

Soluzione dell'esercizio del 2 Febbraio 2004

1. Casi d'uso

I casi d'uso sono riportati in Figura 1.

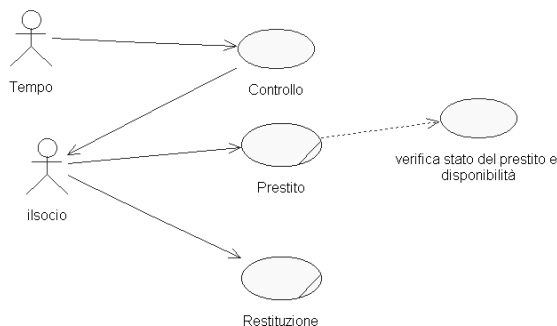


Figura 1: Diagramma dei casi d'uso. E' evidenziato un sotto caso di uso.

2. Modello concettuale

Osserviamo anzitutto che Circolo *possiede* opere, in una relazione uno a molti. Opera è una classe astratta alla quale corrispondono tre classi concrete CD, Video e Libro. Al circolo sono *associati* i soci. Si ha la struttura iniziale di Figura 2. Si noti che si è usata l'associazione di aggregazione dei soci e delle opere da parte del circolo. Il circolo viene a essere il contenitore di tutti i relativi oggetti.

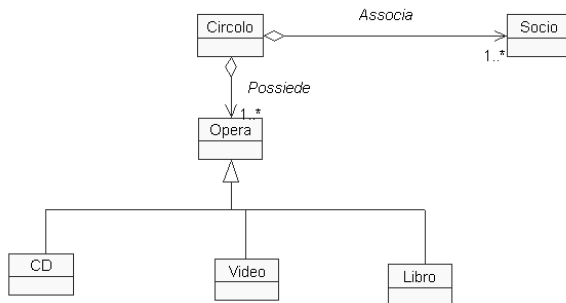


Figura 2: Diagramma UML: schema iniziale

Il diagramma si completa come in Figura 3. Sono stati aggiunti gli attributi che in questa fase appaiono rilevanti. Si faccia caso al fatto che tutti gli attributi sono privati (non visibili dall'esterno) e quindi, se per esempio serve sapere il titolo di una data opera, occorrerà prevedere un metodo che restituisca il titolo. Si noti l'uso dell'ereditarietà anche per le persone.

L'associazione di prestito è una associazione ternaria. Essa stessa ha infatti (almeno) un attributo proprio rappresentato dalla data di restituzione prevista.

3. Diagrammi di sequenza

In Figura 4 viene riportato il diagramma di sequenza corrispondente al caso d'uso della verifica dello stato del prestito e della disponibilità dell'opera. Per convenienza è stato posto un punto interrogativo ai metodi che riportano una risposta booleana.

?PrestConsentito() è il metodo di Socio che risponde se il prestito è consentito. Ovviamente, il prestito è consentito se Socio non ha prestiti in corso, ovvero se non ha associazioni con oggetti della classe Prestito. Se ci sono prestiti in corso si richiede che essi vengano esaminati (col metodo ?scaduto() della classe Prestito).

Per vedere se un'opera è in catalogo viene letto il titolo delle opere (tramite il metodo get_title()). In caso affermativo viene verificato che l'opera non sia in prestito (non abbia associazioni con oggetti della classe

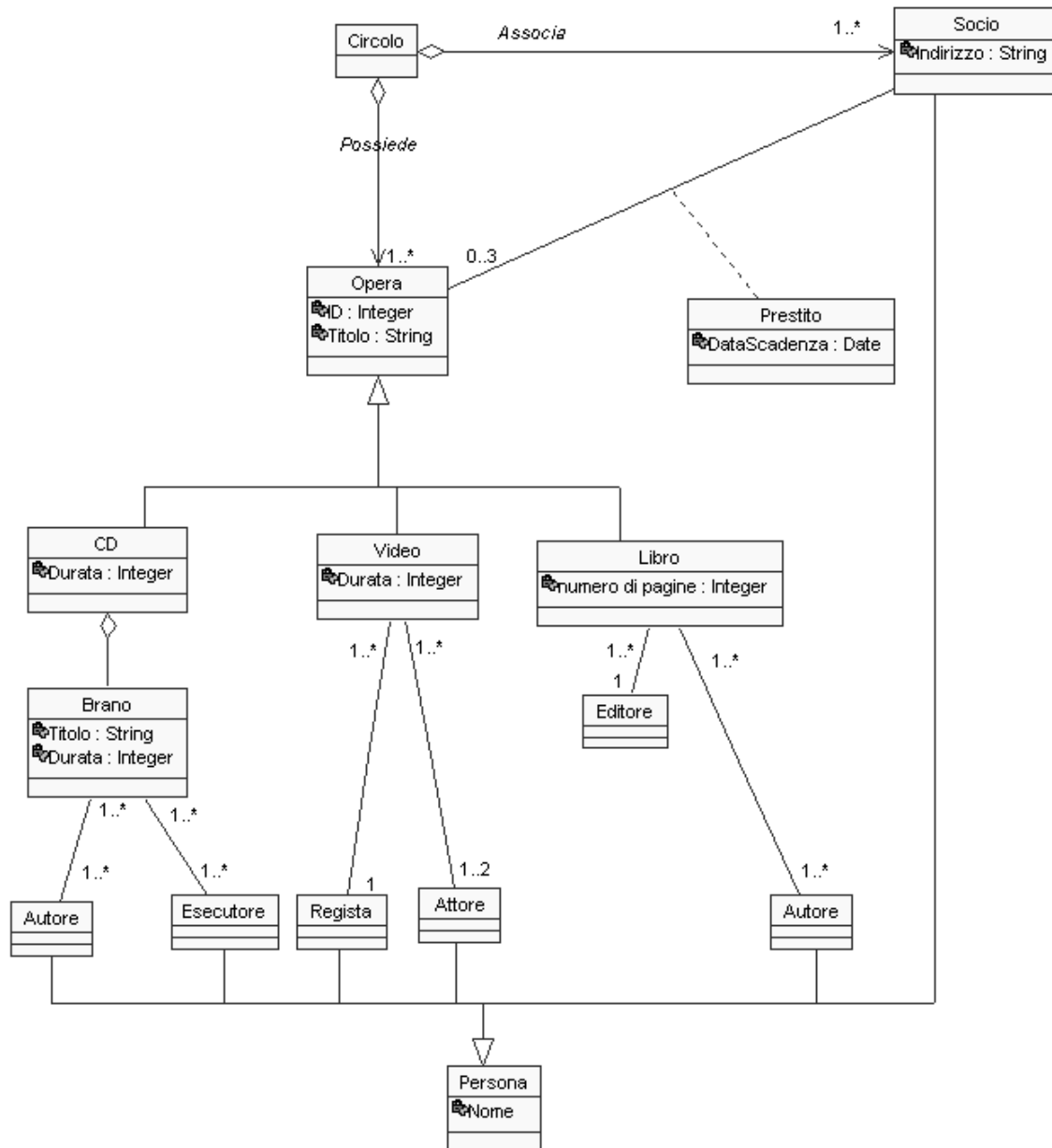


Figura 3: Modello concettuale completo.

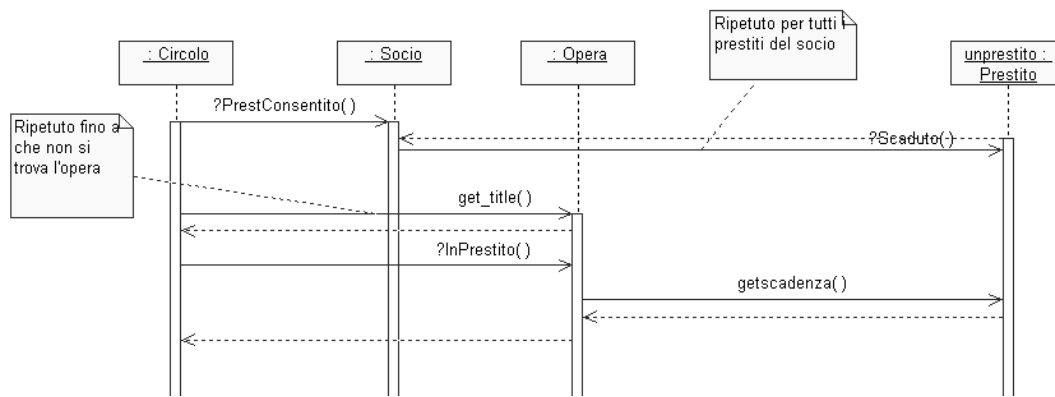


Figura 4: Sequence diagramma della verifica dello stato del prestito e della disponibilità.

Prestito). Se l'opera è in prestito il metodo `getscadenza()` della classe `Prestito` riporta la data di scadenza, che può essere comunicata al richiedente.

In Figura 5 viene riportato il diagramma di sequenza corrispondente al caso d'uso del prestito. Notare che l'opera data in prestito è responsabile di creare la classe `Prestito` e quindi, conoscendo se stessa (CD, Video o libro), può fissarne la scadenza all'atto della creazione del `Prestito` (col suo costruttore). Il metodo `SetInPrestito()` riporta al `Socio` (il riferimento a) il `Prestito` appena istanziato. `Opera` e `Socio` costruiscono quindi l'associazione con `Prestito`.

In Figura 6 viene riportato il diagramma di sequenza corrispondente al caso d'uso della restituzione. Il metodo `rilascia()` di `Socio` ha solo l'effetto di eliminare il riferimento di `Socio` al `Prestito` e di chiamare il metodo `SetDisponibile` di `Opera` che, a sua volta, distrugge il `Prestito` ed elimina il relativo riferimento che essa possiede.

Si noti che il meccanismo delineato non richiede una variabile di stato per indicare se un'opera è in prestito o no. Un'opera è in prestito se c'è una associazione con un `Prestito`, altrimenti è disponibile.

In Figura 7 è riportato il diagramma di sequenza relativo al controllo e della generazione degli avvisi. In questo caso sono stati introdotti due ulteriori oggetti, `ilTempo` e `ilMailer`. Essi esistono in copia unica. Il primo è un oggetto avente funzione di controllo (attivare la sequenza), il secondo è un oggetto di interfaccia a cui arrivano i messaggi confezionati da `Opera`.

4. Responsabilità delle classi

Vedere i diagrammi di sequenza e la Figura 8, in cui si mostrano le operazioni di ciascuna classe. Per le classi derivate da `Opera`, alcuni metodi possono essere definiti in `Opera` e derivati senza modifica (per esempio `get_title()`); altri devono essere ridefiniti nelle classe derivate (per esempio, `avvisa()`).

5.

Vedere i diagrammi di sequenza

6.

Tramite il polimorfismo (NON con statement `if`). Il diagramma di Figura 5 mostra che è `Opera` a creare un `Prestito` e quindi essa sa quanto può durare e fissare la data di scadenza (tramite il costruttore del `Prestito`). La Figura 7 mostra che è sempre `Opera` a chiedere il servizio a `ilMailer`. `Opera` ottiene dal `Prestito` l'indicazione che esso è scaduto (data, etc), aggiunge l'identificazione di se stessa nel messaggio da trasmettere e chiama il metodo `trasm()` del `Mailer`.

La trasmissione dell'eventuale messaggio con richiesta di pagamento avviene allo stesso modo. Per richiedere la cifra occorre prevedere che tra gli attributi di un'opera ci sia anche il costo in modo da calcolare la cifra dovuta dal socio. La Figura 2 deve essere aggiornata con aggiungendo questo attributo.

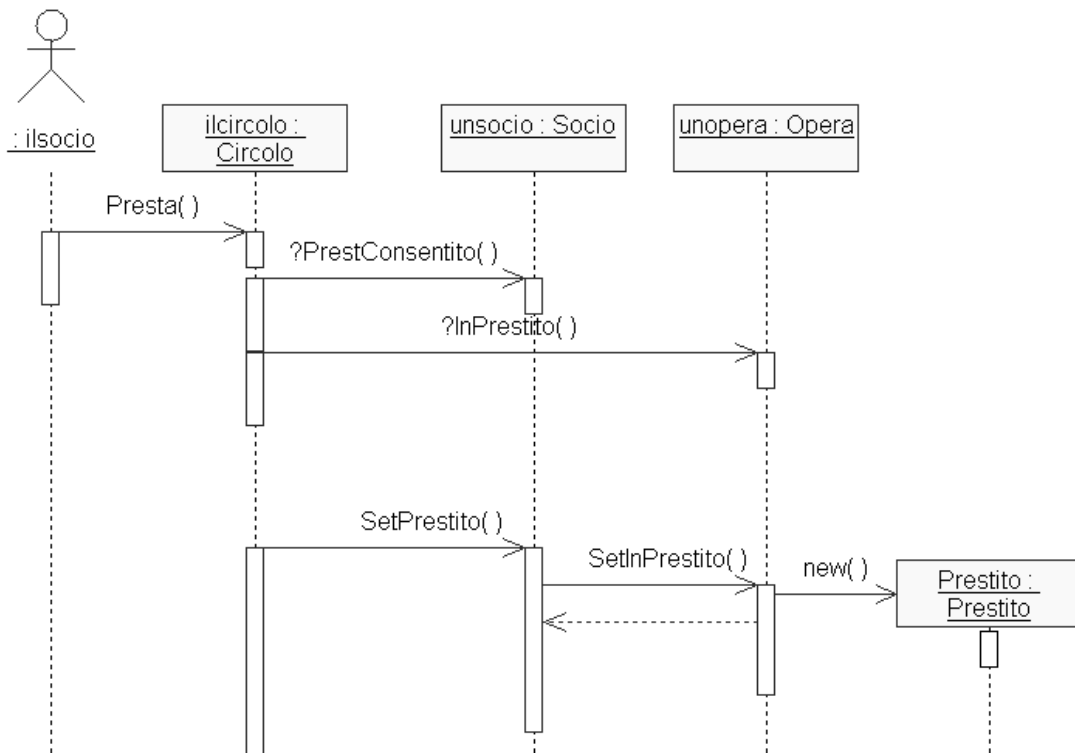


Figura 5: Sequence diagram del prestito. La parte alta corrisponde al caso di uso di Figura 4 di cui però si riporta solo gli elementi principali. Ovviamente la fase di prestito ha luogo solo se la verifica ha dato esito positivo.

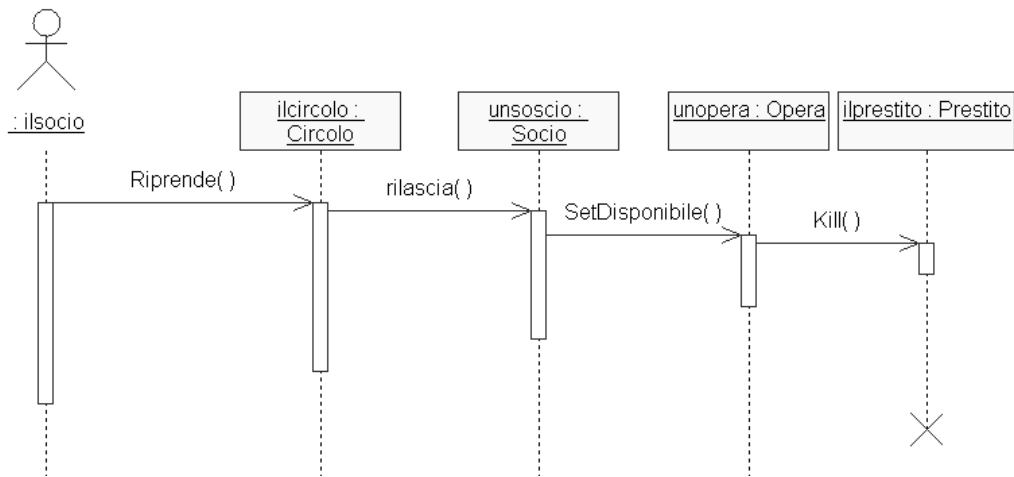


Figura 6: Sequence diagram della restituzione.

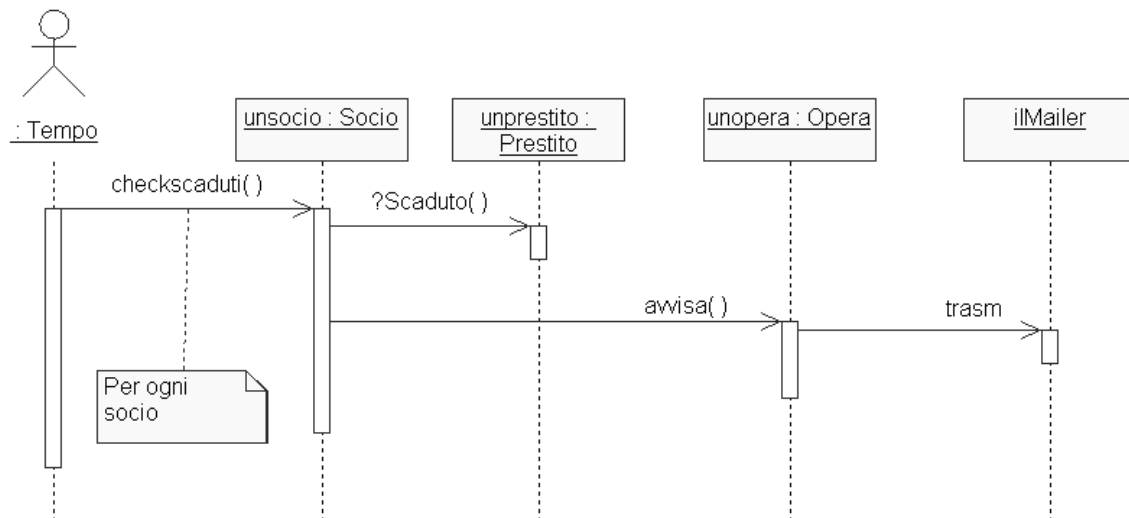


Figura 7: Sequence diagram del controllo e generazione avvisi. In questo caso sono messi in evidenza il Tempo (l'attore che determina l'esecuzione della sequenza e l'oggetto Mailer responsabile di raccogliere i messaggi e di trasmetterli.

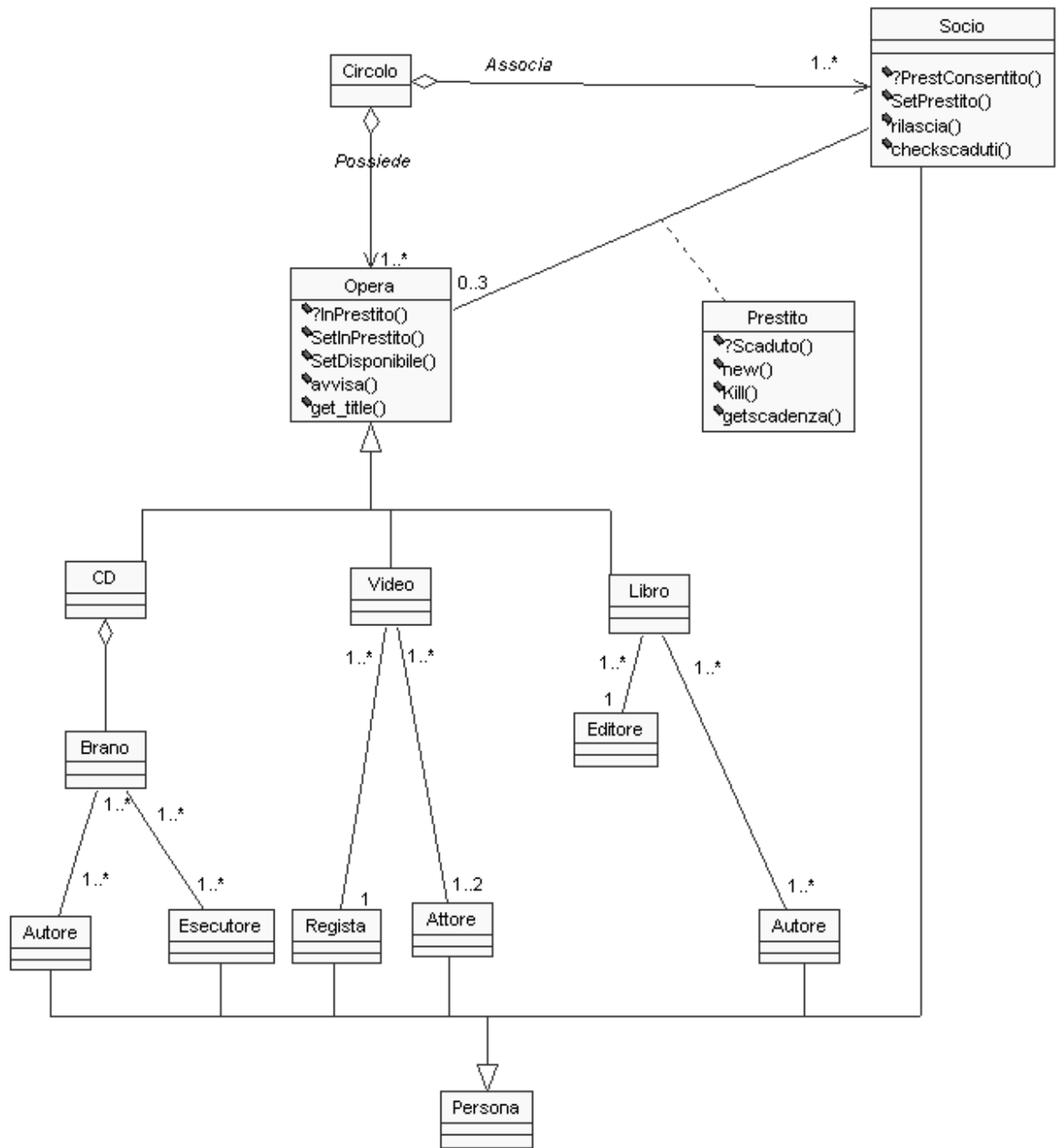


Figura 8: Interfacce delle classi.

Soluzione alternativa

Una caratteristica della soluzione proposta è il ricorso alla associazione ternaria di Prestito. Si pone la domanda se essa è effettivamente necessaria.

Nel caso specifico, l'associazione di Prestito in forma di classe non è essenziale; infatti:

- le opere sono in versione unica e un'opera può essere prestata a un solo socio alla volta;
- non è richiesto che si tenga traccia dei precedenti prestiti e quindi, come è stato fatto nella soluzione proposta, un prestito cessa di esistere alla sua conclusione;
- per tenere traccia di un prestito basta l'associazione Opera-Socio, prevedendo che Opera abbia tra i suoi attributi una variabile di stato che indica se essa è o no in prestito e una variabile che dia la data di scadenza del prestito.

Con queste modifiche il modello concettuale diventa quello di Figura 9, dove è stata riportata solo la parte di interesse.

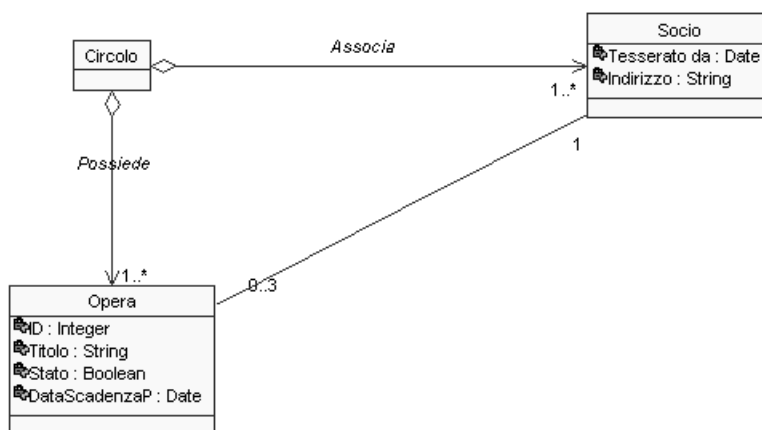


Figura 9: Modello del dominio dopo l'eliminazione della classe di associazione Prestito.

In Figura 10 viene mostrato il diagramma di sequenza del caso di uso di verifica stato del prestito e disponibilità. ?PrestConsentito() è il metodo di Socio che risponde se il prestito è consentito. Questa volta la verifica è se ci sono associazioni con Opera e, in caso positivo, se i prestiti in corso non sono scaduti. Il metodo ?scaduto() è ora un'operazione della classe Opera. La verifica che un'opera sia in prestito coinvolge solo Opera. Il metodo ?InPrestito() può riportare la data di scadenza del prestito se è in corso.

Il caso d'uso del prestito si semplifica come in Figura 11. L'operazione SetInPrestito() fa cambiare lo stato di Opera e assegnare il valore dovuto alla componente DataScadenzaP. Ovviamente stabilisce l'associazione tra Socio e Opera.

Il caso d'uso della restituzione diventa come in Figura 12. Quello del controllo e generazione avvisi diventa come in Figura 13. Il significato dei metodi è scontato.

Il diagramma delle classi di Figura 14 mostra le interfacce delle classi.

Considerazioni

E' lecito domandarsi a cosa serve una associazione ternaria.

Supponiamo che un'opera oltre a essere prestata possa essere prenotata. E' chiaro che una data opera può avere più prenotazioni e che ogni prenotazione si distingue dalle altre per la data. E' perciò evidente che una prenotazione è un oggetto a sé stante, con propri attributi, che definisce una possibile associazione tra un'opera e un socio.

Se il problema in esame avesse richiesto di tenere traccia dei prestiti pregressi saremmo stati nella medesima condizione. Infatti ogni prestito, una volta creato, avrebbe dovuto restare permanentemente nel sistema. In questo caso il Prestito avrebbe richiesto più attributi (data di inizio, di fine e di effettiva restituzione), in modo

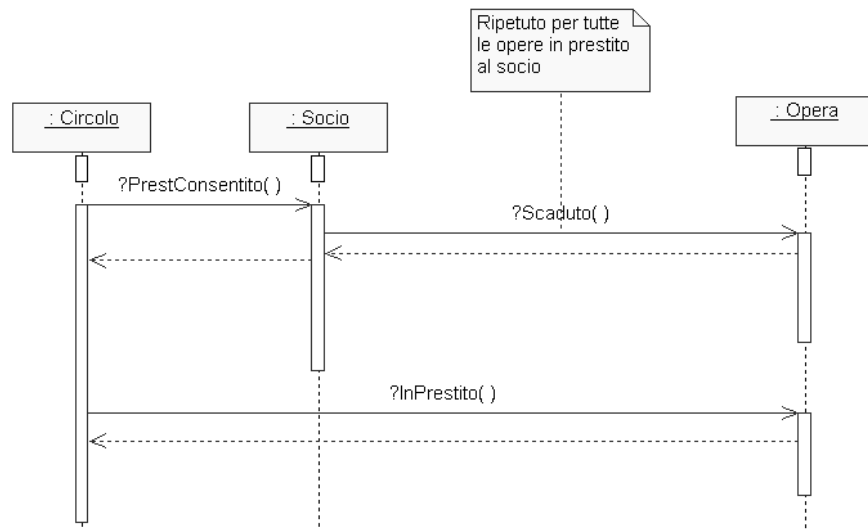


Figura 10: Diagramma della verifica dello stato del prestito e della disponibilità.

da poter decidere del suo stato. Con una tale organizzazione sarebbe possibile chiedersi, ad esempio, quanti prestiti ha ottenuto un socio (in un dato anno, o a partire da una certa data) esaminando i prestiti associati al socio e conteggiando quelli che soddisfano la condizione imposta dalla domanda.

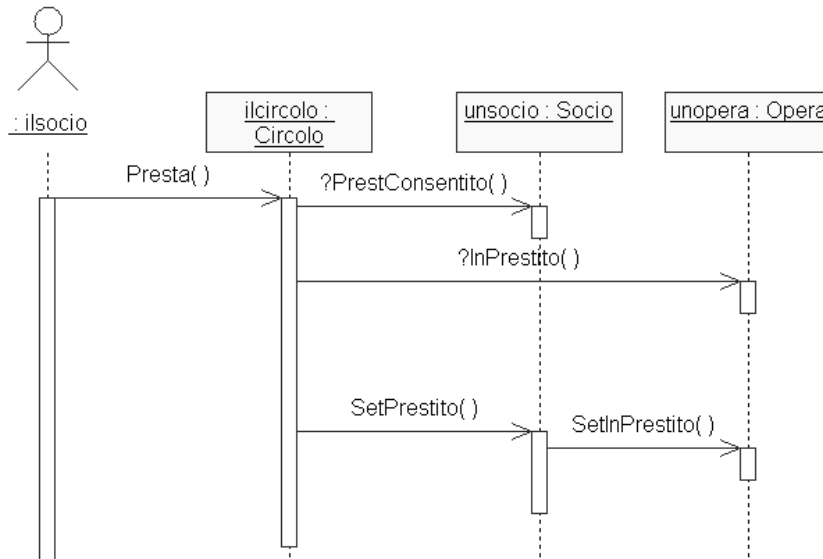


Figura 11: Nuovo diagramma di sequenza del prestito. La parte alta corrisponde sempre al caso di uso della verifica dello stato del prestito e della disponibilità di Figura 10. La fase di prestito ha luogo solo se è la verifica ha dato esito positivo.

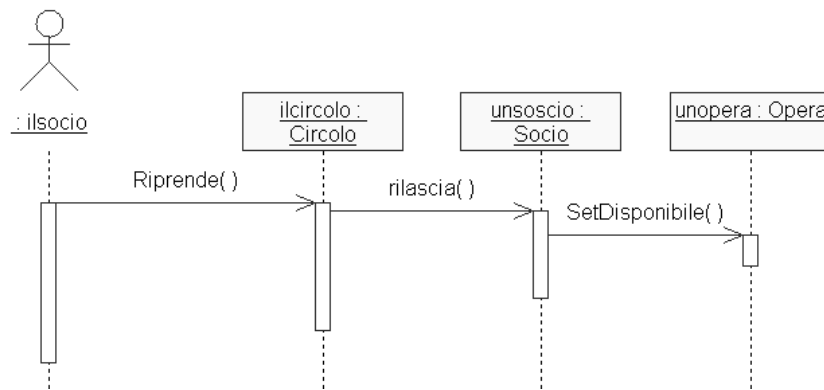


Figura 12: Diagramma di sequenza della restituzione.

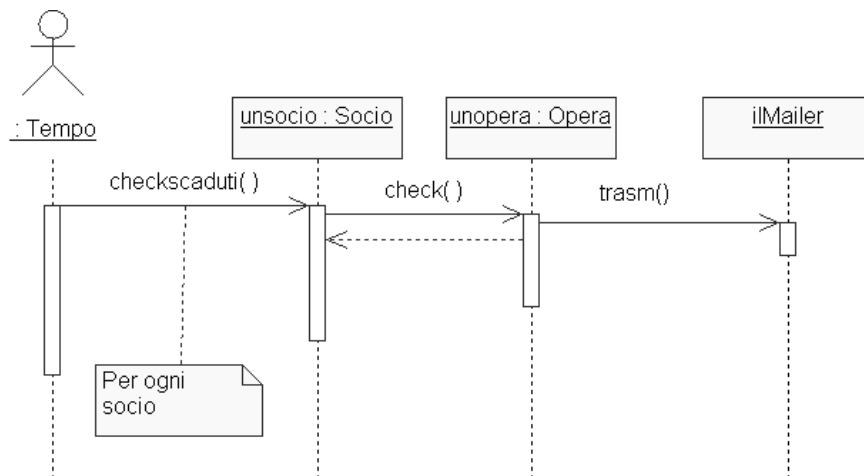


Figura 13: Diagramma di sequenza del controllo e generazione avvisi. E' sempre Opera che genera il messaggio di avviso e chiede al Mailer di trasmetterlo.

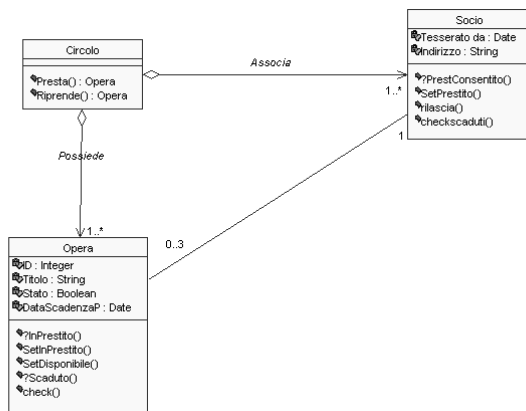


Figura 14: Diagramma delle classi e loro interfacce.