

Introducció a l'Enginyeria del Software (IES)

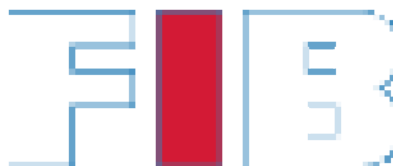
Domain Model:

Exercicis de traducció de l'esquema d'especificació al de disseny



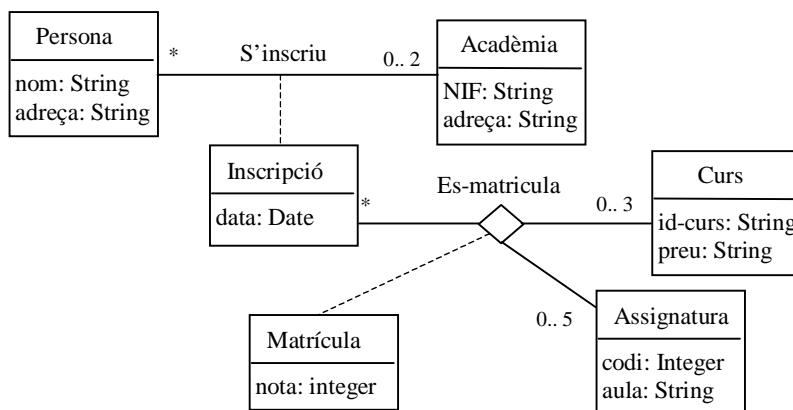
Departament d'Enginyeria de Serveis
i Sistemes d'Informació

UNIVERSITAT POLITÈCNICA DE CATALUNYA



1. Considereu una empresa que gestiona diverses acadèmies que ofereixen cursos de formació i que està interessada en el disseny d'un sistema software de suport a la seva activitat. L'especificació en UML d'aquest sistema és la següent:

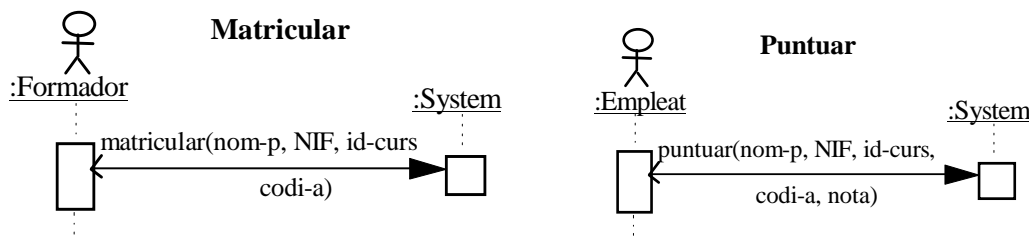
Esquema Conceptual:



R.I. Textual:

- Claus classes no associatives: (Persona, nom); (Acadèmia, NIF); (Curs, id-curs); (Assig., codi)
- Una inscripció no pot tenir més de 25 matrícules
- No es pot suspendre una assignatura un cop ja ha estat aprovada.

Diagrames de seqüència del sistema i contractes de les operacions:



Contracte de l'operació *matricular*:

- Operació:** matricular (nom-p: String, NIF: String, id-curs: String, codi-a: String)
- Semàntica:** Matricular a un alumne inscrit a una acadèmia d'una assignatura en un curs acadèmic determinat
- Precondicions:**
1. Existeix la inscripció de la persona en aquella acadèmia.
 2. Existeixen el curs i l'assignatura.
- Postcondicions:**
- Es crea una instància de l'associació matrícula definida pels objectes corresponents.

Contracte de l'operació *puntuar*:

- Operació:** puntuar (nom-p: String, NIF: String, id-curs: String, codi-a: String, nota: Integer)
- Semàntica:** Assignar la nota a un alumne corresponent a la matrícula
- Precondicions:**
- Existeix una matrícula d'aquella persona, en aquella acadèmia, per aquell curs, d'aquella assignatura.
- Postcondicions:**

S'assigna la nota corresponent, a la matrícula definida pels paràmetres de l'operació.

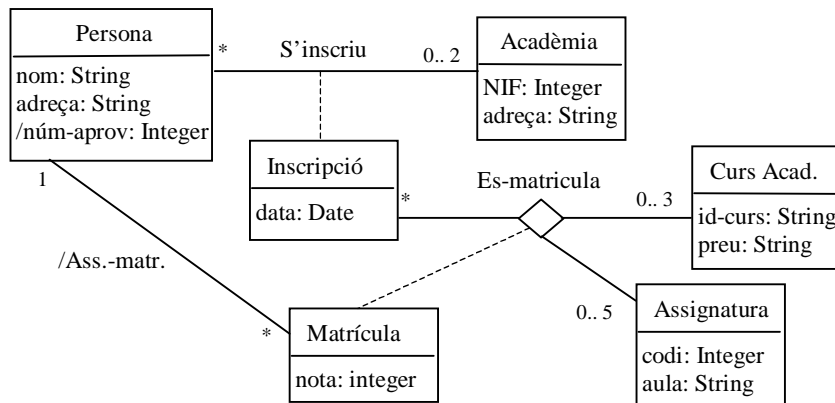
Observació: fixeu-vos que, tal i com està definida, l'operació puntuar permet assignar tants cops com es vulgui la nota d'una matrícula.

Es demana:

- Diagrama de classes de disseny obtingut a partir de l'esquema conceptual de les dades.
- Modificació dels contractes de les operacions, si s'escau, com a conseqüència de la traducció de l'esquema d'especificació al de disseny.

2. Considereu una altra vegada el sistema software de l'exercici anterior, però on s'ha modificat l'esquema conceptual d'especificació per incloure-hi informació derivada. Concretament, s'ha afegit un atribut derivat *núm-aprov* que permet saber el nombre d'assignatures aprovades per un alumne i l'associació derivada *Ass.-matr.* entre les persones i les assignatures que tenen matriculades.

El nou esquema conceptual d'especificació és el següent:



R.I. Textual:

- Claus classes no associatives: (Persona, nom); (Acadèmia, NIF); (Curs, id-curs); (Assig., codi)
- Una inscripció no pot tenir més de 25 matrícules

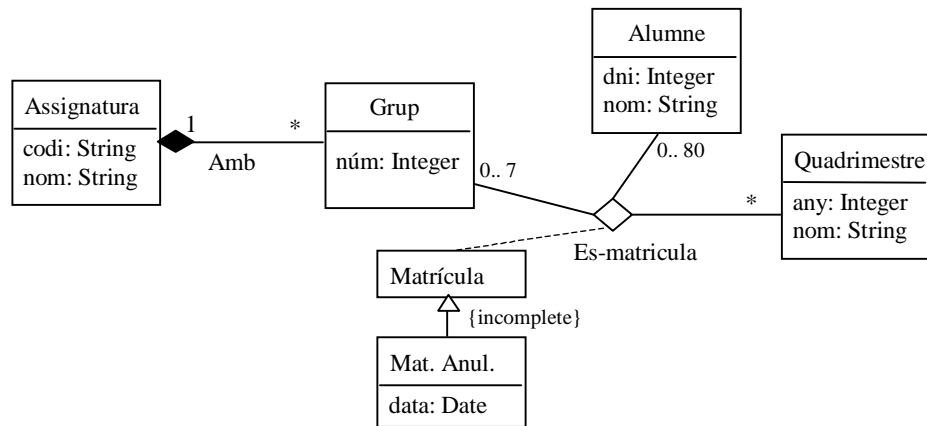
Informació derivada:

- *núm-aprov.* = número d'assignatures aprovades (*nota* ≥ 5) d'aquella persona
- *Ass-matr.*: un alumne s'ha matriculat a un conjunt d'assignatures

Es demana:

- Suposant que *núm-aprov* i *Ass.-matr.* són materialitzats:
 - Diagrama de classes de disseny resultant que inclogui la definició de les operacions necessàries per a tractar aquesta informació (assignades a les classes corresponents).
 - Modificacions necessàries, si n'hi ha, dels contractes de les operacions.
- Feu el mateix que en el cas anterior, suposant que *núm-aprov.* i *Ass.matr.* són calculats.

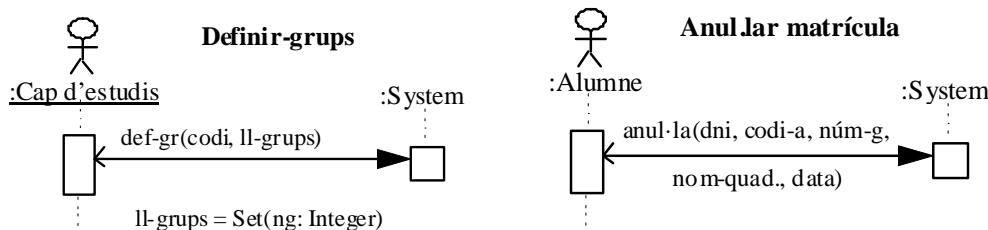
3. Considereu una facultat universitària que està interessada en el disseny d'un sistema software de suport a les matrícules d'assignatures que fan els seus estudiants. L'especificació en UML d'aquest sistema és la següent:



R.I. Textuals:

- Claus classes no associatives: (Assignatura, codi); (Quadrimestre, nom); (Persona, dni)
- Una assignatura no pot tenir dos grups amb el mateix número de grup.
- En un quadrimestre determinat, un alumne no es pot matricular a dos grups d'una mateixa assignatura

Diagrames de seqüència del sistema i contractes de les operacions:



Contracte de l'operació *definir grups*:

Operació: def-gr (codi: String, ll-grups: Set(ng:Integer))

Semàntica: Definir tots els grups de l'assignatura identificada per codi.

Precondicions:

- Existeix l' assignatura identificada per codi.
- L' assignatura identificada per codi no té cap grup definit.

Postcondicions:

- Donar d'alta un objecte grup amb núm-grup
- Crear l'associació entre l'assignatura i els grups que apareixen al paràmetre ll-grups..

Contracte de l'operació *anul·lar matrícula*:

Operació: anul·la (dni:Integer, codi: String, núm-g:Integer, nom-quad: String, data: Date)

Semàntica: Anul·lar la matrícula d'un estudiant d'un grup en un quadrimestre.

Precondicions:

- Existeix la matrícula identificada per l'estudiant, el grup de l'assignatura i el quadrimestre i no està anul·lada.

Postcondicions:

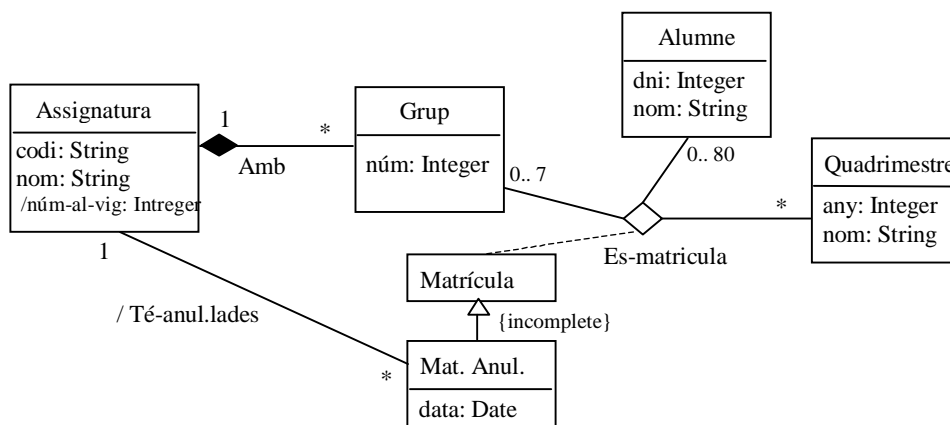
- S'especialitza la matrícula anterior en Matr. Anul., amb l'atribut data corresponent.

Es demana:

- Diagrama de classes de disseny obtingut a partir de l'esquema conceptual de les dades.
- Modificació dels contractes de les operacions, si s'escau, com a conseqüència de la traducció de l'esquema d'especificació al de disseny.

4. Considereu una altra vegada el sistema software de l'exercici anterior, però on s'ha modificat l'esquema conceptual d'especificació per incloure-hi informació derivada. Concretament, s'ha afegit un atribut derivat *núm-alum-vig* que permet saber el nombre d'alumnes d'una assignatura amb matrícula no anul.lada i l'associació derivada *Té-anul.lades* entre les assignatures i les seves matrícules anul.lades.

El nou esquema conceptual d'especificació és el següent:



R.I. Textuals:

- Claus classes no associatives: (Assignatura, codi); (Quadrimestre, nom); (Persona, dni)
- Una assignatura no pot tenir dos grups amb el mateix número de grup.
- En un quadrimestre determinat, un alumne no es pot matricular a dos grups d'una mateixa assignatura

Informació derivada:

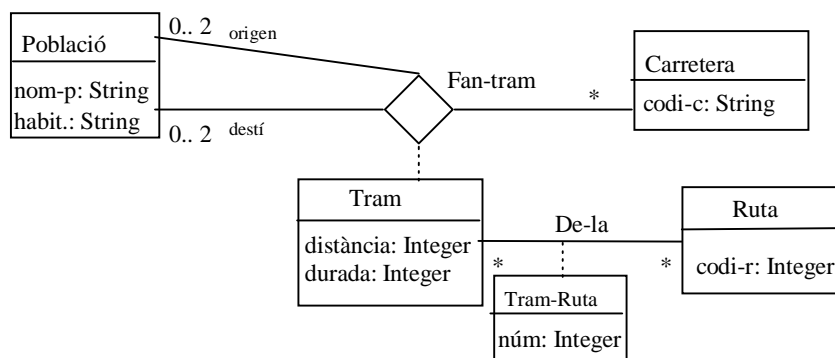
- *núm-al-vig* = nombre d'alumnes de l'assignatura amb matrícula no anul.lada
- *Té-anul.lades*: una assignatura té anul.lades un conjunt de matrícules

Es demana:

- Suposant que *núm-al-vig.* és materialitzat i *Té-anul.lades* és calculada:
 - Diagrama de classes de disseny resultant que inclogui la definició de les operacions necessàries per a tractar aquesta informació (assignades a les classes corresponents).
 - Modificacions necessàries, si n'hi ha, dels contractes de les operacions.
- Feu el mateix que en el cas anterior, suposant que *núm-al.-vig.* és calculat i *Té-anul.lada.* és materialitzada.

5. Considereu una empresa de transports que està interessada en un sistema per la definició dels recorreguts de les rutes de distribució dels seus camions. L'especificació en UML d'aquest sistema és la següent:

Esquema Conceptual:

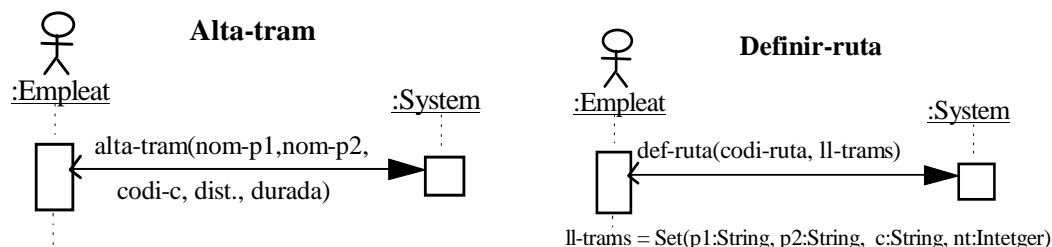


R.I. Textual:

- Claus classes no associatives: (Població, nom-p); (Carretera, codi-c); (Ruta, codi-r)
- La població destí d'un tram d'una ruta ha de coincidir amb la població origen del tram següent (que té $\text{núm} = \text{núm} + 1$) de la ruta, si el tram següent està definit.
- La població origen i la població destí d'un tram han de ser diferents.
- Una carretera no pot tenir més de 10 trams diferents

Observació: fixeu-vos que, tal i com està definida l'associació tram, donada una carretera, el sistema emmagatzema els trams en les dues direccions possibles. O sigui, el tram (Figueres, La Jonquera, N-II) és diferent de (La Jonquera, Figueres, N-II).

Diagrames de seqüència del sistema i contractes de les operacions:



Contracte de l'operació alta-tram:

Operació: alta-tram (nom-p1: String, nom-p2: String, codi-c: String, dist: Integer, durada: Integer)

Semàntica: Donar d'alta un tram entre dues poblacions d'una carretera

Precondicions:

1. Existeixen les dues poblacions i la carretera.

Postcondicions:

Es crea una instància de l'associació fan-tram amb els atributs corresponents on nom-p1 és l'origen i nom-p2 el destí.

Contracte de l'operació def-ruta:

Operació: def-ruta (codi-ruta: Integer, ll-trams: Set(p1:String, p2:String, c:String, nt: Integer))

Semàntica: Donar d'alta un ruta i els trams corresponents

Precondicions:

1. Existeix la ruta.
2. Per cada element (nom-p1, nom-p2, codi-c, núm) de ll-trams, existeix un tram definit per nom-p1, nom-p2 i codi-c.

Postcondicions:

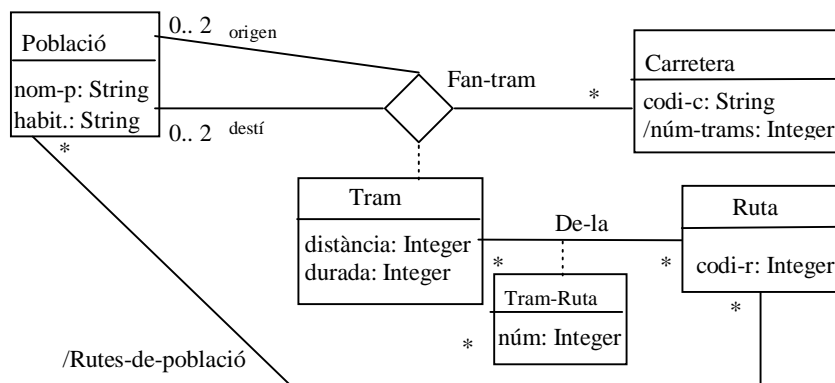
- Per cada element (nom-p1, nom-p2, codi-c, núm) de ll-trams, es crea una instància de l'associació De-la amb l'atribut núm corresponent.

Es demana:

- a) Diagrama de classes de disseny obtingut a partir de l'esquema conceptual de les dades.
- b) Modificació dels contractes de les operacions, si s'escau, com a conseqüència de la traducció de l'esquema d'especificació al de disseny.

6. Considereu una altra vegada el sistema de l'exercici anterior, però on s'ha modificat lleugerament l'esquema conceptual d'especificació per incloure-hi informació derivada. Concretament, s'hi ha afegit un atribut derivat *núm-trams* que permet saber el nombre de trams que té una determinada carretera i l'associació derivada *Rutes-de-població* entre les poblacions i les rutes que hi passen.

El nou esquema conceptual d'especificació és el següent:



R.I. Textual:

- Claus classes no associatives: (Població, nom-p); (Carretera, codi-c); (Ruta, codi-r)
- La població destí d'un tram d'una ruta ha de coincidir amb la població origen del tram següent (que té $núm = núm + 1$) de la ruta, si el tram següent està definit.
- La població origen i la població destí d'un tram han de ser diferents.
- Una carretera no pot tenir més de 10 trams diferents
-- Una ruta no pot tenir dos trams amb el mateix número de tram-ruta

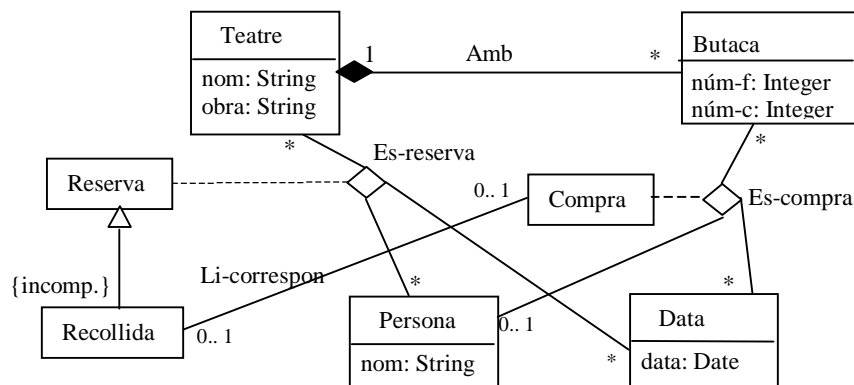
Informació derivada:

- $núm-trams$ = suma dels trams (nombre d'ocurrències de tram) que estan definits per aquella carretera
- Rutes-de-població: per una població hi passen un conjunt de rutes

Es demana:

- a) Suposant que *núm-trams* i *Rutes-de-població* són calculats:
 - a.1) Diagrama de classes de disseny resultant que inclogui la definició de les operacions necessàries per a tractar aquesta informació (assignades a les classes corresponents)
 - a.2) Modificacions necessàries, si n'hi ha, dels contractes de les operacions.
- b) Feu el mateix que en el cas anterior, suposant que *núm-trams* i *Rutes-de-població* són materialitzats.

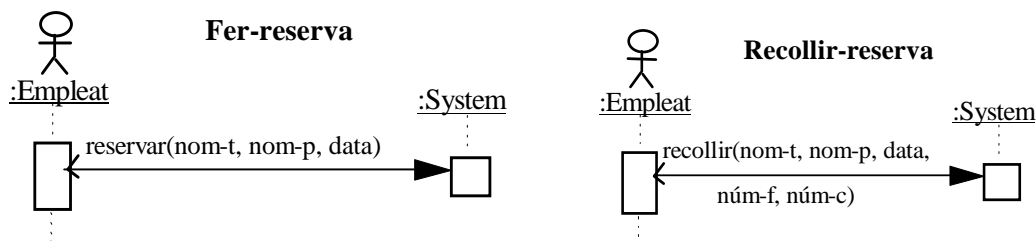
7. Considereu una empresa propietària de diversos teatres que està interessada en el disseny d'un sistema software que li permeti gestionar les reserves i les compres d'entrades de les obres que es representen a aquests teatres. Per simplificar, suposarem que de cada teatre només interessa guardar la informació de compres i reserves de l'obra que s'està representant actualment i que en un teatre concret no s'hi poden representar alhora més d'una obra. L'especificació en UML d'aquest sistema és la següent:



R.I. Textuals:

- Claus classes no associatives: (Teatre, nom); (Data, data); (Persona, nom)
- Un teatre no pot tenir dues butaques amb el mateix número de fila i número de columna
- El nombre de reserves no recollides més el nombre de compres d'un teatre per a una data determinada no pot ser superior al seu nombre de butaques.
- A "Li-correspon" la persona, la data i el teatre de la butaca han de ser els mateixos per ambdós costats.

Diagrames de seqüència del sistema i contractes de les operacions:



Contracte de l'operació *reservar*:

- Operació:** reservar (nom-t: String, nom-p: String, data: Data)
- Semàntica:** Fer una reserva d'una butaca per a la persona nom-p al teatre nom-t, en la data indicada.
- Precondicions:**
1. Existeixen el teatre, la persona i la data.
- Postcondicions:**
- Es crea una instància de l'associació Es-reserva amb els atributs corresponents.

Contracte de l'operació *recollir*:

- Operació:** recollir (nom-t: String, nom-p: String, data: Data, núm-f: Integer, núm-c: Integer)

Semàntica: Recollir una reserva prèvia de la persona nom-p, al teatre nom-t, en data i assignar-li la butaca definida per núm-f i núm-c.

Precondicions:

1. Existeix una reserva no recollida de nom-p a nom-t en data.
2. El teatre nom-t té alguna butaca amb núm-f i núm-c.

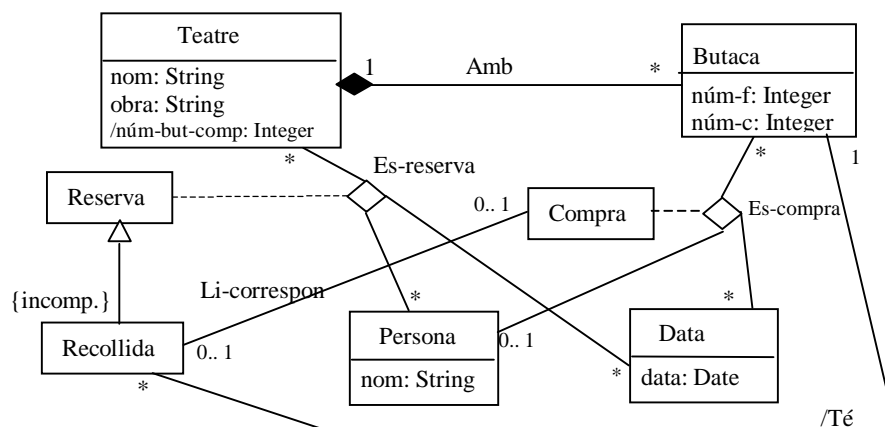
Postcondicions:

1. La reserva definida per nom-t, nom-p i data s'especialitza en Recollida.
2. Es crea una instància de Compra per a la Butaca, Data i Persona.
3. Es crea una instància de l'associació Li-correspon entre la reserva anterior i la compra acabada de crear.

Es demana:

- a) Diagrama de classes de disseny obtingut a partir de l'esquema conceptual de les dades.
- b) Modificació dels contractes de les operacions, si s'escau, com a conseqüència de la traducció de l'esquema d'especificació al de disseny.

8. Considereu una altra vegada el sistema de l'exercici anterior, però on s'ha modificat lleugerament l'esquema conceptual d'especificació per incloure-hi informació derivada. Concretament, s'hi ha afegit un atribut derivat *núm-but-comp* que permet saber el nombre de butaques comprades d'un determinat teatre i l'associació derivada *Té* entre les reserves recollides i les butaques que els hi corresponen. El nou esquema conceptual d'especificació és el següent:



R.I. Textuals:

- Claus classes no associatives: (Teatre, nom); (Data, data); (Persona, nom)
- Un teatre no pot tenir dues butaques amb el mateix número de fila i número de columna
- El nombre de reserves no recollides més el nombre de compres d'un teatre per a una data determinada no pot ser superior al seu nombre de butaques.
- A "Li-correspon" la persona, la data i el teatre de la butaca han de ser els mateixos per ambdós costats.

Informació derivada:

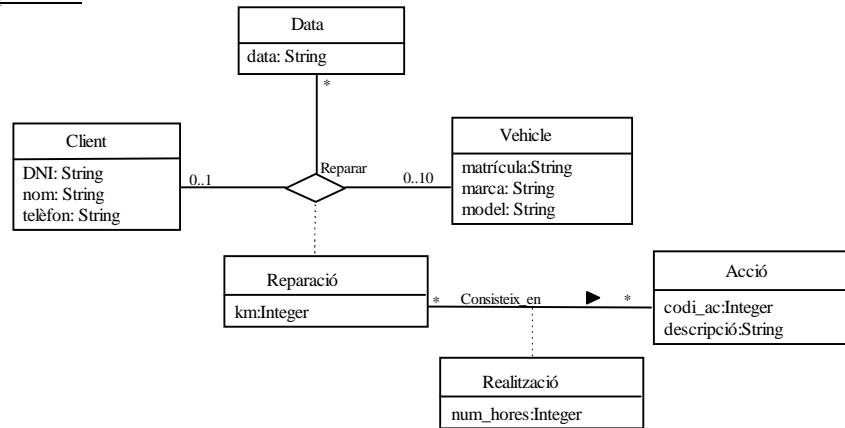
- *núm-but-comp* = suma de les butaques comprades d'aquell teatre
- *Té*: una butaca ha estat recollida per a un conjunt de reserves

Es demana:

- a) Suposant que *núm-but-comp* és materialitzat i *Té* és calculada
 - a.1) Diagrama de classes de disseny resultant que inclogui la definició de les operacions necessàries per a tractar aquesta informació (assignades a les classes corresponents)
 - a.2) Modificacions necessàries, si n'hi ha, dels contractes de les operacions.
- b) Feu el mateix que en el cas anterior, suposant ara que *Té* és materialitzada i que *núm-but-comp* és calculat.

9. Considereu un taller de reparacions d'automòbils que està interessat en el disseny d'un sistema software de suport a les seves reparacions. L'especificació en UML d'aquest sistema és la següent:

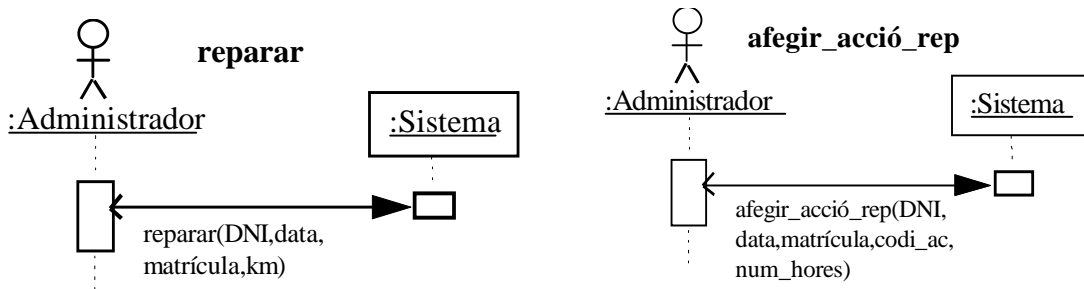
Esquema Conceptual :



R.I Textuals:

- Claus classes no associatives: (Client, DNI); (Data, data); (Vehicle, matrícula); (Acció, codi_ac)
- Un vehicle no pot tenir més de 50 reparacions.
- En una data qualsevol el taller fa com a molt 20 reparacions.

Diagrames de seqüència del sistema i contractes de les operacions:



Contracte de l'operació reparar:

Operació: reparar (DNI: String, data:String, matrícula: String, km: Integer)

Semàntica: Donar d'alta una reparació d'un vehicle

Precondicions:

1. Existeixen el client, la data i el vehicle.

Postcondicions:

1. Es crea una instància de Reparació i se li assigna el km.

Contracte de l'operació afegir acció rep:

Operació: afegir_acció_rep (DNI: String, data: String, matrícula: String, codi_ac: Integer, núm_hores: Integer)

Semàntica: S'afegeix una nova acció a la reparació.

Precondicions:

1. Existeixen la reparació i l'acció.

Postcondicions:

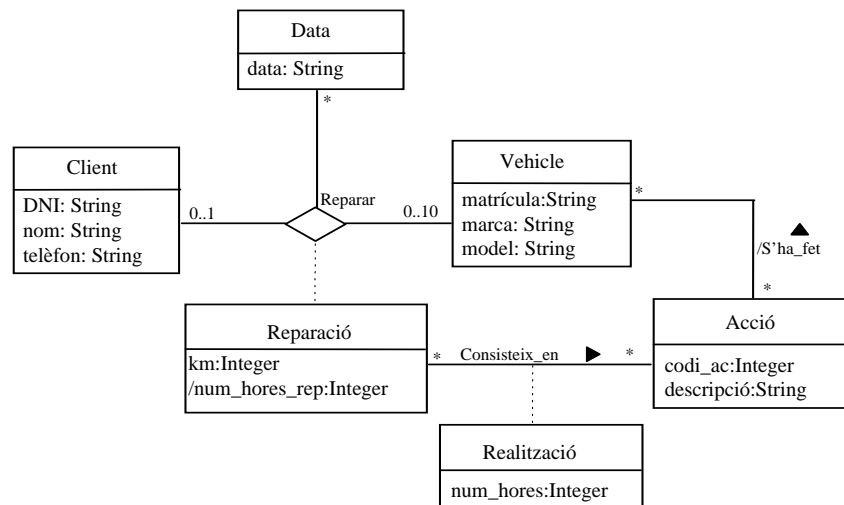
1. Es crea una instància de Realització i se li assigna el num_hores.

Es demana:

- a) Diagrama de classes de disseny obtingut a partir de l'esquema conceptual de les dades.
- b) Modificació dels contractes de les operacions, si s'escau, com a conseqüència de la traducció de l'esquema d'especificació al de disseny.

10. Considereu una altra vegada el sistema software de l'exercici anterior, però on s'ha modificat l'esquema conceptual d'especificació per incloure-hi informació derivada. Concretament, s'ha afegit un atribut derivat *num_hores_rep* que permet saber el nombre d'hores utilitzades en una reparació i l'associació derivada *S'ha_fet* entre un vehicle i totes les accions fetes sobre el vehicle en totes les reparacions.

El nou esquema conceptual d'especificació és el següent:



R.I Textuals:

- Claus classes no associatives: (Client, DNI); (Data, data); (Vehicle, matrícula); (Acció, codi_ac)
- Un vehicle no pot tenir més de 50 reparacions.
- En una data qualsevol el taller fa com a molt 20 reparacions.

Atribut i associació derivada:

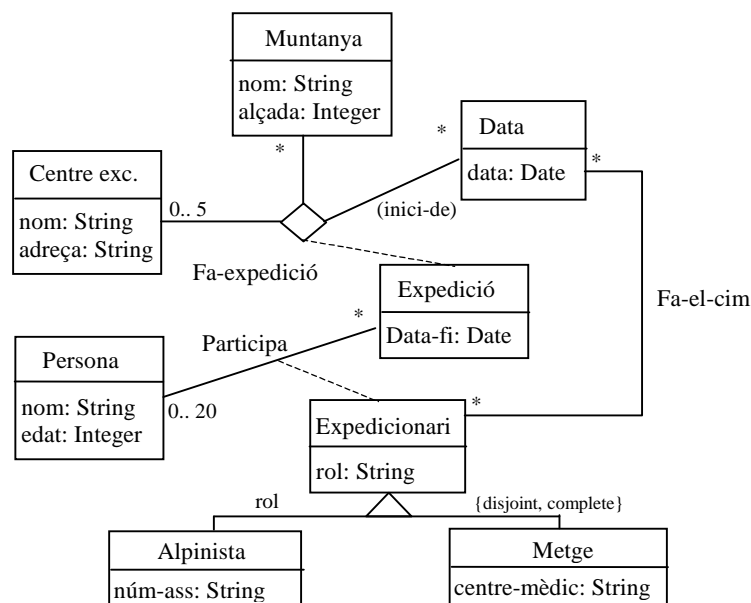
- *num_hores_rep* = número d'hores de realització de cada acció d'una reparació.
- *S'ha fet*: enregistra les accions diferents que s'han fet en un vehicle.

Es demana:

- a) Suposant que *num_hores_rep* i *S'ha_fet* són materialitzats:
 - a.1) Diagrama de classes de disseny resultant que inclogui la definició de les operacions necessàries per a tractar aquesta informació (assignades a les classes corresponents).
 - a.2) Modificacions necessàries, si n'hi ha, dels contractes de les operacions.
- b) Feu el mateix que abans, suposant que *num_hores_rep* i *S'ha_fet* són calculats.

11. Considereu una federació d'entitats excursionistes que gestiona les expedicions que fan els centres excursionistes que hi estan adscrits i que està interessada en el disseny d'un sistema software de suport a la seva activitat. L'especificació en UML d'aquest sistema és la següent.

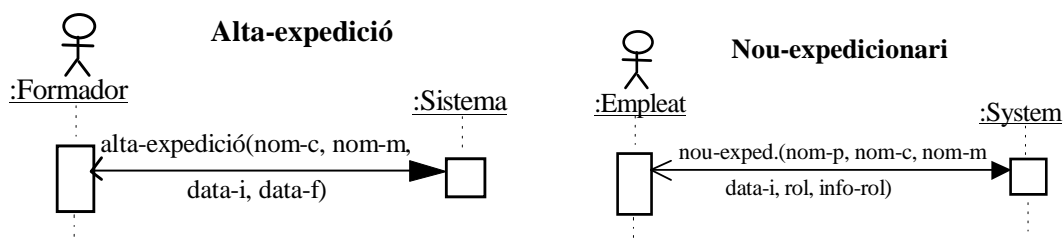
Esquema conceptual:



R.I. Textuals:

- Claus classes no associatives: (Centre-exc., nom); (Muntanya, nom); (Data, data); (Persona, nom).
- La data-fi d'una expedició ha de ser posterior a la seva data d'inici.
- En una data, hi pot haver com a màxim cinc expedicions a una muntanya.
- En una data es poden iniciar com a màxim 25 expedicions.
- La data en què un expedicionari fa el cim ha de pertanyer al període de l'expedició

Diagrames de seqüència del sistema i contractes de les operacions:



Contracte de l'operació *alta-expedició*:

Operació: alta-expedició (nom-c: String, nom-m: String, data-i: Date, data-f: Date)

Semàntica: Donar d'alta una nova expedició del centre nom-c a la muntanya nom-m des de data-i fins data-f.

Precondicions:

1. Existeix el centre identificat per nom-c.
2. Existeix la muntanya identificada per nom-m.
3. Existeix la data identificada per data-i.

Postcondicions:

1. Es crea una instància de l'associació fa-expedició definida pels objectes corresponents.

Contracte de l'operació nou-exped:

Operació: nou-exped. (nom-p: String, nom-c: String, nom-m: String, data-i: Date, rol: String, i-rol: String)

Semàntica: Afegir un nou expedicionari a una expedició.

Precondicions:

1. Existeix una expedició definida per nom-c, nom-m i data-i.
2. Existeix la persona identificada per nom-p.

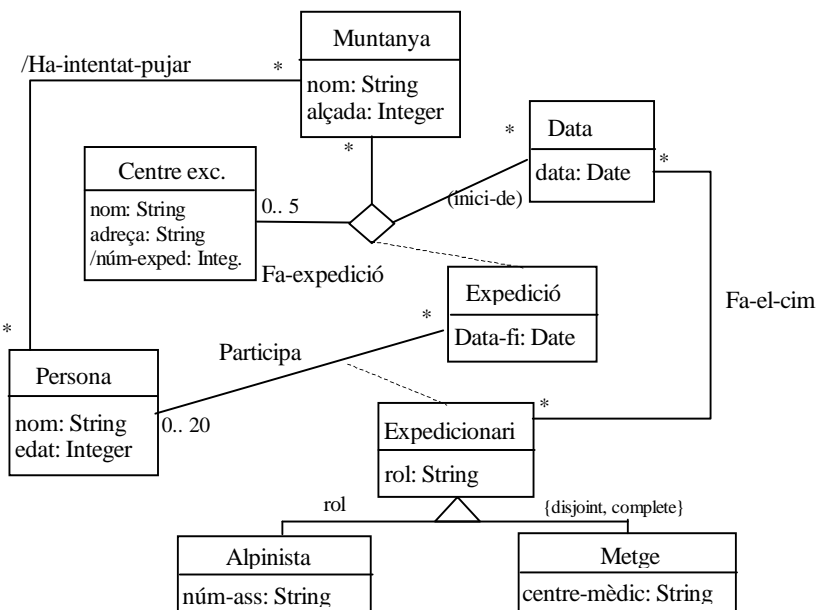
Postcondicions:

1. Es dóna d'alta una nova instància de participa definida pels paràmetres corresponents i especialitzada a Alpinista o Metge, si s'escau.

Es demana:

- a) Diagrama de classes de disseny obtingut a partir de l'esquema conceptual de les dades.
- b) Modificació dels contractes de les operacions, si s'escau, com a conseqüència de la traducció de l'esquema d'especificació al de disseny.

12. Considereu una altra vegada el sistema software de l'exercici anterior, però on s'ha modificat l'esquema conceptual d'especificació per incloure-hi informació derivada. Concretament, s'ha afegit un atribut derivat *núm-expedicions* que permet saber el nombre d'expedicions que ha fet un centre excursionista i l'associació derivada *Ha-intentat-pujar*. entre les muntanyes i les persones que han participat a alguna expedició a la muntanya. El nou esquema conceptual d'especificació és el següent:



R.I. Textual:

- Claus classes no associatives: (Centre-exc., nom); (Muntanya, nom); (Data, data); (Persona, nom).
- La data-fi d'una expedició ha de ser posterior a la seva data d'inici.
- En una data, hi pot haver com a màxim cinc expedicions a una muntanya.
- En una data es poden iniciar com a màxim 25 expedicions.
- La data en què un expedicionari fa el cim ha de pertanyer al període de l'expedició

Informació derivada:

- núm-exped = suma de les expedicions que ha fet un centre excursionista.
- Ha-intenta-pujar: una persona ha participat a alguna expedició a la muntanya

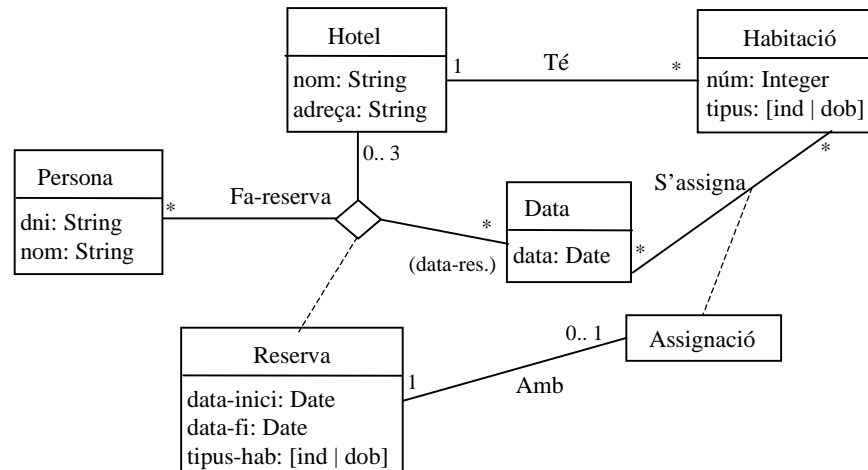
Es demana:

- a) Suposant que *núm-exped* i *Ha-intentat-pujar*. són calculats:

- a.1) Diagrama de classes de disseny resultant que inclogui la definició de les operacions necessàries per a tractar aquesta informació (assignades a les classes corresponents).
- a.2) Modificacions necessàries, si n'hi ha, dels contractes de les operacions.
- b) Feu el mateix que en el cas anterior, suposant que *núm-exped.* i *Ha-intentat-pujar* són materialitzats.

13. Considereu una cadena hotelera que gestiona les reserves que es fan als hotels que hi estan adscrits i que està interessada en el disseny d'un sistema software de suport a la seva activitat. L'especificació en UML d'aquest sistema és la següent:

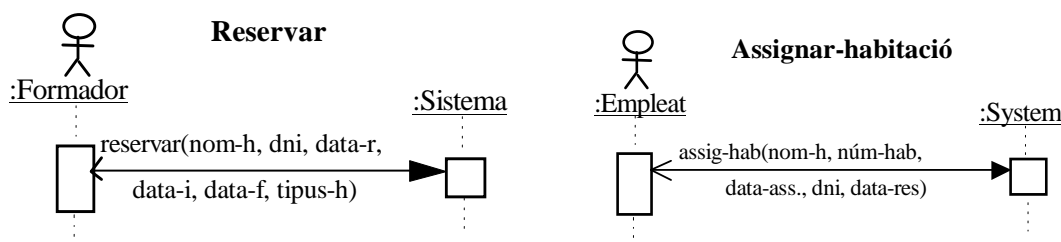
Esquema Conceptual:



R.I. Textual:

- Claus classes no associatives: (Hotel, nom); (Persona, dni); (Data, data).
- Un hotel no pot tenir dues habitacions amb el mateix núm.
- La data-fi d'una reserva ha de ser posterior a la seva data d'inici i aquesta no pot ser inferior a la data de la reserva.
- No es pot fer una reserva si l'hotel no té una habitació del tipus disponible durant tot el període.
- El nom de l'hotel i el tipus de l'habitació d'una assignació coincideixen amb el de la reserva.
- En una data no es poden fer més de 100 reserves.
- La data d'assignació és inferior a la data d'inici de la reserva i posterior (o igual) a la data de la reserva.
- Una habitació no pot tenir dues assignacions que corresponguin a reserves solapades.

Diagrames de seqüència del sistema i contractes de les operacions:



Contracte de l'operació *reservar*:

- Operació:** reservar (nom-h: String, dni: String, data-r: Date, data-i: Date, data-f: Date, tipus-h: [ind | dob])
- Semàntica:** Donar d'alta una nova reserva de la persona identificada per dni a l'hotel nom-h i amb data de reserva data-r.
- Precondicions:**
1. Existeixen l'hotel identificat per nom-h i la persona identificada per dni.
 2. Existeix una data identificada per data-r.

Postcondicions:

1. Es crea una instància de l'associació fa-reserva definida pels objectes corresponents.

Contracte de l'operació *assig-hab*:

Operació: assig-hab (nom-h: String, núm-hab: Integer, data-a: Date, dni: String, data-r: Date)

Semàntica: Assignar en data-a l'habitació núm-hab de l'hotel nom-h a la reserva definida per nom-h, dni i data-r.

Precondicions:

1. Existeix una reserva definida per nom-h, dni i data-r.
2. Existeix una habitació amb núm-hab a l'hotel nom-h.
3. Existeix una data identificada per data-a.

Postcondicions:

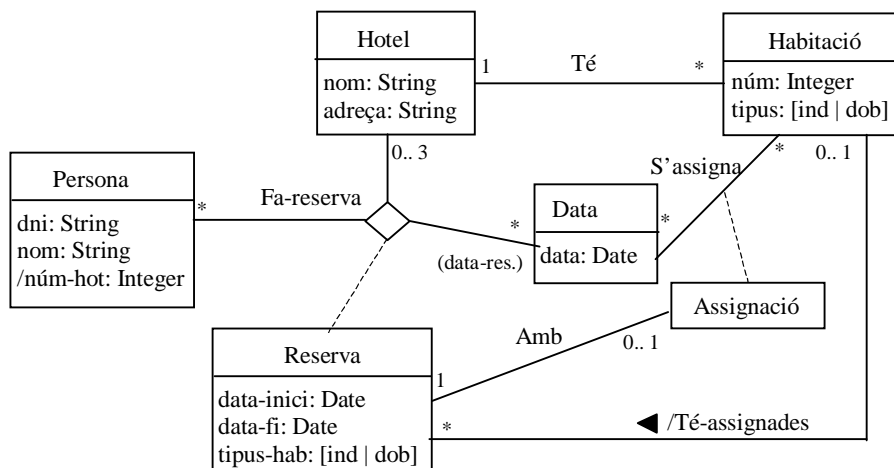
1. Es dona d'alta una nova instància de s'assigna, associada a la reserva definida per nom-h, dni i data-r.

Es demana:

- a) Diagrama de classes de disseny obtingut a partir de l'esquema conceptual de les dades.
- b) Modificació dels contractes de les operacions, si s'escau, com a conseqüència de la traducció de l'esquema d'especificació al de disseny.

- 14.** Considereu una altra vegada el sistema software de l'exercici anterior, però on s'ha modificat l'esquema conceptual d'especificació per incloure-hi informació derivada. Concretament, s'ha afegit un atribut derivat *núm-hotels* que permet saber el nombre d'hotels als que una persona hi ha fet alguna reserva i l'associació *Té-assignades* entre les habitacions i les reserves que hi han estat assignades.

El nou esquema conceptual d'especificació és el següent:

R.I. Textual:

- Claus classes no associatives: (Hotel, nom); (Persona, dni); (Data, data).
- Un hotel no pot tenir dues habitacions amb el mateix núm.
- La data-fi d'una reserva ha de ser posterior a la seva data d'inici i aquesta no pot ser inferior a la data de la reserva.
- No es pot fer una reserva si l'hotel no té una habitació del tipus disponible durant tot el període.
- El nom de l'hotel i el tipus de l'habitació d'una assignació coincideixen amb el de la reserva.
- En una data no es poden fer més de 100 reserves.
- La data d'assignació és inferior a la data d'inici de la reserva.
- Una habitació no pot tenir dues assignacions que corresponguin a reserves solapades.

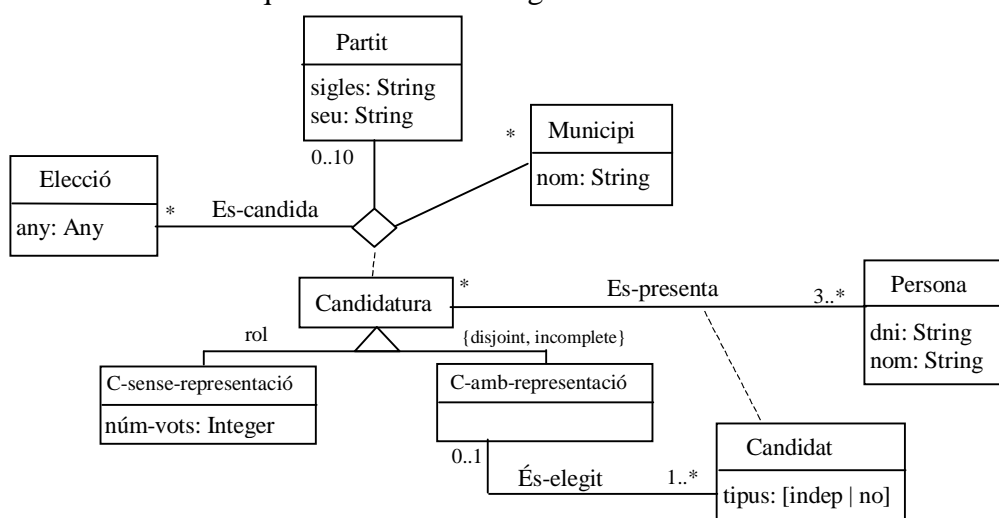
Informació derivada:

- núm-hot = nombre d'hotels als que la persona hi ha fet alguna reserva.
- Té-assignedes: una habitació té assignades diverses reserves.

Es demana:

- Suposant que *núm-hot* i *Té-assignedes*. són calculats:
 - Diagrama de classes de disseny resultant que inclogui la definició de les operacions necessàries per a tractar aquesta informació (assignades a les classes corresponents).
 - Modificacions necessàries, si n'hi ha, dels contractes de les operacions.
- Feu el mateix que en el cas anterior, suposant que *núm-hot* i *Té-assignedes* són materialitzats.

15. Considereu un centre de gestió electoral que està interessat a dissenyar un sistema software per a gestionar les candidatures i els resultats de les successives eleccions municipals que es realitzen. L'especificació en UML d'aquest sistema és la següent:



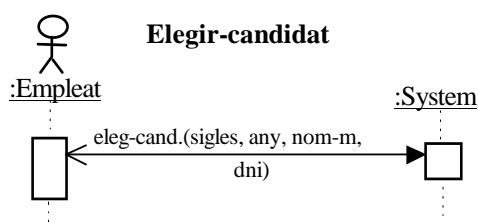
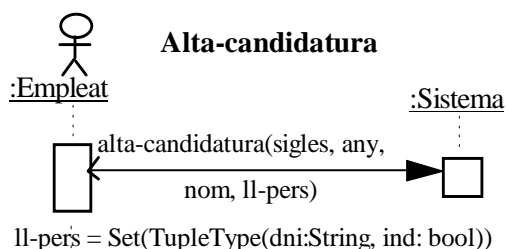
R.I. Textual:

- Claus classes no associatives: (Partit, sigles); (Municipi, nom); (Elecció, any); (Persona, dni).
- Un candidat només es pot elegir a la candidatura a la que es presenta.
- Una persona no pot estar a més d'una candidatura per elecció
- En una elecció com a màxim hi ha 200 candidatures.

Esquema Conceptual:

Fixeu-vos que l'especialització de candidatura és *incomplete* perquè primer es defineixen les candidatures que es presentaran a les eleccions i, al cap d'un temps, es coneixen els seus resultats.

Diagrames de seqüència del sistema i contractes de les operacions:



Contracte de l'operació *alta-candidatura*:

Operació: alta-candidatura (sigles: String, any: Any, nom-m: String, ll-pers: Set(TupleType(dni:String, ind: bool)))

Semàntica: Donar d'alta una candidatura del partit sigles al municipi nom-m en l'any any que presenta els candidats continguts a ll-pers.

Precondicions:

1. Existeix el partit identificat per sigles
2. Existeix el municipi identificat per nom-m.
3. Existeix l'elecció identificada per any.
4. Existeixen les persones corresponents a tots els dni de ll-pers.

Postcondicions:

1. Es crea una instància de l'associació Es-candida definida pels objectes corresponents.
2. Es creen tantes instàncies de Es-presenta com dnis hi hagi a ll-pers.

Contracte de l'operació *elegir-cand*:

Operació: elegir-cand. (sigles: String, nom-m: String, any: Any, dni: String)

Semàntica: Afegir un nou candidat elegit a a una candidatura.

Precondicions:

1. Existeix la candidatura definida per sigles, nom-m i any.
2. La candidatura no és candidatura sense representació.
3. La persona dni existeix.

Postcondicions:

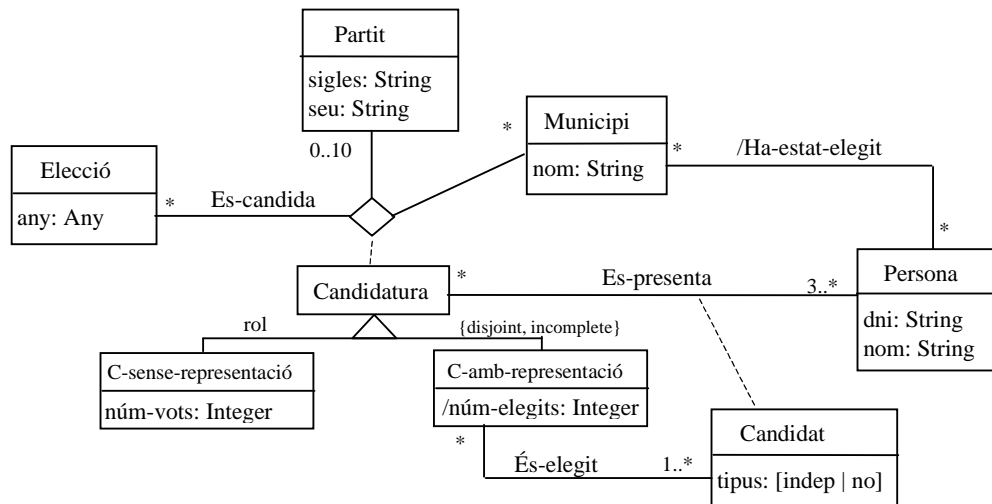
1. Si la candidatura no tenia cap candidat elegit, aleshores s'especialitza a candidatura amb representació.
2. Es dóna d'alta una nova instància de És-elegit definida pels paràmetres corresponents.

Es demana:

- a) Diagrama de classes de disseny obtingut a partir de l'esquema conceptual de les dades.
- b) Modificació dels contractes de les operacions, si s'escau, com a conseqüència de la traducció de l'esquema d'especificació al de disseny.

- 16.** Considereu una altra vegada el sistema software de l'exercici anterior, però on s'ha modificat l'esquema conceptual d'especificació per incloure-hi informació derivada. Concretament, s'ha afegit un atribut derivat *núm-elegits* que permet saber el nombre de persones elegides a una candidatura amb representació i l'associació derivada *Ha-estat-elegit* entre els municipis i les persones que han estat elegits a alguna elecció d'aquell municipi.

El nou esquema conceptual d'especificació és el següent:



R.I. Textual:

- Claus classes no associatives: (Partit, sigles); (Municipi, nom); (Elecció, any); (Persona, dni).
- Un candidat només es pot elegir a la candidatura a la que es presenta.
- Una persona no pot estar a més d'una candidatura per elecció.
- En una elecció com a màxim hi ha 200 candidatures.

Informació derivada:

- núm-elegits = suma dels candidats elegits per una candidatura.
- Ha-estat-elegit: una persona ha estat elegit alguna vegada a un municipi.

Es demana:

a) Suposant que *núm-elegits* i *Ha-estat-elegit* són calculats:

- Diagrama de classes de disseny resultant que inclogui la definició de les operacions necessàries per a tractar aquesta informació (assignades a les classes corresponents).
- Modificacions necessàries, si n'hi ha, dels contractes de les operacions.

b) Feu el mateix que en el cas anterior, suposant que *núm-elegits* i *Ha-estat-elegit* són materialitzats.