



PEGASO
Università Telematica



Indice

1. INTRODUZIONE.....	3
2. IL LINGUAGGIO UML	4
3. ASSOCIAZIONI	6
4. ALTRI COSTRUTTI.....	9
4.1 MOLTEPLICITÀ	9
4.2 ALTRI COSTRUTTI	10
BIBLIOGRAFIA	13

1. Introduzione

Questa unità didattica di apprendimento ha l'obiettivo di presentare un modello alternativo per rappresentare i dati e le loro relazioni, in pratica un'alternativa valida al modello Entità-Relazione.

Nella prima parte viene illustrato, per sommi capi, il linguaggio UML ed il diagramma delle classi, nato per la progettazione del software orientato agli oggetti.

Nella seconda parte si illustra il concetto di associazione, utilizzata come alternativa alla relazione, insieme alle cardinalità che la contraddistinguono.

Nell'ultima parte si illustrano altre corrispondenze come la generalizzazione e gli identificatori.

2. Il linguaggio UML

L'*Unified Modeling Language* è un linguaggio grafico per la modellazione di applicazioni software basate sulla programmazione orientata agli oggetti. Proposto a metà degli anni '90 per unificare alcuni formalismi preesistenti per la modellazione orientata agli oggetti, esso viene utilizzato nell'ingegneria del software e permette di rappresentare tutti gli aspetti di un'applicazione software.

La versione 2.0 è stata consolidata nel 2004 e ufficializzata da OMG nel 2005. UML 2.0 riorganizza molti degli elementi della versione precedente (1.5) in un quadro di riferimento ampliato e introduce molti nuovi strumenti, inclusi alcuni nuovi tipi di diagrammi. Sebbene OMG indichi UML 2.0 come la versione "corrente" del linguaggio, la transizione è di fatto ancora in corso; le stesse specifiche pubblicate da OMG sono ancora non completamente aggiornate e il supporto dei tool a UML 2.0 è, nella maggior parte dei casi, appena abbozzato. L'ultima versione è la 2.5, finalizzata nel 2013¹.

Questo linguaggio si è quindi universalmente imposto come linguaggio di riferimento per la modellazione di applicazioni software. L'UML risulta importante nel contesto dei database poiché esso viene utilizzato talvolta in alternativa al modello E-R con il **Diagramma delle classi**. Infatti, cambia la rappresentazione diagrammatica ma non l'approccio alla progettazione e per questo il diagramma delle classi viene utilizzato anche per rappresentare schemi concettuali.

L'UML racchiude un insieme di diagrammi, ciascuno dei quali modella uno o più aspetti di un'applicazione software orientata agli oggetti:

- Diagrammi UML
 - Diagramma delle classi
 - Diagramma degli oggetti
 - Diagramma dei casi d'uso
 - Diagramma di sequenza
 - Diagramma di collaborazione
 - Diagramma delle attività
 - Diagramma degli stati
 - Diagramma delle componenti
 - Diagramma di distribuzione

¹ https://it.wikipedia.org/wiki/Unified_Modeling_Language

In Figura 1 è illustrato un esempio di diagramma delle classi mentre in Figura 2 un esempio di diagramma dei casi d'uso, dove viene modellata l'interazione utente-sistema.

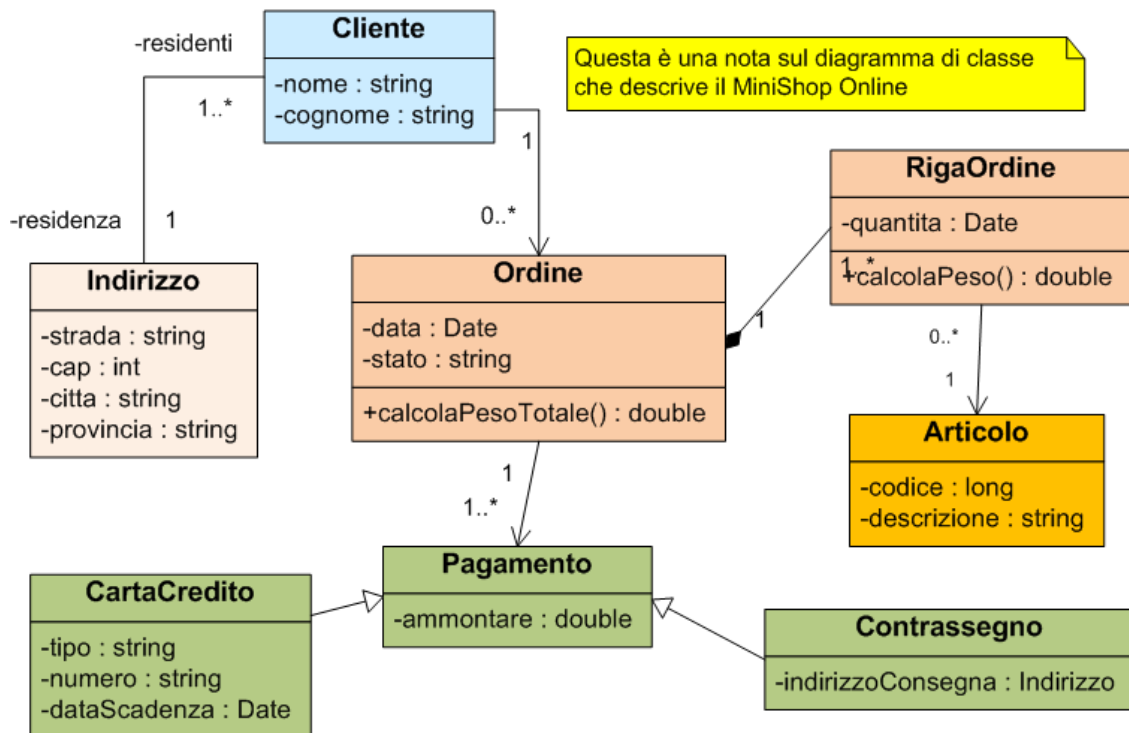


Figura 1: un esempio di diagramma delle classi per un contesto di vendita e pagamento.

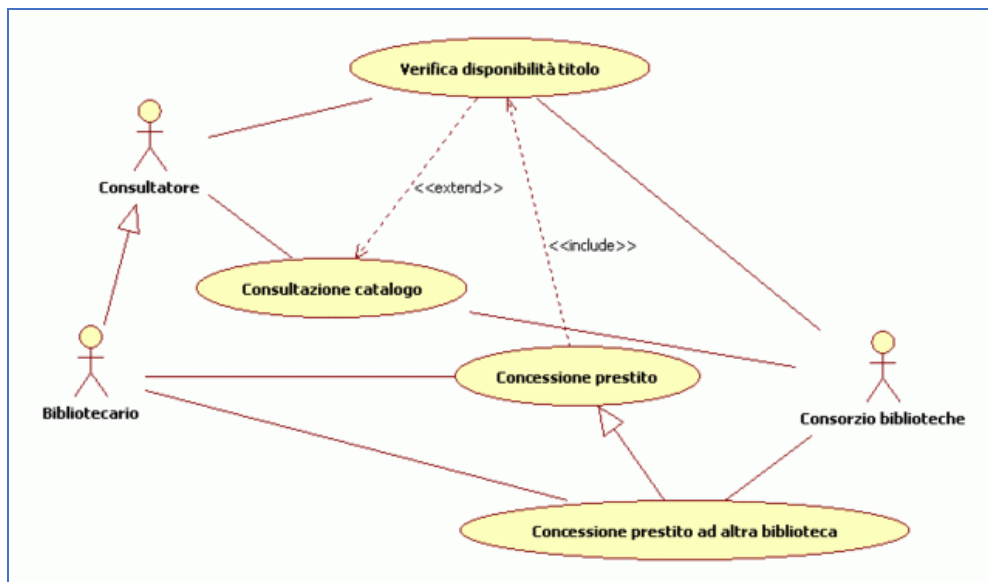


Figura 2: esempio di diagramma dei casi d'uso.

3. Associazioni

Come abbiamo visto precedentemente, il diagramma delle classi modella il dominio oggetto di studio dell'applicazione. È basato su versioni astratte degli oggetti: le classi. Una classe corrisponde ad una entità del modello E-R e viene rappresentata con un rettangolo con all'interno le seguenti informazioni:

- Nome
- Attributi
- Operazioni ammissibili sui dati

In Figura 3 è illustrata una rappresentazione di esempio di due classi: Classe impiegato e classe Progetto. Il rettangolo appare diviso appunto in tre parti. Gli attributi caratterizzano le classi.

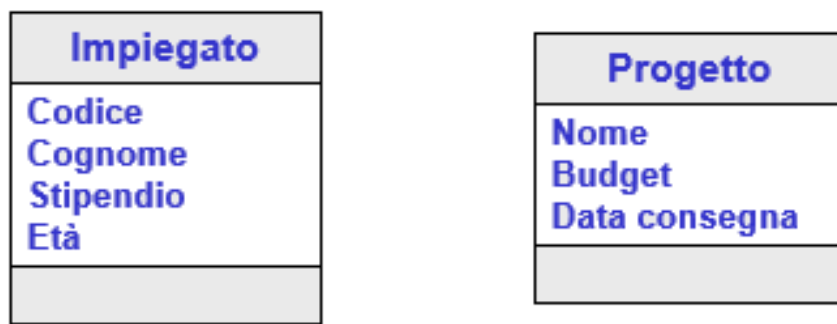


Figura 3: esempio di rappresentazione delle classi.

Nel diagramma delle classi, le associazioni corrispondono alle relazioni del modello E-R. Esse vengono rappresentate con linee dritte che congiungono le classi che partecipano alle associazioni.

Graficamente le associazioni vengono rappresentate come in Figura 4, dove sono rappresentate come in figura (relazioni binarie). Si possono rappresentare più relazioni tra le stesse classi come nel diagramma E-R.

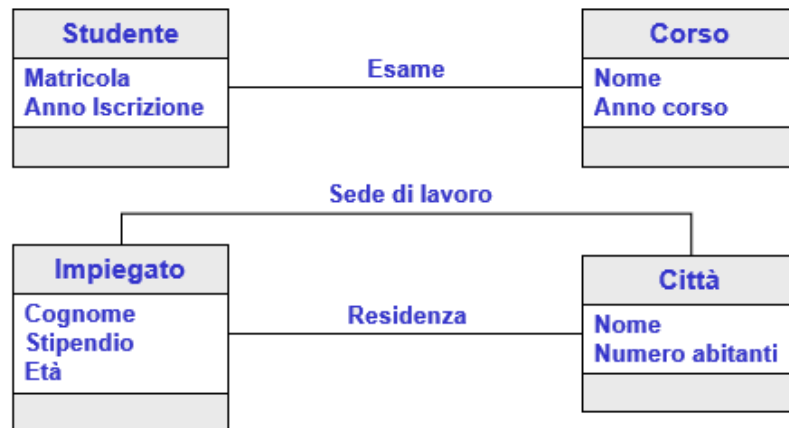


Figura 4: esempio di associazioni tra classi.

È possibile anche associare ruoli alle classi coinvolte in una associazione mentre NON è possibile assegnare attributi alle relazioni. Per questo si fa uso delle **Classi di associazione**. Esse descrivono le proprietà di un'associazione, e vengono graficamente rappresentate con una linea tratteggiata, come rappresentato dal cerchio rosso nella Figura 5.

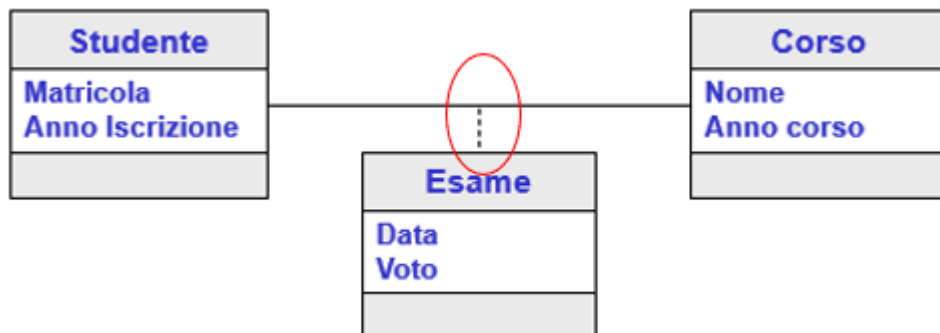


Figura 5: esempio di assegnazione di attributi attraverso le classi di associazione.

Per le associazioni n-arie si adotta la stessa notazione grafica del modello E-R, ovvero l'associazione rappresentata da un rombo, come rappresentato in Figura 6 dove sono indicate tre relazioni, Fornitore, Prodotto e Dipartimento, con l'attributo Quantità della relazione.

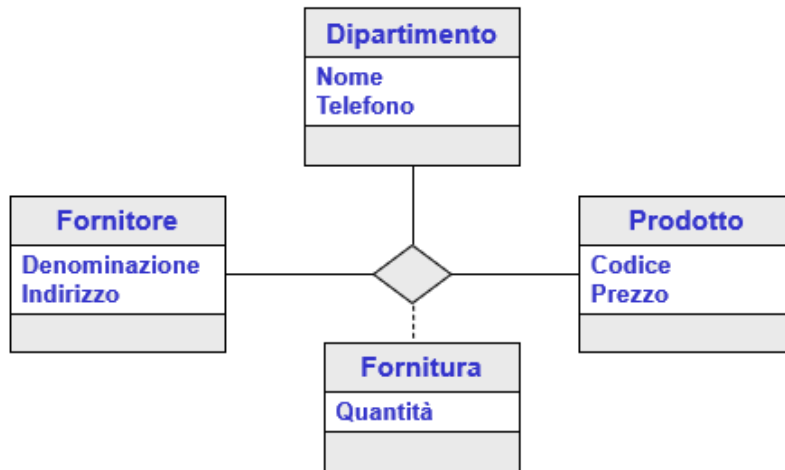


Figura 6: esempio di relazione ternaria.

Nel caso di associazioni ternarie, si usa la **Reificazione di associazione**: si trasforma cioè l'associazione in una classe legata alle classi originarie con associazioni binarie.

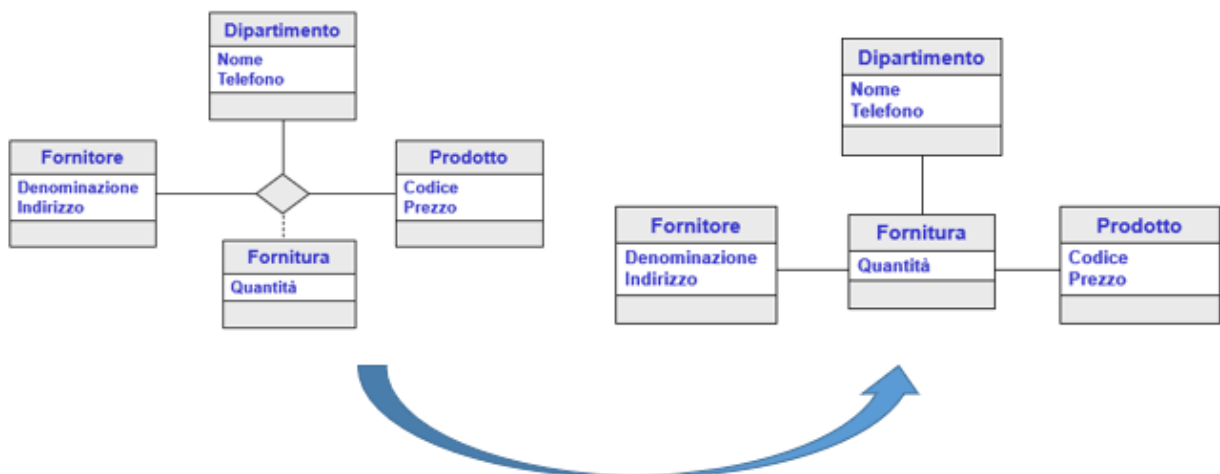


Figura 7: esempio di reificazione.

4. Altri costrutti

Altre proprietà dell'utilizzo del diagramma delle classi sono le seguenti:

- La Navigabilità.
- Le Associazioni come aggregazione di concetti, rappresentata da una linea avente un rombo attaccato alla classe.

In Figura 8 è sono illustrate le due tipologie di aggregazione di concetti con le relative spiegazioni.

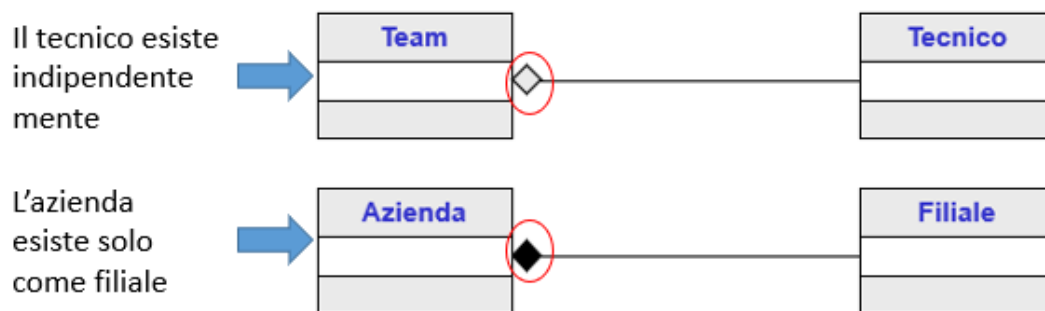


Figura 8: esempio di aggregazione di concetti.

4.1 Molteplicità

Si può anche indicare la cardinalità di partecipazione ma con un formalismo diverso, come indicato in Figura 9.



Figura 9: cardinalità delle relazioni.

Le differenze con il diagramma E.R sono le seguenti, come risulta dalla Figura 9. Lo schema di corrispondenza è il seguente:

- Cardinalità minima separata dalla massima con due punti
 - (0,1) -> 0..1
 - (1,1) -> 1
 - (0,N) -> *
 - (1,N) -> 1..N

Si presti attenzione al fatto che nel diagramma delle classi, le cardinalità sono invertite rispetto al classico diagramma E-R.

4.2 Altri costrutti

Come altri costrutti, abbiamo gli **Identificatori interni**:

- Non esiste in UML una notazione per esprimere identificatori di classi
- Si usano i vincoli utente

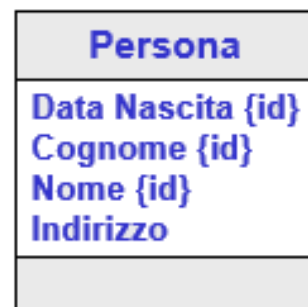
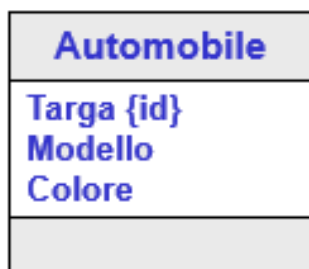


Figura 10: identificatori delle classi.

In pratica si mette il costrutto **{id}** accanto all'attributo identificatore.

Per gli Identificatori esterni, si ricorre agli stereotipi, ovvero un nome racchiuso tra i simboli **<<** e **>>**.

In Figura 11 un esempio.

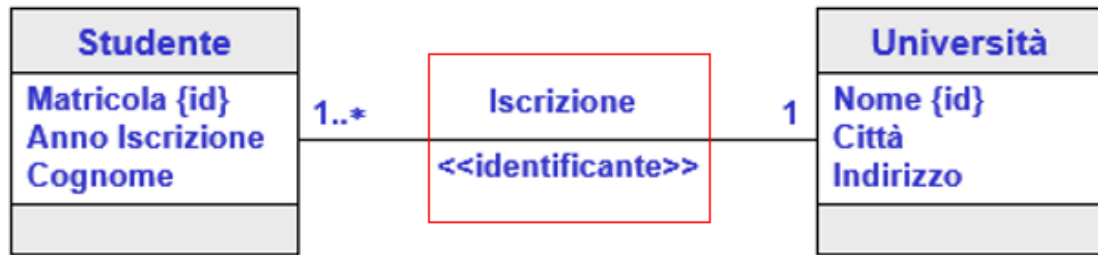


Figura 11: identificatore esterno.

Si possono utilizzare anche le **Generalizzazioni**. Infatti, esiste la possibilità di definire generalizzazioni come per il diagramma E-R. Si possono usare gli stessi costrutti.

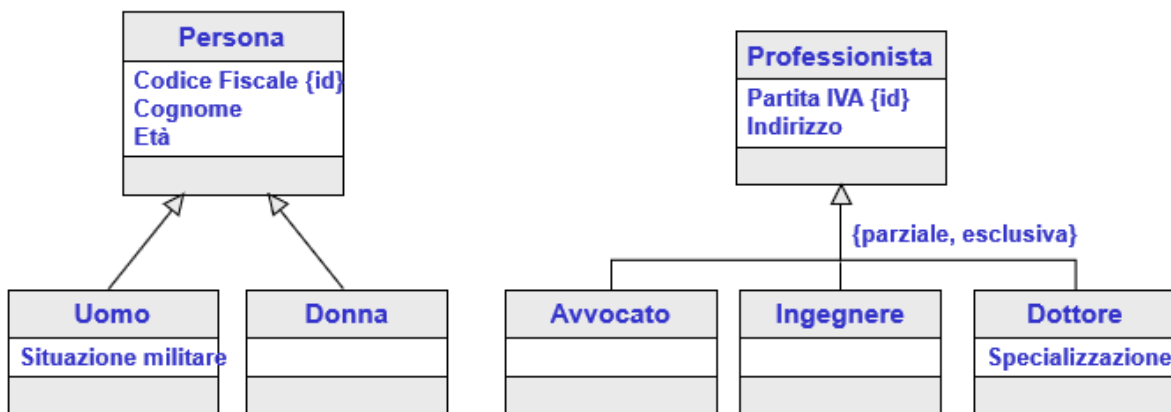


Figura 12: esempi di generalizzazioni.

Infine, per completezza, in Figura 13 l'esempio gi  studiato della modellazione della realt  aziendale.

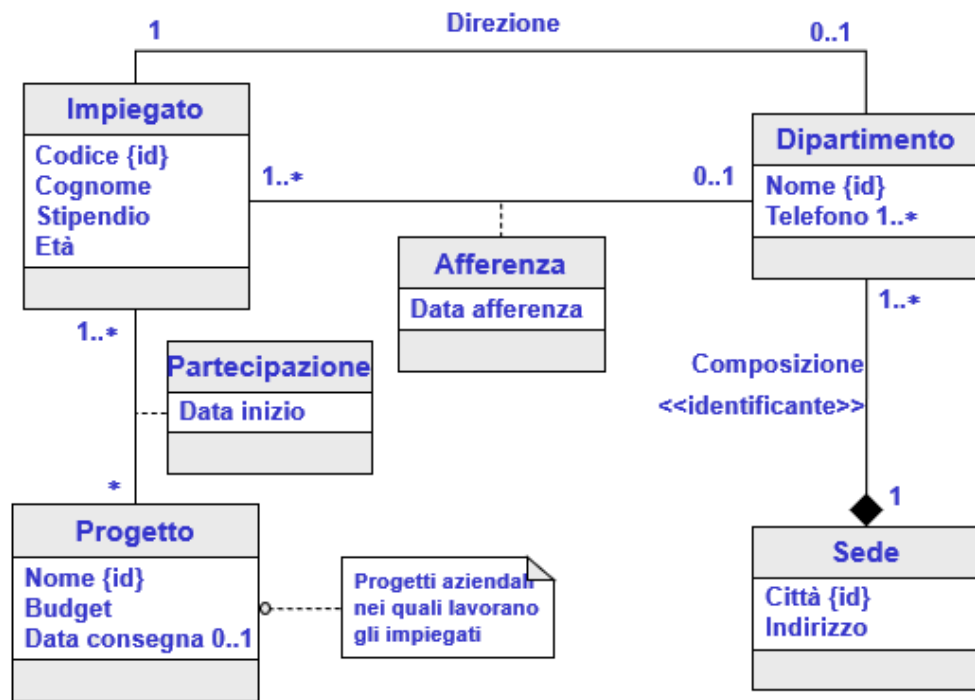


Figura 13: un esempio di diagramma delle classi completo.

Si invita lo studente a confrontare questo schema con quello analogo E-R.

Bibliografia

- Atzeni P., Ceri S., Fraternali P., Paraboschi S., Torlone R. (2018). Basi di Dati. McGraw-Hill Education.
- Batini C., Lenzerini M. (1988). Basi di Dati. In Cioffi G. and Falzone V. (Eds). Calderini. Seconda Edizione.