Ampliació a l'enginyeria del programari

Principis del bon desenvolupador

Què hi ha en aquest material

- Principis que tot i no ser propis del disseny li són del tot aplicables
  - Principis de modelització
  - Principis generals de desenvolupament de software

Principis de modelització

Definició

El concepte com a model

Model: Representació de les propietats rellevants d'allò que volem estudiar

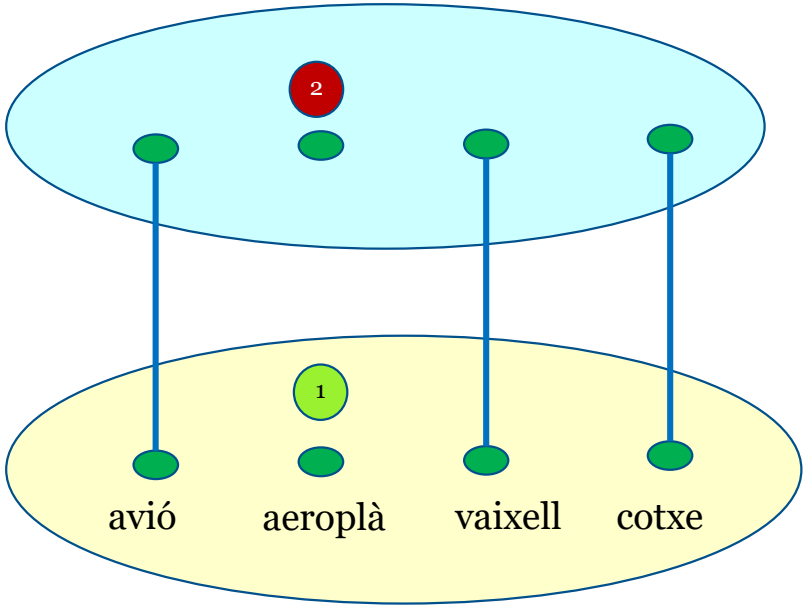
Propietats d'un model

- Focalització
  - Representem només allò que interessa pel nostre estudi
    - Per tant, el model depèn de les necessitats de l'estudi
- Simplicitat
  - Cerquem la representació més simple
    - Fins i tot a canvi d'irrealitat o errors afitats
- Abstracció
  - En termes genèrics
    - Independentment de les circumstàncies o contextos particulars

Espill

6

En la construcció d'un model ens **emmirallem** en la realitat modelada; és a dir, intentem que cadascun dels conceptes, idees, objectes, etc. del model es corresponguin un a un amb conceptes, idees, etc. de la realitat modelada.



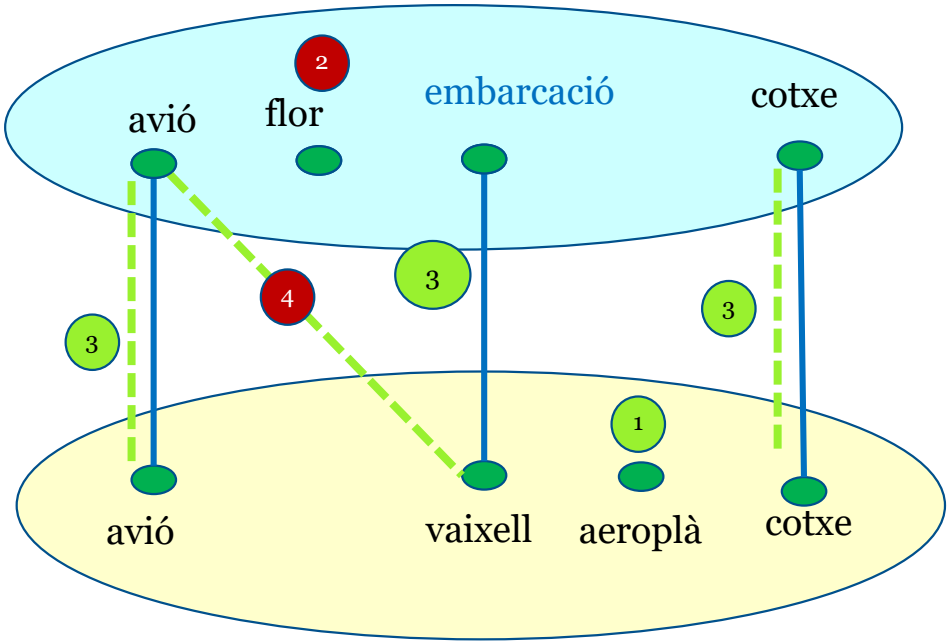
Lligam semàntic

1. Realitat no modelada
2. Viola espill

Principi

Franquícia

- Els elements del model **poden** tenir el mateix nom que un dels elements de la realitat modelada
  - Cal però un correlat semàntic entre els elements que comparteixen nom



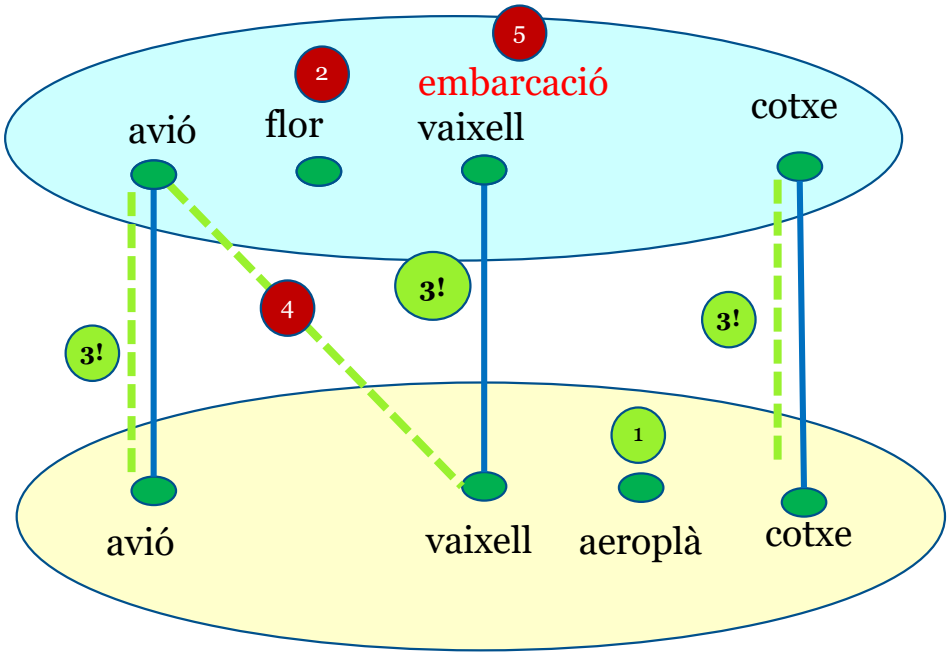
Lligam semàntic

Homonímia

1. Realitat no modelada
2. Viola espill
3. Franquícia
4. Viola Franquícia

Franquícia obligada

Si en el model podem usar un nom de la realitat modelada, **l'hem** d'usar



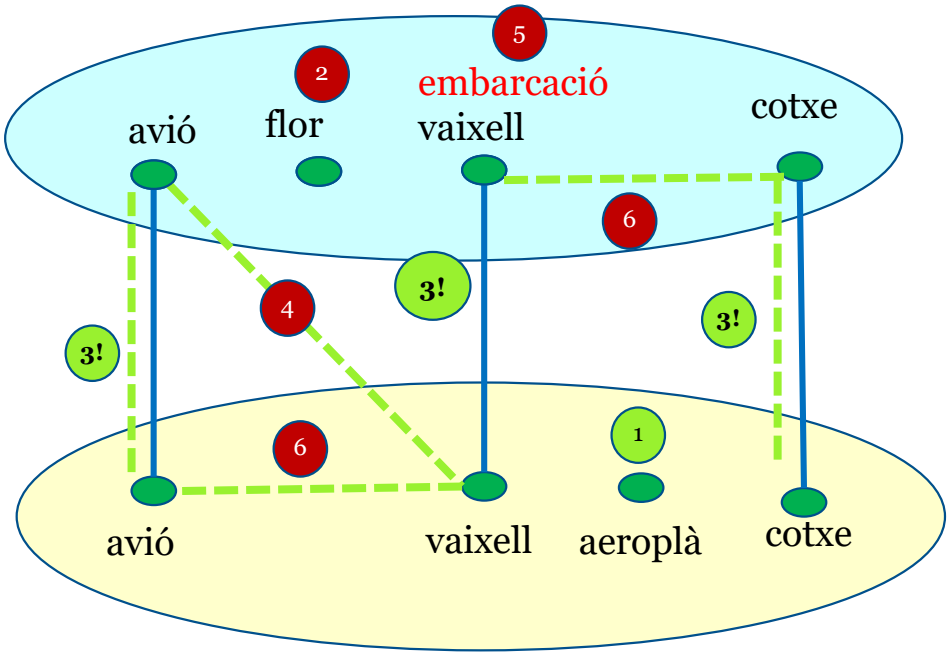
Lligam semàntic

Homonímia

1. Realitat no modelada
2. Viola espill
3. Franquícia
4. Viola Franquícia
5. Viola Franquícia obligada

Referent únic

L'homonímia és sinonímia

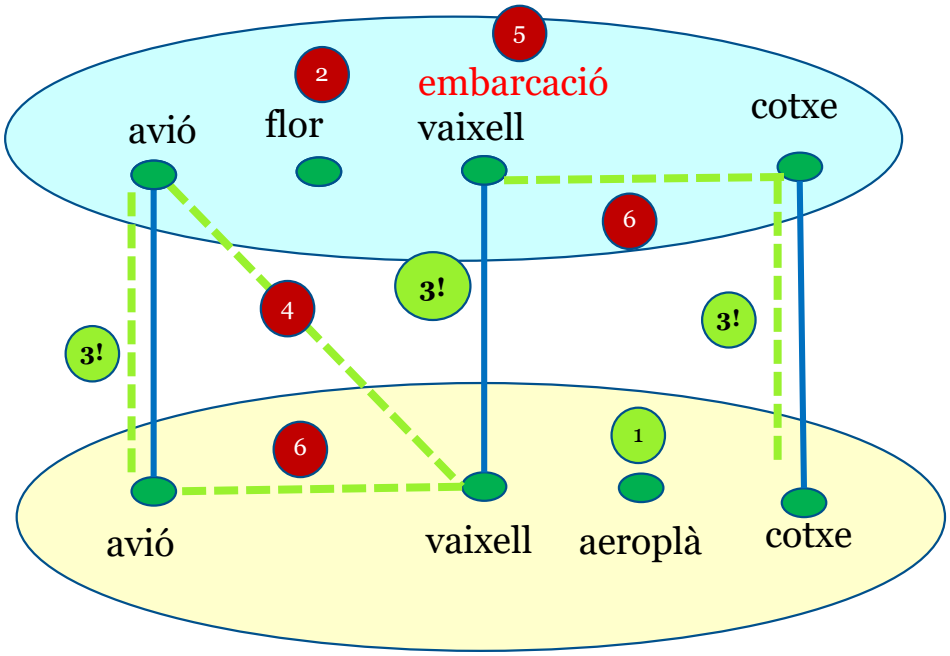


Lligam semàntic

Homonímia

1. Realitat no modelada
2. Viola espill
3. Franquícia
4. Viola Franquícia
5. Viola Franquícia obligada
6. Viola Referent únic

Resum: Principis de modelització



Lligam semàntic

Homonímia

1. Realitat no modelada
2. Viola espill
3. Franquícia
4. Viola Franquícia
5. Viola Franquícia obligada
6. Viola Referent únic

Principi

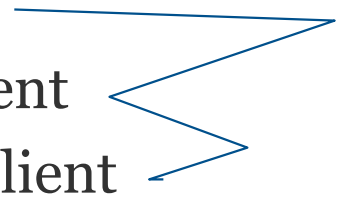
Modelització contínua

- Els noms i elements usats en el model s'han de correspondre amb noms i elements de la realitat modelada

- **Modelització contínua** = Franquícia obligada + Referent únic
  - Cada nom només significa una cosa
  - Els noms es reutilitzen en els diferents nivells de modelització
  - Els noms d'un nivell s'obtenen del nivell inferior

Conseqüències de la Modelització contínua

- La modelització no és trencament
  - Isomorfisme entre model i realitat
- Traçabilitat de la modelització
  - Existeix un camí únic des d'un element d'un nivell de modelització a l'element modelat
- Exemple
  - Anàlisi: Concepte Client
  - Disseny: Component Client
  - Implementació: Classe Client



Traça

Principis de desenvolupament de software

Principi

Encapsulament

- Cada unitat de la solució software té uns **límits ben definits**:
  - Sabem on comença, on acaba i amb qui interactua

- Certa localitat de la comprensió
  - Divideix i vençs
  - Fora el “codi spaghetti”

Principi

Ocultació

- Cada unitat de la solució software **oculta** aquells detalls de la seva construcció que són susceptibles de canvis

- Dues perspectives
  - Ús i construcció
- Dos actors
  - Client i servidor

Problema:
Com sabem quins són els detalls susceptibles de canvi?

- Perspectiva i coneixement
  - Les necessitats de coneixement depenen de la perspectiva
    - Client i servidor tenen coneixements diferents
- Localitat dels canvis
  - L'ocultació evita la propagació dels canvis

Interfície

- Descripció de com usar una unitat de la solució software
  - Totes les possibilitats d'ús
  - Manual d'instruccions
- **Exemple**
  - Prototipus de les capçaleres C
    - El nom fictici dels arguments pròpiament no forma part de la **interfície**
    - El flux de la informació no s'indica ni directament ni explícita

Principi

Interfície obligada

- L'ús de cada unitat de la solució software està perfectament delimitat i explicitat

- És a dir, cada unitat de la solució software té una interfície definida
- Avantatges
  - No ens cal llegir el disseny o el codi per saber com utilitzar-la
  - Documentació integrada

Contracte

- Descripció explícita del comportament i de l'ús (com usar i en quins contextos) una unitat de la solució software
  - Manual del bon d'ús
  - Manual del quan i el perquè en cal el seu ús
- **Exemple**
  - Contracte PRE/POST
- Avantatges
  - No ens cal llegir el disseny o el codi per saber-ne la funcionalitat
  - Documentació integrada

- **Interfície**: Descripció sintàctica
- **Contracte**: Descripció semàntica contextual

Definició

Contracte PRE/POST

- Contracte que s'expressa mitjançant els següents elements:
  - **Interfície**
  - Explicació **semàntica** del comportament i del significat de la informació manipulada per la interfície
  - Condicions contextuals d'ús correcte (**PRE**)
  - Condicions resultants d'un ús correcte (**POST**)

Principi

Desenvolupament sota contracte

- El **client** només s'ha de preocupar d'assegurar les **PRE**
- El **servidor** només s'ha de preocupar d'assegurar les **POST**

- Assignació de responsabilitats diferents
- Delimitació de responsabilitats
  - Reducció de la complexitat
- Confiança mútua
  - Col·laboració

Definició

Caixa blanca

- Unitat software encapsulada i amb interfície ben definida
- Caixa
  - Límits definits (Encapsulament)
- Blanca o transparent
  - Podem conèixer el seu interior

Ocultació total

- El **client** només coneix els **usos** de la unitat software

- Desapareix el problema de decidir què ocultem i què no

Definició

Caixa negra

- Caixa blanca amb ocultació total (l'únic visible és la seva interfície)
- Principis usats en les caixes negres:
  - Encapsulament
    - Límits ben definits
  - Interfície obligada
    - Usos ben definits
  - Ocultació total
    - Desconeixem l'ínterior. L'únic que coneixem és la interfície

Desenvolupament amb caixes negres

- Independència entre l'ús i al construcció
  - Ús amb desconeixement
  - Construcció aïllada

Definició

Component

Caixa negra amb contracte explícit

Desenvolupament amb components

- La doble confiança simplifica els desenvolupaments
  - El contracte comunica al client i al servidor què s'espera de cadascun d'ells
- Desenvolupament independent
  - No cal tenir el servidor per usar la caixa negra
    - Només cal assegurar les PRE
  - No cal tenir el client per construir la caixa negra
    - Només cal assegurar les POST

Component = Caixa negra + Contracte

Avantatges del desenvolupament amb components

- Permet fàcilment el desenvolupament descendent i modular, entre d'altres
- Facilita la divisió de tasques
  - Els contractes es converteixen en l'element comunicatiu
- La documentació forma part del desenvolupament
  - Els contractes són una documentació fonamental

Principis i regles

Principis i regles

- Els **principis** són **recomanacions**
- Les **regles** són **obligacions**

- Principis d'obligat compliment (regles)
  - Caixa negra com a única unitat
    - Encapsulament
    - Ocultació total
    - Interfície obligada
  - Contracte explícit
    - Contracte PRE/POST