

A.A. 2008-2009

ALLIEVI DEL III ANNO IN INGEGNERIA DELL'INFORMAZIONE

PROGETTO DA PRESENTARE

OBBLIGATORIAMENTE COME PROVA (NON ESCLUSIVA)

D'ESAME DELL'INSEGNAMENTO  
**INGEGNERIA DEL SOFTWARE A,**

OPZIONALMENTE COME INTEGRAZIONE

D'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 un sistema modellato mediante due macchine a stati finiti (FSM) *cooperanti*.

## DOMINIO APPLICATIVO

*Macchina a stati finiti.* Una FSM è costituita da un insieme finito di stati, uno dei quali è identificato come *stato corrente*, e da un insieme di transizioni di stato, ciascuna delle quali collega uno stato con un altro, dove stato sorgente e stato destinazione della transizione possono essere coincidenti. Non esistono due transizioni distinte che collegano la stessa coppia ordinata di stati sorgente e destinazione. Non tutte le coppie di stati sono necessariamente collegate da transizioni.

Una transizione può scattare solo se ha come sorgente lo stato corrente della FSM. Lo *scatto* di una transizione determina un passaggio di stato: a scatto concluso, il nuovo stato corrente è lo stato destinazione della transizione stessa.

*Relazioni fra FSM cooperanti.* Una coppia di transizioni afferenti rispettivamente a due generiche FSM cooperanti è costituita da transizioni reciprocamente *sincrone*, oppure *asincrone*, oppure *mutuamente esclusive*. Non esistono ulteriori relazioni reciproche fra due transizioni di FSM cooperanti distinte.

*Abilitazione delle transizioni.* Date due transizioni uscenti dallo stato corrente di due FSM cooperanti distinte (che, incidentalmente, possono essere identiche),

- se le due transizioni sono sincrone, lo scatto dell'una comporta il simultaneo scatto dell'altra; lo scatto di due transizioni sincrone non può iniziare se non simultaneamente;
- se le due transizioni sono asincrone, lo scatto dell'una non comporta necessariamente il simultaneo scatto dell'altra, pertanto ciascuna transizione può scattare singolarmente oppure entrambe possono scattare simultaneamente;
- se le due transizioni sono mutuamente esclusive, lo scatto dell'una inibisce per tutta la sua durata lo scatto dell'altra; lo scatto di due transizioni mutuamente esclusive non può mai temporalmente sovrapporsi, neppure parzialmente.

Pertanto, dati gli stati correnti di due FSM cooperanti, sono abilitate a scattare:

- tutte le singole transizioni asincrone uscenti da tali stati (che non siano sincrone rispetto ad altre uscenti dai medesimi stati),
- tutte le singole transizioni mutuamente esclusive uscenti da tali stati (che non siano sincrone rispetto ad altre uscenti dai medesimi stati),
- tutte le coppie di transizioni reciprocamente sincrone uscenti da tali stati,
- tutte le coppie di transizioni reciprocamente asincrone uscenti da tali stati.

Se, dati gli stati correnti di due FSM cooperanti, più transizioni singole e/o coppie di transizioni risultano abilitate, la scelta dell'evoluzione da seguire è non-deterministica.

## REQUISITI FUNZIONALI

All'inizio di ogni sessione di simulazione, l'utente dell'applicazione le fornisce in ingresso:

- la descrizione di due FSM,
- le relazioni che intercorrono fra le transizioni delle due FSM cooperanti di cui sopra,
- lo stato iniziale di ciascuna delle due FSM.

A ogni passo di simulazione, l'applicazione produce in uscita gli stati via via raggiunti dalle due FSM durante l'evoluzione dinamica simulata.

Sessioni diverse possono prendere in considerazione diversi sistemi e/o stati iniziali.

## I LIVELLO DI DIFFICOLTÀ

All'inizio della simulazione gli stati correnti delle due macchine sono quelli iniziali. La simulazione procede iterativamente, raggiungendo a ogni iterazione una nuova coppia di stati, attraverso lo scatto di una transizione singola o di due transizioni simultanee che sono abilitate dati gli stati correnti. Gli stati raggiunti (che, incidentalmente, possono coincidere con quelli correnti) diventano quelli correnti, e così via finché l'utente non ponga fine alla simulazione o non si sia raggiunta una coppia di stati a partire dai quali nessuna transizione sia abilitata.

Più dettagliatamente, a ogni iterazione della simulazione, dati gli stati correnti delle due macchine, si possono verificare i tre casi sottoelencati.

- 1) Non esiste alcuna transizione abilitata: l'applicazione evidenzia la situazione di blocco e termina la simulazione.
- 2) Esiste un solo possibile passaggio di stato, che può avvenire per opera di una transizione singola (coinvolgendo pertanto una singola macchina) o di due transizioni sincrone (coinvolgendo entrambe le macchine): l'applicazione visualizza tale transizione singola o tale coppia di transizioni sincrone nonché gli stati prossimi raggiunti attraverso il relativo scatto.
- 3) Esistono più passaggi di stato possibili: l'applicazione visualizza tutte le transizioni singole abilitate e tutte le coppie di transizioni simultanee abilitate, chiede all'utente di scegliere una fra le transizioni singole o le coppie di transizioni abilitate e visualizza gli stati prossimi raggiunti attraverso il relativo scatto .

Al termine dell'iterazione condotta secondo i casi 2 e 3, gli stati prossimi raggiunti diventano quelli correnti; l'applicazione chiede all'utente se intende continuare la simulazione, ne acquisisce la risposta e agisce di conseguenza.

## II LIVELLO DI DIFFICOLTÀ

Lo scatto di ciascuna transizione delle due FSM cooperanti è dotato di una durata, che deve essere acquisita in ingresso dall'applicazione sotto forma di valore intero e può variare da sessione a sessione. Sussiste il vincolo che ogni coppia di transizioni reciprocamente sincrone abbiano la stessa durata. (Si noti che il livello di difficoltà precedente si ottiene assumendo che anche ogni coppia di transizioni reciprocamente asincrone abbiano la stessa durata. Tale assunzione aggiuntiva rende inutile tenere conto della durata delle transizioni al fine di simulare l'evoluzione del sistema modellato.)

All'inizio della simulazione gli stati correnti delle due macchine sono quelli iniziali. La simulazione procede iterativamente, come descritto di seguito facendo ricorso alle etichette ITERAZIONE e CONTINUA.

- Se, dati tali stati, non ci sono transizioni abilitate, l'applicazione avverte l'utente e termina la simulazione.
- Se invece esiste una sola transizione singola o una sola coppia di transizioni sincrone abilitate dati gli stati correnti delle due FSM, l'applicazione visualizza tale transizione singola o tale coppia di transizioni sincrone nonché gli stati prossimi raggiunti attraverso il relativo scatto, che diventano i nuovi stati correnti. L'elaborazione continua dall'etichetta CONTINUA.
- Altrimenti l'applicazione chiede all'utente di selezionare una transizione singola o una coppia di transizioni tra quelle abilitate dati gli stati iniziali. L'elaborazione prosegue dall'etichetta ITERAZIONE.

(ITERAZIONE) Ogni iterazione rappresenta il trascorrere di un lasso di tempo pari alla durata minore fra le durate necessarie al completamento dello scatto delle transizioni considerate. Ciò significa che:

- se si sta considerando una transizione singola, essa termina il suo scatto nell'iterazione corrente, raggiungendo un nuovo stato della FSM a cui la transizione appartiene, che diviene lo stato corrente di tale FSM;
- se invece si stanno considerando due transizioni sincrone o (asincrone) richiedenti la medesima durata per giungere a completamento, entrambe le transizioni terminano il loro scatto nell'iterazione corrente, raggiungendo una nuova coppia di stati, che diventano gli stati correnti delle due FSM;
- se, infine, si stanno considerando due transizioni asincrone richiedenti durate diverse per giungere a completamento, termina il suo scatto nell'iterazione corrente la sola transizione richiedente la durata minore, dove lo stato raggiunto dalla FSM a cui tale transizione appartiene diviene lo stato corrente della FSM stessa; l'altra transizione rimane *in atto*, cioè il completamento del suo scatto richiede ancora una durata residua (che sarà pari alla durata che la transizione aveva all'inizio dell'iterazione decurtata della durata della transizione di cui è avvenuto lo scatto nell'iterazione stessa).

(CONTINUA) L'applicazione chiede all'utente se intende continuare la simulazione. In caso di risposta negativa, la simulazione termina. In caso di risposta affermativa, l'applicazione mostra all'utente tutte le transizioni con cui può avvenire la prossima evoluzione del sistema modellato. Si possono verificare i casi elencati di seguito.

a) Non ci sono transizioni in atto e non esistono transizioni abilitate: l'applicazione avverte l'utente e termina la simulazione.

b) Non ci sono transizioni in atto ed esistono più transizioni abilitate dati gli stati correnti delle due FSM: l'utente sceglie una transizione singola o una coppia di transizioni fra quelle abilitate, dando così inizio a una nuova iterazione (l'elaborazione salta cioè all'etichetta ITERAZIONE).

c) Non ci sono transizioni in atto ed esiste una sola transizione singola o una sola coppia di transizioni sincrone abilitate dati gli stati correnti delle due FSM: l'applicazione visualizza tale transizione singola o tale coppia di transizioni sincrone nonché gli stati prossimi raggiunti attraverso il relativo scatto, che diventano i nuovi stati correnti. L'elaborazione continua (dall'etichetta CONTINUA).

d) In una delle due FSM una transizione è ancora in atto e nell'altra FSM non esistono transizioni abilitate, dato lo stato corrente, che siano reciprocamente asincrone rispetto a quella in atto: l'applicazione visualizza lo stato prossimo raggiunto attraverso il completamento della transizione in atto e tale stato diviene quello corrente della FSM a cui appartiene. L'elaborazione continua (dall'etichetta CONTINUA).

e) In una delle due FSM una transizione è ancora in atto e nella seconda FSM esistono una o più transizioni abilitate, dato lo stato corrente, che sono reciprocamente asincrone rispetto a quella in atto: la scelta dell'utente riguarderà solo la seconda FSM.

Se sceglie una transizione, l'utente dà inizio a una nuova iterazione (l'elaborazione salta cioè all'etichetta ITERAZIONE).

Se l'utente sceglie invece di non fare evolvere la seconda FSM, l'applicazione visualizza lo stato prossimo raggiunto attraverso il completamento della transizione in atto e tale stato diviene quello corrente della FSM a cui appartiene. L'elaborazione continua (dall'etichetta CONTINUA).

## REQUISITI NON FUNZIONALI

Il linguaggio di programmazione da adottare è Java.

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

La modalità di introduzione delle informazioni d'ingresso 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.

Tutte le informazioni d'ingresso sono passibili di una rappresentazione grafica. Come noto, una FSM può essere rappresentata sotto forma di grafo orientato, dove lo stato iniziale può essere opportunamente evidenziato (ad esempio, mediante una freccia entrante nello stesso, non proveniente da alcun altro stato).

Le relazioni fra transizioni di FSM cooperanti sono rappresentabili sotto forma di grafi non orientati. Dal momento che esistono tre tipi di relazioni (antiriflessive e simmetriche) che globalmente coinvolgono tutte le possibili coppie distinte di transizioni ciascuna delle quali è relativa a un FSM distinta, è sufficiente rappresentare due relazioni. Ciascuna delle due relazioni può essere rappresentata mediante un grafo non

orientato distinto (che può essere disconnesso). In alternativa, un grafo non orientato unico (che può essere disconnesso), dotato di due tipi di archi, può rappresentare entrambe le suddette relazioni.

La visualizzazione delle rappresentazioni grafiche sopra definite non è un requisito.

Ogni stato corrente raggiunto da una FSM durante la simulazione è passibile di una rappresentazione grafica, ad esempio, mediante un marcatore posto all'interno del nodo del grafo corrispondente a tale stato.

La visualizzazione di questa rappresentazione grafica non è un requisito.

## 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 FSM 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.

## **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 comportamentali (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.