Patró *Domain Model*: de l'esquema d'especificació al de disseny

FIB

Departament d'Enginyeria de Serveis i Sistemes d'Informació
UNIVERSITAT POLITÈCNICA DE CATALUNYA

De l'esquema d'especificació al de disseny

Motivació

Obtenció del diagrama de classes de disseny

- Eliminació d'associacions n-àries i classes associatives
- Eliminació de la classe Data i similars

Obtenció dels contractes de disseny de les operacions

- Incorporació de la comprovació de les restriccions d'integritat
- Incorporació del tractament de la informació derivada

Exemple

Bibliografia

Motivació

Al **disseny** hi tenim **components software** i no conceptes del domini

Limitació tecnològica : no es poden implementar directament tots els conceptes que hem usat a l'especificació:

- associacions n-àries, amb n > 2.
- classes associatives
- control de les restriccions d'integritat
- informació derivada

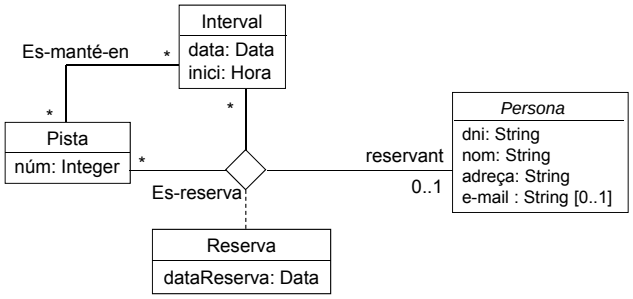
cal una transformació prèvia dels diagrames d'especificació:

- *Obtenció del diagrama de classes de disseny*:
  - eliminar associacions n-àries i classes associatives
  - eliminar la classe Data i similars
- *Obtenció dels contractes de disseny de les operacions:*
  - controlar les restriccions d'integritat
  - tractar la informació derivada

Eliminació d'associacions n-àries i classes associatives
Definició del problema i exemple

Objectiu:
Donat un esquema conceptual de les dades, obtenir un diagrama de classes de disseny que no tingui classes associatives i on totes les associacions siguin binàries

Exemple: esquema conceptual d'especificació inicial:

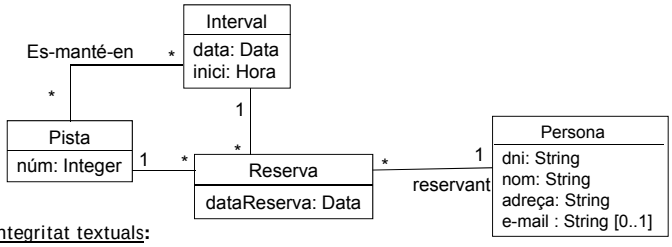


Restriccions d'integritat textuals:

- claus classes no associatives: (Pista, núm); (Interval, data + inici); (Persona, dni)

Eliminació d'associacions n-àries i classes associatives
Obtenció del diagrama de classes de disseny

Diagrama de classes de disseny obtingut:



Restriccions d'integritat textuals:

1. claus classes no associatives: (Pista, núm); (Interval, data + inici); (Persona, dni)
2. **(Afegida)** No hi pot haver dues reserves amb els mateixos Pista, Interval i Persona
3. **(Afegida)** *Donats una pista i un interval, com a màxim els pot tenir reservats una Persona*

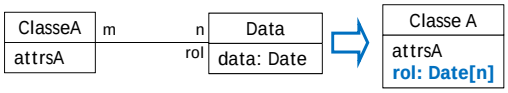
s'elimina perquè està inclosa a la tercera restricció textual

El diagrama de classes de disseny obtingut ha de tenir la mateixa semàntica que l'esquema conceptual d'especificació de partida

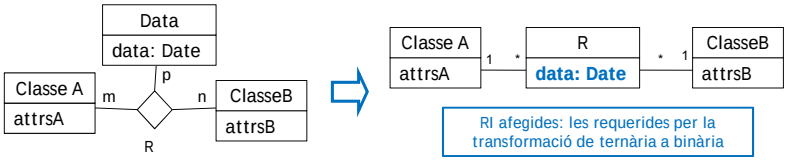
Eliminació de la classe Data i similars

Tractament de la classe Data:

- A l'esquema conceptual de les dades sovint apareix la classe Data
- Per a la majoria de llenguatges, les dates són un tipus de dada més
- Cal eliminar la classe i convertir les associacions en atributs



RI afegida: Per a una data *d*, només hi pot haver *m* instàcies de *ClasseA* tals que *rol* inclou *d*



RI afegides: les requerides per la transformació de ternària a binària

Cal aplicar una transformació similar per a tots els Value Types com ara Any, Import, etc.

Contractes de les operacions de disseny
Són una mica diferents als d'especificació

Operació: nom i paràmetres de l'operació (*signatura de l'operació*)

Precondicions:
Condicions que estan garantides quan es crida l'operació.

Excepcions:
Condicions que l'operació ha de comprovar
L'operació no s'ha d'executar si alguna condició se satisfà.

Postcondicions:
Canvis d'estat que es produeixen com a conseqüència de l'execució:

Sortida:
Descripció de la sortida que proporciona l'operació

Observeu que:
El concepte de precondició no és el mateix a especificació que a disseny
A disseny apareix un nou concepte, excepció, que captura el que a especificació era la precondició

Incorporació de les restriccions d'integritat als contractes
Definició del problema

Els models d'**especificació** són **no redundants** entre ells

Quan dissenyem, cal garantir que el **software** a desenvolupar **satisfaci les restriccions d'integritat** (gràfiques i textuals) de l'esquema conceptual.

En *Domain Model* tota la lògica del sistema està a la capa de domini (contractes)

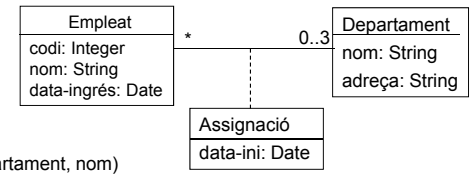
cal afegir a l'excepció de cada operació el control de les restriccions d'integritat que poden ser violades per aquella operació

com a conseqüència, en el patró *domain model* el **model estàtic del disseny no té restriccions d'integritat**
(tot i que les podem mostrar a efectes informatius)

En general, la capa de presentació i la de gestió de dades també controlaran algunes restriccions d'integritat. Per tant, aquest control no es farà únicament des de la capa de domini.

Control de les restriccions d'integritat: exemple
Esquema conceptual d'especificació inicial

Esquema conceptual d'especificació:



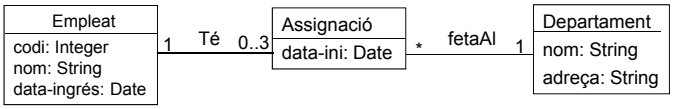
R.I. Textuals:
- Claus: (Empleat, codi); (Departament, nom)
- Data-ini d'assignació ≥ data-ingrés
- Un empleat no pot estar assignat alhora als departaments de Vendes i de Control-Vendes

Operació:	novaAssignació (codiEmp: Integer, nomDept: String, data: Date)
Pre:	- Existeix l'empleat codiEmp i el departament nomDept.
Post:	- Es dóna d'alta l'associació entre l'empleat i el departament.

Objectiu:
Eliminar les restriccions d'integritat de l'esquema de les dades i fer que les condicions que imposen siguin garantides pels contractes de les operacions

Control de les restriccions d'integritat: exemple
Pas 1: eliminar associacions n-àries i classes associatives

Diagrama de classes de disseny:



R.I. Textuals:
- Claus: (Empleat, codi); (Departament, nom)
- Data-ini d'assignació ≥ data-ingrés
- Un empleat no pot estar assignat als departaments de Vendes i de Control-Vendes
- **(Afegida)** *No hi pot haver dues assignacions amb els mateixos Empleat i Departament*

Control de les restriccions d'integritat: exemple
Pas 2: afegir a les excepcions el control de les restriccions

Operació:	novaAssignació (codiEmp: Integer, nomDept: String, data: Date)
Pre:	- Existeix l'empleat codiEmp i el departament nomDept.
Post:	- Es dóna d'alta l'associació entre l'empleat i el departament.

Contracte de disseny de l'operació *novaAssignació*:

Operació: novaAssignació (codiEmp: Integer, nomDept: String, data: Date)
Precondicions:
Excepcions:
- *empNoExisteix*: no existeix cap Empleat identificat per codiEmp
- *deptNoExisteix*: no existeix cap Departament identificat per nomDept
- *jaAssignat*: l'empleat codiEmp ja estava assignat al departament nomDept
- *jaTé3Depts*: l'empleat codiEmp ja està assignat a tres departaments
- *assigAmbRetard*: la data d'ingrés de l'empleat codiEmp és superior a data
- *assigIncompatible*: l'empleat passaria a estar assignat a vendes i control-vendes

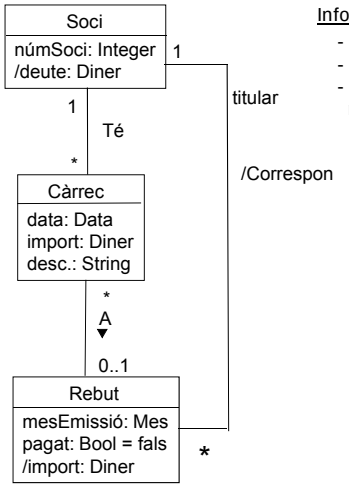
Postcondicions:
- Es dóna d'alta l'associació entre l'empleat i el departament.

Tractament de la informació derivada
Definició del problema

- Els atributs i les associacions derivats es poden:
  - *Calcular* quan es necessiten mitjançant l'execució d'una funció
  - *Materialitzar* (o sigui, emmagatzemar físicament el seu valor)
- Si es *calcula*:
  - **Desapareix** (explícitament) la **informació derivada** que es decideix calcular
  - **Apareixen** noves **operacions** per obtenir la informació derivada que es calcula
- Si es *materialitza*:
  - Cal **modificar** la postcondició dels **contractes** de les operacions que provoquen canvis al valor de la informació que es materialitza
  - S'elimina del diagrama de classes la indicació que la informació és derivada
- En la decisió hi influeix el temps de càlcul, la freqüència d'accés i l'espai ocupat.

Tractament de la informació derivada

Punt de partida: esquema conceptual d'especificació



Informació derivada (regles de derivació):

- Soci::deute = suma dels imports dels rebuts no pagats del soci
- Rebut::import = suma dels càrrecs d'aquell rebut
- Correspon = associa un rebut amb el soci que té els càrrecs del rebut

Operació: ferRebuts(m: Mes)

Precondició: no hi ha cap rebut del mes m

Postcondició:

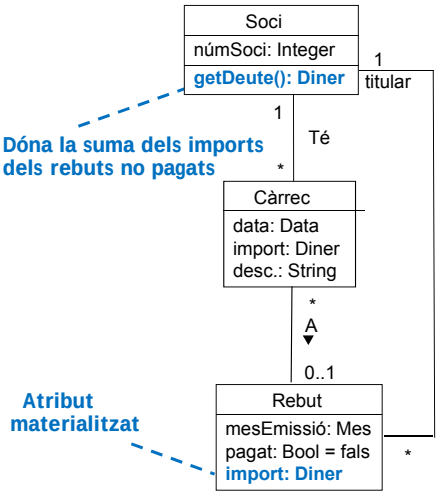
Per cada soci que té algun càrrec al Mes m:
es crea un rebut amb mesEmissió=m associat a tots els càrrecs del soci en m

- Decidim:
  - *calcular l'atribut deute de Soci*
  - *materialitzar import de Rebut*
  - *materialitzar l'associació Correspon*

Tractament de la informació derivada

Resultat: diagrama de classes i contracte de les operacions de disseny

Diagrama de classes de disseny resultant:



Contracte de disseny obtingut:

Operació: ferRebuts(m: Mes)

Excepció: ja hi ha cap rebut del mes m

Postcondició:

Per cada soci que té algun càrrec al Mes m:
es crea un rebut amb mesEmissió=m associat a tots els càrrecs del soci en m i amb import igual a la suma dels imports d'aquests càrrecs

s'associa el rebut creat amb el soci

Correspon

Associació materialitzada

Exemple

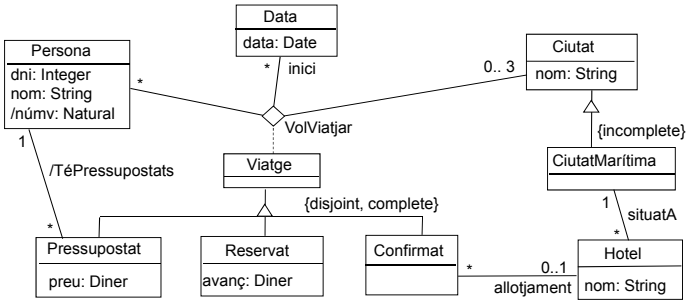
Especificació:

- Esquema conceptual de les dades
- Esquema del comportament

Obtenció de l'esquema de disseny:

- Diagrama de classes de disseny
- Contractes de les operacions de disseny

Especificació: Esquema conceptual de les dades



R.I. Textuals:

- Claus classes no associatives: (Persona, dni); (Data, data); (Ciutat, nom)
- Una Ciutat Marítima no pot tenir més d'un Hotel amb el mateix nom
- Un viatge confirmat es fa a un Hotel de la mateixa ciutat on es fa el viatge
- Una persona no pot tenir més d'un viatge confirmat amb una mateixa data d'inici

Informació derivada:

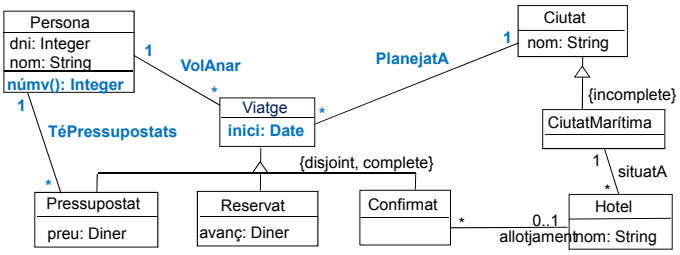
- númv: és el nombre de viatges confirmats d'aquella Persona
- téPressupostats: és igual al conjunt de viatges pressupostats d'una Persona

Suposicions:

- L'atribut derivat *númv* s'ha de calcular.
- L'associació *téPressupostats* s'ha de materialitzar

Diagrama de classes de disseny

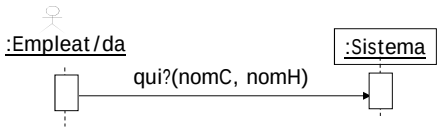
Eliminació de n-àries i associatives, tractament de la informació derivada



R.I. Textuals:

- Claus classes no associatives: (Persona, dni); (Data, data); (Ciutat, nom)
- Una Ciutat Marítima no pot tenir més d'un Hotel amb el mateix nom
- Un viatge confirmat es fa a un Hotel de la mateixa ciutat on es fa el viatge
- Una persona no pot tenir més d'un viatge confirmat amb una mateixa data d'inici
- **Per una persona i una data no poden existir més de tres viatges**
- **No poden existir dos viatges de la mateixa persona a la mateixa ciutat amb la mateixa data d'inici**

Especificació: contracte operació "qui?"



Operació: qui? (nomCiutat: String, nomHotel: String): Set(String)

Semàntica: retorna la llista de noms de les persones que s'han allotjat alguna vegada a l'hotel nomHotel de la ciutat nomCiutat.

Precondicions:

- L'hotel nomHotel ha d'estar situat a la ciutat nomCiutat
- Hi ha d'haver com mínim una persona que s'hagi allotjat a l'hotel

Postcondicions:

- Es retorna la llista que conté els noms de totes les persones que s'han allotjat alguna vegada a l'hotel nomHotel de la ciutat nomCiutat.

Disseny: contracte de l'operació "qui?"

Operació: qui? (nomCiutat: String, nomHotel: String): Set(String)

Semàntica: retorna la llista de noms de les persones que s'han allotjat alguna vegada a l'hotel nomHotel de la ciutat nomCiutat.

Precondicions:

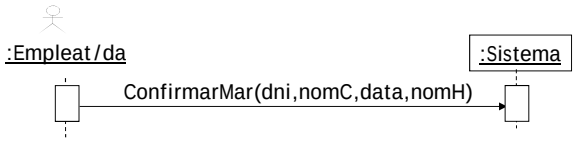
Excepcions:

- hotelIncorrecte: l'hotel nomHotel no està situat a la ciutat nomCiutat
- ningúAllotjat: no hi ha cap persona que s'hagi allotjat a l'hotel nomHotel de la ciutat nom Ciutat

- **Postcondicions**:
  - Es retorna la llista que conté els noms de totes les persones que s'han allotjat alguna vegada a l'hotel nomHotel de la ciutat nomCiutat.

Les postcondicions de les operacions de consulta no es veuen mai afectades al passar de l'esquema d'especificació al de disseny

Especificació: contracte"ConfirmarViatgeAlMar"



Operació: confirmarMar (dni: Integer, nomCiutat: String, data: Date, nomHotel: String)

Semàntica: confirmar un viatge a una ciutat de mar i assignar-li l'hotel

Precondicions:

- El viatge identificat per (dni, data, nomCiutat) ha d'estar Pressupostat o Reservat
- L'hotel nomHotel està situat a la ciutat nomCiutat

Postcondicions:

- El viatge passa a estar Confirmat, i deixa d'estar Pressupostat o Reservat
- El viatge Confirmat s'associa a l'hotel nomHotel

Disseny: contracte de l'operació "confirmarMar"

Operació: confirmar-mar (dni: Integer, nomCiutat: String, data: Date, nomHotel: String)

Semàntica: confirmar un viatge a una ciutat marítima i assignar-li l'hotel

Precondicions:

Excepcions:

- estatIncorrecte: el viatge identificat per (dni, data, nomCiutat) no està pressupostat ni reservat
- hotelInexistent: l'hotel nomHotel no és de la ciutat nomCiutat
- jaTéViatgeConfirmat: la persona dni ja té un viatge confirmat en data

Postcondicions:

- El viatge passa a estar Confirmat , i deixa d'estar Pressupostat o Reservat
- El viatge Confirmat s'associa a l'hotel nomHotel
- Si el viatge era Pressupostat, s'elimina la instància de l'associació téPressupostat entre Pressupostat i Persona **(afegida).**

Bibliografia

- Pressman, R.G. *"Software Engineering. A Practitioner's Approach"*, Mc Graw-Hill, 2015 (8a edició).
- Larman, C. "*Applying UML and Patterns. An Introduction to Object-oriented Analysis and Design*", Prentice Hall, 2005, (3ª edició)
- Meyer, B. "*Object-oriented Software Construction*", Prentice Hall, 1997.