

# Traccia di soluzione dell'esercizio del 14/07/2005

## 1 Attori

Gli attori sono “il socio”, il “non socio” e, volendo considerare le azioni effettuate sul sistema come la messa a cartellone (cosa non richiesta) l'impiegato. Si noti che è vero che col sistema interagisce direttamente l'impiegato, nel senso che sarà lui a battere sulla tastiera del computer, ma chi chiede di svolgere attività al sistema sono il socio e il non socio.

## 2 Modello concettuale

La difficoltà dell'esercizio deriva dal fatto che alcune associazioni sono espresse tramite classi (associazioni ternarie). Mostriamo come si ottiene il modello concettuale, procedendo passo passo.

Anzitutto si ha che l'ufficio gestisce una molteplicità di spettacoli e che l'ufficio ha una molteplicità di soci. In uno spettacolo recitano 1 o più attori (Figura 1).

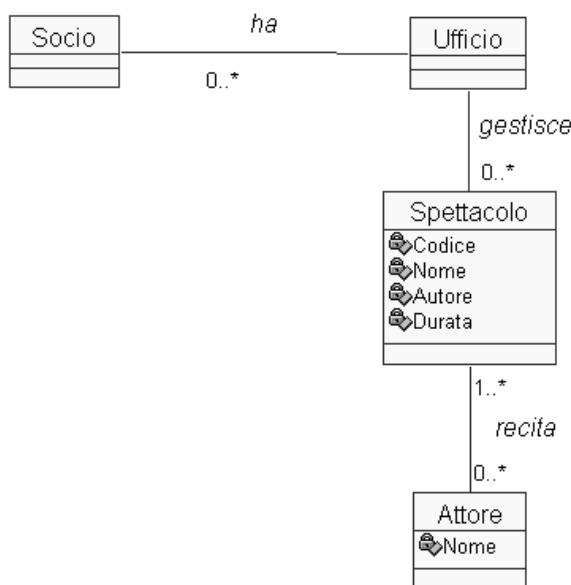


Figura 1: Inizio della costruzione del modello.

Uno spettacolo può essere tenuto in più teatri, ma in date diverse. L'associazione tra spettacolo e teatro può essere data attraverso la classe DataPresentaz (Figura 2). A sua volta il teatro è suddiviso in settori numerati e ciascun settore contiene posti numerati.

A questo punto bisogna tener conto del fatto che ogni data di presentazione può prevedere una o 2 presentazioni, e che ogni presentazione ha i suoi posti prenotati. Conviene prevedere una associazione tra Presentazione e Posti attraverso la classe Prenotazione. Una prenotazione ha uno stato che può assumere questi valori: (a) non prenotato; (b) prenotato, ma non pagato (non può essere la prenotazione di un non socio); (c) prenotato e pagato (prenotazioni di non soci ma anche di soci). A una prenotazione può essere o no associato un biglietto.

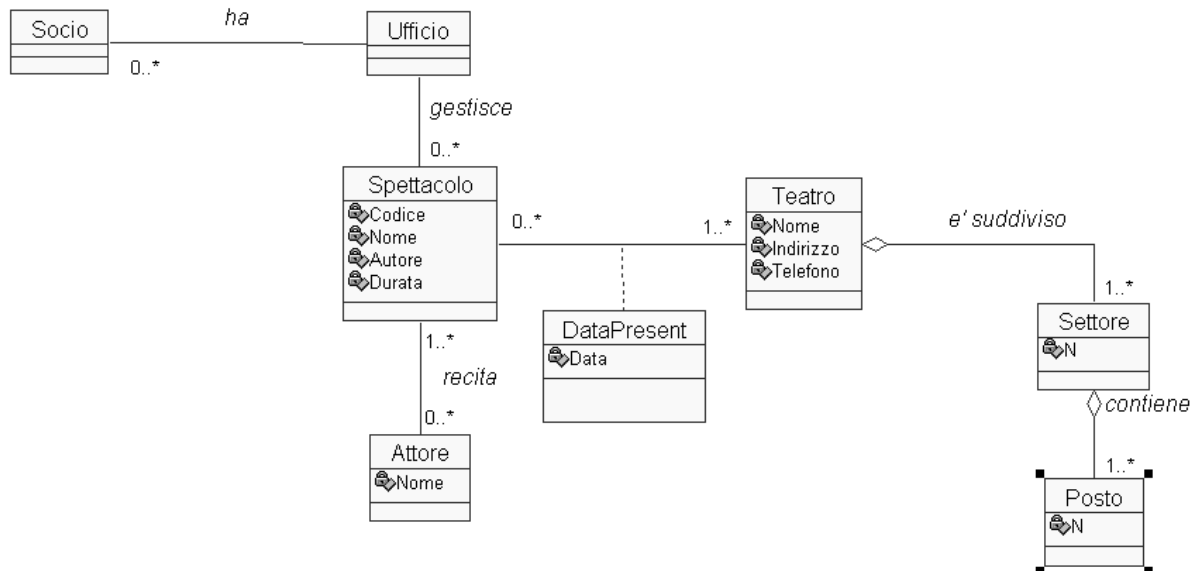


Figura 2: Aggiunta della associazione tra Spettacolo e Teatro.

Lo stato della prenotazione dice direttamente se il biglietto è stato emesso o no (ovvero se c'è l'associazione) Si arriva così alla Figura 3.

Vale la pena di osservare che il modello rappresenta bene il modo di operare delle agenzie che effettuano prenotazioni e vendono biglietti. Queste per ogni spettacolo tengono una pianta del teatro e segnano i posti man mano che vengono prenotati e/o venduti.

Il Biglietto deve possedere una funzione `costo()`. Navigando il diagramma, da ogni biglietto si risale a quale presentazione, a quale settore ecc. esso si riferisce. In base a questi parametri il metodo `costo()` ne calcola il costo.

Il testo dice che si deve tenere traccia solo dei soci. Dei non soci, ovvero del generico spettatore, non si deve tenere traccia. Dunque nel sistema c'è solo la classe `Socio` quale rappresentativa degli spettatori. Un `Socio` potrà effettuare 0 o più Prenotazioni. In conclusione si ottiene il diagramma UML di Figura 5

Vale la pena di osservare che il modello consente di ricostruire quali e quanti biglietti un dato `Socio` ha acquistato nell'anno.

In Figura ?? viene data una soluzione alternativa per il modello. Il lettore è invitato a fare un confronto con la precedente.

### 3 Altre domande

Si lascia come esercizio la verifica della fattibilità del modello (dei modelli) attraverso i diagrammi di sequenza.

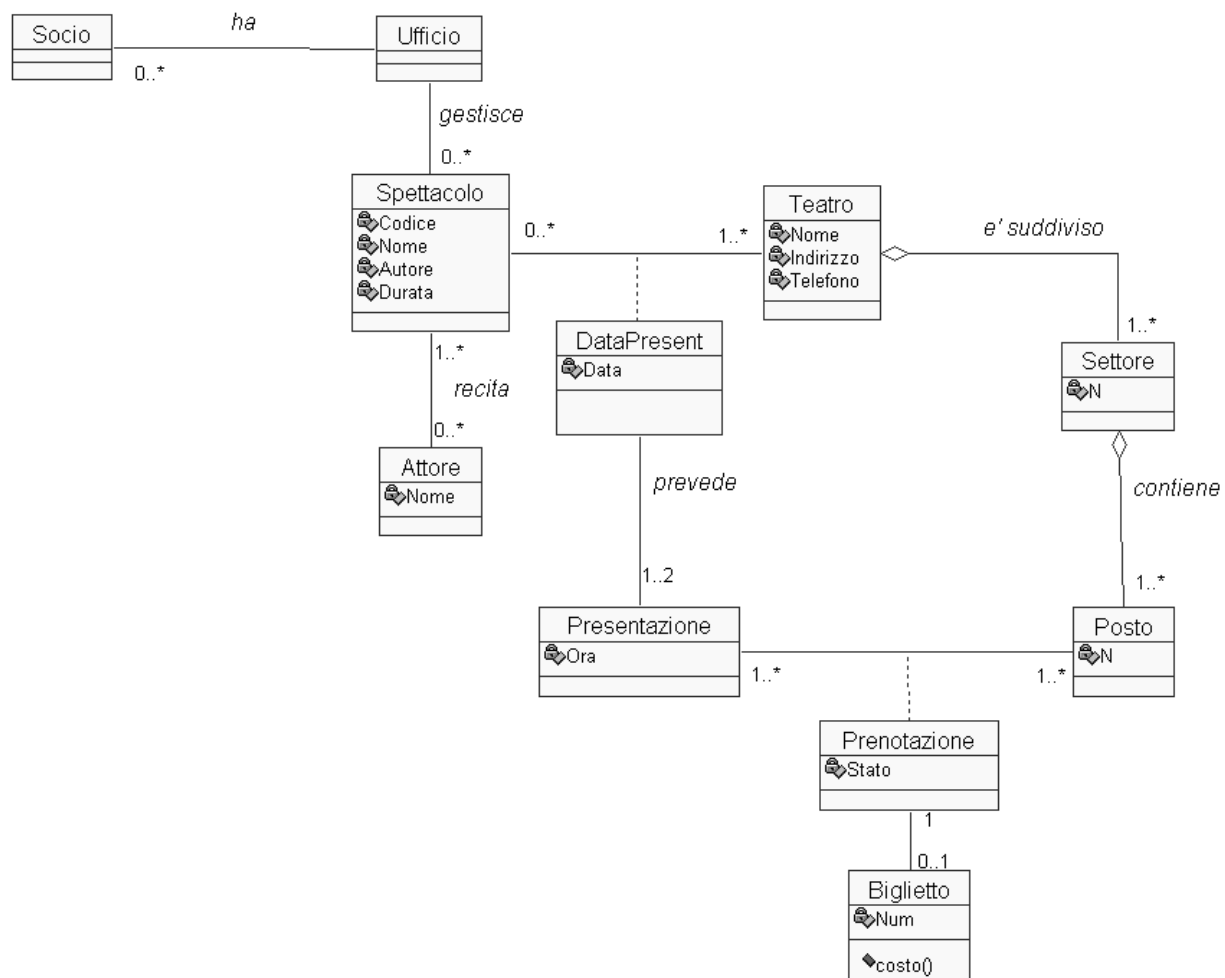


Figura 3: Presentazioni, Prenotazioni e Posti.

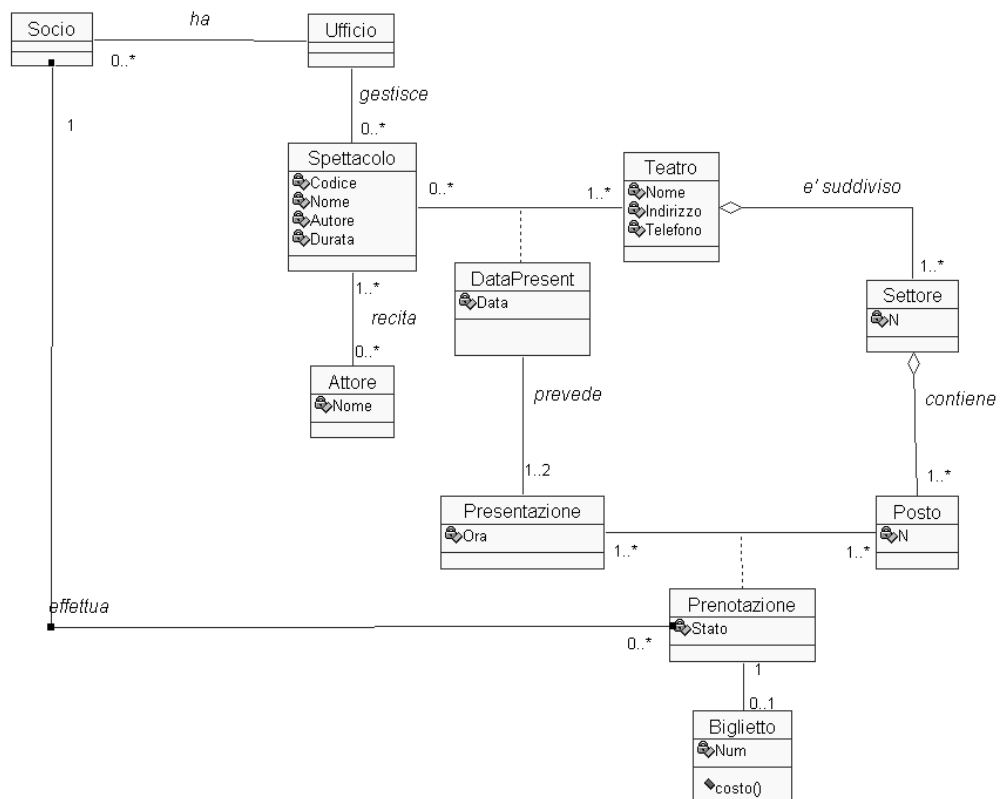


Figura 4: Modello concettuale completo.

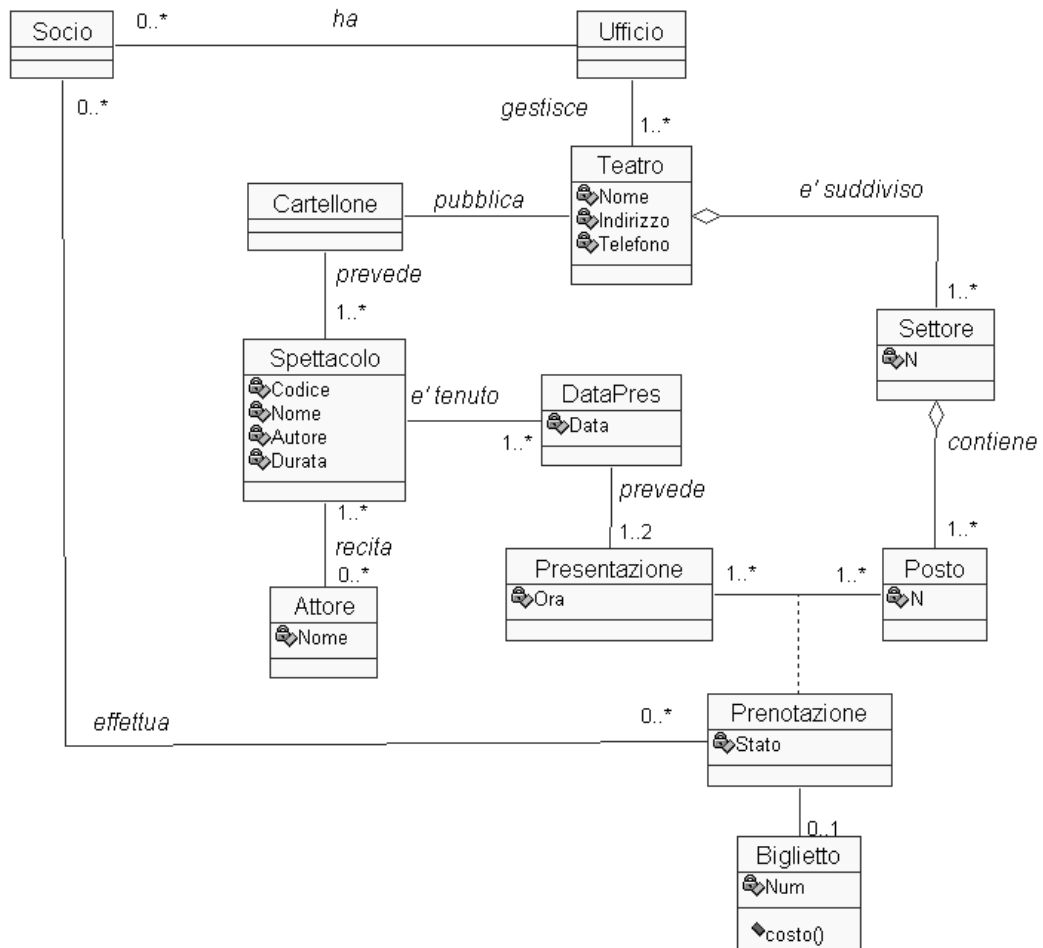


Figura 5: Soluzione alternativa.