

## **Disseny de la Capa de Gestió de Dades: Estratègies de Gestió de Persistència de Dades**

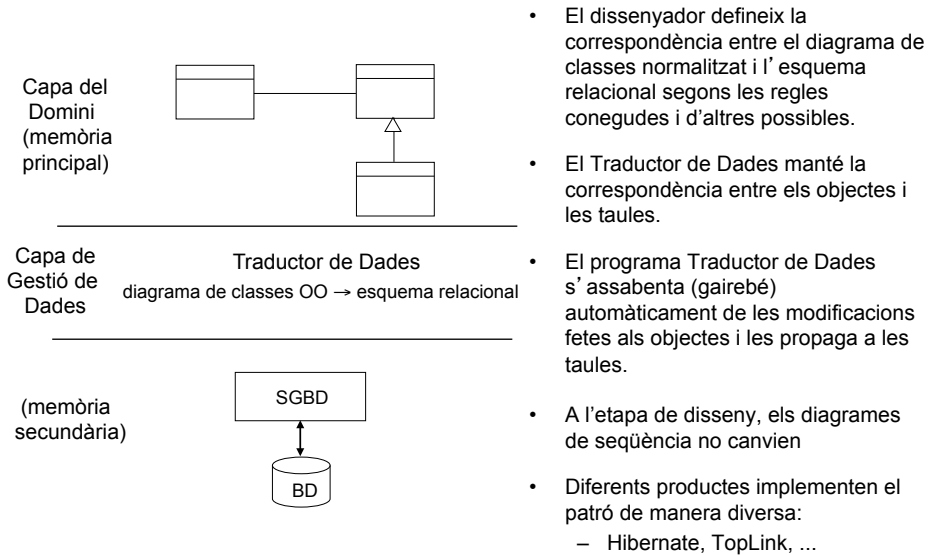
- Generació automàtica de la persistència
  - Patró Traductor de Dades (*Data Mapper*)
  - Variant: operacions de consulta a la capa de dades
- Disseny directe de la persistència
  - Disseny directe d'operacions
    - Controlador de la Capa de Domini amb lògica simple
    - Controlador de la Capa de Domini com a intermediari
- Exemple
- Disseny de les capes de domini i gestió de dades, resum
- Bibliografia

## **Estratègies de gestió de la persistència Generació automàtica**

- Generador automàtic de la persistència:
  - Sistema que proporciona una traducció automàtica per emmagatzemar les dades en memòria secundària
  - Transforma les actualitzacions fetes a memòria principal en actualitzacions en memòria secundària
  - Requereix especificar la correspondència entre el diagrama de classes (ja normalitzat) i l'esquema de la base de dades
- Característiques:
  - La gestió de la persistència és transparent al dissenyador
  - Pot no aprofitar-se completament de les funcionalitats ofertes pels SGBD

S'utilitza en conjunció amb el patró Domain Model

### Generació automàtica de la persistència Patró Traductor de Dades (*Data Mapper*)

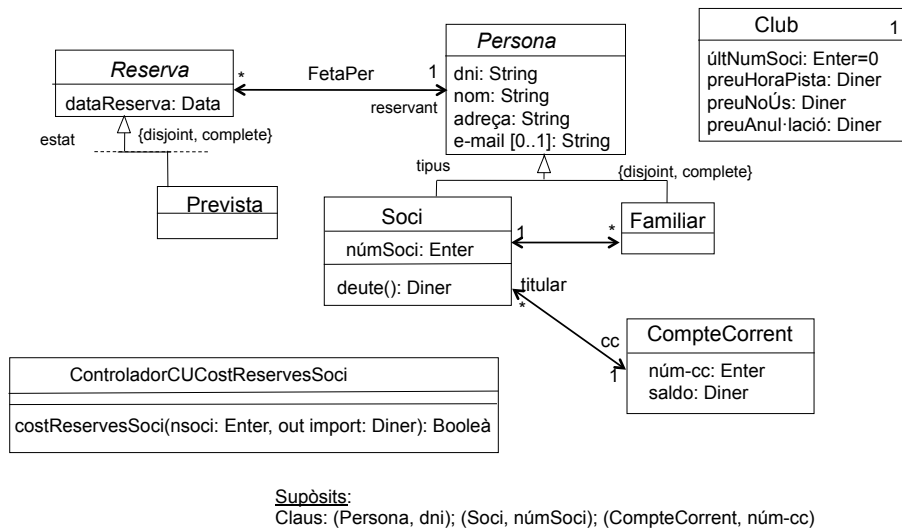


### Generació automàtica de la persistència Operacions arbitràries de consulta a la capa de gestió de dades

- La generació automàtica de persistència és molt còmode per l'actualització de la BD però no tant per certes consultes
  - Diagrames de seqüència farragosos
  - Possible pèrdua d'eficiència
- Opció: la capa de dades ofereix les operacions de consulta requerides pel problema
  - La capa de domini fa de simple intermediària
  - Les consultes són arbitràriament complexes, i poden involucrar diverses taules
- Les opcions tecnològiques actuals de generació automàtica acostumen a oferir aquesta possibilitat:
  - Ex: Hibernate, *createQuery*

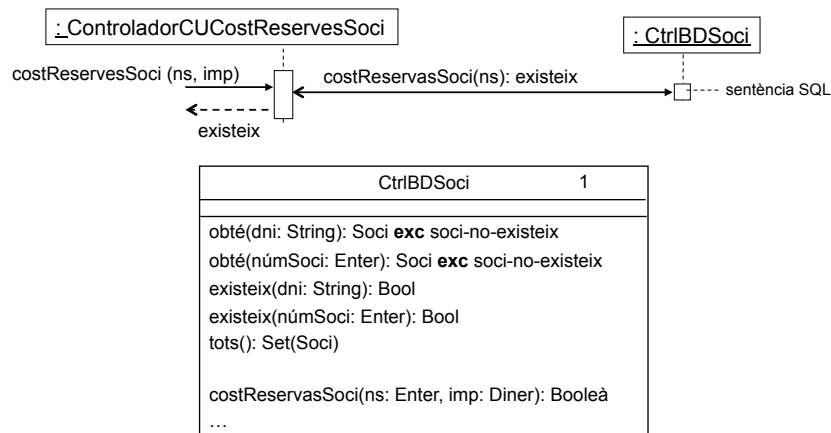
### Generació automàtica de la persistència

#### Operacions arbitràries de consulta a la capa de gestió de dades, exemple



### Generació automàtica de la persistència

#### Operacions arbitràries de consulta a la capa de gestió de dades, exemple



### **Estratègies de gestió de la persistència**

#### **Disseny directe**

- Disseny directe de la persistència:
  - El disseny del software es fa tenint en compte el SGBD que s'utilitzarà a la implementació
  - Serà diferent segons usem un SGBD orientat a objectes, relacional, etc.
- Característiques:
  - El dissenyador necessita conèixer el SGBD per dissenyar la capa de gestió de dades (i el sistema software en general)
  - Pot aprofitar-se completament de les funcionalitats ofertes pels SGBD
  - El disseny que es fa és similar per tots els sistemes concrets d'una mateixa família

S'utilitza en conjunció amb el patró Transaction Script

### **Disseny directe de la persistència**

#### **Disseny de les operacions, estratègies**

- Controlador de la capa de domini amb lògica simple
  - El controlador crida operacions que accedeixen a la BD per obtenir dades i actualitzar-les
    - ✓ Ús de patrons d'accés a dades
  - El controlador fa manipulacions simples sobre aquestes dades
- Controlador de la capa de domini com a intermediari
  - El controlador delega en un procediment emmagatzemat que s'encarrega de tota la feina

És possible combinar les dues estratègies en un mateix sistema

En qualsevol dels dos casos, també es poden usar els elements de SQL vistos per assignar algunes de les responsabilitats

### Disseny directe de la persistència

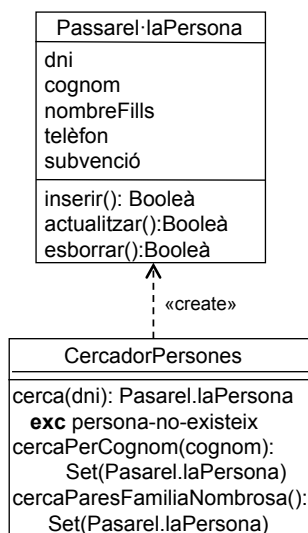
#### Controlador de la capa de domini amb lògica simple

#### Patrons de la capa de dades per al disseny directe en BD relacionals

- Passarel·la Taula (*Table Data Gateway*)
  - Una classe (d'una sola instància) per cada taula de la base de dades.
  - Encapsula l'accés (cerca, actualització) a una taula.
- Passarel·la Fila (*Row Data Gateway*)
  - Una classe per una taula de la base de dades. Una instància és una fila de la taula.
  - Les cerques poden ser operacions de classe o estar en una altra classe.
  - Encapsula l'actualització d'una fila.
  - Aplicable a models de disseny simples.
- Enregistrament actiu (*Active Record*)
  - Passarel·la fila amb operacions pròpies del domini.

### Disseny directe de la persistència

#### Patró Passarel·la Fila



- Una *classe* passarel·la per *taula* (o vista).  
Les *instàncies* són *tuples* de la taula.  
Cada *columna* de la taula és un *atribut* de la classe.
- Les operacions de cerca consulten la taula corresponent mitjançant sentències SQL (o procediments emmagatzemats) i creen les instàncies de la passarel·la amb el resultat.
- Les operacions *inserir*, *actualitzar* i *esborrar* també contenen sentències SQL (o procediments emmagatzemats).

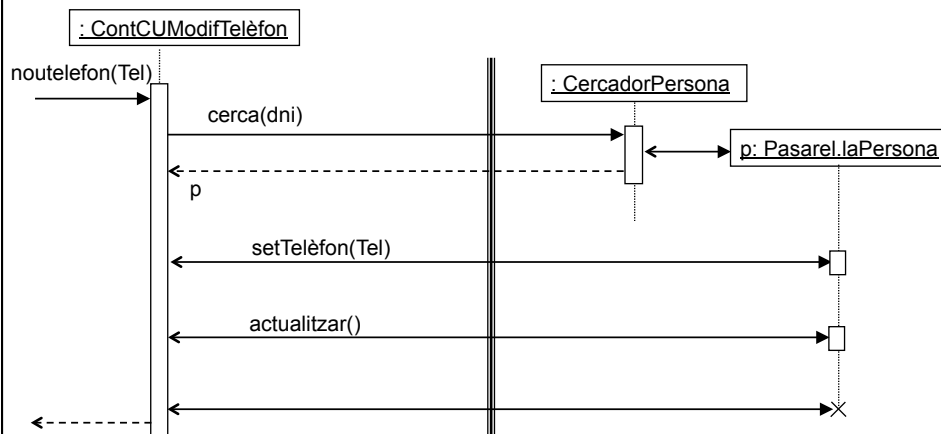
### Disseny directe de la persistència

#### Disseny usant el patró Pasarel.la Fila

- El disseny de les operacions pot **aprofitar** la potència del **SQL**
- Els controladors de la capa de domini:
  - Creen pasarel.les per obtenir objectes
  - Manipulen aquests objectes a memòria principal
  - Els tornen a la capa de gestió de dades usant la pasarel.la

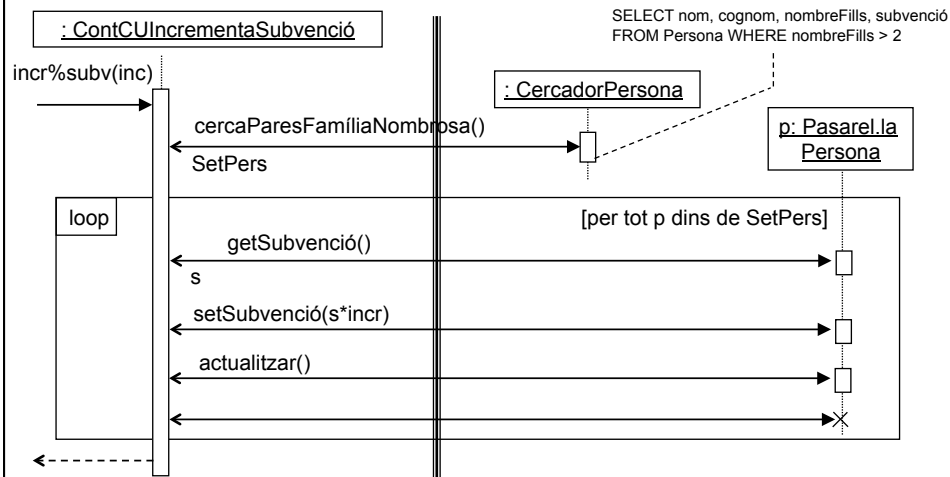
### Disseny directe de la persistència

#### Disseny usant el patró Pasarel.la Fila, exemple



### Disseny directe de la persistència

#### Disseny usant el patró Pasarel.la Fila, un altre exemple



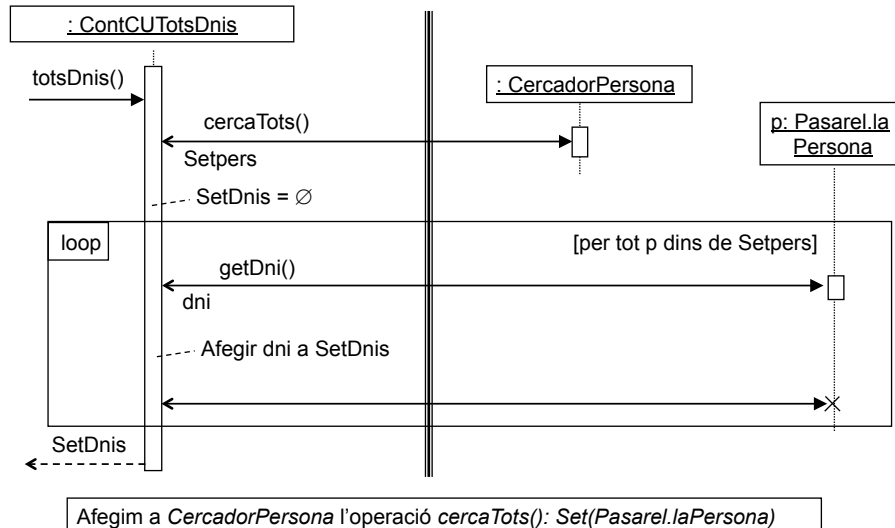
### Disseny directe de la persistència

#### Disseny usant el patró Pasarel.la fila amb operacions arbitràries de consulta

- Igual que passava amb Domain Model, algunes operacions de consulta poden ser feixugues usant el patró estrictament
- Solució: permetre operacions de consulta en els cercadors
  - Obtenció de col.leccions
  - Càlcul de resultats
- Aquestes operacions arbitràries de consulta permeten consultar diverses taules de la base de dades, i retornar valors de qualsevol tipus

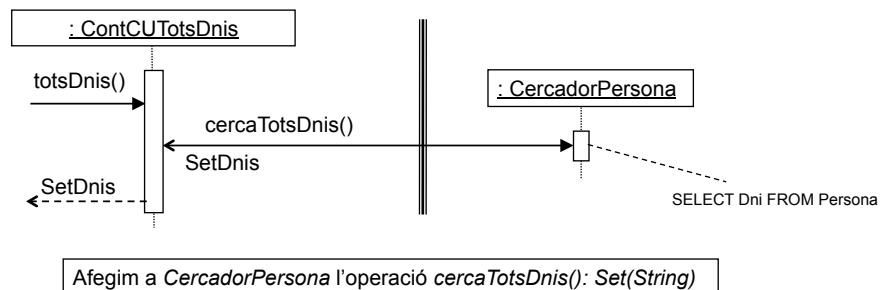
### Disseny directe de la persistència

Disseny usant el patró Pasarel.la Fila, exemple amb operacions de consulta simples



### Disseny directe de la persistència

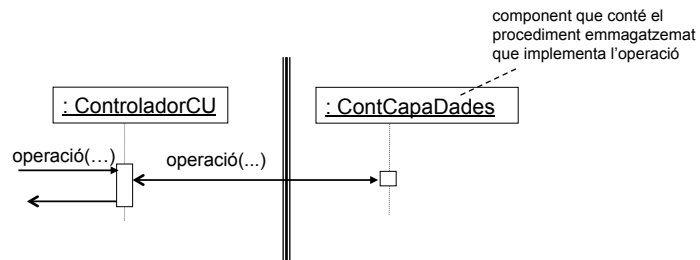
Disseny usant el patró Pasarel.la Fila, exemple amb operacions arbitràries de consulta





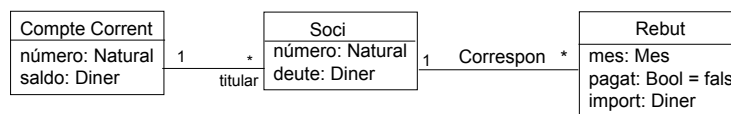
### Disseny directe de la persistència

#### Controlador de la capa de domini com a simple intermediari



### Exemple de gestió de la persistència

#### Enunciat



**context** pagarRebut (ns: Enter, m: Mes)

-- operació per pagar el rebut del mes *m* del Soci *ns*

**exc** rebut-no-existeix: no existeix un rebut del soci *ns* el mes *n*

rebut-ja-pagat: el rebut ja està pagat

saldo-insuficient: el client no té prou saldo al seu compte

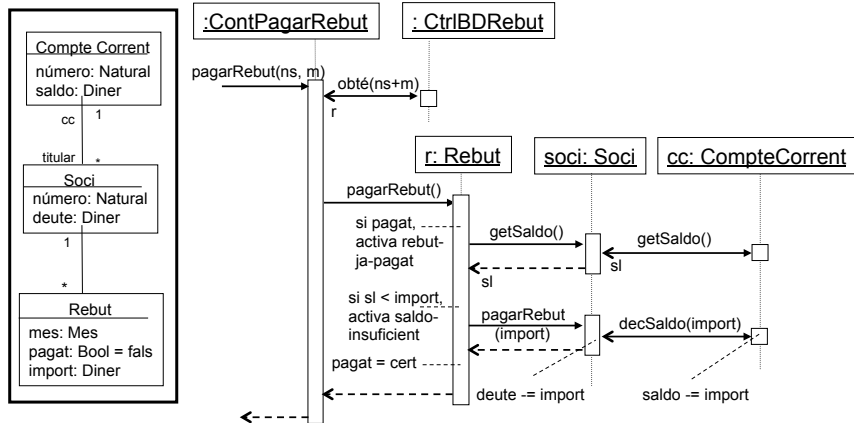
**post:**

paga-rebut: l'atribut *pagat* de *Rebut* es posa a cert

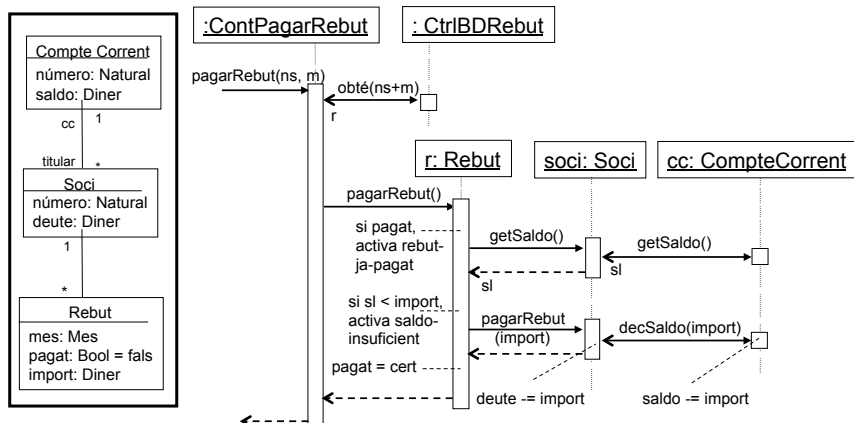
decrementa-saldo: el saldo del compte corrent del soci es decrementa en l'import del rebut

decrementa-deute: el deute del soci es decrementa en l'import del rebut

### Exemple de gestió de la persistència Solució sense considerar persistència

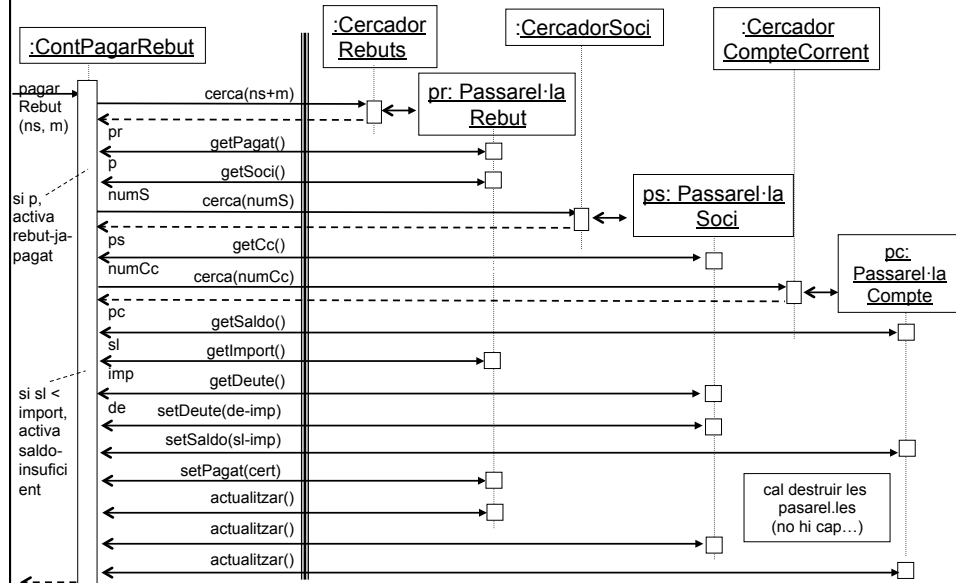


### Exemple de gestió de la persistència Solució amb generació automàtica de persistència



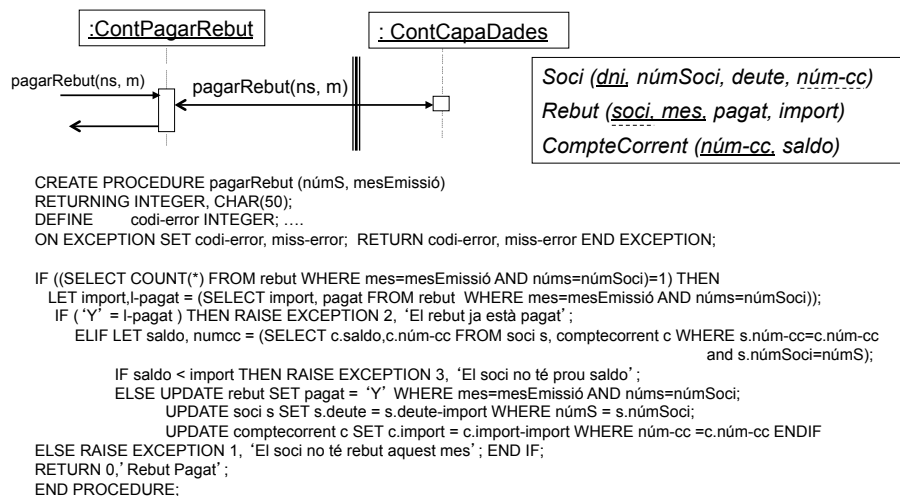
### Exemple de gestió de la persistència

#### Solució amb gestió directe de la persistència, pasarel·la fila



### Exemple de gestió de la persistència

#### Solució amb gestió directa de la persistència, procediment emmagatzemat



### **Disseny de les capes de domini i gestió de dades, resum**

- Els patrons de les capes de domini i de gestió de dades estan relacionats:
  - Domain Model + Data Mapper
    - variant: Domain Model amb consultes a la capa de dades
  - Transaction Script: permet dues variants
    - Transaction Script + Pasarel.la fila
      - Variant: Pasarel.la fila amb consultes arbitràries
    - Transaction Script + Procediment emmagatzemats
    - Poden combinar-se en el mateix sistema
- En qualsevol opció, es pot implementar directament en l'esquema de la BD les responsabilitats assignades a la capa de gestió de dades:
  - Provenients de restriccions d'integritat i informació derivada (e.g., claus)
  - Provenients dels contractes d'especificació (e.g., esborrat en cascada)
  - En el cas de Data Mapper, depèn del que la tecnologia emprada permeti fer

### **Bibliografia**

- M. Fowler  
Patterns of Enterprise Application Architecture  
Addison-Wesley, 2003
- H. Garcia-Molina, J. Ullman, J. Widom  
*Database Systems Implementation*  
Prentice-Hall, 2000.
- *Understanding SQL and Java Together*  
J.Melton, A.Eisenberg  
Morgan-Kaufmann, 2000.
- *Hibernate in Action*  
Christian Bauer, Gavin King  
Manning Publications Company, 2004