



Introducció al disseny descendent

Introducció

Josep Vilaplana

ETSEIB/UPC

Introducció

Els problemes es poden classificar en:

- **Senzills** *Els seus enunciats fan referència a objectes que són directament representables amb els tipus elementals del llenguatge algorísmic, i el que cal fer amb els objectes és fàcilment expressable en l'algorisme corresponent.*
- **Complexos** *Els seus enunciats fan referència a objectes que no tenen una correspondència directa amb els tipus elementals de la notació algorísmica, i/o el que cal fer amb els objecte no és elemental.*

Motivació

Encara que els tipus elementals, accions i funcions elementals de la notació algorísmica són suficients per dissenyar problemes complexos, les seves **solucions** poden resultar **llargues, feixugues, difícils d'entendre i de seguir**, i en conseqüència, propenses a **tenir molts errors difícils de trobar i corregir**.

Motivació

Dos conceptes clau ens ajudaran a no caure en les solucions anteriors:

- **L'abstracció**
- **El mètode de disseny descendent**

A més, aquests dos conceptes hauran de quedar reflectits en el propi disseny de l'algorisme i per aquest motiu parlem de **disseny descendent**.

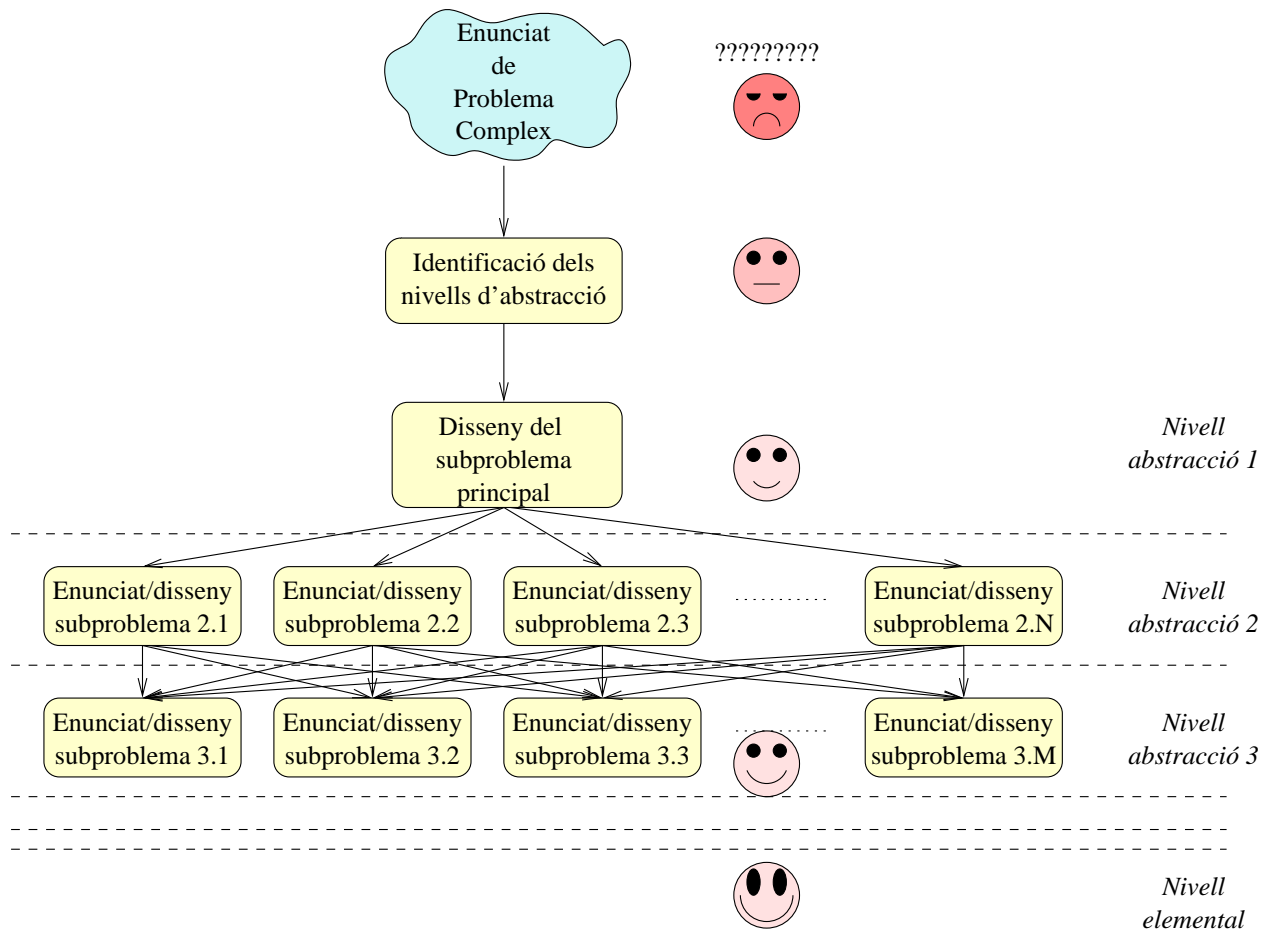
Disseny Descendent

El disseny descendent consisteix en descomposar un problema complex en problemes més senzills. Si és així,

- perquè no el titulem “disseny per parts”?
- d'on surt això de “descendent”?

La paraula “descendent” indica que la descomposició cal fer-la des del nivell més abstracte del problema al més concret i no al revés.

Disseny Descendent



Disseny Descendent

- No tota descomposició d'un problema en subproblemes és adient per obtenir un bon disseny.
- La bondat de la descomposició rau en les abstraccions que convenim adequades pel problema que analitzem i que porten a una descomposició en subproblemes que són molt independents entre sí.

Disseny Descendent

Quan decidim les abstraccions a fer, les podem orientar a:

- **Objectes** del problema no elementals.
Definirem un tipus apropiat a l'objecte i associarem mitjançant accions i/o funcions, les operacions que podrà tenir.
- Tractament del problema (**funcional**). *El tractament del problema convé formularlo en varies etapes funcionals.*

Formalització Disseny Descendent

La notació algorísmica ens dona les eines per definir els tipus nous que ens calen, i les accions i/o funcions noves que necessitem per expressar millor els nostres algorismes.

- Quan utilitzem una acció o una funció en el disseny expressem en l'algorisme QUÈ cal fer en aquell punt.
- Quan dissenyem el cos d'una acció o una funció expressem COM fer el QUÈ ha de fer (el QUÈ es l'especificació de l'acció i/o funció) .
- Per tant, la definició i ús d'accions i/o funcions ens permeten separar problemes i són l'eina excel·lent per aplicar la metodologia del disseny descendent al disseny.

Formalització Disseny Descendent

Els paràmetres comuniquen les dades que calen i els resultats assolits entre problemes aïllats.

- La construcció de nous tipus mitjançant els constructors taula i tupla permeten representar objectes complexos. Si associem a un nou tipus quines operacions podrà fer, (accions i/o funcions que poden treballar sobre l'objecte) augmenta la claredat i control sobre el disseny de l'aplicació complexa.

Correctesa del mètode en el disseny

- L'especificació d'una acció diu QUÈ ha de fer.
- Per saber si una acció és correcta només cal comprovar que el assoleixi la postcondició a partir de la precondició.
- L'especificació de l'acció o funció és a la vegada les instruccions d'ús per qui l'ha de fer servir.
- La correctesa aïllada de cadascuna de les accions i/o funcions portarà a la correctesa global del disseny. L'especificació rigorosa i exacte de cada acció i funció ajuda molt.

Exercici 1 disseny descendent

Es vol dissenyar un algorisme que calculi la facturació individual d'un conjunt de telegrams enviats. Cada paraula curta (10 o menys caràcters) d'un telegrama val 0.20 euros. Les paraules llargues valen 0.40 euros. Les paraules STOP i END són gratuïtes. El contingut del telegrama acaba amb la paraula END. Donada una seqüència de caràcters que té tots els telegrams i que acaba amb un telegrama que només conté la paraula END, dissenyeu l'algorisme que escrigui la facturació de cada telegrama. Indiqueu quines abstraccions trobeu que podrien ser adients per resoldre el problema.

Solució exercici 1

- Pensar-ho en termes de la seqüència de caràcters de l'entrada pot resultar en la realització un disseny enrevesat.
- Per fer un disseny intel·ligible i fàcil de mantenir ens convé d'entrada pensar que la seqüència a l'entrada és una seqüència de telegrams.
- Un cop resolt en el primer nivell l'algorisme que tracta tots els telegrams, haurem de pensar com tractar un telegrama donat (obtenció i tractament).
- En aquest nou nivell ens convé pensar el telegrama com una seqüència de paraules. Un cop resoltes totes les accions i funcions del nivell de telegrama, passem a resoldre el nivell de paraules.

Solució exercici 1

Així doncs, tenim les següents abstraccions:

1. Tots els telegrams: seqüència de telegrams.
2. Telegrama: seqüència de paraules
3. Paraula: seqüència de caràcters

L'orientació seguida per les abstraccions és pels objectes del problema.

Exercici 2 Disseny Descendent

A l'entrada tenim una seqüència de caràcters que representa un poema. Cada vers del poema està separat pel caràcter '/'. Les paraules del vers estan separades per espais. El poema acaba amb una estrofa que no conté cap vers. Dos versos rimen quan les dues síl·labes de les respectives darreres paraules coincideixen. Es vol dissenyar un algorisme que indiqui si el poema segueix una rima amb mètrica AABCCB. Indiqueu quines abstraccions trobeu que podrien ser adients per resoldre el problema.

Solució exercici 2 Disseny Descender

Les següents abstraccions ajudaran molt el disseny a fer:

1. Poema: seqüència d'estrofes Necessitem saber si una estrofa rima correctament
2. Estrofa: seqüència de versos Cal que una estrofa consisteixi de 6 versos.
3. Vers: seqüència de paraules Ens interessa la darrera paraula del vers
4. Paraula: seqüència de sil·labes Cal analitzar les sil·labes de la darrera paraula.
5. Sil·laba: seqüència de caràcters

Solució exercici 3 Disseny Descendent

A l'entrada tenim una seqüència de caràcters que representa un poema. Cada vers del poema està separat pel caràcter '/'. Les paraules del vers estan separades per espais. El poema acaba amb una estrofa que no conté cap vers. Es vol dissenyar un algorisme que indiqui si el poema té paraules més llargues de 10 caràcters. Indiqueu quines abstraccions trobeu que podrien ser adients per resoldre el problema.

Solució exercici 3 Disseny Descendent

Les següents abstraccions ajudaran molt al disseny a fer:

1. Poema: seqüència de paraules Necessitem saber el nombre de caràcters de cada paraula.
2. Paraula: seqüència de caràcters Cal comptar els caràcters de la paraula. Noteu que l'estructura que pugui tenir el poema (estrofes, versos, etc) no és necessària aquí pel tractament a fer.

Recordeu que escollim les abstraccions que més convenen per resoldre el problema. L'orientació seguida per les abstraccions és la dels objectes del problema.