

A.A. 2006-2007

ALLIEVI DEL III ANNO IN INGEGNERIA DELL'INFORMAZIONE
(N.O.)

PROGETTO DA PRESENTARE

OBBLIGATORIAMENTE COME PROVA (NON ESCLUSIVA)
D'ESAME DELL'INSEGNAMENTO
INGEGNERIA DEL SOFTWARE A,

OPZIONALMENTE COME INTEGRAZIONE DELL'ESAME
DELL'INSEGNAMENTO
IMPIANTI INFORMATICI

E, OPZIONALMENTE, COME “ALTRA ATTIVITÀ FORMATIVA”
DEL VALORE DI 5 CFU

Si desidera realizzare un'applicazione che consenta la simulazione “accelerata” del comportamento di una rete di Petri.

REQUISITI FUNZIONALI

L'applicazione acquisisce in ingresso la descrizione di una rete di Petri, che può variare da sessione a sessione, e, interagendo con l'operatore, produce in uscita un'evoluzione dinamica simulata (strettamente sequenziale) di suddetta rete.

I LIVELLO DI DIFFICOLTÀ

All'inizio della simulazione la marcatura corrente è quella iniziale. La simulazione procede iterativamente, raggiungendo a ogni iterazione una nuova marcatura, attraverso lo scatto contemporaneo di una o più transizioni abilitate data la marcatura corrente. La marcatura raggiunta (che, incidentalmente, può coincidere con quella corrente) diventa quella corrente, e così via finché l'operatore non ponga fine alla simulazione o non si sia raggiunta una marcatura che evidenzia un blocco critico della rete di Petri.

A ogni iterazione della simulazione, data la marcatura corrente, si possono verificare tre casi:

- 1) non esiste alcuna transizione abilitata: l'applicazione evidenzia la situazione di blocco critico e termina la simulazione;
- 2) esiste una sola transizione abilitata: l'applicazione visualizza tale transizione nonché la marcatura prossima raggiunta attraverso lo scatto della stessa;
- 3) esistono più transizioni abilitate: l'applicazione procede come descritto di seguito.
 - A) Sia S il sottoinsieme delle transizioni abilitate che non sono in conflitto con alcuna transizione abilitata (pertanto tutte le transizioni appartenenti a S sono reciprocamente concorrenti); se S non è vuoto, l'applicazione visualizza le transizioni in esso contenute e chiede all'operatore di selezionare fra esse quelle che vuole fare scattare contemporaneamente (eventualmente nessuna, una o tutte);
 - B) Se non esiste alcuna transizione abilitata non appartenente a S , l'applicazione salta al successivo punto E per terminare l'iterazione; se esistono transizioni abilitate non appartenenti a S , l'applicazione le mostra all'operatore e chiede se desideri fare scattare una di esse;

- C) Se l'operatore risponde negativamente, l'applicazione salta al successivo punto E per terminare l'iterazione, altrimenti consente all'operatore di scegliere una di suddette transizioni;
- D) Sia S_C l'insieme corrente delle transizioni abilitate non appartenenti a S che non sono in conflitto con alcuna delle transizioni di cui l'operatore ha già richiesto di simulare lo scatto. Se S_C non è vuoto, l'applicazione mostra all'operatore tutte le transizioni contenute in S_C e chiede se desideri fare scattare una di esse, quindi torna al punto precedente;
- E) Se l'insieme completo delle transizioni di cui l'operatore ha richiesto lo scatto non è vuoto, l'applicazione visualizza tale insieme e la marcatura raggiunta effettuando contemporaneamente lo scatto di tutte le suddette transizioni a partire dalla marcatura corrente; se l'insieme completo delle transizioni di cui l'operatore ha richiesto lo scatto è vuoto, l'applicazione avverte l'operatore che la marcatura non è cambiata perché non è stata scelta alcuna transizione da fare scattare.

Al termine dell'iterazione condotta secondo i casi 2 e 3, l'applicazione chiede all'operatore se intende o meno continuare la simulazione, ne acquisisce la risposta e agisce di conseguenza.

II LIVELLO DI DIFFICOLTÀ

La rete di Petri di volta in volta considerata può essere (o non essere) temporizzata. Nel caso la rete di Petri sia temporizzata, l'applicazione deve acquisire in ingresso anche gli intervalli temporali relativi a ciascuna transizione (che, a parità di rete di Petri, possono variare da sessione a sessione). Gli estremi di tali intervalli devono essere interi. La simulazione di una rete di Petri temporizzata visualizza iterativamente la marcatura valida a ogni istante intero positivo di tempo, raggiunta a partire dalla marcatura valida all'istante intero precedente. Si assume che all'istante 0 la marcatura corrente sia quella iniziale, qui indicata come $M(0)$. La simulazione procede finché l'operatore non ponga fine alla stessa o non si sia raggiunta una marcatura che evidenzia un blocco critico della rete di Petri. La prima iterazione determina pertanto la marcatura valida all'istante 1, indicata con $M(1)$, raggiunta a partire da $M(0)$ prendendo in considerazione il lasso di tempo $[0,1[$; la seconda iterazione determina la marcatura valida all'istante 2, indicata con $M(2)$, raggiunta a partire da $M(1)$ prendendo in considerazione il lasso di tempo $[1,2[$, e così via.

A ogni iterazione della simulazione, data la marcatura corrente $M(t)$, si possono verificare due casi:

- 1) non esiste alcuna transizione abilitata secondo $M(t)$ (ovvero abilitata in senso convenzionale e il cui istante finale di possibile attivazione è maggiore o uguale a t): l'applicazione evidenzia la situazione di blocco critico e termina la simulazione;
- 2) esistono una o più transizioni abilitate secondo $M(t)$: l'applicazione procede come descritto di seguito. (Si ricordi che l'intervallo considerato dall'iterazione corrente è $[t, t+1[$, pertanto si potrà simulare lo scatto solo di transizioni il cui istante iniziale di possibile attivazione è minore o uguale a t .)

A) Sia S l'insieme delle transizioni abilitate secondo la marcatura $M(t)$, il cui istante iniziale di possibile attivazione è minore o uguale a t , che non sono in conflitto con alcuna transizione abilitata (pertanto tutte le transizioni appartenenti a S sono reciprocamente concorrenti). Sia S' il sottoinsieme di S contenente tutte e sole le transizioni il cui istante finale di possibile attivazione è minore o uguale a $t+1$. Se l'insieme $S \setminus S'$ non è vuoto, l'applicazione visualizza le transizioni in esso contenute e chiede all'operatore di selezionare fra esse quelle che vuole fare scattare contemporaneamente (eventualmente nessuna, una o tutte);

- B) Sia S_C l'insieme corrente delle transizioni abilitate secondo $M(t)$, il cui istante iniziale di possibile attivazione è minore o uguale a t , che non sono appartenenti a S , che non sono in conflitto con alcuna delle transizioni di cui l'operatore ha già richiesto di simulare lo scatto. Se S_C è vuoto, l'applicazione salta al successivo punto D; se S_C non è vuoto, l'applicazione mostra le transizioni in esso contenute all'operatore e chiede se desideri fare scattare una di esse;
- C) Se l'operatore risponde affermativamente, l'applicazione consente allo stesso di scegliere una di suddette transizioni, quindi ritorna al punto precedente;
- D) Sia S_{c1} l'insieme corrente delle transizioni abilitate secondo $M(t)$, non appartenenti a S , il cui istante iniziale di possibile attivazione è minore o uguale a t , il cui istante finale di possibile attivazione è minore o uguale a $t+1$, di cui l'operatore non ha già richiesto di simulare lo scatto e che non sono in conflitto con alcuna transizione di cui l'operatore ha già richiesto di simulare lo scatto. Se S_{c1} non è vuoto, l'applicazione mostra all'operatore tutte le transizioni in esso contenute affinché ne scelta (obbligatoriamente) una da fare scattare e torna a eseguire il passo D;

- E) Sia T l'insieme completo delle transizioni di cui l'operatore ha richiesto lo scatto, unito all'insieme S' . Se T non è vuoto, l'applicazione visualizza sia il contenuto di T , sia la marcatura $M(t+1)$ raggiunta effettuando contemporaneamente lo scatto di tutte le transizioni appartenenti a T a partire dalla marcatura $M(t)$. (Se T è vuoto, necessariamente $M(t+1)$ è identica a $M(t)$.)

Al termine dell'iterazione condotta secondo il caso 2, l'applicazione chiede all'operatore se intende o meno continuare la simulazione, ne acquisisce la risposta e agisce di conseguenza.

REQUISITI NON FUNZIONALI

La forma di ingresso della descrizione della rete di Petri (temporizzata o meno) può essere interattiva o batch (o, eventualmente, entrambe le forme possono essere supportate). Non è richiesta la creazione di una interfaccia grafica per le sessioni interattive.

NOTA

I requisiti (funzionali e non) di cui sopra sono deliberatamente espressi a un alto livello di astrazione (ad esempio, non si sono imposti limiti alle dimensioni delle reti di Petri considerate né al numero di iterazioni della simulazione) al fine di consentire agli ingegneri del software di fornire un'interpretazione personale, che comporta sempre l'aggiunta di ulteriori requisiti. Inoltre, al punto D del secondo livello di difficoltà, nel caso limite S_{c1} contenga una singola transizione, l'operatore è tenuto a selezionare la stessa: questa selezione, essendo obbligatoria, potrebbe essere automatizzata.

Ingegneria del Software A

Impianti Informatici

Agli studenti è richiesto di realizzare in linguaggio Java un'applicazione software che soddisfi i requisiti sopra esposti, sino a coprire almeno il primo livello di difficoltà.

La realizzazione dell'applicazione secondo un'architettura stand alone, è adeguata per la presentazione dell'elaborato ai fini del superamento dell'esame di *Ingegneria del software A*.

La realizzazione dell'applicazione secondo un'architettura client-server è richiesta quando si intenda presentare l'elaborato non solo ai fini del superamento dell'esame di *Ingegneria del software A* ma anche come integrazione di quello di *Impianti informatici*.

Attività formativa

La realizzazione in linguaggio Java di un'applicazione che adotti un'architettura client-server e soddisfi tutti i requisiti sopra esposti sino a coprire il secondo livello di difficoltà è invece necessaria al fine dell'accreditamento di ulteriori 5 CFU (voce “Laboratorio/progetto/tirocinio o altre attività formative volte ad una migliore conoscenza del mondo del lavoro” del Manifesto degli Studi).

Richieste

Ogni gruppo dovrà:

- 1) indicare il modello di processo adottato;
- 2) presentare in formato cartaceo la documentazione di progetto, comprendente
 - casi d'uso (comprensivi dell'espressione dei requisiti aggiuntivi),
 - diagramma UML delle classi,
 - diagrammi UML dinamici (opzionali),
 - diagramma UML di deployment (se l'architettura è client-server),e qualsiasi altra specifica ritenuta opportuna;
- 3) presentare in formato cartaceo un breve manuale d'uso;
- 4) consegnare codice sorgente + codice interpretabile + (preferibilmente) codice eseguibile;
- 5) preparare ed effettuare un'intera dimostrazione.