

Esercizio **E.III.20080918**

versione del 20 settembre 2008

Un consorzio di teatri vuole sviluppare un'applicazione web che consenta agli utenti di prenotare biglietti per spettacoli teatrali negli enti consorziati.

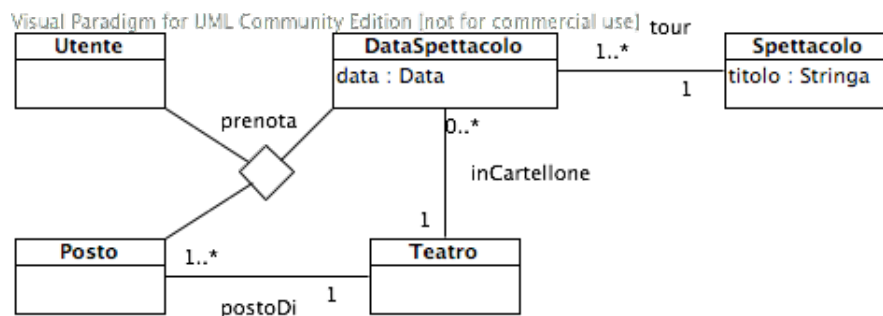
Tra gli altri, la specifica completa dei requisiti contiene il seguente frammento:

Gli spettacoli, di cui interessa il titolo, hanno più date di rappresentazione e, in generale, sono itineranti. Pertanto, il sistema deve consentire di mantenere l'informazione circa il teatro in cui avrà luogo la rappresentazione di uno spettacolo in ogni sua data.

I teatri hanno un insieme di posti prenotabili dagli utenti.

Il sistema deve consentire ad un utente (e memorizzare le informazioni richieste) di prenotare uno o più posti per uno spettacolo in una certa data.

Un brillante analista ha definito il seguente diagramma delle classi UML concettuale per l'applicazione (molti degli attributi sono stati omessi per semplicità; il rombo indica una associazione ternaria):



1. Il diagramma soffre di alcuni seri problemi. Quali? (Indicarne almeno uno)
2. Quale strumento software utilizzereste per dimostrare il punto 1?

3. Per lo strumento scelto, scrivere il relativo file di input e l'output atteso.
4. Come correggereste i problemi riscontrati?

**Soluzione.**

1. Ad esempio, il diagramma permette che lo stesso posto sia prenotato, per lo stesso spettacolo nella stessa data, da due utenti diversi. Esistono altri seri problemi: ad esempio un utente può addirittura prenotare, per uno spettacolo, un posto che si trova in un altro teatro! 2. Per dimostrare la presenza di tali eventualità va usato un ricercatore di modelli, ad es., Maice (un theorem prover sarebbe inutile, visto che il diagramma ammette anche scenari che rispettano i requisiti, quindi la formula ottenuta non è contraddittoria). 4. Vanno aggiunti opportuni vincoli esterni al diagramma: per il primo problema ad es., va aggiunto il vincolo  $\forall p, ds, n_1, n_2 (Posto(p) \wedge DataSpettacolo(ds) \wedge Utente(n_1) \wedge Utente(n_2)) \rightarrow \neg (prenota(n_1, p, ds) \wedge prenota(n_2, p, ds))$ , ovvero un posto può essere prenotato, per uno spettacolo in una certa data, da al più un utente. Per correggere il secondo errore invece va aggiunto il vincolo:  $\forall p, ds, n prenota(n, ds, p) \rightarrow \forall t_p, t_{ds} (postoDi(t_p, t_p) \wedge inCartellone(ds, t_{ds}) \rightarrow t_p = t_{ds})$ .