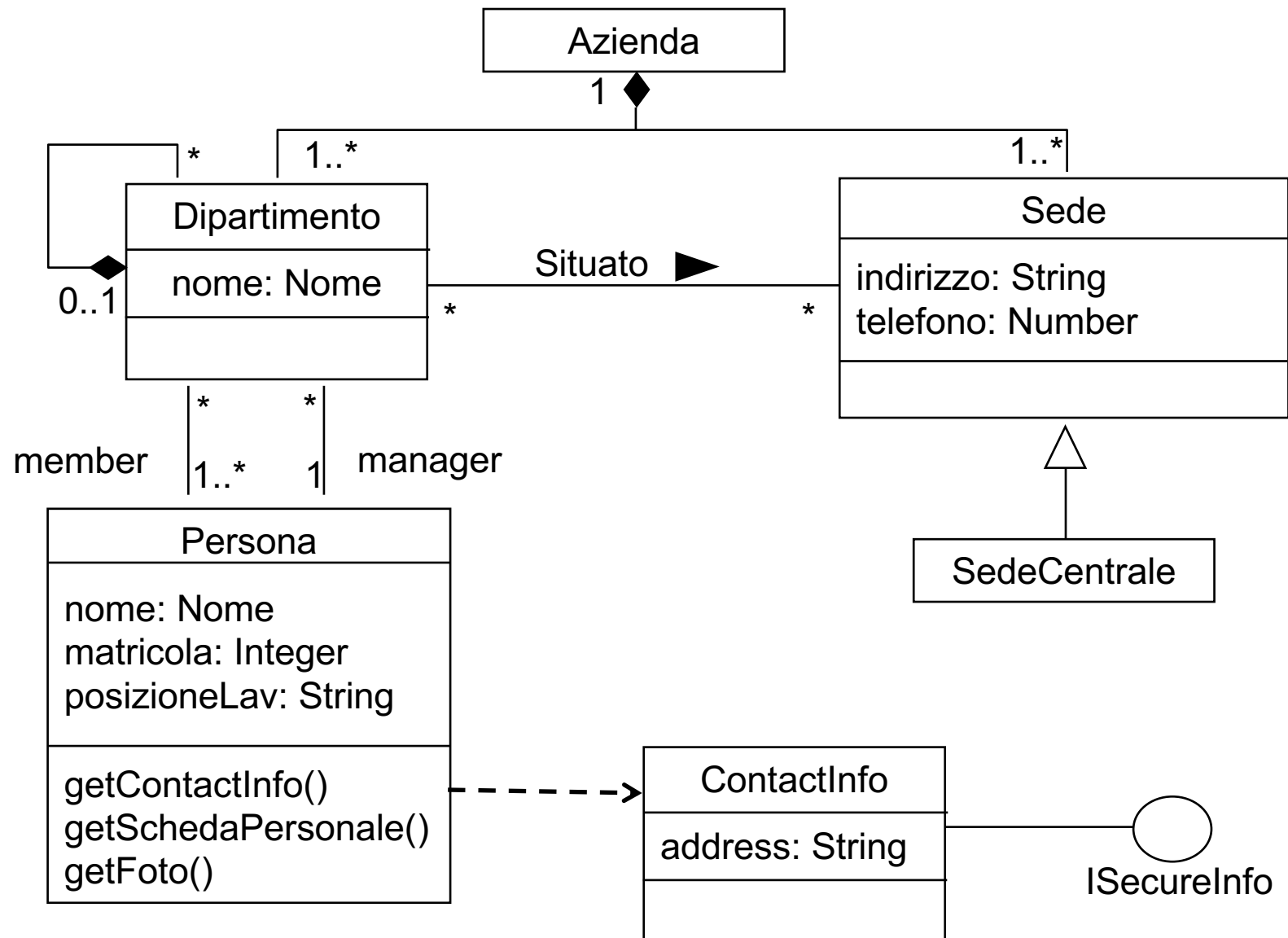


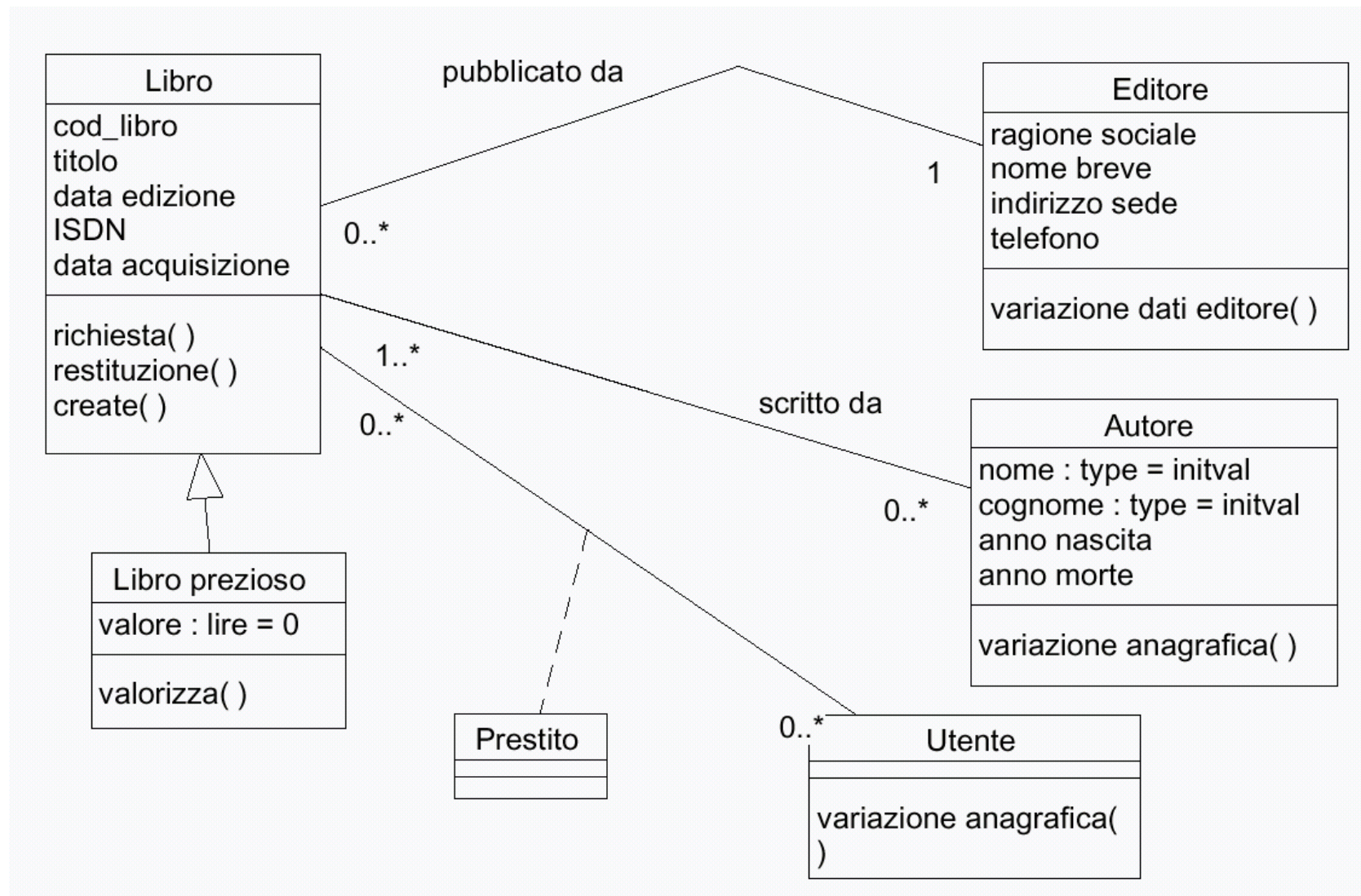
Class Diagram

- ◆ Un class diagram rappresenta un insieme di class, interface, collaboration e le loro relationship
- ◆ Un class diagram è tipicamente usato per modellare
 - la vista di static design di un sistema, che supporta principalmente i requisiti funzionali del sistema
 - se include active class descrive la vista statica di un processo di un sistema
 - il glossario di un sistema: sono prese decisioni relativamente alle astrazioni da considerare
 - semplici collaborazioni
 - lo schema concettuale di un database

Esempio di class diagram



Un altro esempio



Costruzione di un class diagram (1)

- ◆ Obiettivo:
 - identificare e caratterizzare gli elementi del modello a oggetti e come sono in relazione tra loro
- ◆ I diagrammi non sono il modello ma una vista sul modello
- ◆ Non cercare di mettere tutto su un solo diagramma
- ◆ Ogni diagramma deve avere uno scopo:
 - mostrare le classi e gli oggetti che partecipano in una singola collaborazione
 - mostrare una tassonomia di generalizzazione
 - mostrare la suddivisione delle partizioni logiche (packages)
- ◆ Non mostrare gli attributi e le operazioni nei diagrammi (se non per un motivo fondato)

Costruzione di un class diagram (2)

- ◆ Un class diagram ben strutturato è incentrato sul ‘comunicare’ un solo aspetto della vista dello static design
 - contiene solamente gli elementi che sono essenziali a comprendere quell’ aspetto
 - fornisce dettagli consistenti con il suo livello di astrazione, con solo quegli adornments che sono essenziali alla comprensione
- ◆ Ma non è così minimalista da disinformare il lettore circa la reale semantica
- ◆ Comprensibilità
 - non possiede troppi tipi di relationships (se si hanno relationship complicate è meglio mettere tali elementi in un ulteriore diagramma di dettaglio)
 - ha un layout che ne facilita la lettura
 - ha un nome significativo,
 - possiede eventualmente, note e colori per evidenziare aspetti particolari

Attività

- ◆ L'ordine non è rigido e le iterazioni sono numerose
 - Identificare le classi di oggetti
 - Preparare un dizionario dei dati
 - Identificare le associazioni (incluse le aggregazioni) tra classi
 - Identificare gli attributi delle classi e dei legami
 - Identificare le operazioni
 - Riorganizzare usando l'ereditarietà

Esempio: Problema del Bancomat

Il sistema software da progettare per gestire una rete bancaria automatizzata prevede che i cassieri e gli sportelli automatici (Bancomat) siano condivisi da un consorzio di banche.

Ogni banca fornisce il proprio computer per gestire i propri conti ed elaborare le transazioni su questi conti.

I terminali dei cassieri sono posseduti dalle singole banche e comunicano direttamente con il computer della propria banca. I cassieri inseriscono dati su conti e transazioni.

I Bancomat comunicano con un computer centrale che passa le transazioni alle banche appropriate.

Un Bancomat accetta una scheda magnetica, interagisce con l'utente, comunica con il sistema centrale per portare a termine la transazione, distribuisce il denaro, e stampa la ricevuta.

Il sistema richiede appropriati provvedimenti per la registrazione e la sicurezza.

Il sistema deve gestire correttamente accessi concorrenti allo stesso conto.

Ogni banca fornirà il proprio software per il proprio computer; bisogna progettare il software per i Bancomat e per la rete.

Il costo del sistema condiviso sarà distribuito proporzionalmente tra le banche secondo il numero di clienti con carta Bancomat.

Identificare le classi di oggetti

- ◆ Dove cercare
 - conoscenza generale del problema, descrizioni testuali, figure
- ◆ Cosa cercare
 - oggetti fisici, altri sistemi, dispositivi esterni, eventi da ricordare, ruoli, locazioni, unità organizzative
- ◆ Cosa considerare
 - candidati aventi confini ben definiti e identità distinte
 - candidati con proprietà da ricordare
 - candidati che forniscono o richiedono servizi
- ◆ Scartare:
 - candidati ridondanti, irrilevanti, o vaghi
 - candidati che rappresentano singoli oggetti, o operazioni su oggetti, o ruoli in associazioni
 - candidati legati alla realizzazione

Esempio: Identificare le classi di oggetti

◆ Classi selezionate

- Conto
- Bancomat
- Banca
- Scheda
- Cassiere
- Terminale del cassiere
- Cliente
- Transazione

◆ Classi scartate

- Vaghe
 - » Sistema, Provvedimento di sicurezza
 - » Provvedimento per la registrazione
 - » Rete bancaria
- Singoli oggetti
 - » consorzio
- Attributi
 - » Dati del conto, Ricevuta, Contante
 - » Dati della transazione
- Irrilevanti
 - » Costo
- Realizzazione
 - » Identificazione della transazione
 - » Accesso, Software, Linea di comunicazione

Preparare un dizionario dei dati

- ◆ Definire cosa la classe rappresenta nel contesto del problema
- ◆ Specificare quali sono le responsabilità della classe nel sistema

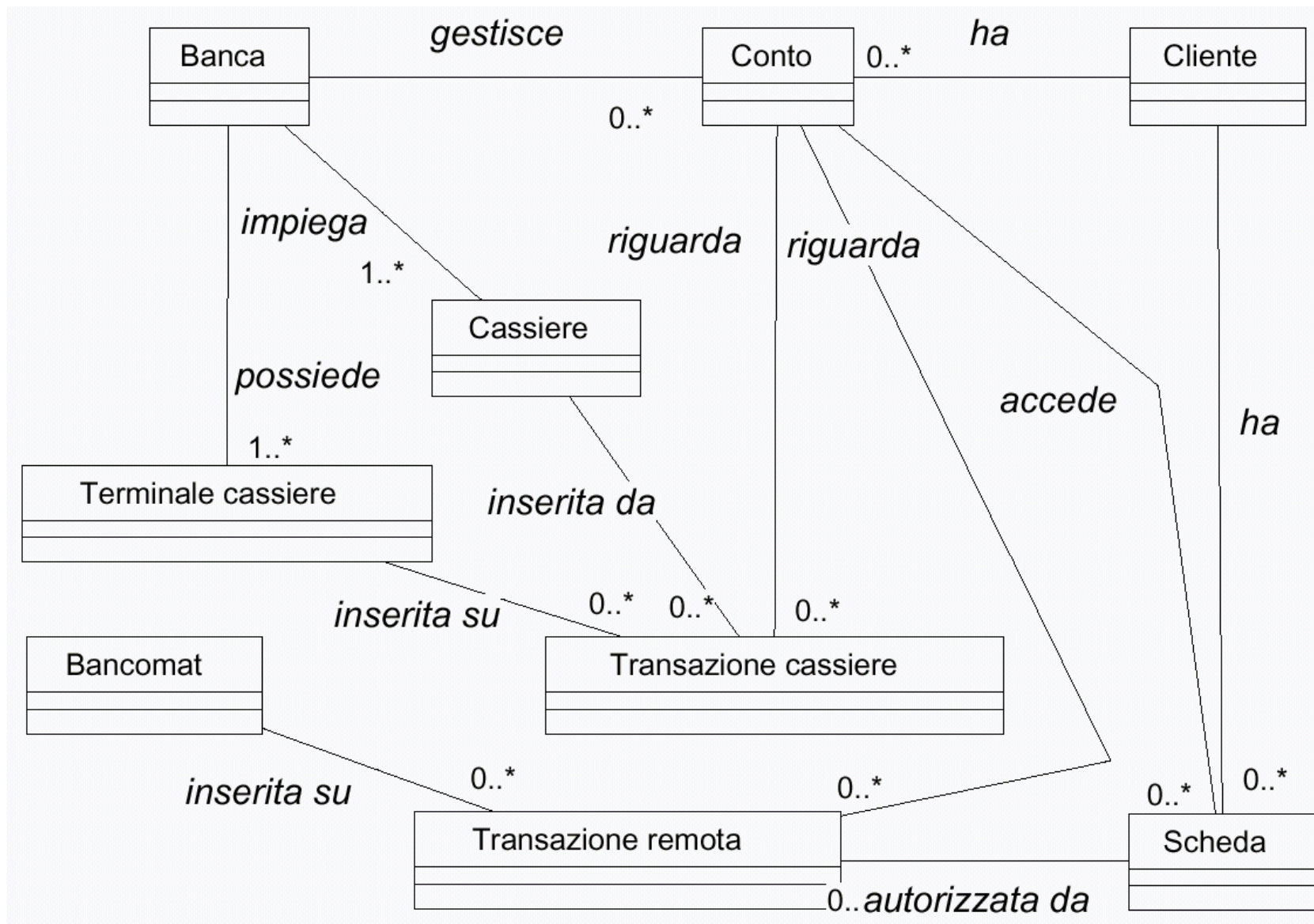
Esempio Bancomat

- ◆ *Conto*
 - *Rappresenta un singolo conto di un cliente in una banca. Tutti gli accessi e le modifiche al conto della banca devono avvenire attraverso questa classe*
- ◆ *Transazione Remota*
 - *Una richiesta integrale di operazioni sul conto effettuata dal cliente attraverso un Bancomat*

Identificare le associazioni

- ◆ Cercare le dipendenze tra classi
 - considerare le espressioni verbali nella descrizione del problema
- ◆ La relazione di aggregazione è un tipo speciale di associazione (“è parte di” o “si compone di”)
- ◆ Aggiungere i nomi alle associazioni o ai ruoli delle classi associate
- ◆ Specificare la molteplicità se possibile
- ◆ Scartare le associazioni:
 - non rilevanti per il problema specifico
 - troppo legate alla realizzazione
 - derivabili da altre associazioni

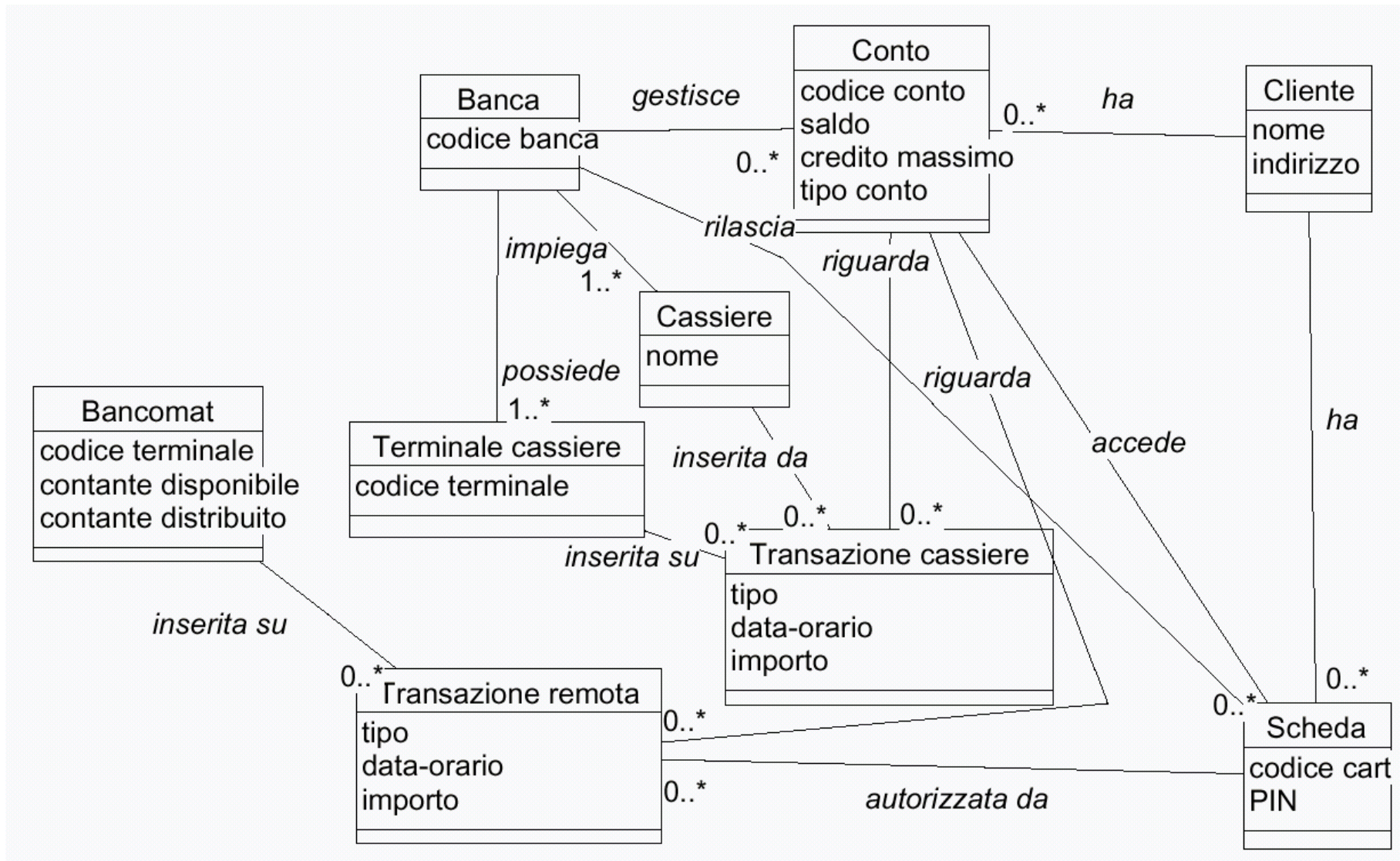
Esempio: Identificare le associazioni



Identificare gli attributi degli oggetti e dei legami

- ◆ Gli attributi necessari dipendono dal problema specifico
- ◆ In fase di analisi
 - Non essere esaustivi
 - Omettere
 - » attributi derivati
 - » attributi che descrivono stati interni alla classe

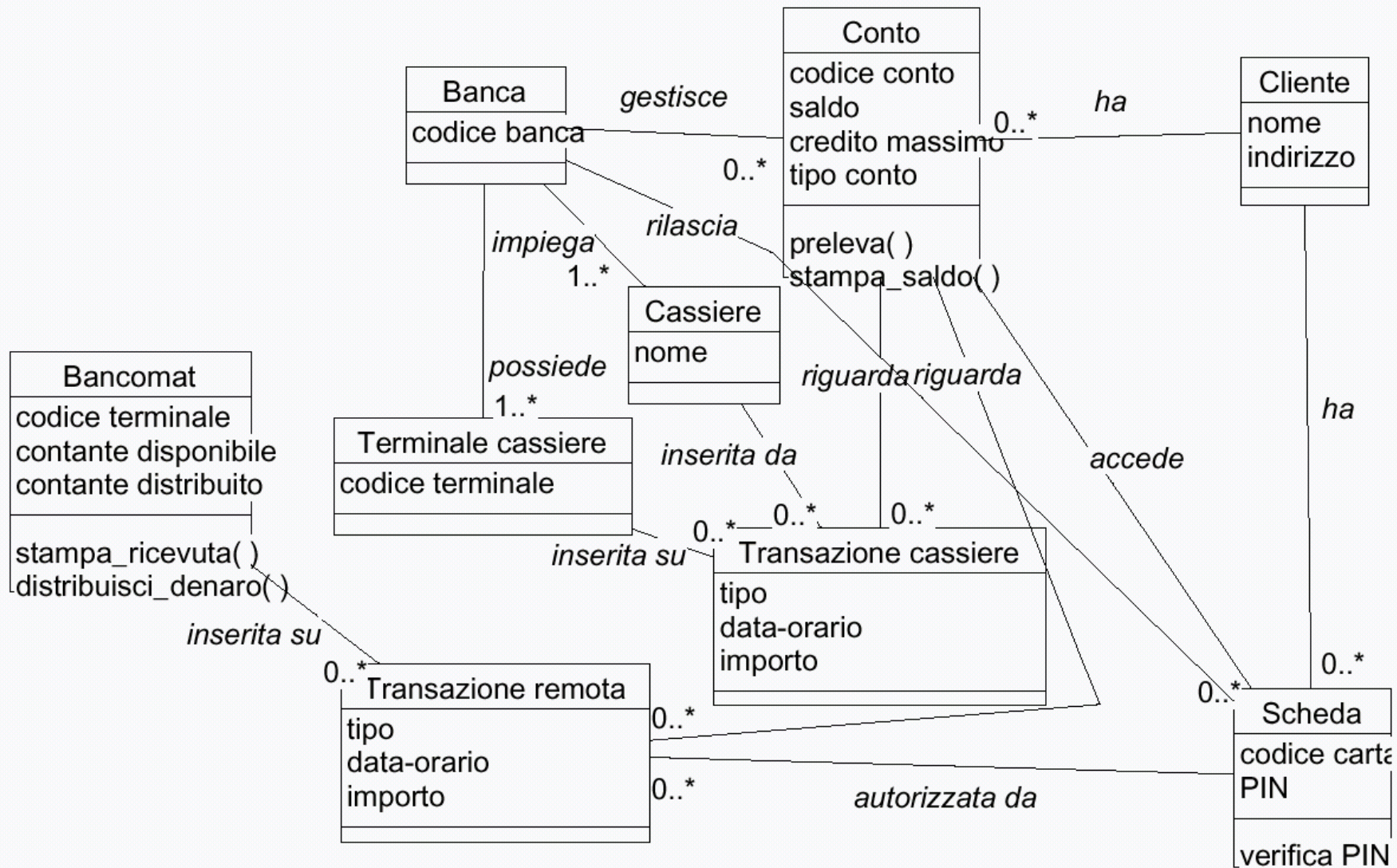
Esempio: Identificare gli attributi degli oggetti e dei legami



Identificare le operazioni

- ◆ Operazioni base (costruttore, selettore, modificatore, distruttore)
 - aggiungerle in questa fase solo se utili nella comprensione del problema o se menzionate esplicitamente nella descrizione
- ◆ Altre operazioni
 - Consultare gli scenari che descrivono i casi d'uso
 - Attribuire le funzionalità richieste alle classi identificate
 - Considerare le operazioni che modificano lo stato degli oggetti
- ◆ Non essere esaustivi in fase di analisi

Esempio: Identificare le operazioni



Riorganizzare usando l' ereditarietà

- ◆ Considerare solo classificazioni utili per il problema specifico
 - Specializzazione
 - » E' possibile raffinare una classe in sottoclassi con attributi e operazioni specifici?
 - Generalizzazione
 - » E' possibile identificare una superclasse che astrae attributi e operazioni comuni?
- ◆ Posizionare nella superclasse gli attributi e le operazioni comuni a tutte le sottoclassi

Esempio: Riorganizzare usando l'ereditarietà

