

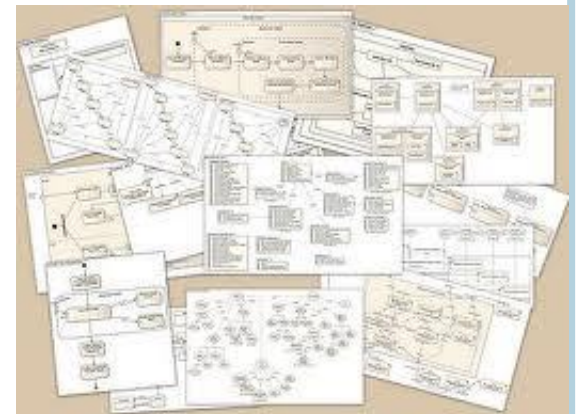


UML 2.0: INTRODUZIONE

Ingegneria del Software a.a. 2023-2024

AGENDA

- Che cosa è UML?
 - Cosa non è UML ...
- Come usare UML?
- Prospettiva Concettuale vs. Software
- Un po di storia ...
- Perché è importante conoscere UML?
- Notazioni e meta-modello
- I 14 diagrammi (UML 2.5)
 - Ne vedremo solo alcuni
- Come integrare UML nel processo?





“questi è per me fratello, sorella e madre ...”

Mt 12,46–50

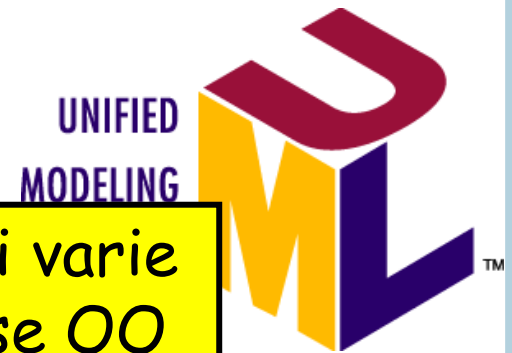
CHE COS'È UML?

- UML = Unified Modeling Language
- Famiglia di **notazioni grafiche** utili a supportare la descrizione e il progetto dei sistemi software
 - In particolare sistemi OO
 - Ma anche altro:
 - Sistemi hardware
 - Dominio del business
 - ...
- Standard OMG
 - Object Management Group
 - Consorzio di aziende IT
 - Not-for-profit
 - Obiettivo: creare standard nel contesto IT



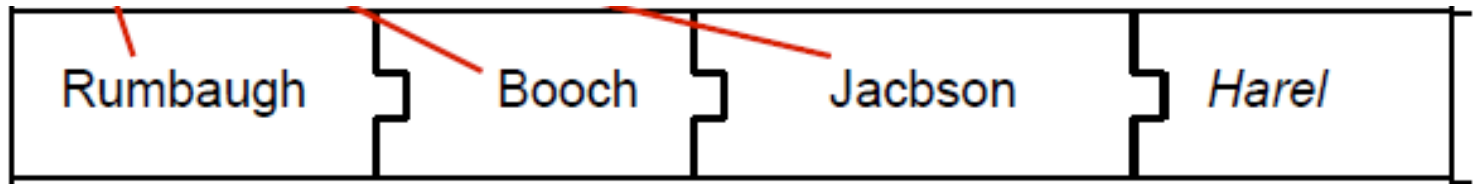
CHE COS'È UML?

- UML = Unified Modeling Language
- Famiglia di **notazioni grafiche** utili a supportare la descrizione e il



Unificato = in pratica una collezione di varie notazioni preesistenti, integrate e rese OO

- Ma anche altro:



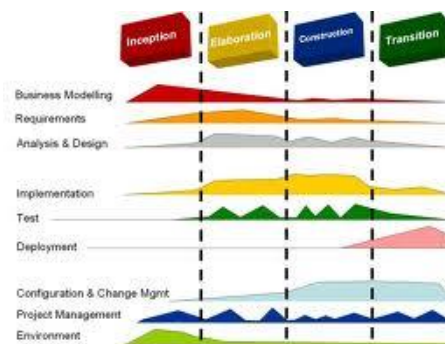
○ Standard OMG

- Object Management Group
 - Consorzio di aziende IT
 - Not-for-profit
 - Obiettivo: creare standard nel contesto IT



COSA NON È UML

- **Non è un metodo di sviluppo (metodologia)!**
 - È un linguaggio/notazione
- **Non è un linguaggio di programmazione**
 - È un linguaggio di modellazione
- **Non è legato a UP**
 - Non è legato a nessun processo particolare
 - Può essere usato sia nei metodi plan-driven che in quelli agili



No!

I MOLTI MODI DI USARE UML

○ **Abbozzo (sketch)**

- **Scopo:** aiutare la comunicazione e la discussione
 - Solo info + importanti (diagrammi incompleti)
- **Tool:** editor grafici o lavagna

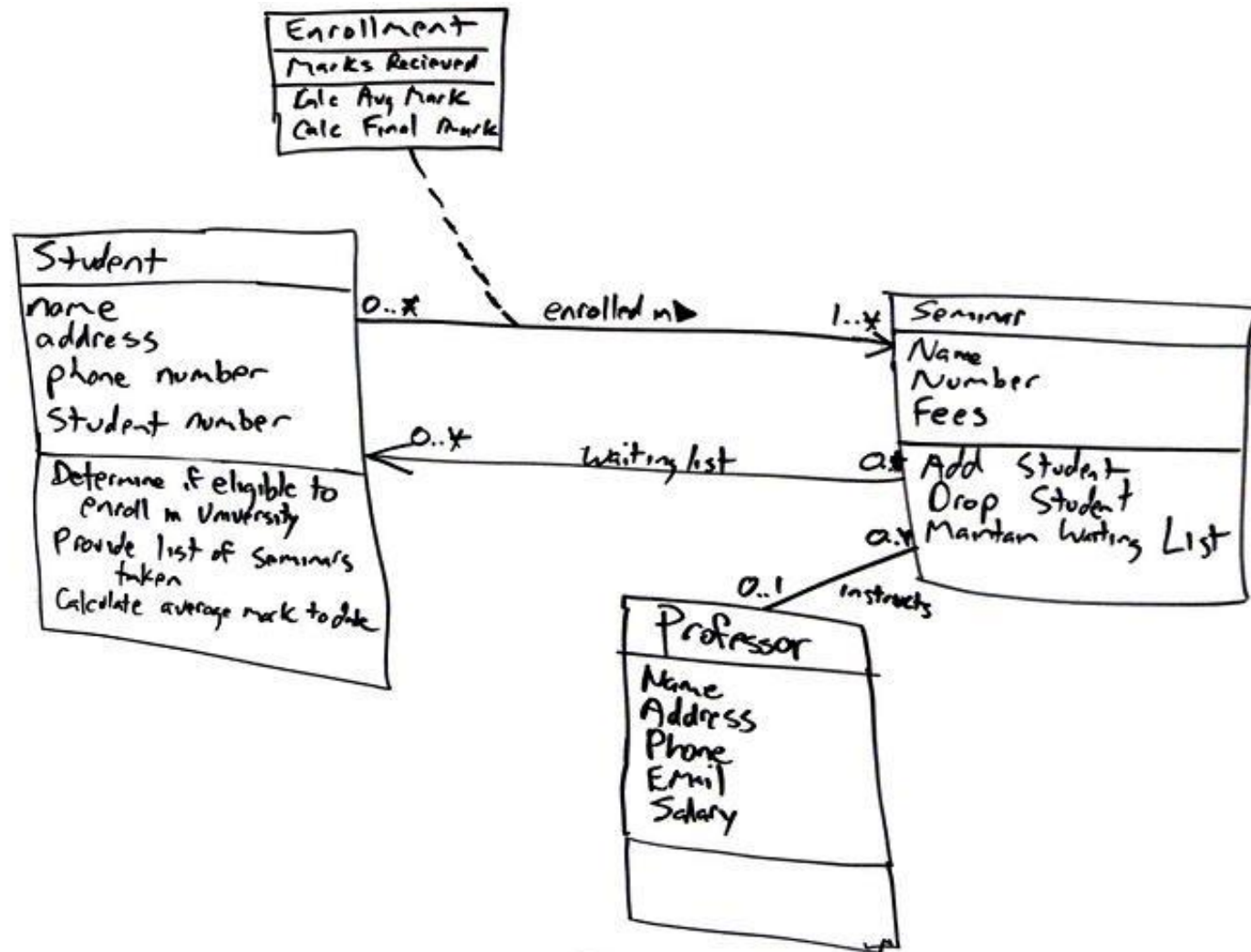
○ **Progetto dettagliato (blueprint)**

- **Scopo:** fornire un modello completo/dettagliato da implementare
 - Codifica = attività meccanica
- **Tool:** UML modeler
 - Es. Visual Paradigm, Papyrus

○ **Linguaggio di programmazione**

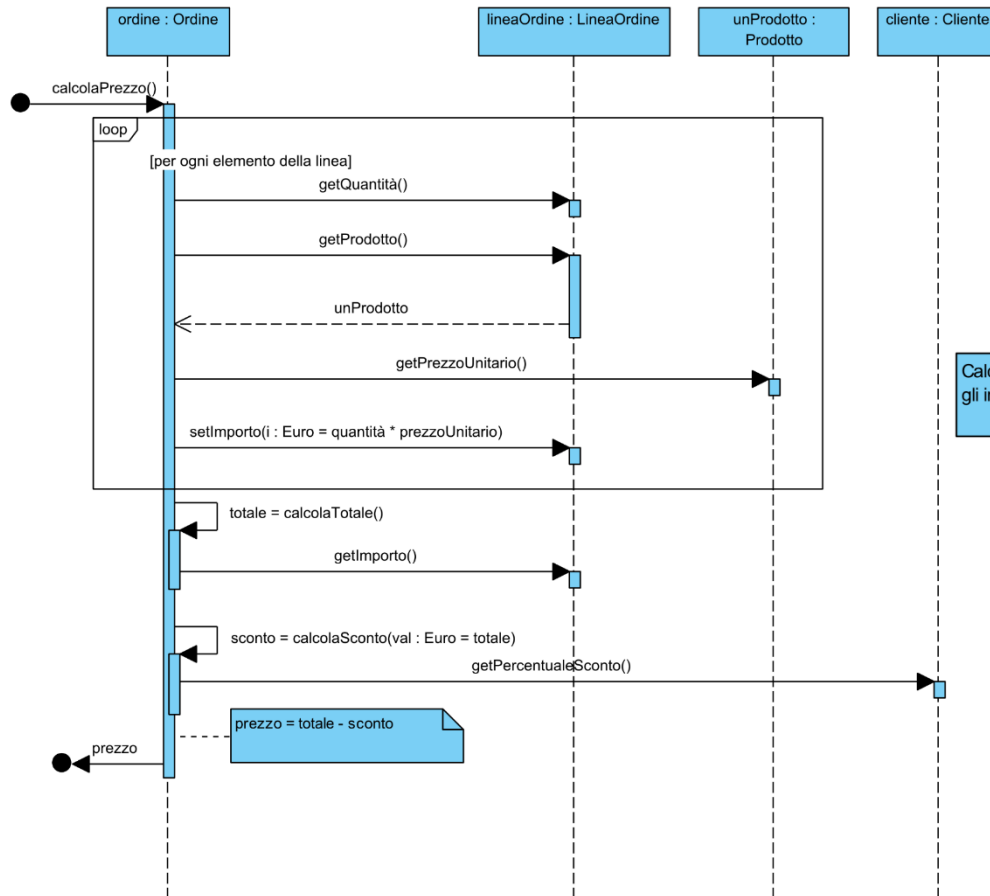
- **Scopo:** fornire un “modello eseguibile”
- **Tool:** specifici per eseguire/trasformare il modello
 - Es. UniMod / WebRatio (IFML), trasformazioni ad-hoc

ESEMPIO: ABBOZZO

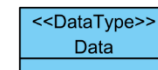
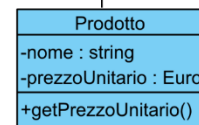
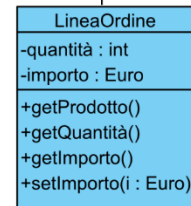
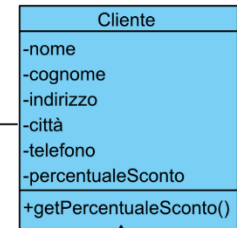
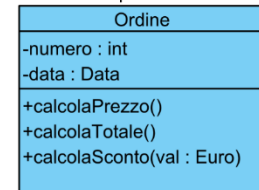


ESEMPIO: DESIGN DETTAGLIATO

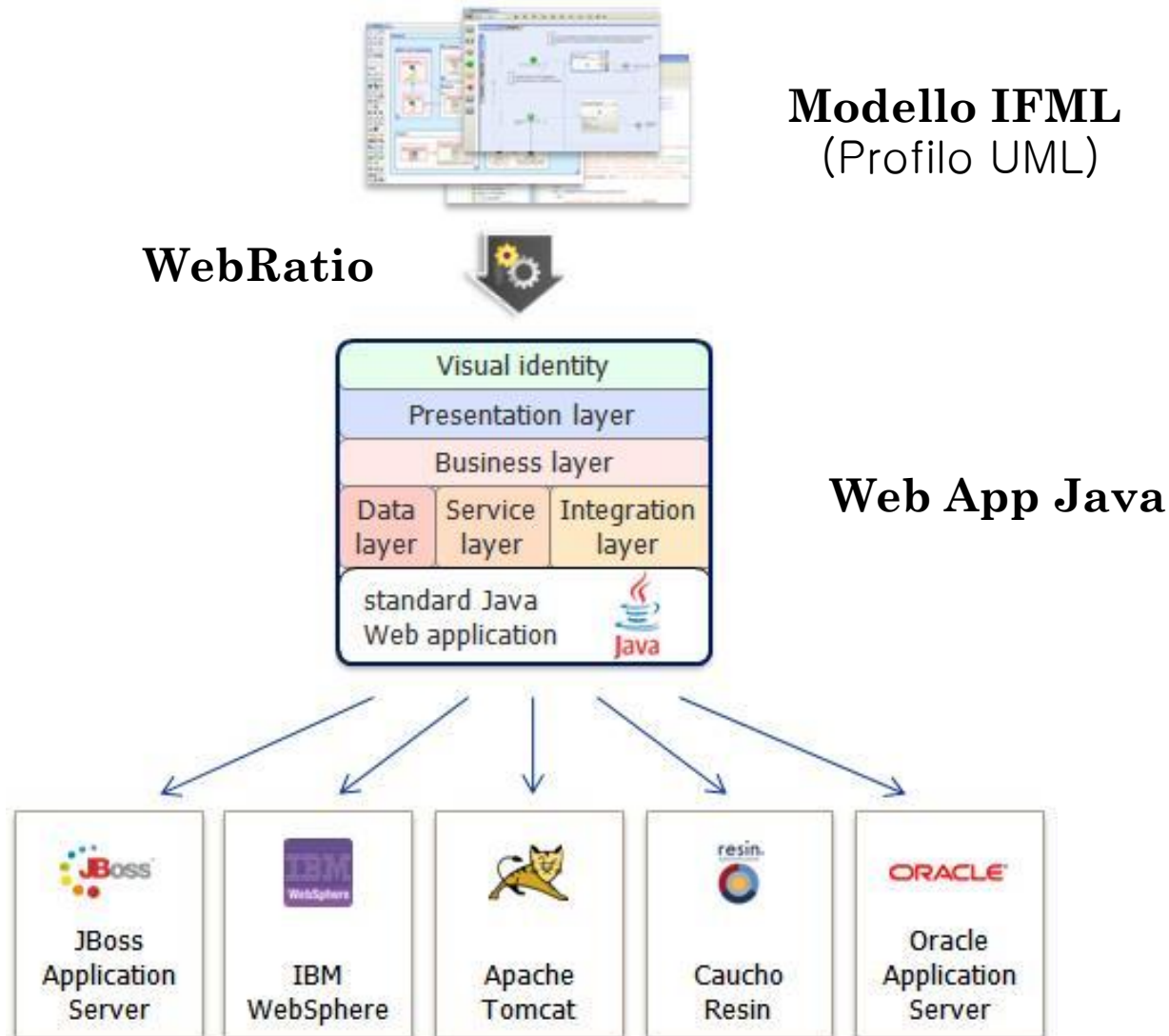
Visual Paradigm



CalcolaTotale() ritorna la somma di tutti gli importi di linea d'ordine



ESEMPIO: LINGUAGGIO DI PROGRAMMAZIONE



○ Prospettiva software

- Elementi rappresentati sono *elementi di un sistema software*
 - Es. Utente: è la classe in un linguaggio di programmazione, ad esempio con gli attributi nome, cognome e indirizzo
- Serve per **descrivere il design di un sistema software**
- Serve per documentare un sistema software

○ Prospettiva concettuale

- Elementi rappresentati sono *concetti del dominio*
 - Es. Utente: è la persona fisica
- Serve per modellare il **dominio del business e i processi**

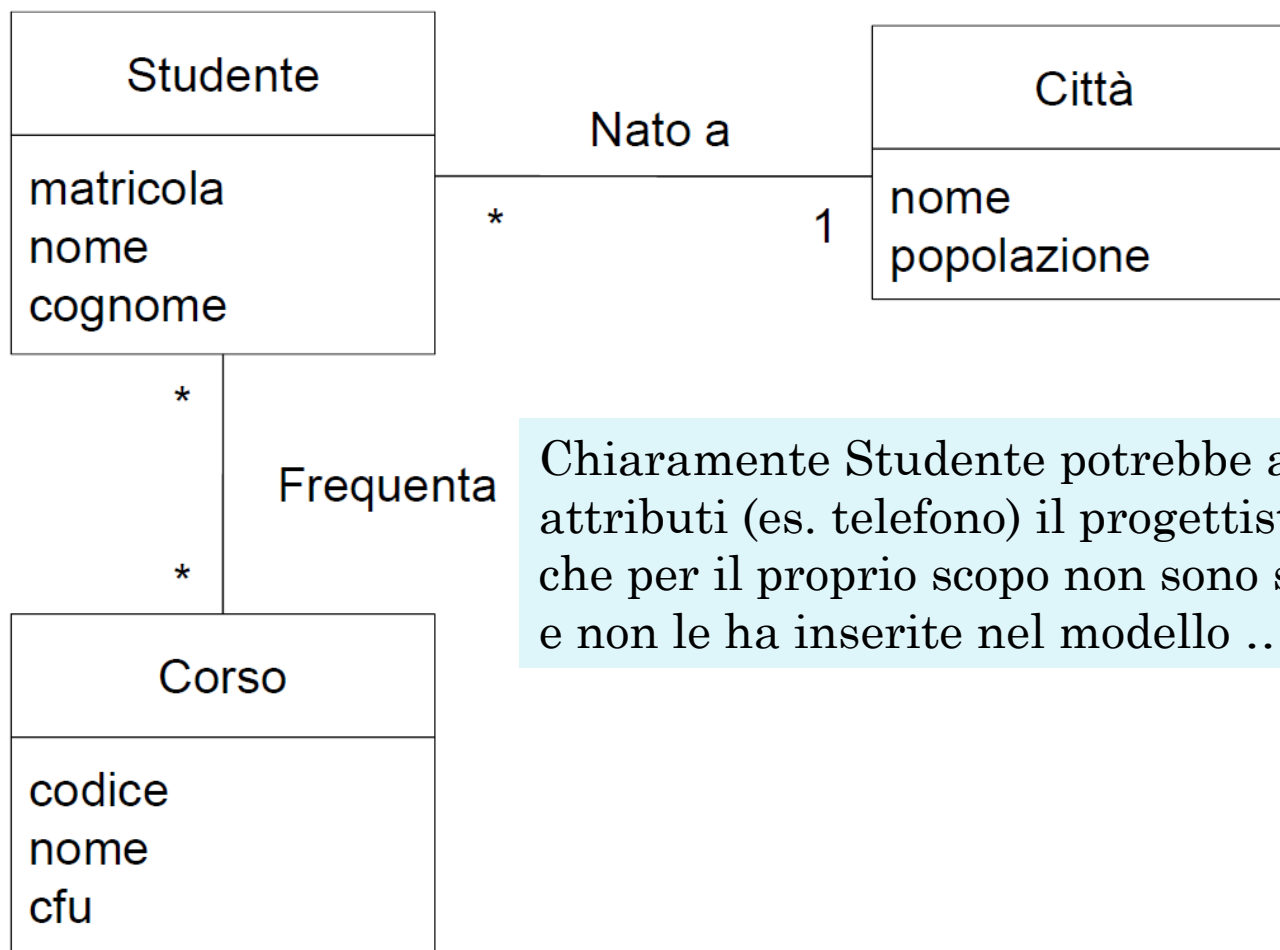
Spesso nello sviluppo di un software il primo passo è modellare il dominio del business ...

MODELLO DI DOMINIO (DEL BUSINESS)

- E' una rappresentazione *visuale* di classi concettuali o di **oggetti del mondo reale** di un dominio [Fowler]
- E' un **glossario visuale** delle **astrazioni significative** del dominio
 - Non si modella tutto! Solo i concetti più importanti e rilevanti nel contesto
- Utile:
 - Per chiarire i concetti di dominio nel quale il sistema software che dobbiamo progettare andrà inserito
 - Glossario comune tra i vari stakeholder
 - Per ispirare nomi di alcune classi e attributi software nel design
 - Per la progettazione del sistema
 - Punto di partenza

ESEMPIO DI MODELLO DI DOMINIO

Rettangoli rappresentano classi concettuali

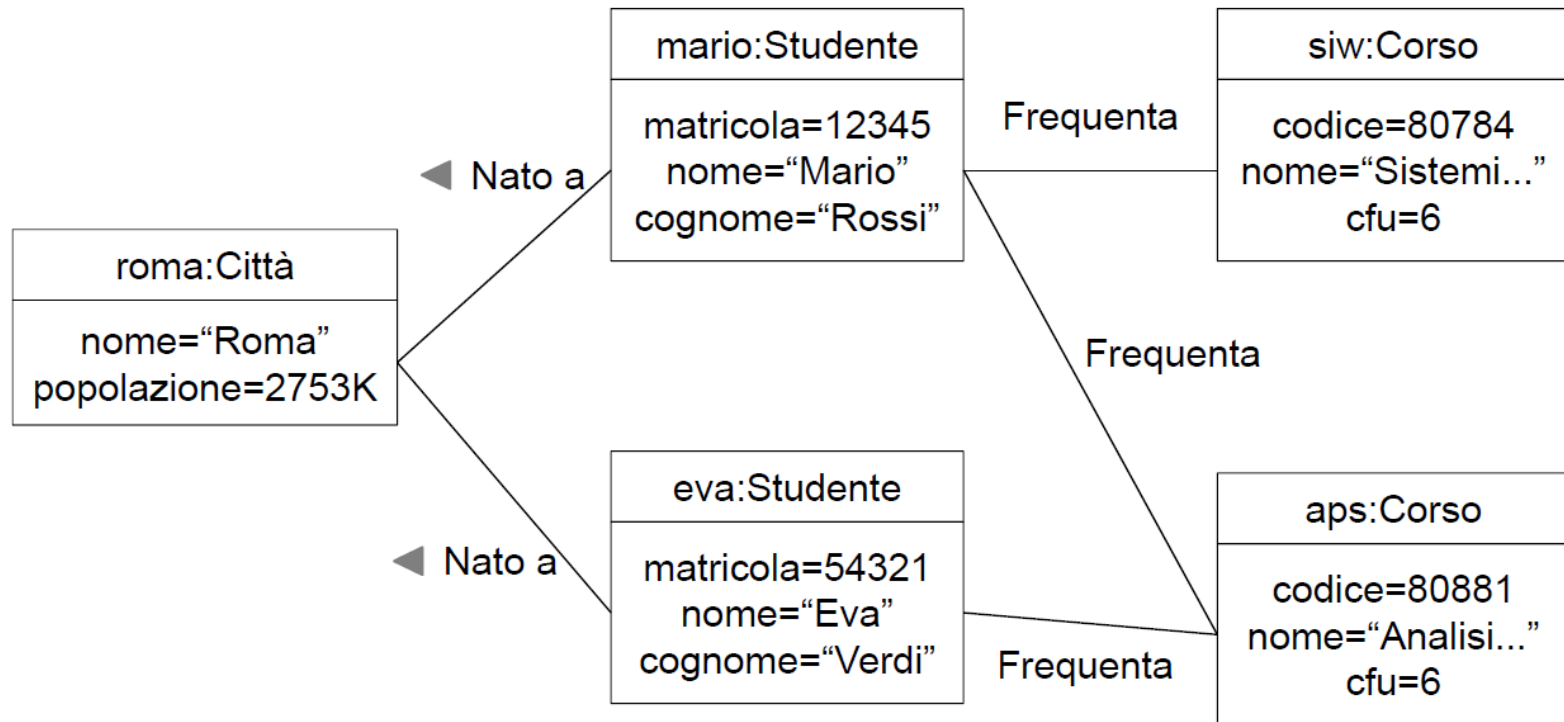


Chiaramente Studente potrebbe avere più attributi (es. telefono) il progettista ha deciso che per il proprio scopo non sono significative e non le ha inserite nel modello ...

OGGETTI DI DOMINIO

Oggetti di dominio per rappresentare

- uno studente Mario Rossi, nato a Roma, frequenta SIW e APS
- una studentessa Eva Verdi, nata a Roma, frequenta APS



UN PO' DI STORIA ...



Rumbaugh : Object Modelling Technique (OMT)

Booch : metodologia nota come "Booch 93"

Jacobson : Object-oriented software engineering (OOSE)

Incorporate richieste di altri vendor

UN F

...



Anni '80.
linguaggi
ad oggetti
iniziano a
prendere
piede
(Smalltalk)

1980-1995.
"babele" di
linguaggi di
modellazione

1994.
Rumbaugh si
unisce a
Booch alla
Rational
Software
(azienda)

1995.
Booch e
Rumbaugh
presentano a
OOPSLA Unified
Method v. 0.8.

Viene dato
l'annuncio che
Rational ha
comprato
Objectory dove
lavora Jacobson

1997.
proposal
parte di
OMG.
"amigos"
pongono
UML 1.0

Nov 1997.
nasce UML
1.1
(standard
OMG)

Mar 2005.
UML 2.0
diventa la
versione
ufficiale

Adesso.
UML 2.5.1

Rumbaugh : Object Modelling Technique (OMT)

Booch : metodologia nota come "Booch 93"

Jacobson : Object-oriented software engineering (OOSE)

PERCHÈ STUDIARE UML?

- È uno standard
 - UML svolge un'importantissima funzione di “**lingua franca**” nel mondo ICT
- Abbastanza usato/richiesto dall'industria
 - Almeno a livello di abbozzo
 - E' richiesto in alcuni appalti pubblici
- Un immagine (diagramma) vale 1000 parole!
- Indispensabile per le figure di **Analista** e **Progettista** (architetto) software

Ovvero (si spera) le vostre future occupazioni ...

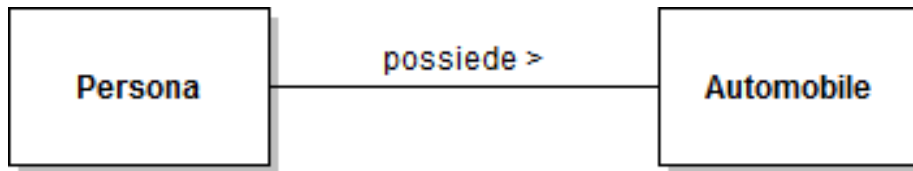
PERCHÈ NON STUDIARLO IN MODO APPROFONDITO?

- Complesso per un corso di base come questo
- Approccio **blueprint** poco usato nell'industria
 - Complesso anche per i progettisti + bravi proporre un design completo che sia privo di errori
 - Difficile perché il modello non si può testare/eseguire
- Non esistono **studi empirici** che provano che l'approccio blueprint è il migliore
- Mio punto di vista uguale a quello di M. Fowler

“I progetti dettagliati sono troppo complessi da realizzare e rallentano il processo di sviluppo”

NOTAZIONE E META-MODELLO

- UML definisce una (famiglia di) notazione(i) e un meta-modello
- **Notazione = sintassi grafica del linguaggio**
 - Es. la notazione dei diagrammi delle classi definisce come rappresentare classi e associazioni

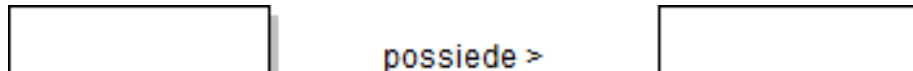


- **Meta-modello = definisce i concetti stessi del linguaggio**
 - Es. Cosa è una Classe, un Attributo, un Operazioni
- Quale è l'impatto del meta-modello sull'uso della notazione?

Poco o nulla se usiamo UML come abbozzo

NOTAZIONE E META-MODELLO

- UML definisce una (famiglia di) notazione(i) e un meta-modello
- **Notazione = sintassi grafica del linguaggio**
 - Es. la notazione dei diagrammi delle classi definisce come rappresentare classi e associazioni

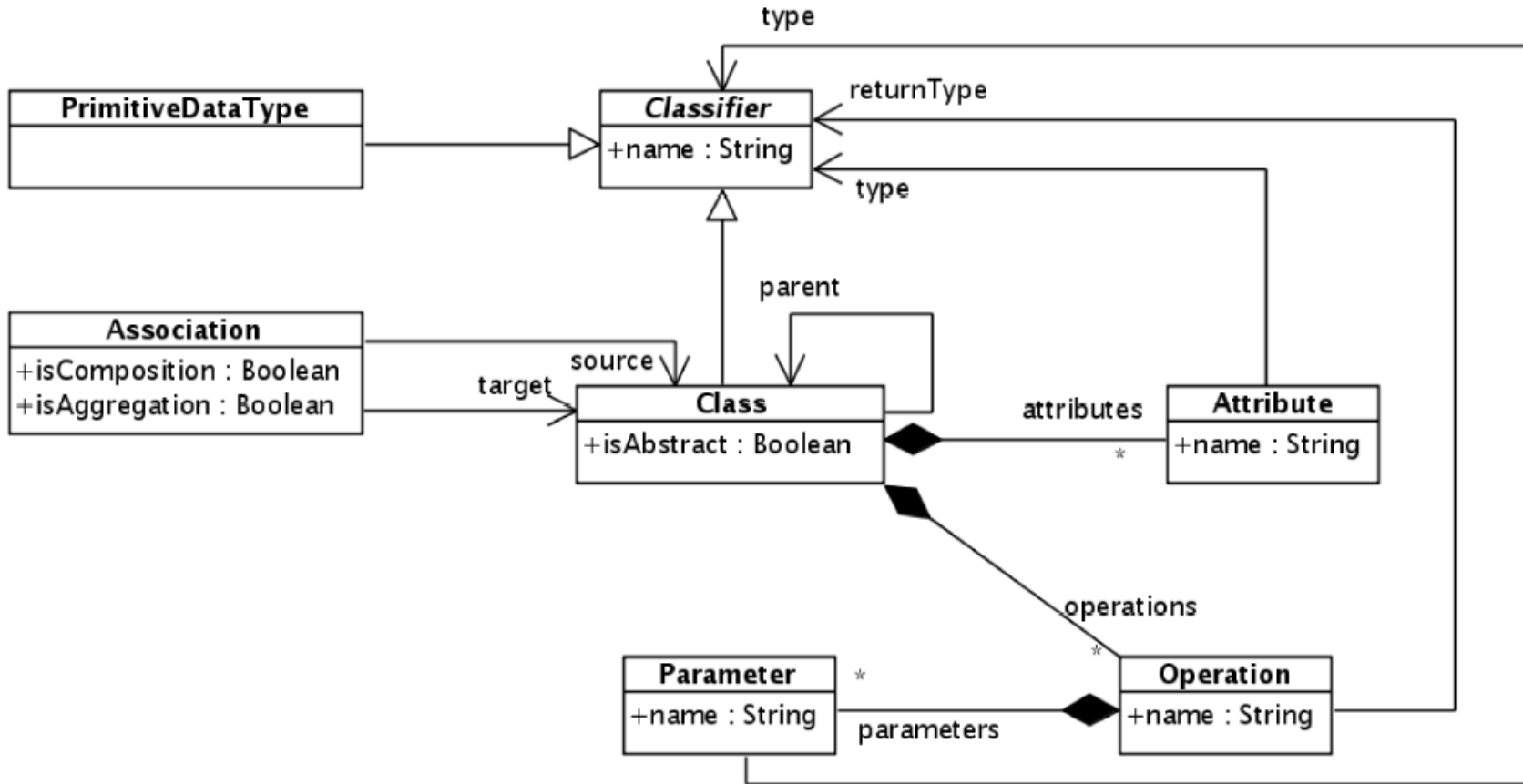


Possiamo vivere anche senza meta-modello!!!

- **Meta-modello = definisce i concetti stessi del linguaggio**
 - Es. Cosa è una Classe, un Attributo, un Operazioni
- Quale è l'impatto del meta-modello sull'uso della notazione?

Poco o nulla se usiamo UML come abbozzo

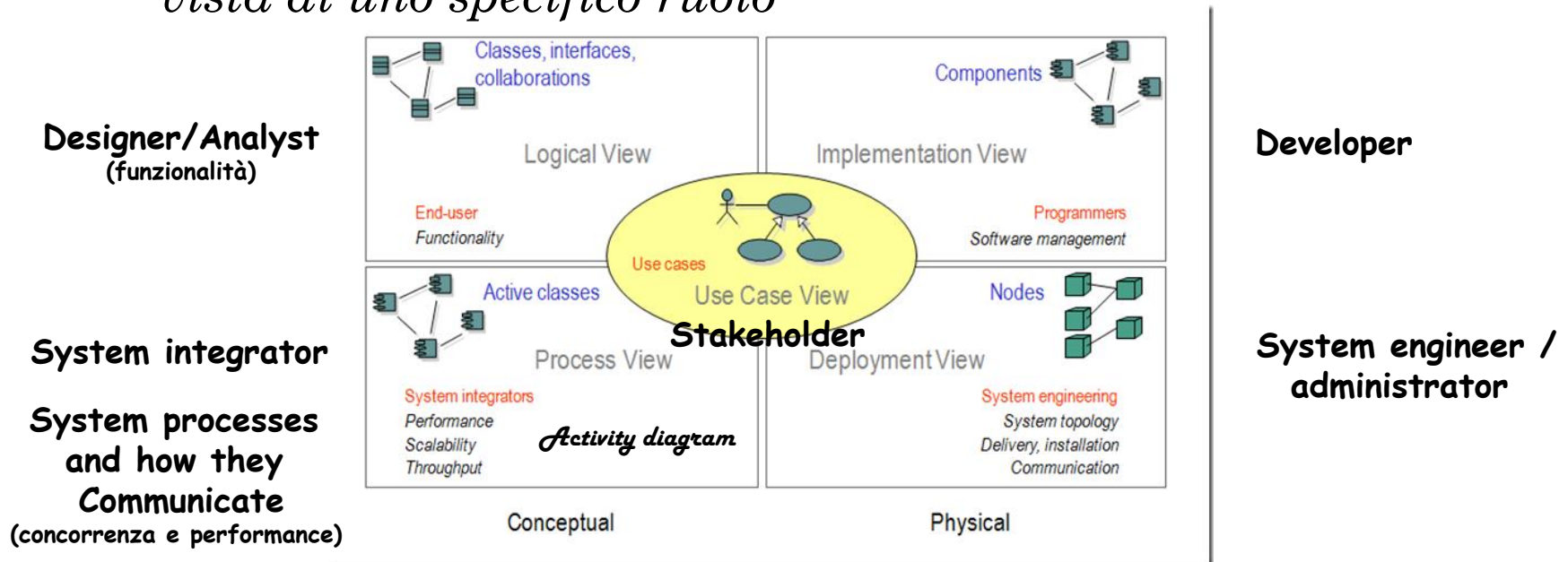
ESEMPIO DI META-MODELLO



Espresso a sua volta in UML

VISTE, DIAGRAMMI E MODELLO (IN UML)

- In UML, un sistema viene descritto utilizzando diverse **viste** e diversi **diagrammi**:
 - Vista = “*particolare aspetto di un Sistema dal punto di vista di uno specifico ruolo*”



- Un diagramma descrive/presenta **una view/subview**
- Chiameremo **Modello** un insieme di diagrammi che descrivono aspetti diversi dello stesso sistema

VISTE, DIAGRAMMI E MODELLO (IN UML)

- In UML, un sistema viene descritto utilizzando diverse **viste** e diversi **diagrammi**:
 - Vista = “*particolare aspetto di un Sistema dal punto di vista di uno specifico ruolo*”



Deployment diagram

Attenzione in un modello:

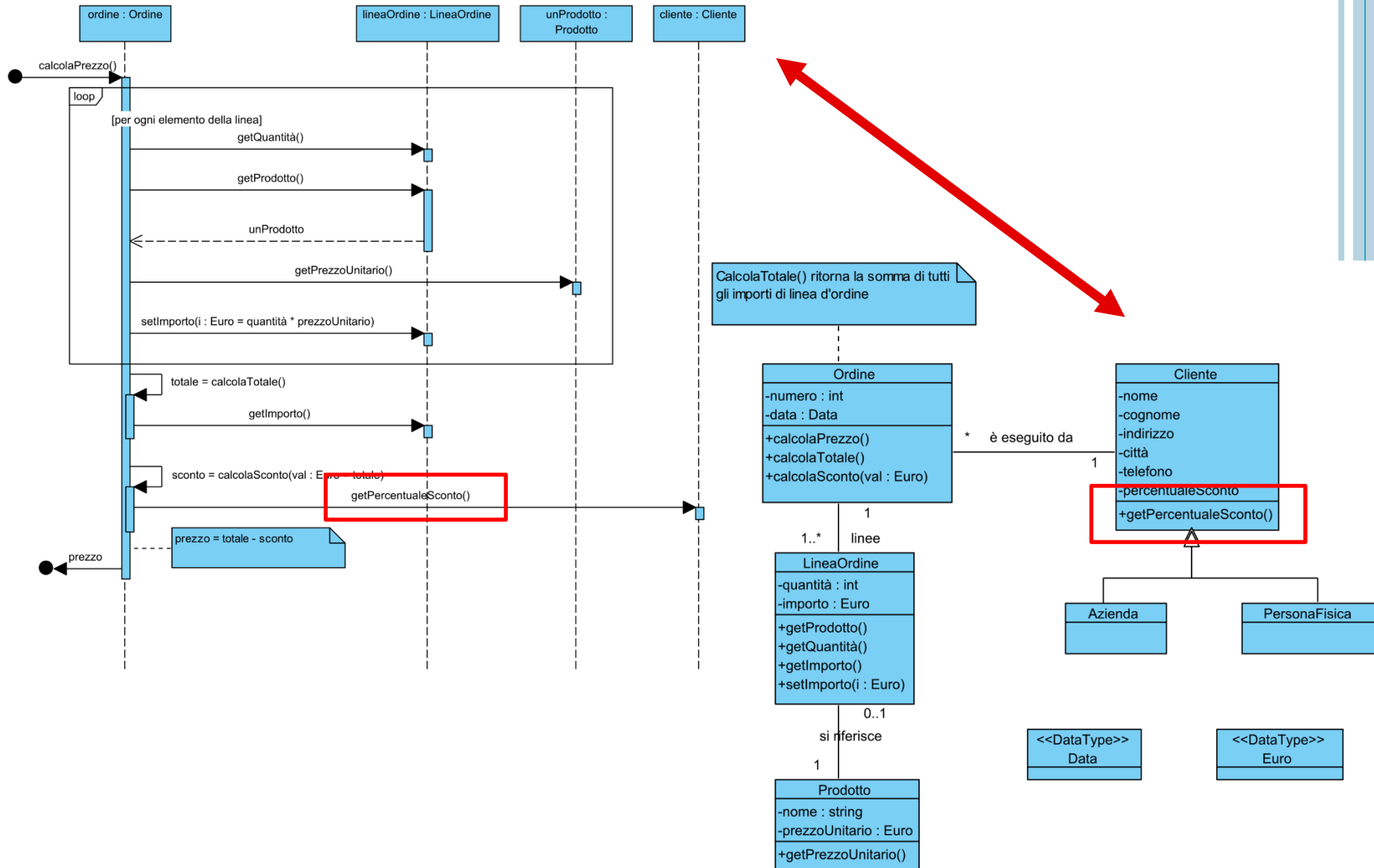
- viste possibilmente sovrapposte (problema consistenza)
- viste possibilmente incomplete (problema completezza)



- Un diagramma descrive/presenta **una view/subview**
- Chiameremo **Modello** un insieme di diagrammi che descrivono aspetti diversi dello stesso sistema

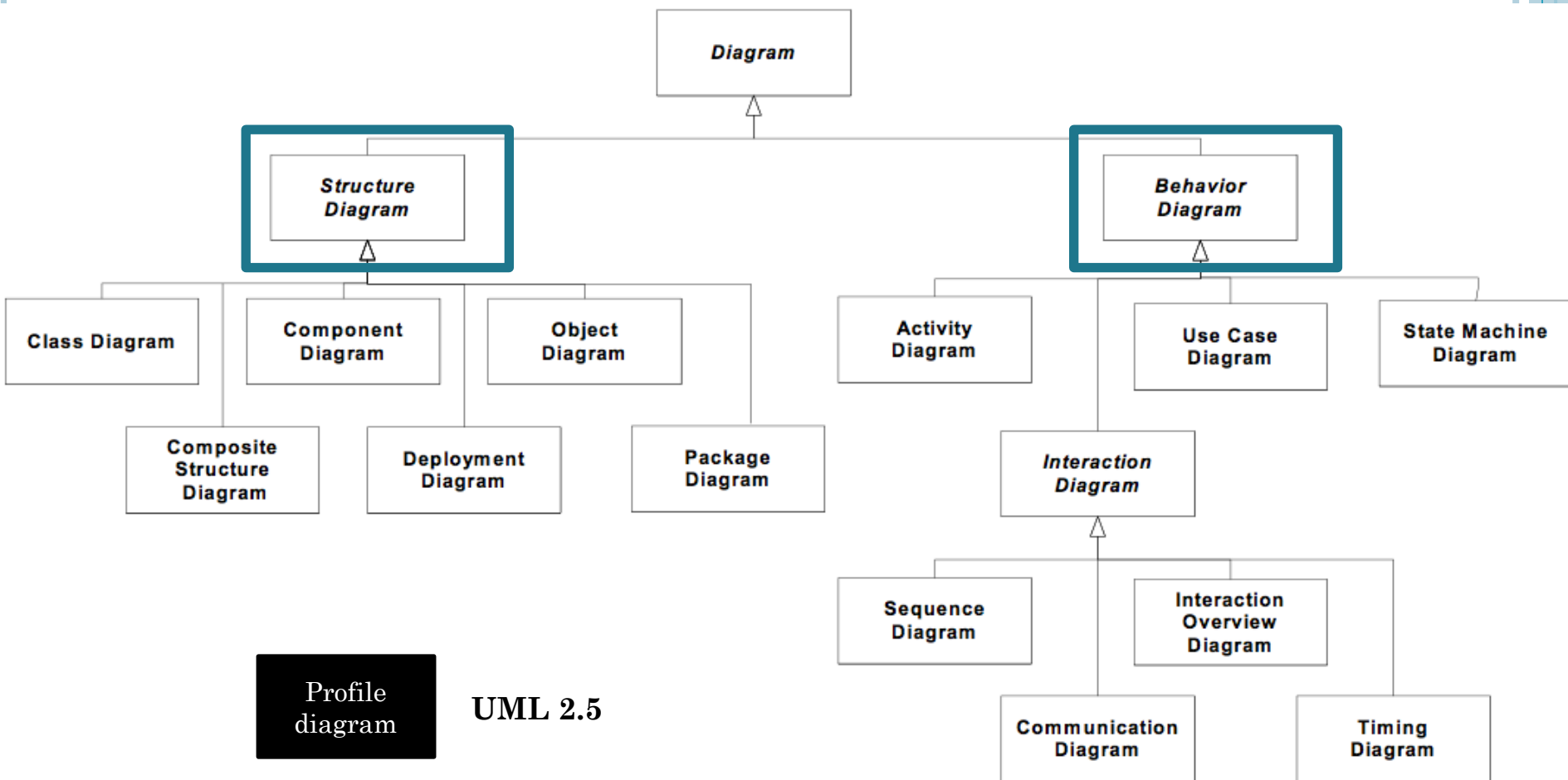
ESEMPIO: DESIGN DETTAGLIATO

Visual Paradigm

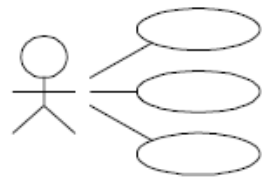


DIAGRAMMI UML

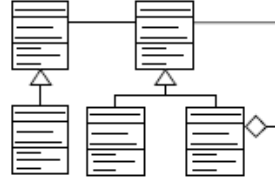
UML 2.0 comprende 13 tipi di diagrammi (nel 2.5 sono 14)



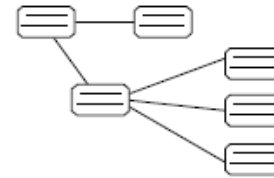
QUELLI CHE VEDREMO



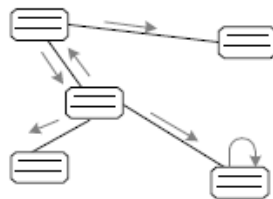
Use Case
(Casi D'uso)



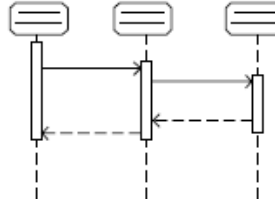
Class Diagram
(Diagrammi delle Classi)



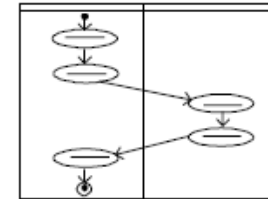
Object Diagram
(Diagrammi degli Oggetti)



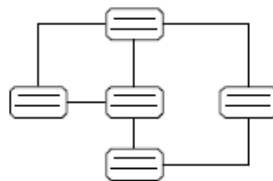
Communication Diagram
(Diagramma di comunicazione)



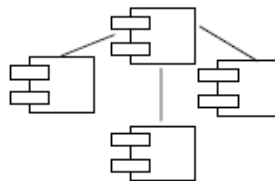
Sequence Diagram
(Diagramma di Sequenza)



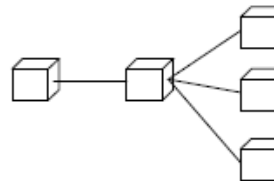
Activity Diagram
(Diagramma di Attivita')



State machine Diagram
(dmacchina a stati)

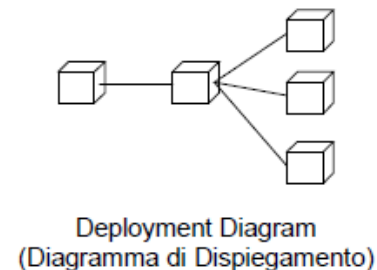
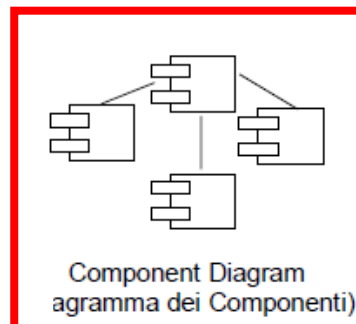
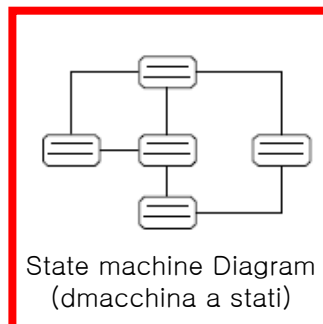
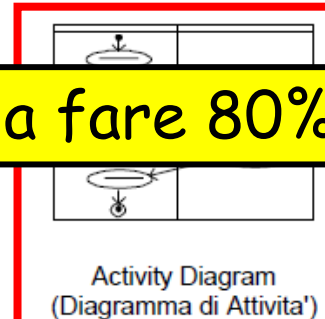
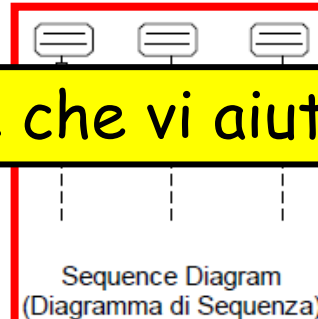
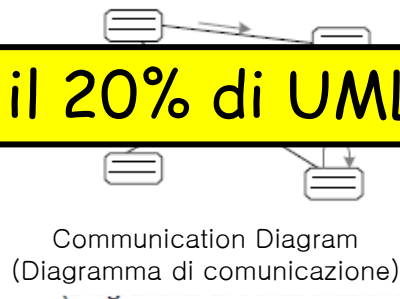
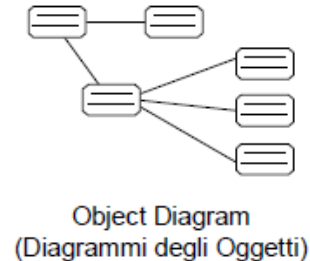
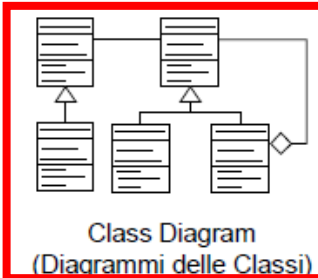
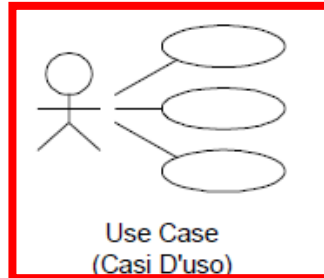
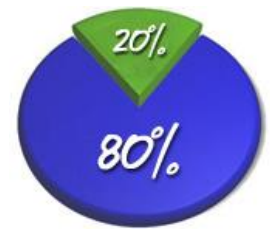


Component Diagram
agramma dei Componenti)



Deployment Diagram
(Diagramma di Dispiegamento)

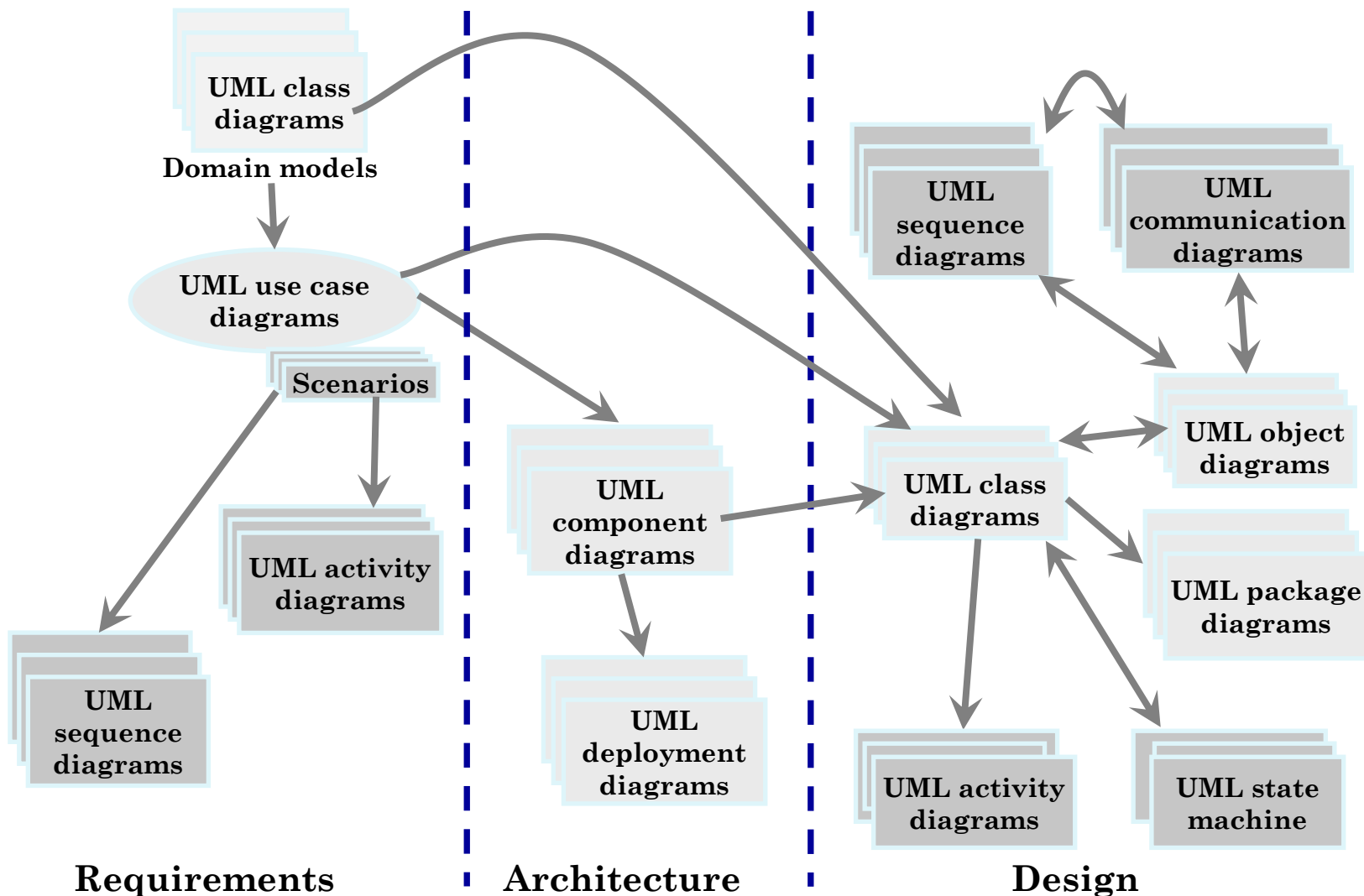
QUELLI CHE VEDREMO



Ovvero il 20% di UML che vi aiuta a fare 80% del lavoro!

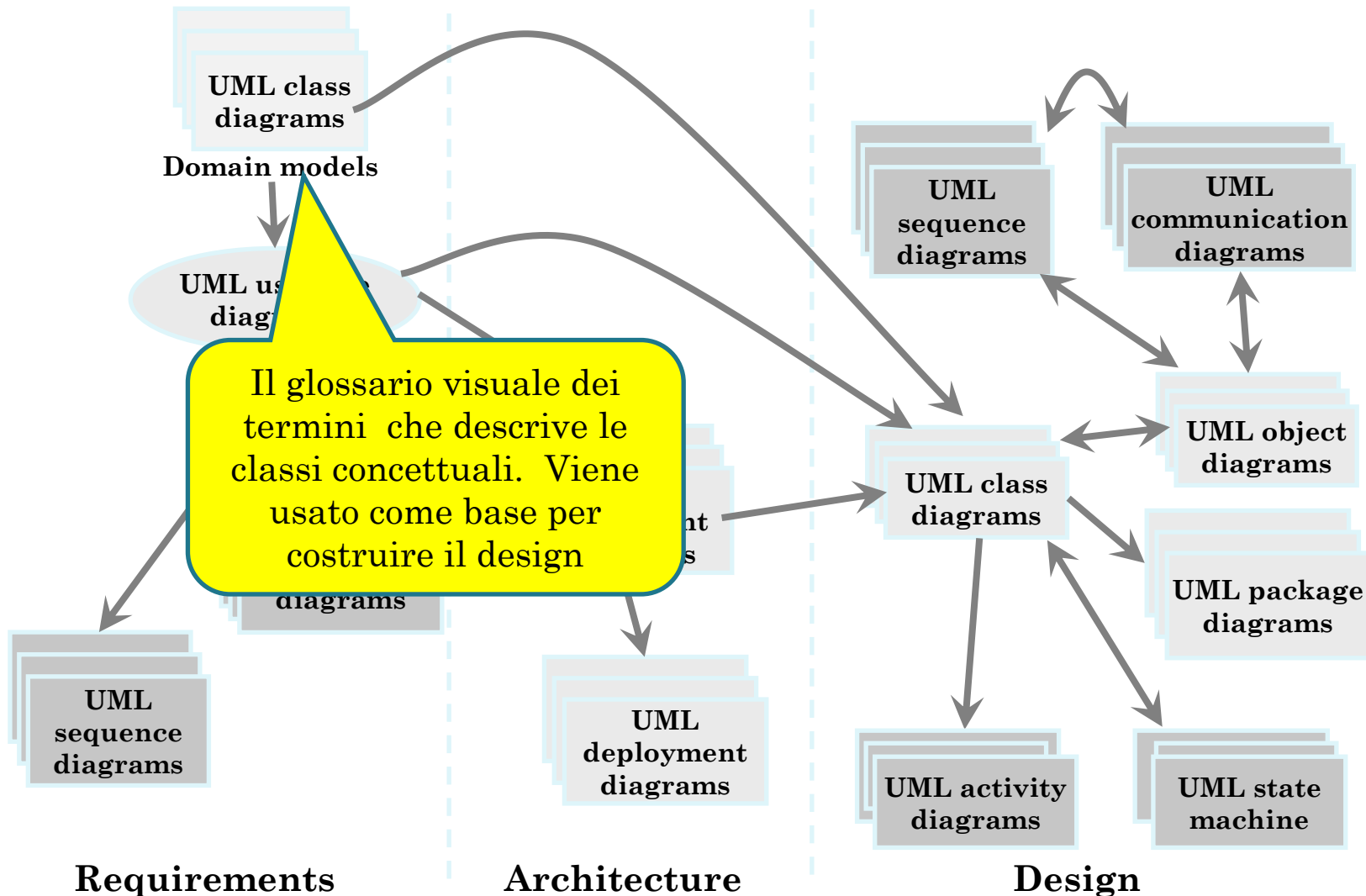
INTEGRARE UML NEL PROCESSO

- UML può esser usato con qualsiasi processo nelle seguenti fasi:



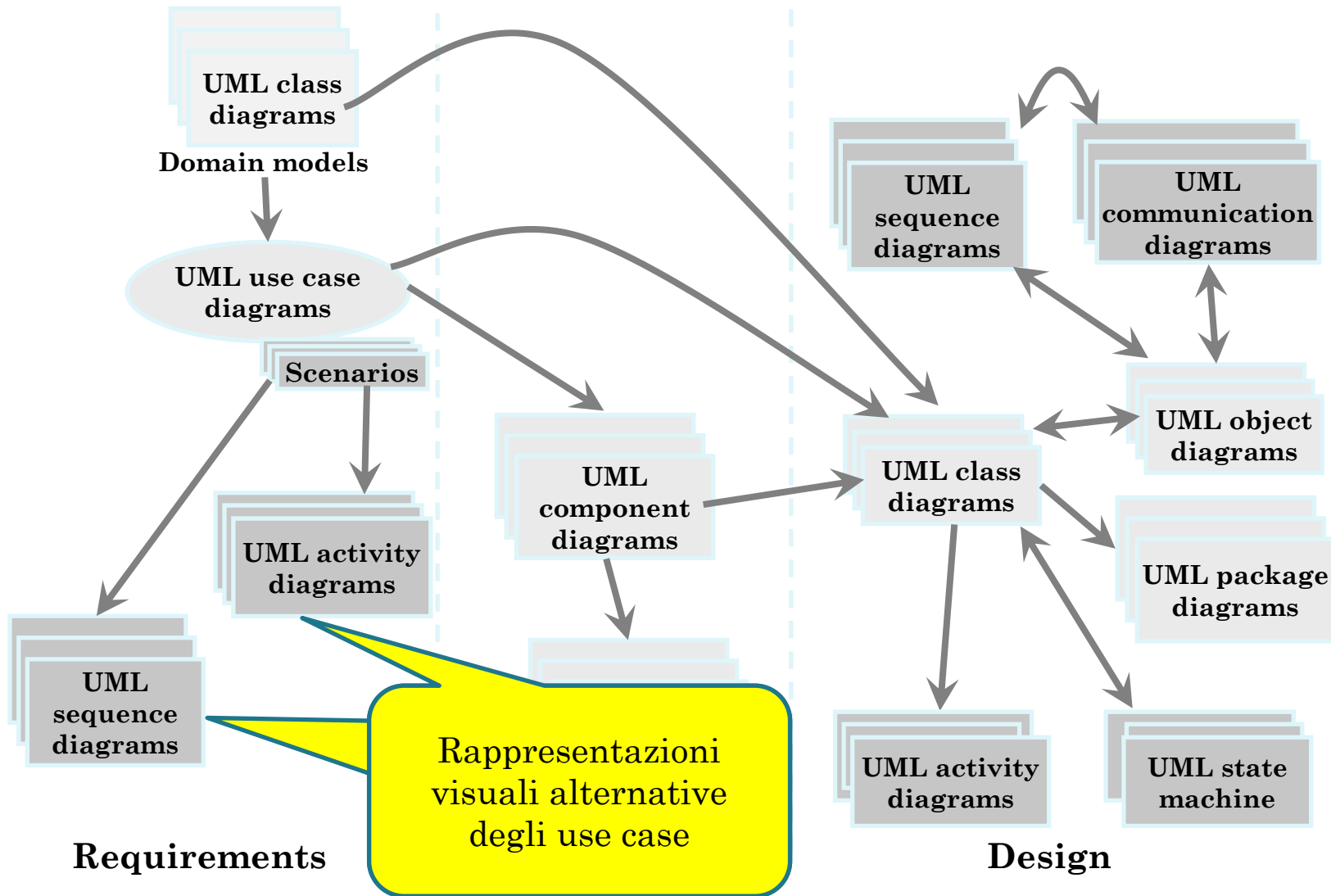
INTEGRARE UML NEL PROCESSO

- UML può esser usato con qualsiasi processo nelle seguenti fasi:



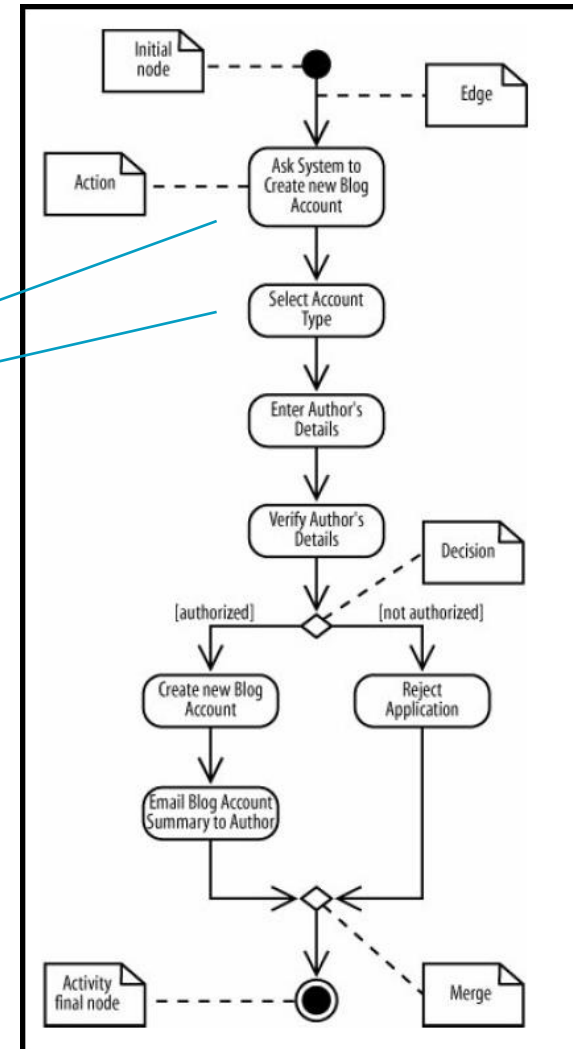
INTEGRARE UML NEL PROCESSO

- UML può esser usato con qualsiasi processo nelle seguenti fasi:



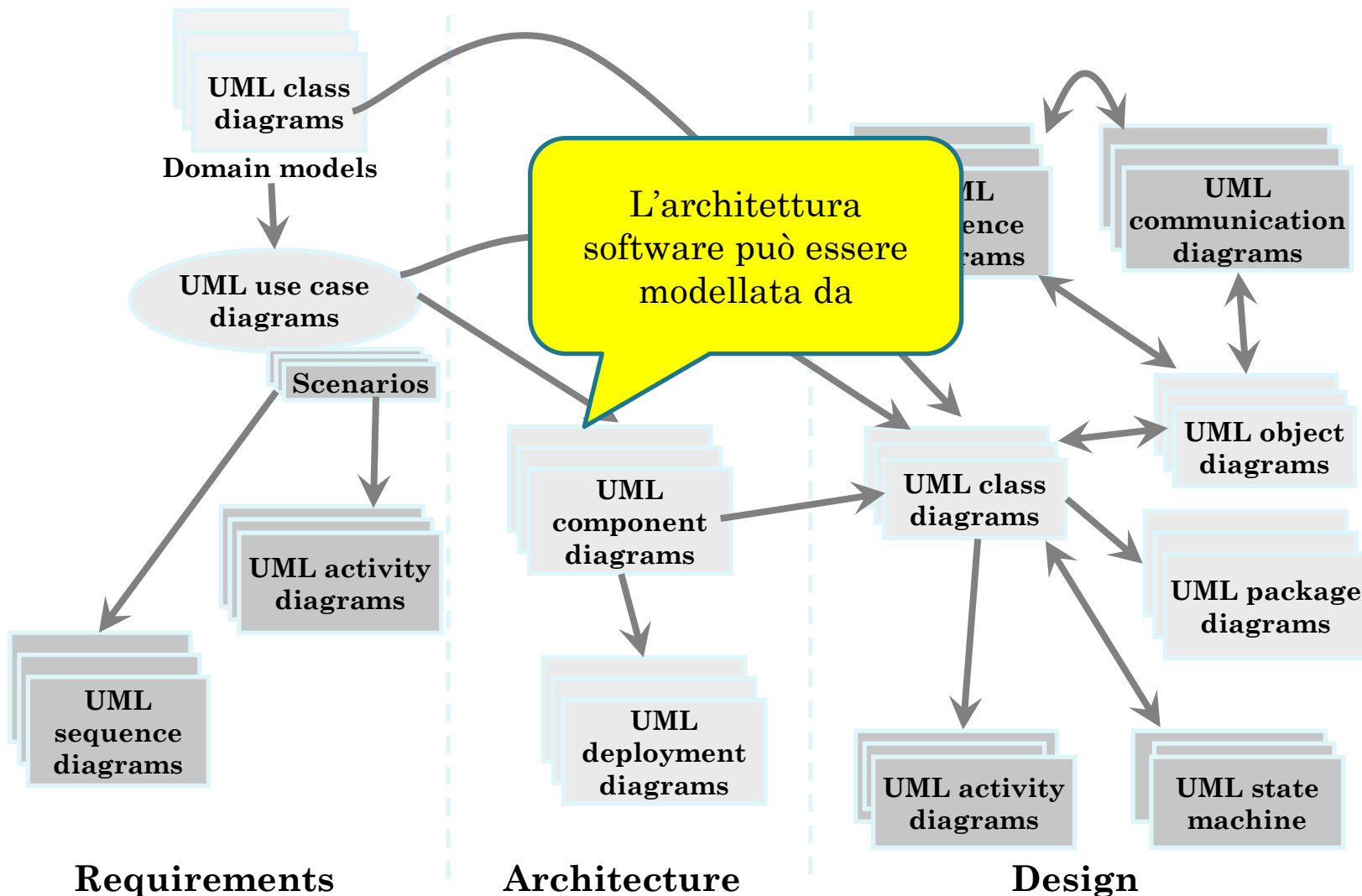
USE CASE RAPPRESENTATO CON ACTIVITY DIAGRAM

| Use case name | Create a new Blog Account | | | | | | | | | | | | | | |
|--------------------------|---|------|------------------|-----|---|-----|--|---|--|---|--|---|----------------------------------|---|--|
| Related Requirements | Requirement A.1. | | | | | | | | | | | | | | |
| Goal In Context | A new or existing author requests a new blog account from the Administrator. | | | | | | | | | | | | | | |
| Preconditions | The system is limited to recognized authors, and so the author needs to have appropriate proof of identity. | | | | | | | | | | | | | | |
| Successful End Condition | A new blog account is created for the author. | | | | | | | | | | | | | | |
| Failed End Condition | The application for a new blog account is rejected. | | | | | | | | | | | | | | |
| Primary Actors | Administrator. | | | | | | | | | | | | | | |
| Secondary Actors | Author Credentials Database. | | | | | | | | | | | | | | |
| Trigger | The Administrator asks the Content Management System to create a new blog account. | | | | | | | | | | | | | | |
| Main Flow | <table><tr><th>Step</th><th>Action</th></tr><tr><td>1</td><td>The Administrator asks the system to create a new blog account.</td></tr><tr><td>2</td><td>The Administrator selects an account type.</td></tr><tr><td>3</td><td>The Administrator enters the author's details.</td></tr><tr><td>4</td><td>The author's details are verified using the Author Credentials Database.</td></tr><tr><td>5</td><td>The new blog account is created.</td></tr><tr><td>6</td><td>A summary of the new blog account's details are emailed to the author.</td></tr></table> | Step | Action | 1 | The Administrator asks the system to create a new blog account. | 2 | The Administrator selects an account type. | 3 | The Administrator enters the author's details. | 4 | The author's details are verified using the Author Credentials Database. | 5 | The new blog account is created. | 6 | A summary of the new blog account's details are emailed to the author. |
| Step | Action | | | | | | | | | | | | | | |
| 1 | The Administrator asks the system to create a new blog account. | | | | | | | | | | | | | | |
| 2 | The Administrator selects an account type. | | | | | | | | | | | | | | |
| 3 | The Administrator enters the author's details. | | | | | | | | | | | | | | |
| 4 | The author's details are verified using the Author Credentials Database. | | | | | | | | | | | | | | |
| 5 | The new blog account is created. | | | | | | | | | | | | | | |
| 6 | A summary of the new blog account's details are emailed to the author. | | | | | | | | | | | | | | |
| Extensions | <table><tr><th>Step</th><th>Branching Action</th></tr><tr><td>4.1</td><td>The Author Credentials Database does not verify the author's details.</td></tr><tr><td>4.2</td><td>The author's new blog account application is rejected.</td></tr></table> | Step | Branching Action | 4.1 | The Author Credentials Database does not verify the author's details. | 4.2 | The author's new blog account application is rejected. | | | | | | | | |
| Step | Branching Action | | | | | | | | | | | | | | |
| 4.1 | The Author Credentials Database does not verify the author's details. | | | | | | | | | | | | | | |
| 4.2 | The author's new blog account application is rejected. | | | | | | | | | | | | | | |



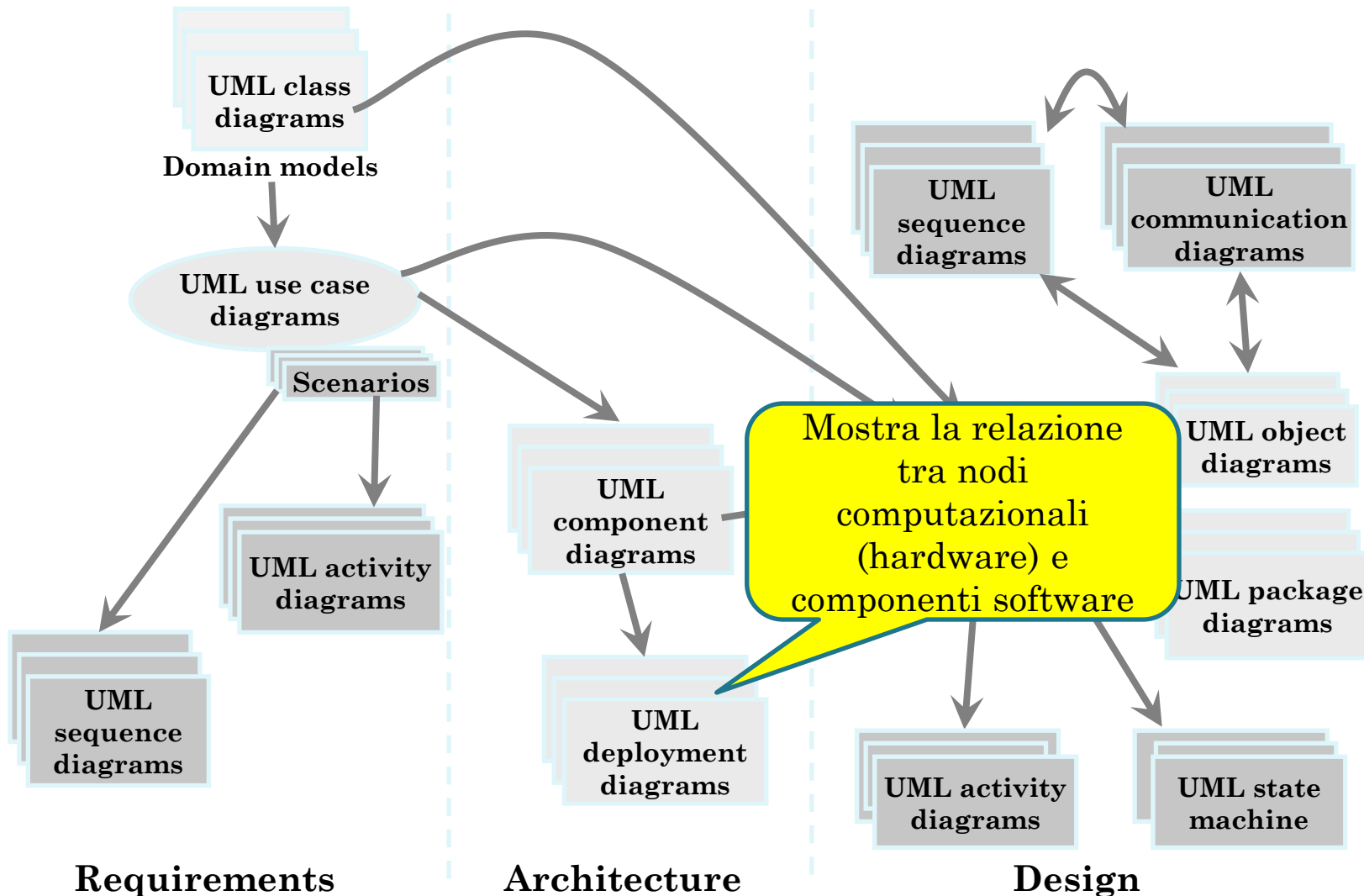
INTEGRARE UML NEL PROCESSO

- UML può esser usato con qualsiasi processo nelle seguenti fasi:



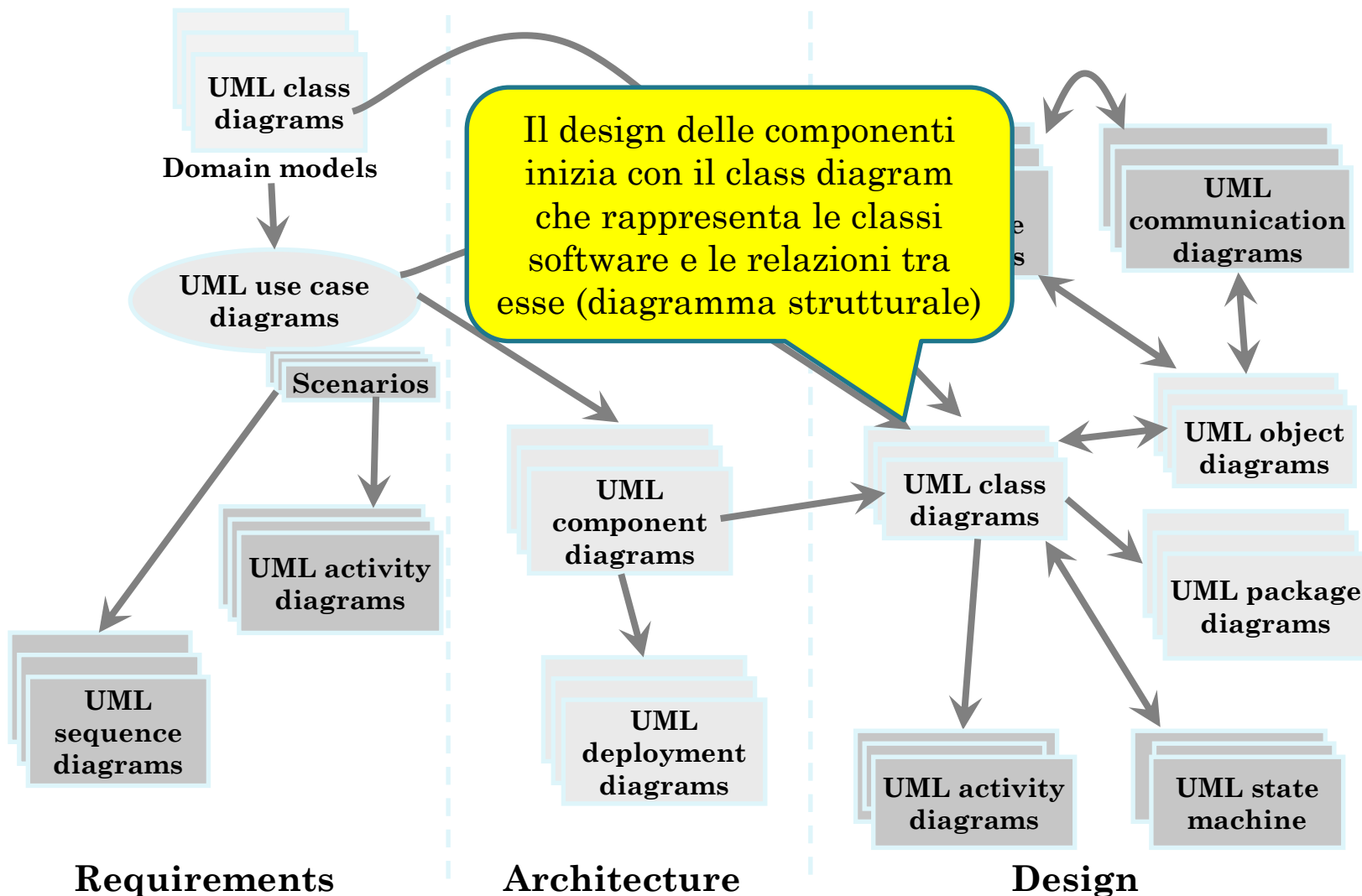
INTEGRARE UML NEL PROCESSO

- UML può esser usato con qualsiasi processo nelle seguenti fasi:



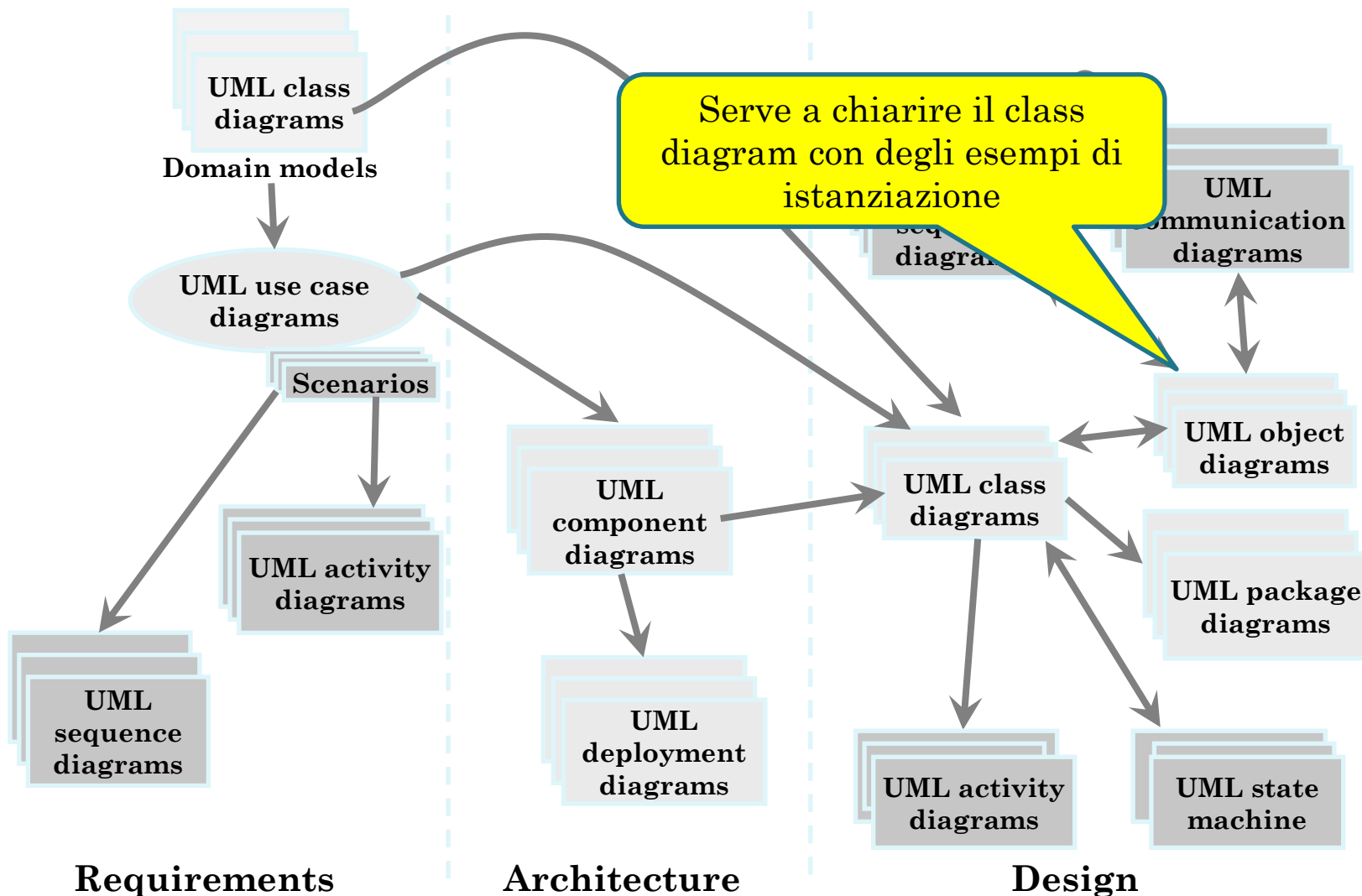
INTEGRARE UML NEL PROCESSO

- UML può esser usato con qualsiasi processo nelle seguenti fasi:



INTEGRARE UML NEL PROCESSO

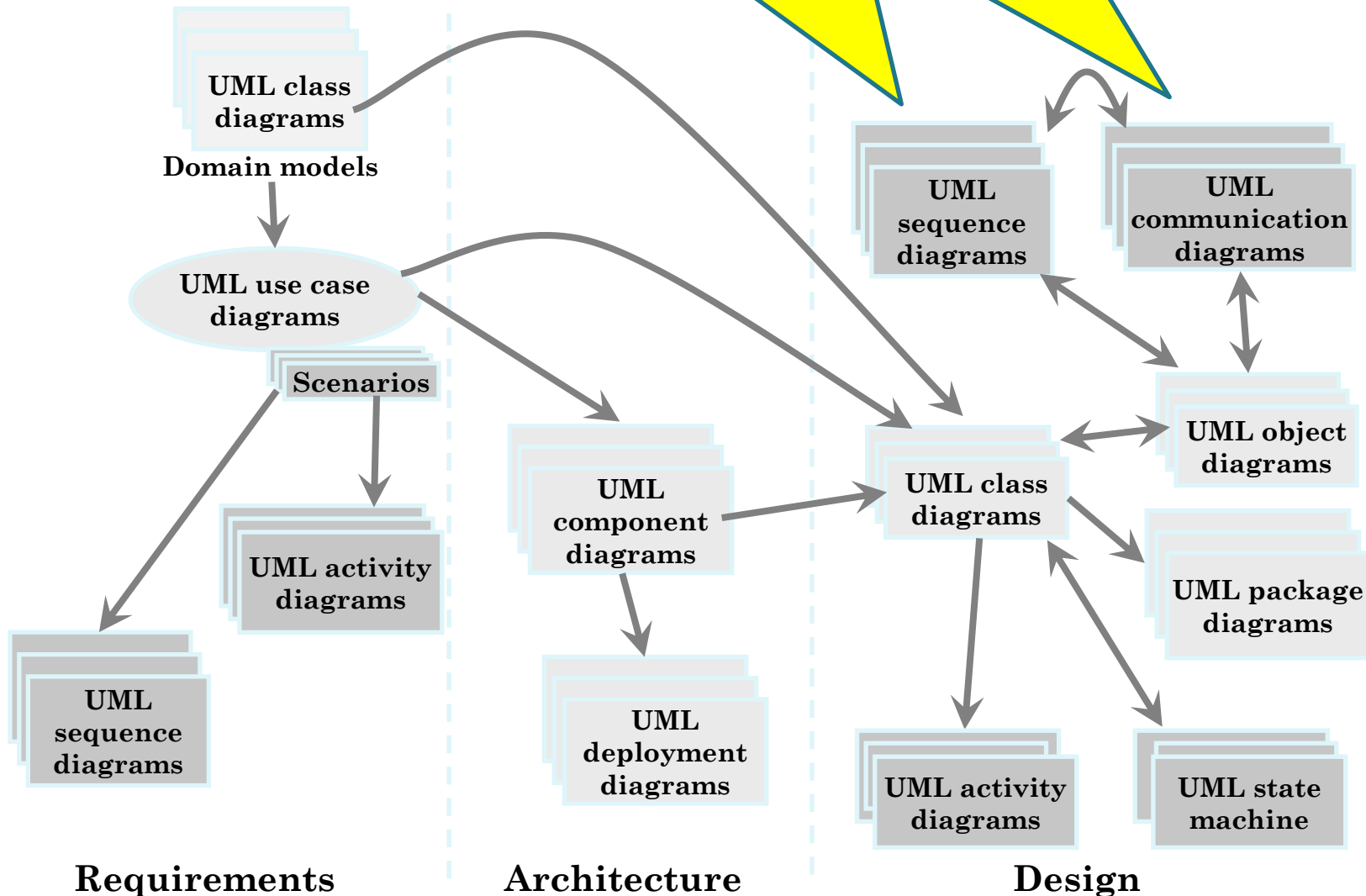
- UML può esser usato con qualsiasi processo nelle seguenti fasi:



INTEGRARE UML NEL PROCESSO

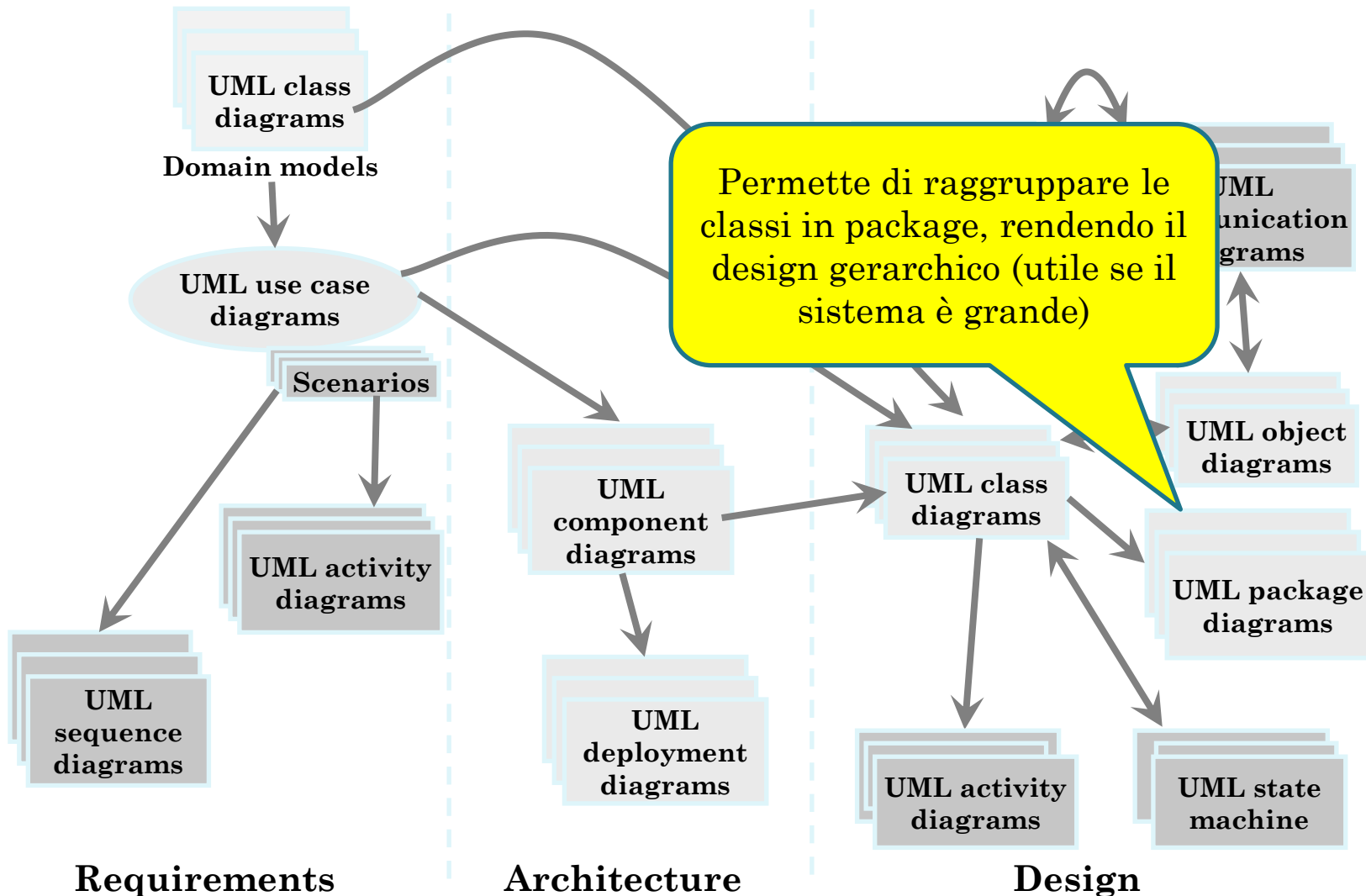
- UML può esser usato nelle seguenti fasi:

Mostrano come i messaggi fluiscono da un oggetto ad un altro



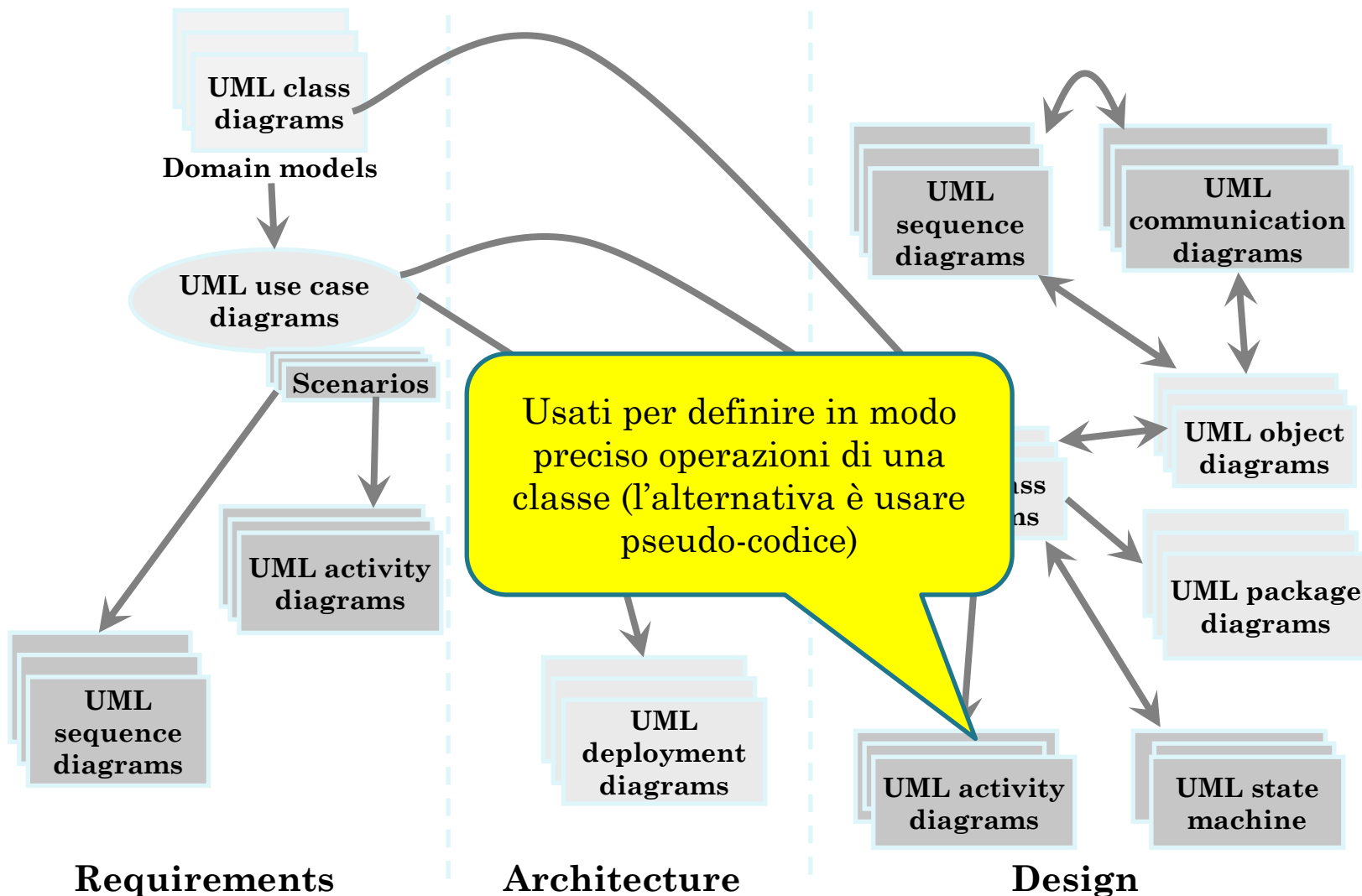
INTEGRARE UML NEL PROCESSO

- UML può esser usato con qualsiasi processo nelle seguenti fasi:



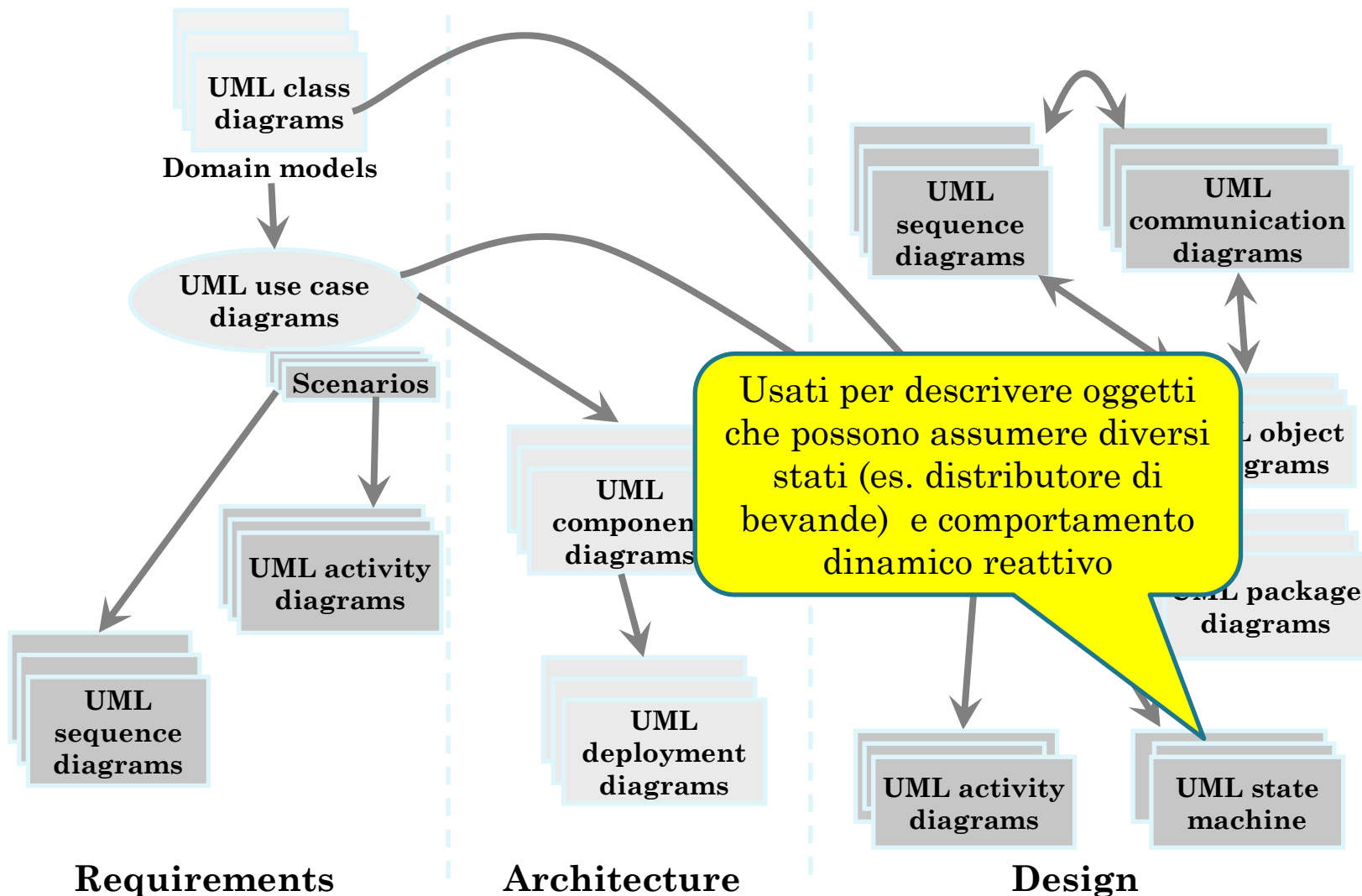
INTEGRARE UML NEL PROCESSO

- UML può esser usato con qualsiasi processo nelle seguenti fasi:



INTEGRARE UML NEL PROCESSO

- UML può esser usato con qualsiasi processo nelle seguenti fasi:

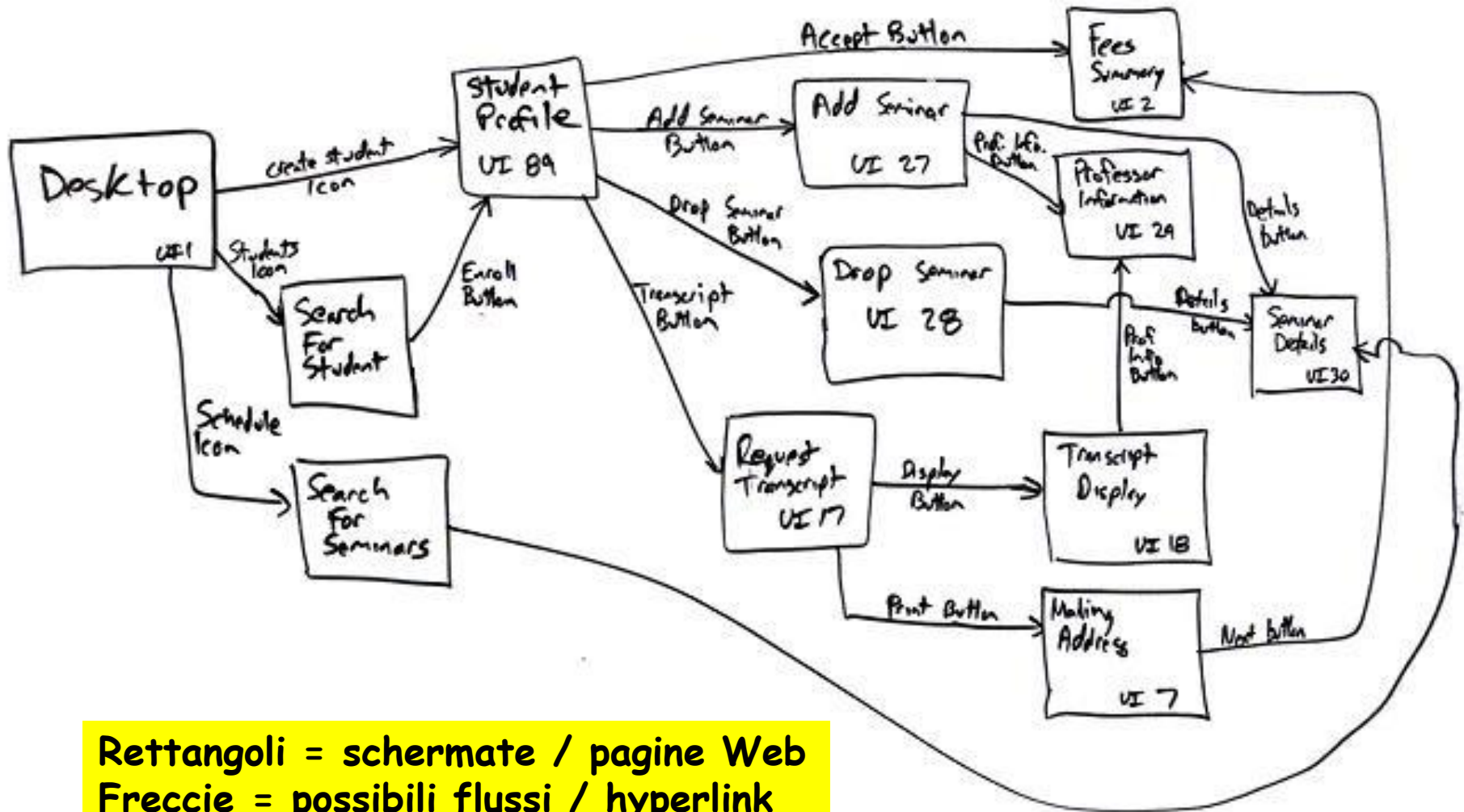


UML NON È ABBASTANZA

- Benchè i diagrammi UML siano tanti l'insieme non è esaustivo ...
- Esistono **due soluzioni** a questo problema:
 1. Si cerca o si produce un “estensione” di UML
 - Vedi il concetto di profilo (+ avanti)
 2. Si usano diagrammi non UML
 - M. Fowler: *“non esitare a farlo se UML non ne comprende uno adatto al vostro scopo”*

Ma si perde uno dei vantaggi di UML, la standardizzazione

DIAGRAMMA DELLE SCHERMATE (STORYBOARD)



Rettangoli = schermate / pagine Web
Freccie = possibili flussi / hyperlink

PROFILO

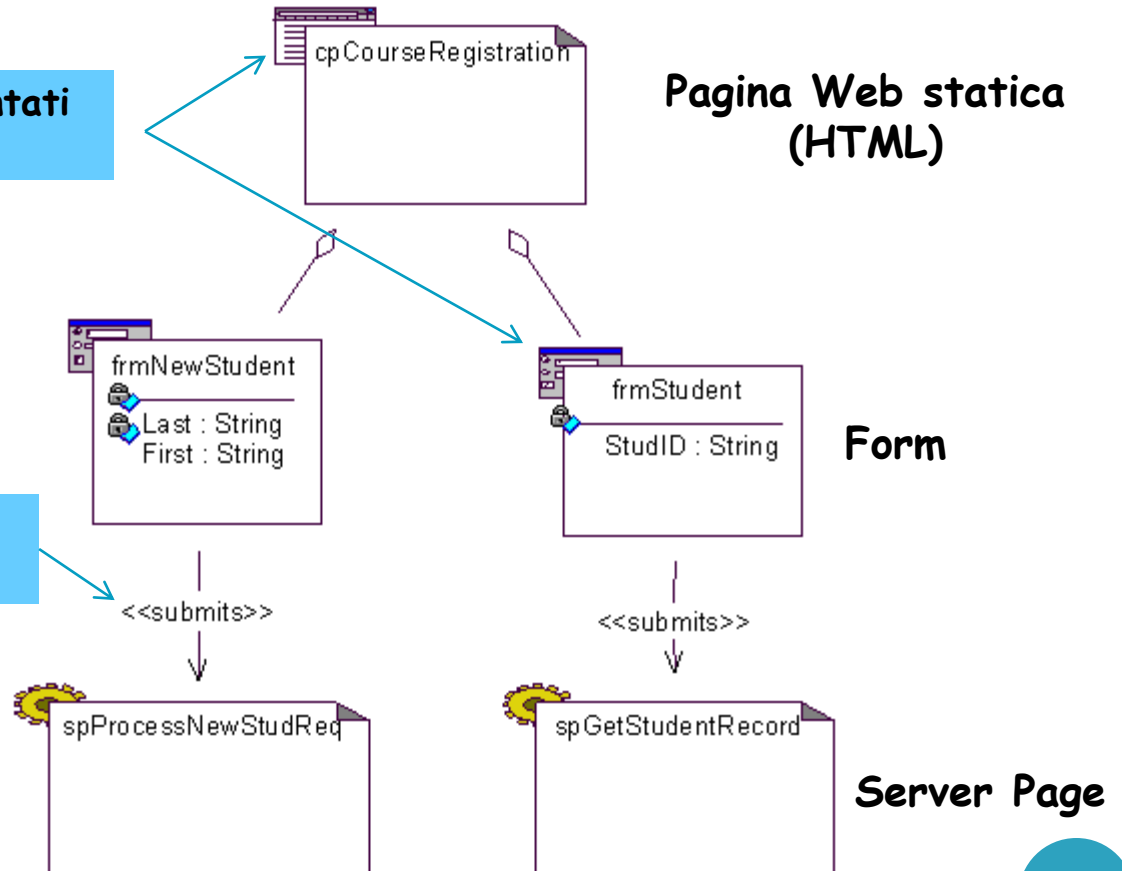
- UML si può **estendere** tramite il concetto di **profilo** per avere la possibilità di specificare particolari **domini applicativi** o **tipologie di applicazioni**
 - Es.
 - **Dominio applicativo:** dominio ferroviario, spaziale
 - **Tipologia:** applicazioni WEB o sistemi embedded
- Un **profilo** è costituito principalmente da:
 - **Stereotipi**, ovvero elementi aggiuntivi ottenuti modificando quelli esistenti
 - **Vincoli aggiuntivi** (Constraint) rispetto a quelli standard presenti in UML
 - per *vietare* l'uso di alcuni elementi del linguaggio
 - **Informazioni semantiche aggiuntive** relative agli elementi aggiuntivi

PROFILO PER APPLICAZIONI WEB (CONALLEN)

Stereotipi rappresentati
da icone nuove

Pagina Web statica
(HTML)

Altra possibilità per rappresentare
gli stereotipi è usare << ... >>



UML E IL CODICE ...

- **Aspetto fastidioso:** non esiste una definizione della corrispondenza tra UML e un qualsiasi linguaggio di programmazione ...
- Non si può guardare un diagramma UML e dire esattamente cosa farà il codice corrispondente ...
 - Ma solo avere una buona approssimazione di cosa farà ...
- Modelli UML sono ad un livello di astrazione più alto!!
 - Unica eccezione se usiamo UML come linguaggio di programmazione (estendendolo)



QUALE UML È LEGALE?

- Quello definito dalla specifica del linguaggio
 - Ovvero dal documento OMG
 - <http://www.omg.org/>, cercare UML
- Ma UML è molto complesso e si presta a multiple interpretazioni
- Così le persone lo usano adottando **convenzioni particolari**
- **“Uno, due molti UML ...” [Fowler]**
 - normativo/standard/legale rispetta lo standard
 - UML convenzionale che non rispetta lo standard ma di ampio uso
 - Di solito si aggiunge nei diagrammi **“non-normativo”**

QUALE UML È LEGALE?

- Quello definito dalla specifica del linguaggio
 - Ovvero dal documento OMG
 - <http://www.omg.org/>, cercare UML
- Ma UML è molto complesso e si presta a multiple

Dato che lo scopo primario dei diagrammi UML è quello di comunicare è meglio capirsi che rispettare lo standard [Fowler]

- **“Uno, due molti UML ...” [Fowler]**
 - normativo/standard/legale rispetta lo standard
 - UML convenzionale che non rispetta lo standard ma di ampio uso
 - Di solito si aggiunge nei diagrammi **“non-normativo”**

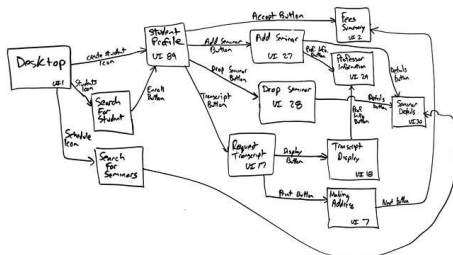
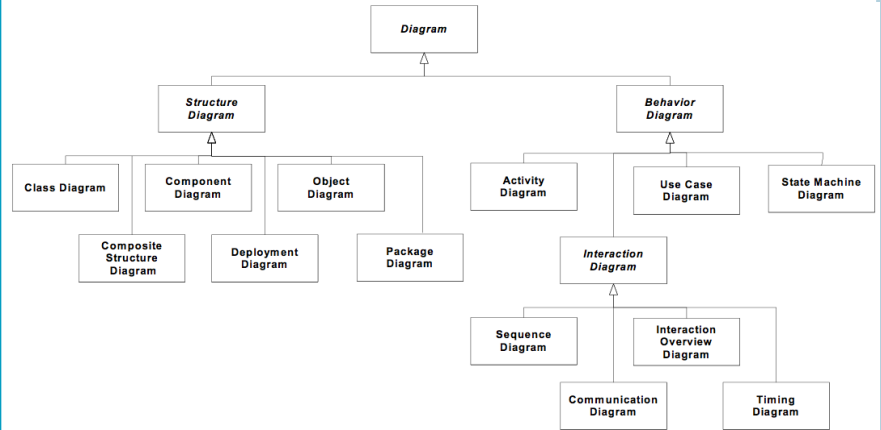
CONSIGLI SU COME USARE UML

1. Capire il punto di vista dell'autore per capire i diagrammi
 - Utilizzo/Prospettiva
 - Es. Abbozzo/Software
2. Meglio dei buoni diagrammi “**illegali**” che altri formalmente corretti ma ‘scarsi’
3. UML è molto complesso da prestarsi a multiple interpretazioni: usate un sottoinsieme e solo i diagrammi che ritenete utili (e che avete compreso bene)
4. La completezza è nemica della chiarezza
5. Attenzione: **qualsiasi info può essere sempre soppressa**
 - l'assenza di qualcosa non dimostra nulla!
 - Es. X: int senza inizializzazione, il valore esiste ma è stato omesso oppure non esiste e quindi si inizializza con il valore di default?

RIASSUNTO

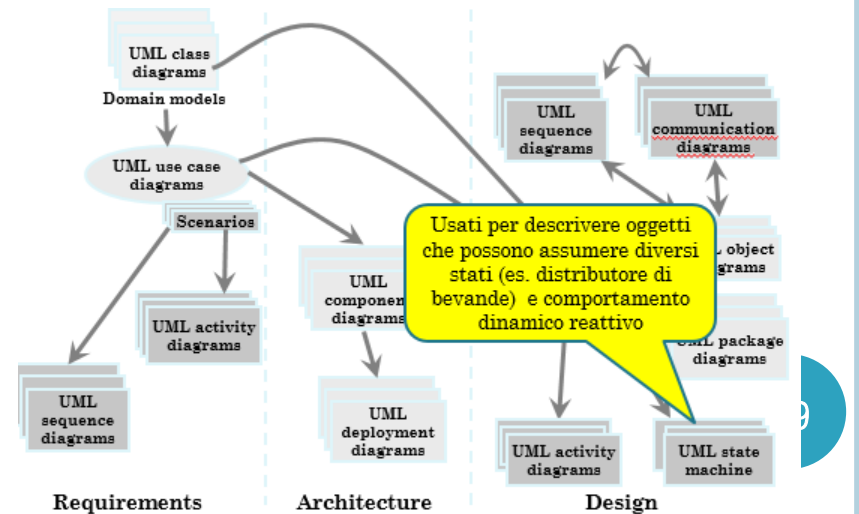
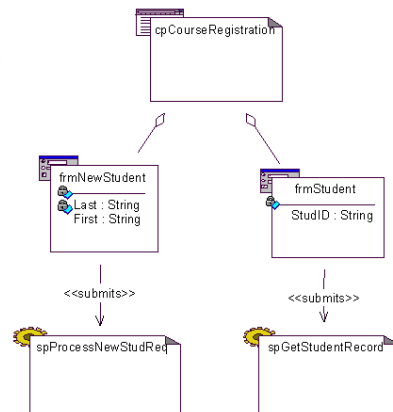


Cosa è, cosa non è, storia, come usarlo



Diagrammi aggiuntivi

Profilo



INFO SU UML

- Definizioni ufficiali (<http://www.omg.org/>)
 - Unified Modeling Language: Superstructure
 - Il documento di riferimento
 - sintassi e semantica
- J. Rumbaugh, G. Booch and I. Jacobson. *The Unified Modeling Language Reference Manual* (piuttosto vecchio)
- M. Fowler, *UML Distilled: A Brief Guide to the Standard Object Modeling Language*
- R. S. Pressman, *Principi di Ingegneria del software* 5/ed,
 - Capitolo 7 relativo ad UML (a cura di Cerioli e Reggio)
- **Attenzione al materiale che si trova su Web**
 - **attenzione anche a cose che si riferiscono a UML 1**

THE END ...



Domande?