UNIVERSITA' DEGLI STUDI DI
NAPOLI FEDERICO II
Facoltà di Ingegneria
Corso di Studi in Ingegneria Informatica

tesi di laurea

Test in automatico di programmi Android

Anno Accademico 2012/2013

relatore
Ch.mo prof. Porfirio Tramontana

candidato
Andrea Macera
Matr. 534002556

Scopo della tesi

Creare una web application dalla quale sia possibile effettuare un testing in automatico di un programma Android da accesso remoto

Perché da accesso remoto?

Affinché il *ripper* possa essere utilizzato tramite web, bisogna renderlo accessibile da accesso remoto, in modo tale da poter essere adoperato da qualsiasi tipo browser.

Il ripper non è altro che un tool che in ingresso riceve un'applicazione Android e la testa.

A cosa serve una web application?

Le operazioni che compie la web application sono:

- **creazione di un interfaccia web, dove l'utente inserisce gli eventi, che verrano utilizzati per testare l'applicazione Android**
- **Dopo che l'utente ha effettuato inserimento di tutti i dati necessari per il testing, li invia, tramite delle *servlet*, ad un programma (scritto in java), che li acquisisce, li elabora, e crea un file XML, chiamato preferences, costruito in modo tale da poter essere letto ed assimilato dal ripper.**

La web application si divide in due parti

- **Client(**interfaccia**)**
- **Server**

CLIENT 1/3

- Scritto in HTML e JavaScript
  - HTML usato per creare l'interfaccia utente
  - JavaScript usato per controllare i dati in input inseriti dall'utente
- Utente inserisce i dati utili al ripper per effettuare il testing

Client 2/3

I dati inseriti dall'utente sono:

- Tipo di scheduler da utilizzare:
  - Systematic : esplorazione sistematica
  - Random : esplorazione casuale
- Tempi di attesa:
  - Evento : tempo che il ripper deve attendere dopo un evento
  - Restart : tempo che il ripper deve attendere dopo un riavvio
  - Throbber : tempo che il ripper deve attendere con un Throbber
  - Task : tempo che il ripper deve attendere dopo un task

Cliente 3/3

I dati inseriti dall'utente sono:

- Comparazione: si selezionano i possibili elementi sui quali effettuare una comparazione
- Interazione : testa le possibili interazioni che si possono avere con applicazioni
- Input : testa i possibili input che si possono inserire nell'applicazione

Server 1/2

- Scritto in Java, si occupa del trattamento dei dati inseriti dall'utente
- Crea il file XML che andrà letto dal ripper

Server 2/2

Per il passaggio dei dati utilizza il metodo "*request.getParameter*" implementato nella classe *javax.servlet.http.HttpServletRequest*

Es:

Lato client:

Inserisci testo <input type="text" name="esempio" id="esempio" value=""/>

Lato server

String testo=request.getParameter("esempio")

Gestione File 1/3

Per creare il file XML è stata scritta una procedura chiamata "*create*" che si occupa della creazione, gestione e cancellazione del file

Ecco come il file viene creato:

```
DocumentBuilderFactory  docFactory =
    DocumentBuilderFactory.newInstance();
      DocumentBuilder docBuilder =
    docFactory.newDocumentBuilder();
      Document doc = docBuilder.newDocument();
```

Gestione File 3/5

Creazione nodo pricipale:

```
radice=doc.createElement("preferences");
radice.setAttribute("EXTERNAL_XML_VERSION","1.0");
doc.appendChild(radice);
```

Gestione File 2/3

Creazione sotto-nodo

```
root=doc.createElement("root");
root.setAttribute("type","user");
radice.appendChild(root);
```

Gestione File 4/4

Eliminazione file:

```
if(xmlFile.exists())// VERIFICO SE IL FILE ESISTE
xmlFile.delete();//SE ESISTE LO CANCELLO
//SE NON ESISTE LO CREO
if (!xmlFile.exists()) {
xmlFile.createNewFile();
```

Conclusioni e Sviluppi Futuri

La web application rende possibile configurare un esecuzione del ripper du di una macchina remota, che possa essere ad esso dedicata.

Il Testing Automatico ha lo scopo di ottimizzare e migliorare i test che di solito verrebbero eseguiti manualmente

Il mondo dell'informatica è sempre in continua evoluzione, quindi in futuro verranno create sempre nuove web application con ulteriori aggiornamenti, per considerare sempre nuove situazioni da testare