

A.A. 2005-2006

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 un sistema reattivo descritto mediante due macchine a stati finiti (FSM) comunicanti, in modo unidirezionale, attraverso un buffer.

## DOMINIO APPLICATIVO

*Macchina a stati finiti.* Nell'ambito di questo elaborato ciascuna *transizione* di stato di ciascuna FSM è fatta scattare da un *evento*. Per entrambe le FSM tale evento può provenire dal mondo esterno. Per la FSM collegata all'uscita del buffer, tale evento può essere anche estratto dal buffer. Ciascuna transizione di stato della FSM collegata all'ingresso del buffer può infatti produrre un evento in uscita, che viene scritto nel buffer, al fine di essere consumato dall'altra FSM.

*Evento.* È un valore simbolico appartenente a un insieme finito. L'insieme degli eventi provenienti dal mondo esterno (*eventi esterni*) è disgiunto rispetto a quello degli eventi transitanti nel buffer (*eventi interni*). Un evento si ritiene *consumato* dopo che ha fatto scattare una (singola) transizione di una FSM. Non esistono eventi interni che non possono fare scattare alcuna transizione della FSM collegata all'uscita del buffer.

*Buffer.* È un registro, in grado di memorizzare al più due eventi (interni). Da una simulazione all'altra può variare

- la sua politica di gestione (ad esempio, può essere FIFO o LIFO),
- il trattamento di un tentativo di scrittura nel caso di buffer pieno (ad esempio, il dato che si voleva scrivere può venire perso, oppure può sovrascrivere quello attualmente contenuto nella prima o nella seconda posizione).

## REQUISITI FUNZIONALI

All'inizio di ogni sessione di simulazione, l'utente dell'applicazione le fornisce in ingresso, in forma interattiva o batch (o, eventualmente, entrambe le forme di ingresso possono essere supportate):

- la descrizione di ciascuna delle due FSM e del buffer,
- lo stato iniziale del sistema da simulare,
- una sequenza di eventi esterni.

L'applicazione produce in uscita gli stati globali via via raggiunti dal sistema reattivo durante l'evoluzione dinamica simulata, che è strettamente sequenziale.

Sessioni diverse possono prendere in considerazione diversi sistemi reattivi e/o stati iniziali e/o sequenze di eventi esterni.

## I LIVELLO DI DIFFICOLTÀ

Ciascuna delle due FSM è deterministica (cioè non esistono più transizioni uscenti dallo stesso stato che sono fatte scattare dallo stesso evento, indipendentemente dallo stato di destinazione e dagli eventi di uscita delle stesse), il buffer è inizialmente vuoto e l'insieme degli eventi esterni a cui ciascuna FSM reagisce è disgiunto rispetto a quello dell'altra.

Nell'evoluzione simulata, dopo avere prodotto la sequenza di stati globali raggiungibili attraverso la successione di transizioni innescata dal primo evento esterno (appartenente alla sequenza) ed essere pervenuti in uno stato globale in cui nessuna transizione (delle due FSM) alimentata da un evento interno può più scattare (perché non esiste alcun evento interno che debba essere consumato, o perché non esiste alcuna transizione uscente dallo stato globale corrente che sia alimentata dal prossimo evento interno da estrarre dal buffer, o perché non esiste alcuna transizione uscente dallo stato globale corrente che sia alimentata da eventi interni), si considera la reazione del sistema all'evento esterno successivo, e così via finché si sono esauriti tutti gli eventi provenienti dal mondo esterno.

La simulazione termina però prima di avere esaurito gli eventi esterni nel caso in cui il prossimo evento esterno da considerare non faccia scattare alcuna transizione a partire dallo stato corrente del sistema reattivo. Questa evenienza deve essere esplicitata nell'uscita dell'applicazione.

## II LIVELLO DI DIFFICOLTÀ

Ciascuna delle due FSM può essere non-deterministica, il buffer può non essere inizialmente vuoto e possono esistere eventi esterni a cui entrambe le FSM sono sensibili.

All'inizio della simulazione, si considera la reazione del sistema al primo evento esterno solo dopo avere consumato tutti gli eventi interni consumabili presenti nel buffer. Nell'evoluzione simulata, a ogni passo in cui siano presenti più transizioni che possono scattare, si delega all'utente la scelta non-deterministica di una di esse.

Le condizioni di terminazione della simulazione sono le stesse del livello di difficoltà precedente.

## III LIVELLO DI DIFFICOLTÀ

Livello di difficoltà precedente + possibilità di produrre in uscita tutte (e sole) le evoluzioni compatibili con l'intera sequenza di eventi esterni assegnata, senza interagire con l'utente.

## REQUISITI NON FUNZIONALI

Il meccanismo di simulazione non deve ricorrere alla creazione della macchina a stati finiti che rappresenta il modello globale del comportamento del sistema descritto.

### 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 due FSM considerate) al fine di consentire agli ingegneri del software di fornire un'interpretazione personale, che comporta sempre l'aggiunta di ulteriori requisiti. Questa intenzione del committente è valida in un contesto di gara d'appalto, dove il sistema sviluppato può fungere da prototipo dimostrativo delle specifiche che ciascun partecipante alla gara intende soddisfare.

## **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 l'ultimo 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.