



Tesi di Laurea

**Automazione del testing delle Interfacce utente di applicazioni WEB:
un caso di studio**

Anno accademico 2009 / 2010

Relatore

Ch.mo prof. Porfirio Tramontana

Correlatore

Ch.mo Ing. Domenico Amalfitano

Candidato

Pino Raffaele Oliva

Matr. 576/118



Problematiche affrontate:

Automazione del Testing delle interfacce utente di Applicazioni Web di tipo Rich Internet Application (RIA)

Obiettivi:

- Proporre un processo di testing valido per applicazioni web di tipo RIA che preveda l'utilizzo di alcuni tools sviluppati nell'ambito del laboratorio di Ingegneria del Software dell'università di Napoli.
- proporre una strategia di generazione di Test Suite e di oracolo a partire da una sessione utente.
- Verifica dell'efficacia del processo proposto e dei tools utilizzati attraverso esperimenti su una applicazione nota nella quale sono stati iniettati opportuni difetti.



Caratteristiche principali delle applicazioni web moderne:

- | | |
|---|---|
| ✓ Interfacce utente intelligenti | ✓ Elaborazione lato client |
| ✓ Comunicazione asincrona con il web server | ✓ funzionalità più ricche come il drag & drop |
| ✓ Controlli interattivi e dinamici | ✓ Interfacce personalizzate per l'utente |

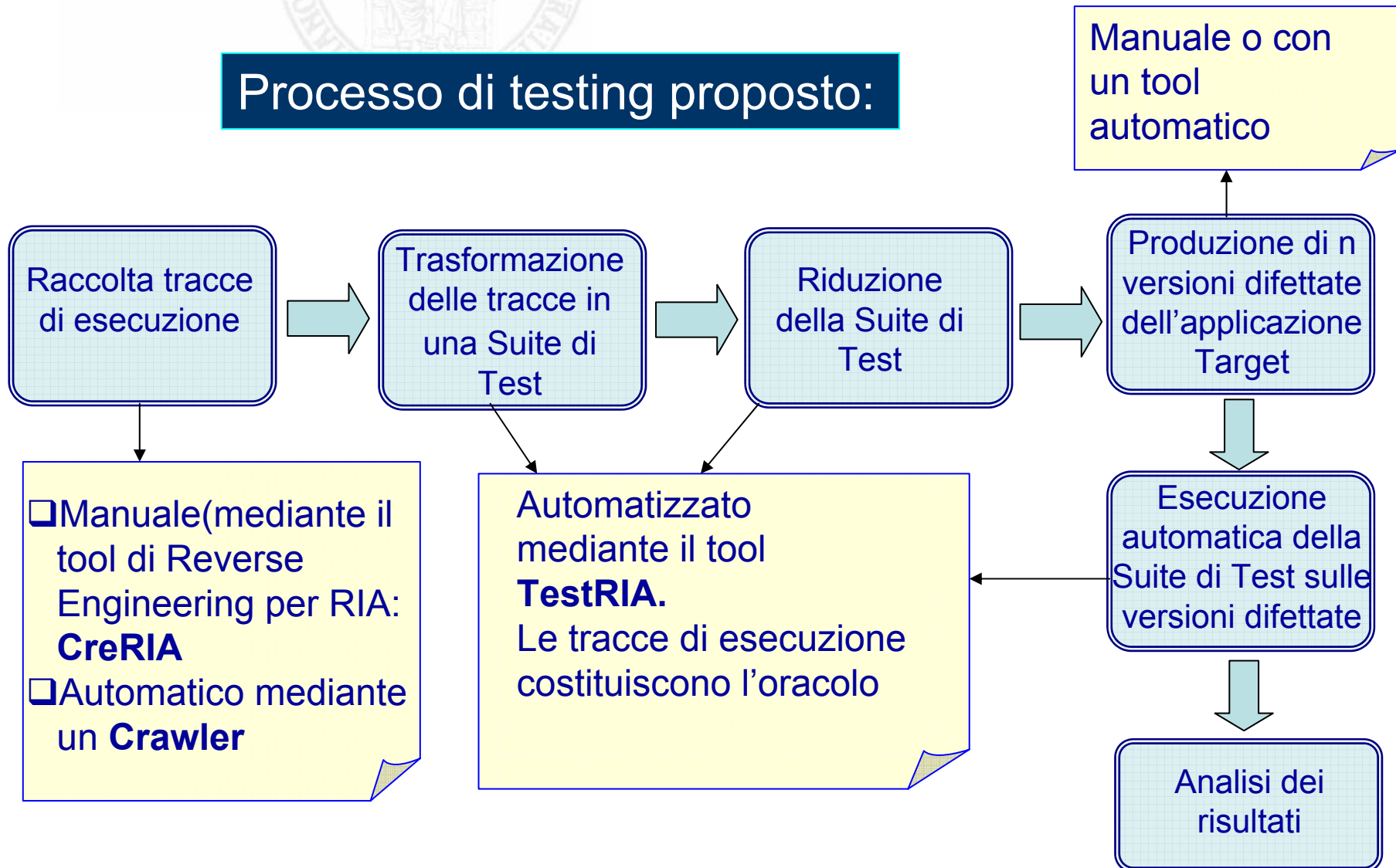
Conseguenze sull'attività di testing:

Il codice eseguito lato client dipende dallo stato dell'interfaccia.

Occorre pertanto una metodologia che sia in grado di valutare:

- ✓ lo stato iniziale delle interfacce;
- ✓ gli eventi scatenati sia dall'utente che dal sistema;
- ✓ lo stato finale dell'interfaccia;

Processo di testing proposto:





I tools utilizzati

CreRIA

Permette l'attività di:

- ❖ collezione, analisi e classificazione delle tracce di esecuzione;
- ❖ astrazione e validazione dell'FSM risultante.

TestRIA

Permette l'attività di:

- ❖ Trasformazione delle tracce di esecuzione in Suite di Test;
- ❖ Riduzione della Suite di test in una equivalente, possibilmente più piccola;
- ❖ generazione di asserzioni;
- ❖ Esecuzione automatica della suite di test.

Applicazione target

- Applicazione realizzata con tecnologia ASP NET 3.5;
- Utilizza widget forniti dalla libreria AJAX NET;
- Utilizza una base dati su SQLServer 2008



Costituisce una sorta di scrivania virtuale per gli addetti ai lavori.

Su tale sito si trovano tutti gli strumenti utili allo svolgimento dell'attività lavorativa giornaliera, nonché un sistema di messaggistica gestito da esperti che informa sulle novità sia legislative che tecniche.



Produzione delle tracce di esecuzione

A copertura di tutti gli scenari dei casi d'uso dell'applicazione target è stata ricavata una Suite di Test completa.

La Suite di Test è stata eseguita manualmente con il tool CreRIA per memorizzare nella base dati MySQL tutte le tracce di esecuzione.

Numero Test Case	293
Interfacce navigate	2.645
Tempo di esecuzione	30 (ore)



Produzione delle versioni difettate

Sono state create 30 versioni difettate ognuna contenente un solo difetto

I difetti iniettati sono stati suddivisi in:

- difetti che si riflettono sicuramente sull'interfaccia utente;
- difetti che non si riflettono sull'interfaccia utente.

In questa fase di Faults Injection si è cercato di simulare quegli errori che più frequentemente si verificano durante la scrittura del codice JavaScript.
Esempi: operatore booleano errato, operatore di confronto errato, riferimento a elementi di array fuori limite, riferimento ad un oggetto null, chiamata a funzione con parametro errato, riferimento ad una risorsa server inesistente, ecc.



Riduzione ed esecuzione della Suite di Test

La Suite di Test è stata eseguita in modo automatizzato con TestRIA su tutte le 30 versioni difettate.

Tempo di esecuzione dei 293 Test su ogni versione difettata dell'applicazione

5 ore circa

Esito della sperimentazione:

Difetti rilevati

23

Difetti non rilevati

7



Considerazioni sui risultati della sperimentazione

I dati sul rilevamento dei difetti sono stati:

Difetti che hanno comportato un effetto visibile sull'interfaccia	17 su 17	100%
Difetti senza effetti visibili sull'interfaccia	6 su 13	46%

Analizzando i difetti della seconda categoria (senza effetti visibili sull'interfaccia), si è riscontrato che il 46% cioè quelli individuati durante gli esperimenti, pur non producendo effetti visibili sull'interfaccia, sono caratterizzati dal provocare una anomalia sulla struttura del DOM mentre ciò non avviene per gli ulteriori 54% che non sono stati rilevati. Questi infatti provocano solo un crash JavaScript.

Conclusioni:

La sperimentazione ha dimostrato che il processo di testing è sicuramente efficace e gli strumenti utilizzati hanno dato risultati più che soddisfacenti.

Abbiamo potuto rilevare che essi funzionano in modo eccellente per quei difetti che comportano effetti visibili sull'interfaccia o una variazione della struttura del DOM.

Sviluppi futuri:

- ❖ Ulteriore miglioramenti degli strumenti utilizzati al fine di rilevare anche quei difetti che non hanno effetti né sull'interfaccia, né sulla composizione del DOM delle pagine.
- ❖ Utilizzare lo stesso processo di testing e gli stessi strumenti in ulteriori fasi sperimentali magari utilizzando un criterio di clusterizzazione delle interfacce diverso da quello utilizzato in questo lavoro di tesi oppure verificare se i difetti non rilevati da TestRIA sono rilevabili con DynaRIA.
- ❖ Automatizzare maggiormente il processo di generazione dei casi di test nella fase di raccolta delle tracce di esecuzione per migliorare l'efficienza del processo, salvo poi monitorare che non ci sia stato una perdita in efficacia.