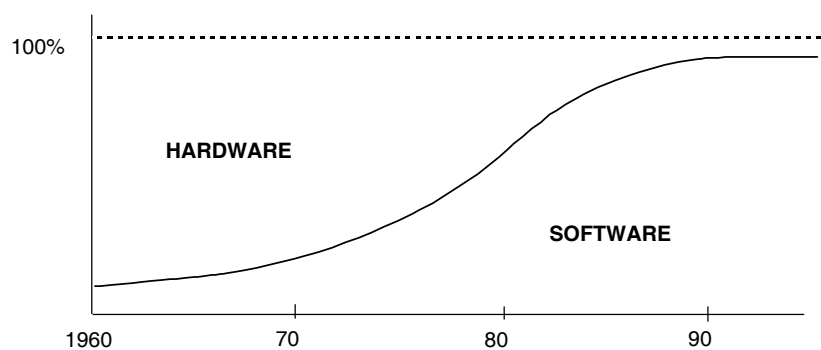


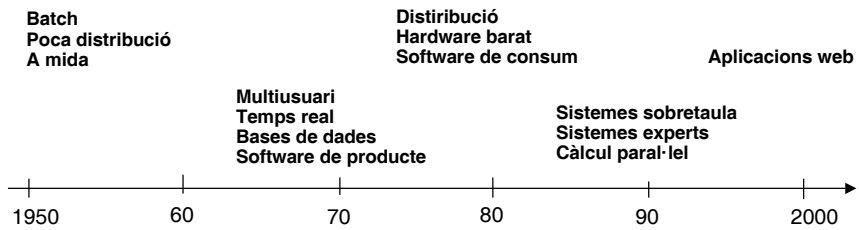
Introducció a l'Enginyeria del Software

- Software
 - Importància
 - Evolució
 - Característiques
 - Crisi del software
- Enginyeria del Software
 - Definicions
 - Característiques
 - Visió genèrica
- Paradigmes
 - Cicle de vida clàssic
 - Prototipatge
 - Model en espiral
- Bibliografia

Evolució de Costos del Hardware i del Software



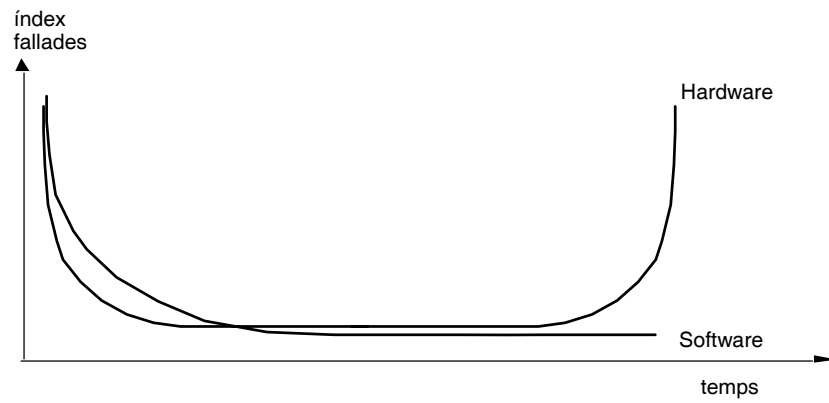
Evolució del software



Característiques del software

- Es desenvolupa, no es fabrica
- No s'espantia
- Manteniment més difícil que el hardware
- Es construeix a mida, no es reusa massa

Fallades H/S



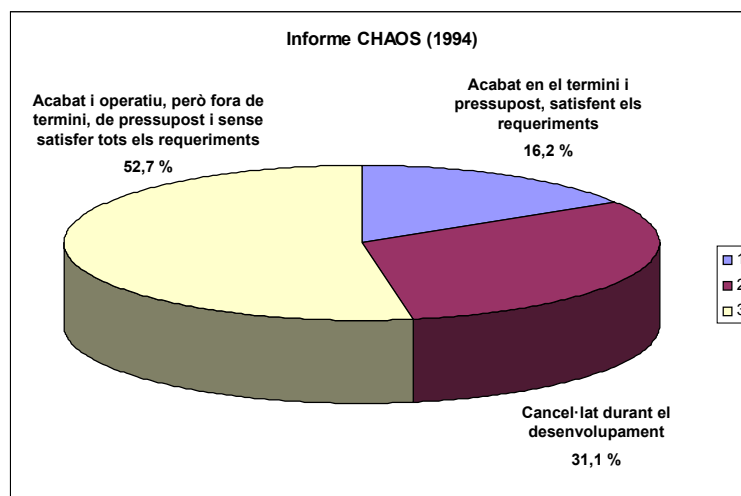
Aplicacions del Software

- Sistemes
- Temps real
- Gestió
- D'enginyeria
- Científic
- Empotrat
- De PC
- D'Intel·ligència Artificial

Crisi del software

- Crisi?
 - Aflicció crònica !!!
- Problemes:
 - Evolució contínua
 - Insatisfacció dels usuaris
 - Poca qualitat
 - Manteniment difícil
- Causes:
 - Naturalesa del software
 - Complexitat
 - Gestió
- Mites:
 - De gestió
 - Dels clients
 - Dels dissenyadors

Resultat de projectes de software



Enginyeria del software

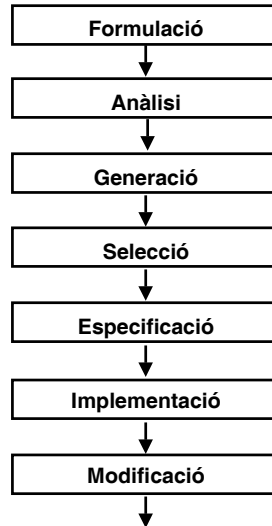
Establiment i ús de principis de l'enginyeria orientats a obtenir software:

- Econòmic
- Fiable
- Que funcioni eficientment
- Que satisfaci les necessitats dels usuaris

Un enginyer ...

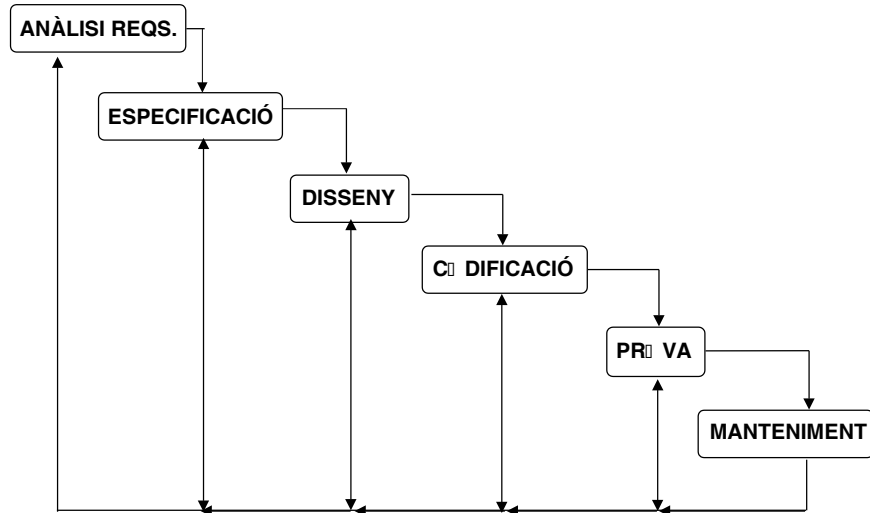
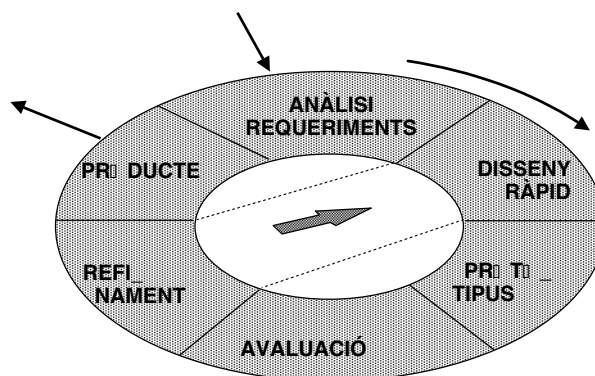
- Disposa d'un ventall de tècniques provades que donen resultats precisos.
- Es preocupa de la fiabilitat i del rendiment.
- Tracta de reduir costos i complexitat.
- Basa els seus models en teories matemàtiques sòlides.
- Construeix prototipus dels nous dissenys.
- Utilitza diagrames formals.

El procés de l'enginyeria

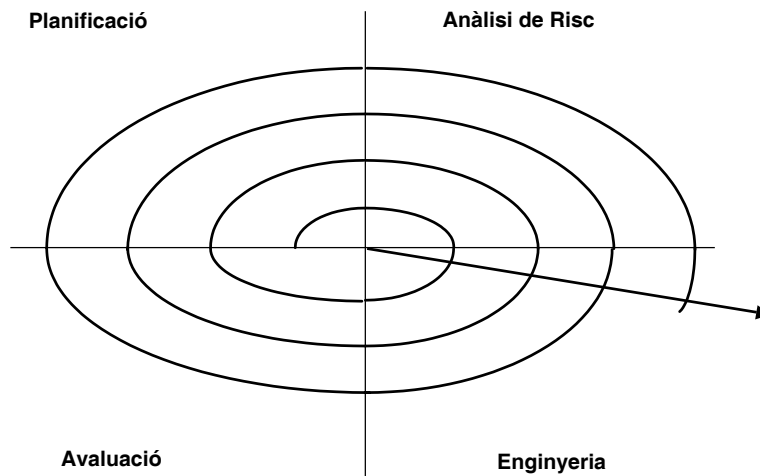


Visió genèrica de l'Enginyeria del Software

- Definició:
 - Anàlisi del sistema
 - Planificació del projecte
 - Anàlisi de requeriments del software
- Desenvolupament:
 - Disseny del software
 - Codificació
 - Prova
- Manteniment:
 - Correcció
 - Adaptació
 - Millora

Cicle de vida clàssic**Prototipatge**

Model en Espiral



Bibliografia

- R.S. Pressman
Software Engineering: A Practitioner's Approach.
5a edició.
McGraw Hill, 2001. (Cap. 1 i 2)
- The CHAOS Report, 1994
http://www.standishgroup.com/sample_research/chaos_1994_1.php

Requeriments i Especificacions

- **Determinació dels requeriments del software**
 - Requeriments del sistema versus requeriments del software
 - Etapes
 - Estratègies
- **Especificacions de sistemes software**
 - Requeriments funcionals i no funcionals
 - Propietats desitjables de les especificacions
 - Estàndards de documentació
- **Bibliografia**

Requeriments del sistema vs. requeriments del software

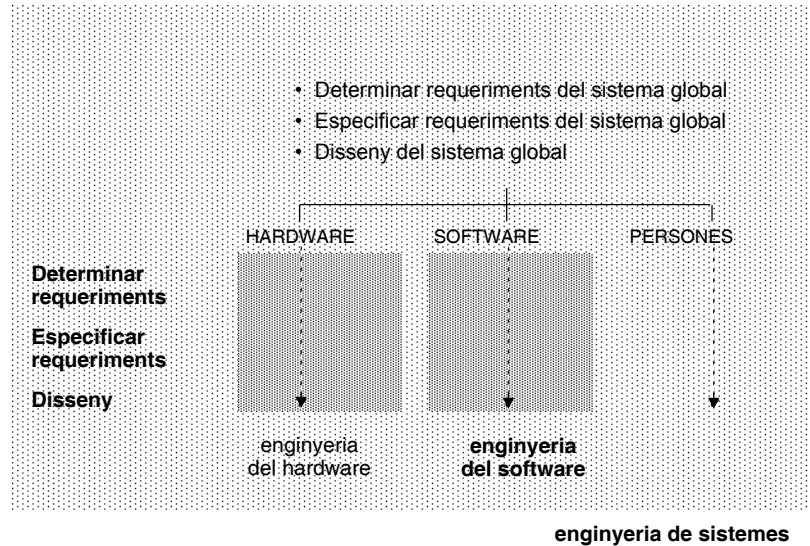
Requeriment: Condició o capacitat necessitada per un usuari per tal de solucionar un problema o aconseguir un objectiu

- La solució al problema es pot realitzar amb software, hardware, manualment, o amb una combinació de tots tres.
- Si la solució és composta, abans de dissenyar els detalls d'un component software concret, cal dissenyar el sistema global.

Exemple de sistema compost: refinaria automatitzada

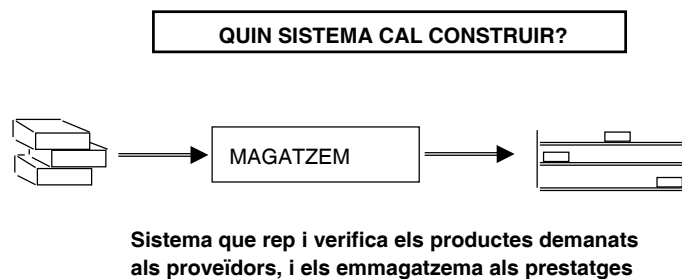
Exemple de sistema només software: control d'estocs

Etapes de l'enginyeria de sistemes



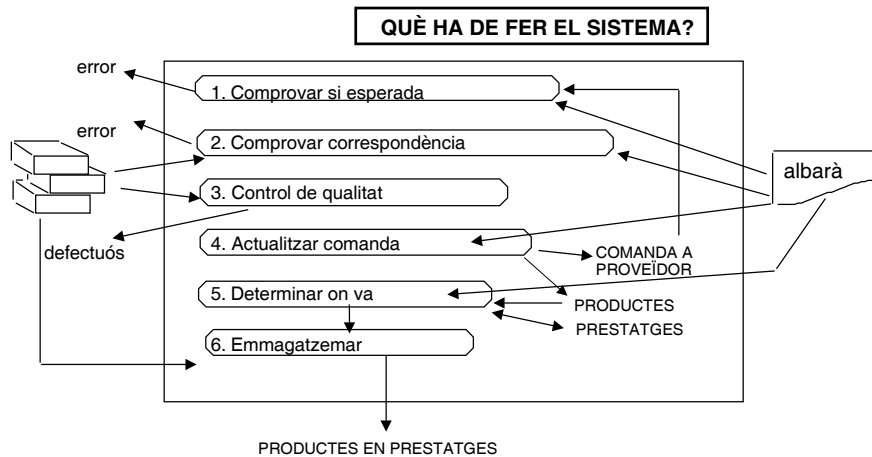
Determinar requeriments del sistema global

- Comprensió dels objectius i necessitats de l'usuari
- Definir el conjunt de sistemes que podrien satisfer les necessitats o objectius i avaluar-los
- Triar el sistema més adient



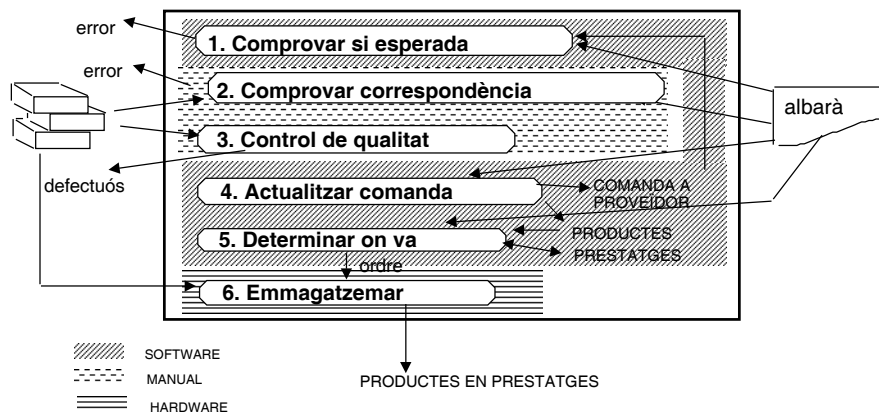
Especificar els requeriments del sistema global

- Descriure el comportament extern del sistema, des del punt de vista de l'usuari, o de l'entorn

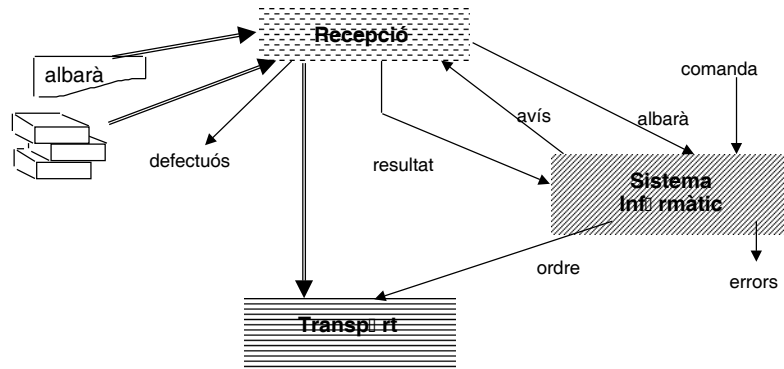


Disseny del sistema global

- Determinar l'arquitectura general del sistema que millor satisfà els requeriments , en termes de:
 - components físics: hardware, software, persones
 - comunicació entre ells

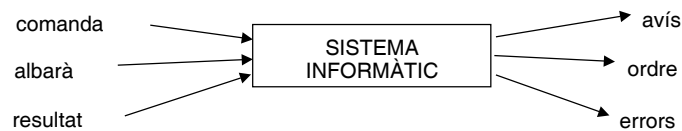


Disseny del sistema global



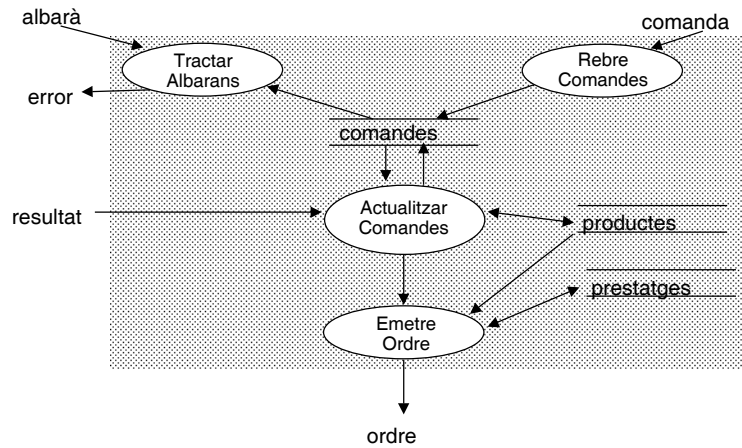
Determinar requeriments del software

- És el subconjunt dels requeriments del sistema global que han estat assignats al component software concret



Especificar els requeriments del software

- Descriure amb detall el comportament extern del component software concret



Estratègies de determinació dels requeriments

- Demanar-ho a l'usuari
- Treure-ho d'un sistema software existent
- Sintetitzar-ho a partir del sistema global
- Descobrir-ho mitjançant experimentació

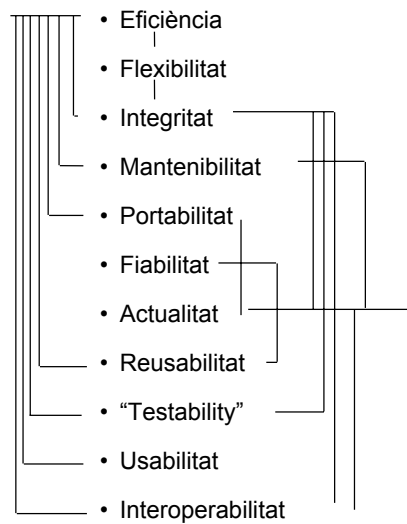
Requeriments del software

- Funcionals
 - Descriuen totes les entrades i sortides, i la relació entre ambdues
 - de dades
 - de procés
- No funcionals
 - Defineixen les qualitats generals que ha de tenir el sistema en realitzar la seva funció
 - Eficiència
 - Tipus d'interfícies
 - Econòmics
 - Estructurals
 - Polítics
 - Qualitat

Factors de qualitat del software

- Eficiència
- Flexibilitat
- Integritat
- Mantenibilitat
- Portabilitat
- Fiabilitat
- Actualitat
- Reusabilitat
- "Testability"
- Usabilitat
- Interoperabilitat

Factors de qualitat del software: conflictes



Propietats desitjables de les especificacions

- No ambigües
- Completes
- Verificables
- Consistents
- Modificables
- "Traçables"
- Usables durant l'operació i el manteniment

Com organitzar una especificació de requeriments? estàndard de documentació ANSI/IEEE (I)

1. Introducció
 - 1.1 Propòsit de l'especificació
 - 1.2 Abast del producte
 - 1.3 Definicions i abreviatures
 - 1.4 Referències
 - 1.5 Visió general de l'especificació
2. Descripció General
 - 2.1 Perspectiva del producte
 - 2.2 Funcions del producte
 - 2.3 Característiques dels usuaris
 - 2.4 Restriccions generals
 - 2.5 Supòsits i dependències
3. Requeriments específics
4. Apèndixos
5. Índex

Com organitzar una especificació de requeriments? estàndard de documentació ANSI/IEEE (II)

3. Requeriments específics
 - 3.1 Requeriments d'interfícies externes
 - 3.2 Requeriments funcionals
 - 3.3 Requeriments de rendiment
 - 3.4 Requeriments lògics de la base de dades
 - 3.5 Restriccions de disseny
 - 3.6 Atributs del sistema software
 - a) Fiabilitat
 - b) Disponibilitat
 - c) Seguretat
 - d) Mantenibilitat
 - e) Portabilitat
 - 3.7 Organització dels requeriments específics
 - 3.8 Altres requeriments

Bibliografia

- K.Shumate, M.Keller
Software Specification and Design - A Disciplined Approach for Real-Time Systems.
John Wiley, 1992. (Cap. 3)
- A. M. Davis
Software Requirements - Objects, Functions and States.
Prentice-Hall, 1993. (Cap. 1-5)
- *IEEE Recommended Practice for Software Requirements Specifications*
IEEE Std 830-1998 , 20 Oct 1998.

Introducció a l'orientació a objectes

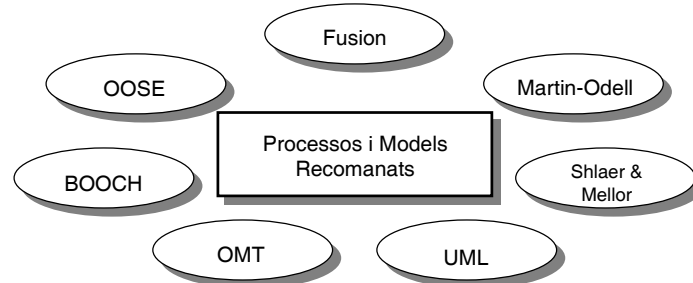
- Motivació i orígens
- Visió d'un sistema software
 - Aspecte estàtic
 - Aspecte dinàmic
- La notació UML

Motivació

- Aparició dels llenguatges de programació orientats a objectes.
 - SIMULA: finals dels 1960's
 - SMALLTALK: principis dels 1970's
- L'ús d'aquests llenguatges requereix un nou enfocament d'anàlisi i de disseny.
- Altres factors:
 - Desenvolupament de noves aplicacions
 - Èmfasi principal en l'estructura de dades

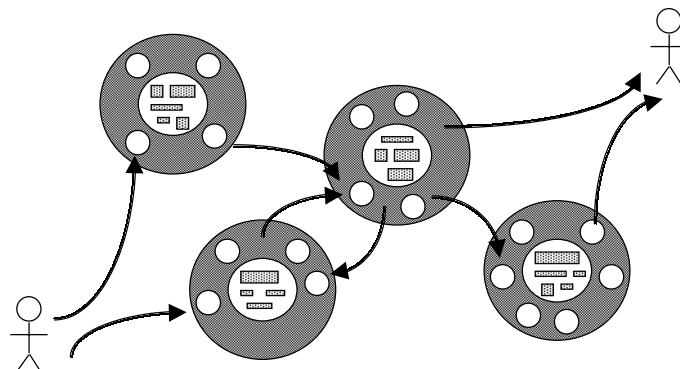
**Aparició dels primers mètodes de disseny i d'anàlisi
orientats a objectes**

Orígens



Rumbaugh, Blaha, Premerlani (OMT) - 1991
Coad, Yourdon - 1991
Shlaer, Mellor - 1992
Booch - 1992
Odell, Martin - 1992
Jacobson (OOSE) - 1993
Fusion - 1994
Booch, Rumbaugh, Jacobson (UML) - 1997

Visió d'un sistema software

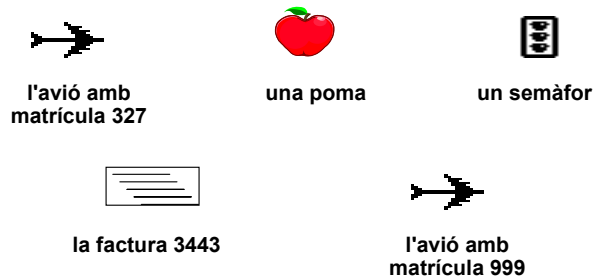


Objectes

Objecte:

Entitat que existeix al món real

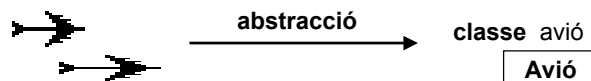
Tenen identitat i són distingibles entre ells



Classes d'objectes

Classe d'objectes: descriu un conjunt d'objectes amb:

- les mateixes propietats
- comportament comú
- idèntica relació amb altres objectes
- semàntica comuna



Abstracció: eliminar distincions entre objectes per a poder observar aspectes comuns

Els objectes d'una classe tenen les mateixes propietats i els mateixos patrons de comportament

Atributs

Atribut: és una propietat compartida pels objectes d'una classe.

Exemples:

Persona => nom, adreça, telèfon, ...

Avió => model, capacitat, color, ...

Persona
Nom
Adreça
Telèfon
Data_naixement
Edat

Avió
Model
Capacitat
Color

- Cada atribut té un valor (probablement diferent) per cada objecte
- Els atributs poden ser bàsics o derivats

Operacions (I)

Operació: és una funció o transformació que es pot aplicar als objectes d'una classe.

Persona
Nom
Adreça
Telèfon
Canvi_adreça
Canvi_feina

Avió
Model
Capacitat
Colors
Reparar
Moure

- Les operacions d'un objecte són invocades pels altres objectes

Mètode: especificació procedural (implementació) d'una operació dins d'una classe.

Encapsulament: consisteix a separar els aspectes externs d'un objecte dels detalls d'implementació.

Operacions (II)

- En les operacions, cal indicar també el tipus dels arguments i del resultat.

Triangle
Color
Posició
Girar (angle: Real)
seleccionar (p:Punt):Booleà

Quadrat
Color
Posició
Girar (angle: Real)

Polimorfisme:

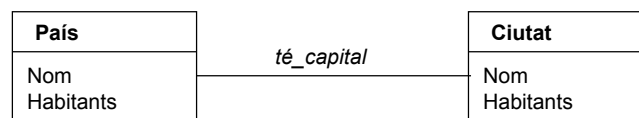
- una mateixa operació es pot aplicar a diferents classes
- la seva implementació depèn de cada classe

Associacions

Associació:

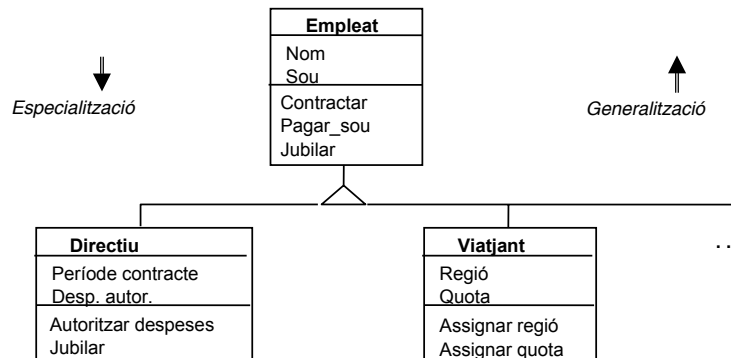
defineix la manera d'enllaçar o connectar objectes de diferents classes

Exemple: Un país té una única capital.



Generalització / Especialització

Generalització: és l'acte o resultat de distingir un concepte que és més general que un altre



Herència: permet que les propietats i les operacions d'una classe siguin accessibles a les seves subclasses

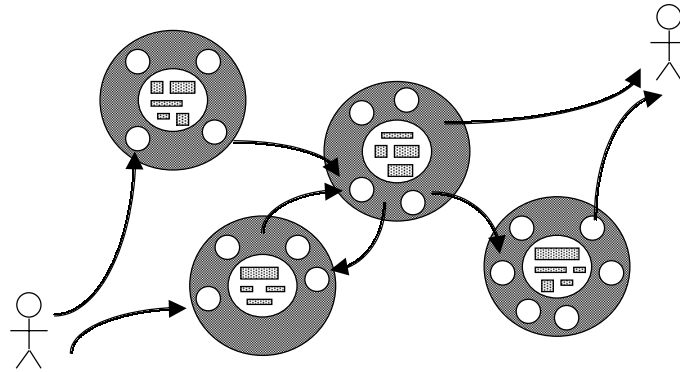
Orientació a objectes (I)

Aspecte estàtic: descriu l'estructura estàtica dels objectes del sistema i les seves interrelacions

	Intra - objecte	Inter - objectes
Aspecte estàtic	Classes d'objectes Atributs Operacions	Associació Generalització ...

Descripció del comportament (I)

Els objectes es comuniquen mitjançant la invocació d'operacions d'altres objectes



Descripció del comportament (II)

Aspecte dinàmic (de comportament): descriu els aspectes d'un sistema que canvien amb el temps

L'**aspecte dinàmic** d'un sistema descriu:

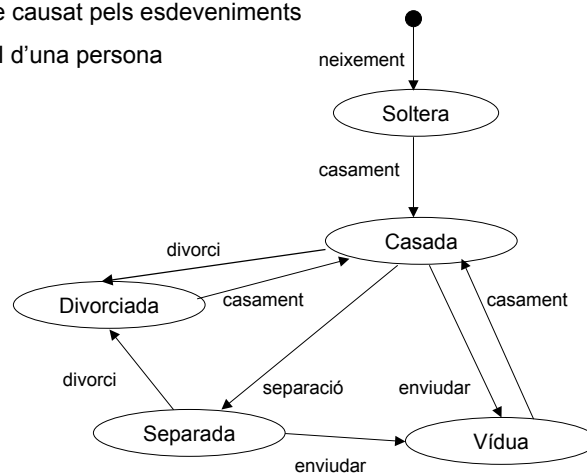
- Interaccions entre els objectes
- Possibles estats d'un objecte
- Transicions entre estats
- Quins esdeveniments es produeixen
- Quines operacions s'executen

Hi ha molta divergència entre els mètodes actuals a l'hora de tractar l'aspecte dinàmic

Descripció del Comportament (III)

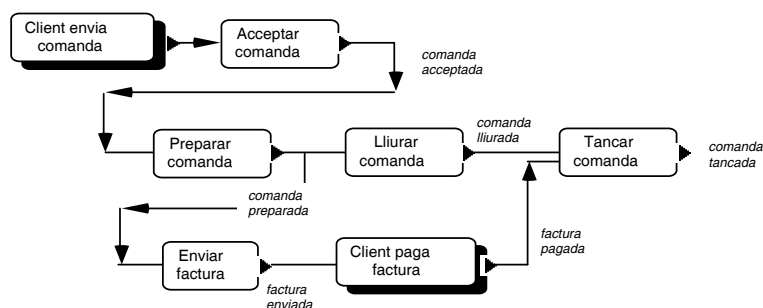
Diagrama de transició d'estats: especifica el canvi d'estat d'un objecte causat pels esdeveniments

Exemple: estat civil d'una persona



Descripció del comportament (IV)

Esquema d'esdeveniments: permet especificar la interacció entre diferents objectes (usat pel mètode de Martin i Odell [MO92])



Orientació a objectes (II)

Aspecte estàtic: descriu l'estructura estàtica dels objectes del sistema software i les seves interrelacions

Aspecte dinàmic (de comportament): descriu els aspectes d'un sistema software que canvien amb el temps

	Intra - objecte	Inter - objectes
Aspecte estàtic	Classes d'objectes Atributs Operacions	Associació Generalització ...
Aspecte dinàmic	Diagrama de transició d'estats	Esquema d'esdeveniments

Anàlisi i disseny orientats a objectes

Anàlisi:

- Creació d'una especificació del problema i dels requeriments
- **Què** ha de fer el sistema software

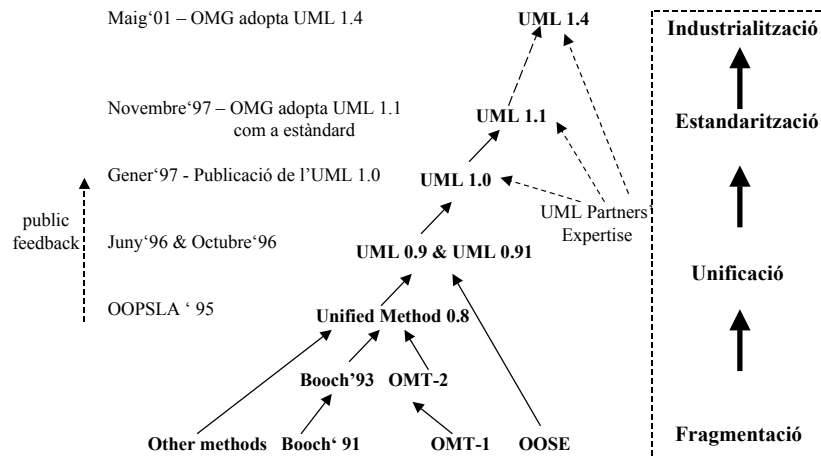
Disseny:

- definició d'una solució software que satisfaci els requeriments
- **Com** ho farà el sistema

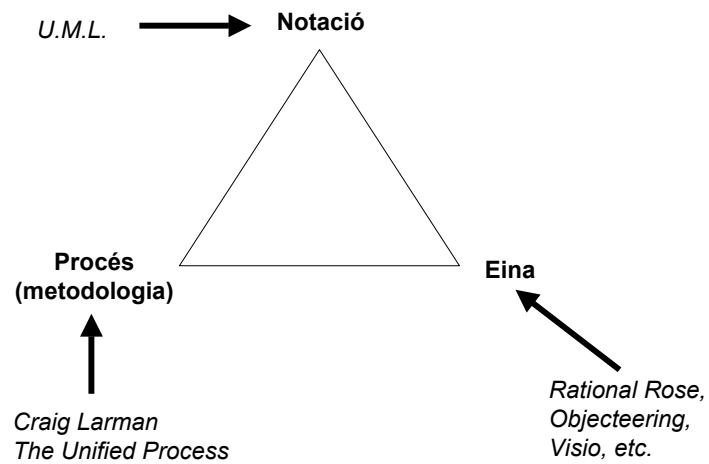
... orientats a objectes

- S'usen els mateixos conceptes a l'anàlisi i al disseny
- És difícil determinar on acaba l'anàlisi orientada a objectes i on comença el disseny:
 - estratègia de desenvolupament iterativa
 - diferències de criteris segons els autors

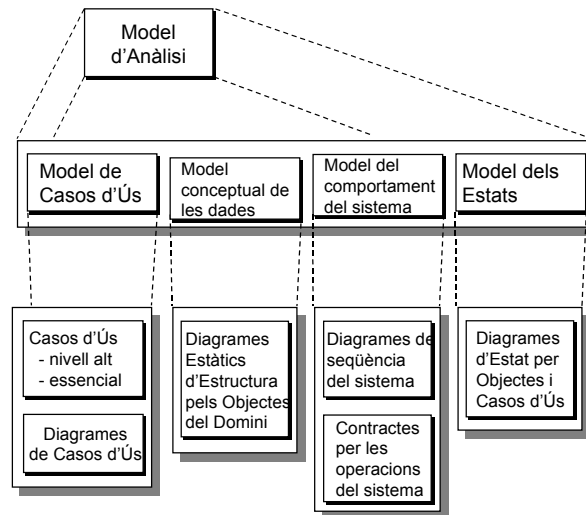
El llenguatge UML (Unified Modeling Language)



El triangle de l'èxit



Model d'Anàlisi (especificació)



Bibliografia

- *Software Engineering: A Practitioner's Approach (5th edition)*
R. S. Pressman
Mc-Graw-Hill, 2001 (Cap. 20 i 21)
- *Object-Oriented Analysis and Design*
G.Booch
Benjamin/Cummings, 1994
- *Object Oriented Software Engineering: A Use Case Driven Approach*
I. Jacobson et al.
Addison-Wesley, 1992
- *Object-Oriented Modelling and Design*
J. Rumbaugh et al.
Prentice-Hall, 1991
- *Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and the Unified Process*
C. Larman
Prentice-Hall 2002
- *The Unified Software Development Process*
I.Jacobson, G.Booch, J.Rumbaugh
Addison-Wesley, 1999.