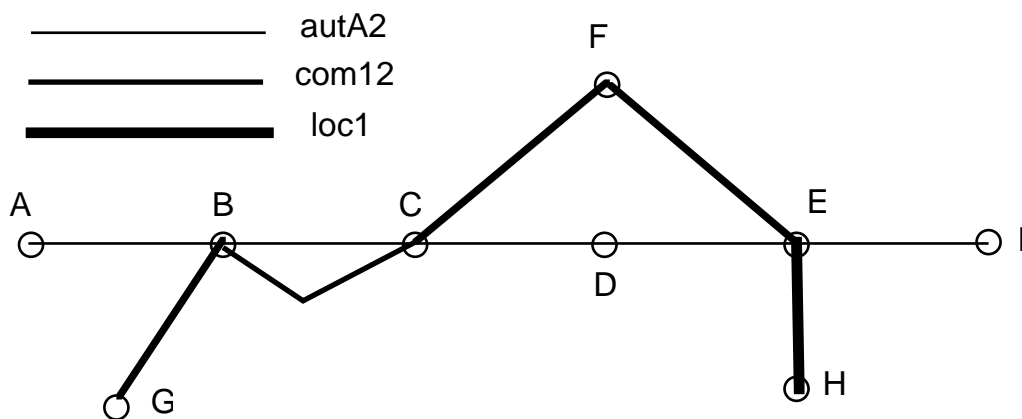
L'horari indica per cada grup d'una assignatura (per exemple, ES:E grup 10) quins dies de la setmana hi ha classe, en quina aula i en quines hores. Els períodes de classe podeu suposar que són d'una hora. Cada assignatura té un cert nombre d'hores de classe (no cal distingir entre hores de teoria, problemes i laboratoris, ni tenir en compte el concepte de subgrup).

Expresseu gràficament totes les restriccions que pugueu. Les restriccions que no es poden expressar gràficament i les regles de derivació dels atributs derivats, si n'hi ha, especifiqueu-les en el llenguatge OCL.

2. Feu el Model Conceptual amb la notació UML d'una empresa de transports que conté informació relativa a rutes, carreteres i poblacions.

L'empresa cobreix un àmbit geogràfic comprès per unes 1000 poblacions. Cada població té un codi i un nom. Aquestes poblacions estan unides per unes 50 carreteres. Cada carretera té un codi. Una carretera consta d'una sèrie de trams consecutius. En mitjana, hi ha 100 trams per carretera. Un tram d'una carretera es defineix per les dues poblacions que uneix i la distància en kilòmetres entre elles. També es considera la duració del trajecte, en minuts, entre aquestes dues poblacions, que, en el cas general, pot ser diferent segons sigui un sentit o l'altre. Per una mateixa població poden passar diverses carreteres.

Un exemple amb 9 poblacions, 3 carreteres i 10 trams podria ser:



El trajecte que ha de recórrer un camió de l'empresa és una ruta. Hi ha unes 30 routes, cada una de les quals té un codi. Una ruta parteix i acaba en una mateixa població, i consta d'una sèrie de trams consecutius de la mateixa o diferents carreteres, que s'han de recórrer en una certa direcció. Per exemple, la ruta 10 podria ser:

(B,C,autA2),(C,F,com12),(F,E,com12),(E,H,loc1),(H,E,loc1),(E,D,autA2),
(D,C,autA2),(C,B,com12)

3. Feu el Model Conceptual amb la notació UML d'un sistema que guardi i processi dades sobre les següents relacions entre persones (no cal que aquestes informacions apareguin explícitament al model, poden obtenir-se indirectament):

1. Casaments (entre dues persones).
2. Divorcis (id.)
3. Fills (o Pare de i Mare de)

El model hauria de mostrar explícitament les restriccions (suposades) següents:

- a) Els casaments són entre un home i una dona.
- b) Dues persones només es poden divorciar si estaven casades.
- c) Una persona és filla d'un home i d'una dona que es van casar.

Pots suposar que:

1. Dues persones no es casen entre elles més d'una vegada.
2. Les persones s'identifiquen pel su nom.
3. De les persones en sabem només el seu sexe.
4. No estem interessats en cap altre atribut.

Un cop fet el model, mostreu-ne la instanciació completa amb les dades següents:

En Joan i la Maria es van casar.
L'Helena és una filla del Joan i la Maria.
En Joan i la Maria es van divorciar.
La Maria es va casar amb el Jordi.
L'Albert és un fill del Jordi i la Maria.

4. Feu el Model Conceptual amb la notació UML d'un sistema que guardi i processi dades sobre les següents relacions entre persones i cotxes:

1. Propietari: un cotxe té una o més persones propietàries.
2. Conductor habitual: un cotxe només té una, i només una, persona que n'és la conductora habitual.
(No cal que aquestes informacions apareguin explícitament al model: poden obtenir-se indirectament).

El diagrama hauria de mostrar explícitament les restriccions (suposades) següents:

- a) El conductor habitual d'un cotxe és un dels seus propietaris (o l'únic).
- b) Una persona pot ésser conductora habitual de més d'un cotxe.

Pots suposar que:

1. Els cotxes s'identifiquen per la seva matrícula.
2. Les persones s'identifiquen pel su nom.
3. No estem interessats en cap altre atribut.

Un cop fet el diagrama, mostreu-ne la instanciació completa amb les dades següents:

En Joan i la Maria són propietaris del cotxe A.
En Jordi i la Maria són propietaris del cotxe C.
En Joan és propietari del cotxe B.
En Joan és el conductor habitual del cotxe B.
La Maria és la conductora habitual del cotxe A.
La Maria és la conductora habitual del cotxe C.

5. Feu el Model Conceptual amb la notació UML d'un sistema que gestioni un conjunt d'escaleres mecàniques d'una gran superfície comercial. Cada escala s'identifica per un codi. En un moment determinat, una escala pot estar en funcionament o en reparació. Independentment d'això, una escala pot estar pujant, baixant o parada, però si està en reparació està necessàriament parada.

Si una escala està en reparació, el sistema guarda quin era l'estat (pujant, baixant, parada) que tenia l'escala en el moment de passar a reparació.

6. Considera una empresa que es dedica a la fabricació d'aparells electro-mecànics i està interessada en construir un sistema que inclogui, entre altres coses, informació sobre la composició dels aparells. Cada aparell consta d'una o més unitats d'un o més components. Un component pot ésser una matèria primera o un altre aparell (que alhora tindrà components). Una matèria primera és un producte que s'adquireix a un (i només un) proveïdor, i no és fabricada per l'empresa. Tant els aparells com les primeres matèries tenen un codi identificador tipus string(5) i un nom tipus string(50).

Per exemple, l'aparell A requereix 5 unitats de l'aparell B, 8 unitats de l'aparell C i 4 unitats de la matèria primera D (la qual és subministrada pel proveïdor P1). L'aparell B consta de 10 unitats de la matèria primera E (del proveïdor P2). L'aparell C consta d'una unitat de l'aparell F, el qual consta de 5 unitats de la matèria primera G (del proveïdor P1).

Un component en la fabricació d'un cert aparell pot tenir zero o més substituïts. Un substituït és un altre component que es pot utilitzar si no es disposa del previst en la fabricació d'un aparell determinat.

Per exemple, si no es disposa d'un aparell B quan es fabrica l'A es pot utilitzar el F en el seu lloc.

Un substituït d'un component que és un aparell pot ésser una matèria primera o un altre aparell. Un substituït d'un component que és una matèria primera només pot ésser una altra matèria primera del mateix proveïdor.

Per exemple, si no es disposa de la matèria primera G quan es fabrica l'aparell F, es pot utilitzar en el seu lloc la matèria primera H. Aquesta matèria primera H és subministrada per P1. Per tant, compleix la restricció indicada.

Feu el Model Conceptual d'aquest sistema amb la notació UML. Expresseu gràficament totes les restriccions que pugueu. Les restriccions que no es poden expressar gràficament i les regles de derivació dels atributs derivats, si n'hi ha, especifiqueu-les en el llenguatge OCL. Heu d'indicar també, necessàriament, la instanciació del model amb les dades de l'exemple que s'ha donat.

Si en fer aquest exercici us cal més informació, feu els supòsits que creieu més adients, i indiqueu-los ben clarament.

7. Considera un Comitè organitzador d'un congrés que està interessat en construir un sistema que inclogui, entre altres coses, informació sobre les ponències que s'hi presentaran. Cada ponència té un codi i un títol i està escrita per un o més autors. Un cop rebudes, cada ponència s'envia a un o més revisors. Els revisors no poden ser autors de cap ponència. Al cap d'un temps, els revisors envien els seus informes de cada ponència que han de revisar. A vegades, un revisor no envia cap dels informes, o no n'envia algun dels que havia de fer. De tota manera, sempre es té almenys un informe de cada ponència. Cada informe, quan es reb, dona una puntuació (de 0 a 10) de la ponència i classifica la ponència en una de les sessions que hi haurà al congrés. Cada sessió correspon a un dels dies del congrés, amb una hora inicial i una hora final.

Per exemple, la ponència 10, de títol 'YSM' és escrita pels autors A1 i A2. La ponència s'envia als revisors Ra, Rb i Rc. El primer li dona un 5 i la classifica en la sessió 'Anàlisi Estructurada Moderna'. El segon li dona un 8 i la classifica en la sessió 'Nous mètodes d'especificació'. El tercer es va oblidar d'enviar l'informe.

La ponència 23, de títol 'La importància dels esdeveniments' és escrita pels autors A2 i A5 i s'envia al revisor Rb, que no contesta, i al Rd, que la qualifica amb un 3 i la classifica en la sessió 'Nous mètodes d'especificació'.

De cada autor o revisor, el sistema en té un codi identificador, el seu nom i la seva adreça.

En una certa data, el Comitè es reuneix i, partint dels informes, decideix quines ponències accepta i quines refusa. Les ponències acceptades s'assignen a una sessió, que ha de ser una de les suggerides pels revisors. Per les ponències rebutjades, es guarda el motiu. Totes les ponències acaben sent acceptades o refusades.

Per exemple, es decideix acceptar la ponència 10 i assignar-la a la sessió 'Anàlisi estructurada moderna'. La sessió 'Anàlisi estructurada moderna' es farà el dia 29/04/94, de 11 a 13. La ponència 23 es refusa pel motiu "molt llarga".

Feu el Model Conceptual d'aquest sistema amb la notació UML. Expresseu gràficament totes les restriccions que pugueu. Les restriccions que no es poden expressar gràficament i les regles de derivació dels atributs derivats, si n'hi ha, especifiqueu-les en el llenguatge OCL. Heu d'indicar també, necessàriament, la instanciació del model amb les dades de l'exemple que s'ha donat.

Si en fer aquest exercici us cal més informació, feu els supòsits que creieu més adients, i indiqueu-los ben clarament.

8. Considera el cas d'una Federació de ciclisme, que vol construir un sistema que tracti, entre d'altres coses, dels resultats de les curses ciclistes. Les curses s'organitzen en edicions de sèries. Cada sèrie consta d'edicions, que es van fent periòdicament. Una sèrie s'identifica per un nom i una edició per la sèrie i l'any. Una edició consta d'un conjunt d'etapes, que varien d'una edició a l'altra. Cada etapa té un número d'etapa, una longitud, una població inici i una població final.

Per exemple, de la sèrie "Volta a Catalunya" se n'han fet tres edicions. La tercera edició tenia dues etapes. La primera anava de Barcelona a Montserrat (50 Km.) i la segona de Montserrat a Poblet (200 Km.)

Un altre exemple pot ser la sèrie "Tour de France", de la qual se n'han fet 20 edicions. La darrera tenia cinc etapes. La primera d'aquestes anava de Paris a Lió, etc.

Interessa també que el sistema enregistri els ciclistes i la participació a les curses. De cada ciclista se n'haurà de saber el seu nom, data de naixement, etc. Cada ciclista que participa a una cursa la pot acabar o no. Si l'acaba és en una certa posició (classificació) i si no l'acaba és per un cert motiu, i interessa saber en quina etapa va córrer per darrera vegada. També cal enregistrar el resultat de ciclista en cada etapa. Com és obvi, un ciclista només pot tenir un resultat en una etapa si participava en la cursa corresponent.

Per exemple, els ciclistes C1, C2 i C3 participen a la tercera edició de la Volta a Catalunya. El C3 va ser el primer, el C1 el segon i el C2 no va acabar, per malatia. La darrera etapa que havia fet era la Barcelona a Montserrat. En la primera etapa el primer va ser el C3, el segon el C2 i el tercer el C1. En la segona etapa, el primer va ser el C3 i el segon el C1.

Algunes etapes tenen Premi de Muntanya a l'arribada. Per aquestes etapes, cal enregistrar quin dels ciclistes que hi va participar va guanyar el premi.

Per exemple, l'etapa Barcelona-Montserrat de la cursa que estem considerant tenia aquest premi (l'altra no). El premi el va guanyar el C2, que, naturalment, havia participat a l'etapa.

Feu el Model Conceptual d'aquest sistema amb la notació UML. Expressen gràficament totes les restriccions que pugueu. Les restriccions que no es poden expressar gràficament i les regles de derivació dels atributs derivats, si n'hi ha, especifiqueu-les en el llenguatge OCL. Heu d'indicar també, necessàriament, la instanciació del model amb les dades de l'exemple que s'ha donat.

Si en fer aquest exercici us cal més informació, feu els supòsits que creieu més adients, i indiqueu-los ben clarament.

9. Considera una empresa que està interessada en construir un sistema que inclogui, entre altres coses, informació sobre els seus empleats. Cada empleat té un número de document d'identitat, un nom i una adreça. Els empleats estan assignats a un, i només un, departament. Cada departament té un nom. Els departaments estan estructurats jeràrquicament, i cada departament pot dependre com a màxim d'un altre departament. Cada Departament té un sol director, que ha de ser un dels empleats que hi estan assignats.

Per exemple, en Joan, la Maria, la Rosa i l'Albert i el Jordi són empleats de l'empresa. En Joan treballa al departament de Vendes, la Maria al Servei Tècnic Postvenda, la Rosa al Laboratori i l'Albert i el Jordi a Recepció. Vendes depèn de Direcció Comercial que, alhora, depèn de Direcció General. El Servei Tècnic Postvenda depèn de Vendes, etc. La directora del Laboratori és la Rosa. El director de Recepció és el Jordi.

Cada empleat és d'una (sola) categoria determinada. Només hi ha tres categories: Venedor, Tècnic i Administratiu. De cada categoria s'ha d'enregistrar els dies de vacances i el plus de sou que tenen.

Per exemple, la Categoria Venedor té 30 dies de vacances i un plus de 10.000 Pts. La Categoria Administratiu té 20 dies de vacances i 20.000 Pts. de plus. El Joan és venedor, la Maria i la Rosa són tècnics i l'Albert administratiu.

Pels empleats que són venedors, s'ha d'enregistrar la zona on treballen. Una zona té un codi i un nom. Un venedor només treballa en una zona. Pels empleats que són tècnics s'ha d'enregistrar els estudis que tenen. Cada estudi té un codi, un nom i el Centre on s'imparteix. Un mateix tècnic pot tenir diversos estudis. Pels empleats que són administratius, s'ha d'enregistrar els cursos de perfeccionament que han fet. Cada curs té també un codi, un títol i una data de realització. Un mateix administratiu pot haver fet diversos cursos.

Per exemple, el Joan treballa a la zona de Girona. La Maria té els estudis d'enginyer en informàtica, i la Rosa el d'electricista i el de mecànica. L'Albert ha fet dos cursos de perfeccionament: Mecanografia i Arxiu.

Feu el Model Conceptual d'aquest sistema amb la notació UML. Expressen gràficament totes les restriccions que pugueu. Les restriccions que no es poden expressar gràficament i les regles de derivació dels atributs derivats, si n'hi ha, especifiqueu-les en el llenguatge OCL. Heu d'indicar també, necessàriament, la instanciació del model amb les dades de l'exemple que s'ha donat.

Si en fer aquest exercici us cal més informació, feu els supòsits que creieu més adients, i indiqueu-los ben clarament.

- 10.** Considera el cas d'un aficionat a la música rock, que vol construir un sistema que gestioni informació sobre la seva fonoteca. El sistema ha de permetre guardar i consultar dades sobre els discos (compactes, LPs o cassettes) i els seus intèrprets.

Els discos són editats per cases discogràfiques. Cada casa discogràfica s'identifica per un nom, i cada disc s'identifica per un codi alfanumèric i també té un nom. Els discs s'estructuren en seqüències de talls. Un tall és una gravació continuada, normalment d'una única cançó, i s'identifica mitjançant un codi alfanumèric. També tenen un nom i una data de gravació. En un disc compacte hi ha una sola seqüència de talls, i cada tall apareix en una certa posició de la seqüència. En un LP o un cassette hi ha una seqüència de talls per cada cara i cada tall apareix en una posició d'una cara. Un mateix tall pot aparèixer a diversos discos (per exemple, al disc original i en una recopilació) però no pot aparèixer més d'un cop al mateix disc.

Per exemple, la casa discogràfica Zafiro ha editat el disc compacte D1 i el disc LP D2. D1 té la cançó C1 com a primer tall i la cançó C2 com a segon tall. El disc D2 té la cançó C1 com a primer tall de la primera cara, i la cançó C3 com a primer tall de la segona cara, i no té més talls.

També es vol tenir informació sobre els intèrprets de cada tall. Un intèrpret s'identifica mitjançant un codi alfanumèric; té un nom, un any de naixement i, si ja és mort, un any de defunció. Un intèrpret pot participar en la gravació d'un tall jugant diversos papers (vocal, guitarra solista, piano, bateria, etc.), però un paper determinat d'un tall només el pot jugar un intèrpret. Malauradament, de vegades es desconeixen els intèrprets d'un tall, o els papers que hi han jugat.

Per exemple, l'intèrpret I1 va néixer l'any 1930 i encara és viu. I2 va néixer el 1920 i es va morir el 1965. Tots dos participaven en la gravació de C1: I1 com a vocalista i piano, i I2 com a bateria.

Feu el Model Conceptual d'aquest sistema amb la notació UML. Expressen gràficament totes les restriccions que pugueu. Les restriccions que no es poden expressar gràficament i les regles de derivació dels atributs derivats, si n'hi ha, especifiqueu-les en el llenguatge OCL. Heu d'indicar també, necessàriament, la instanciació del model amb les dades de l'exemple que s'ha donat.

Si en fer aquest exercici us cal més informació, feu els supòsits que creieu més adients, i indiqueu-los ben clarament.

- 11.** Considereu el cas d'una companyia ferroviària que vol construir un sistema que mantingui els horaris previstos dels seus trens, i els horaris reals.

Cada línia de tren té un codi que la identifica, un tipus de tren, una estació d'origen i una hora de sortida (prevista). Suposarem que cada línia de tren circula diàriament. Cada línia consta d'una sèrie de trams, que van d'una estació a una altra. Les estacions s'identifiquen per un codi. Com a mínim hi ha un tram, que surt de l'estació origen de la línia. Per cada tram d'una línia, ens interessarà mantenir l'hora de sortida prevista de l'estació origen i l'hora d'arribada prevista a l'estació destí. Podem suposar que l'hora d'arribada i la de sortida d'una estació són del mateix dia.

Per exemple, la línia R12 surt de Manresa, cada dia a les 10:00. És un tren Talgo i consta de dos trams. Va de Manresa a Terrassa (on ha d'arribar a les 10:15 i sortir a les 10:17) i de Terrassa a Barcelona (on ha d'arribar a les 10:31).

La línia C240 surt de Barcelona, cada dia a les 23:00. És un tren Correu i consta de 3 trams. Va de Barcelona a Terrassa (on ha d'arribar a les 23:20 i sortir a les 23:30), d'aquesta a Monistrol (on ha d'arribar a les 00:05 i sortir a les 00:15) i d'aquesta a Sant Vicenç de Castellet (on ha d'arribar a les 01:00).

Cada dia, el cap de l'estació origen d'una línia de tren enregistra l'hora de sortida real del tren i la comunica immediatament al sistema. Si el tren no ha pogut sortir (és a dir, si s'ha cancel·lat) s'apunta el motiu. Totes aquestes dades, i les que es mencionen posteriorment, s'han d'enregistrar al sistema, a efectes d'informació al públic i estadístics. Suposarem que, si els trens surten de l'estació origen, acaben arribant a l'estació final.

Per exemple, el dia 26/04/95 el tren de la línia R12 va sortir de Manresa a les 10:01. En canvi, el dia 27/04/95 aquest tren no va poder sortir per 'Tempesta'.

A cada estació que para un tren, el cap d'estació corresponent enregistra l'hora real d'arribada i (si no és l'estació final) l'hora real de sortida i també ho comunica immediatament al sistema. Com és natural, els trens només paren a les estacions que està previst que parin.

Per exemple, el tren de la línia R12 del dia 26/04/95 va arribar a Terrassa a les 10:14 i va sortir a les 10:17. El mateix tren arriba a Barcelona a les 10:33.

Finalment, si el maquinista observa alguna anomalia en un tram, enregistra el tram corresponent (que serà un tram de la línia), l'hora d'observació i l'anomalia detectada. Aquestes observacions es comuniquen al sistema en arribar el tren a l'estació final.

Per exemple, en el tram de Terrassa a Barcelona, el maquinista del tren de la línia R12, del dia 26/04/95, va observar que 'Hi ha tall de corrent' a les 10:23.

Feu el Model Conceptual d'aquest sistema amb la notació UML. Expresseu gràficament totes les restriccions que pugueu. Les restriccions que no es poden expressar gràficament i les regles de derivació dels atributs derivats, si n'hi ha, especifiqueu-les en el llenguatge OCL. Heu d'indicar també, necessàriament, la instanciació del model amb les dades de l'exemple que s'ha donat.

Si en fer aquest exercici us cal més informació, feu els supòsits que creieu més adients, i indiqueu-los ben clarament.

- 12.** Considereu una companyia d'assegurances que està interessada en un sistema pel control dels sinistres en que intervenen els cotxes que té assegurats. Un sinistre el té un cotxe determinat, essent conduït per una certa persona, i ocorre en una certa data. La companyia identifica els cotxes per la seva matrícula, i necessita enregistrar-ne la seva marca i el seu model, entre altres dades. Les persones són identificades pel seu document d'identitat, enregistrent-se també altres dades com ara el seu nom i la seva adreça. No és impossible que un cotxe tingui dos sinistres, amb el mateix o diferent conductor, però seria en dies diferents.

Per exemple, el cotxe 10 (marca Renault, model R6) va tenir un sinistre el dia 10/10/95, quan el conduïa el Joan.

Alguns sinistres requereixen que el cotxe accidentat es porti a un (o més) tallers per a la seva reparació. La companyia té "fixats" els tallers possibles, amb un codi identificador, el seu nom comercial, adreça, etc. No tots els sinistres acaben amb el cotxe a un taller. La companyia indica a quins tallers s'ha de portar el cotxe, i els dies que s'ha de portar. El sistema ha d'anotar on s'assignen els cotxes.

Per exemple, com a conseqüència del sinistre anterior, calia portar el cotxe a dos tallers. Primer, el dia 15/10/95, a el "Mecànic" i després, el 20/10/95 a el "Pintor de cotxes".

Un taller tractarà de reparar el cotxe sinistrat, en la part que li correspongui. En això hi esmerçarà un cert nombre d'hores de ma d'obra. Per altra banda, la reparació pot exigir

(però no sempre) l'ús d'unes certes quantitats de materials determinats. La companyia té codificats aquests materials, i per cada un d'ells es té també el seu nom i el seu preu unitari. Quan un taller acaba una reparació, ho comunica a la Companyia, indicant les dades esmentades, que el sistema ha d'enregistrar.

Per exemple, la reparació indicada anteriorment va requerir al taller "Pintor de cotxes" 15 hores de ma d'obra, i l'ús de 2 litres de pintura blava i d'1 litre de pintura blanca. La pintura blava té el codi ABC, i va a 1000 Pts. el litre, etc.

De quan en quan, els tallers facturen a la Companyia d'assegurances les reparacions que han fet. Una factura pot incloure diverses reparacions, però una reparació només pot estar inclosa en una factura (les reparacions no es facturen parcialment). De cada factura es guarda el seu número i la data de la factura. El sistema ha de poder saber, d'una manera o altra, el codi del taller emissor de la factura. Una factura no pot incloure reparacions de dos o més tallers diferents.

Per exemple, el taller "Pintor de cotxes" esmentat va emetre la factura nº 100 el dia 30/12/95. La factura incloïa la reparació anterior, i la d'altres cotxes.

Feu el Model Conceptual d'aquest sistema amb la notació UML. Expressen gràficament totes les restriccions que pugueu. Les restriccions que no es poden expressar gràficament i les regles de derivació dels atributs derivats, si n'hi ha, especifiqueu-les en el llenguatge OCL. Heu d'indicar també, necessàriament, la instanciació del model amb les dades de l'exemple que s'ha donat.

Si en fer aquest exercici us cal més informació, feu els supòsits que creieu més adients, i indiqueu-los ben clarament.

13. Descriu què són els diagrames de transició d'estats de classes d'objectes en UML. Digues quan es poden definir, i quins components tenen. Posa'n un exemple.
14. Considera un Model Conceptual, especificat amb la notació UML, de les dades d'un sistema que conté *només* una relació ternària R entre les entitats A, B i C. Siguin **a,b,c** ocurrències qualssevol de les entitats A, B, C, respectivament. Indica com s'haurien d'expressar en aquell model les restriccions següents:
 1. Cap **c** pot participar en més de 10 ocurrències de R.
 2. Una parella **a,b** qualsevol pot estar relacionada, via R, amb un màxim de 3 ocurrències de C.
 3. Una tripleta **a,b,c** qualsevol pot estar relacionada, via R, com a màxim una vegada.
15. Considereu una mútua sanitària que està interessada en un sistema pel control dels ingressos hospitalaris en què intervenen els seus socis. Un ingrés el té una persona determinada en un cert centre mèdic i ocorre en una certa data. La mútua identifica els seus socis pel seu número d'associat i n'enregistra també el seu nom i la seva adreça. Els centres mèdics són identificats pel seu nom i s'enregistren també la informació de si el centre té signat un conveni o no amb la mútua. És impossible que una mateixa persona tingui dos ingressos en centres mèdics en una mateixa data, encara que si que pot tenir diversos ingressos en dates diferents.

Per exemple, la soci número 17 (nom Maria, adreça C/Diputació) va ser ingressada a l'hospital de Santa Maria del Mar (que no té conveni amb la mútua) el dia 8/8/1995.

Els ingressos requereixen que a la persona ingressada se li administrin un o més medicaments per a la seva curació. El sistema guarda informació dels possibles

medicaments, que tenen un codi identificador, un nom i un preu unitari. Tots els ingressos requereixen l'administració d'algun medicament, indicant-se en cada cas quin és el nombre diari d'unitats a administrar i la data d'inici i la data final d'administració. Un mateix medicament es pot administrar diverses vegades durant un ingrés. El sistema ha d'enregistrar únicament quins són els medicaments administrats a un pacient que està ingressat en un centre que no té conveni amb la mútua.

Per exemple, com a conseqüència de l'ingrés anterior, la soci 17 va rebre tres medicaments. El medicament 3, en una quantitat de 3 unitats diàries, des del dia 8/8/1995 fins el 10/8/1995. El medicament 5, en una quantitat de 5 unitats diàries, des del dia 8/8/1995 fins el 11/8/1995. Finalment, una segona administració del medicament 3, en una quantitat de 7 unitats diàries, des del dia 12/8/1995 fins el 14/8/1995.

El sistema coneix també els medicaments que són incompatibles als socis. Un soci pot tenir més d'un medicament incompatible i s'enregistrarà, per cada medicament, el motiu de la incompatibilitat. Durant un ingrés, no es poden administrar a un soci medicaments que li són incompatibles.

Per exemple, el medicament 33 és incompatible per a la soci 17 ja que conté penicilina.

Feu el Model Conceptual d'aquest sistema amb la notació UML. Expressen gràficament totes les restriccions que pugueu. Les restriccions que no es poden expressar gràficament i les regles de derivació dels atributs derivats, si n'hi ha, especifiqueu-les en el llenguatge OCL. Heu d'indicar també, necessàriament, la instanciació del model amb les dades de l'exemple que s'ha donat.

Si en fer aquest exercici us cal més informació, feu els supòsits que creieu més adients, i indiqueu-los ben clarament.

- 16.** Considera un Model Conceptual, especificat amb la notació UML, de les dades d'un sistema que conté només una relació ternària R entre les entitats A, B i C. Siguin **a,b,c** ocurrències qualssevol de les entitats A, B, C, respectivament. Indica com s'haurien d'expressar en aquell model les restriccions següents:

1. Tots els **c** han de participar com a mínim en dues ocurrències de R.
2. Una parella **a,b** qualsevol ha d'estar relacionada, via R, amb un mínim de 3 ocurrències de C.
3. Una ocurrència **c** qualsevol ha d'estar relacionada com a màxim amb tres ocurrències distintes de B.
4. Una tripleta **a,b,c** qualsevol pot estar relacionada, via R, com a màxim una vegada, i com a mínim cap.

- 17.** Considereu que la federació d'hoquei patins està interessada en el control dels partits que es disputen en el decurs d'una lliga. Per simplificar, suposeu que convé enregistrar només la informació corresponent a una única lliga. Un partit d'hoquei patins se celebra entre un equip que juga a casa i un equip que juga a fora. Un partit correspon a una determinada jornada. En una jornada es juguen sempre vuit partits. Els equips s'identifiquen pel seu nom i s'enregistra també la seva adreça i el color de la samarreta. Les jornades s'identifiquen per un número de jornada. És impossible que un mateix equip jugui dos partits diferents en una mateixa jornada. Tampoc pot passar que un equip jugui al seu camp en dues jornades diferents amb el mateix equip visitant.

Per exemple, l'equip Vic (adreça C/Guilleries, color Blanc) va jugar contra l'equip Voltregà (adreça C/Osona, color Blau) a la jornada 3. El Tordera és un equip d'hoquei amb adreça C/Riera i color Groc.

La federació també vol guardar informació dels jugadors que juguen els partits de la lliga i dels àrbitres que xiulen aquests partits. Tant jugadors com àrbitres s'identifiquen pel seu document d'identitat, i s'enregistra també el seu nom en ambdós casos. No pot passar que algú sigui jugador i àrbitre alhora. En el cas dels jugadors cal enregistrar també quina és la seva posició (podeu suposar que un jugador té una única posició: porter, defensa, mig, etc.) i el nom de l'equip al que pertanyen. La federació d'hoquei patins vol enregistrar també la informació dels àrbitres que estan recusats pels diversos equips. En una lliga els equips poden recusar fins un màxim de 3 àrbitres si consideren que aquests els han perjudicat en lligues anteriors, enregistrent-se per cada cas el motiu de la recusació.

Per exemple, en Joan té el DNI 111, és porter i pertany al Vic. En Pep té el DNI 222, és davanter i pertany al Voltregà. L'Oriol té el DNI 333 i és un àrbitre. El Tordera ha recusat a l'Oriol perquè els va pitar un penal injust.

Els jugadors dels equips participen en partits durant un determinat nombre de minuts. Un jugador només pot participar en partits en els que hi juga el seu equip. A més, cal enregistrar també la informació del nombre de gols marcats per un jugador en un partit. Un jugador només pot marcar gols en els partits en què participa. El sistema coneix també els àrbitres que xiulen els partits i quina és la qualificació atorgada a l'àrbitre cada cop que xiula un partit. Un àrbitre pot xiular més d'un partit (sempre i quan siguin de jornades diferents), un partit només el xiula un únic àrbitre. Un àrbitre no pot xiular un partit en el que hi juga un equip que l'ha recusat.

Per exemple, en Joan va jugar 40 minuts i en Pep 25 del partit esmentat anteriorment. En Pep va marcar 3 gols en aquest partit. El partit va ser xiutat per l'Oriol, qui va ser qualificat amb Notable.

Feu el Model Conceptual d'aquest sistema amb la notació UML. Expressau gràficament totes les restriccions que pugueu. Les restriccions que no es poden expressar gràficament i les regles de derivació dels atributs derivats, si n'hi ha, especifiqueu-les en el llenguatge OCL. Heu d'indicar també, necessàriament, la instanciació del model amb les dades de l'exemple que s'ha donat.

Si en fer aquest exercici us cal més informació, feu els supòsits que creieu més adients, i indiqueu-los ben clarament.

- 18.** Considereu el cas d'una entitat bancària que està interessada en un sistema pel control de peticions i de concessions de préstecs hipotecaris. Els préstecs demanats s'identifiquen per un codi i s'enregistra també la quantitat de diners sol·licitada i el nombre d'anys en què es tornarà el préstec. Un préstec està demanat per una o més persones. Les persones s'identifiquen pel seu nom i es guarda també informació de la seva adreça i edat. Tot préstec té un únic primer titular. El primer titular d'un préstec ha de ser una de les persones que l'ha demanat.

Per exemple, en Joan (adreça C/València, 25 anys) i la Maria (C/Prat, 24 anys) han demanat el préstec de codi 111 (per un valor de 3 milions, a retornar en 15 anys). La Maria és el primer titular d'aquest préstec. La Carme (C/Balmes, 27 anys) ha demanat el préstec 222 (5 milions, 20 anys) i n'és el primer i únic titular.

Els préstecs demanats són assignats a un o més avaluadors (que s'identifiquen pel seu nom i dels quals es coneix també la seva adreça i edat) per a què els estudiïn. Un avaluador no pot haver sol·licitat cap préstec en aquella entitat bancària. Al cap d'un temps, els avaluadors envien els informes dels préstecs que els hi han estat assignats. A vegades, un avaluador no envia algun dels informes que se li havien assignat. Cada informe, quan es reb, diu si l'avaluador aconsella o no la concessió del préstec. En cas afirmatiu, cal indicar també el tipus d'interés al qual s'hauria de fer efectiu el préstec i en cas negatiu el motiu pel qual no s'hauria de concedir el préstec.

Per exemple, el préstec 111 s'ha assignat als avaluadors Jordi, Anna i Roser. En Jordi i la Roser consideren que cal concedir el préstec amb un interès del 5,5% i 6%, respectivament, i l'Anna no envia l'informe preceptiu. D'altra banda, el préstec 222 s'envia als revisors Pol i Anna que envien un informe negatiu amb el motiu de què s'han sol·licitat massa diners.

En una certa data, l'entitat bancària decideix si concedir o no un préstec demanat a partir dels informes dels avaluadors. Als préstecs concedits se'ls hi assigna un tipus d'interès igual al promig del tipus d'interès suggerits pels revisors que havien emès un informe positiu. Pels préstecs denegats, cal enregistrar el motiu pel qual l'entitat bancària ha decidit de no concedir-los. Un préstec no es pot concedir si té un mínim de dos informes negatius. Cal guardar també informació de la data en què s'ha fet l'avaluació del préstec. Pels préstecs concedits, cal guardar també la informació del compte corrent del qual s'hauran de treure els diners (únic per a cada préstec, identificat per número de compte) i el dia del mes en què s'efectuarà el traspàs dels diners del compte a l'entitat bancària.

Per exemple, el préstec 111 ha estat concedit el dia 5/4/97 a un interès del 5,75%. El pagament d'aquest préstec es farà a partir del compte C567, el dia 4 de cada mes. El préstec 222 ha estat denegat el dia 7/4/97 ja que l'havia demanat una única persona.

Feu el Model Conceptual d'aquest sistema amb la notació UML. Expressen gràficament totes les restriccions que pugueu. Les restriccions que no es poden expressar gràficament i les regles de derivació dels atributs derivats, si n'hi ha, especifiqueu-les en el llenguatge OCL. Heu d'indicar també, necessàriament, la instanciació del model amb les dades de l'exemple que s'ha donat.

Si en fer aquest exercici us cal més informació, feu els supòsits que creieu més adients, i indiqueu-los ben clarament.

- 19.** Considereu el cas d'una companyia propietària de diversos teatres i que està interessada en un sistema pel control de les compres d'entrades de les obres que es representen en els seus teatres. Les obres de teatre s'identifiquen pel seu nom i s'enregistra també el seu autor, el seu director i el nombre d'actors que hi intervenen. Les obres de teatre es representen en diverses sessions (cadascuna de les quals s'identifica pel dia i per l'ordre dins el dia) i en un determinat teatre. Cal guardar també la informació de l'hora d'inici de la representació.

Els teatres s'identifiquen pel seu nom i s'enregistra també el seu aforament (nombre total de butaques de què disposen) i la ciutat on es troben. En un mateix dia no es poden representar obres de teatre diferents en un mateix teatre. Una obra de teatre no es pot projectar en teatres diferents en un mateix dia. En una determinada representació d'un teatre s'hi representa una única obra.

Per exemple, el teatre Monumental és a Figueres i té un aforament de 400 butaques. L'obra "El visitant" és d'en E.Schmitt, està dirigida per la R.M.Sardà i hi participen 25 actors. A la tercera sessió del dia 26/4/96 es representarà al teatre Monumental l'obra "El visitant". Aquesta representació començarà a les 21 hores.

Cada teatre té un cert nombre de butaques, cadascuna de les quals s'identifica (per a un teatre determinat) pel número de la fila i el número de seient dins la fila. El sistema ha de guardar també la informació de les entrades que s'han venut per a una determinada representació. Les entrades es poden vendre de dues maneres diferents: directament a finestreta o bé per telèfon. Per les entrades venudes directament a finestreta cal enregistrar la representació per la qual s'ha venut l'entrada i la butaca assignada.

En el cas d'entrades venudes per telèfon cal enregistrar la mateixa informació que pel cas d'entrades venudes a finestreta i, a més, el número de document d'identitat de la persona que ha comprat l'entrada i del número de tarja de crèdit al que s'ha de carregar la venda. En

una representació no es poden vendre entrades a les que se'ls assigni una butaca que no existeix en el teatre on es fa la representació.

Per exemple, s'han venut tres entrades de la representació anterior: dues a finestra i una per telèfon. Les entrades venudes a finestra ocupen les butaques (1,18) i (1,20), on 1 correspon al número de fila i 18 i 20 al número de seient dins la fila. L'entrada venuda per telèfon ocupa la butaca (5,20), ha estat comprada per una persona amb document d'identitat 111, i s'ha de carregar a la tarja de crèdit 333.

El sistema ha d'emmagatzemar també informació de les persones que estan abonades a la companyia. Un abonat s'identifica pel seu document d'identitat i s'enregistra també el seu nom i la seva adreça. Només poden comprar entrades per telèfon les persones que estan abonades a la companyia propietària del teatre.

Per exemple, en Joan està abonat a la companyia, té el document d'identitat 111 i viu al C/Escudellers.

Feu el Model Conceptual d'aquest sistema amb la notació UML. Expressen gràficament totes les restriccions que pugueu. Les restriccions que no es poden expressar gràficament i les regles de derivació dels atributs derivats, si n'hi ha, especifiqueu-les en el llenguatge OCL. Heu d'indicar també, necessàriament, la instanciació del model amb les dades de l'exemple que s'ha donat.

Si en fer aquest exercici us cal més informació, feu els supòsits que creieu més adients, i indiqueu-los ben clarament.

- 20.** Considereu una federació d'entitats excursionistes que està interessada en un sistema pel control de les expedicions efectuades pels centres excursionistes adscrits a la federació. Una expedició l'efectua un centre excursionista a una certa muntanya, amb una data d'inici i una de finalització. La federació identifica els centres excursionistes pel seu nom i n'enregistra també la seva adreça. Les muntanyes s'identifiquen pel seu nom i se n'enregistra també la seva alçada. Un centre excursionista pot efectuar diverses expedicions a la mateixa muntanya, amb dates diferents. A una muntanya s'hi poden fer diverses expedicions, però en una data concreta hi pot d'haver un màxim de 5 expedicions.

Per exemple, el centre excursionista CEC (adreça Gran Via), va efectuar una expedició a l'Everest (alçada 8848 m) del dia 5/5/1994 al 20/7/1994.

En una expedició hi participen diverses persones (com a mínim una). Una persona pot participar a més d'una expedició. El sistema guarda informació de totes les persones (que tenen el dni com a identificador, un nom i una edat) que han participat a alguna expedició. Algun dels components d'una expedició pot assolir el cim. En aquest cas, s'enregistrarà aquest fet i també la data en què s'ha fet el cim. Una persona pot assolir el cim més d'una vegada en una mateixa expedició.

Per exemple, les persones amb dni 111 (nom Joan, edat 25 anys), 222 (Maria, 23), 333 (Pere, 24) i 444 (Ann, 22) varen participar a l'expedició anterior. Les persones 111 i 222 varen fer el cim el dia 24/6/1994. A més, la persona 222 va tornar a fer el cim el dia 30/6/1994.

Quan una persona participa en una expedició hi desenvolupa un determinat paper. Per simplificar, suposarem que els papers possibles són: metge, alpinista, encarregat de material i cap d'expedició. En una expedició un paper pot ser desenvolupat per més d'una persona. Una mateixa persona pot desenvolupar més d'un paper en una expedició. No tots els papers tenen perquè desenvolupar-se en una expedició.

Quan una persona desenvolupa el paper d'alpinista en una expedició, el sistema ha d'enregistrar també l'assegurança mèdica (que té un nom d'assegurança identificador i el

nom de la mútua que la cobreix) contractada per l'ocasió. En el cas de què una persona d'una expedició hi desenvolupi el paper de metge, cal enregistrar també el nom del centre mèdic en el que treballa actualment (que suposarem que és únic).

Per exemple, totes quatre persones feien el paper d'alpinista a l'expedició anterior i tenien una assegurança de nom "Accidents a l'Himàlaia" (coberta per la Mútua de Terrassa). La persona 222 era el metge de l'expedició i treballava a l'Hospital del Mar. La persona 444 era el cap de l'expedició.

Feu el Model Conceptual d'aquest sistema amb la notació UML. Expressen gràficament totes les restriccions que pugueu. Les restriccions que no es poden expressar gràficament i les regles de derivació dels atributs derivats, si n'hi ha, especifiqueu-les en el llenguatge OCL. Heu d'indicar també, necessàriament, la instanciació del model amb les dades de l'exemple que s'ha donat.

Si en fer aquest exercici us cal més informació, feu els supòsits que creieu més adients, i indiqueu-los ben clarament.

- 21.** Considereu que un club de tennis està interessat en el control dels partits que disputen els seus socis en el torneig social del club. Per simplificar, suposeu que convé enregistrar només la informació corresponent a un únic torneig. Un partit de tennis se celebra entre dos socis i correspon a una determinada jornada. En una jornada es juguen sempre vint partits. Els socis s'identifiquen pel seu nom i s'enregistra també la seva adreça i edat. Les jornades s'identifiquen per un número de jornada. És impossible que un mateix soci jugui dos partits diferents en una mateixa jornada. Tampoc pot passar que dos jugadors s'enfrontin entre ells en dues jornades diferents.

Per exemple, en Joan (adreça C/Guilleries, 27 anys) va jugar contra en Josep (C/Osona, 25 anys) a la jornada 3.

El club de tennis vol guardar també informació dels jutges principals que arbitren els partits disputats. Els jutges, com els socis, s'identifiquen pel seu nom i se n'enregistra també la seva adreça i edat. No pot passar que algú sigui soci del club de tennis i jutge de la competició alhora. El club de tennis vol enregistrar també la informació dels jutges que estan recusats pels diversos socis.

En un torneig els socis poden recusar fins un màxim de 5 jutges si consideren que aquests els han perjudicat en tornejos anteriors, enregistrent-se per cada cas el motiu de la recusació. El sistema coneix també els jutges que arbitren els partits i quina és la qualificació atorgada al jutge cada cop que arbitra un partit. Un jutge pot arbitrar més d'un partit; un partit només l'arbitra un únic jutge. Un jutge no pot arbitrar un partit en el que hi participi un jugador que l'ha recusat.

Per exemple, l'Oriol viu al C/Tortosa, té 29 anys i és jutge. La Maria viu al C/de Mar, té 29 anys i és jutge. En Joan ha recusat a l'Oriol perquè l'any passat li va fer perdre un partit. El partit entre en Joan i en Josep va ser arbitrat per la Maria, qui va ser qualificada amb Excel·lent.

Cada partit de tennis es disputa a un màxim de tres sets. Guanya un partit el primer jugador que guanya 2 sets. El club de tennis vol guardar també informació dels resultats de tots els sets d'un partit i, en cas que un set es decideixi per tie-break (és a dir, si el resultat final del set és de 7-6), cal enregistrar també el resultat del tie-break.

Per exemple, el partit entre en Joan i en Josep va durar tres sets. El primer el va guanyar en Joan per 6 a 2. El segon set el va guanyar en Josep per 7 a 6 (7-2 al tie break). El tercer el va tornar a guanyar en Josep per 6-3.

Feu el Model Conceptual d'aquest sistema amb la notació UML. Expressen gràficament totes les restriccions que pugueu. Les restriccions que no es poden expressar gràficament i les regles de derivació dels atributs derivats, si n'hi ha, especifiqueu-les en el llenguatge OCL. Heu d'indicar també, necessàriament, la instanciació del model amb les dades de l'exemple que s'ha donat.

Si en fer aquest exercici us cal més informació, feu els supòsits que creieu més adients, i indiqueu-los ben clarament.

- 22.** Considereu el cas d'una biblioteca que està interessada en el control dels préstecs i de les reserves que efectuen els seus usuaris. La biblioteca disposa d'un ampli fons bibliogràfic de llibres i d'un gran nombre d'exemplars de cada llibre. Els llibres s'identifiquen pel seu títol (això ho fem per a simplificar) i se n'enregistra també el seu autor i el nom de l'editorial. Els exemplars d'un llibre s'identifiquen per un número d'exemplar correlatiu (començant des de l'1) dins de cada llibre.

Per exemple, del llibre titulat "Orient, Occident" (escrit per la M. de la Pau Janer i publicat per l'editorial Columna) se'n tenen dos exemplars: l'1 i el 2. Del llibre "L'Atles Furtiu" (Alfred Bosch, Columna) se'n tenen cinc exemplars: l'1, el 2, el 3, el 4 i el 5.

La biblioteca admet que tots els seus usuaris puguin fer tant préstecs com reserves. Un usuari s'identifica per un número d'usuari i el sistema guarda també informació del seu nom i adreça. Una reserva la fa un usuari, per a un llibre determinat, amb una data de recollida de la reserva i s'indica també el nombre de dies reservats. Un préstec el fa un usuari, per a un exemplar de llibre determinat, en una certa data i s'indica també la data màxima en què l'usuari pot retornar l'exemplar que s'endú en préstec.

Un usuari pot fer com a màxim 3 reserves per a una mateixa data. Un usuari pot fer un únic préstec d'un mateix exemplar en un mateix dia. Un mateix exemplar d'un llibre en un cert dia es pot prestar com a màxim a una única persona. No es pot fer una reserva d'un llibre si no queda cap exemplar disponible d'aquell llibre durant tot el període que es vol reservar. No es pot prestar un exemplar si no està disponible durant tot el període de préstec o bé si fer efectiu aquest préstec afecta la disponibilitat de les reserves ja fetes.

Per exemple, el soci número 22 (de nom Bernat i adreça C/Cristina) ha fet dos préstecs diferents de l'exemplar 3 del llibre "L'Atles Furtiu". El primer d'aquests préstecs el va fer el dia 1/9/98 amb data màxima de retorn el 15/9/98. El segon el va fer el dia 20/9/98 amb data màxima de retorn el 20/10/98. El soci 33 (Ruth, C/Sant Bonaventura) ha fet una reserva del llibre "L'Atles Furtiu" amb data de recollida el 10/10/98 i per a un total de 15 dies.

Els préstecs que s'han fet es retornen en una certa data. El sistema ha d'enregistrar quins préstecs s'han retornat i la data en què s'ha fet efectiu el retorn. Les reserves que es fan es poden recollir en la data que s'havia indicat al fer la reserva. El sistema ha d'enregistrar l'exemplar del llibre que s'ha assignat a la reserva. Una reserva només es pot recollir en la data de recollida que s'havia indicat inicialment (és a dir, ni abans ni després).

Per exemple, el primer dels préstecs que havia fet el soci 22 va ser retornat el dia 14/9/98. La reserva que havia fet la sòcia 33 del llibre "L'Atlas Furtiu" va ser recollida el dia 10/10/98 i se li va assignar l'exemplar número 5 d'aquell llibre.

Feu el Model Conceptual d'aquest sistema amb la notació UML. Expressen gràficament totes les restriccions que pogueu (les altres, si n'hi ha, doneu-les en forma de text). Indiqueu els atributs de les classes d'objectes i, si és el cas, els atributs de les associacions. Heu d'indicar també necessàriament la instanciació del model amb les dades de l'exemple que s'ha donat. Si en fer aquest exercici us cal més informació, feu els supòsits que creieu més adients i indiqueu-los ben clarament.

- 23.** Considereu el cas d'una escola bressol que està interessada en construir un sistema que inclogui, entre d'altres coses, informació sobre els nens que té al seu càrrec. Cada nen té un nom, que l'identifica, una adreça i una data de naixement. Els nens tenen diverses persones adultes que són els seus responsables i que el poden recollir a l'escola. De cada adult que és responsable d'un nen, l'escola vol conèixer el seu nom (que l'identifica), el seu número de telèfon i el tipus de responsabilitat que exerceix respecte al nen (pare, mare, avi, àvia, cangur, etc.). Com us podeu imaginar hi ha adults que són responsables de més d'un nen (per exemple, de germans que van plegats a l'escola). Cada nen està inclòs a la cobertura mèdica d'un adult, que ha de ser un dels seus adults responsables. En aquest cas cal conèixer el número de seguretat social (o pòlissa mèdica) de l'adult.

Per exemple, l'Albert, l'Iris, la Judit i l'Ariadna són nens de l'escola. L'Albert té tres adults responsables: la seva mare (Maria), el seu pare (Josep) i la seva àvia (Concepció), i està inclòs a la pòlissa mèdica d'en Josep (número 5643578).

Cada nen està assignat a una (sola) classe. Només hi ha tres classes: "Nadons", "Patufets" i "Llops". De cada classe se n'ha de conèixer el seu nom, el nom de l'educadora i l'aula que ocupa. No pot passar que un nen de més d'un any sigui "nadó". Pels nens que són nadons cal enregistrar la llet que prenen (les llets s'identifiquen pel seu nom i se'n guarda també la seva composició). Pels nadons i pels patufets cal conèixer els bolquers que usen (que també s'identifiquen pel seu nom i es guarda informació també de les possibles contraindicacions). Un nen pot usar diferents tipus de bolquers.

Per exemple, la classe dels Llops la porta la Dolors i ocupa l'aula A1; la dels Nadons la porta la Carme i ocupa l'aula A2; i la dels Patufets la porta la Isabel i ocupa l'aula A3. L'Ariadna és un Nadó, l'Iris és un Patufet i l'Albert i la Judit són Llops. L'Ariadna pren la llet "Dorlat2" i usa els bolquers "Dodot" i "Ausonia". La Iris usa els bolquers "Dodot".

Per tal de coordinar-se entre ells, els pares dels nens de cada classe organitzen una "cadena de comunicació", que ha d'estar enregistrada. D'aquesta manera, quan cal que tothom s'assabenti d'alguna cosa, es truquen els uns als altres en l'ordre que estableix la cadena. En una cadena, tots els nens són de la mateixa classe. A més, a l'escola es fan informes diaris de cada nen on s'hi indica: com ha dormit la migdiada (bé o malament) i com ha menjat (tot, poc o res). De vegades pot passar que un dia no es faci l'informe d'un nen..

Per exemple, a la classe dels Llops la primera de la cadena és la Judit, la segueix l'Albert i així successivament fins a completar tots els nens de la classe. El dia 1-11-98 l'Albert s'ho va menjar tot i va dormir bé la migdiada. En canvi, el dia 2-11-98 l'Albert va dormir malament la migdiada i no va menjar res.

Feu el Model Conceptual d'aquest sistema amb la notació UML. Expresseu gràficament totes les restriccions que pogueu (les altres, si n'hi ha, doneu-les en forma de text). Indiqueu els atributs de les classes d'objectes (en particular, com s'identifiquen) i, si és el cas, els atributs de les associacions. Heu d'indicar també necessàriament la instanciació del model amb les dades de l'exemple que s'ha donat.

Si en fer aquest exercici us cal més informació, feu els supòsits que creieu més adients i indiqueu-los ben clarament.

- 24.** Considereu un centre d'ensenyament que està interessat en un sistema per la gestió de les preguntes que es fan als examens efectuats al centre. Un examen correspon a una certa assignatura, a un cert curs en el que l'assignatura s'imparteix i es realitza en una data determinada. El centre identifica les assignatures pel seu nom i n'enregistra també la seva àrea. Els cursos s'identifiquen pel seu nom i se n'enregistra l'edat habitual dels alumnes que el cursen. Una assignatura pot impartir-se a més d'un curs. El sistema enregistra els crèdits que té una assignatura en un curs determinat i el fet de si una assignatura és obligatòria o opcional en un curs. Es poden fer com a màxim 5 examens per assignatura i curs en el que l'assignatura s'imparteix.

Per exemple, l'assignatura Biologia (àrea Ciències) s'imparteix a ESO-1 (edat 12) com a opcional i amb 6 crèdits. Es va fer un examen de Biologia al curs ESO-1 el dia 24/3/1999.

El sistema gestiona informació de preguntes que s'han posat en algun examen o que poden posar-se en algun examen futur. Les preguntes s'identifiquen per un número i s'enregistra també el seu text, el seu tipus (teòrica, pràctica, test), la seva assignatura i el professor que n'és l'autor. Els professors s'identifiquen pel seu nom i s'enregistra també el seu telèfon.

Per exemple, la pregunta número 1 (text "Expliqueu els descobriments de Mendel", tipus teòrica) és de Biologia i el seu autor és en Joan (telèfon 935286748). La pregunta 2 (text "Trobeu el factor Rh que es pot obtenir de ...", tipus pràctica) és de Biologia i la seva autora és la Núria (telèfon 937226432). La pregunta número 3 (text "Digueu quina ...", tipus test) és també de Biologia i el seu autor és en Lluís (telèfon 937226432).

Tots els examens que s'efectuen al centre tenen un únic professor organitzador de l'examen. A un examen s'hi han de posar entre 1 i 10 preguntes de les que el sistema té enregistrades. Totes les preguntes han de ser de l'assignatura de l'examen. Per cada pregunta d'un examen, s'enregistra el pes que té i la nota promig que han obtingut els alumnes per la pregunta en aquell examen.

Per exemple, l'examen anterior l'organitza la Núria i té tres preguntes: la número 1 (pes 0.4, nota promig 5), la número 2 (pes 0.4, nota promig 7.5) i la número 3 (pes 0.2, nota promig 3).

Per les preguntes teòriques d'un examen, el sistema ha d'enregistrar també la nota mínima i la nota màxima que s'ha obtingut, per les preguntes pràctiques el percentatge d'alumnes que han contestat la pregunta i per les preguntes de test els punts negatius que tenen les respostes errònies.

Per exemple, a l'examen anterior la pregunta 1 té una nota mínima de 1.5 i una màxima de 10, la pregunta 2 té un percentatge de respostes del 90 per cent i la pregunta 3 té 5 punts negatius si és errònia.

Feu el Model Conceptual d'aquest sistema amb la notació UML. Expresseu gràficament totes les restriccions que pogueu (les altres, si n'hi ha, doneu-les en forma de text). Heu d'indicar també necessàriament la instanciació del model amb les dades de l'exemple que s'ha donat. Si en fer aquest exercici us cal més informació, feu els supòsits que creieu més adients i indiqueu-los ben clarament.

- 25.** Considereu que un consorci d'empreses està interessat en informatitzar el seguiment dels projectes informàtics que desenvolupa. Tots els projectes desenvolupats dins el consorci són projectes subcontractats. És a dir, en qualsevol cas, una empresa subcontracta a una altra empresa (o a ella mateixa) perquè porti a terme el projecte. Tot projecte informàtic s'inicia en una certa data i s'emmagatzema també la data prevista de finalització. Lògicament, una empresa pot subcontractar diverses vegades a una altra empresa per a desenvolupar projectes diferents. Les empreses s'identifiquen pel seu codi i s'enregistra també la seva població. Una empresa no pot subcontractar a una mateixa empresa dos projectes diferents amb una mateixa data d'inici.

Per exemple, l'empresa 111 (de El Vendrell) va subcontractar l'empresa 222 (d'Altafulla) per desenvolupar un projecte informàtic que començava el 2-2-1999 i es preveia acabar el 5-5-1999. A més, l'empresa 111 va subcontractar també l'empresa 222 per desenvolupar un projecte del 3-3-1999 al 4-4-1999.

Tot projecte informàtic ha de portar a terme alguna de les etapes tradicionals del cicle de vida d'un projecte software (és a dir, especificació, disseny o implementació). Lògicament, no tot projecte ha de comprendre totes les etapes i una etapa es porta a terme una única vegada en un projecte.

Per exemple, el primer dels projectes anteriors comprenia les etapes d'especificació, disseny i implementació. En canvi, el segon projecte consistia només en una implementació.

El sistema ha de guardar també la informació dels empleats (identificats per nom i dels que s'enregistra també la seva adreça i edat) que treballen a les empreses. Per simplificar, suposarem que un empleat comença a treballar en una empresa en una certa data i que no es dóna mai de baixa. Els empleats que treballen a una empresa, poden participar en els projectes que aquella empresa està desenvolupant a partir d'una certa data (per a simplificar, suposarem també que quan un empleat comença a treballar en un projecte no deixa mai de treballar-hi). Un empleat només pot participar en un projecte en què la seva empresa és la subcontractada.

Per exemple, la Montse (Camí Ral, 21) treballa a l'empresa 111 des del 1-1-1998, i en Joan (Carrer Nou, 22) i la Maria (La Riera, 20) treballen a l'empresa 222 des del 2-2-1998. En Joan i la Maria participen al primer dels projectes anteriors, en Joan des del 2-2-1999 i la Maria des del 3-3-1999.

Quan un empleat treballa en un projecte, ho fa en el marc d'alguna de les etapes de desenvolupament del projecte. Cal enregistrar la informació de les etapes a les que s'ha assignat cada empleat (en el marc d'un projecte) i de les hores treballades en aquesta etapa.

Per exemple, en el marc del primer projecte informàtic, en Joan va ser assignat a l'etapa d'especificació i hi va dedicar 40 hores. En canvi, la Maria va ser assignada a l'etapa de disseny, dedicant-hi també 40 hores, i a l'etapa d'implementació, dedicant-hi 60 hores.

Feu el Model Conceptual d'aquest sistema amb la notació UML. Expressen gràficament totes les restriccions que pugueu (les altres, si n'hi ha, doneu-les en forma de text). Indiqueu els atributs de les classes d'objectes i els atributs de les classes associatives. Heu d'indicar també necessàriament la instanciació del model amb les dades de l'exemple que s'ha donat. Si en fer aquest exercici us cal més informació, feu els supòsits que creieu més adients i indiqueu-los ben clarament.

- 26.** Considereu el cas d'una ciutat de l'illa de Menorca que està interessada en construir un sistema que li permeti gestionar, entre d'altres coses, la informació de les propietats i els lloguers de les seves finques i dels pisos d'aquestes. Per poder localitzar les finques, es necessita saber la informació dels carrers (identificats per nom i dels que s'enregistra també el seu barri) que formen la ciutat. Una finca s'identifica per un número dins un carrer. Per tant, dues finques d'un mateix carrer han de tenir números diferents. Tot carrer ha de contenir alguna finca. Una finca pot estar dividida en diferents pisos, que s'identifiquen pel número de pis i número de porta dins la finca.

Per exemple, el carrer Son Saura (barri del Port) i el carrer Maó (barri del Centre) són carrers de la ciutat. El carrer Son Saura té dues finques, situades als números 2 i 26, i el carrer Maó en té una situada al número 17. La finca situada al número 2 del carrer Son Saura té dos pisos (1er-1a i 1er-2a) i la finca del número 17 del carrer Maó només en té dos (1er-1a i 1er-2a).

Les finques són posseïdes per persones (identificades per nom i de les que s'enregistra també la seva adreça). Una persona pot posseir diverses finques, i una finca pot ser posseïda per diverses persones. Una persona posseeix una certa finca a partir d'una data i en un determinat percentatge. Lògicament, la suma dels percentatges de possessió d'una finca no pot superar el 100%. Per a simplificar, suposarem que el percentatge de possessió d'una finca per part d'una certa persona no varia mai.

Per exemple, la finca situada al número 2 del carrer Son Saura és posseïda per en Joan (C/Algaiarens) en un 40% des del 1-1-1999 i per la Maria (C/Macarella) en un 60% des del 2-2-1999. En canvi, la finca del carrer Maó és posseïda íntegrament per la Maria.

Les persones poden llogar pisos de les finques. Una persona lloga un pis, des d'una certa data i per un nombre de mesos determinat. Una persona pot tenir diversos pisos llogats alhora. Donada una data d'inici i un nombre de mesos de lloguer, un pis és llogat per una única persona. Una persona pot llogar diversos cops el mateix pis, però amb dates d'inici diferents. Una persona no pot llogar un pis d'una finca que posseeix.

Per exemple, en Joan ha llogat el 1er-1a de la finca del número 17 del carrer Maó des del 4-4-1999, per 3 mesos. La Marta (C/Macarelleta) ha llogat el 1er-2a del número 2 del carrer de Son Saura des del 3-3-1999, per 12 mesos.

Els lloguers d'un pis es facturen a un propietari de la finca on es troba situat el pis, concretament al propietari que té el percentatge de possessió més alt d'aquella finca (que podríem anomenar propietari principal). Cal enregistrar la informació del número de compte corrent al que s'ha d'efectuar el pagament. Si una persona és propietària principal de diverses finques, el compte corrent al que es facturen els seus lloguers pot ser diferent per cada finca.

Per exemple, el lloguer de la Marta es facturava a la Maria com a propietària principal de la finca on estava situat el pis, a càrrec del compte corrent 1234. En canvi, el lloguer d'en Joan que també es facturava a la Maria com a propietària principal s'ingressava al compte 6789.

Feu el Model Conceptual d'aquest sistema amb la notació UML. Expresseu gràficament totes les restriccions que pugueu (les altres, si n'hi ha, doneu-les en forma de text). Indiqueu els atributs de les classes d'objectes i de les classes associatives. Heu d'indicar també necessàriament la instanciació del model amb les dades de l'exemple que s'ha donat. Si en fer aquest exercici us cal més informació, feu els supòsits que creieu més adients i indiqueu-los ben clarament.

- 27.** Considereu el cas d'una empresa que està interessada en construir un sistema software que inclogui, entre d'altres coses, informació sobre els seus empleats. Cada empleat té un número de document d'identitat (que l'identifica), un nom i una data d'ingrés a l'empresa. En el decurs de la seva vida laboral, els empleats es van assignant a diversos departaments. Cada departament té un nom (que l'identifica), una adreça i està a una ciutat. L'empresa té un total de 10 departaments. Quan s'assigna un empleat a un departament, se'n registra la seva data d'assignació i quan es desassigna la seva data de desassignació.

Un empleat es pot assignar com a màxim tres vegades al mateix departament, però sempre amb dates d'inici diferents i amb períodes que no se solapin entre sí. En tota la seva vida laboral, un empleat es pot assignar com a màxim a cinc departaments diferents. En un moment determinat, un empleat pot no estar assignat a cap departament, i totes les assignacions s'han de fer en dates posteriors a la data d'ingrés a l'empresa.

Per exemple, l'empleat amb dni 111 (Pere Màrtir, 1-10-98) va estar assignat al departament de Vendes (C/Còrsega, Colera) des del 1-10-1998 al 5-9-1999 i des del 5-10-1999 fins al 20-10-1999 i al departament de Gerència (C/Sardenya, Darnius) des del 21-10-1999 al 1-11-1999. L'empleat amb dni 222 (Maria Dolça, 2-2-99) està assignada al departament de Màrqueting des del 2-2-1999.

Quan un empleat està assignat al departament de Vendes o al de Màrqueting, el sistema ha de guardar informació addicional. Un empleat pot pertànyer alhora al departament de Vendes i al de Màrqueting. Pels empleats assignats al departament de vendes, cal enregistrar els càrrecs que han desenvolupat. Hi ha tres càrrecs possibles: Venedor, Responsable de Vendes i Director General de Vendes, i cal enregistrar la data en què l'empleat va començar a ocupar el càrrec en qüestió.

Per exemple, quan l'empleat amb dni 111 va estar al departament de Vendes durant el període comprès del 1-10-1998 al 5-9-1999 va desenvolupar dos càrrecs: Venedor (des del 1-10-1998) i Responsable de Vendes (des del 1-1-1999). En canvi, quan va tornar a estar al departament de

Vendes, va fer de Venedor (des del 6-10-1999) i de Director General de Vendes (des del 10-10-1999).

Els empleats assignats al departament de Màrketig participen en diversos projectes. Un projecte s'identifica per un nom i té una durada determinada. Quan els empleats participen en projectes hi porten a terme certes activitats (que s'identifiquen pel nom de l'activitat) i dediquen un determinat nombre d'hores setmanals a fer aquesta activitat. Un empleat del departament de Màrketig pot fer més d'una activitat dins el projecte. Una activitat i un projecte qualssevol han de tenir com a mínim la participació d'un empleat del departament de Màrketig.

Per exemple, mentre l'empleat amb dni 222 ha estat al departament de Màrketig, ha participat al projecte "Ara, Lleida" i hi ha fet les activitats de "Responsable d'Imatge", dedicant-hi 20 hores setmanals, i de "Relació amb els mitjans de comunicació", amb una dedicació de 7 hores/setmana.

Atesa la seva importància estratègica, el departament de Màrketig pot rebutjar empleats. Un empleat rebutjat pel departament de Màrketig no es pot assignar mai a aquest departament.

Per exemple, l'empleat amb dni 111 fou rebutjat pel departament de Màrketig i, per tant, no s'hi podrà assignar mai.

Feu el Model Conceptual d'aquest sistema amb la notació UML. Expresseu gràficament totes les restriccions que pogueu (les altres, si n'hi ha, i els possibles atributs derivats doneu-los en forma de text). Indiqueu els atributs de les classes d'objectes i, si és el cas, els atributs de les associacions. Heu d'indicar també necessàriament la instanciació del model amb les dades de l'exemple que s'ha donat.

Si en fer aquest exercici us cal més informació, feu els supòsits que creieu més adients i indiqueu-los ben clarament.

- 28.** Considereu una empresa que es dedica a l'organització de congressos i que està interessada en un sistema per a la gestió de les ponències i dels assistents als diversos congressos que organitza. Un congrés té unes sigles i un any que l'identifiquen i cal que el sistema enregistri també l'idioma que s'utilitza en el congrés, la ciutat on es realitza i el preu de la inscripció.

Per exemple, el congrés ER de l'any 1998 té: idioma anglès, ciutat Singapur i preu 50000 pts. El congrés ER de l'any 1999 té: idioma anglès, ciutat París i preu 60000 pts.

A cada congrés s'hi envien moltes ponències. A cada ponència se li assigna un codi que l'identifica i també s'enregistra el seu títol i qui són els seus autors. De cada autor, el sistema ha de tenir el seu nom que es considerarà identificador i també el seu e-mail. Algunes de les ponències enviades són escollides per a ser presentades al congrés. Totes les ponències escollides per a un determinat congrés han d'estar entre les que s'havien enviat per a aquell congrés. És possible que una mateixa ponència es presenti a diversos congressos però aleshores pot ser escollida com a màxim per a un de sol. De les ponències escollides, el sistema ha d'enregistrar la puntuació que se'ls ha atorgat.

Per exemple, a l'ER del 1999 s'hi ha presentat la ponència de codi 'C125' (títol 'Evolució de jerarquies') que té per autors en Jordi (email jordi@fib.upc.es) i la Rosa (email rosa@fib.upc.es). Aquesta ponència ha estat escollida i se li ha atorgat una puntuació de 7.

Un congrés s'estructura en sessions. Cada sessió té un nom que indica la temàtica de la sessió i que no es pot repetir en sessions diferents d'un mateix congrés però que sí es pot repetir en congressos diferents. Cada ponència escollida s'assigna a una sessió del congrés

per al qual se l'ha escollit. A una sessió se li poden assignar com a màxim quatre ponències.

Per exemple, a l'ER del 1999 hi ha una sessió que té el nom 'Modelització conceptual' i la ponència 'C125' ha estat assignada a l'esmentada sessió.

Les persones que volen assistir a un congrés s'hi inscriuen i el sistema ha d'enregistrar el seu nom que es considera identificador i el seu e-mail. Hi ha dos tipus d'inscripcions: les inscripcions normals i les inscripcions d'estudiants. Per a les inscripcions d'estudiants cal enregistrar el nom dels estudis que està realitzant la persona inscrita en el moment de la inscripció. Cal considerar que una mateixa persona pot tenir inscripcions de tipus diferents (per congressos diferents) i també que una persona pot tenir diverses inscripcions com a estudiant cadascuna amb uns estudis diferents.

Per exemple, a l'ER del 1998 s'hi va inscriure en Jordi com a estudiant (nom estudis Informàtica). A l'ER del 1999 s'hi ha inscrit en Jordi amb inscripció normal i la Rosa també amb inscripció normal. Una altra persona, la Maria (email maria@fib.upc.es) s'ha inscrit a l'ER del 1999 amb inscripció normal.

Per cada congrés, existeixen alguns hotels que serviran per allotjar als inscrits al congrés que ho desitgin (no obligatòriament). El sistema ha d'enregistrar quins són els hotels de cada congrés. Els hotels s'identifiquen pel seu nom. Cal que també s'enregistrin tots els allotjaments de les persones inscrites en un congrés corresponents a hotels del congrés. De cada allotjament s'enregistrarà la data d'inici i el nombre de nits. Es considerarà que una persona inscrita en un congrés pot tenir diversos allotjaments diferents en períodes que no se solapin.

Per exemple, els hotels de l'ER del 1999 són el Montmartre i el Sena. En Jordi, per a la seva inscripció a l'ER del 1999, s'allotja a l'hotel Montmartre durant 2 nits des del 5-11-1999, s'allotja a l'hotel Sena durant 2 nits des del 7-11-1999 i s'allotja a l'hotel Montmartre durant 3 nits des del 9-11-1999.

Feu el Model Conceptual d'aquest sistema amb la notació UML. Expresseu gràficament totes les restriccions que pogueu (les altres, si n'hi ha, doneu-les en forma de text). Heu d'indicar també necessàriament la instanciació del model amb les dades de l'exemple que s'ha donat. Si en fer aquest exercici us cal més informació, feu els supòsits que creieu més adients i indiqueu-los ben clarament.

- 29.** Considereu el cas d'una cadena de botigues de roba que està interessada en construir un sistema software que inclogui, entre d'altres coses, informació sobre els seus productes i les botigues on es venen.

Cada producte té un codi (que l'identifica), una descripció, un preu, una data d'alta i, si ja no es fabrica, una data de baixa. De cada producte se'n dissenyen diverses talles i de cada talla se'n fabriquen diverses peces (aquesta cadena de botigues no dissenya models "exclusius"). Les talles s'identifiquen pel número de talla. Els productes poden ser de tres tipus: infantil, juvenil i adult. Un producte es considera de tipus infantil si s'ha dissenyat per alguna talla entre la 2 i la 12; de tipus juvenil si s'ha dissenyat per alguna talla entre la 34 i la 38; i de tipus adult si s'han dissenyat talles superiors a la 38. Res impedeix que un producte sigui de més d'un tipus (per exemple, infantil i juvenil).

Per exemple, el producte amb codi 333 ("forro polar de flors", 6500, 1-9-1999) es va dissenyar en les talles 2,4 i 6. El producte amb codi 444 ("abric blau marí", 14000, 1-9-1998) es va dissenyar en les talles 12,34 i 36, i es va donar de baixa el 30-6-1999). El producte amb codi 555 ("guants de llana", 1500, 1-9-1998) es va dissenyar en les talles 2 i 4. Per tant, els productes 333 i 555 són infantils, i el producte 444 és infantil i juvenil.

Les botigues de la cadena tenen un número, que les identifica, i una adreça. El sistema ha de guardar, per cada producte, la quantitat de peces de cada talla que estan disponibles (a la venda) a cada botiga.

Per exemple, a la botiga número 1 (Rambla Catalunya) del producte 333 tenen tres peces de la talla 4, una peça de la talla 2 i la talla 6 està exhaurida. A la botiga número 2 (Diagonal) del producte 444 tenen dos peces de la talla 12, una peça de la talla 34 i una peça de la talla 36.

De vegades, les botigues fan promoció de productes. El sistema ha d'enregistrar la botiga, el producte promocionat, les dates d'inici i final de la promoció i el descompte que s'aplica al preu del producte. Un producte pot ser promocionat fins a 3 vegades en una mateixa botiga, però amb dates d'inici diferents. En una botiga determinada no hi pot haver més de 10 productes en promoció simultàniament.

Per exemple, a la botiga numero 2 es promociona el producte 444 des del dia 28-12-99 fins el dia 30-1-2000 aplicant-hi un descompte del 50%.

Dos cops l'any l'empresa elabora els catàlegs dels seus productes per promocionar-los en les temporades de primavera/estiu i tardor/hivern, respectivament. S'elaboren tres tipus de catàlegs: l'infantil, el juvenil i l'adult. El sistema ha d'enregistrar, per cada tipus de catàleg i temporada, la data en que entra en vigència i el conjunt de productes que en formen part. Com és natural, en el catàleg infantil només hi poden haver productes de tipus infantil, en el juvenil productes de tipus juvenil i en l'adult productes de tipus adult.

Per exemple, al catàleg infantil tardor/hivern99 (1-9-1999) hi trobem el producte 333 i el producte 555.

Feu el Model Conceptual d'aquest sistema amb la notació UML. Expresseu gràficament totes les restriccions que pugueu (les altres, si n'hi ha, i els possibles atributs derivats doneu-los en forma de text). Indiqueu els atributs de les classes d'objectes i, si és el cas, els atributs de les associacions. Heu d'indicar també necessàriament la instanciació del model amb les dades de l'exemple que s'ha donat.

Si en fer aquest exercici us cal més informació, feu els supòsits que creieu més adients i indiqueu-los ben clarament.

- 30.** Considereu un club de jubilats que està interessat en un sistema que gestioni els viatges que organitzen dins el club. Els jubilats s'identifiquen pel seu nom i cal que el sistema enregistri el seu any de naixement. Els viatges s'identifiquen per un número i cal enregistrar el tipus de transport que s'hi utilitza. Els jubilats que desitgen fer un viatge han de fer una reserva pel viatge (que s'ha d'enregistrar). També cal enregistrar quins jubilats d'entre els que havien reservat un viatge l'han fet finalment juntament amb el seu grau de satisfacció pel funcionament del viatge. Si un viatge no té com a mínim dues persones que el facin, el viatge s'anula i el sistema no l'enregistra.

Per exemple, en Joan (any naixement 1930), la Carme (any naixement 1933) i en Carles (any naixement 1931) són jubilats del club. S'ha organitzat un viatge de número 25 (tipus transport autobús). Per al viatge 25 s'han fet tres reserves: una d'en Joan, una de la Carme i una d'en Carles. Finalment, els que han fet el viatge 25 han estat en Joan amb un grau de satisfacció de 5 i la Carme amb un grau de 10.

Quan un jubilat fa un viatge té la possibilitat de contractar una assegurança pel viatge. En aquests casos, s'enregistra el número de l'assegurança i la mutua asseguradora.

Pel viatge 25, en Joan ha contractat una assegurança que té el número 111 de la mutua Segur.

En un viatge es fan parades a diverses localitats. Per cada parada del viatge cal enregistrar la localitat, la data d'inici de la parada i la data de fi de la parada. En una determinada data un viatge no pot iniciar parada a més d'una localitat. Un viatge pot parar com a màxim dues vegades a la mateixa localitat i pot fer com a màxim deu parades en total. Les localitats s'identifiquen pel nom i es vol saber també el seu nombre d'habitants.

El viatge 25 té una parada a Figueres (30000 habitants) que s'inicia el dia 6-10-1999 i finalitza el dia 8-10-1999. També té una parada a Castelló d'Empúries (5000 habitants) que s'inicia el 8-10-1999 i finalitza el 9-10-1999.

Les parades d'un viatge poden incloure diverses visites. A cada visita se li assigna un número identificador i el sistema ha d'enregistrar per cada visita a quina parada correspon (una sola), el nom de la visita, i la seva data.

Per exemple, la parada que el viatge 25 inicia el dia 6-10-1999 a Figueres inclou la visita número 1234 (nom 'museu Dalí', data 7-10-1999). La parada que el viatge 25 inicia el dia 8-10-1999 a Castelló d'Empúries inclou la visita número 1235 (nom 'museu de la Catedral', data 8-10-1999).

Els jubilats que fan un viatge poden donar, si volen, la seva valoració (molt bona, bona, regular, dolenta) de les visites incloses a les parades del viatge. Aquestes valoracions han de ser enregistrades pel sistema.

Per exemple, en Joan ha valorat la visita 1234 com a regular i la visita 1235 com a molt bona.

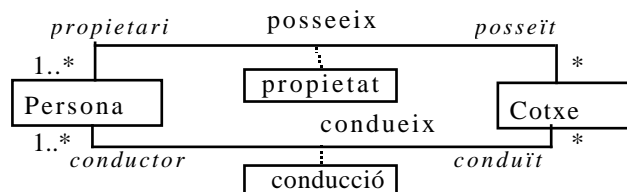
Entre els jubilats que fan un viatge es fa un sorteig i el guanyador rep un regal. El sistema ha d'enregistrar qui és el guanyador i quin ha estat el seu regal.

Per exemple, pel viatge 25 la guanyadora del sorteig ha estat la Carme que ha tingut com a regal un rellotge.

Feu el Model Conceptual d'aquest sistema amb la notació UML. Expressen gràficament totes les restriccions que pogueu (les altres, si n'hi ha, doneu-les en forma de text). Heu d'indicar també necessàriament la instanciació del model amb les dades de l'exemple que s'ha donat. Si en fer aquest exercici us cal més informació, feu els supòsits que creieu més adients i indiqueu-los ben clarament.

Object Constraint Language (O.C.L.)

1. A partir del Model Conceptual següent, expressen en OCL les restriccions textuais a) i b). La classe Persona té un únic atribut *nom* i la classe Cotxe un únic atribut *matrícula*, ambdós identificadors.



- a) Una persona no pot conduir un cotxe que posseeix.
- b) Tot conductor d'un cotxe ha de ser un dels propietaris d'aquell cotxe.

Part III : Altres models d'UML

Problemes

1. Considereu un sistema senzill de control de préstecs en una biblioteca. El sistema ha d'admetre altes i baixes de socis i de llibres. Els socis poden demanar llibres en préstec, però no es poden tenir més de tres llibres en préstec en un moment determinat. Els llibres s'han de tornar abans d'un mes de la data del préstec. Cada cop que un soci torna un llibre més enllà de la data de devolució, es penalitza reduint en una unitat el nombre de llibres que pot tenir simultàniament. Quan arriba a zero, el soci es dona de baixa automàticament.

Us demanem que feu els models següents (tots ells mitjançant la notació UML):

- *Model Conceptual.*
- *Model de Casos d'Ús:* diagrama de casos d'ús i especificació dels casos d'ús.
- *Model del Comportament del Sistema:* diagrames de seqüència i contractes de les operacions.
- *Model dels Estats:* diagrama d'estats per objectes i casos d'ús.

2. Considereu un sistema senzill de reserva i utilització de pistes de tennis d'un club. L'ús de les pistes està reservat als socis de club, i s'han de preveure les altes i baixes de socis. El club té tres pistes, que es poden reservar per blocs d'una hora. Les reserves es poden cancel·lar, si no són pel mateix dia. No hi ha límit en les reserves que pot fer un soci, però no es poden fer reserves per més enllà d'un mes. Cada final de mes s'envia una factura als socis, carregant l'ús que han fet de les pistes durant el mes. Cada hora reservada costa un cert preu, i l'import de la factura es calcula multiplicant la suma de les hores reservades durant el mes per aquell preu.

Cal tenir en compte si les pistes reservades s'ocupen o no. La primera "no ocupació" d'una pista durant un any natural, no es carregarà a la factura del soci, però la resta es carregaran com si s'hagués utilitzat realment.

Us demanem que feu els models següents (tots ells mitjançant la notació UML):

- *Model Conceptual.*
- *Model de Casos d'Ús:* diagrama de casos d'ús i especificació dels casos d'ús.
- *Model del Comportament del Sistema:* diagrames de seqüència i contractes de les operacions.
- *Model dels Estats:* diagrama d'estats per objectes i casos d'ús.

3. Considera el cas d'una empresa que es dedica a gestionar l'adopció de gossos. L'empresa disposa d'uns gossos, cadascun dels quals té un identificador i és d'una raça determinada. Aquests gossos poden ésser adoptats pels clients de l'empresa. Cada client té un identificador i una raça preferida de gossos. En un moment determinat, cada client pot tenir adoptat cap o un gos. Alhora, un gos pot estar lliure o adoptat per un client determinat. El sistema ha de respondre a tres esdeveniments: Alta de gos, Alta de client i Canvi de gos.

Quan es produeix una alta d'un gos, ens comuniquen el seu identificador i la seva raça. Si hi ha algun client qualsevol que estigui esperant gossos d'aquesta raça, se li assigna el nou gos, i es produeix una sortida "Adopció feta", indicant l'identificador del gos i el del client. En cas contrari, el gos queda lliure per ser adoptat en el futur. No es considera que els gossos es puguin donar de baixa.

Quan es produeix una alta d'un client, ens comuniquen el seu identificador i la raça que prefereix. Si hi ha algun gos qualsevol d'aquesta raça lliure, se l'assigna al client i es

produeix una sortida "Adopció feta", indicant l'identificador del gos i el del client. En cas contrari, el client queda a l'espera d'adoptar un gos en el futur. No es considera que els clients es puguin donar de baixa.

Quan es produeix l'esdeveniment de Canvi de gos, ens comuniquen l'identificador de la persona que ens torna el gos (no cal que ens indiquin el gos que torna). El canvi només s'accepta si tenim algun altre gos lliure de la raça preferida pel client. En aquest cas, s'assigna un gos qualsevol d'aquests al client i es produeix una sortida "Adopció feta", indicant l'identificador del gos i el del client. No cal guardar la història de les adopcions fetes pels clients. Podeu suposar que les races estan codificades.

Us demanem que feu els models següents (tots ells mitjançant la notació UML):

- *Model Conceptual.*
- *Model de Casos d'Ús:* diagrama de casos d'ús i especificació dels casos d'ús.
- *Model del Comportament del Sistema:* diagrames de seqüència i contractes de les operacions.
- *Model dels Estats:* diagrama d'estats per objectes i casos d'ús.

4. Considera un sistema que enregistra les compres que fan els clients, i que les factura. El sistema ha de respondre a tres esdeveniments: nova compra, fer factures i fi d'any. Quan es produeix una **nova compra** ens comuniquen el codi del client que l'ha fet, el codi del producte que ha comprat, la quantitat comprada d'aquest producte i la data de la compra. Se suposa que en una compra el client només compra un sol producte.

El sistema accedeix a dos arxius ja existents: **Productes** i **Clients**. L'arxiu de productes conté, per cada producte, el seu codi, la descripció i el preu unitari. L'arxiu de Clients conté, per cada client, el seu codi, el seu nom i el total comprat en el darrer any.

Un producte pot ésser comprat per un nombre indeterminat de clients. Alhora, un client pot arribar-nos a comprar un nombre indeterminat de productes. Un client pot comprar-nos el mateix producte en diverses ocasions.

Quan es produeix l'esdeveniment **fer factures** el sistema ha de generar factures de totes les compres que encara no s'han facturat. Una factura s'identifica per un número de factura (que el sistema assigna correlativament). Interessa enregistrar, de cada factura, la data en què s'ha fet i el seu import. S'ha de generar una factura per cada client que té una o més compres no facturades. L'import de la factura és la suma dels imports de les compres. En acabar el procés, ha de sortir un missatge que digui el primer i el darrer número de factura generat o, si no n'ha generat cap (perquè totes les compres ja estaven facturades), l'avís: 'Cap factura feta'. (No tindrem en compte el llistat de les factures).

Quan es produeix l'esdeveniment **fi d'any**, el sistema ha de calcular, per cada client, l'import total de les compres que ens ha fet durant l'any, i enregistrar-lo a l'arxiu Clients.

Us demanem que feu els models següents (tots ells mitjançant la notació UML):

- *Model Conceptual.*
- *Model de Casos d'Ús:* diagrama de casos d'ús i especificació dels casos d'ús.
- *Model del Comportament del Sistema:* diagrames de seqüència i contractes de les operacions.
- *Model dels Estats:* diagrama d'estats per objectes i casos d'ús.

5. Una federació de ciclisme ha decidit facilitar l'organització de curses per etapes per aficionats. Amb aquesta intenció ha encarregat a una empresa de software la construcció d'un sistema informàtic capaç de gestionar les dades d'una única cursa.

La cursa consta de diverses etapes, numerades correlativament. Els ciclistes tenen un número de dorsal i es vol conèixer també el seu nom i el del seu equip. A cada etapa, els corredors arriben en un cert ordre i havent tardat un determinat temps. La classificació general de la cursa s'estableix segons el temps acumulat durant totes les etapes disputades.

El sistema ha de respondre als tres esdeveniments següents: donar d'alta un corredor, registrar el resultat d'una etapa i generar un llistat amb la classificació general. Per donar d'alta un corredor es proporciona el seu número de dorsal, el seu nom i el del seu equip. El sistema ha de comprovar que no existeix un altre ciclista amb el mateix número. Els resultats d'una etapa es registren un cop arribats tots els corredors (exceptuant els que abandonen o arriben fora de control). Per cada un d'ells s'introdueixen el número, la posició en la que ha arribat i el temps que ha consumit. El sistema comprova que les dades entrades són correctes, genera un número d'etapa (el més alt que ja existeixi més un) i emmagatzema els resultats. A més, s'imprimeix un llistat en el qual es mostra el número d'etapa i, per cada corredor, la classificació, el número, el temps, el nom i l'equip. En qualsevol moment es pot demanar un llistat amb la classificació general (que serà provisional si no ha acabat la cursa). S'ha de mostrar el número de l'última etapa recorreguda i, per cada corredor que l'ha acabada, la classificació, el número, el temps acumulat, el nom i l'equip.

Us demanem que feu els models següents (tots ells mitjançant la notació UML):

- *Model Conceptual.*
- *Model de Casos d'Ús:* diagrama de casos d'ús i especificació dels casos d'ús.
- *Model del Comportament del Sistema:* diagrames de seqüència i contractes de les operacions.
- *Model dels Estats:* diagrama d'estats per objectes i casos d'ús.

6. Els gestors de *El setè segell*, una sala de cinema, han decidit oferir als seus clients un servei de reserva per telèfon d'entrades numerades. Amb aquest propòsit, han encarregat a un estudiant d'ES:E la construcció d'un sistema informàtic.

Els films es projecten en sessions, cadascuna de les quals s'identifica pel dia i per l'ordre dins del dia. Cada sessió comença en una hora determinada i projecta una determinada pel·lícula (de la qual només ens interessa el seu nom). En un mateix dia, es poden projectar pel·lícules diferents. La sala disposa d'un nombre fix de butaques, cadascuna de les quals s'identifica pel número de la fila i el número dins la fila.

De tots els esdeveniments als quals ha de respondre el sistema, només ens n'interessen quatre: 1) compra directa a la finestra (sense reserva prèvia) d'una entrada, 2) petició (telefònica) de reserva d'una entrada, 3) recollida i pagament (a finestra) d'una entrada reservada i 4) consulta de l'ocupació d'una sessió. Altres esdeveniments (p.e., altes i baixes de sessions) poden ignorar-se de cara a l'examen.

Per comprar directament una entrada es proporcionen la sessió, la fila i el número de la butaca. Un cop fetes les comprovacions pertinents, s'anota l'ocupació de la butaca i s'imprimeix l'entrada amb les dades necessàries. En el cas d'una reserva es requereixen les mateixes dades i el DNI de la persona que ha de recollir l'entrada. Després de les validacions oportunes, s'anota la reserva amb el DNI. Per recollir una reserva es torna a entrar la mateixa informació; es fan, com sempre, les comprovacions necessàries, es registra que l'entrada ha sigut pagada i s'imprimeix com en el cas de la compra directa. Finalment, l'ocupació d'una sessió es pot consultar mitjançant el seu identificador. Com a resposta s'obté una llista de butaques reservades i/o comprades directament, i una llista de butaques lliures.

Us demanem que feu els models següents (tots ells mitjançant la notació UML):

- *Model Conceptual.*
- *Model de Casos d'Ús:* diagrama de casos d'ús i especificació dels casos d'ús.
- *Model del Comportament del Sistema:* diagrames de seqüència i contractes de les operacions.
- *Model dels Estats:* diagrama d'estats per objectes i casos d'ús.

7. El president d'un comitè de programa d'un congrés internacional vol construir un sistema software que l'ajudi a controlar les ponències enviades al congrés, les persones que revisaran aquestes ponències i les revisions que aquestes fan.

Cada ponència té un codi, que l'identifica, un títol i està escrita per una o més persones. De tots els autors d'una ponència n'hi ha un que és l'autor principal la resta es consideren autors secundaris. Un cop rebuda, cada ponència s'envia a un o més revisors.

Les persones s'identifiquen pel seu nom. Pot passar que una persona sigui revisora d'una ponència i autora d'altres ponències.

El sistema ha de respondre als cinc esdeveniments següents: donar d'alta un revisor, recepció d'una ponència, assignar una ponència a un revisor, registrar la valoració que un revisor fa d'una ponència i generar un llistat amb les revisions pendents de valoració.

Per donar d'alta un revisor es proporciona el seu nom. El sistema ha de comprovar que no existeix cap altre revisor amb el mateix nom.

Quan es reb una ponència, s'indica el codi de la ponència, el seu títol, l'autor principal i la resta d'autors.

El president del comitè de programa assigna cada ponència a diversos revisors indicant, per cada assignació, el codi de la ponència i el nom del revisor. Un mateix revisor pot tenir assignades diverses ponències. No es pot assignar una ponència a un revisor que sigui autor d'aquesta mateixa ponència.

Al cap d'uns dies, els revisors envien el resultat de la revisió que han fet d'una ponència indicant el codi de la ponència, el nom del revisor i la qualificació que expressa, en una escala de 0 a 10, la valoració global que el revisor fa de la ponència. Com és lògic, un revisor no pot enviar la valoració d'una ponència que no li havia estat assignada. Per altra banda, un revisor no pot enviar dues valoracions d'una mateixa ponència.

En qualsevol moment, el president pot demanar un llistat amb les revisions pendents de valoració. S'ha de mostrar, per cada ponència pendent de valoració per part d'un revisor, el codi de la ponència, el seu títol i el nom del revisor.

Us demanem que feu els models següents (tots ells mitjançant la notació UML):

- *Model Conceptual.*
- *Model de Casos d'Ús:* diagrama de casos d'ús i especificació dels casos d'ús.
- *Model del Comportament del Sistema:* diagrames de seqüència i contractes de les operacions.
- *Model dels Estats:* diagrama d'estats per objectes i casos d'ús.

8. Considereu una federació d'entitats excursionistes que està interessada en un sistema pel control de les expedicions efectuades pels centres excursionistes adscrits a la federació. Una expedició l'efectua un centre excursionista a una certa muntanya, amb una data d'inici i una de finalització. La federació identifica els centres excursionistes pel seu nom i n'enregistra també la seva adreça. Les muntanyes s'identifiquen pel seu nom i se

n'enregistra també la seva alçada. Un centre excursionista pot efectuar diverses expedicions a la mateixa muntanya, amb dates d'inici diferents. A una muntanya s'hi poden fer diverses expedicions, però en una data qualsevol hi pot d'haver un màxim de 5 expedicions.

En una expedició hi participen diverses persones (com a mínim una). Una persona pot participar a més d'una expedició. El sistema guarda informació únicament de les persones (que tenen el dni com a identificador, un nom i una edat) que han participat a alguna expedició que tenia com objectiu una muntanya de més de 8.000 metres.

Algun dels components d'una expedició a una muntanya de més de 8.000 m. pot assolir el cim. En aquest cas, s'enregistrarà aquest fet i també la data en què s'ha fet el cim. Per a simplificar, suposeu que una persona pot assolir el cim una única vegada en una mateixa expedició.

El sistema a desenvolupar únicament ha de consultar les dades referents a centres excursionistes i muntanyes, atès que ja existeix algun altre sistema encarregat de mantenir-les

El sistema ha de permetre efectuar les operacions següents: alta d'expedició, alta de participant a una expedició a una muntanya de més de 8.000 metres, alta de persona que ha fet el cim i emetre un llistat de les dades d'una expedició determinada.

Quan s'efectua una alta d'expedició, cal indicar el nom del centre excursionista, el nom de la muntanya a la que s'efectua l'expedició, la data d'inici i la data de final de l'expedició.

Quan s'efectua una alta d'un participant a una expedició a una muntanya de més de 8.000 metres, s'indica el dni de la persona, el nom del centre excursionista que fa l'expedició, la muntanya i la data d'inici de l'expedició.

Quan s'efectua una alta de persona que ha fet el cim s'indica el nom del centre excursionista que fa l'expedició, la muntanya i la data d'inici de l'expedició, el dni de la persona i la data en què assoleix el cim.

Per demanar el llistat de les dades d'una expedició cal indicar el el nom del centre excursionista que fa l'expedició, la muntanya i la data d'inici de l'expedició. Es mostren el nom del centre, la muntanya, la seva alçada, la data d'inici i final d'expedició i, si és el cas, el nom de totes les persones que han participat en l'expedició i, opcionalment, la data en què han fet el cim.

En totes les operacions anteriors, cal realitzar totes les comprovacions que es considerin adients.

Us demanem que feu els models següents (tots ells mitjançant la notació UML):

- *Model Conceptual.*
- *Model de Casos d'Ús:* diagrama de casos d'ús i especificació dels casos d'ús.
- *Model del Comportament del Sistema:* diagrames de seqüència i contractes de les operacions.
- *Model dels Estats:* diagrama d'estats per objectes i casos d'ús.

9. Una gran organització ha decidit posar en funcionament un sistema que li permeti gestionar els cursos de formació als que assisteixen els seus empleats, ja sigui com a oients o bé mitjançant una matrícula oficial.

Els empleats s'identifiquen per codi, i tenen també un nom i una categoria. Els cursos de formació s'identifiquen per nom, i s'enregistra també qui n'és l'organitzador. En ambdós

casos, el sistema a desenvolupar únicament n'ha de consultar les seves dades, atès que ja existeix un altre sistema que manté tant les dades d'empleat com les de curs de formació.

Un empleat s'inscriu a un curs de formació en una certa data. Cada inscripció fa referència a un curs concret. Un empleat es pot inscriure a tants cursos com vulgui i a un curs s'hi poden escriure diversos empleats. Un empleat s'inscriu una única vegada a un mateix curs de formació. La inscripció d'un empleat a un curs de formació pot ser de dos tipus: com a "oient" o bé com a "matrícula oficial". En el primer cas, el sistema guardarà informació únicament del motiu pel qual s'ha realitzat la inscripció. En el segon, s'enregistraran tant el motiu com el preu de la inscripció.

En el cas de les inscripcions corresponents a matrícules oficials, l'empleat té un màxim de tres convocatòries per tal d'aprovar el curs al qual s'ha inscrit. Per cadascuna de les convocatòries a les que té dret un empleat per a un curs determinat, caldrà enregistrar-ne també la seva nota.

El sistema ha de permetre efectuar les operacions següents: nova inscripció, assignar nota, emetre un llistat d'inscripcions d'un empleat i emetre un llistat de convocatòries.

Quan s'efectua una nova inscripció, cal indicar el codi d'empleat, el nom del curs, la data d'inscripció, el motiu que la justifica, el tipus d'inscripció i, en cas de tractar-se d'una "matrícula oficial", el preu.

Quan s'assigna una nota d'una convocatòria, s'indica el codi d'empleat, el nom del curs i la nota corresponent. El número de la convocatòria s'assignarà, automàticament i de forma correlativa, pel propi sistema i es mostrarà a l'usuari si l'operació s'ha pogut efectuar satisfactòriament.

Les inscripcions d'un empleat es poden consultar indicant-ne el seu codi i, opcionalment, el tipus d'inscripció (si es proporciona només es llisten les inscripcions d'aquell tipus). Es mostren, per cada curs de formació al que s'ha inscrit l'empleat, el nom del curs, el seu organitzador, la data d'inscripció, el motiu i, si s'escau, el preu.

Per demanar el llistat de convocatòries cal indicar el codi de l'empleat i el nom del curs dels quals es demana el llistat. Es mostren el nom i la categoria de l'empleat, el nom del curs i, per cada convocatòria d'aquell empleat en aquell curs, el número de la convocatòria i la seva nota.

En totes les operacions anteriors, cal realitzar totes les comprovacions que esconsiderin adients.

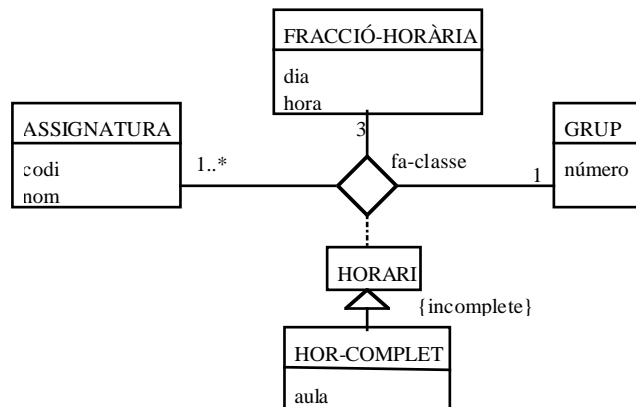
Us demanem que feu els models següents (tots ells mitjançant la notació UML):

- *Model Conceptual.*
- *Model de Casos d'Ús:* diagrama de casos d'ús i especificació dels casos d'ús.
- *Model del Comportament del Sistema:* diagrames de seqüència i contractes de les operacions.
- *Model dels Estats:* diagrama d'estats per objectes i casos d'ús.

10. Considereu una facultat universitària que està interessada en un sistema per la definició dels horaris dels grups de les diferents assignatures. L'horari indica, per cada grup d'una assignatura (per exemple, ES:E – grup 10) quins dies de la setmana hi ha classe i a quina hora (per a simplificar, suposarem que els períodes de classe són sempre d'una hora). A més, també cal guardar la informació de l'aula de cada horari.

El sistema a desenvolupar només ha de consultar les dades d'ASSIGNATURA i les FRACCIONS HORÀRIES, atès que existeix un altre sistema encarregat de donar-los

d'alta. Els diversos números de GRUP es donen d'alta a mesura que es coneixen els horaris d'algun d'ells. El Model Conceptual en UML d'aquest sistema és el següent:



El sistema ha de permetre efectuar les funcionalitats següents: definició-d'horaris-d'un-grup, assignació-d'aula, llistat-d'horaris-sense-aula i llistat-d'horaris de totes les assignatures.

Quan el Cap d'Estudis defineix l'horari d'un grup, indica el codi de l'assignatura, el número de grup a donar d'alta i, per cada fracció horària en què es fa classe d'aquell grup de l'assignatura, el dia i l'hora. És el propi Cap d'Estudis qui comunica aquestes dades al sistema. Feu que la interacció necessària per a portar a terme aquesta funcionalitat requereixi més d'un esdeveniment.

Quan s'efectua una assignació d'aula, s'indica el codi de l'assignatura, el número de grup, el dia, l'hora i l'aula assignada. Aquesta operació és efectuada pels empleats de secretaria a requeriment del Cap d'Estudis.

Per demanar el llistat d'horaris sense aula d'una assignatura determinada, el Cap d'Estudis indica al sistema el codi de l'assignatura i el sistema mostra, per a cada horari d'aquella assignatura sense aula assignada, el número de grup, el dia i la hora.

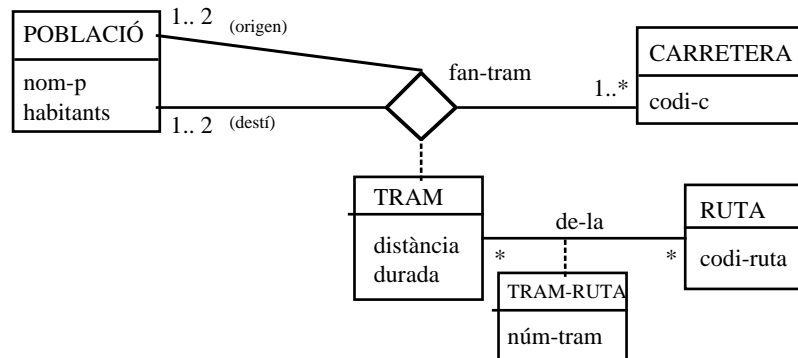
En qualsevol moment, qualsevol usuari d'aquest sistema (inclosos Professors i Alumnes) pot demanar el llistat d'horaris de totes les assignatures. El sistema mostra, per a cada assignatura, el seu codi i, per a cada grup de l'assignatura, el seu número, els dies i hores en què es fa classe i, si es coneix, l'aula.

Us demanem que feu els models següents (tots ells mitjançant la notació UML). En *tots* els casos, cal realitzar *totes* les comprovacions que siguin necessàries. Si n'hi ha, cal indicar les modificacions necessàries a fer en el Model Conceptual de partida.

- *Model de Casos d'Ús*: Diagrama de casos d'ús. Especificació del cas d'ús de l'assignació d'aula.
- *Model del Comportament del Sistema*: tots els diagrames de seqüència i contractes de les operacions corresponents a la definició dels horaris d'un grup i al llistat d'horaris sense aula d'una assignatura.

11. Considereu una empresa de transports que està interessada en un sistema per la definició dels recorreguts de les rutes de distribució dels seus camions. Una ruta està formada per una sèrie de trams consecutius (que es distingeixen pel número de tram dins la ruta). Un tram es defineix per una població d'origen, una de destí i una carretera que uneix les dues poblacions. A més, també cal guardar la informació de la distància i de la durada de recorregut d'un tram.

El sistema a desenvolupar només ha de consultar les dades de POBLACIÓ i de CARRETERA, atès que existeix un altre sistema encarregat de donar-les d'alta. El Model Conceptual en UML d'aquest sistema és el següent:



R.I. Textual:

- La població destí d'un tram de la ruta ha de coincidir amb la població origen del tram següent de la mateixa ruta.
- La població origen i la població destí d'un tram han de ser diferents.

El sistema ha de permetre efectuar les funcionalitats següents: alta-tram, alta-ruta, llistat-de-trams-sense-ruta i llistat-dels-trams-d'una-ruta.

Quan es dona d'alta un tram, s'indica el nom de la població d'origen, el nom de la població de destí, el codi de la carretera, la distància i la durada del tram. Aquesta operació és efectuada pels empleats de l'empresa, a requeriment del Director de Distribució.

Quan el Director de Distribució dóna d'alta una ruta, indica ell mateix al sistema el codi de la ruta i, per cada tram que forma part de la ruta que s'està donant d'alta, les poblacions d'origen i destí del tram, el codi de carretera i el número de tram dins la ruta. Feu que la interacció necessària per a portar a terme aquesta funcionalitat requereixi més d'un esdeveniment.

El llistat dels trams sense ruta assignada s'emet sempre a final de mes i va dirigit al Director de Distribució. El resultat d'aquest llistat inclou, per cada tram que no està assignat a cap ruta, el nom de les poblacions d'origen i destí del tram i el codi de la carretera que les uneix.

En qualsevol moment, els empleats de l'empresa poden demanar el llistat dels trams d'una ruta. El sistema mostra, per a cada tram de la ruta indicada per l'empleat, el nom de la població d'origen i de destí, el codi de la carretera, la distància i la durada del tram i el número de tram dins la ruta.

Us demanem que feu els models següents (tots ells mitjançant la notació UML). En *tots* els casos, cal realitzar *totes* les comprovacions que siguin necessàries. Si n'hi ha, cal indicar les modificacions necessàries a fer en el Model Conceptual de partida.

- *Model de Casos d'Ús:* Diagrama de casos d'ús. Especificació del cas d'ús de l'alta tram.
- *Model del Comportament del Sistema:* Tots els diagrames de seqüència. Contractes de les operacions corresponents l'alta de ruta i al llistat dels trams d'una ruta.

12. Considereu un club d'automobilistes que està interessat en desenvolupar un sistema software per a gestionar assegurances de vehicles. Un vehicle (identificat pel seu número de matrícula i del que se n'enregistra també el model) és comprat per un únic conductor (identificat per número de llicència i del que se'n coneix el nom) en una certa data. Un conductor, pot haver comprat diversos vehicles. El sistema guardarà únicament la informació dels conductors que han comprat algun vehicle.

Tot vehicle comprat ha de contractar una assegurança d'accidents a alguna companyia. Quan es fa un d'aquests contractes, s'enregistra també la data d'inici del mateix. Un vehicle no pot estar contractat a més d'una companyia, mentre que una companyia pot tenir diversos contractes de vehicles diferents. De les companyies d'assegurances ens interessa enregistrar el seu nom i l'adreça (que suposarem única per cada companyia).

Cal enregistrar també la informació dels conductors que no són acceptats per les companyies d'assegurances, conjuntament amb el motiu de la no acceptació. Lògicament, una companyia no podrà tenir assegurances de cotxes comprats per conductors que no són acceptats per la companyia. Un conductor pot no ser acceptat per diverses companyies.

Per simplificar, suposarem que la informació de conductors, vehicles i compres de vehicles és mantinguda per algun sistema extern i, per tant, el sistema a desenvolupar únicament l'haurà de consultar. En canvi, el sistema a desenvolupar ha de permetre efectuar altes de contracte d'assegurança, baixes de contracte, enregistrar conductors no acceptats per les companyies i llistat d'assegurances d'una companyia.

Quan es vol fer una alta de contracte d'assegurança cal indicar la matrícula del cotxe, el nom de la companyia on es fa l'assegurança i la data en què es fa efectiu el contracte. Quan es vol donar de baixa un contracte, s'indica la matrícula del cotxe i el nom de la companyia. En ambdós casos, són els empleats qui comuniquen les dades al sistema, a requeriment d'algun conductor.

Les companyies d'assegurances envien al club d'automobilistes els llistats de conductors que no són acceptats per la companyia. Algun empleat del club, quan així ho consideri convenient, comunicarà al sistema el llistat de tots els conductors no acceptats per les diverses companyies. S'indicarà, per cada companyia, el número de llicència de cadascun dels conductors no acceptats. Feu que la interacció necessària per a portar a terme aquesta funcionalitat requereixi més d'un esdeveniment.

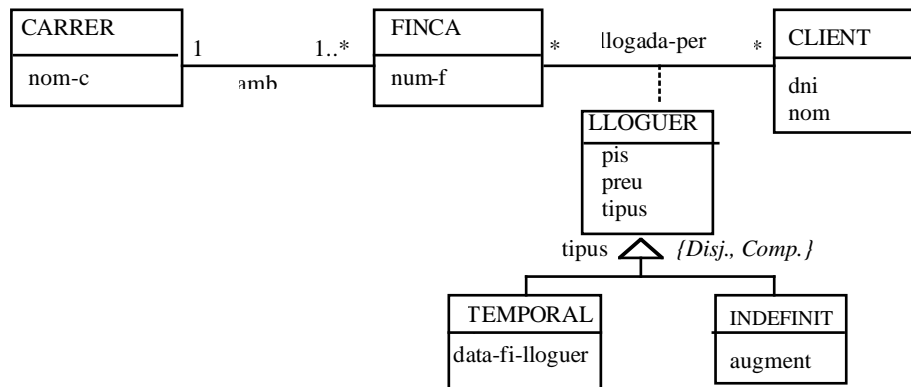
Per demanar el llistat dels contractes d'una companyia, el Director del club d'automobilistes indica directament al sistema el nom de la companyia i el sistema mostra, per a cada contracte d'assegurances d'aquella companyia, la matrícula del cotxe, el número de llicència del seu propietari i la data de contractació de l'assegurança.

En tots els casos, cal realitzar totes les comprovacions que es considerin adients.

Us demanem que feu els models següents (tots ells mitjançant la notació UML):

- *Model Conceptual*, que ha d'incloure totes les restriccions textuais necessàries i una instanciació del model que demostrï la seva validesa.
- *Model de Casos d'Ús*:
 - Diagrama de casos d'ús.
 - Especificació del cas d'ús de l'alta de contracte, que ha d'incloure totes les comprovacions a realitzar.
- *Model del Comportament del Sistema*:
 - Diagrames de seqüència de "enregistrar conductors no acceptats" i de "llistat de contractes d'una companyia".
 - Contractes de les operacions corresponents a aquests diagrames de seqüència.

13. Considereu una agència immobiliària d'un poble turístic de l'Alt Empordà que està interessada en un sistema per gestionar els lloguers de les finques que posseeix al poble. El sistema guarda informació dels carrers (identificats per nom) on té situades les finques (identificades per número dins el carrer) i dels lloguers de les finques que fan els clients de l'agència (identificats per dni). Els lloguers fan referència a un pis de la finca (1er 1a, 1er 2a, etc) i poden ser temporals o bé indefinits. El sistema ha de guardar també informació del preu del lloguer i d'atributs específics per a cada tipus de lloguer. A més, el sistema només ha de consultar les dades de CLIENT atès que hi ha un altre sistema encarregat de donar-les d'alta. El Model Conceptual en UML d'aquest sistema és el següent:



R.I.textuals:

- Claus de les classes no associatives: (Carrer, nom-c), (Client, dni).
- Un carrer no pot tenir dues finques amb el mateix número.

El sistema ha de permetre efectuar les funcionalitats següents: alta-carrer, alta-lloguers-finca, baixa-lloguers i llistat-dels-lloguers-amb-augment-elevat.

Quan es dona d'alta un carrer, s'indica el nom del carrer i, per cada finca d'aquell carrer, el seu número de finca. Aquesta operació és efectuada pels empleats de l'agència, a requeriment de la Directora de l'agència.

Quan el Responsable de Lloguers dona d'alta tots els lloguers d'una finca, indica ell mateix al sistema el nom del carrer, el número de la finca i, per cada lloguer d'aquella finca que es dona d'alta, el dni del client, el pis llogat, el preu, el tipus de lloguer (temporal o indefinit) i la data-fi-lloguer o l'augment anual establert, segons s'escaigui. Feu que la interacció necessària per a portar a terme aquesta funcionalitat requereixi més d'un esdeveniment.

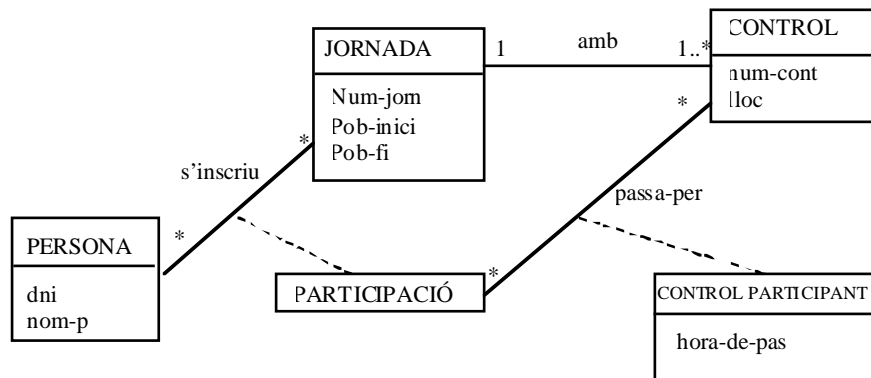
A final del dia, es donen de baixa automàticament tots els lloguers que finalitzen en aquesta data. A més, el sistema genera un llistat que inclou, per cada lloguer donat de baixa, el nom del carrer, el número de finca, el dni del client i el pis.

En qualsevol moment, tots els empleats de l'empresa poden demanar el llistat dels lloguers amb augment elevat. El sistema mostra, per cada lloguer indefinit amb un augment superior al 5%, el nom del carrer, el número de finca i el nom del client que té el pis llogat.

Us demanem que feu els models següents (tots ells mitjançant la notació UML). En *tots* els casos, cal realitzar *totes* les comprovacions que siguin necessàries. Si n'hi ha, cal indicar les modificacions necessàries a fer en el Model Conceptual de partida.

- Model de Casos d'Ús:* Diagrama de casos d'ús. Especificació del cas d'ús d'alta carrer.
- Model del Comportament del Sistema:* Tots els diagrames de seqüència. Contractes de les operacions corresponents l'alta de lloguers d'una finca i al llistat dels lloguers amb augments elevats.

14. Considereu un centre excursionista que ha decidit organitzar la travessa dels Pirineus a peu, des de Biarritz al Cap de Creus. Aquesta travessa consta de diverses jornades (identificades per un número) que es recorren entre una població d'inici i una de final. Tota jornada té una sèrie de controls (identificats per un número dins la jornada) per garantir que cap participant no es perd. Les persones es poden inscriure a una jornada de la travessa. El sistema emmagatzema també les hores de pas de tots els participants per a cadascun dels controls que passen. A més, el sistema només ha de consultar les dades de PERSONA atès que hi ha un altre sistema encarregat de donar-les d'alta. El Model Conceptual en UML d'aquest sistema és el següent:



R.I.textuals:

- Claus de les classes no associatives: (Jornada, num-jornada), (Persona, dni).
- Una jornada no pot tenir dos controls amb el mateix num-cont.
- La població final d'una jornada i la d'inici de la jornada següent han de coincidir
- La persona que participa a "control-participant" ha d'estar inscrita a la jornada d'aquell control

El sistema ha de permetre efectuar les funcionalitats següents: alta-jornada, alta-participant, assignar-passos-per-control i llistat-dels-controls-d'una-jornada.

Quan es dona d'alta una jornada, s'indica el número de la jornada, la població d'inici, la població de final i, per cada control d'aquella jornada que s'està donant d'alta en aquell moment, el seu número de control i el lloc. Les jornades no es donen d'alta totes de cop, sinó que es van comunicant al sistema a mesura que es pren la decisió del seu recorregut concret. A més, ja no es poden donar d'alta jornades un cop es coneixen els primers control de pas de la travessa. L'alta de jornades és efectuada pels empleats de l'agència, a requeriment de la Responsable de Recorregut. Cal fer que la interacció necessària per a portar a terme aquesta funcionalitat requereixi més d'un esdeveniment.

Quan una persona vol participar en una jornada de la travessa, ho fa saber a l'empleat que és qui comunica aquesta informació al sistema. N'hi ha prou indicant el dni de la persona i el número de jornada.

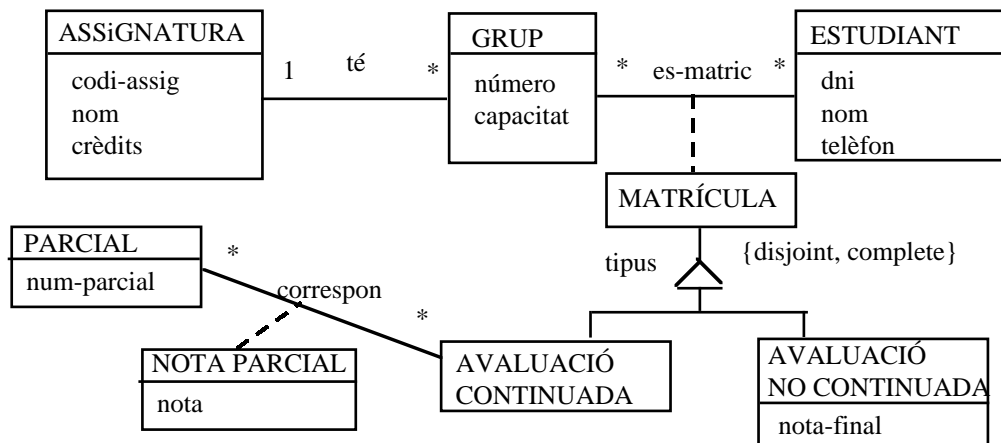
Quan el Controlador en Cap dona d'alta tots els controls dels participants d'una jornada, indica ell mateix al sistema el número de jornada i, per cada persona que ha participat en aquella jornada el número de control i l'hora de pas de tots els controls perquè ha passat. La interacció necessària per portar a terme aquesta funcionalitat ha de requerir també més d'un esdeveniment.

En qualsevol moment, el Controlador en Cap pot demanar un llistat dels controls d'una jornada. Donada una jornada, que és indicada per ell mateix al sistema, aquest mostra el dni del participant, el número de control i l'hora de pas de tots els controls de participant.

Us demanem que feu els models següents (tots ells mitjançant la notació UML). En *tots* els casos, cal realitzar *totes* les comprovacions que siguin necessàries. Si n'hi ha, cal indicar les modificacions necessàries a fer en el Model Conceptual de partida.

- *Model de Casos d'Ús*: Diagrama de casos d'ús. Especificació del cas d'ús d'alta jornada.
- *Model del Comportament del Sistema*: Tots els diagrames de seqüència. Contractes de les operacions corresponents l'alta de controls i al llistat de controls.

15. Considereu una facultat universitària que està interessada en un sistema per la definició dels grups de les diferents assignatures. Una assignatura pot tenir diversos grups (per exemple, grups número 10 i 20 de l'assignatura FBD; grups número 10, 20 i 30 de l'assignatura ES:E, etc.). Una matrícula es defineix per un estudiant i per un grup on l'estudiant es matricula. Les matrícules poden ser de dos tipus: d'avaluació continuada i d'avaluació no continuada. Per les matrícules d'avaluació no continuada s'enregistra la nota final de l'estudiant. Per les d'avaluació continuada s'entregistren les diverses notes parcials de l'estudiant. El Model Conceptual en UML d'aquest sistema és el següent:



R.I. textuals:

- Claus de les classes no associatives: (Assignatura, codi-assig), (Estudiant, dni), (Parcial, num-parcial)
- No hi pot haver cap assignatura que tingui dos grups amb el mateix número
- Un estudiant no pot matricular-se de més d'un grup de la mateixa assignatura

El sistema a desenvolupar només ha de consultar les dades d'ASSIGNATURA, ESTUDIANT i PARCIAL, atès que existeix un altre sistema encarregat de donar-los d'alta. El sistema ha de permetre efectuar entre d'altres les funcionalitats següents: alta-grups-d'una-assignatura, matrícula-estudiant, assignació-notes-parcials, consulta-d'estudiants-amb-notes-finals-aprovades.

Quan el Cap d'Estudis vol definir els grups d'una assignatura, indica el codi de l'assignatura i, per cada grup a donar d'alta, el número de grup i la capacitat del grup. És el propi Cap d'Estudis qui comunica aquestes dades al sistema. Feu que la interacció necessària per a portar a terme aquesta funcionalitat requereixi més d'un esdeveniment.

Quan un estudiant es matricula en un grup, indica el codi de l'assignatura, el número de grup, el seu dni i el tipus de matrícula (aval. cont. o no). Aquesta operació és efectuada pel propi estudiant.

Quan un professor vol fer una assignació de notes parcials a estudiants amb matrícula d'avaluació continuada en un determinat grup, s'indica el codi de l'assignatura, el número de grup i el número de parcial i, per cada estudiant que té nota parcial, el dni de l'estudiant i la nota. Aquesta operació és efectuada per l'empleat de secretaria a requeriment del professor. Feu que la interacció necessària per a portar a terme aquesta funcionalitat requereixi més d'un esdeveniment.

El Cap d'Estudis en qualsevol moment pot consultar els estudiants amb notes finals aprovades. El sistema mostra, per cada matrícula d'avaluació no continuada amb nota final més gran o igual que 5, el dni de l'estudiant, el codi de l'assignatura i el número de grup.

Us demanem que feu els models següents en UML. Cal realitzar totes les comprovacions necessàries. Si n'hi ha, cal indicar els afegits necessaris al Model Conceptual de partida.

- *Model de Casos d'ús*: Diagrama de casos d'ús (per les funcionalitats especificades). Especificació del cas d'ús de l'alta de grups d'una assignatura.
- *Model del Comportament del Sistema*: Tots els diagrames de seqüència. Contractes de les operacions corresponents a l'assignació de notes parcials i a la consulta d'estudiants amb notes finals aprovades.

16. Considereu un centre de gestió electoral que està interessat en desenvolupar un sistema software per a gestionar les candidatures i resultats de les successives eleccions municipals.

Una candidatura s'identifica mitjançant el partit que la presenta, el municipi al qual correspon i les eleccions per a les que s'ha confeccionat. A cada candidatura s'hi presenta un conjunt de persones candidates (com a mínim 3) que poden presentar-se com a candidats independents o no. Una mateixa persona no pot estar a més d'una candidatura per elecció. Cal que el sistema tingui enregistrades quines són les persones candidates de cada candidatura i també si es presenten com a independents o no (tipus de candidat). Les persones s'identifiquen pel seu dni i també vol enregistrar-se el seu nom.

Cadascuna de les eleccions municipals s'identifica pel seu any de realització, els municipis s'identifiquen pel seu nom i els partits per les seves sigles. A més, el sistema ha d'enregistrar el nom de cada partit i la comarca de cada municipi.

Un cop s'han efectuat unes eleccions, per les candidatures que hagin obtingut representació, el sistema haurà de permetre tenir enregistrat el número total de representants obtingut i quins són els candidats (de la candidatura) que queden escollits com a representants. Per les candidatures que no hagin obtingut representació, el sistema haurà de permetre enregistrar el número de vots obtinguts per la candidatura.

Per simplificar, suposarem que la informació d'eleccions, municipis, partits i persones és mantinguda per algun sistema extern i, per tant, el sistema a desenvolupar únicament l'haurà de consultar. En canvi, el sistema a desenvolupar ha de permetre efectuar altes de candidatures, entrades de resultats de candidatures sense representació, entrades de resultats de candidatures amb representació i llistats de candidats escollits d'una candidatura.

Quan es vol donar d'alta una candidatura cal indicar les sigles del partit, el nom del municipi, l'any de les eleccions i, per cada candidat, el dni, el nom i el seu tipus (independent o no). Una candidatura es dona d'alta tota de cop i, una vegada feta l'alta, no s'hi poden afegir candidats. A més, no es poden donar d'alta noves candidatures per eleccions que ja tinguin algun resultat entrat. L'alta de candidatures és efectuada pels empleats del centre a requeriment dels interlocutors dels partits. Feu que la interacció necessària per a portar a terme aquesta funcionalitat requereixi més d'un esdeveniment.

Quan un empleat del centre vol entrar els resultats d'una candidatura sense representació, indica ell mateix al sistema les sigles del partit, el nom del municipi, l'any de les eleccions i el número de vots obtingut per la candidatura.

Quan un empleat del centre vol entrar els resultats d'una candidatura amb representació, indica ell mateix al sistema les sigles del partit, el nom del municipi i l'any de les eleccions. A més, per cada candidat que queda escollit com a representant, indica el seu dni. Els resultats d'una candidatura amb representació s'entren tots de cop. Feu que la interacció necessària per a portar a terme aquesta funcionalitat requereixi més d'un esdeveniment.

En qualsevol moment, els interlocutors dels partits poden demanar un llistat de candidats escollits d'una candidatura. Aleshores, un empleat del centre indica les sigles del partit, el nom del municipi i l'any de les eleccions de la candidatura. Com a resposta el sistema mostra, per cada candidat de la candidatura que hagi quedat escollit com a representant, el dni, el nom i el tipus de candidat (independent o no). Evidentment, per poder demanar aquest llistat, cal que la candidatura tingui representació.

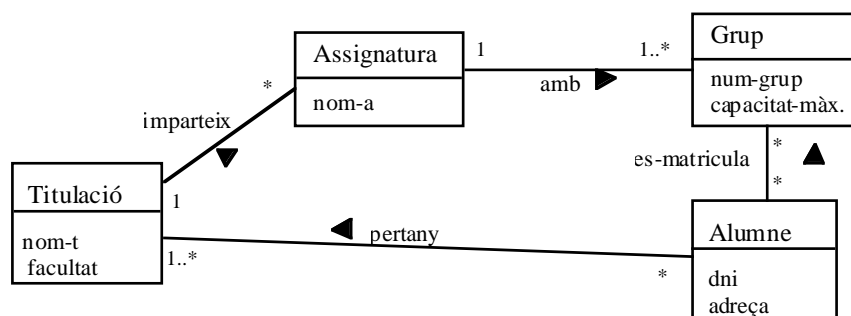
Us demanem que feu els models següents (tots ells mitjançant la notació UML).

- *Model Conceptual*, que ha d'incloure totes les restriccions textuais i atributs derivats necessaris (en narrativa, no en OCL) i una instanciació del model que demostrï la seva validesa.
- *Model de Casos d'ús*: Diagrama de casos d'ús. Especificació del cas d'ús de l'entrada de resultats d'una candidatura amb representació (que inclogui totes les comprovacions a realitzar)

- *Model del Comportament del Sistema:* Diagrames de seqüència de “alta candidatura” i “llistat de candidats escollits d’una candidatura”. Contractes de les operacions corresponents a aquests diagrames de seqüència (que incloguin totes les comprovacions a realitzar). Si n’hi ha, indiqueu els afegits necessaris al Model Conceptual de partida. Expressen les sortides amb l’ajuda del llenguatge OCL.

17. Considereu una Universitat que està interessada en un sistema per a la gestió de la matrícula dels seus alumnes a les diferents titulacions que imparteix. Per simplificar, suposarem que interessa emmagatzemar únicament la informació referent a un sol curs acadèmic.

La Universitat ofereix diverses titulacions. Una titulació imparteix un conjunt d’assignatures, cadascuna de les quals consta de diversos grups de l’assignatura. Un alumne pertany a una o més titulacions de la Universitat i es pot matricular de grups de les assignatures d’aquestes titulacions. El Model Conceptual en UML d’aquest sistema és el següent:



R.I.textuals:

- Claus de les classes no associatives: (Titulació, nom-t), (Alumne, dni).
- Una assignatura no pot tenir dos grups amb el mateix num-grup.
- Una titulació no pot tenir dues assignatures amb el mateix nom-a.
- Un alumne no es pot matricular a dos grups diferents d’una mateixa assignatura
- Un alumne només es pot matricular d’assignatures que són impartides en titulacions a les que ell pertany.

El sistema a desenvolupar no ha de donar d’alta les dades de Titulació, Alumne, Pertany i Es-matricula, atès que hi ha un altre sistema encarregat de fer-ho. El sistema ha de permetre efectuar, entre d’altres, les funcionalitats següents: alta-d’assignatures-impartides i informació-alumne.

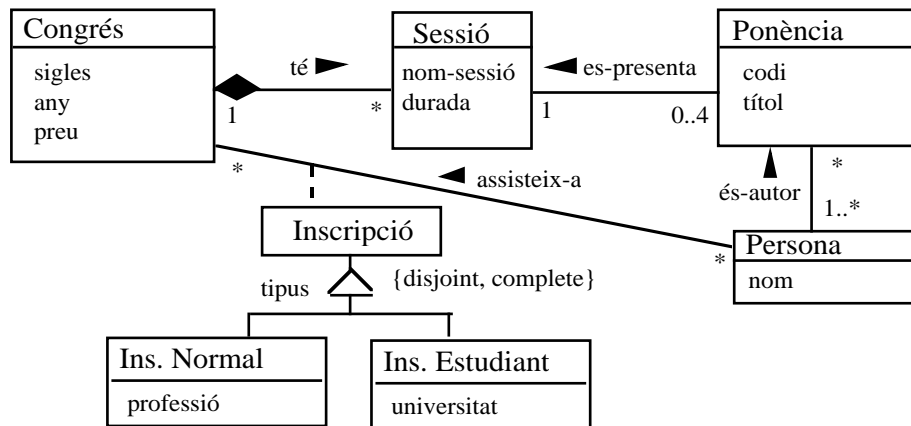
Quan la Cap d’Estudis d’una titulació defineix les assignatures que s’imparteixen a aquella titulació indica el nom de la titulació i, per cada assignatura, el nom de l’assignatura i la informació referent a tots els grups de l’assignatura (o sigui, per cada grup de l’assignatura, el número de grup i la seva capacitat màxima). Com a conseqüència d’aquesta interacció, la pròpia Cap d’Estudis rep un llistat en el qual es mostra, per tot grup amb una capacitat màxima superior a 80 alumnes, el número de grup i el nom de la seva assignatura. Per una titulació, l’alta de les assignatures impartides es realitza tota de cop. Feu que la interacció necessària per a portar a terme aquesta funcionalitat requereixi més d’un esdeveniment.

Quan ho desitgin, els alumnes poden demanar la informació dels estudis que estan fent a la Universitat. Per a fer-ho indiquen ells mateixos al sistema el seu dni i aquest mostra, per cada titulació a la que pertany l’estudiant el nom de la titulació més el nom de l’assignatura i el número de grup de tots els grups dels que l’estudiant s’ha matriculat en aquesta titulació.

Us demanem que feu els models següents (tots ells mitjançant la notació UML). En *tots* els casos, cal realitzar *totes* les comprovacions que siguin necessàries.

- *Model de Casos d’Ús:* Diagrama de casos d’ús (de les funcionalitats especificades).
- *Model del Comportament del Sistema:* diagrames de seqüència i contractes de les operacions corresponents a l’alta-d’assignatures-impartides i a informació-alumne. Expressen les sortides de les operacions amb l’ajut del llenguatge OCL.

18. Considereu una empresa que es dedica a l'organització de congressos i que està interessada en un sistema per a la gestió de les sessions, de les ponències i dels assistents als diversos congressos que organitza. Un congrés s'estructura en sessions. Cada sessió d'un congrés es dedica a la presentació de diverses ponències. Les ponències tenen diversos autors. Les persones que volen assistir a un congrés s'hi han d'inscriure. Les inscripcions poden ser de dos tipus: de tipus "normal" o de tipus "estudiant". Per les inscripcions d'estudiant s'enregistra el nom de la universitat on està estudiant la persona inscrita en el moment de la inscripció. Per les inscripcions normals s'enregistra la professió de la persona inscrita en el moment de la inscripció. El Model Conceptual en UML d'aquest sistema és el següent:



R.I. textuals:

- Claus de classes no associatives: (Congrés: sigles, any), (Ponència: codi), (Persona: nom)
- No hi pot haver cap congrés que tingui dos sessions amb el mateix nom de sessió.

El sistema a desenvolupar no ha de donar d'alta les dades de les inscripcions, atès que existeix un altre sistema encarregat de fer-ho. El sistema ha de permetre efectuar entre d'altres les funcionalitats següents: alta-de-congrés-i-sessions, alta-de-ponències-d'una-sessió i consulta-d'estudiants-i-autors.

Quan el President del Comitè Organitzador d'un congrés vol donar d'alta el congrés i les seves sessions, indica les sigles, any i preu del congrés i, per a cada sessió del congrés, el nom de la sessió i la seva durada. És el propi President del Comitè Organitzador qui comunica aquestes dades al sistema. Finalment, el sistema mostra el nombre total de sessions donades d'alta pel congrés. Feu que la interacció necessària per a portar a terme aquesta funcionalitat requereixi més d'un esdeveniment.

Quan el President del Comitè de Programa d'un congrés vol donar d'alta les ponències d'una sessió del congrés, s'indiquen les sigles i any del congrés i també el nom de la sessió. A més, per a cada ponència de la sessió, s'indiquen el codi i títol de la ponència juntament amb els noms de tots els autors de la ponència. Cal considerar que alguns d'aquests autors existiran a la base d'informació mentre que d'altres no. També cal tenir en compte que no es poden donar d'alta ponències a sessions de congressos que ja tenen alguna persona inscrita. Aquesta alta és efectuada per un administratiu de l'empresa a requeriment del President del Comitè de Programa. Feu que la interacció necessària per a portar a terme aquesta funcionalitat requereixi més d'un esdeveniment.

Finalment, quan una persona qualsevol indica que vol fer una consulta d'estudiants i autors, el sistema mostra els noms de les persones que assisteixen amb inscripció d'estudiant a algun congrés i que, alhora, són autors d'alguna ponència que es presenta en alguna sessió del mateix congrés.

Us demanem que feu els models següents (tots ells mitjançant la notació UML).

- *Model de Casos d'ús*: Diagrama de casos d'ús (per les funcionalitats especificades).
- *Model del Comportament del Sistema*: Diagrames de seqüència i contractes de les operacions corresponents a l'alta de congrés i sessions, a l'alta de les ponències d'una sessió i a la consulta d'estudiants i autors. Expressau les sortides de les operacions amb l'ajut del llenguatge OCL.

Teoria

1. Model del Comportament en UML:

- a) És possible en UML que l'especificació del contracte d'una operació no tingui ni preconditionió ni postcondició, però en canvi si que tingui sortida?
- b) És possible en UML que l'especificació del contracte d'una operació no tingui preconditionió, però en canvi si que tingui postcondició?

En ambdòs casos, justifica breument la teva resposta i, si ho creus convenient, il·lustra-la mitjançant un exemple.

- ### **2.**
- En un diagrama d'estats d'UML, té sentit especificar una transició que porti d'un estat a ell mateix com a conseqüència d'un esdeveniment determinat? Justifica la teva resposta i, en cas afirmatiu, posa'n un exemple que il·lustri el perquè.