

INGEGNERIA DEL SOFTWARE A.A. 2012-13
PROVA SCRITTA DEL 19 GIUGNO 2013 (SOLUZIONE)

Esercizio 1

Si supponga di dover implementare un “immaginario e semplificato” **sistema esperto basato su regole** che sia di ausilio ad un medico generico. Un sistema esperto è un programma che cerca di riprodurre le prestazioni di una o più persone esperte in un determinato campo di attività.

I sistemi esperti basati su regole sono dei programmi composti da regole nella forma IF *condizione* THEN *azione*. Dati una serie di fatti, i sistemi esperti, grazie alle regole di cui sono composti, riescono a dedurre nuovi fatti.

Per esempio, supponiamo di avere un problema di salute, forniamo al sistema esperto i seguenti fatti:

- ho mal di testa
- sono raffreddato
- la temperatura corporea è di 38 gradi

il sistema esperto assume i fatti e sceglie una regola così formata:

```
IF ((mal di testa) AND (raffreddore) AND (temperatura = 38)) THEN (Il suo
problema è INFLUENZA)
```

e conclude che il problema è l'influenza.

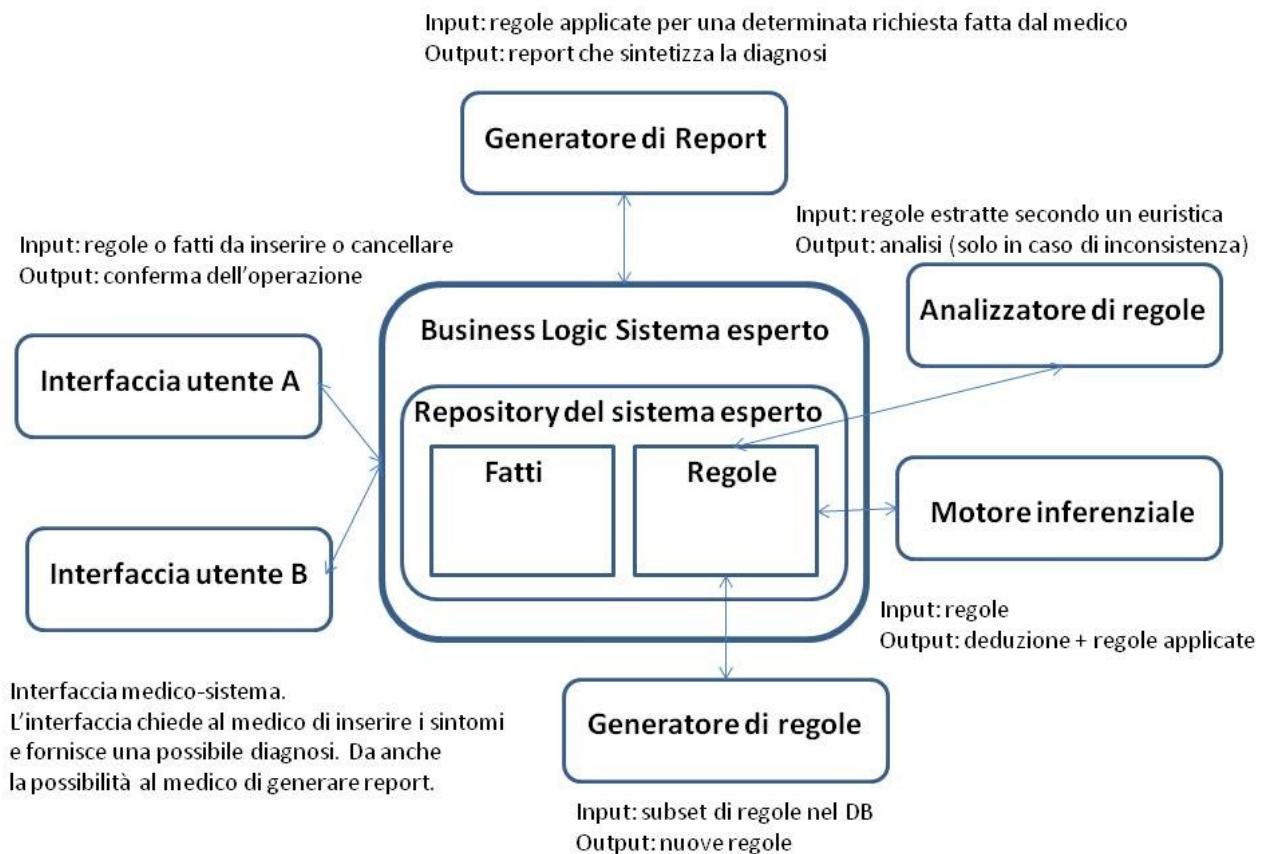
Il sistema esperto che si vuole implementare si compone in sostanza di:

1. una base di dati che contiene i fatti;
 2. una base di conoscenza, in cui sono accumulate le regole (deduttive) di cui il sistema si serve nel suo operato;
 3. un motore inferenziale che si occupa di applicare le regole per fornire la risposta (più regole possono essere applicate per dedurre un unico fatto);
 4. un analizzatore di regole che quando eseguito verifica la consistenza delle regole immagazzinate nella base di conoscenza;
 5. un generatore “intelligente” di regole che a partire dalle regole inserite ne deduce di nuove (che saranno validate in un secondo tempo dall'esperto medico);
 6. un generatore di report, che riassume le regole e i fatti utilizzati per formulare la diagnosi (ovvero dettaglia come è arrivato alla conclusione) ;
 7. un interfaccia utente A, che permette di modificare la base di dati e la base di conoscenza. Cioè permette di inserire e cancellare fatti e regole;
 8. un'interfaccia utente B, che permette interazione fra il medico e il sistema esperto che deve dare risposta ai suoi problemi.
- a) Specificare il nome dello stile architetturale che ritenete più appropriato per il sistema software sopra descritto motivando brevemente la vostra scelta, cioè elencando i pro e i contro (questi ultimi se presenti)
- b) Disegnare un diagramma che descrive l'architettura del sistema in accordo a tale stile, commentando brevemente a parole il funzionamento e marcando (quando utile) input e output dei vari componenti software

Bozza di Soluzione

a) Lo stile architetturale più appropriato è il repository model in quanto consente un efficiente condivisione di dati (nel nostro caso fatti e regole). Questo stile è adatto ad applicazioni dove i dati sono generati da un sottosistema e utilizzati da un altro. Pro: gestione centralizzata di backup e sicurezza. Forse il contro più grande di questo stile è l'evoluzione del modello dei dati perchè se richiesto rende necessario tradurre una grosse mole di dati (per rendere i dati consistenti con il nuovo modello). Inoltre occorre anche modificare "l'interfaccia" tra repository e sottosistemi.

b) Sketch di diagramma ad alto livello:



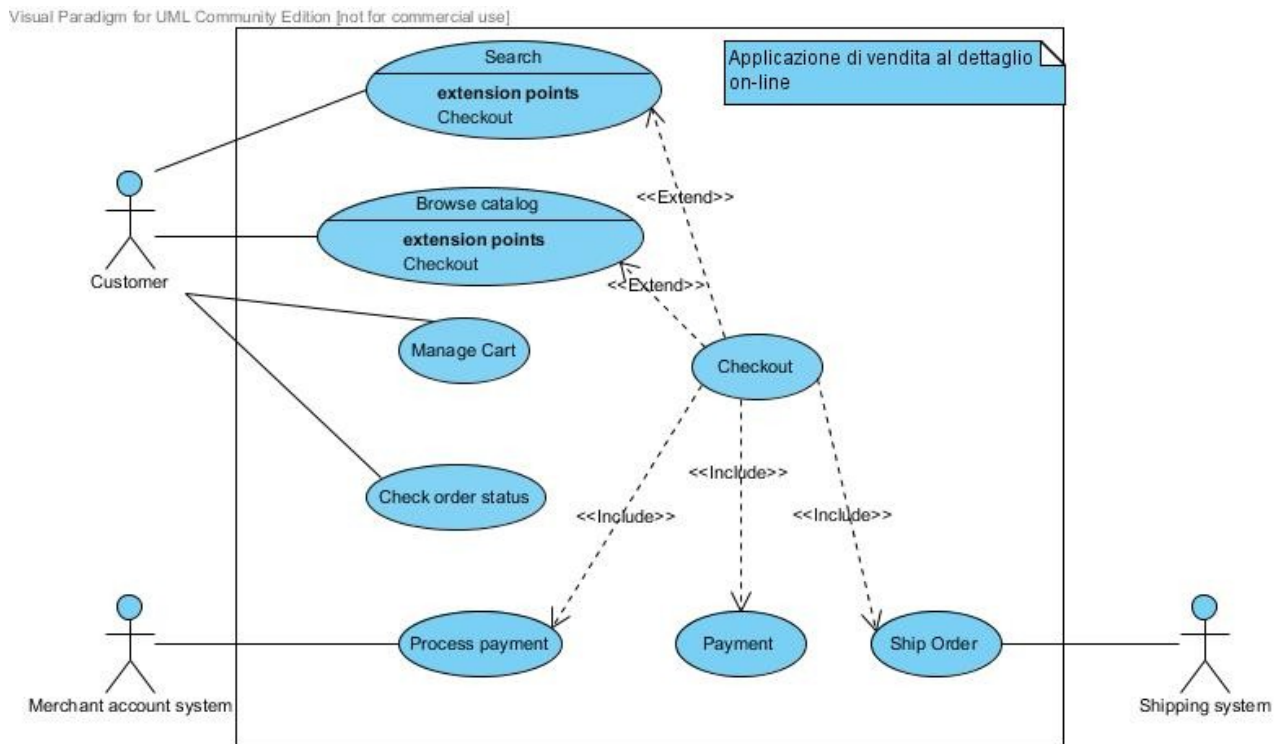
Esercizio 2

Si supponga di dover sviluppare un applicazione Web di vendita al dettaglio on-line. Un cliente può scorrere il catalogo di prodotti (**Browse catalog**) o fare una ricerca (**Search**), acquistare uno o più prodotti all'interno del catalogo (**Checkout**), gestire il carrello dei prodotti (**Manage cart**) o verificare lo stato dell'ordine una volta che questo è stato eseguito (**Check Order status**). La funzionalità di Checkout prevede anche: 1) il pagamento del prodotto (**Payment**) -- solo tramite carta di credito, 2) la trasmissione dell'avvenuta entrata ad un sistema esterno di contabilità (**Merchant account system**) e la richiesta di spedizione dell'ordine ad un altro sistema esterno (**Shipping system**).

- Disegnare il diagramma dei casi d'uso facendo attenzione agli attori primari e secondari
- Rappresentare con un diagramma delle attività il caso d'uso **Checkout**. Si consiglia di usare due swimlane che rappresentano rispettivamente il cliente e l'applicazione Web di vendita al dettaglio
- (facoltativo) Supponendo di aver scelto di implementare il sistema con un architettura three-tier ed aver popolato il primo livello con pagine Web, il secondo con classi di business e il terzo con classi che gestiscono il collegamento con il DBMS rappresentare con un diagramma di sequenza il caso d'uso **Search**.

Bozza di Soluzione

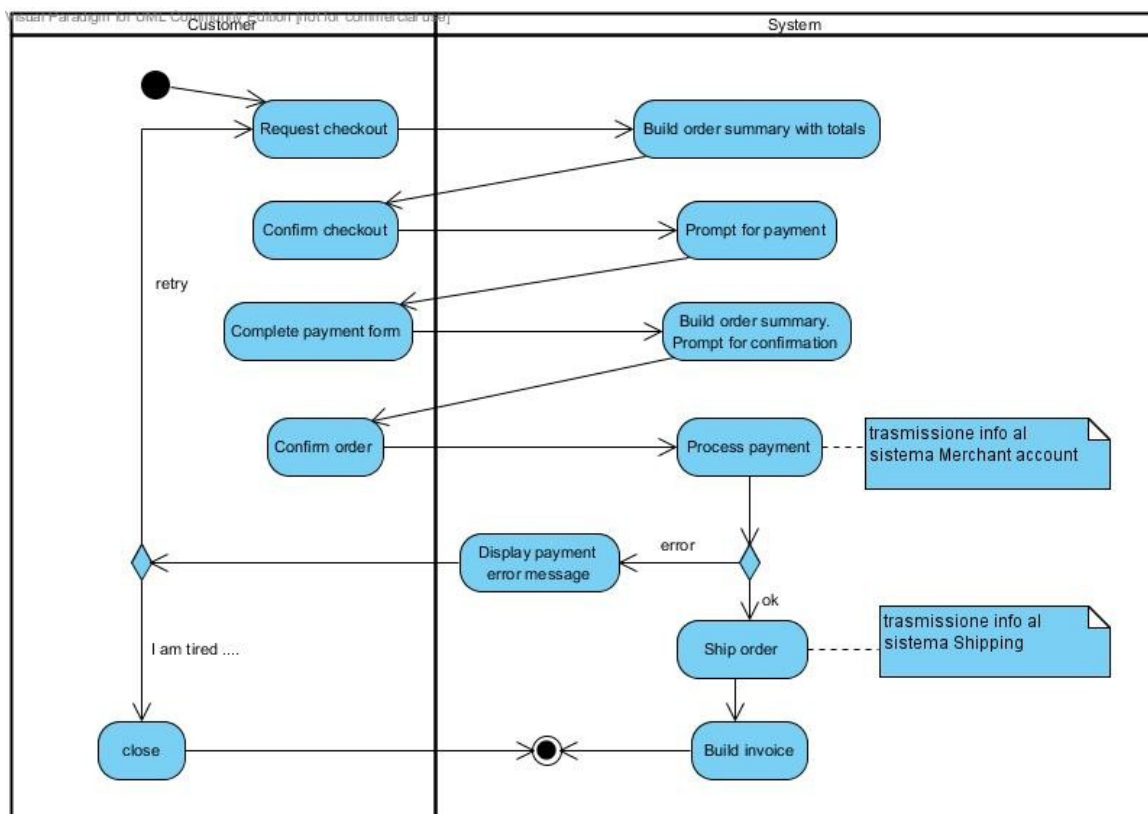
a) use case diagram



Attore primario: Customer

Attori secondari: Merchant account system, Shipping system

b) activity diagram



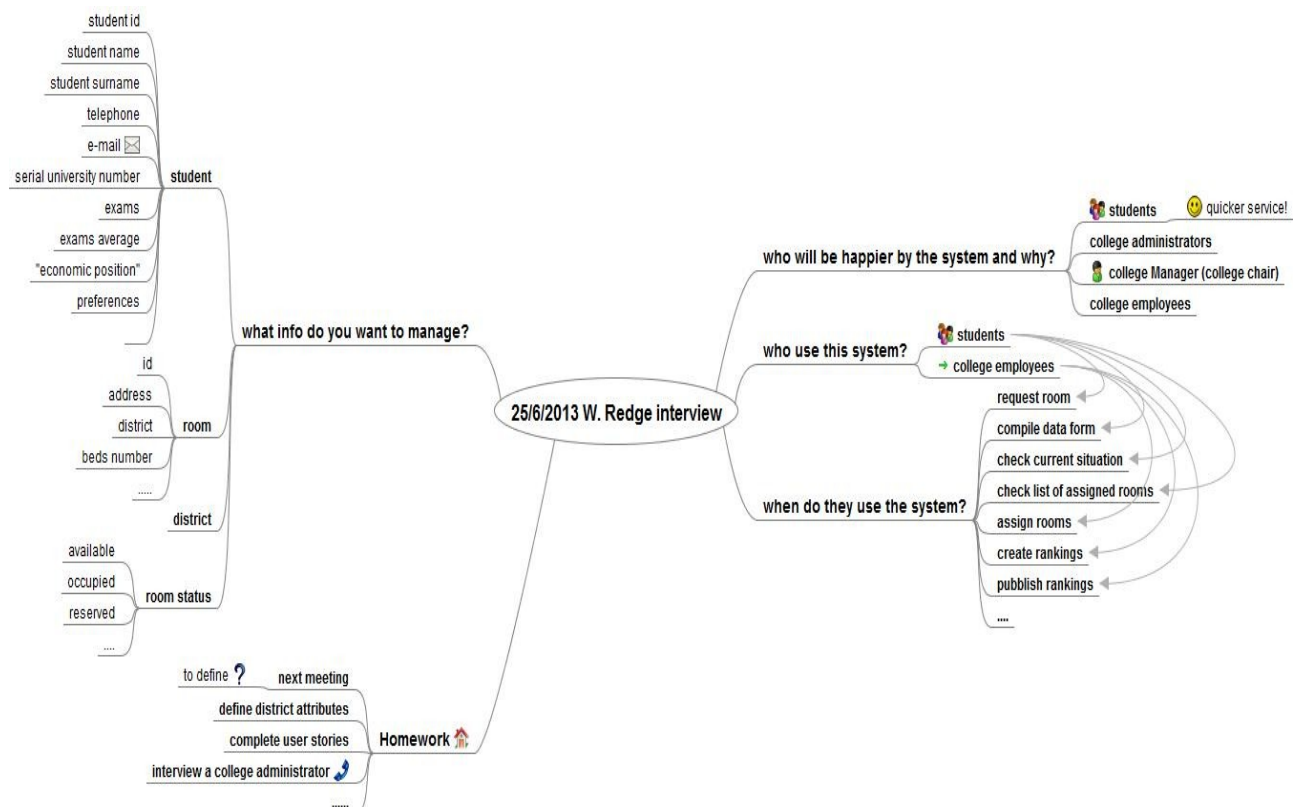
Esercizio 3

Si supponga di dover implementare un sistema software di gestione delle prenotazioni di stanze per un college. Durante le vacanze estive, gli studenti universitari possono prenotare *online* delle stanze all'interno del college. Gli studenti devono specificare il proprio nome, la propria matricola, il corso di laurea, l'anno di corso, tre preferenze riguardo i comprensori selezionabili, la loro posizione economica, il numero e tipo di esami sostenuti e la media complessiva. Il sistema fornisce un posto letto agli studenti che ne hanno fatto richiesta fino ad esaurimento posti, seguendo una graduatoria di merito che si basa sugli esami sostenuti e sul reddito, cercando dove possibile di soddisfare le preferenze fornite.

- Creare un mind-map come visto a lezione che riassume un'ipotetica intervista ad un studente "modello" interessato al sistema
- Scrivere due requisiti funzionali sensati per il sopracitato sistema utilizzando il formato delle **User stories**
- Eseguire una **"formal peer review"** dei seguenti requisiti, evidenziando eventuali problematiche (cioè violazioni alle proprietà dei requisiti viste a lezione) e fornendo possibili soluzioni. *L'applicazione deve essere in grado di funzionare, oltre che su laptop e smartphone di ultima generazione, anche su computer client con schermo da 640x480 pixel monocromatico (b/w). L'interfaccia utente per tutte queste categorie di computer deve essere altamente intuitiva. Lo studente inserirà la prenotazione in una form HTML (method="get") che trasmetterà i dati alla servlet ManageReservation.java avente funzione di controller. Durante l'inserimento di una prenotazione il codice Javascript lato client convaliderà tutti i campi immessi dallo studente mostrando in rosso eventuali errori (esempio tipico di una mail senza il simbolo @). Il sistema fornirà una risposta (numero posto letto e comprensorio nel caso ci sia ancora posto) molto velocemente.*
- (facoltativo) Descrivere in pseudocodice o con un activity diagram la componente del sistema che si occupa di calcolare la graduatoria di merito ed assegnare i posti letto

Bozza di Soluzione

a) mind map



b) User stories

- As a student, I want to check the current situation of the rooms (i.e., the number and district of available rooms)
- As a college employee, when the deadline for the request is passed, I want to create the temporary rankings

c) Formal peer review

- ci sono due requisiti contraddittori: schermo monocromatico (b/w) e mostrando in rosso eventuali errori
- ci sono dei requisiti non testabili: il sistema fornirà una risposta molto velocemente, l'interfaccia deve essere altamente intuitiva
- vengono utilizzati termini tecnici facenti parte dello "spazio della soluzione e non del problema" (cioè si usano termini informatici non adatti a stakeholders non informatici): servlet, form HTML, metodo "get"
- requisiti funzionali e non funzionali sono "mischiati"