

A.A. 2007-2008
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 del comportamento di una rete di Petri le cui transizioni sono dotate di priorità intere positive (dove, fra un insieme di valori di priorità, la priorità più elevata – detta priorità massima – è quella di valore numerico minimo).

REQUISITI FUNZIONALI

L'applicazione acquisisce in ingresso la descrizione di una rete di Petri pura, che può variare da sessione a sessione, e i valori delle priorità delle transizioni della stessa, che, a parità di rete di Petri pura, possono variare da sessione a sessione; quindi, 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 di una transizione che è abilitata data la marcatura corrente e la cui priorità è la più elevata fra quelle di tutte le transizioni abilitate. 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.

Più dettagliatamente, a ogni iterazione della simulazione, data la marcatura corrente, si possono verificare i tre casi sottoelencati.

- 1) Non esiste alcuna transizione abilitata: l'applicazione evidenzia la situazione di blocco critico e termina la simulazione.
- 2) Esiste una sola transizione abilitata dotata di priorità massima: l'applicazione visualizza tale transizione nonché la marcatura prossima raggiunta attraverso lo scatto della stessa.
- 3) Esistono più transizioni abilitate dotate della medesima priorità massima: l'applicazione chiede all'utente di sceglierne una e visualizza la marcatura prossima raggiunta attraverso lo scatto della stessa.

Al termine dell'iterazione condotta secondo i casi 2 e 3, la marcatura raggiunta diventa quella corrente; l'applicazione chiede all'operatore se intende continuare la simulazione, ne acquisisce la risposta e agisce di conseguenza.

II LIVELLO DI DIFFICOLTÀ

La rete di Petri di volta in volta considerata, oltre a essere dotata di priorità, 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 possono variare da sessione a sessione). Gli estremi di tali intervalli o sono entrambi numeri naturali finiti (eventualmente uguali) oppure l'estremo inferiore è un numero naturale finito e l'estremo superiore assume un valore infinito.

All'istante iniziale, convenzionalmente istante 0, la marcatura corrente è quella iniziale, indicata come $M(0)$. L'applicazione opera di volta in volta su un intervallo $]t, t+1]$, consentendo all'operatore di scegliere, una per una, le transizioni che desidera far scattare nell'intervallo $]t, t+1[$ e nell'istante $t+1$, rispettivamente. Quando tutti questi scatti sono stati simulati, l'attenzione si concentra sull'intervallo di lunghezza unitaria successivo (ovvero dapprima si considera l'intervallo $]0, 1]$, poi l'intervallo $]1, 2]$, poi l'intervallo $]2, 3]$, e così via). 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.

Più dettagliatamente, la simulazione deve produrre lo stesso effetto operativo di un processo ciclico, dove il corpo del ciclo (il cui inizio è indicato dall'etichetta CICLO) prende in considerazione un intervallo $[t, t+1]$, dove t è l'istante corrente prima della prima iterazione relativa a tale intervallo. Il medesimo intervallo può essere considerato per più iterazioni successive, come descritto di seguito, al termine dell'ultima delle quali l'istante corrente è $t+1$. Si noti che, secondo la semantica delle reti di Petri temporizzate, relativamente a tale intervallo è simulabile lo scatto solo di transizioni il cui istante iniziale di possibile attivazione è minore o uguale a $t+1$. Prima della prima iterazione il valore dell'istante corrente t è 0 e la marcatura corrente è $M(0)$.

CICLO - Data la marcatura corrente, si possono verificare i tre casi sotto riportati.

- 1) Non esiste alcuna transizione abilitata nel senso convenzionale delle reti di Petri pure: l'applicazione evidenzia la situazione di blocco critico e termina la simulazione.
- 2) Esistono transizioni abilitate nel senso convenzionale delle reti di Petri pure ma nessuna di esse è abilitata nell'intervallo $[t, t+1]$: l'applicazione attribuisce a $M(t+1)$ il valore della marcatura corrente, $M(t+1)$ diventa la marcatura corrente, $t+1$ diventa l'istante corrente e l'esecuzione riprende da CICLO, considerando l'intervallo di lunghezza unitaria successivo, la cui marcatura di partenza è appunto $M(t+1)$.

3) Esistono una o più transizioni abilitate nell'intervallo $]t, t+1]$ (ovvero abilitate nel senso convenzionale delle reti di Petri pure e i cui istanti iniziale e finale di possibile attivazione sono rispettivamente minore o uguale a $t+1$ e maggiore di t).

a) Se non esistono transizioni abilitate nell'intervallo $]t, t+1[$, l'applicazione continua come descritto nel successivo punto 3b.

Se esistono una o più transizioni abilitate nell'intervallo $]t, t+1[$ dotate della medesima priorità massima, l'applicazione le visualizza e chiede all'operatore di selezionarne una. L'applicazione simula lo scatto della transizione scelta dall'operatore, mostra la marcatura raggiunta $M(t')$, che diventa la nuova marcatura corrente, dove il nuovo (indefinito) istante corrente t' cade nell'intervallo $]t, t+1[$, chiede all'operatore se intende continuare la simulazione, ne acquisisce la risposta e agisce di conseguenza.

All'operatore che ha manifestato l'intenzione di continuare la simulazione, l'applicazione chiede se desidera simulare il compimento dell'istante $t+1$. In caso di risposta negativa dell'operatore, l'esecuzione ciclica continua a considerare l'intervallo $]t, t+1]$ (riprendendo da CICLO); in caso di risposta affermativa, l'applicazione continua come descritto nel successivo punto 3b.

b) L'applicazione determina l'insieme S delle transizioni abilitate data la marcatura corrente, dotate dalla medesima priorità massima m , che dovrebbero scattare entro l'istante $t+1$ (cioè il cui estremo superiore dell'intervallo temporale è $t+1$). Se S non è vuoto, chiede all'operatore quali desideri fare scattare tutte insieme all'istante $t+1$. Si noti che, poiché la scelta di una transizione da far scattare può disabilitare altre transizioni (che erano in conflitto con quella selezionata), l'ordine di selezione è pregnante e quindi la scelta deve essere condotta iterativamente, cosicché la prossima selezione da parte dell'operatore avvenga entro un insieme di transizioni che non contiene né quelle già scelte né quelle disabilitate dallo scatto di queste ultime. Le iterazioni continuano fino a quando non esistono più transizioni abilitate di priorità m che devono scattare entro l'istante $t+1$. La marcatura $M(t+1)$, raggiunta attraverso lo scatto di tutte le transizioni selezionate dall'operatore all'interno del punto corrente, viene visualizzata.

Se invece S è vuoto, l'applicazione attribuisce a $M(t+1)$ il valore della marcatura corrente.

Al termine di questo punto 3b, $M(t+1)$ è la nuova marcatura corrente, $t+1$ diventa l'istante corrente e l'esecuzione riprende da CICLO, prendendo in considerazione l'intervallo di lunghezza unitaria successivo, la cui marcatura di partenza è appunto $M(t+1)$.

REQUISITI NON FUNZIONALI

La modalità 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.

Requisito non prescrittivo ma importante in sede di valutazione è l'impiego di precondizioni, postcondizioni e invarianti di classe entro il codice Java.

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 passi di 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 3b del secondo livello di difficoltà, nel caso tutte le transizioni abilitate di priorità massima che dovrebbero scattare entro l'istante $t+1$ siano reciprocamente concorrenti, l'operatore è tenuto a selezionarle tutte (una per una): questa selezione globale, 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 (costituito al più da quattro persone) dovrà:

- 1) indicare (e giustificare) in forma scritta il modello di processo adottato;
- 2) produrre la documentazione di progetto, comprendente
 - casi d'uso (comprensivi dell'espressione dei requisiti aggiuntivi), sia in forma testuale, sia in forma di diagramma UML,
 - 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) redigere un breve manuale d'uso;
- 4) presentare in formato sia cartaceo, sia elettronico quanto richiesto ai precedenti punti da 1 a 3;
- 5) consegnare codice sorgente + codice interpretabile + (preferibilmente) codice eseguibile;
- 6) preparare ed effettuare un'intera dimostrazione.