

# Elaborazione + Modello dominio

Friday, 31 March 2023

14:08

- Minimo 2 iterazioni deve durare

Si ha bisogno di una buona pianificazione a seconda di:

- Rischio, aka complessità tecnica
- Copertura, tutte le parti del sistema
- Criticità, ciò che il cliente vuole

Questi vengono ordinati attraverso es votazione e poi si iniziano con i scenari di voto maggiore.

Voto	Requisito (Caso d'uso o Caratteristica)	Commento
Alto	Elabora Vendita Logging	Ottiene voti alti per tutti i criteri. Pervasivo. Difficile da aggiungere in un secondo momento.
...	...	...
Medio	Gestire utenti	Influisce sul sottodominio di sicurezza.

Questa classifica viene aggiornata ad ogni iterazione

- Scoperto maggior parte dei requisiti ed altri nel dettaglio
  - Stimare il piano di lavoro e risorse complessive più precisamente
  - Se viene eseguito tutto prima abbiamo sbagliato qualcosa
- Fatti prima i rischi maggiori -> Nucleo
  - Si inizia presto a programmare così abbiamo i primi feedback
  - Testing siccome il codice andrà in produzione
  - Deve per forza produrre qualcosa l'elaborazione

Però non bisogna progettare tutto
- Vengono iniziati vari modelli per un'analisi ad oggetti:
  - Modello di progetto
    - Diagrammi con progettazione logica, con diagrammi software
  - Documento architettura software
  - Modello di dati
    - Mappare oggetti e basi di dati
  - Storyboard dei casi d'uso ed UI
    - Interfacce utenti
  - Modello dominio
    - Visualizzazione concetti di dominio
    - Descrive un punto di vista concettuale gli oggetti del mondo reale
      - Senza attributi, ruolo puramente comportamentale
      - Non è un modello dati, che mostra i dati che devono essere memorizzati
    - Descrive classi concettuali e le relazioni

→ Si usa il testo dei casi d'uso

→ E' un dizionario visuale

Aka rappresentazione visuale di classi concettuali/mondo reale

In un dominio di interesse

E mostrano classi/oggetti, associazioni ed attributi

- Ispirazione per la classe software

- ☐ **NON** è una raffigurazione di oggetti software

- ☐ Sono astrazione di oggetti reali

Quindi usare come nomi oggetti reali e non software

Es. SalesDatabase oppure mettere una funzione nella classe

- Sviluppato iterativamente ed incrementalmente

- Crearlo per:

- ☐ Comprendere dominio del sistema ed il vocabolario

- ☐ Ispirazione per lo strato di dominio

E per crearlo:

- ☐ Trovare classi concettuali

- ◆ Riusare/modificare modelli esistenti

- ◆ Elenco categorie comuni

Dobbiamo inserire gli esempi da queste categorie

Categoria di classe concettuale	Esempi
<b>transazioni commerciali</b> <i>Linea guida:</i> sono aspetti critici (riguardano denaro), dunque si inizi con le transazioni.	<i>Sale, Payment (o CashPayment)</i> <i>Reservation</i>
<b>elementi/righe di transazioni</b> <i>Linea guida:</i> le transazioni spesso sono composte da righe per gli articoli correlati, quindi queste vanno considerate subito dopo le transazioni.	<i>SalesLineItem</i>
<b>prodotto o servizio correlato a una transazione o a una riga di transazione per articolo</b> <i>Linea guida:</i> le transazioni sono per qualcosa (un prodotto o un servizio). Vanno considerate subito dopo.	<i>Item</i> <i>Flight, Seat, Meal</i>
<b>dove viene registrata la transazione?</b> <i>Linea guida:</i> importante.	<i>Register, Ledger (o SalesLedger), FlightManifest</i>
<b>ruoli di persone o organizzazioni correlati alle transazioni; attori nei casi d'uso</b> <i>Linea guida:</i> normalmente dobbiamo sapere quali sono le parti coinvolte in una transazione.	<i>Cashier, Customer, Store</i> <i>MonopolyPlayer Passenger, Airline</i>
<b>luogo della transazione; luogo del servizio</b>	<i>Store Airport, Plane, Seat</i>
<b>eventi significativi, spesso con un'ora o un luogo che è necessario ricordare</b>	<i>Sale, Payment (o CashPayment)</i> <i>MonopolyGame Flight</i>
<b>oggetti fisici</b> <i>Linea guida:</i> questo è particolarmente importante quando si crea software per il controllo di dispositivi, oppure simulazioni.	<i>Item, Register Board, Piece, Die</i> <i>Airplane</i>
<b>descrizioni di oggetti</b> <i>Linea guida:</i> vedere il Paragrafo 12.13 per una discussione.	<i>ProductDescription</i> <i>FlightDescription</i>
<b>cataloghi</b> <i>Linea guida:</i> le descrizioni sono spesso contenute in un catalogo.	<i>ProductCatalog FlightCatalog</i>
<b>contenitori di oggetti (fisici o informazioni)</b>	<i>Store, Bin Board Airplane</i>
<b>oggetti in un contenitore</b>	<i>Item Square (in un Board)</i> <i>Passenger</i>
<b>altri sistemi che collaborano</b>	<i>CreditAuthorizationSystem</i> <i>AirTrafficControl</i>
<b>registrazioni di questioni finanziarie, di lavoro, contrattuali e legali</b>	<i>Receipt, Ledger MaintenanceLog</i>
<b>strumenti finanziari</b>	<i>Cash, Check, LineOfCredit</i> <i>TicketCredit</i>
<b>piani, manuali, documenti cui si fa regolarmente riferimento per eseguire il lavoro</b>	<i>DailyPriceChangeList</i> <i>RepairSchedule</i>

- ◆ Identificare nomi/locuzioni nominali

Analizziamo gli elaborati testuali (es casi d'uso) e teniamo traccia

Di tutti i nomi e locuzioni nominali (Cosa sono le locuzioni nominali)

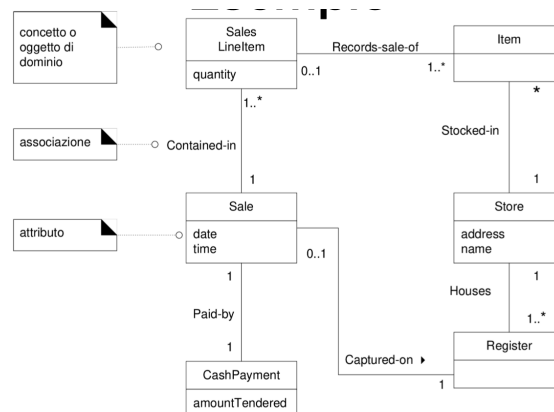
E dopo le riportiamo, a sinistra classe destra nome trovato

<i>Sale</i>	<i>(Vendita)</i>
<i>CashPayment</i>	<i>(Pagamento in contanti)</i>
<i>SalesLineItem</i>	<i>(Riga di vendita per l'articolo)</i>

E dopo possiamo rappresentarlo in uml la prima colonna

- ☐ Disegnare come classi non diagramma delle classi UML
- ☐ Aggiungere associazioni ed attributi
- ☐ Quando si hanno dei dubbi, chiediamoci:
  - ◆ Ci potrebbe servire la classe [CLASSE]?
  - ◆ La possiamo rappresentare?
- Classe concettuale è un'idea, una cosa o un oggetto con:
  - ☐ Simbolo, parola/immagina per rappresentare classe concettuale
    - ◆ Usare nomi esistenti nel territorio/dizionario/semplifici
    - ◆ Non aggiungere cose che non ci sono
    - ◆ No irrilevanti
  - ☐ Intenzione, definizione della classe concettuale
  - ☐ Estensione, insieme degli oggetti descritti

Le istanze quindi sono a livello estensionale
- Rappresentato con il modello delle classi in uml



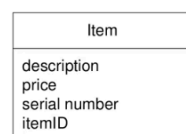
- ☐ Classe: un descrittore per un insieme di oggetti che possiedono le stesse caratteristiche (attributi, operazioni, metodi, relazioni)

Quindi le classi concettuali sono un sottoinsieme delle classi UML

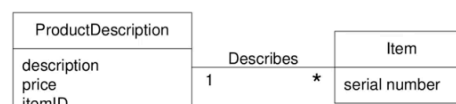
- ◆ Una classe particolare è la classe descrizione che contiene informazioni che descrivono qualcos'altro

Es. Item: un oggetto fisico del negozio - potrebbe avere una serie di attributi

Ed item estende lavastoglie, aspirapolvere, e quindi evitiamo la ridondanza siccome a tutti gli item serve una serie



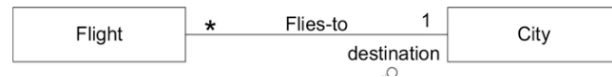
Sconsigliato



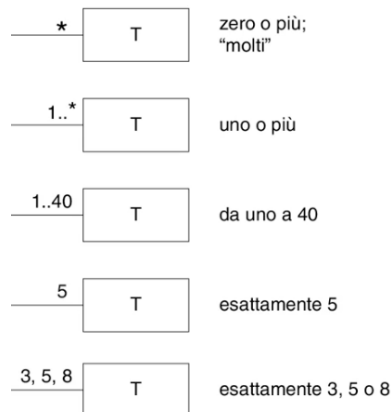
Migliore

- Associazione: Relazione tra classi che indica una connessione significativa
  - ◆ Mostrare quelle che implicano conoscenze che deve essere memorizzata per tempo
  - ◆ Ogni associazione è fatta da:
    - ◇ Nome associazione
    - ◇ Molteplicità

Che sono dentro ad ogni ruolo (ogni ruolo può avere un



E ogni molteplicità rappresenta quante istanze abbiamo

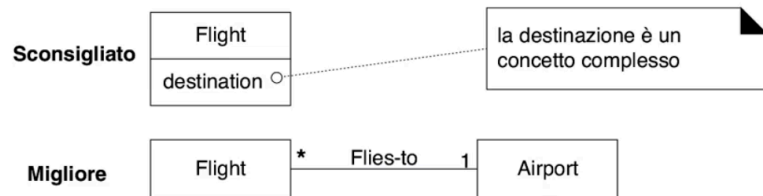


- ◆ Esistono associazioni riflesse ed associazioni multiple
- ◆ Associazioni identificate attraverso associazioni comuni

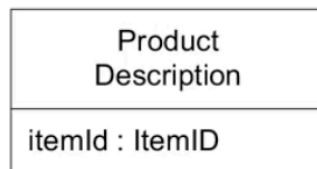
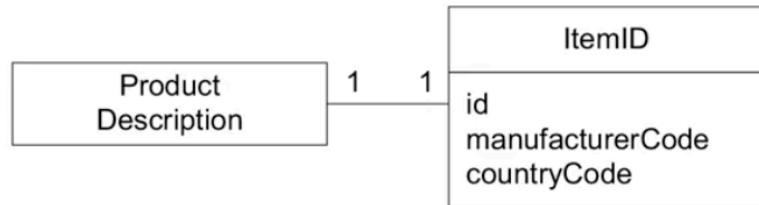
Categoria	Esempi
A è una transazione correlata a un'altra transazione B	CashPayment—Sale Cancellation—Reservation
A è un elemento/riga di una transazione B	SalesLineItem—Sale
A è un prodotto o servizio per una transazione (o riga per l'articolo) B	Item—SalesLineItem (o Sale) Flight—Reservation
A è un ruolo relativo a una transazione B	Customer—Payment Passenger—Ticket
A è una parte fisica o logica di B	Drawer—Register Square—Board Seat—Airplane
A è contenuto fisicamente o logicamente in B	Register—Store, Item—Shelf Square—Board Passenger—Airplane
A è una descrizione per B	ProductDescription—Item FlightDescription—Flight
A è un membro di B	Cashier—Store Player—MonopolyGame Pilot—Airline
A è una sottounità organizzativa di B	Department—Store Maintenance—Airline
A utilizza o gestisce o possiede B	Cashier—Register Player—Piece Pilot—Airplane
A è vicino/prossimo a B	SalesLineItem—SalesLineItem Square—Square City—City

- ◆ Evitare troppe associazioni
- Attributo:
  - ◆ Proprietà elementare di un oggetto di una classe
  - ◆ Ogni attributo deve avere un valore
  - ◆ Rappresenta le informazioni
  - ◆ Devono essere semplici, elementari con dati del dominio
    - ◇ In caso di dubbio, definire classe concettuale con associati
  - ◆ Non corrispondere ad un concetto complesso

- ▼ non corrispondere ad un concetto complesso



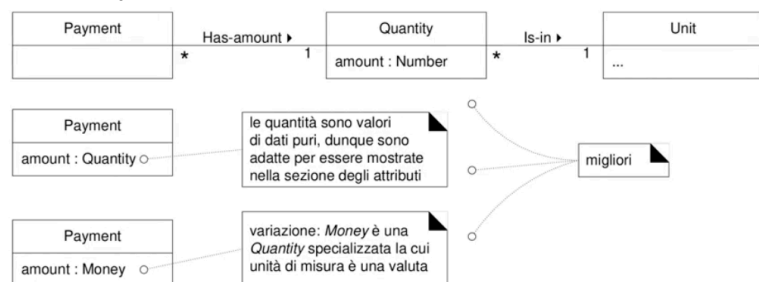
- ◆ In caso di relazioni 1-1, si possono creare le classi dato  
Che sono classi con solamente dati senza funzioni  
E rendere il tipo della variabile il nome della classe



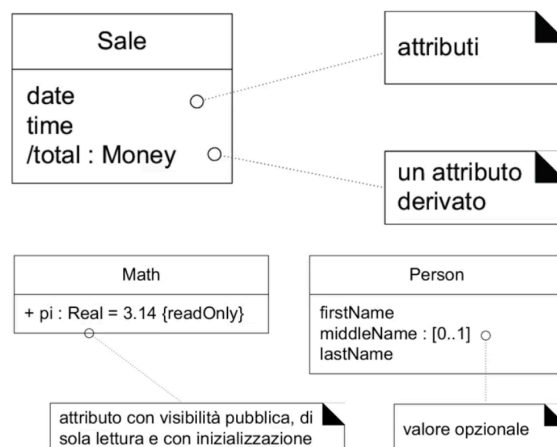
Però **non** possiamo creare un tipo semplice come puntatore

- ◆ Ricordare di specificare sempre, es:  
Payment ha un attributo amount di tipo Number
  - ◇ Che vuol dire number?
  - ◇ E' in euro? Dollari? E' la quantità?

Usare questi



- ◆ Attributi derivati hanno il / (Derivati dalle relazioni, non diretti)



Quest'ultima immagine è un uml però non modello dominio

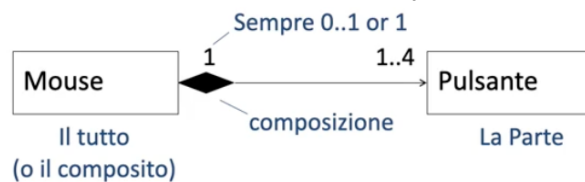
□ Aggregazione:

- ◆ E' una associazione aggrego parte
  - ◆ Debolmente collegati, uno può esistere senza l'altro
- Un computer può avere una stampante, una stampante è indipendente  
Però la stampante è incompleta senza computer



□ Composizione

- ◆ Aggregazione più forte
- ◆ Una parte appartiene ad 1 composto alla volta
- ◆ Non esiste X senza Y → Dipendenza



Da questo si crea il diagramma degli oggetti di dominio

