

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



De l'esquema d'especificació al de disseny

Traducció de l'esquema conceptual al diagrama de classes de disseny
Incorporació de les restriccions d'integritat als contractes de les operacions
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:

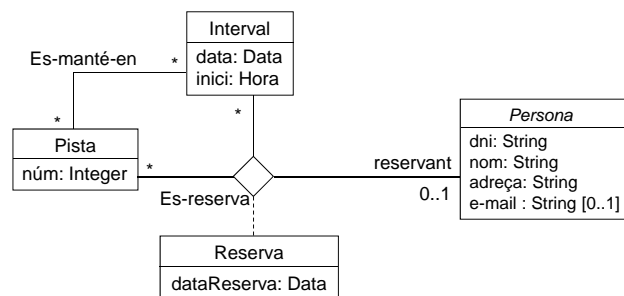
- *eliminar associacions n-àries i classes associatives*
- *controlar les restriccions d'integritat*
- *tractar la informació derivada*

que impacta tant al diagrama de classes com als contractes de les operacions.

3

Eliminació d'associacions n-àries i classes associatives

Esquema conceptual d'especificació:



Restriccions d'integritat textuais:

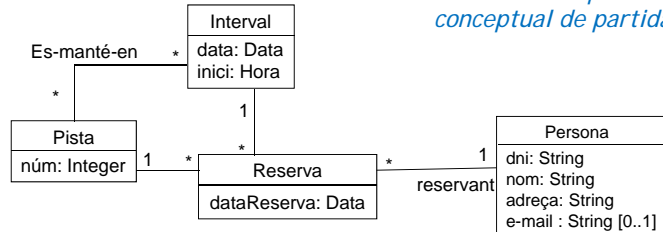
- claus classes no associatives: (Pista, núm); (Interval, data + inici); (Persona, dni)

4

Obtenció del diagrama de classes de disseny

Diagrama de classes de disseny:

el resultat obtingut ha de tenir la mateixa semàntica que l'esquema conceptual de partida



Restriccions d'integritat textuals:

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

~~..... (Afe gida) No hi pot haver dues reserves amb els mateixos Pista, interval i Persona~~

- ↑ - (Afe gida) 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

5

Incorporació de les restriccions d'integritat als contractes

Els models d'especificació són no redundants entre ells

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



cal modificar els contractes de les operacions perquè controlin totes les restriccions d'integritat

com a conseqüència, 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

6

Disseny per Contracte

Pre/postcondicions i la fórmula de correctesa

- Tota operació A del sistema té:
 - una precondició P
 - una postcondició Q

on P i Q són assercions tals que se satisfà l'anomenada *fórmula de correctesa*:

$$\{P\} A \{Q\}$$



"Qualsevol execució d' A en un estat que satisfà P acabarà en un estat que satisfà Q ."

- Els contractes garanteixen la **fiabilitat del software** mitjançant:
 - *precondicions*: condicions que estan garantides quan es crida l'operació
 - *postcondicions*: condicions que l'operació ha de garantir

*no representen el mateix
que a l'especificació*

7

Disseny per Contracte

Contractes de les operacions de disseny

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

Precondicions:

Condicions que estan garantides quan es crida l'operació.

Excepcions:

Condicció que impedeix l'execució de l'operació quan no se satisfà.

Postcondicions:

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

Sortida:

Descripció de la sortida que proporciona l'operació

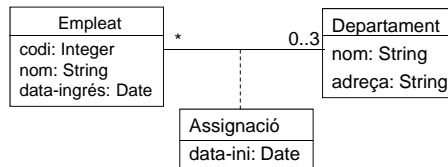
Observeu que:

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

8

Control de les restriccions d'integritat: exemple

Esquema conceptual d'especificació:



R.I. Textuals:

- Claus: (Empleat, codi); (Departament, nom)
- Data-ini d'assignació \geq 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)

Precondició:

- L'empleat codiEmp i el departament nomDept existeixen

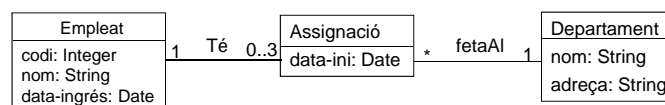
Postcondició:

- Es dona d'alta l'associació entre l'empleat i el departament.

9

Exemple: obtenció del diagrama de classes de disseny

Diagrama de classes de disseny:



R.I. Textuals:

- Claus: (Empleat, codi); (Departament, nom)
- Data-ini d'assignació \geq data-ingrés
- Un empleat no pot estar assignat alhora als departaments de Vendes i de Control-Vendes
- **(Afegeida)** No hi pot haver dues assignacions amb els mateixos Empleat i Departament

10

Exemple: obtenció dels contractes de disseny

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

11

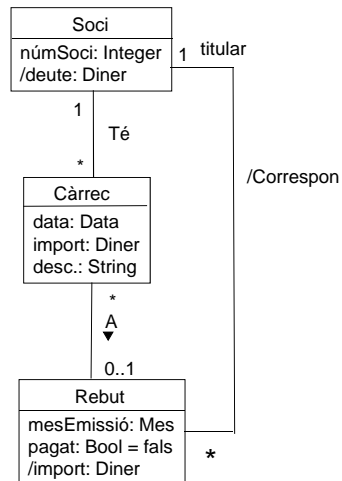
Tractament de la informació derivada

- Els atributs i les associacions derivats es poden:
 - *Calcular* quan es necessiten.
 - *Materialitzar*
- 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** els **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.

12

Tractament de la informació derivada: exemple

Esquema conceptual:



• Cal decidir si calculem o materialitzem la informació derivada.

• Decidim:

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

Operació: ferRebut(m: Mes)

Precondició: no hi ha cap rebut del mes m

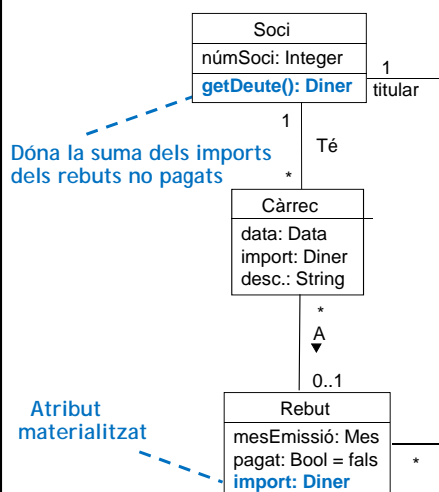
Postcondició:

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

13

Tractament de la informació derivada: resultat

Diagrama de classes de disseny resultant:



Contracte de disseny:

Operació: ferRebut(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 i
amb import igual a la suma dels imports
d'aquests càrrecs
s'associa el rebut creat amb el soci

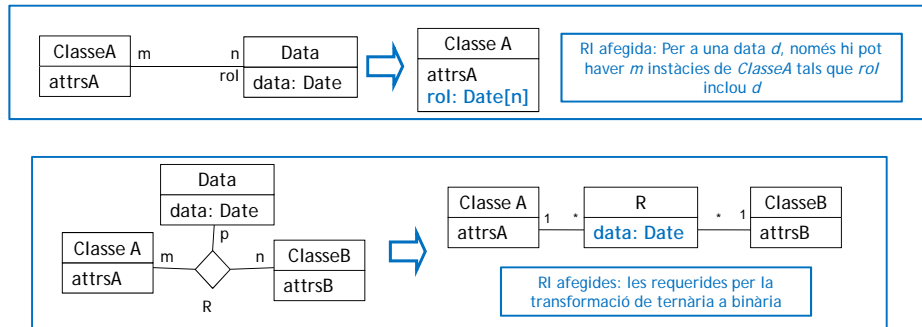
Associació
materialitzada

14

Patró Domain Model: tractament de la classe Data

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



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

15

Exemple

Especificació:

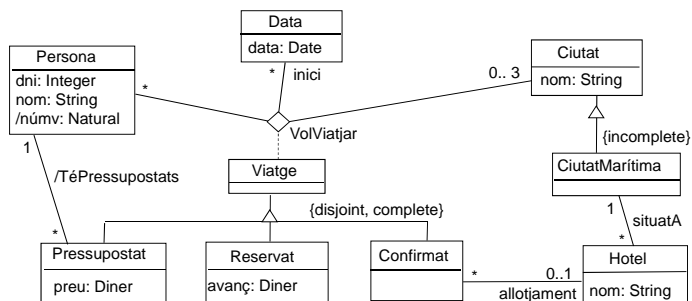
- Esquema conceptual de les dades
- Esquema del comportament

Obtenció de l'esquema de disseny:

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

16

Especificació: Esquema conceptual de les dades



R.I. Textuals:

- Claus classes no associatives: (Persona, dni); (Data, data); (Ciutat, nom)
- Una Ciutat 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

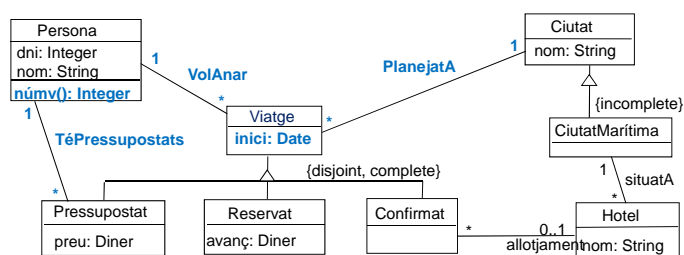
Suposicions:

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

17

Diagrama de classes de disseny

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

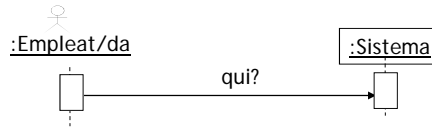


R.I. Textuals:

- 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

18

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.

19

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

20

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

21

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
- `ciutatNoMarítima`: la ciutat nomCiutat no és de Mar
- `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).

22

Bibliografia

- Pressman, R.G. *"Software Engineering. A Practitioner's Approach"*, Mc Graw-Hill, 2005 (7a 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.