

tesi di laurea

Testing di applicazioni flex: confronto tra gli strumenti FlexUnit e FlexMonkey

Anno Accademico 2009/2010

relatore

Ch.mo prof. Porfirio Tramontana

candidato

Pasquale Ludi

Matr. 534\000438



•Obbiettivi del progetto:

- Sviluppo di un'applicazione Flex in AdobeFlashBuilder 4



- Testing dell'applicazione sviluppata, con l'ausilio degli strumenti Open Source FlexUnit e FlexMonkey, che offrono supporto al testing di applicazioni Flex.



Applicazione target

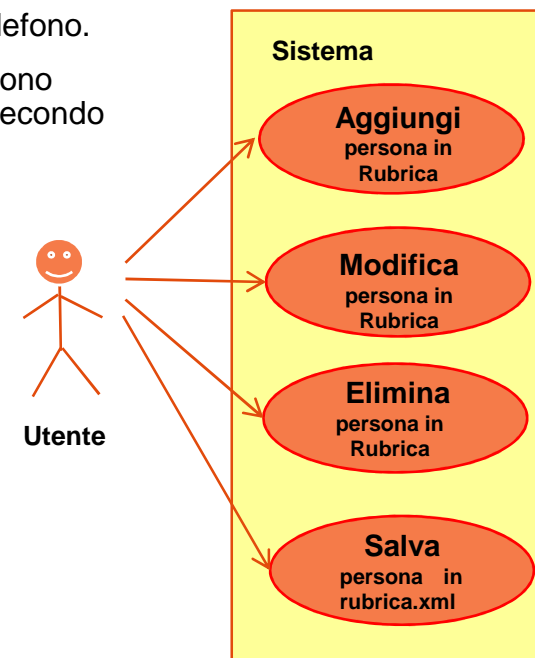
• Analisi dei requisiti

L'applicazione creata è una rubrica, realizzata utilizzando FlashBuilder4. Durante l'esecuzione dell'applicazione viene visualizzato una Rubrica inizialmente vuota, che successivamente verrà popolata e modificata dall'utente finale, attraverso le operazioni previste su di essa: Aggiungi, Modifica, Elimina e Salva.

- Rubrica ha come chiave primaria i tre campi: nome-cognome-telefono.
- I campi nome e cognome accettano solo caratteri, il campo telefono solo numeri, mentre il campo email accetta solo indirizzi validi secondo un'espressione letterale.
- I pulsanti Modifica ed Elimina si attivano solo se viene selezionato un record in Rubrica.
- Il pulsante Salva memorizza il contenuto di Rubrica in modo ordinato secondo il cognome in un file xml(rubrica.xml).

• Use Case Diagram

Dall'analisi dei requisiti è possibile estrarre le funzionalità del sistema, intese come le possibili interazioni dell'utente con esso.





Interfaccia Utente

COGNOME	NOME	TELEFONO	EMAIL

PANNELLO GESTIONE UTENTI

COGNOME

NOME

TELEFONO

EMAIL

- La parte grafica è stata scritta in MXML creato da Macromedia, utilizzato per la strutturazione delle interfacce , simile a XML.
- La parte della logica client, è stata scritta nel linguaggio di scripting ActionScript 3 di Adobe Flash.



FlexUnit

Il framework FlexUnit è un tool che facilita l'accumulo di test unitari per Flex e ActionScript, da eseguire in modo automatico, integrato in flash builder.

La fase di testing è stata articolata in due passi:

- Si fa funzionare il sistema con dati di input per i quali si conoscono i risultati (test case).
- Si verificano i risultati ottenuti.

La verifica dei risultati:

consiste nel confrontare le informazioni contenute nella Rubrica con un array contenente i risultati attesi (oracolo), attraverso i metodi messi a disposizione dal tool, tra i quali troviamo : AssertEquals, AssertTrue, AssertNull. I metodi sono tutti simili e prendono un messaggio di testo opzionale e fanno un test di tipo booleano.

La Suite di test

- I casi di test sono stati scelti cercando di stimolare l'applicazione con almeno un caso test per ogni tipo di scenario di esecuzione definiti a margine dei casi d'uso.
- sono state definite tutte le classi di equivalenza necessarie per la copertura:
 - dei valori normali interni al dominio;
 - dei valori illegali per i campi obbligatori.
 - Dei valori illegali secondo la restrizione sui campi.
- La scelta dei dati di input è stata fatta basandosi sulle specifiche del programma.
- I test case sono scritti in ActionScript.



Esempio di un Test Case in FlexUnit

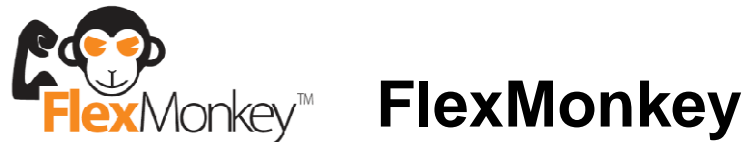
```
package NewTestCaseRubrica{
import flexunit.framework.TestCase;

public class TC1Aggiungi extends TestCase
{
public var math:ClasseRubrica;
private var ar:Array=new Array();
private var array:Array=new Array();

public override function setUp():void {
math = new ClasseRubrica();
ar.push({ cognomeInfo:"Rossi", nomeInfo:"Roberto",
telefonolInfo:"345768798", emailInfo:"Rossi@live.it"});
}

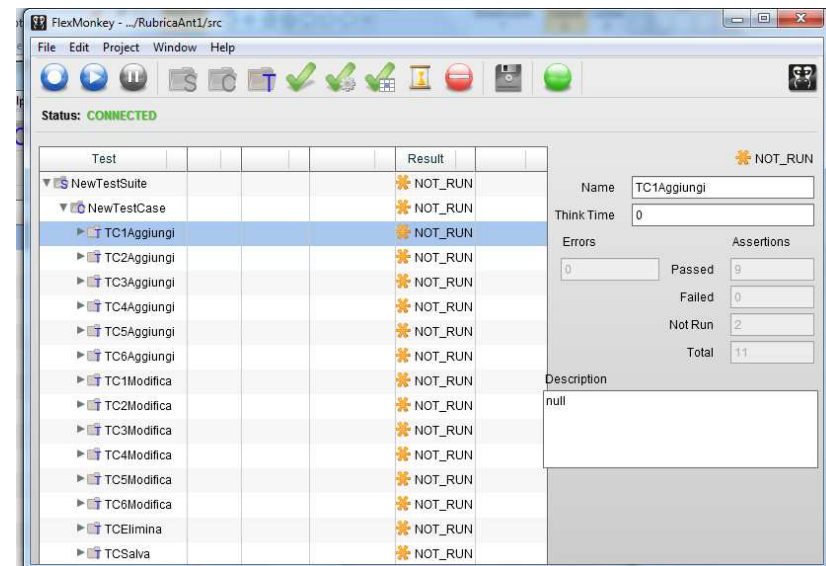
public function testAggiungi():void {
assertEquals("L'utente non è stato aggiunto",ar,
math.aggiungi(array,"Rossi","Roberto","345768798 ","Rossi@live.it"));
}
}
```

- E' stata importata la libreria *TestCase*, che ci consentirà di definire la classe di test estendendone i metodi.
- Il metodo di *setUp()*, viene invocato automaticamente dal framework , prima dell'esecuzione di ogni test case. Inizializza l'oggetto *math*, e fa una push su *ar*.
- FlexUnit usa le proprietà di introspezione di AS3 (reflection) per eseguire i metodi per il test: tutti i metodi della classe *TC1Aggiungi* che iniziano con la parola *test* verranno eseguiti all'interno della *TestSuite*.
- Il test vero e proprio del metodo *Aggiungi()* viene effettuato attraverso l'asserzione *assertEquals ()* che verifica se il valore di ritorno del metodo sia uguale al valore atteso, specificato da *ar* (oracolo).
- FlexUnit mediante un pannello integrativo aggiunto all'ambiente di sviluppo, consente di visualizzare l'esito dell'esecuzione dei test case e di tener traccia, nel caso di fallimenti, degli errori riscontrati.

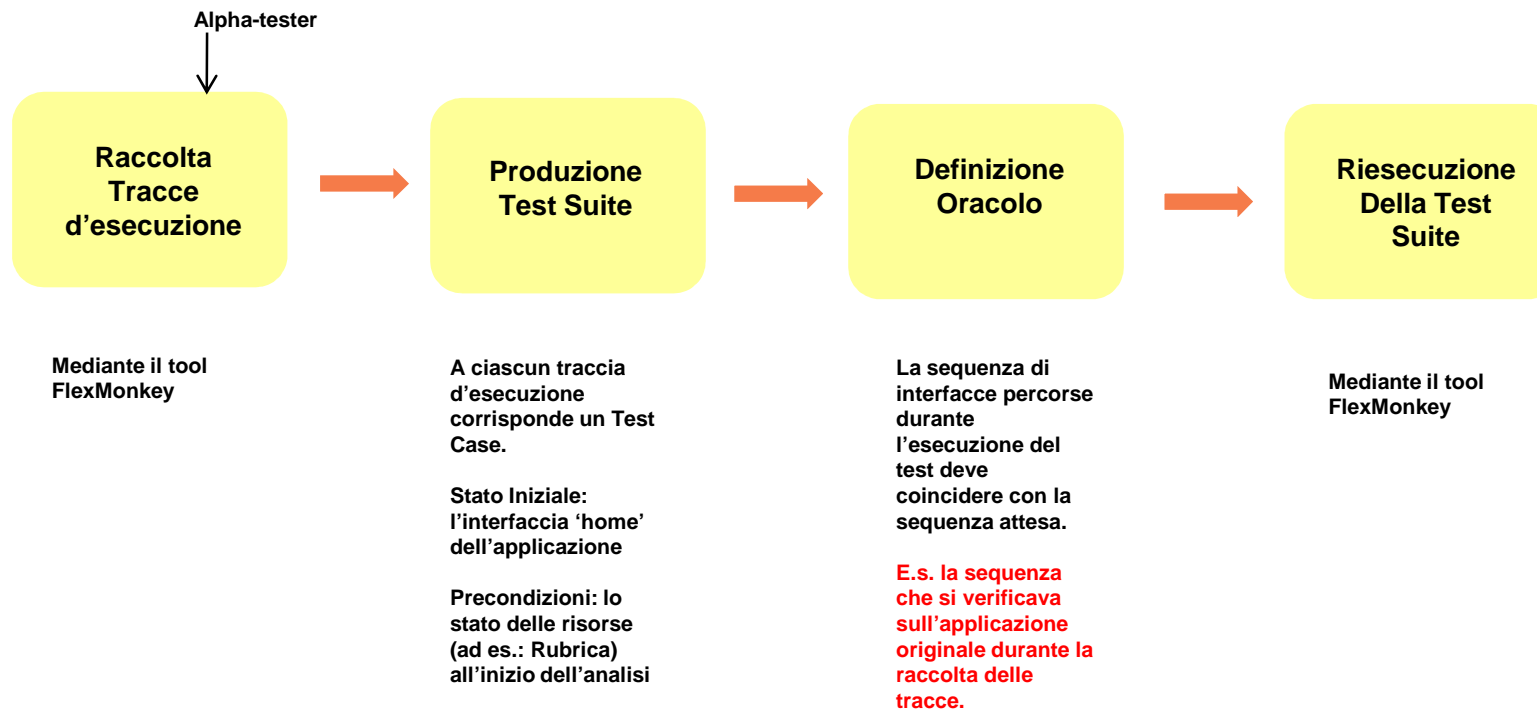


FlexMonkey è un tool open source distribuito da Gorilla Logic, che sfrutta il framework per l'automazione Flex, per registrare e riprodurre le interazioni dell'utente con la GUI di Flex.

- Il tool si basa sul tipico approccio appartenente alla categoria di testing di tipo black-box ossia il capture/replay.
- Questa tecnica è molto spesso utilizzata in fase di regression testing, dopo una modifica al codice sorgente dell'applicazione, è possibile eseguire nuovamente la test suite registrata sul codice modificato per stabilire se le modifiche apportate hanno alterato elementi precedentemente funzionanti.
- Genera script di test in Actionscript-based, che può essere utilizzato all'interno di un processo di continua integrazione.
- Si collega all'applicazione in esecuzione nel browser.
- E' possibile aggiungere dei passaggi di verifica alla prova registrata, attraverso i pulsanti messi a disposizione dalla console, che sono: addVerify, AddVerifyProperty e addVerifyGrid, e rieseguirli, verificando se il risultato ottenuto è quello atteso.



Il processo di produzione di Test Suite da user session data



La comparazione tra l'oracolo ed il sistema da testare avviene verificando opportuni asserts (asserzioni)

Le asserzioni sono verificate attraverso i pulsanti messi a disposizione dalla console, quali :
addVerify, AddVerifyProperty e addVerifyGrid.

Gorilla Logic prevede un'attività personalizzata Ant per l'esecuzione dei tests ActionScript generati da console, come parte di FlexMonkey, in ambiente FlashBuilder.



Apache Ant è una libreria Java e uno strumento da riga di comando il cui scopo, è quello di guidare i processi descritti nel file di compilazione (il Build.xml), come obiettivi e punti di estensione dipendenti l'una dall'altra. La libreria Ant deve essere installata su adobeFlashBuilder prima di poterla utilizzare. Al termine dell'esecuzione della Test Suite, Ant genera il reporting in formato HTML e JUnit.

Unit Test Results.

Class	Name	Status	Type	Time(s)
NewTestCase	TC1Aggiungi	Success		5.904
NewTestCase	TC1Modifica	Success		3.215
NewTestCase	TC2Aggiungi	Success		1.965
NewTestCase	TC2Modifica	Success		2.833
NewTestCase	TC3Aggiungi	Success		1.924
NewTestCase	TC3Modifica	Success		2.668
NewTestCase	TC4Aggiungi	Success		1.970
NewTestCase	TC4Modifica	Success		2.608
NewTestCase	TC5Aggiungi	Success		1.349
NewTestCase	TC5Modifica	Success		2.739
NewTestCase	TC6Aggiungi	Success		1.452
NewTestCase	TC6Modifica	Success		4.529
NewTestCase	TCElimina	Success		1.806
NewTestCase	TCSalva	Success		1.645

testSuites.NewTestSuite.tests.NewTestCase

Runs: 14/14 Errors: 0 Failures: 0

Test Case	Time (s)
TC1Aggiungi	5.849
TC1Modifica	3.144
TC2Aggiungi	1.898
TC2Modifica	2.691
TC3Aggiungi	1.861
TC3Modifica	2.736
TC4Aggiungi	1.917
TC4Modifica	2.620
TC5Aggiungi	1.341
TC5Modifica	2.653
TC6Aggiungi	1.426
TC6Modifica	4.625
TCElimina	1.773
TCSalva	1.624



Confronto



- Organizza i tests in Suite e testcase.
- Si basa su test di tipo unitario.
- I test unitari sono scritti per definire la funzionalità *prima che* il codice sia scritto.
- Più affidabile perché ha una copertura di codice migliore e meno costosa rispetto al codice che viene testato da esplorazione manuale.
- Gira costantemente durante il ciclo di sviluppo, piuttosto che alla fine di un ciclo di sviluppo a cascata.
- Uno svantaggio è il “costo” in termini di tempo perché va a gravare ulteriormente sul lavoro, durante il ciclo di sviluppo del software.



- Organizza i tests in Suite e testcase.
- Si basa sul testing di tipo capture\replay.
- E' molto semplice da utilizzare.
- Genera script di test in Actionscript-based, che può essere utilizzato all'interno di un processo di continua integrazione.
- Non è necessaria la conoscenza di altri linguaggi oltre all'ActionScript.
- Non è necessario apportare modifiche al sorgente dell'applicazione Flex o ai suoi metodi per usarlo.
- Una modifica all'interfaccia grafica, comporterebbe la ri-registrazione della prova.