L'identificazione di un ricco insieme di classi concettuali è al centro dell'analisi OO. Se viene effettuata con perizia e con un *breve* investimento di tempo (ovvero, non più di qualche ora in ciascuna delle iterazioni iniziali), solitamente ripaga durante la progettazione, favorendo una migliore comprensione e comunicazione.

Linea guida

Si eviti un grosso sforzo di modellazione secondo l'approccio a cascata per creare un modello di dominio completo o "corretto". Infatti, non sarà mai né completo né corretto, e questi sforzi di modellazione eccessivi portano alla cosiddetta *paralisi da analisi*, con poco o nessun ritorno sull'investimento fatto.

9.2 Che cos'è un modello di dominio

Il passo essenziale dell'analisi orientata agli oggetti è la decomposizione di un dominio in concetti o oggetti significativi.

Un **modello di dominio** è una rappresentazione *visuale* di classi concettuali o di oggetti del mondo reale di un dominio [MO95, Fowler96]. I modelli di dominio sono stati chiamati anche **modelli concettuali, modelli degli oggetti di dominio** e **modelli degli oggetti di analisi.**[2]

Definizione

In UP, il termine "Modello di Dominio" indica una rappresentazione di classi concettuali del mondo reale, non di oggetti software. Il termine *non* indica un insieme di diagrammi che descrivono classi software, lo strato del dominio di un'architettura software o oggetti software con responsabilità.

UP definisce il Modello di Dominio[3] come uno degli elaborati che si possono creare nella disciplina di Modellazione del business. Più precisamente, il Modello di Dominio di UP è una specializzazione del **Modello degli Oggetti di Business** (BOM) di UP che "è incentrato sulla spiegazione di "cose" e prodotti importanti per un dominio di business" [RUP]. Ciò significa che un Modello di Dominio è incentrato su un solo dominio, come per esempio tutto ciò che riguarda il POS. Il BOM più ampio, che non verrà trattato in questo testo introduttivo e che si sconsiglia di creare (poiché può portare a un'eccessiva modellazione preliminare), è un modello multi-dominio esteso, spesso molto ampio e difficile da creare, che riguarda l'intero business e tutti i relativi sottodomini.

---

2. I modelli di dominio sono anche correlati ai modelli concettuali Entità-Relazione, che consentono di mostrare viste puramente concettuali dei domini, ma che sono stati considerati come modelli di dati per la progettazione di basi di dati. I modelli di dominio non sono modelli di dati.

3. Le iniziali maiuscole di "Modello di Dominio" o di altri termini sono utilizzate per sottolineare il fatto che si tratta di un nome di modello ufficiale definito in UP, rispetto al ben noto e più generale concetto di "modello di dominio".

Applicando la notazione UML, un modello di dominio è illustrato con un insieme di **diagrammi delle classi** in cui non sono definite operazioni (firme di metodi). Esso fornisce un *punto di vista concettuale* e può mostrare:

- oggetti di dominio o classi concettuali
- associazioni tra classi concettuali
- attributi di classi concettuali

Definizione: il Modello di Dominio è un "dizionario visuale"

Si rifletta per un istante sulla Figura 9.2 e si noti come sono visualizzate e correlate le parole o i concetti del dominio. La figura mostra inoltre un'*astrazione* delle classi concettuali, poiché ci sono molte altre informazioni che si potrebbero comunicare in merito ai registratori di cassa, alle vendite e così via.
Le informazioni che la figura rappresenta (utilizzando la notazione UML) potevano essere espresse, in alternativa, come del semplice testo (nel Glossario di UP). L'uso di un linguaggio visuale permette tuttavia di capire più facilmente i termini e soprattutto le relazioni tra di essi, poiché il cervello umano è bravo a comprendere elementi mostrati graficamente e linee di connessione.
Pertanto il modello di dominio è un *dizionario visuale* delle astrazioni significative, della terminologia del dominio e del contenuto informativo del dominio.

Definizione: il Modello di Dominio non è una raffigurazione degli oggetti software aziendali

Un Modello di Dominio di UP, come mostrato nella Figura 9.3, è una visualizzazione di oggetti del mondo reale in un dominio di interesse, *non* di oggetti software (come classi Java o C#) oppure di oggetti software con responsabilità (Figura 9.4).
Pertanto i seguenti elementi non sono adatti a essere mostrati in un modello di dominio.

- Elementi software, come una finestra o una base di dati, a meno che il dominio che si sta modellando non riguardi concetti software, come un modello per interfacce grafiche utente.
- Responsabilità o metodi.[4]

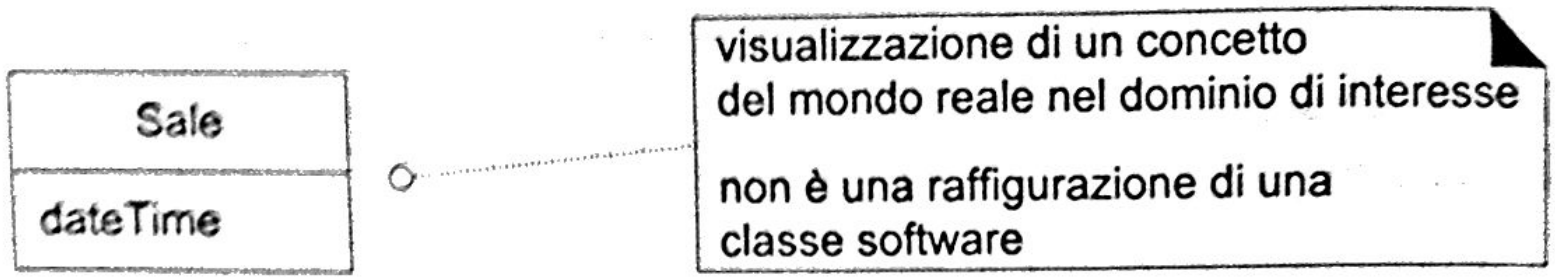


Figura 9.3 Un modello di dominio mostra classi concettuali del mondo reale, non classi software.

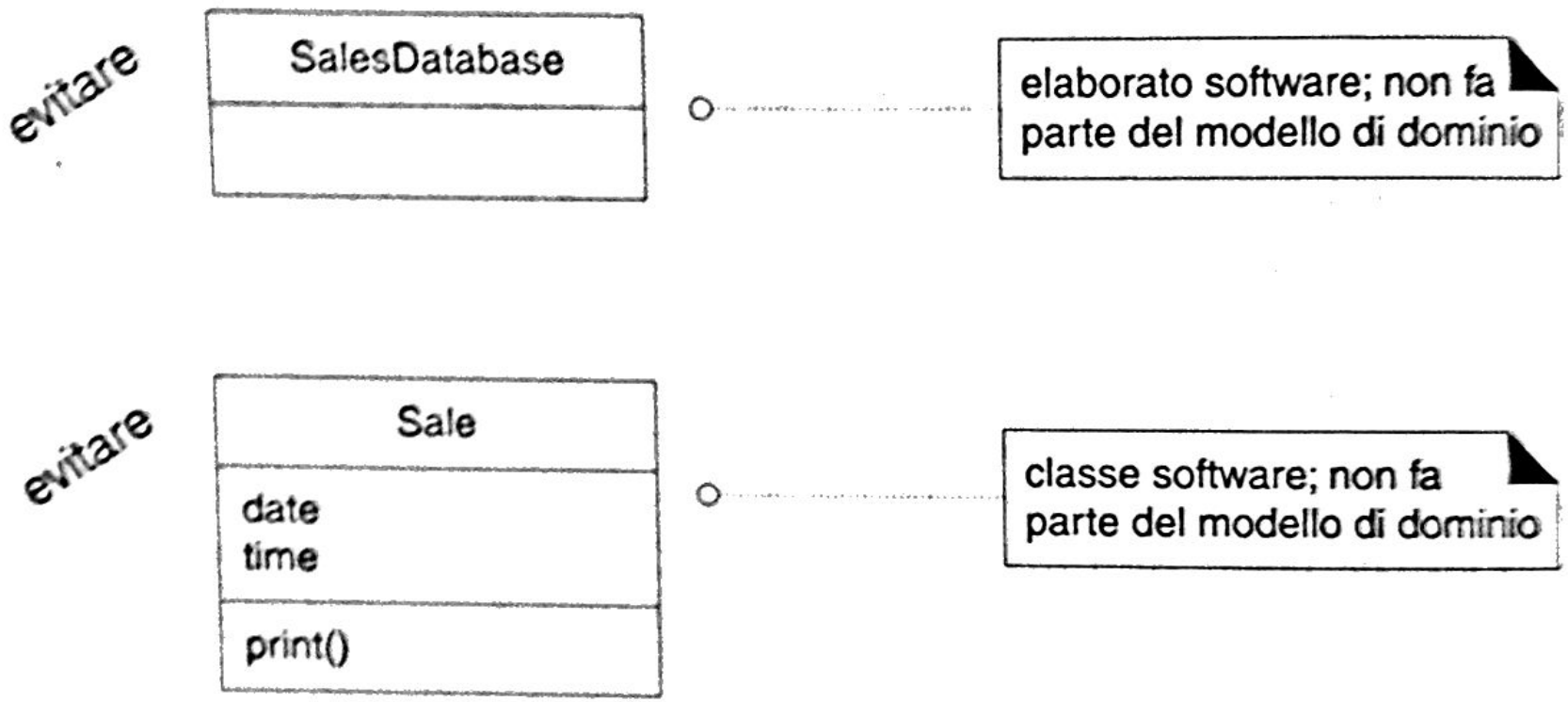


Figura 9.4 Un modello di dominio non mostra elementi o classi software.

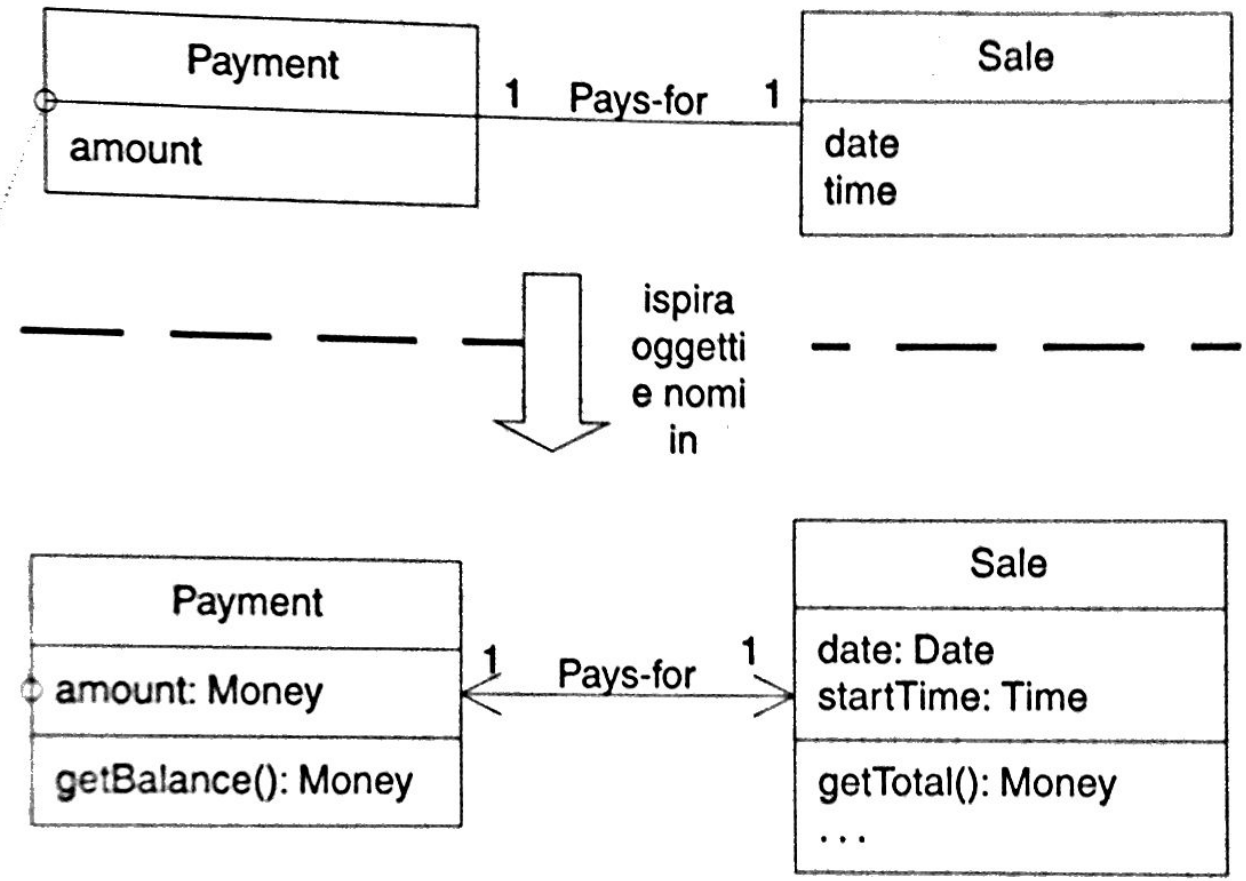
Payment nel Modello di Dominio è un concetto, ma Payment nel Modello di Progetto è una classe software. Non sono la stessa cosa, ma il primo ha ispirato il nome e la definizione del secondo.

In tal modo si riduce il salto rappresentazionale.

Questa è una delle grandi idee della tecnologia a oggetti.

Modello di Dominio di UP

Vista delle parti interessate dei concetti significativi del dominio.



Modello di Progetto di UP

Lo sviluppatore orientato agli oggetti ha tratto ispirazione dal dominio del mondo reale per la creazione di classi software.

Pertanto, il salto rappresentazionale tra il modo in cui le parti interessate concepiscono il dominio e la sua rappresentazione nel software è diminuito.

Figura 9.6 Salto rappresentazionale basso con la modellazione OO.

Tabella 9.1 Elenco di categorie di classi concettuali.

Categoria di classe concettuale	Esempi
transazioni commerciali *Linea guida*: sono aspetti critici (riguardano denaro), dunque si inizi con le transazioni.	*Sale, Payment, Reservation*
elementi/righe di transazioni *Linea guida*: le transazioni spesso sono composte da righe per gli articoli correlati, quindi queste vanno considerate subito dopo le transazioni.	*SalesLineItem*
prodotto o servizio correlato a una transazione o a una riga di transazione per articolo *Linea guida*: le transazioni sono *per* qualcosa (un prodotto o un servizio). Vanno considerate subito dopo.	*Item* *Flight, Seat, Meal*
dove viene registrata la transazione? *Linea guida*: importante.	*Register, Ledger* *FlightManifest*
ruoli di persone o organizzazioni correlati alle transazioni; attori nei casi d'uso *Linea guida*: normalmente dobbiamo sapere quali sono le parti coinvolte in una transazione.	*Cashier, Customer, Store* *MonopolyPlayer* *Passenger, Airline*
luogo della transazione; luogo del servizio	*Store* *Airport, Plane, Seat*
eventi significativi, spesso con un'ora o un luogo che è necessario ricordare	*Sale, Payment* *MonopolyGame* *Flight*
oggetti fisici *Linea guida*: questo è particolarmente importante quando si crea software per il controllo di dispositivi, oppure simulazioni.	*Item, Register* *Board, Piece, Die* *Airplane*
descrizioni di oggetti *Linea guida*: vedere il Paragrafo 9.13 per una discussione.	*ProductDescription* *FlightDescription*
cataloghi *Linea guida*: le descrizioni sono spesso contenute in un catalogo.	*ProductCatalog* *FlightCatalog*
contenitori di oggetti (fisici o informazioni)	*Store, Bin* *Board* *Airplane*
oggetti in un contenitore	*Item* *Square (in una Board)* *Passenger*
altri sistemi che collaborano	*CreditAuthorizationSystem* *AirTrafficControl*
registrazioni di questioni finanziarie, di lavoro, contrattuali e legali	*Receipt, Ledger* *MaintenanceLog*
strumenti finanziari	*Cash, Check, LineOfCredit* *TicketCredit*
piani, manuali, documenti cui si fa regolarmente riferimento per eseguire il lavoro	*DailyPriceChangeList* *RepairSchedule*

Linea guida

Occorre prestare particolare attenzione con questo metodo; una corrispondenza meccanica da nome a classe non è possibile, poiché le parole nel linguaggio naturale sono ambigue.

Nondimeno, l'analisi linguistica è un'altra fonte di ispirazione. I casi d'uso in formato dettagliato sono un'ottima descrizione a cui ispirarsi per questa analisi. Per esempio, si può utilizzare lo scenario corrente del caso d'uso *Process Sale*.

Scenario principale di successo (o Flusso di base):

1. Il **Cliente** arriva alla **cassa POS** con gli **articoli** e/o i **servizi** da acquistare.
2. Il **Cassiere** inizia una nuova **vendita**.
3. Il **Cassiere** inserisce il **codice identificativo dell'articolo**.
4. Il Sistema registra la **riga di vendita per l'articolo** e mostra la **descrizione dell'articolo**, il suo **prezzo**, il **totale** parziale. Il prezzo è calcolato in base a un insieme di regole di prezzo.

Il Cassiere ripete i passi 2-3 fino a che non indica che ha terminato.

5. Il Sistema mostra il totale con le **imposte** calcolate.
6. Il Cassiere riferisce il totale al Cliente e richiede il **pagamento**.
7. Il Cliente paga e il Sistema gestisce il pagamento.
8. Il Sistema registra la vendita completata e invia informazioni sulla **vendita** e sul pagamento ai sistemi esterni di **Contabilità** (per la contabilità e le **commissioni**) e di **Inventario** (per l'aggiornamento dell'inventario).
9. Il Sistema genera la **ricevuta**.
10. Il Cliente va via con la ricevuta e gli articoli acquistati.

Estensioni (o Flussi alternativi):

...

7a. Pagamento in contanti:

1. Il Cassiere inserisce l'**importo in contanti** presentato dal Cliente.
2. Il Sistema mostra il **resto dovuto** e apre il cassetto del registratore di cassa.
3. Il Cassiere deposita il contante presentato e restituisce il resto in contanti al Cliente.
4. Il Sistema registra il pagamento in contanti.

9.10 Linea guida: pensare come un cartografo; utilizzare i termini del dominio

La strategia del cartografo può essere applicata sia alle mappe che ai modelli di dominio.

Linea guida

Creare un modello di dominio usando lo spirito con cui lavora un cartografo.

- Utilizzare i nomi esistenti sul territorio. Per esempio, se si sviluppa un modello per le prenotazioni aeree, si assegni al cliente il nome "Passeggero", termine utilizzato dal personale delle compagnie aeree, anziché il troppo generico "Cliente".
- Escludere caratteristiche irrilevanti o fuori dalla portata. Per esempio, nel modello di dominio del Monopoly per l'iterazione 1 non vengono utilizzate le carte (come la carta "Esci gratis di prigione"), quindi non è opportuno mostrare una classe *Card* nel modello per questa iterazione.
- Non aggiungere cose che non ci sono.

Il principio è simile alla strategia *Use the Domain Vocabulary* [Coad95].

9.11 Linea guida: modellare il mondo *non reale*

Alcuni sistemi software riguardano domini che hanno ben poche analogie con i domini naturali o aziendali; un esempio è il software per le telecomunicazioni. Tuttavia è possibile creare un modello di dominio anche per questi domini. È necessario un grado elevato di astrazione, evitando di usare tecniche di progettazione non orientate agli oggetti e prestando attenzione alla terminologia principale e ai concetti che utilizzano gli esperti del dominio.

Per esempio, ecco alcune classi concettuali candidate per il dominio di un commutatore per telecomunicazioni: *Message, Connection, Port, Dialog, Route* e *Protocol.*

Linea guida: quando sono utili le classi descrizione

Linea guida

Aggiungere una classe descrizione (per esempio, *ProductDescription*) nei seguenti casi.

- Quando è necessaria una descrizione di un articolo o servizio, indipendentemente dall'attuale esistenza di istanze di tali articoli o servizi.
- Quando l'eliminazione delle istanze degli oggetti che descrivono (per esempio, *Item*) darebbe luogo a una perdita di informazioni che invece è necessario conservare, ma che sono state erroneamente associate all'oggetto eliminato.
- Quando si vogliono ridurre le informazioni ridondanti o ripetute.

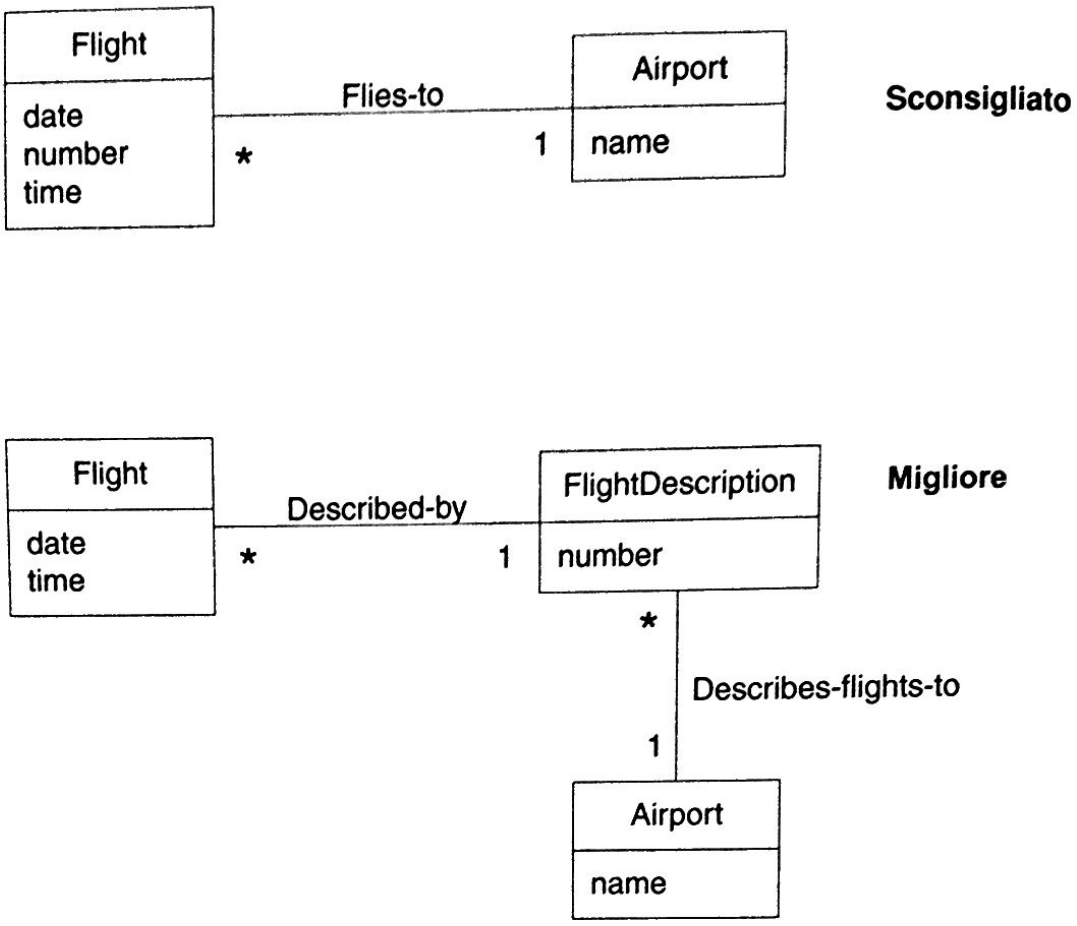


Figura 9.10 Descrizioni di altri oggetti.

9.14 Associazioni

È utile trovare e mostrare associazioni necessarie per soddisfare i requisiti informativi degli scenari correnti in corso di sviluppo, nonché quelle che contribuiscono alla comprensione del dominio.

Un'**associazione** è una relazione tra classi (più precisamente, tra istanze di queste classi) che indica una connessione significativa e interessante (Figura 9.11).

In UML, un'associazione è definita come "la relazione semantica tra due o più classificatori che comporta connessioni tra le rispettive istanze".

Linea guida: quando mostrare un'associazione

Le associazioni che è utile mostrare sono solitamente quelle che implicano la conoscenza di una relazione che deve essere memorizzata per una certa durata di tempo, che a seconda del contesto potrebbe essere di millisecondi o di anni. In altre parole, *tra quali oggetti è necessaria una certa* ***memoria*** *di una relazione*?

Per esempio, dobbiamo *ricordare* quali istanze di *SalesLineItem* sono associate a un'istanza di *Sale*? Certamente sì, altrimenti non sarebbe possibile ricostruire una vendita, stampare una ricevuta o calcolare il totale di una vendita.

E dobbiamo ricordare le *Sale* completate in un *Ledger*, ai fini legali e di contabilità.

Linea guida

Considerare l'inclusione delle seguenti associazioni in un modello di dominio.

- Associazioni per cui la conoscenza della relazione deve essere conservata per una qualche durata (associazioni "da ricordare").
- Associazioni derivate dall'elenco di associazioni comuni.

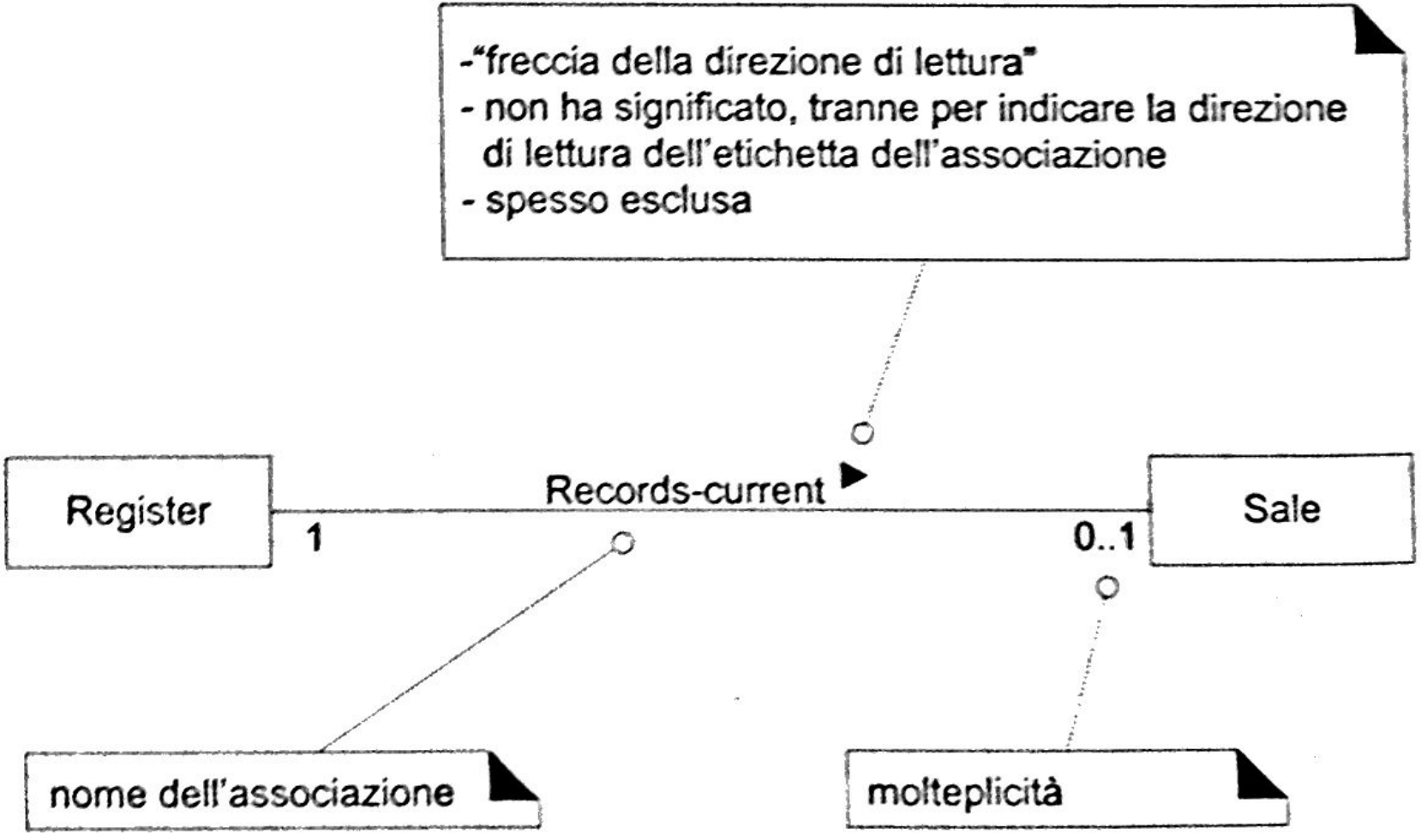


Figura 9.12 Notazione UML per le associazioni.

Attenzione

La freccia della direzione di lettura non ha significato in termini di modello; è solo un aiuto per il lettore del diagramma.

Linea guida: come assegnare il nome a un'associazione in UML

Linea guida

Assegnare il nome a un'associazione in base al formato *NomeClasse-LocuzioneVerbale-NomeClasse*, in cui la locuzione verbale (ovvero, una frase formata da un verbo principale, più eventuali complemento, oggetto e avverbi) crea una sequenza leggibile e significativa.

Nomi di associazioni semplici come "Ha" oppure "Usa" vanno se possibile evitati, poiché raramente migliorano la comprensione del dominio.
Per esempio:

- *Sale Paid-by CashPayment*
  - un cattivo esempio (perché non migliora la comprensione) è: *Sale Uses CashPayment*
- *Player Is-on Square*
  - un cattivo esempio è: *Player Has Square*

I nomi delle associazioni devono iniziare con una lettera maiuscola, poiché un'associazione rappresenta un classificatore di collegamenti tra istanze; in UML, i nomi dei classificatori devono iniziare con una lettera maiuscola. Due formati comuni e ugualmente validi per il nome composto di una associazione sono:

- *Records-current*
- *RecordsCurrent*

Tabella 9.2 Elenco di 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 noto/registrato/memorizzato/ riportato/acquisito in B	*Sale—Register* *Piece—Square* *Reservation—FlightManifest*
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*

9.16 Attributi

È utile identificare gli attributi delle classi concettuali necessari per soddisfare i requisiti informativi per gli scenari correnti in corso di sviluppo. Un **attributo** è un valore logico (un dato) di un oggetto.

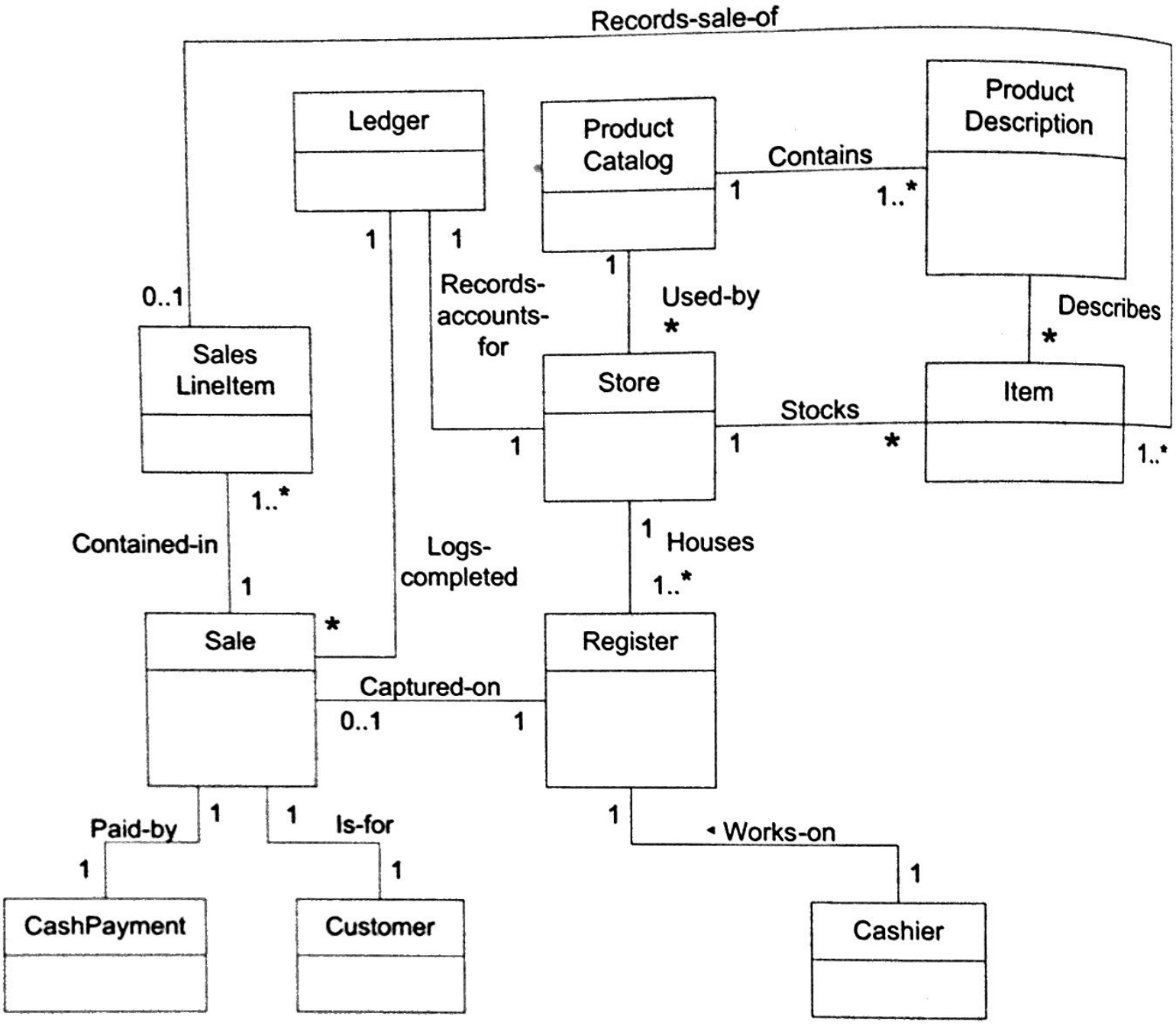


Figura 9.17 Modello di dominio parziale per POS NextGen.

Per esempio, una ricevuta (che riporta le informazioni di una vendita) nel caso d'uso *Process Sale* normalmente comprende, tra l'altro, una data e un'ora, il nome e l'indirizzo del negozio e l'ID del cassiere.
Pertanto:

- *Sale* necessita di un attributo *dateTime;*
- *Store* necessita di *name* e *address*;
- *Cashier* necessita di un *ID*.

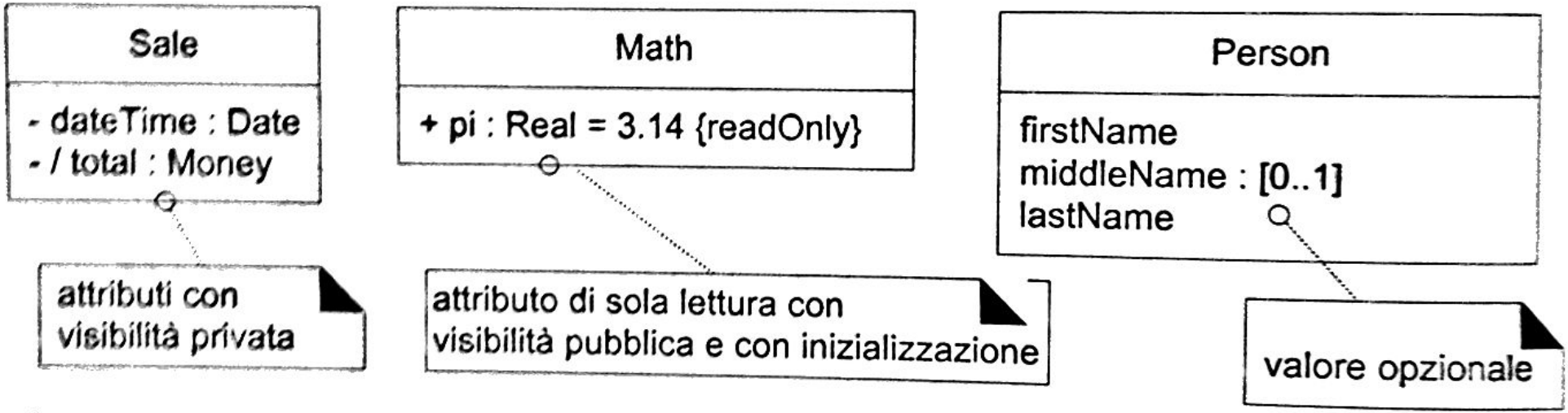


Figura 9.20 Notazione UML per gli attributi.

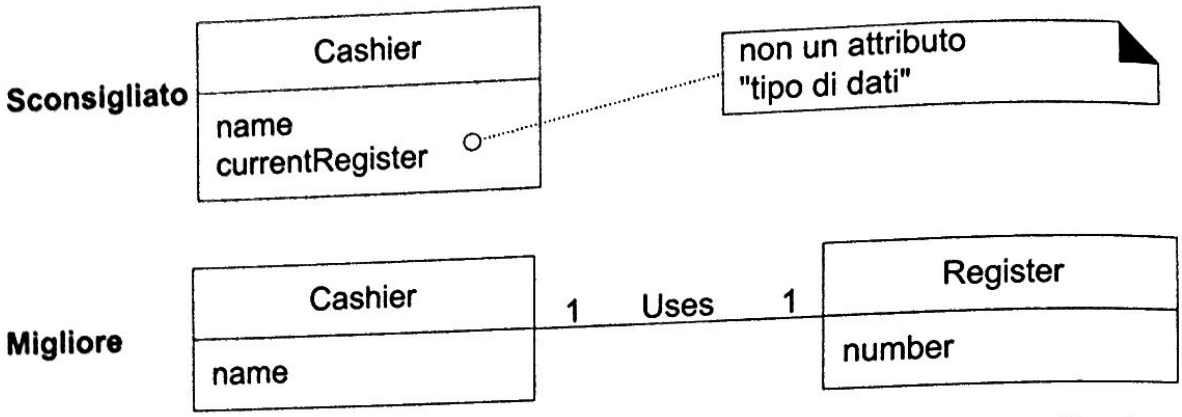


Figura 9.22 Classi concettuali vanno correlate con associazioni, non con attributi.

plice (quali *Number* o *String*). Il metodo più utile per esprimere il fatto che un *Cashier* utilizza un *Register* è tramite un'associazione, non con un attributo.

Linea guida

Gli attributi in un modello di dominio devono preferibilmente essere di **tipi di dato**. Tra i tipi di dato più comuni ci sono: *Boolean, Date* (o *DateTime*), *Number, Character, String* (*Text*), *Time.*

Altri tipi comuni sono *Address, Color, Geometrics (Point, Rectangle), Phone Number, Social Security Number* (*Codice Fiscale*), *Universal Product Code* (*UPC*), *SKU, ZIP* (*CAP*) o *codici postali, tipi enumerati.*

Per riprendere un esempio precedente, un equivoco comune consiste nel modellare un concetto complesso del dominio come un attributo. Per esempio, l'aeroporto di destinazione di un volo non è solo una stringa; è qualcosa di complesso, che occupa molti chilometri quadrati di spazio. Pertanto, *Flight* dovrebbe essere correlato ad *Airport* mediante un'associazione, non un attributo, come mostrato nella Figura 9.23.

Linea guida

Classi concettuali vanno correlate con un'associazione, non con un attributo.

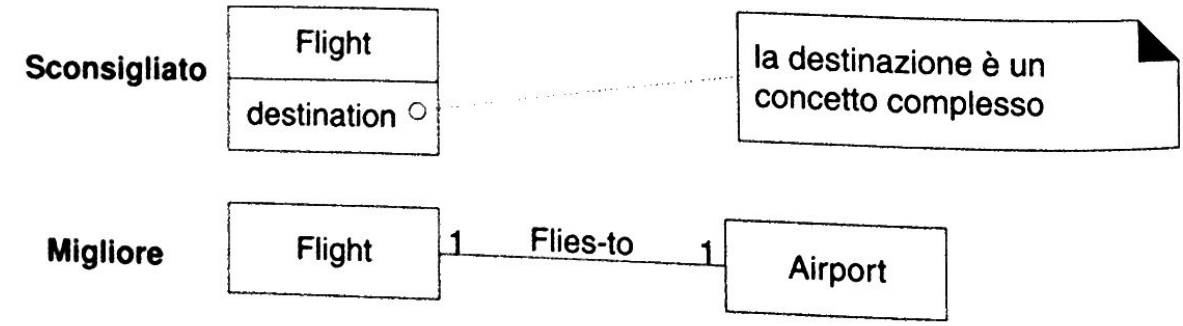


Figura 9.23 Concetti complessi non vanno mostrati come attributi; si utilizzino delle associazioni.

Ma non è soltanto questo (per esempio, i codici identificativi degli articoli hanno anche delle parti secondarie). In effetti, è utile avere nel modello di dominio una classe chiamata *ItemID* (o *ItemIdentifier*), e designare il tipo dell'attributo come tale. Per esempio, *itemID : ItemIdentifier*.
La Tabella 9.3 fornisce linee guida relative a quando è utile modellare con i tipi di dato. Applicando le linee guida agli attributi del modello di dominio per POS si ottiene la seguente analisi.

- Il codice identificativo degli articoli è un'astrazione di vari schemi di codifica comuni, compreso UPC-A, UPC-E e la famiglia degli schemi EAN. Questi schemi di codifica numerica hanno parti secondarie che identificano il produttore, il prodotto, il paese (per EAN), nonché una cifra che corrisponde alla somma di controllo per la convalida. Pertanto, ci dovrebbe essere una classe tipo di dato *ItemID*, dato che soddisfa molte delle linee guida proposte.
- Gli attributi *price* e *amount* dovrebbero essere di una classe tipo di dato *Money*, poiché sono quantità in un'unità monetaria.
- L'attributo *address* dovrebbe essere di una classe tipo di dato *Address*, poiché contiene sezioni separate.

Tabella 9.3 Linee guida per la modellazione con i tipi di dato.

Linea guida
Rappresentare nel modello di dominio come una nuova classe tipo di dato ciò che inizialmente può essere considerato un numero o una stringa se:
• È composto da sezioni separate.   ▪ per esempio, numero di telefono, nome di persona
• Ci sono operazioni a esso associate, come il parsing o la convalida.   ▪ numero di previdenza sociale, codice fiscale
• Ha altri attributi.   ▪ un prezzo promozionale potrebbe avere una data di inizio (effettiva) e una data di fine
• È una quantità con un'unità di misura.   ▪ l'importo di un pagamento ha un'unità monetaria
• È un'astrazione di uno o più tipi con alcune di queste qualità.   ▪ il codice identificativo di articolo nel dominio delle vendite è una generalizzazione di tipi come Universal Product Code (UPC) e European Article Number (EAN)

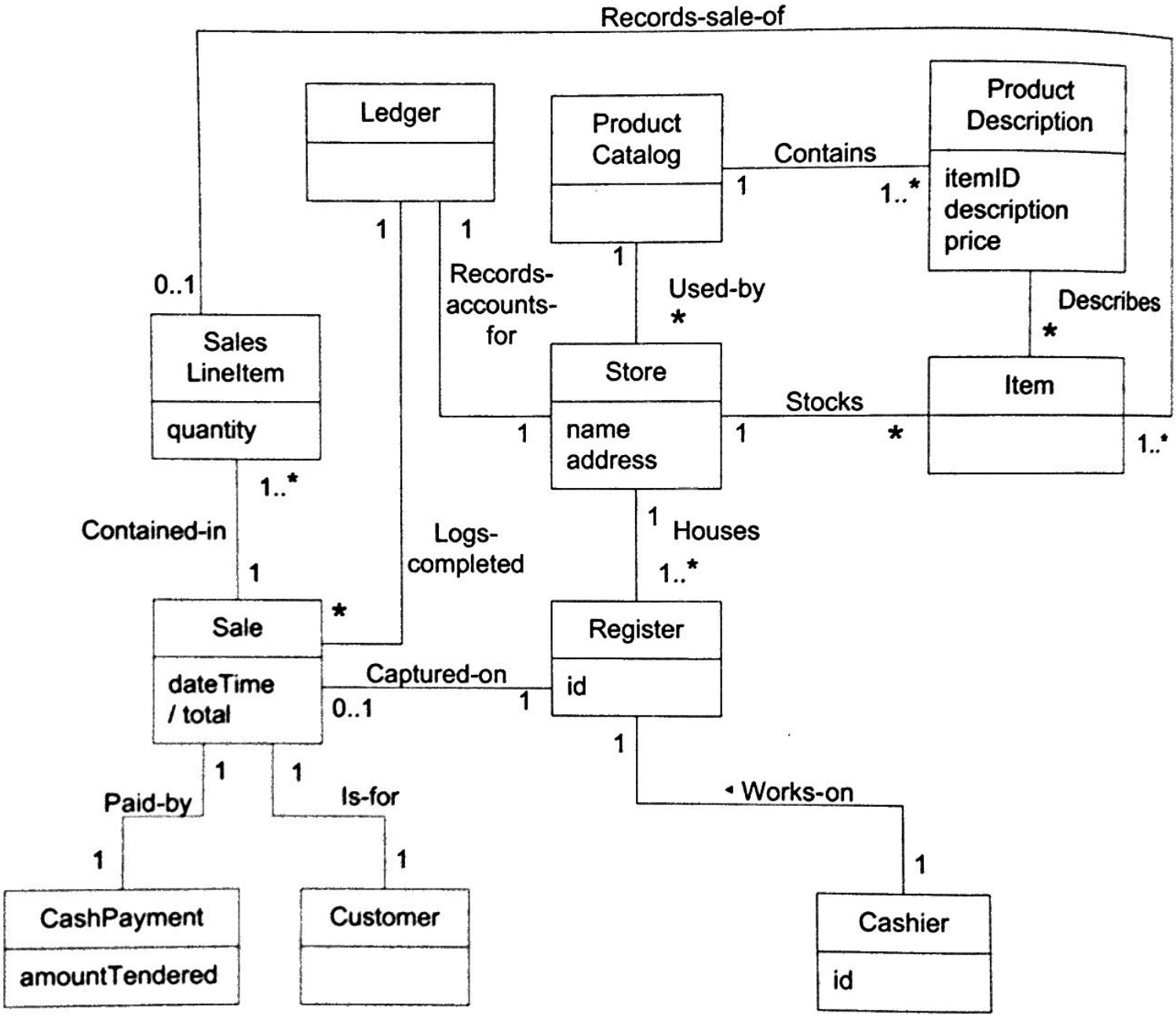


Figura 9.27 Modello di dominio parziale per POS NextGen.

9.18 Conclusione: il modello di dominio è corretto?

Non esiste un solo modello di dominio corretto. Tutti i modelli sono approssimazioni del dominio che si sta tentando di capire; il modello di dominio è in primo luogo uno strumento di comprensione e di comunicazione all'interno di un gruppo particolare. Una domanda migliore è: il modello di dominio è utile? Un modello di dominio utile coglie le astrazioni essenziali e le informazioni richieste per capire il dominio nel contesto dei requisiti correnti, e aiuta le persone nella comprensione del dominio (i suoi concetti, la terminologia e le relazioni esistenti).