

Ampliació a l'enginyeria del programari

MOMO: Un modisme sobre els multiobjectes

Què hi ha en aquest material

2

- Es repassa el concepte de multiobjecte
 - No hi ha material nou. És una relectura del que ja s'ha vist
- **MOMO**: Modisme de l'objecte multiobjecte
 - S'introdueix un modisme per resoldre els problemes que, en la majoria de llenguatges de programació, el mecanisme dels multiobjectes genera en la implementació
- **Creació primerenca**
 - Els objectes s'han de crear sencers, amb tots els seus atributs i visibilitats d'atribut
- **Creació primerenca i MOMO**
 - En crear l'origen d'una visibilitat d'atribut multiavaluada cal crear un objecte que simula el multiobjecte

El concepte multiobjecte

3

Multiobjectes: repàs

4

- Un **multiobjecte** és una **ficció** o artifici que facilita el treball amb les visibilitats multiavaluades
 - Comunicació o col·laboració
 - ✦ Permet usar enllaços dirigits multiavaluats en les col·laboracions, sense necessitat d'explicitar-ne els objectes que formen la destinació
 - Simplicitat
 - ✦ Permet usar la visibilitat generalitzada
 - ✦ El missatge arriba a tots els objectes de la destinació

Essència d'un multiobjecte

5

- S'assembla a un objecte

Apareix als **DC**

- Pot rebre i emetre missatges

- ✦ Tot i que distingeix entre els missatges d'enllaç i els missatges de grup

- Es diferencia d'un objecte

No apareix a **MComp**

- No és la realització de cap component

- Ni hi ha cap component que sigui l'abstracció del multiobjecte

En quin model?

6

- Ha d'aparèixer en el model dinàmic
 - Per tal d'expressar les col·laboracions el multiobjecte ha d'aparèixer en els diagrames d'interacció (diagrames de col·laboració o de seqüència)
 - ✦ Justament amb aquest motiu s'han introduït
- **No pot aparèixer a Mcomp**
 - No és un component, sinó que és una realització
 - ✦ Rep i envia missatges
 - Com a ficció no hi ha cap component que sigui la seva abstracció

Parèntesi sobre implementació

7

El salt a la implementació

8

- **I què passa si el meu llenguatge de programació no ofereix el mecanisme dels multiobjectes?**
 - Caldrà **gestionar els missatges enviats** a un multiobjecte

Possibles gestions

9

- **Intrusiva**

- L'emissor del missatge es preocupa d'enviar tants missatges com destinataris

- **Transparent**

- S'introdueix un element de la implementació que simula el comportament del multiobjecte
 - ✦ En rebre un missatge el propaga a tots els objectes que té enregistrats
 - Ex. Classe `Repositori` que conté una `List` dels objectes pertinents

MOMO: Un modisme pels multiobjectes

10

- Conjunt de decisions de disseny que prenen la seva rellevància en l'ús posterior d'una tecnologia o un llenguatge de programació determinat
 - En cas de no usar aquesta tecnologia o llenguatge, les decisions preses per mor del modisme o bé són innòcues o bé són fàcils de retrocedir

La problemàtica dels multiobjectes

12

- Molts llenguatges de programació (LP) no ofereixen el mecanisme dels multiobjectes
- La **gestió transparent** dels missatges exigeix l'aparició d'un element software que fa el paper del multiobjecte
 - Si sabem que usarem un d'aquests LP podem demanar al disseny que consideri, **en el model de comportament**, el multiobjecte com un objecte qualsevol
 - ✦ En concret el disseny s'haurà de preocupar de crear-lo quan calgui

En el model d'estructura (**MComp**) **no apareix cap component** explícit associat a la idea de multiobjecte

- **Modisme de l'Objecte MultiObjecte**
 - Assumim que **els multiobjectes són objectes especials**
 - ✦ No són la realització de cap component
 - Com a objectes el disseny ha de definir el seu cicle de vida
 - ✦ Caldrà crear-los
 - (En cas de llenguatges amb gestió explícita de la memòria dinàmica, com el C++, es poden usar modismes de destrucció dels objectes)

Decisió no local

14

- La decisió de qui és el responsable de crear l'objecte multiobjecte és una decisió **no local**
 - No és local al context d'emissió que usa l'enllaç dirigit multiavaluat a què es correspon el multiobjecte
 - Pot no ser local al component origen de la visibilitat
 - ✦ Penseu en les visibilitats de paràmetre

Àmbit de les decisions

15

- El **disseny** tracta de problemes **globals**
 - L'àmbit de la responsabilitat és local, però l'àmbit de la decisió d'assignació és global
- Les decisions d'**implementació** han de ser **locals**
 - Aquest és un dels sentits de l'afirmació “implementació fàcil” en presentar l'objectiu bàsic del disseny

Avantatges de MOMO

16

- Oportunitat

- L'ús de MOMO elimina la necessitat que la implementació hagi de resoldre el problema de la creació de l'objecte multiobjecte, que és una decisió d'àmbit global
 - ✦ I el disseny tracta les decisions d'àmbit global

- Cost nul

- L'ús de MOMO obliga a què algú envii el missatge `create()` sobre l'objecte multiobjecte
 - ✦ Retrocedir les decisions de MOMO és trivial

Creació de l'objecte multiobjecte

17

CREACIÓ COMPLETA

Una nova operació de grup

18

- **MOMO** exigeix emprar un objecte multiobjecte per cada visibilitat multiavaluada
 - No és la realització de cap component
 - Però el disseny l'ha de crear
- Per expressar això introduïm un missatge de grup sobre els multiobjectes
 - `create()` : crea un multiobjecte buit
 - ✦ Es correspon a un enllaç dirigit mutliavaluat que de moment no té destinacions

Missatges a multiobjectes

19

- Un **missatge de grup** a un multiobjecte s'envia al contenidor, **no es retransmet** als membres de la col·lecció
 - **create()** : creació de la col·lecció representada pel multiobjecte
 - **find()** : buscar un element per clau
 - **add()** : afegir un element a la col·lecció
- Un **missatge d'enllaç** a un multiobjecte arriba a tots i cadascun dels membres de la col·lecció

El problema plantejat per MOMO

20

- Qui crea l'objecte multiojecte exigeix per MOMO?
 - Hem d'analitzar qui l'usa
 - ✦ `create()` crea un multiojecte buit
 - ✦ Per fer un `find()` primer hem d'haver fet un o més `add()`
- **Solució:** En un moment anterior a qualsevol `add()`

En el moment de crear un objecte donem valor a tots els seus atributs i a les seves visibilitats d'atribut

Creació completa i MOMO

22

- La **Creació completa** diu que en crear un objecte que té visibilitat d'atribut multiavaluada, cal crear l'objecte multiobjecte exigint per **MOMO**
 - Simplifica el disseny en no haver d'estar comprovant l'existència de l'objecte multiobjecte
 - Dóna eficiència a la implementació, pel mateix motiu
- El resultat és que assegurarem que en el moment de fer un `add ()` l'objecte multiobjecte ja existeix



M1: Multiobjecte d'inscripcions

23

- El controlador manté un multiobjecte de totes les inscripcions
 - Qui crea l'objecte multiobjecte???
- El controlador emmagatzema
 - L'objecte multiobjecte s'ha de crear en el moment de crear el controlador
- L'objecte multiobjecte s'ha de crear, juntament amb el controlador, en la **inicialització del sistema**



M2: Multiobjecte d'inscripcions

24

- La caminada manté un multiobjecte de les seves inscripcions
 - Qui crea l'objecte multiobjecte???
- La caminada emmagatzema
 - L'objecte multiobjecte s'ha de crear en el moment de crear la caminada
- L'objecte multiobjecte s'ha de crear, juntament amb la caminada, en la creació d'aquesta