

tesi di laurea

Sviluppo di processi per l'automatizzazione del testing per applicazioni Android

Anno Accademico 2011/2012

relatori

Ch.mo prof. Porfirio Tramontana

candidato

Enrico Solimeo

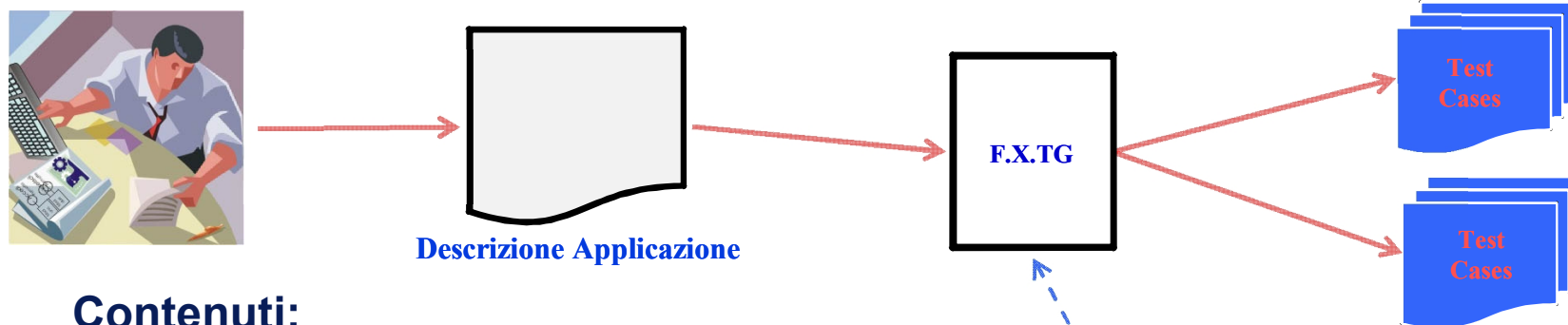
Matr. 534002361

Contesto:

 Testing di applicazioni basate sul sistema operativo Android

Obiettivi:

 Realizzare un processo per la generazione automatica di test per applicazioni Android a partire dal modello progettuale



Contenuti:

 Background

 Processo F.X.TG

 Casi di studio

*F.X.TG : Processo per la
generazione automatica
dei Test Cases*





Background 1/2



Framework Junit



Modellazione F.S.M.

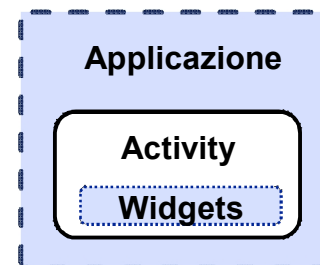


Linguaggio XML



Ogni applicazione Android è costituita da almeno :

- Una Activity
- Una moltitudine di Widgets



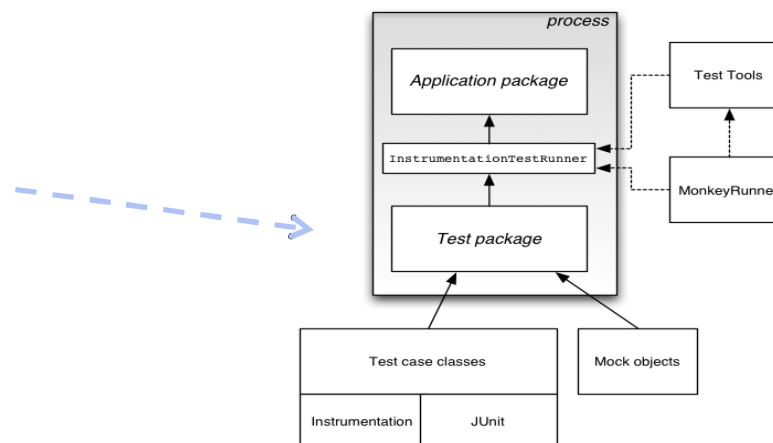
Framework Junit

Processo di Testing

Il testing è un punto cruciale nella realizzazione di ogni applicazione. A supporto del testing per applicazioni Android c'è il Framework Junit che fornisce tutta una serie di strumenti (ad esempio Stub e Mock) per la realizzazione di test funzionali.

I test realizzati con Junit sono programmi in grado di interagire con l'applicazione sotto test.

- *Stub E Mock : si sostituiscono ai componenti reali e consentono di testare i componenti in isolamento*

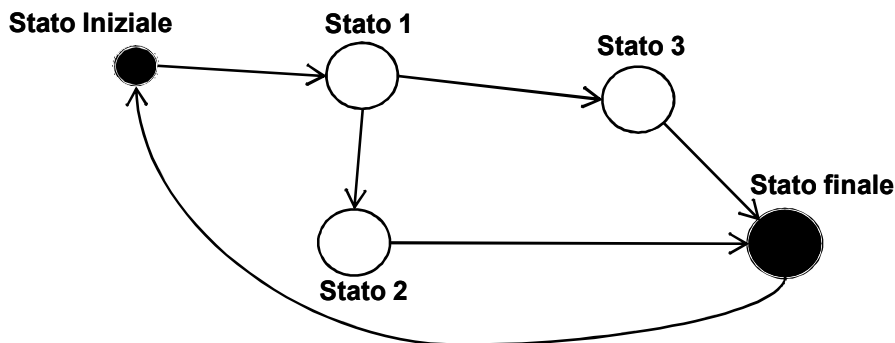




Background 2/2

Modellazione F.S.M.

*Ogni Applicazione Android interagisce con l'ambiente esterno ed in particolare con l'Utente. Essa, pertanto, muta il suo "stato" nel tempo.
Il funzionamento di una applicazione è per certi versi riconducibile a quello di una macchina a stati finiti e, pertanto, può essere descritto in UML mediante uno statechart diagram.*



Evento : *Causa delle transizioni*

Linguaggio XML

XML è un linguaggio di Markup basato su un meccanismo sintattico che consente di definire e controllare il significato degli elementi contenuti in un documento e consente di definire tag per descrivere struttura e tipo dei dati

Rubrica
Nome : Enrico
Cognome : Solimeo
Numero : 534002361

Traduzione in XML

```
<Rubrica> → nodo  
<Nome>Enrico</Nome> → tag  
<Cognome>Solimeo</Cognome> → tag  
<Numero>534002361</Numero> → tag  
</Rubrica>
```



Da modello descrittivo ad F.S.M.

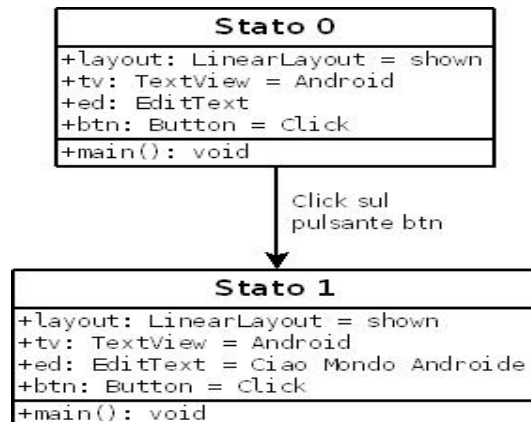
Il modello descrittivo di una applicazione fornisce una serie di caratteristiche delle quali solo alcune sono utili alla realizzazione di un modello F.S.M. Scendendo nello specifico individuiamo precisamente tre elementi utili :

- *Eventi (interazione dell'ambiente esterno con l' applicazione)*
- *Stati (caratterizzati dai widgets e dal loro valore)*
- *Transizioni (caratterizzate da stato iniziale, stato finale, evento, eventuali widgets coinvolti)*

Da F.S.M. a modello XML

Il modello XML è realizzato codificando informazioni ricavabili dal modello descrittivo a dal modello F.S.M. In Particolare saranno codificati :

- *Eventi, Stati e Transizioni con relative caratteristiche così come individuabili sui modelli F.S.M. e Descrittivo*
- *Widgets (tipo, valore e relativi attributi), Funzioni (eseguite dall'applicazione) ed altre eventuali informazioni*

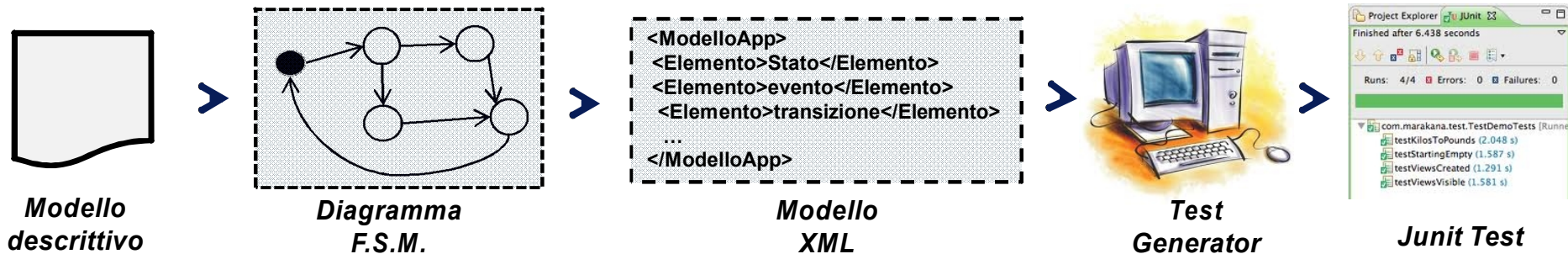


```

<Elemento id="0">
<descrizione>stato</descrizione>
<componete id= ><tipo>widget</tipo></componete>
</Elemento>
<Elemento id="1">
<descrizione>stato</descrizione>
<componete id=><tipo>widget</tipo></componete>
</Elemento>
<Elemento id="0" tipo="click">
<descrizione>evento</descrizione>
</Elemento>
<Elemento id="0">
1"stato_iniziale="0"stato_finale="1"evento="1"widget="3">
<descrizione>transizione</descrizione>
</Elemento>
  
```

Processo F.X.TG 1/2

F.X.TG è un processo di generazione automatica di test funzionali per applicazioni Android.



Punto di partenza del processo è il modello descrittivo dell'applicazione. Questo modello non è altro che una descrizione delle funzionalità e del comportamento dell'interfaccia utente dell'applicazione. Viene poi individuato ogni singolo componente dell'applicazione evidenziandone il cambiamento di valore in funzione del ciclo di vita di quest'ultima. Successivamente vengono definiti i vari stati (in funzione dei componenti e del loro valore) e gli eventi che causano le transizioni da uno stato all'altro.

Una volta raccolte tutte queste informazioni è possibile realizzare uno statechart diagram che descrive il comportamento dell'applicazione. Successivamente, tale diagramma sarà tradotto in linguaggio XML.



Processo F.X.TG 2/2

Punto molto importante del processo F.X.TG è la traduzione del diagramma F.S.M. in codice XML. Il file XML è strutturato in elementi (nodi). Ogni elemento ha un tag che ne specifica il tipo ed eventuali "nodi figlio" a loro volta dotati di tag che ne specificano tipo e caratteristiche.

Ultimo step del processo F.X.TG è il software Test Generator. Potremmo dire che esso dà un senso al lavoro svolto durante tutti gli step precedenti del processo fino alla realizzazione del codice XML. Rimane solo da capire come il modello XML venga utilizzato ...

Come viene usato il codice XML?

Il software Test Generator è in grado di interpretare il valore di nodi e tag. Sulla base delle informazioni acquisite verranno modellati automaticamente i casi di Test.

Possiamo sintetizzare il processo F.X.TG in 3 momenti

 ***Costruzione di un modello F.S.M.*** 

 ***Traduzione del modello F.S.M. in codice XML*** 

 ***Generazione dei Test attraverso il software TestGenerator*** 



Criteri di generazione dei test

Quando si scrivono dei test è importante :

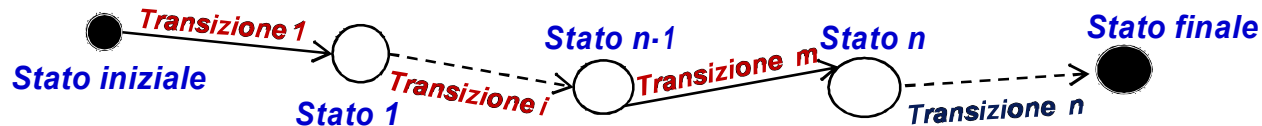


- Controllare mediante degli assert che il programma si comporti come atteso
- Valutare quanto del programma è stato testato.

Ogni test funzionale realizzato dal Test Generator è mirato a testare un singolo cammino.

Ogni cammino è costituito da una o più transizioni e pertanto passerà attraverso un numero variabile di stati dipendente dalla lunghezza del cammino in esame.

Il buon esito del test su una transizione facente parte di un determinato cammino implica la correttezza di tutte le transizioni che la precedono lungo tale cammino. Ad esempio il buon esito della transizione n nel seguente cammino, implica il buon esito di tutte le transizioni che la precedono.



Per valutare la copertura del codice utilizziamo il tool EMMA

EMMA Coverage Report (generated Tue Jul 31 17:59:59 CEST 2012)				
[all classes]				
OVERALL COVERAGE SUMMARY				
name	class, %	method, %	block, %	line, %
all classes	100% (6/6)	100% (12/12)	100% (563/563)	100% (119/119)
OVERALL STATS SUMMARY				
total packages: 1				
total executable files: 1				
total classes: 6				
total methods: 12				
total executable lines: 119				
COVERAGE BREAKDOWN BY PACKAGE				
name	class, %	method, %	block, %	line, %
enrico.Calculator2	100% (6/6)	100% (12/12)	100% (563/563)	100% (119/119)
[all classes]				
EMMA 0.0.0 (unsupported private build) (C) Vladimir Roubtsov				

EMMA Coverage Report (generated Thu Aug 02 12:55:38 CEST 2012)				
[all classes]				
OVERALL COVERAGE SUMMARY				
name	class, %	method, %	block, %	line, %
all classes	100% (4/4)	100% (8/8)	100% (410/410)	100% (87/87)
OVERALL STATS SUMMARY				
total packages: 1				
total executable files: 2				
total classes: 4				
total methods: 8				
total executable lines: 87				
COVERAGE BREAKDOWN BY PACKAGE				
name	class, %	method, %	block, %	line, %
enrico.Massimale	100% (4/4)	100% (8/8)	100% (410/410)	100% (87/87)
[all classes]				
EMMA 0.0.0 (unsupported private build) (C) Vladimir Roubtsov				

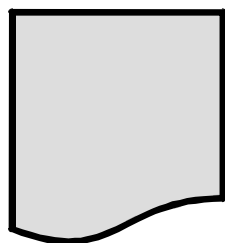


CASI DI STUDIO

Di seguito è schematizzato il processo F.X.TG (dal modello descrittivo all'esecuzione dei test) relativo a due applicazioni utilizzate come casi di studio. Le applicazini in questione sono :
Massimale : semplice applicazione pratica dedicata agli appassionati di fitness
Calculator : applicazione che riproduce una semplice calcolatrice in grado di eseguire le quattro operazioni elementari



Massimale



Modello descrittivo

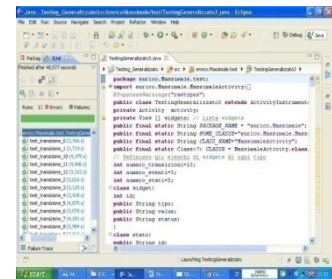


F.S.M



```
<Elemento id="0" tipo="
"LinearLayout" valore=" "
attributi="nn">
<descrizione>widget</descrizione>
</Elemento>
<Elemento id="1" tipo="TextView"
valore="PROGRAMMA PER IL
CALCOLO DEL MASSIMALE"
attributi="costante">
<descrizione>widget</descrizione>
</Elemento>
...
```

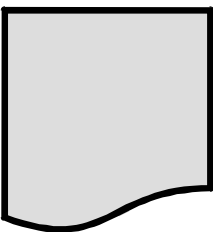
XML



Esecuzione dei Test



Calculator



Modello descrittivo

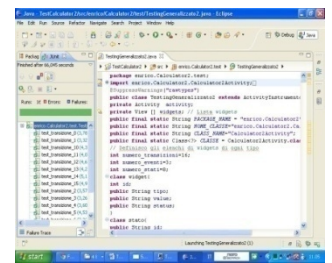


F.S.M



```
<Elemento id="0" tipo="
"LinearLayout" valore=" "
attributi="nn">
<descrizione>widget</descrizione>
</Elemento>
...
<Elemento id="18" tipo="
"LinearLayout" valore=" "
attributi="nn">
<descrizione>widget</descrizione>
</Elemento>
```

XML



Esecuzione dei Test



Sviluppi futuri

Un limite dell'attuale processo F.X.TG. è costituito dal tipo dei widgets gestibili con TestGenerator. Osservando da vicino i Test generati notiamo che essi si limitano ad un controllo sul valore degli "EditText" mediante l'uso di assertEquals. Le applicazioni Android prevedono widgets di svariata natura e l'utilizzo di componenti software ed hardware anche abbastanza complessi.



I punti cruciali da sviluppare sono pertanto :



Ampliamento dei widgets gestibili



Ampliamento del tipo di applicazioni testabili

