



Dalla progettazione concettuale alla modellazione di dominio

Capitolo 91
marzo 2012

Se qualcuno vi avvicinasse in un vicolo buio dicendo "psst, vuoi vedere un diagramma UML?":
ci sono buone probabilità che sarebbe un
diagramma delle classi.

Martin Fowler

Introduzione

Lo studente dovrebbe essere familiare con le nozioni relative alla **progettazione concettuale** di basi di dati, parte della progettazione di basi di dati, studiate nel corso di *Basi di dati*

- queste nozioni sono *simili* (attenzione, non *uguali*) a quelle della **modellazione di dominio**, parte dell'analisi orientata agli oggetti

Vogliamo allora comprendere questa similitudine, in termini di

- nozioni analoghe
- nozioni differenti

A P S Progettazione concettuale

La **progettazione concettuale** di basi di dati ha lo scopo di

- “rappresentare le specifiche informali della realtà di interesse
- in termini di una descrizione formale e completa
- ma indipendente dai criteri di rappresentazione utilizzati nei sistemi di gestione di basi di dati”

In effetti, è un’attività di **analisi** – e non un’attività di **progettazione**

- è interessata al **che cosa** – e non al **come**
 - sarebbe pertanto più corretto chiamarla “**analisi concettuale**”
- in particolare, è interessata alle **informazioni** della realtà di interesse

A P S Modellazione di dominio

Nell’analisi orientata agli oggetti, la **modellazione di dominio** ha lo scopo di descrivere le informazioni della realtà di interesse secondo una rappresentazione concettuale e a oggetti

- in questo contesto, la “realtà di interesse” viene chiamata “dominio del problema”
- dunque, progettazione concettuale e modellazione di dominio hanno scopi *simili*
 - vedremo anche che sono attività svolte con strumenti *simili* e metodi *simili*

Osservazione

- la modellazione di dominio è un modo particolare di fare analisi orientata agli oggetti
 - esistono anche altri modi di fare analisi
 - non tutti sono interessati alle informazioni della realtà di interesse

A P S Un po' di terminologia

Nelle basi di dati

- i diagrammi si chiamano **schemi**
- i formalismi per esprimere schemi si chiamano **modelli** – ad es., il modello ER

Il risultato della progettazione concettuale

- è uno **schema concettuale**
- espresso mediante un **diagramma di un modello concettuale** (ER)

Nell'ingegneria del software

- i diagrammi si chiamano **modelli**
- i formalismi per esprimere modelli si chiamano **linguaggi** – ad es., il linguaggio UML

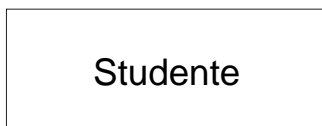
Il risultato della modellazione di dominio

- è un **modello di dominio**
- espresso mediante un **diagramma delle classi di UML** – usato dal *punto di vista concettuale*

A P S Entità e classi concettuali

ER

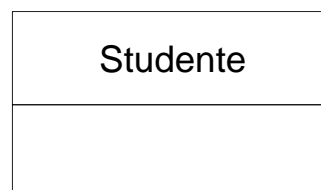
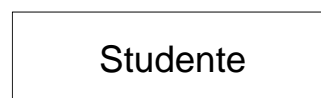
- un'**entità** rappresenta una classe di oggetti che hanno proprietà comuni



- le relative istanze sono chiamate **istanze** o **occorrenze**

UML (concettuale)

- una **classe concettuale** rappresenta un insieme di cose o concetti con caratteristiche simili



- le relative istanze sono chiamate **istanze** o **oggetti**

A P S Attributi

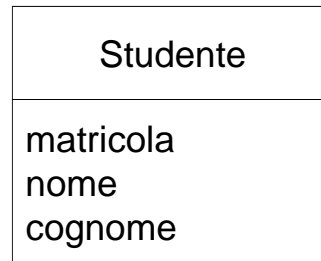
ER

- un **attributo** descrive una proprietà elementare di un'entità o di una relazione



UML (concettuale)

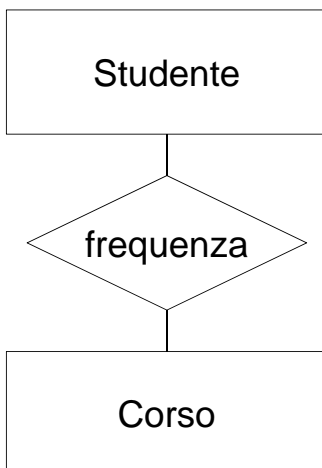
- un **attributo** rappresenta una proprietà elementare degli oggetti di una classe



A P S Relazioni e associazioni

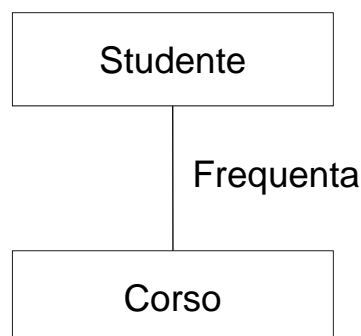
ER

- una **relazione** rappresenta un legame logico tra due o più entità



UML (concettuale)

- un'**associazione** rappresenta una relazione (una connessione significativa) tra classi



- le istanze di un'associazione si chiamano **collegamenti**

A P S Relazioni e associazioni

ER

- le relazioni possono essere binarie o anche N-arie
- le relazioni possono avere attributi
- il nome è generalmente un *sostantivo* che indica una relazione

UML (concettuale)

- le associazioni sono in genere binarie – le N-arie sono possibili, ma poco comuni
- è possibile rappresentare associazioni con attributi, ma è poco comune
- il nome è generalmente un *verbo* che indica una relazione

A P S Identificatori

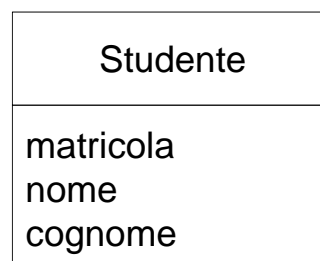
ER

- per ciascuna entità va indicato (almeno) un **identificatore**



UML (concettuale)

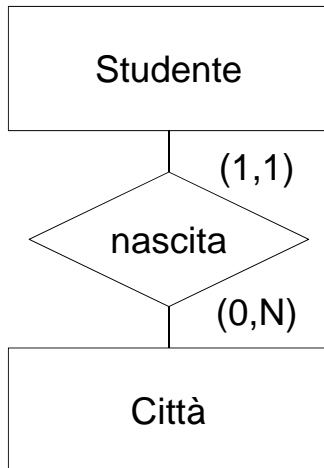
- è possibile associare identificatori alle classi, ma è poco comune



A P S Cardinalità e molteplicità

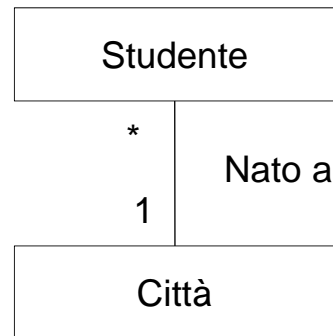
ER

- una **cardinalità** caratterizza la partecipazione (minima e massima) di un'entità a una relazione



UML (concettuale)

- una **molteplicità** indica quante istanze di una classe possono essere associate a un'istanza dell'altra classe



A P S Cardinalità e molteplicità

ER

UML (concettuale)

(A,B)	A..B
(A,A)	A
(0,N)	*
(1,N)	1..*

Attenzione, le posizioni sono invertite!

A P S Generalizzazioni

La nozione di generalizzazione è presente sia nel modello ER che in UML

- la semantica delle generalizzazioni nei due formalismi è però un po' differente
- in ogni caso, per ora non le prendiamo in considerazione

A P S Altre differenze (significative!)

ER

- per rappresentare solo informazioni che devono essere gestite in modo persistente
- un'entità rappresenta sia una classe di istanze che il relativo "insieme"
- in genere nessuna entità ha una sola istanza
- le entità hanno in genere almeno un attributo
- le entità rappresentano informazioni

UML (concettuale)

- per rappresentare tutte le informazioni che devono essere gestite da un'applicazione
- una classe rappresenta una classe di oggetti – ma non il relativo "insieme"
- può essere ok avere classi che hanno un solo oggetto
- può essere ok avere classi senza attributi – con cautela
- può essere ok avere classi che rappresentano solo comportamento – con cautela

A P S Dalla PCBD alla MD

Alcune idee importanti della modellazione di dominio corrispondono (con alcune varianti) a quelle della progettazione concettuale di basi di dati

- bisogna rappresentare le informazioni del dominio del problema – senza preoccuparsi dell'implementazione di queste informazioni
- bisogna concentrarsi solo su ciò che è rilevante ai fini della descrizione del dominio del problema
- le classi concettuali rappresentano classi di oggetti o fatti che hanno proprietà comuni ed esistenza autonoma
- le associazioni rappresentano legami logici significativi tra due classi concettuali
- gli attributi descrivono le proprietà elementari delle classi concettuali

A P S Dalla PCBD alla MD

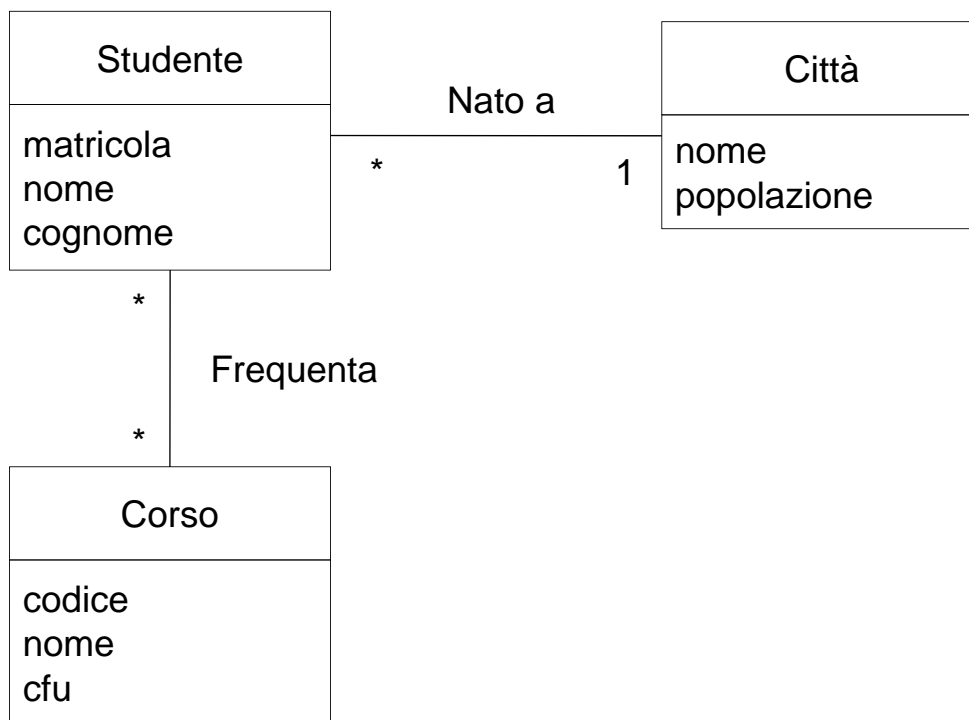
Criteri generali di rappresentazione

- se un concetto ha proprietà significative e/o descrive una classe (insieme) di oggetti con esistenza autonoma, è opportuno rappresentarlo come una classe concettuale
- se un concetto ha una struttura semplice e pensiamo ad esso come ad un valore, è opportuno rappresentarlo come un attributo, associato alla classe concettuale a cui si riferisce
- se sono state individuate due classi concettuali e c'è un concetto che le associa, questo concetto può essere rappresentato come un'associazione

Strategie

- top down, bottom up, inside-out (a macchia d'olio), mista

A P S Esempio di modello di dominio (parziale)



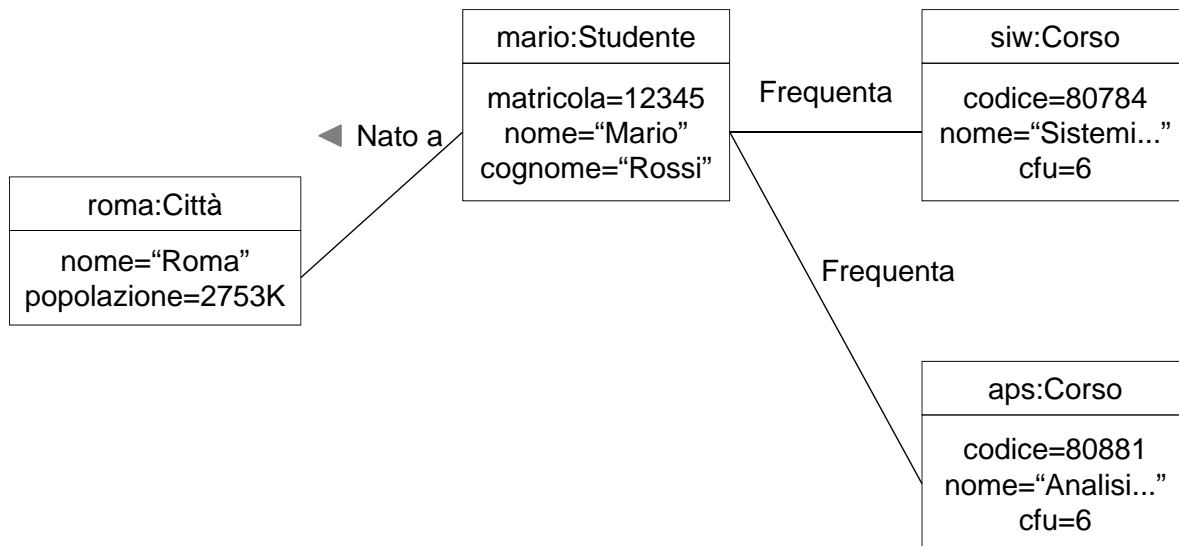
A P S Oggetti di dominio

Un diagramma degli **oggetti di dominio** rappresenta, in modo visuale, un insieme di oggetti di esempio del mondo reale – nel contesto del dominio del problema

- un grafo di oggetti
 - ciascun nodo rappresenta un oggetto – etichettato con il nome di una classe
 - ciascun oggetto è decorato con valori per gli attributi della classe
 - ciascun arco rappresenta un collegamento – etichettato con il nome di un'associazione (tra quelle che collegano le classi dei due oggetti)

A P S Oggetti di dominio

Oggetti di dominio per rappresentare uno studente Mario Rossi, nato a Roma, che frequenta i corsi di SIW e APS



A P S 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

