

A.A. 2020-2021
ALLIEVI DEL III ANNO IN INGEGNERIA INFORMATICA

PROGETTO DA PRESENTARE
OBBLIGATORIAMENTE
PER LA PROVA ORALE D'ESAME DELL'INSEGNAMENTO
INGEGNERIA DEL SOFTWARE (9 CFU)

N.B. Una opportuna ulteriore attività proposta dai docenti, tesa a estendere o approfondire il progetto (prima e seconda parte) realizzato nell'ambito dell'insegnamento di Ingegneria del Software e svolta autonomamente dal singolo studente, con produzione di un elaborato individuale, può essere l'oggetto della PROVA FINALE (3 CFU) per il conseguimento della LAUREA IN INGEGNERIA INFORMATICA.

TEMA

Si desidera realizzare, secondo un processo di sviluppo incrementale/iterativo, un sistema software che consenta la simulazione del comportamento di reti di Petri le cui transizioni siano eventualmente dotate di priorità.

L'applicazione prevede due tipologie di utente, il *configuratore* e il *fruitore*. Il primo è deputato a descrivere le reti di Petri, il secondo può scegliere una rete di Petri fra quelle disponibili ed effettuare delle simulazioni relative alla stessa. Il configuratore potrebbe, per esempio, essere un insegnante che, al fine di fare acquisire ai suoi allievi dimestichezza con il concetto di rete di Petri, fornisce agli stessi delle descrizioni di alcuni esemplari di tali reti. Nel caso esemplificato, ciascun allievo è un fruitore.

VERSIONE 1 – REQUISITI FUNZIONALI

La prima versione dell'applicazione interagisce col solo configuratore al fine di acquisire la descrizione di reti (N) - non ancora reti di Petri - ciascuna delle quali deve contenere almeno un posto e almeno una transizione. Ogni singola N deve essere sintatticamente corretta, ovvero tutti i posti e tutte le transizioni devono essere connessi, ciascun elemento della relazione di flusso deve essere una coppia ordinata (posto, transizione) o (transizione, posto) e non devono esistere elementi della relazione di flusso ripetuti (in altre parole, ogni elemento della relazione di flusso è costituito da una coppia ordinata distinta). Al fine di garantire che ogni N sia sintatticamente corretta, l'applicazione può adottare strategie diverse. Essa può, ad esempio, guidare l'acquisizione interattiva della descrizione della N in modo da evitare che il configuratore compia errori. In alternativa, la correttezza sintattica della N può essere verificata a valle dell'introduzione della sua descrizione, avvertendo il configuratore nel caso si siano riscontrati difetti.

Dietro richiesta del configuratore, ciascuna descrizione di N sintatticamente corretta può essere salvata in forma persistente. Tutte le N salvate devono essere diverse l'una dall'altra. Le descrizioni salvate devono essere accessibili al configuratore, che, in qualunque sessione, può richiedere la visualizzazione di ognuna di esse.

VERSIONE 2 – REQUISITI FUNZIONALI

La seconda versione estende la prima consentendo al configuratore di scegliere una N fra quelle salvate in forma persistente e usarla al fine di definire una rete di Petri (PN) che abbia quella topologia di rete. Per definire la PN, è necessario che il configuratore inserisca i pesi degli elementi della relazione di flusso e una marcatura iniziale per la N prescelta. Per semplificare l'inserimento di pesi e marcature, si può fare ricorso a dei valori di default. In particolare, il valore di default del peso di un elemento della relazione di flusso è pari a 1 e quello della marcatura iniziale di un singolo posto è pari a 0. La stessa N può essere utilizzata per descrivere più PN distinte, che condividono la topologia ma differiscono per pesi e marcatura iniziale. Ogni singola PN deve essere sintatticamente corretta, ovvero tutti i pesi devono assumere valori interi positivi e la marcatura iniziale di ciascun posto deve essere un valore intero non negativo.

Dietro richiesta del configuratore, ciascuna descrizione di PN sintatticamente corretta può essere salvata in forma persistente. Tutte le PN salvate devono essere diverse l'una dall'altra. Le descrizioni salvate devono essere accessibili al configuratore, che, in qualunque sessione, può richiedere la visualizzazione di ognuna di esse.

VERSIONE 3 – REQUISITI FUNZIONALI

La terza versione dell'applicazione, in aggiunta alle funzionalità delle due versioni precedenti, consente al fruitore di scegliere una PN fra quelle salvate in forma persistente dal configuratore e simulare una o più evoluzioni della stessa. 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 delle transizioni abilitate data la marcatura corrente. La marcatura raggiunta (che, incidentalmente, può coincidere con quella corrente) diventa quella corrente, e così via finché il fruitore non ponga fine alla simulazione o non si sia raggiunta una marcatura che denoti un blocco critico della PN.

Più in dettaglio, 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 le mostra, chiede al fruitore di selezionarne una e visualizza la marcatura prossima raggiunta attraverso lo scatto della stessa.

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

VERSIONE 4 – REQUISITI FUNZIONALI

La quarta versione estende le precedenti consentendo la descrizione, da parte del configuratore, e la simulazione, da parte del fruitore, di PN con priorità (PNp).

Più specificamente, la nuova versione deve consentire al configuratore di scegliere una PN fra quelle salvate in forma persistente e usarla al fine di definire una PNp. Per definire una PNp, il configuratore deve attribuire una priorità a ciascuna transizione della PN prescelta. Per semplificare l'inserimento delle priorità, si può fare ricorso a dei valori di default. In particolare, il valore di default della priorità di una transizione è quello minimo, pari a 1. La stessa PN può essere utilizzata per descrivere più PNp distinte, che differiscono unicamente per i valori delle priorità. Ogni singola PNp deve essere sintatticamente corretta, ovvero tutte le priorità devono assumere valori interi positivi.

Dietro richiesta del configuratore, ciascuna descrizione di PNp sintatticamente corretta può essere salvata in forma persistente. Tutte le PNp salvate devono essere diverse l'una dall'altra. Le descrizioni salvate devono essere accessibili al configuratore, che, in qualunque sessione, può richiedere la visualizzazione di ognuna di esse.

La nuova versione deve consentire al fruitore di scegliere una PNp fra quelle salvate in forma persistente e simulare una o più evoluzioni della stessa. 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 delle transizioni la cui priorità è la più elevata fra quelle di tutte le transizioni abilitate data la marcatura corrente. La marcatura raggiunta (che, incidentalmente, può coincidere con quella corrente) diventa quella corrente, e così via finché il fruitore non ponga fine alla simulazione o non si sia raggiunta una marcatura che denoti un blocco critico della PNp.

Più dettagliatamente, a ogni iterazione della simulazione di una PNp, 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 al fruitore 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 al fruitore se intende continuare la simulazione, ne acquisisce la risposta e agisce di conseguenza.

VERSIONE 5 – REQUISITI FUNZIONALI

La quinta versione dell'applicazione, in aggiunta a tutte le funzionalità di cui alle quattro versioni precedenti, dà al configuratore la facoltà di importare da file la descrizione di N, PN e PNp senza doverne effettuare l'inserimento interattivo. Ciascun file contiene la descrizione di una singola entità, cioè di una singola N o PN o PNp.

L'importazione della descrizione di una PN può avvenire solo se il sistema software dispone già della descrizione della N che ne stabilisce la topologia, perché essa è stata precedentemente salvata in forma persistente.

Analogamente, l'importazione della descrizione di una PNp può avvenire solo se il sistema software dispone già della descrizione della PN su cui essa si basa, perché essa è stata precedentemente salvata in forma persistente.

Si noti che l'importazione delle descrizioni di N, PN e PNp deve coesistere nell'applicazione con la possibilità di inserire tali dati interattivamente.

REQUISITI NON FUNZIONALI

Il modello di processo da adottare è incrementale/iterativo.

Il linguaggio di programmazione da utilizzare è Java.

L'architettura esterna da realizzare per l'applicazione è stand alone.

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

Non è richiesta la creazione di una interfaccia utente grafica (tuttavia è bene che l'architettura interna sia progettata in modo da ridurre gli effetti collaterali e lo sforzo connesso al cambiamento se in futuro il sistema di interazione testuale fosse sostituito da una GUI).

Non è richiesto l'impiego di alcun DBMS (Data Base Management System).

NOTA

Ogni versione richiesta estende le funzionalità della precedente, senza modificarle.

I requisiti funzionali della prima versione esigono che ogni N salvata in forma persistente sia *diversa* da quelle già salvate. Si intendono qui uguali due N indistinguibili, ovvero due N che non solo esprimono la medesima topologia ma usano anche gli stessi identificatori per posti e transizioni. Due N che esprimono la medesima topologia ma usano identificatori diversi sono da ritenere diverse.

I requisiti non funzionali non impongono alcuna tecnologia da utilizzare per la memorizzazione persistente dei dati né per la loro importazione (non interattiva). Per quanto riguarda la prima, la scelta dei progettisti potrebbe ricadere banalmente sulla serializzazione di oggetti. Per quanto riguarda la seconda, si possono impiegare file di testo aventi una sintassi definita appositamente dai progettisti stessi (ad esempio, file in formato CSV o JSON o XML ...). Si noti che tali file potrebbero essere adottati anche per il salvataggio.

L'enfasi del progetto non è sulla sicurezza, pertanto l'attribuzione di credenziali a configuratori e fruitori non è ritenuta un'operazione critica.

I requisiti (funzionali e non) delle cinque versioni dell'applicazione da realizzare sono deliberatamente espressi a un alto livello di astrazione (ad esempio, non si sono imposti limiti alle dimensioni delle N 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. Tali aggiunte devono essere chiaramente documentate.

ESTENSIONI FUTURE

Si elencano di seguito alcuni possibili punti di estensione o modifica dell'applicazione, non affinché i requisiti a essi relativi siano soddisfatti ma perché anticipare i cambiamenti è un importante principio di progettazione (teso a rendere l'applicazione insensibile agli stessi).

- Il sistema di interazione potrebbe divenire grafico.
- L'architettura esterna potrebbe diventare distribuita (per descrivere N, PN e PNp, il configuratore agirebbe sul back-end, installato sul server, mentre il fruitore interagirebbe col front-end, installato sul client).
- L'archivio dei dati potrebbe essere gestito attraverso un DBMS.
- La simulazione potrebbe evidenziare a ogni iterazione la presenza di eventuali conflitti fra le transizioni abilitate.
- La simulazione potrebbe evidenziare a ogni iterazione la presenza di gruppi di transizioni abilitate reciprocamente concorrenti.
- La simulazione potrebbe essere accelerata, consentendo a ogni iterazione lo scatto simultaneo di più transizioni concorrenti.
- Si potrebbero considerare anche PN temporizzate e PNp temporizzate.

Richieste relative alla PRIMA PARTE DEL PROGETTO

Agli studenti è richiesto di realizzare evolutivamente cinque versioni software che soddisfino i requisiti sopra esposti. Ogni gruppo (costituito al più da tre persone), dovrà:

- 1) per ogni versione, produrre la documentazione di progetto, contenente
 - casi d'uso (comprensivi dell'espressione di eventuali requisiti aggiuntivi), sia in forma testuale, sia in forma di diagramma UML; si invita a rendere evidenti a colpo l'occhio le integrazioni/modifiche apportate ai casi d'uso (testuali e grafici) della versione precedente per ottenere quelli della versione corrente;
 - diagramma UML delle classi,
 - diagrammi UML comportamentali (opzionali),e qualsiasi altra specifica ritenuta opportuna; la documentazione relativa alle cinque versioni deve essere raccolta in un unico file;
- 2) per l'ultima versione, redigere un unico manuale di installazione e uso (il cui contenuto potrebbe eventualmente divenire parte dell'help in linea dell'applicazione); si sottolinea la necessità di documentare accuratamente il da farsi al fine di importare nell'applicazione le descrizioni di N/PN/PNp;
- 3) consegnare in formato elettronico quanto richiesto ai punti precedenti;
- 4) per ogni versione, consegnare codice sorgente + codice interpretabile + (preferibilmente) codice eseguibile.

MODALITÀ DI CONSEGNA E ALTRO

Come riportato dalla guida in linea relativa all'insegnamento, a partire dall'a.a. 2020-2021, ai fini del superamento dell'esame, ogni studente dovrà sostenere **una sola prova orale** (discussione di entrambe le parti dell'elaborato).

Il materiale relativo alla PRIMA PARTE del progetto, di cui ai precedenti punti 3 e 4, deve essere consegnato attraverso la **piattaforma Moodle**, in anticipo rispetto al materiale relativo alla SECONDA PARTE e rispetto alla prova orale. Più precisamente, il materiale relativo alla PRIMA PARTE deve essere consegnato (al più tardi) entro la data in cui è fissata la prova orale dell'appello precedente rispetto a quello in cui si intende sostenere la prova orale. Le date delle prove orali sono indicate nella *pagina di appelli e prove parziali del portale di Ateneo*. La consegna del materiale relativo alla PRIMA PARTE dell'elaborato deve avvenire senza alcuna iscrizione ad alcuna prova da parte dei componenti del gruppo.

Si rammenta che il progetto corrente, assegnato nell'a.a. 2020-2021, potrà essere discusso in ciascuna delle sue due parti solo **fino alla sessione d'esame di Giugno – Luglio 2022**. Gli studenti che non avessero effettuato la discussione entro tale sessione dovranno svolgere il progetto assegnato nell'anno accademico successivo.