

Ampliació a l'enginyeria del programari

Ús dels identificadors

Què hi ha en aquest material

2

- Amb els principis de les **Connexions**, la **Cistella** i l'**Encaixada** hem delimitat l'**ús** dels identificadors
- Ara estudiem les **propietats** que ha de tenir el **sistema** per poder usar els identificadors
 - **Identificació referencial**
 - ✦ Tot identificador permet recuperar un objecte
 - **Comprovació d'identificador**
 - ✦ Mecanisme per comprovar si un possible valor d'identificador permetrà recuperar un objecte
 - **Identificació robusta**
 - ✦ Per tot identificador visible a l'exterior cal oferir un mecanisme de comprovació d'identificador
- I què ha de complir el sistema per assegurar aquestes propietats
 - **Globalitat accessible**
 - ✦ Tot identificador global visible des de l'exterior es modelitza amb un repositori centralitzat
 - **Àmbits accessibles**
 - ✦ Tot identificador local que és l'argument d'un ES ha d'anar acompanyat d'un altre argument que representi el seu context de localitat

Exigències dels identificadors

3

**RECUPERACIÓ REFERENCIAL
IDENTIFICACIÓ ROBUSTA**

El sistema disposa de mecanismes de recuperació referencial per tots i cadascun dels identificadors usats

- El principi de la **identificació referencial** diu que si en algun moment usem un identificador és perquè tenim un mecanisme per recuperar l'objecte referenciat pel valor donat de l'identificador
- És a dir, els identificadors fan allò que s'espera d'ells

Mecanisme de comprovació d'identificador

Definició

5

És un mecanisme ofert pel sistema que donat un identificador ens diu si el valor donat pot ser correcte o no com a valor d'identificació

- En cas de tractar-se d'un valor d'identificació vàlid, el principi de la **Identificació referencial** assegura que el sistema ens ofereix els mecanismes necessaris per a recuperar l'objecte en qüestió

Comprovació dels id de les PRE

6

- Per tal que l'usuari del sistema pugui comprovar les PRE dels ES, cal que el sistema ofereixi un **mecanisme de comprovació d'identificador per cadascun dels identificadors que apareixen com a argument d'un ES**
 - El valor d'identificació donat és vàlid
 - ✦ Exigim la no existència d'un altre objecte amb aquest valor d'identificació, o bé
 - ✦ Exigim l'existència d'un altre objecte amb aquest valor d'identificació

El sistema ha d'assegurar que per tot identificador que aparegui en un ES hi ha un **mecanisme de comprovació d'identificador.**

- El principi de la *Identificació robusta*, juntament amb el de la *Identificació referencial*, diu que els únics **sistemes d'identificació útils**, de cara a la comunicació entre el sistema i l'exterior, són aquells que **permeten assegurar la recuperació referencial** i fer les **comprovacions** necessàries sobre les PRE dels ES.

Repàs: Els models construïts fins ara

8

- **Model 1**

- **GLS**

- ✦ Responsabilitats
 - Controlador
 - Creador orb d'Inscripció
- ✦ Emmagatzematge
 - Repositori centralitzat d'Inscripcions
 - Inscripció activa

- **Caminada**

- **Inscripció**

- ✦ Visibilitat d'atribut directa de la Caminada

- **Model 2**

- **GLS**

- ✦ Responsabilitats
 - Controlador
- ✦ Emmagatzematge
 - Caminada activa

- **Caminada**

- ✦ Responsabilitats
 - Creador canònic d'inscripció
- ✦ Emmagatzematge
 - Repositori de les inscripcions pròpies
 - Inscripció activa

- **Inscripció**

Repàs: Flux comunicatiu

9

- Què és caminada a novaInscripció (caminada)?

- **Objecte**

- ✦ Cal que la recuperació exhaustiva sigui visible en la interfície del sistema

- **Missatge generat per l'usuari del sistema (ES):**

`obtenirCaminada(id: IdCaminada): Caminada`



Connexions

Hem usat recuperació referencial, però no cal

- **Identificador**

- ✦ El controlador ha de poder fer la recuperació referencial

- **Missatge generat pel controlador:**

`obtenirCaminada(id: IdCaminada): Caminada`



Encaixada

Independentment de la decisió cal dissenyar la recuperació (exhaustiva o referencial)

Robustesa dels identificadors (1)

10

- En el cas que caminada a novaInscripció (caminada) sigui un **identificador**, el sistema ha d'oferir un mecanisme de **comprovació d'identificador**



Identificació robusta

- Exemple

- ✦ `existeixCaminada(id: IdCaminada): bool`

- {POST: retorna `cert` si i només sí en el repositori de caminades actiu hi ha una `c:Caminada` amb `c.id=id`}

Robustesa dels identificadors (2)

11

- Si caminada és un **objecte** el principi de les **connexions** exigeix que la **recuperació exhaustiva** sigui visible en la interfície del sistema
 - Si usem la **recuperació referencial** llavors apareix un ES amb un **identificador** com a argument
 - ✦ `obtenirCaminada(id: IdCaminada) : Caminada`
 - La **identificació robusta** exigeix un mecanisme de **comprovació d'identificador** sobre aquest identificador

Conclusions: Robustesa dels identificadors

12

Segons la decisió que haguem pres sobre què és caminada com a argument de l'ES `novaInscripció` tenim:

• Identificador

○ GLS

- ✦ Responsabilitats
 - Recuperació referencial
 - Comprovació d'identificador
- ✦ Emmagatzematge
 - Repositori centralitzat de caminades
- ✦ Interfície
 - Comprovació d'identificador

• Objecte

○ GLS

- ✦ Responsabilitats
 - Recuperació exhaustiva
 - Comprovació d'identificador 1
- ✦ Emmagatzematge
 - Repositori centralitzat de caminades 1
- ✦ Interfície
 - Recuperació exhaustiva
 - Comprovació d'identificador 1

En el cas que la recuperació exhaustiva sigui una recuperació referencial 1

Un missatge, dues responsabilitats

13

- Sovint la recuperació referencial i la comprovació d'identificador es resolen amb un sol missatge
- **Exemple**
 - El `find(id)` dels multiobjectes retorna `null` quan dins el multiobjecte no hi ha cap objecte amb l'identificador donat
 - Malgrat tot sempre que sigui possible exercirem les dues responsabilitats amb missatges diferenciats
 - El disseny és més clar i simple

Contracte robust

14

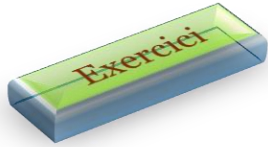
Sobre la robustesa

15

- El principi de la **Identificació robusta** és un cas particular d'un principi més general: el principi del **Contracte robust**
- La idea és que tot el que es demana a una PRE que depèn de l'estat intern del domini, la capa de presentació pugui preguntar al domini si es compleix o no
 - La **Presentació** és qui ha d'assegurar les PRE
 - El **Domini** ofereix les facilitats necessàries per tal que la Presentació pugui comprovar si les condicions són les correctes

- Per tota **PRE** que depengui de l'**estat intern** del sistema, cal oferir a la interfície del sistema els mecanismes necessaris per a **comprovar** si es compleix la condició

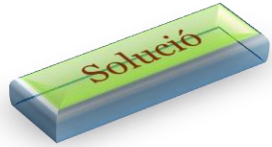
D'aquesta manera, la **Presentació** pot col·laborar amb el **Domini** per tal d'assegurar la seva part del **contracte** (és a dir, que es compleixen les PRE en el moment de fer la petició)



Contracte robust

17

- Suposem les següents PRE, d'ES independents. Què ens demana el **Contracte robust**?
 1. Hi ha algun client
 2. Existeix un client amb més de 10 factures pendents
 3. L'albarà conté algun producte descatalogat
 4. Hi ha un client amb el valor d'identificació donat
 5. Hi ha suficient estoc del producte demanat



Existència d'un client

18

- N'hi ha prou amb el següent ES:
 - `existeixenClients () : <Bool>`
 - ✦ PRE: Cert
 - ✦ POST: Retorna cert si i només si en el sistema hi ha algun client



Client amb factures pendents: criteris

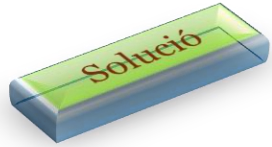
19

- PRE: Existeix un client amb més de 10 factures pendents
- Ara tenim diferents possibilitats
 - Preguntar tota la PRE, tal qual
 - ✦ Molt poc flexible i reusable
 - Preguntar tota la PRE, però després de parametritzar-la
 - ✦ Millora la flexibilitat i la reusabilitat
 - Demanar la informació mínima per tal que la Presentació pugui fer la comprovació
 - ✦ Dificulta la tasca de la presentació

x, enlloc de 10

Demanem al domini el nombre de factures pendents del client

La Presentació ha de poder iterar sobre tots els clients (recuperació exhaustiva iterable)



Client amb factures pendents: proposta

20

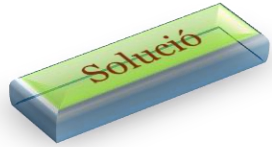
- **Tota la PRE**
 - ES `existeixClientAmbDeuFacturesPendents ()`
 - ✦ PRE: Cert
 - ✦ POST: Cert si i només si hi ha un client amb un mínim de 10 factures pendents
- **Tota la PRE parametritzada**
 - ES `existeixClientAmbXFacturesPendents (x:nat)`
 - ✦ PRE: $n \geq 0$
 - ✦ POST: Cert si i només si hi ha un client amb un mínim de 10 factures pendents
- **Informació necessària per fer la comprovació**
 - Conjunt d'ES que permetin iterar sobre els clients, i
 - ES `nombreFacturesPendents (client)`
 - ✦ PRE: Cert
 - ✦ POST: retorna el nombre de factures pendents que té el client donat
 - ✦ La Presentació haurà d'iterar sobre tots els clients, i comprovar el valor obtingut amb el valor 10.



Productes descatalogats

21

- A l'igual que en el cas anterior tenim diferents possibilitats:
 - `conteProductesDescatalogats` (albarà)
 - `nombreProductesDescatalogats` (albarà)
 - ✦ Per assegurar la PRE cal assegurar que el valor retornat no sigui 0
 - `ésDescatalogat` (producte)
 - ✦ S'ha de combinar amb una recuperació exhaustiva iterable dels productes d'un albarà
 - ✦ Per assegurar la PRE cal fer una cerca sobre els productes de l'albarà i veure si algun està descatalogat



Comprovació d'identificació

22

- Cal un mecanisme per assegurar si el valor donat permet recuperar un objecte
 - `existeixClient(idClient)`
- Això és justament el que demana el principi de la **Identificació robusta**



Estoc suficient

23

- L'ES del nostre CU és:
 - novaCompra (producte, quantitat)
- Per poder comprovar la PRE, cal oferir a la Presentació algun dels següents ES:
 - estoc (producte)
 - ✦ PRE: Cert
 - ✦ POST: Retorna la quantitat d'unitats en estoc del producte indicat
 - estocSuficient (producte, quantitat)
 - ✦ PRE: $quantitat > 0$
 - ✦ POST: Retorna cert si la quantitat d'unitats en estoc del producte indicat és superior o igual a la quantitat donada

Dos principis fonamentals

24

**ÀMBITS CONEGUTS
GLOBALITAT ACCESSIBLE**

Què hem de poder fer amb els identificadors

25

- **Encaixada**

- La primera responsabilitat que s'ha de resoldre, just després de la captura de l'ES, és la d'**encaixar** els arguments identificadors
 - ✦ Simplificant això significa que el controlador ha de poder fer la recuperació referencial dels arguments identificadors

- **Identificació robusta (Contracte robust)**

- Per cada argument identificador d'un ES cal oferir a la interfície del sistema un mecanisme de **comprovació d'identificació**

- **Recuperació referencial**

- Per tot identificador (sigui o no un argument d'un ES), el sistema ofereix un mecanisme de **recuperació referencial** sistema

No és tant el que hem de fer

26

- Si tenim **Recuperació referencial**, l'encaixada és possible. Una altra cosa és si la pot fer el controlador o no
- Si tenim **Recuperació referencial**, la **comprovació d'identificació** és factible
 - Intentem la recuperació; si és exitosa retornem cert; si falla, retornem fals
 - Una altra cosa és fer visible aquest mecanisme de comprovació
- Si assegurem que tot identificador té una **recuperació referencial**, ja ho tenim tot
 - El que manca és fer la petició al més aviat possible (**Encaixada**) o oferir el mecanisme en termes d'ES (**Identificació robusta**)

Com ho fem

27

- Ara cal assegurar que es pot fer allò que cal poder fer
 - Com hem vist, només ens hem de preocupar de la recuperació referencial
- Limitem l'anàlisi als identificadors que apareixen com a argument d'un ES
- Els identificadors que apareixen com a argument d'un ES han de ser:
 - D'àmbit local o d'àmbit global
- **Àmbits coneguts**
 - En cas d'identificació d'àmbit local, la localitat també és un argument (explícit o implícit)
- **Globalitat accessible**
 - Cal un repositori centralitzat

- Tot identificador que és **argument d'un ES**:
 - o bé és **global**
 - o bé el seu **context de localitat és un argument** (implícit o explícit) del mateix ES
 - ✦ El context de localitat es pot donar en forma d'objecte o d'identificador

novaInscripció (idCaminada)

29

- No hi ha cap argument implícit
- L'únic argument explícit és un identificador
- Per tant podem concloure que `idCaminada` és el valor d'identificació d'una **identificació global** de `Caminada`

novaParticipació (nom)

30

- Assumim que `nom` és l'identificador de la `participació`
- Hi ha un **argument implícit**:
 - Hi ha una `i:Inscripció` activa
- Per tant tenim dues possibilitats:
 1. `nom` és un identificador d'**àmbit global** de la `Participació`
 - ✦ Impedeix que dues *Aines* diferents participin en les caminades
 2. `nom` és un identificador d'**àmbit local** a la `i:Inscripció`
 - ✦ Semànticament és millor solució: dins d'una inscripció no admetem noms repetits; però sí en inscripcions diferents

obsequi (idInscripció, nom) (1)

31

- Assumim que no hi ha cap **argument implícit**
- Tenim dues possibilitats:
 1. Ambdós arguments són **identificadors globals**
 - ✦ Considerar el nom de la participació com un identificador global no té massa sentit
 2. **Inscripció** és **global**; nom (Participació) és **local** a la **inscripció**
 - ✦ El context de localitat de la Participació (és a dir, del valor nom) és la inscripció, que és un argument de l'ES
- Possibilitats descartades
 1. **Inscripció** és **local**
 - ✦ El context de localitat no pot ser la Participació (que és l'altre argument) ja que la participació sempre apareix en un temps posterior a la inscripció
 - ✦ No hi ha cap altre context de localitat possible perquè no tenim més arguments (ni implícits ni explícits)

obsequi (idInscripció, nom) (2)

32

- Assumim que hi ha una caminada com a **argument implícit**
- Tenim les següents possibilitats:
 1. **Inscripció** és **global**; **nom** és **local a la inscripció**
 - ✦ El context de localitat de la Participació (és a dir, del valor nom) és la inscripció, que és un argument de l'ES
 2. **Inscripció local a la caminada**; **nom local a la inscripció**
 - ✦ Un context de localitat és un argument implícit; l'altre és un argument explícit
 3. **Inscripció local a la caminada**; **nom local a la caminada**
 - ✦ Ambdós contextos de localitat són un argument implícit
 - ✦ Un mateix nom no pot aparèixer ni en una mateixa inscripció, ni en dues inscripcions de la mateixa caminada
- Possibilitats descartades
 1. **Participació d'àmbit global**
 - ✦ Considerar el nom de la participació com un identificador global no té massa sentit
 2. **Inscripció d'àmbit local a la Participació**
 - ✦ El context de localitat no pot ser la Participació (que és l'altre argument) ja que la participació sempre apareix en un temps posterior a la inscripció

- Per cada argument d'un ES que és un identificador d'àmbit **global** ha d'existir un **repositori centralitzat** visible (via monoavaluació) des del controlador

- ES: $m(\text{id}A) \Rightarrow K \xrightarrow{*} A^*$

El **controlador** o és l'origen d'un repositori centralitzat de l'objecte identificat **globalment**, o bé té una **visibilitat indirecta** sobre aquest **repositori centralitzat**

Demostració de la utilitat dels principis

34

Camí que seguirem

35

- Tot seguit demostrarem que amb els principis de la **Globalitat accessible** i dels **Àmbits coneguts** n'hi ha prou perquè la **recuperació referencial** sempre sigui possible
- Ho farem per inducció
 - Casos base:
 - ✦ Argument identificador d'àmbit global
 - ✦ Argument identificador que el seu context de localitat ve donat per un argument objecte
- Primer presentem un resum de la demostració, i després la fem en detall

Recuperació referencial assegurada: resum previ

36

- Sigui un ES que té un identificador com a argument
 - Si és un **identificador global**, el principi de la **globalitat accessible** assegura que el controlador pot fer (o delegar) l'encaixada
 - Si és un **identificador local**, el principi dels **àmbits coneguts**, assegura que el context de localitat és un argument del mateix ES
 - ✦ Si el context el donem en forma d'identificador, o bé és local, i per tant s'hi aplica un raonament recursiu, o bé és global, i per tant encaixable
 - ✦ Si el context el donem en forma d'objecte vol dir que el controlador té accés a aquest objecte (o bé l'ha rebut com a paràmetre o bé en té visibilitat d'atribut), i per tant se li pot delegar l'encaixada

Identificador global, result

37

- Suposem un ES tal que:
 - Té un argument id_A (que identifica l'objecte $a : A$) que és un identificador d'**àmbit global**
- El principi de la **globalitat accessible** assegura que el controlador pot fer (o delegar) l'encaixada de $a : A$

Identificador local: consideracions prèvies

38

- El principi dels **àmbits accessibles** assegura que tot identificador que és una argument d'un ES o bé és d'àmbit global, o bé ve **acompanyat del seu context**
- Així, en cas d'un identificador d'**àmbit local**, hem d'assumir l'existència d'un altre argument (explícit o implícit) que defineix el **context** d'identificació
- L'argument que defineix el context pot ser un objecte o un identificador

Identificador local, amb un objecte de context

39

- Suposem un ES tal que:
 - Té un argument id_A (que identifica l'objecte $a : A$) que és un identificador d'**àmbit local**
 - Té un argument (implícit o explícit) $b : B$ que és el context d'identificació de id_A
- El controlador té una visibilitat de paràmetre (cas de l'argument explícit) o d'atribut (cas de l'argument implícit) sobre el context d'identificació $b : B$
 - Per tant el controlador pot delegar a $b : B$ la recuperació de $a : A$ a partir d' id_A

Identificador local, amb un identificador de context (1)

40

- Suposem un ES tal que:
 - Té un argument id_A (que identifica l'objecte $a : A$) que és un identificador d'**àmbit local**
 - Té un argument (implícit o explícit) id_B , que identifica **globalment** l'objecte $b : B$
 - $b : B$ és el context d'identificació de id_A
- Com que el context és un identificador global, el principi de la **globalitat accessible** assegura que el controlador en pot fer la recuperació
- Un cop recuperat el context $b : B$, li pot delegar la recuperació de $a : A$ a partir de id_A

Identificador local, amb un identificador de context (2)

41

- Suposem un ES tal que:
 - Té un argument id_A (que identifica l'objecte $a:A$) que és un identificador d'**àmbit local**
 - Té un argument (implícit o explícit) id_B , que identifica **localment** l'objecte $b:B$
 - $b:B$ és el context d'identificació de id_A
- Com que el context és un identificador local, el principi dels **àmbits coneguts** assegura que el context és un dels arguments de l'ES
 - Si aquest segon context és d'àmbit global, ja ho tenim. El recuperem (si no és un objecte); li deleguem la recuperació del context $b:B$; i a aquest li deleguem la recuperació de $a:A$
 - Si aquest segon context és d'àmbit local, apliquem el raonament de manera recursiva
 - ✦ La base de la recursió és que el context és un objecte o un identificador d'àmbit global

Recuperació referencial assegurada

42

Els principis de la **Globalitat accessible** i dels **Àmbits coneguts** asseguruen que el controlador pot resoldre o delegar la **recuperació referencial** dels arguments identificadors dels ES

Conclusions en termes planers

43

- Si seguim els principis de la **Globalitat accessible** i dels **Àmbits coneguts** aconseguim que:
 - El controlador pot fer (o delegar) totes les **encaixades**
 - El controlador pot realitzar (o delegar) les **comprovacions d'identificació**
 - ✦ Principi de la **Identificació Robusta** assegurat

Generalització de la identificació per accés

44

Recordatori sobre principis d'identificació

45

- **Identificació atributiva**

- Tota identificació d'àmbit global es modelitza amb un atribut identificador

- **Identificació per accés**

- Tota identificació d'àmbit local es modelitza amb una visibilitat qualificada

Generalització de la identificació per accés

46

- La identificació per accés s'introdueix davant l'observació que el disseny és un model comunicatiu
 - Només si un objecte és accessible pot intervenir en una col·laboració
 - Identificar vol dir referenciar un objecte; i aquest només és rellevant si és accessible
- En el seu plantejament inicial, però, el principi d'identificació per accés es limitava a parlar de la identificació d'àmbit local
 - La localitat implica un context
 - El context és qui accedeix a l'objecte
- En el cas de la identificació global el principi no en diu res
 - Tot context d'accés és vàlid
- Ara però el principi de la Globalitat accessible assegura que hi ha un repositori centralitzat
 - Hi aquest repositori és el context d'accés

L'accés és la clau

47

- La identificació només té sentit en un accés, és a dir, en una visibilitat
- La identificació d'**àmbit local** es correspon a una **visibilitat qualificada**
 - Principi d'**Identificació per accés**
- La identificació d'**àmbit global** es correspon a un **repositori centralitzat**
 - Principi de la **Globalitat accessible**

COMPTE. Estem parlant d'identificació d'àmbit **local** i **global**; no pas d'identificació **contextual** i **incontextual**