MCrawlerT Wiki

Stassia R.Zafimiharisoa

29 juillet 2014

1 Key features

The MCrawlerT tool can :

1. automatically expand the user interfaces elements and retrieve its structure using crawler mechanism,
2. follow exploration strategies as Depth First Search (DFS), Breadth First Search (BFS) and BFS-Semantic strategies,
3. detect crashes of the Android application under exploration,
4. generate re-executable test cases,
5. construct models from the GUI structure, and generate the following output :
   - the story board (simple transition graph),
   - the STS models,
   - the minimised models
6. measure the code coverage of the source code

2 Download

Please, checkout the project from here [1] and compile. Or only download the toolset from here [2]. Then, setup the MCrawlerT environment variables, eg.,

```
export MCrawlerT=\$DOC\_HOME/MCRAWLERT/tools
export PATH=$PATH:$MCrawlerT:$MCrawlerT/scripts
```

3 Requirements for the execution of MCrawlerT

1. Linux(Ubuntu 12.04)/MacOSX(10.x)
2. JDK v.1.7.0_55
3. Android sdk v.22 or better

[1] https ://github.com/statops/MCrawlerT

[2] https ://www.dropbox.com/s/0hcxk3tjb91npln/MCRAWLERT.zip

4. Libraries for Android 2.3.4 or better
5. Ant version 1.9.3 or better
6. System variables :
   – ANDROID_HOME must be set to the sdk path of Android
   – JAVA_HOME must be set to the Java sdk path
   – MCrawleT must be set to the MCrwalerT tool path
   – Ant, android platform-tools and apktool path.

4 Step 0

1. At least an Android Virtual Device (AVD) or a mobile device is required. Please, refer here[3] to see how to create a new AVD otherwise directly use your Android phone (recommanded).

5 Step 1

The model generation can be performed in two manners :

1. Export environment variables, then call the script (*mct or mctApk*). A simple *export.sh* file where a list of useful runtime information must be filled.
2. Fill step by step the required parameters by calling the command : ***mct step***
3. For any help, launch the command ***mct help or mct -h*** to see the argument's list that mus be set.

5.1 The input files description

5.1.1 The *export.sh* script

If the first option is choosen, the required arguments are listed below :

```
OPTION : codeGen , display , explore , activityCoverage ,
   stress , inj , storyBoard
PROJECT : the project's path
PROJECTPACKAGE : the project's package
LAUNCHERACTIVITY : the launcher activity
EXPECTACTIVITY : the expected activity
TESTPROJECT : the test project's path
TESTPROJECTNAME : the test project's name
TESTDATA :  the user's test data
STRATEGY : the type of exploration's strategy
[0 : DFS
 1 : DFS-BFS
```

[3] http ://developer.android.com/tools/devices/index.html

```
 12: LOGGING semantic]
THREADNUMBER : number of device that will be launched
   in parallel
OUTPUTDIRECTORY : the output directory's path
MAININFOLDER_OF_PROJECT : the parent folder containing
    the project under test
```

Others argument may be setted as the $MAXTIME$ (the maximum execution duration), etc. Please refer to the help command to have the list of all arguments.

5.1.2 The *testdata.xml* file

An exemple of the input data structure is shown below, it is composed of user data serialised into xml, Test data are later generated from this file according to the type of the editable field. For instance, if a couple of *logging/password* is detected on the GUI structure, input data that will fill this field will be retrieved from *logging* and *password* elements.

```
<?xml version="1.0" encoding="UTF-8"?>
<RandomValue>
    <string>
        <value>text</value>
    </string>
    <int>
        <value>15</value>
        <value>8950355</value>
    </int>
    <password>
        <value>passwd1</value>
    </password>
    <logging>
        <value>test</value>
    </logging>
    <inj>
        <value>{data}\' OR 1=1--;</value>
    </inj>
    <mail>
        <value>s.zafimiharisoa@openium.fr</value>
    </mail>
    <stress>
        <value>%</value>
        <value>8950355</value>
    </stress>
    <semantic>
        <value>sign in</value>
    </stress>
</RandomValue>
```

6 Step 3

At the end of each operation, the execution report, the models, the code Coverage report and the generated Test cases are generated. These results are available in the output folder. An example of a readable final report is shown below :

```
Application package: fr.converter
Execution time: 989
Test number: 73
Code coverage:     100 %
Activity coverage: 100.0 %
Number of detected bugs: 1
==================================================
Reported error:
fr.converter.test.MainTest:
INSTRUMENTATION_RESULT: shortMsg=java.lang.
   NumberFormatException
INSTRUMENTATION_RESULT: longMsg=java.lang.
   NumberFormatException: text
INSTRUMENTATION_CODE: 0

Error path
source: fr.converter.MainActivity
edit text  fr.converter:id/question text
click radio_button  fr.converter:id/radio1
==================================================
```

Test cases are genetated for crash replay purpose if at least a crash is detected during the application's exploration. An example is shown below :

```
public void test_1() {
        MCrawlerVariable.Widget s00uielt0= mVariable.
           new Widget("fr.converter:id/ok","android.
           widget.Button","","","0","true","false","
           true","false", "Convert");
        MCrawlerVariable.Widget s00uielt1= mVariable.
           new Widget("fr.converter:id/radio2","
           android.widget.RadioButton","","","0","true
           ","false","true","false", "          liter
           - pinte");
        MCrawlerVariable.Widget s00uielt2= mVariable.
           new Widget("fr.converter:id/radio1","
           android.widget.RadioButton","","","0","true
           ","false","true","false", "          euro -
            livre sterling ");
        MCrawlerVariable.Widget s00uielt3= mVariable.
           new Widget("fr.converter:id/radio0","
           android.widget.RadioButton","","","0","true
           ","false","true","true", "          meter -
            feet");
        MCrawlerVariable.Widget s00uielt4= mVariable.
           new Widget("fr.converter:id/textView4","
           android.widget.TextView","","","0","true","
           false","true","false", "Meter  ");
        MCrawlerVariable.Widget s00uielt5= mVariable.
           new Widget("fr.converter:id/EditText01","
           android.widget.TextView","","","0","true","
           false","true","false", "");
        MCrawlerVariable.Widget s00uielt6= mVariable.
           new Widget("fr.converter:id/textView3","
           android.widget.TextView","","","0","true","
           false","true","false", "Feet  ");
        MCrawlerVariable.Widget s00uielt7= mVariable.
           new Widget("fr.converter:id/question","
           android.widget.EditText","","","0","true","
           false","true","false", "");
        MCrawlerVariable.Widget s00uielt8= mVariable.
           new Widget("android:id/title","android.
           widget.TextView","","","0","true","false","
           true","false", "Simple Converter");
        MCrawlerVariable.State s0 = mVariable.new
           State("fr.converter.MainActivity", true,
           false,"0_2",s00uielt0,s00uielt1,s00uielt2,
           s00uielt3,s00uielt4,s00uielt5,s00uielt6,
           s00uielt7,s00uielt8);
        MCrawlerVariable.State s0_current =
           getCurrentState();
        assertTrue(MCrawlerFunctions.isEqualState(s0,
           s0_current));
        MCrawlerVariable.Widget t0uielt0 = mVariable.
           new Widget("fr.converter:id/radio1","
           RadioButton","","", "", "","","","", "");
        MCrawlerVariable.Action t0_action_0= mVariable
           .new Action("click radio_button",t0uielt0);
        MCrawlerVariable.Widget t0uielt1 = mVariable.
           new Widget("fr.converter:id/question","
           EditText","","", "", "","","","", "text");
        MCrawlerVariable.Action t0_action_1= mVariable
           .new Action("edit text",t0uielt1);
        performAction(0L,t0_action_0,t0_action_1);
    }
```

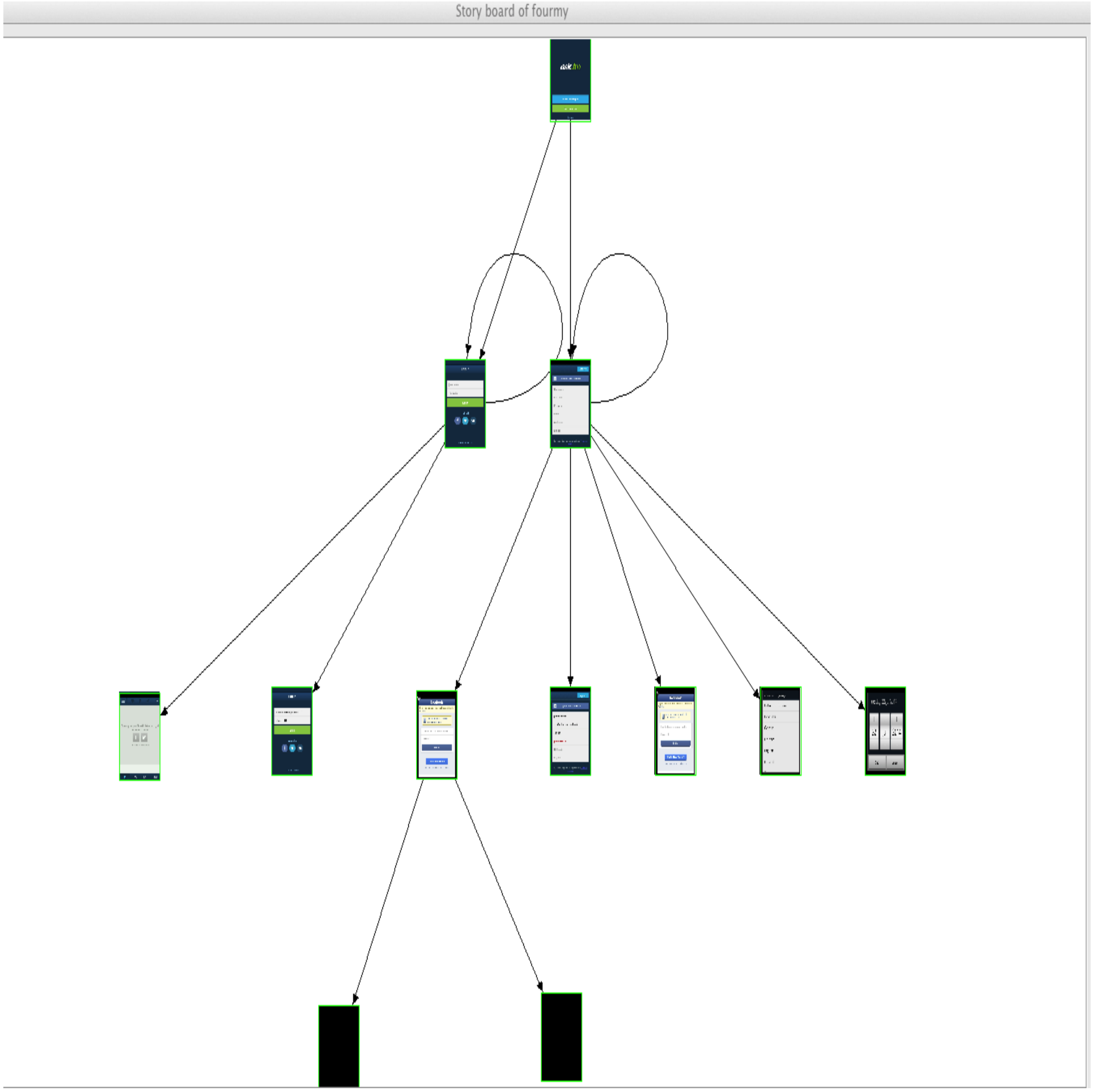

Figure 1 – StoryBoard of the Android Askfm application

An example of the generated simple story board is obtained after the launch of the command : *mct storyBoard*. An example of generated storyBoard is show in figure **??**.

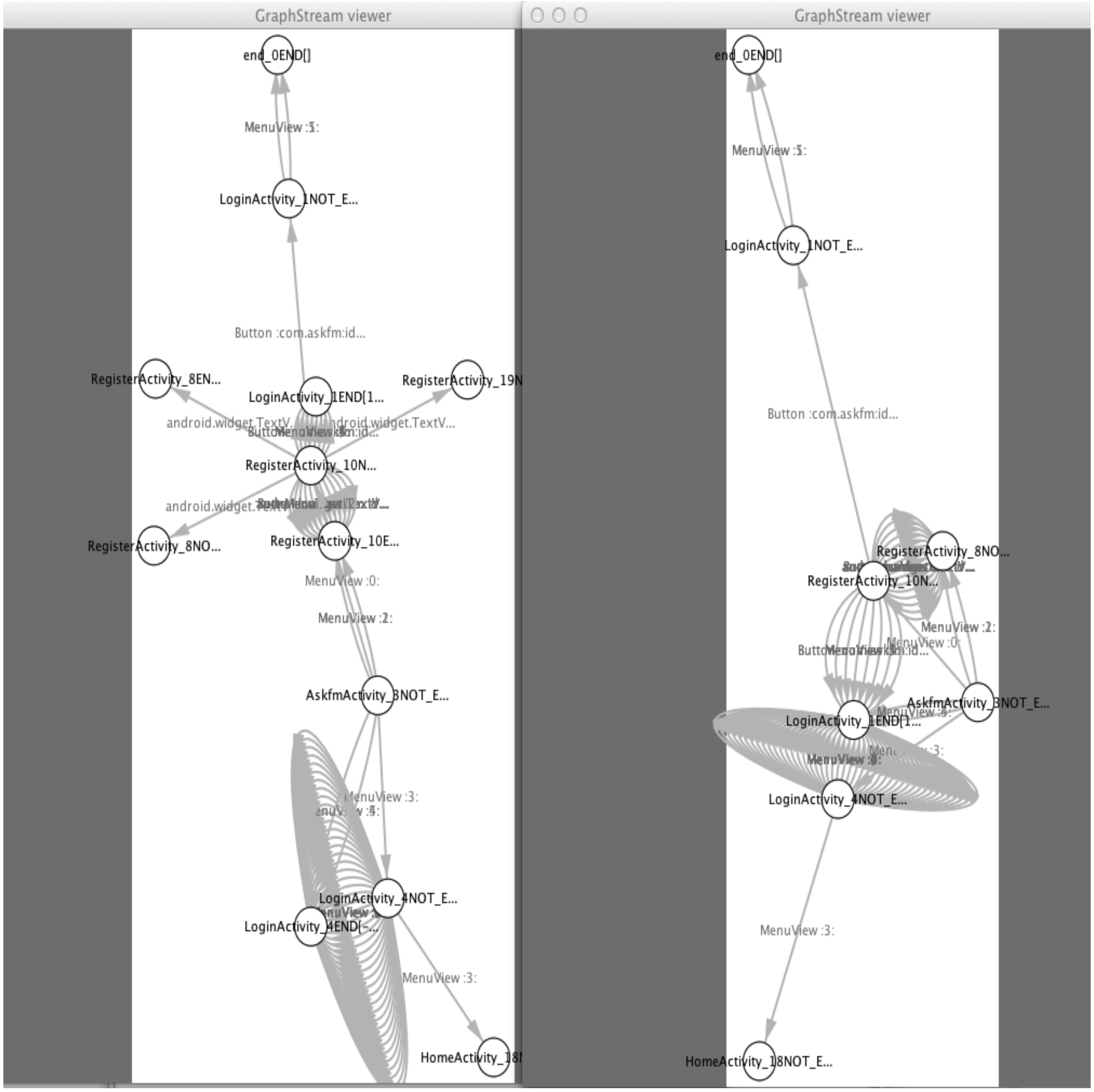

Figure 2 – STS and minimised STS tree of Android Askfm application

7 Examples

7.1 Android Ask application

7.1.1 Model generation

1. **Report**. A summary of the model report (*file : modelReport*) is presented below :

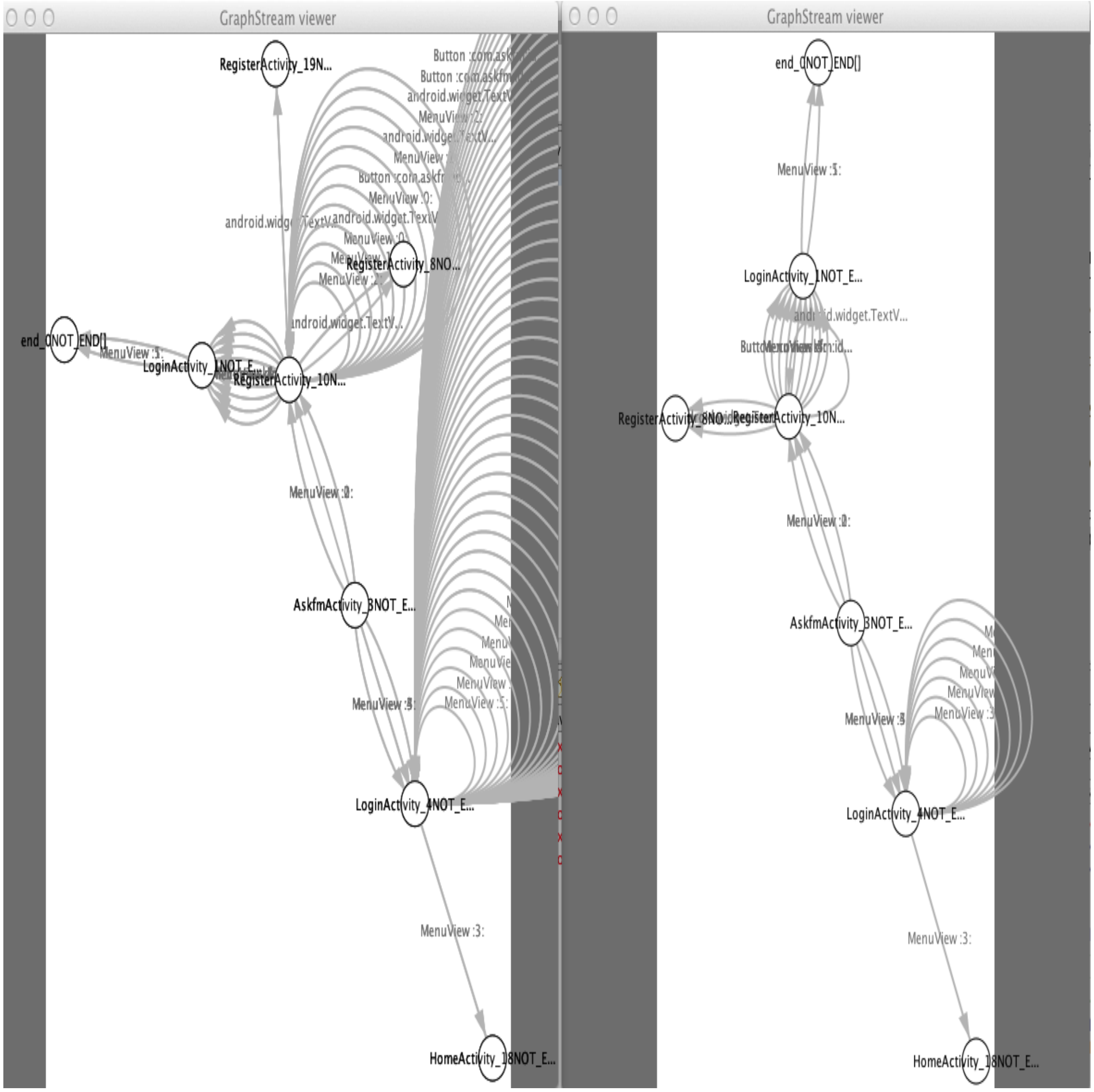

Figure 3 – Extrapolated STS and minimised Extrapolated STS tree of Android Askfm application

```
Application package: com.askfm
Execution time: 5417
Number of test: 79
Code coverage:
```

```

Activity coverage: 11.0 %
Number of detected bugs: 0
===========================================================================

===========================================================================
```

79 tests have been executed during 5417 seconds. No crash have been detected, and no coverage has been retrieved because we tested an apk package.

2. **Storyboard**. A simple storyBoard of the application is presented in **??**. It shows the different unique interface instance that have been explored. The black screen represent interface that cannot be explored, for instance a *web browser*, or an error.
3. **STS models.**
   - The STS tree and the minimised STS tree is illustarted in Figure **??**.
   - The Extrapolated models are illustrated in **??**.

   The way how these models have been generated is available in our paper [4].
4. **JUNIT**. an example of JUNIT output that can be used for a *re test* purpose is presented below to reach the *RegisterActivity*.

```
public class ModelValidationTest
     extends MCrawlerTTestCase
{

     private final static String TARGET_PACKAGE_ID
         = "com.askfm.test";
     private final static String
         LAUNCHER_ACTIVITY_FULL_CLASSNAME = "com.
         askfm.AskfmActivity";
     private final static String
         EXPECTED_INITIAL_STATE_ACTIVITY = "com.
         askfm.AskfmActivity";
     private static MCrawlerVariable var_0;
     private static MCrawlerFunctions var_1;
     private static State var_2;
     private static ScenarioData var_3;

     public void test_1() {
          MCrawlerVariable.Widget s10uielt0=
              mVariable.new Widget("android:id/
              content","FrameLayout","","","0","true"
              ,"false","true","false", "");
```

[4] Model Reverse-engineering of Mobile Applications with Exploration Strategies

```
        MCrawlerVariable.Widget s10uielt1=
           mVariable.new Widget("com.askfm:id/
           logolayout","RelativeLayout","","","0",
           "true","false","true","false", "");
        MCrawlerVariable.Widget s10uielt2=
           mVariable.new Widget("com.askfm:id/
           buttonlayout","LinearLayout","","","0",
           "true","false","true","false", "");
        MCrawlerVariable.State s0 = mVariable.new
           State("com.askfm.AskfmActivity", true,
           false,"0_0",s10uielt0,s10uielt1,
           s10uielt2);
        MCrawlerVariable.State s0_current =
           getCurrentState();
        assertTrue(MCrawlerFunctions.isEqualState(
           s0, s0_current));
        MCrawlerVariable.Widget t1uielt0 =
           mVariable.new Widget("0","MenuView","",
           "", "", "","","","", "");
        MCrawlerVariable.Action t1_action_0=
           mVariable.new Action("menu",t1uielt0);
        performAction(0L,t1_action_0);
        MCrawlerVariable.Widget s101uielt0=
           mVariable.new Widget("android:id/
           content","FrameLayout","","","0","true"
           ,"false","true","false", "");
        MCrawlerVariable.Widget s101uielt1=
           mVariable.new Widget("com.askfm:id/
           register_signup","android.widget.Button
           ","","","0","true","false","true","
           false", "Sign up");
        MCrawlerVariable.Widget s101uielt2=
           mVariable.new Widget("com.askfm:id/
           register_facebook","android.widget.
           Button","","","0","true","false","true"
           ,"false", "Sign up with Facebook");
        MCrawlerVariable.Widget s101uielt3=
           mVariable.new Widget("com.askfm:id/
           register_uid","android.widget.EditText"
           ,"","","0","true","false","true","false
           ", "");
        MCrawlerVariable.Widget s101uielt4=
           mVariable.new Widget("com.askfm:id/
           register_fullName","android.widget.
           EditText","","","0","true","false","
           true","false", "");
```

```
MCrawlerVariable.Widget s101uielt5=
   mVariable.new Widget("com.askfm:id/
   register_password","android.widget.
   EditText","","","0","true","false","
   true","false", "");
MCrawlerVariable.Widget s101uielt6=
   mVariable.new Widget("com.askfm:id/
   register_email","android.widget.
   EditText","","","0","true","false","
   true","false", "");
MCrawlerVariable.Widget s101uielt7=
   mVariable.new Widget("com.askfm:id/
   register_birthDate","android.widget.
   TextView","","","0","true","false","
   true","false", "");
MCrawlerVariable.Widget s101uielt8=
   mVariable.new Widget("com.askfm:id/
   register_language","android.widget.
   TextView","","","0","true","false","
   true","false", "English");
MCrawlerVariable.Widget s101uielt9=
   mVariable.new Widget("com.askfm:id/
   register_disclaimer","android.widget.
   TextView","","","0","true","false","
   true","false", "By clicking Sign up,
   you agree to our Terms of service");
MCrawlerVariable.State s01 = mVariable.new
    State("com.askfm.RegisterActivity",
   false, false,"0_0",s101uielt0,
   s101uielt1,s101uielt2,s101uielt3,
   s101uielt4,s101uielt5,s101uielt6,
   s101uielt7,s101uielt8,s101uielt9);
MCrawlerVariable.State s01_current =
   getCurrentState();
assertTrue(MCrawlerFunctions.isEqualState(
   s01, s01_current));
MCrawlerVariable.Widget s01uielt0=
   mVariable.new Widget("android:id/
   content","FrameLayout","","","0","true"
   ,"false","true","false", "");
MCrawlerVariable.Widget s01uielt1=
   mVariable.new Widget("com.askfm:id/
   register_signup","android.widget.Button
   ","","","0","true","false","true","
   false", "Sign up");
MCrawlerVariable.Widget s01uielt2=
   mVariable.new Widget("com.askfm:id/
   register_facebook","android.widget.
   Button","","","0","true","false","true"
   ,"false", "Sign up with Facebook");
MCrawlerVariable.Widget s01uielt3=
   mVariable.new Widget("com.askfm:id/
   register_uid","android.widget.EditText"
   ,"","","0","true","false","true","false
   ", "");
MCrawlerVariable.Widget s01uielt4=
   mVariable.new Widget("com.askfm:id/
   register_fullName","android.widget.
   EditText","","","0","true","false","
   true","false", "");
MCrawlerVariable.Widget s01uielt5=
   mVariable.new Widget("com.askfm:id/
   register_password","android.widget.
   EditText","","","0","true","false","
   true","false", "");
MCrawlerVariable.Widget s01uielt6=
   mVariable.new Widget("com.askfm:id/
   register_email","android.widget.
   EditText","","","0","true","false","
   true","false", "");
MCrawlerVariable.Widget s01uielt7=
   mVariable.new Widget("com.askfm:id/
   register_birthDate","android.widget.
   TextView","","","0","true","false","
   true","false", "");
MCrawlerVariable.Widget s01uielt8=
   mVariable.new Widget("com.askfm:id/
   register_language","android.widget.
   TextView","","","0","true","false","
   true","false", "English");
MCrawlerVariable.Widget s01uielt9=
   mVariable.new Widget("com.askfm:id/
   register_disclaimer","android.widget.
   TextView","","","0","true","false","
   true","false", "By clicking Sign up,
   you agree to our Terms of service");
MCrawlerVariable.State s1 = mVariable.new
   State("com.askfm.RegisterActivity",
   false, false,"0_0_8",s01uielt0,
   s01uielt1,s01uielt2,s01uielt3,s01uielt4
   ,s01uielt5,s01uielt6,s01uielt7,
   s01uielt8,s01uielt9);
```

```
        MCrawlerVariable.State s1_current =
           getCurrentState();
        assertTrue(MCrawlerFunctions.isEqualState(
           s1, s1_current));
        MCrawlerVariable.Widget t0uielt0 =
           mVariable.new Widget("com.askfm:id/
           register_uid","EditText","","", "", "",
           "","","", "pseudoTest2");
        MCrawlerVariable.Action t0_action_0=
           mVariable.new Action("edit text",
           t0uielt0);
        MCrawlerVariable.Widget t0uielt1 =
           mVariable.new Widget("com.askfm:id/
           register_password","EditText","","", ""
           , "","","","", "test1010");
        MCrawlerVariable.Action t0_action_1=
           mVariable.new Action("edit text",
           t0uielt1);
        MCrawlerVariable.Widget t0uielt2 =
           mVariable.new Widget("2","MenuView","",
           "", "", "","","","", "");
        MCrawlerVariable.Action t0_action_2=
           mVariable.new Action("menu",t0uielt2);
        MCrawlerVariable.Widget t0uielt3 =
           mVariable.new Widget("com.askfm:id/
           register_email","EditText","","", "", "
           ","","","", "s.zafimiharisoaopenium.fr"
           );
        MCrawlerVariable.Action t0_action_3=
           mVariable.new Action("edit text",
           t0uielt3);
        MCrawlerVariable.Widget t0uielt4 =
           mVariable.new Widget("com.askfm:id/
           register_fullName","EditText","","", ""
           , "","","","", "s.zafimiharisoaopenium.
           fr");
        MCrawlerVariable.Action t0_action_4=
           mVariable.new Action("edit text",
           t0uielt4);
        performAction(0L,t0_action_0,t0_action_1,
           t0_action_2,t0_action_3,t0_action_4);
        MCrawlerVariable.Widget s011uielt0=
           mVariable.new Widget("com.askfm:id/
           register_disclaimer","android.widget.
           TextView","","","0","true","false","
           true","false", "By clicking Sign up,
       you agree to our Terms of service");
    MCrawlerVariable.Widget s011uielt1=
       mVariable.new Widget("com.askfm:id/
       register_language","android.widget.
       TextView","","","0","true","false","
       true","false", "English");
    MCrawlerVariable.Widget s011uielt2=
       mVariable.new Widget("com.askfm:id/
       register_birthDate","android.widget.
       TextView","","","0","true","false","
       true","false", "");
    MCrawlerVariable.Widget s011uielt3=
       mVariable.new Widget("com.askfm:id/
       register_email","android.widget.
       EditText","","","0","true","false","
       true","false", "s.zafimiharisoaopenium.
       fr");
    MCrawlerVariable.Widget s011uielt4=
       mVariable.new Widget("com.askfm:id/
       register_password","android.widget.
       EditText","","","0","true","false","
       true","false", "test1010");
    MCrawlerVariable.Widget s011uielt5=
       mVariable.new Widget("com.askfm:id/
       register_fullName","android.widget.
       EditText","","","0","true","false","
       true","false", "s.zafimiharisoaopenium.
       fr");
    MCrawlerVariable.Widget s011uielt6=
       mVariable.new Widget("com.askfm:id/
       register_uid","android.widget.EditText"
       ,"","","0","true","false","true","false
       ", "pseudoTest2");
    MCrawlerVariable.Widget s011uielt7=
       mVariable.new Widget("com.askfm:id/
       register_facebook","android.widget.
       Button","","","0","true","false","true"
       ,"false", "Sign up with Facebook");
    MCrawlerVariable.Widget s011uielt8=
       mVariable.new Widget("com.askfm:id/
       register_signup","android.widget.Button
       ","","","0","true","false","true","
       false", "Sign up");
    MCrawlerVariable.Widget s011uielt9=
       mVariable.new Widget("android:id/
       content","FrameLayout","","","0","true"
```

```
                ,"false","true","false", "");
            MCrawlerVariable.State s11 = mVariable.new
                 State("com.askfm.RegisterActivity",
                false, true,"0_0_8",s011uielt0,
                s011uielt1,s011uielt2,s011uielt3,
                s011uielt4,s011uielt5,s011uielt6,
                s011uielt7,s011uielt8,s011uielt9);
            MCrawlerVariable.State s11_current =
                getCurrentState();
            assertTrue(MCrawlerFunctions.isEqualState(
                s11, s11_current));
        }
    }
```

7.1.2 Crash test

1. **Security Report.**
2. **Junit Crash replayer.**

7.1.3 Injection test

1. **SQL injection report**.

2. Exemple of security report where injection was successed is shown below. It is the obtained result after testing the Notpad application where the implementation of the content provider have been modified. The injection ***’ WHERE _ id = 1;DROP TABLE notes*** have been inserted and deleted the note table. Afterward, the application could not access to the note table anymore.

```
Number of tested location : 1
Number of detected injection vulnerability: 1
===========================================================

Test Number: 1
Target location : NoteEditor
Code coverage: 44
Vulnerability status : is vulnerable
Error description :

com.example.android.notepad.test.
    InjTest:INSTRUMENTATION_RESULT: shortMsg=
    android.database.sqlite.SQLiteException
INSTRUMENTATION_RESULT: longMsg=android.database.
    sqlite.SQLiteException: no such table: notes
INSTRUMENTATION_CODE: 0
```

```

Path to reach the target location
source: com.example.android.notepad.NotesList
menu  0


List of executed  events
edit text with value ' WHERE _id = 1;DROP TABLE
    notes; in the field com.example.android.
    notepad:id/note
menu element number : 0
================================================================
```

7.1.4 Brute force test

1. Security Report

7.2 Budget application

1. **Report**. A summary of the model report is presented below :

```
Application package: com.siri.budgetdemo
Execution time: 24963
Number of test: 255
Code coverage:

Activity coverage: 37.0 %
Number of detected bugs: 0
================================================================

================================================================
```

   255 tests have been executed during 24963 seconds.

2. **Storyboard**. The light storyBoard is presented in figure 7.1.4. The tree storyBoard is presented in figure 7.1.4.
3. **STS models.**
   - The STS tree and the minimised STS tree is illustarted in Figure 7.1.4 where the initial model and its minimisation have respectively 74 and 38 locations.
   - The Extrapolated models are illustrated in 7.1.4 where the extrapolated model and its minimisation have respectively 44 and 35 locations .

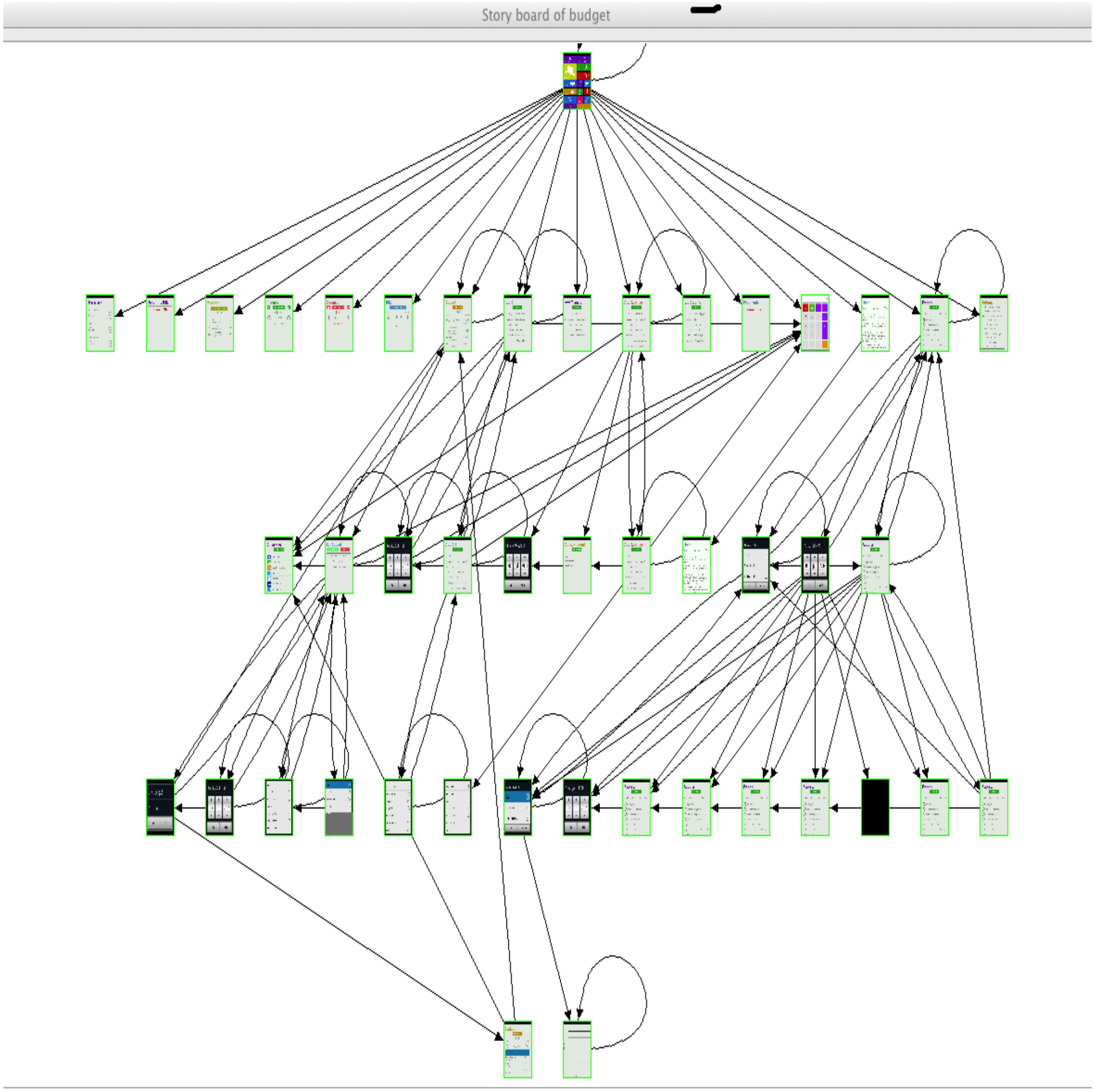

Figure 4 – The StoryBoard of Android Budget application

8 Annexe

8.1 Some useful commands

Commands	Definition
mct -h	Help command, it will detail the different option that must be setted
mct explore	explore the Application
mct stress	perform crash testing
mct inj	perform injection testing
mct bruteforce	perform bruteforce testing
mct storyBoard	show the story board of the application
mct display STS	display the STS model of the application
mct display MinSTS	display the minimised STS model of the application
mct display ExSTS	display the Extrapolated STS model of the application
mct display MinExSTS	display the minimised Extrapolated STS model of the application
mct activityCoverage	give information about the activity coverage

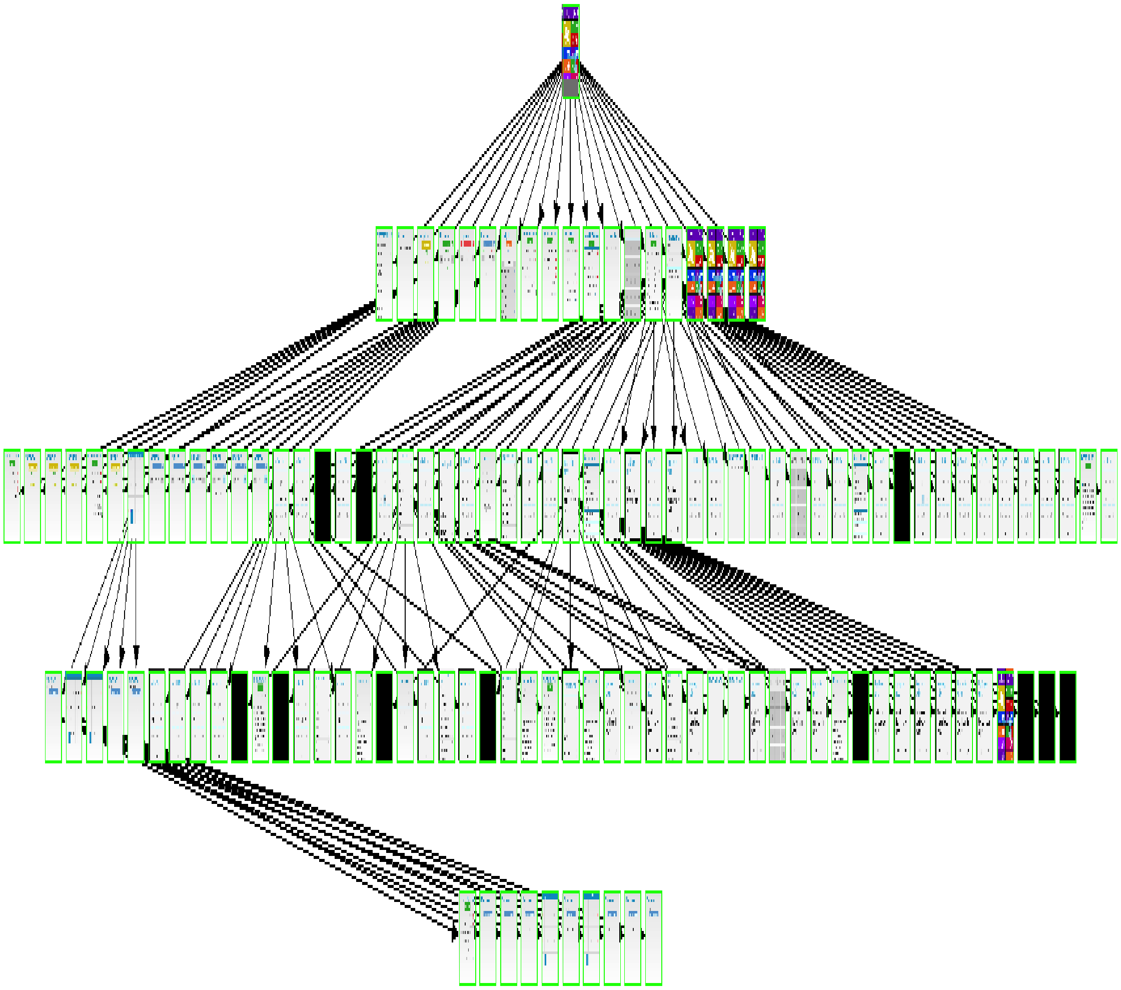

Figure 5 – The Tree StoryBoard of Android Budget application

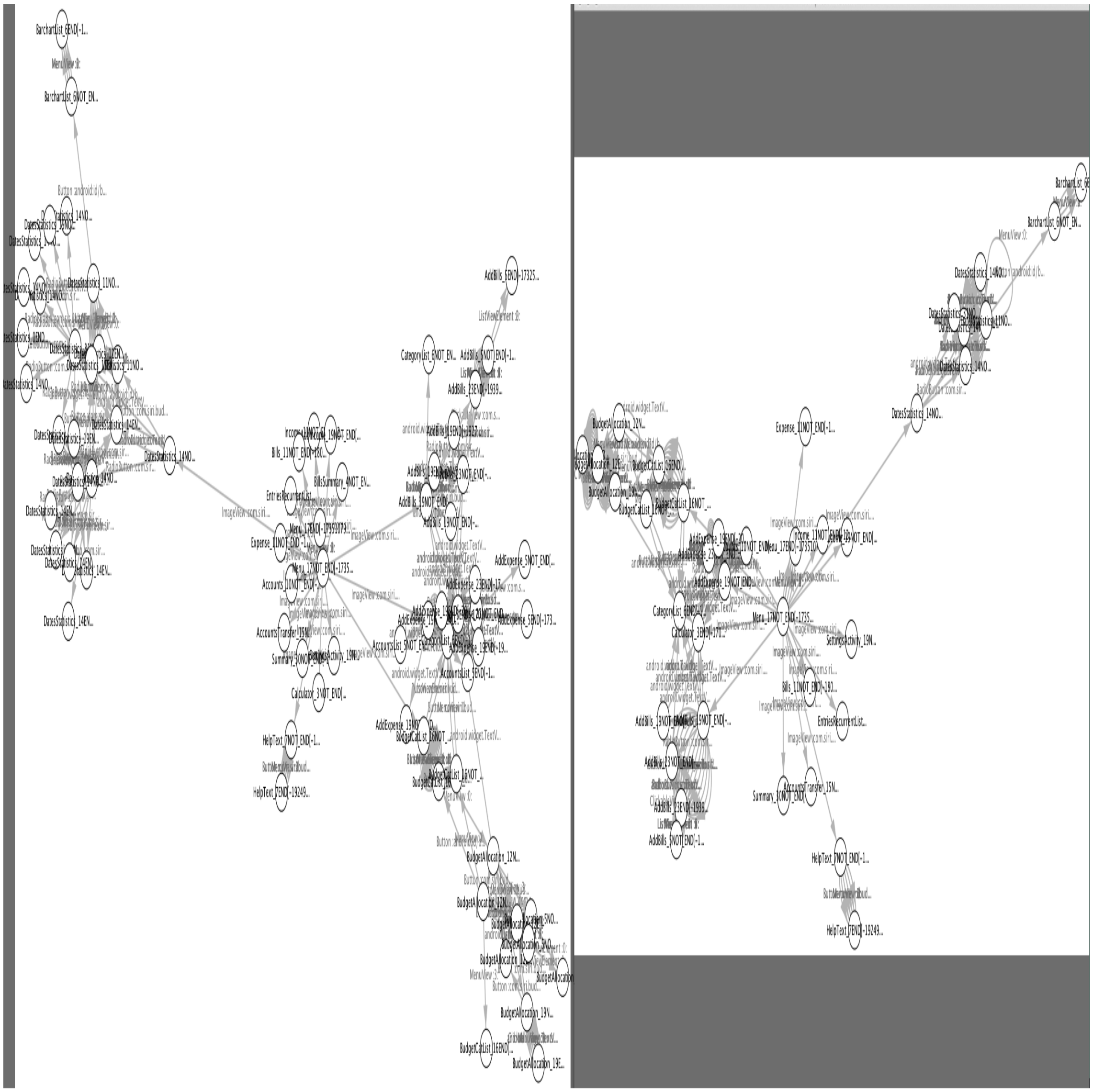

Figure 6 – The STSs models of Android Budget application

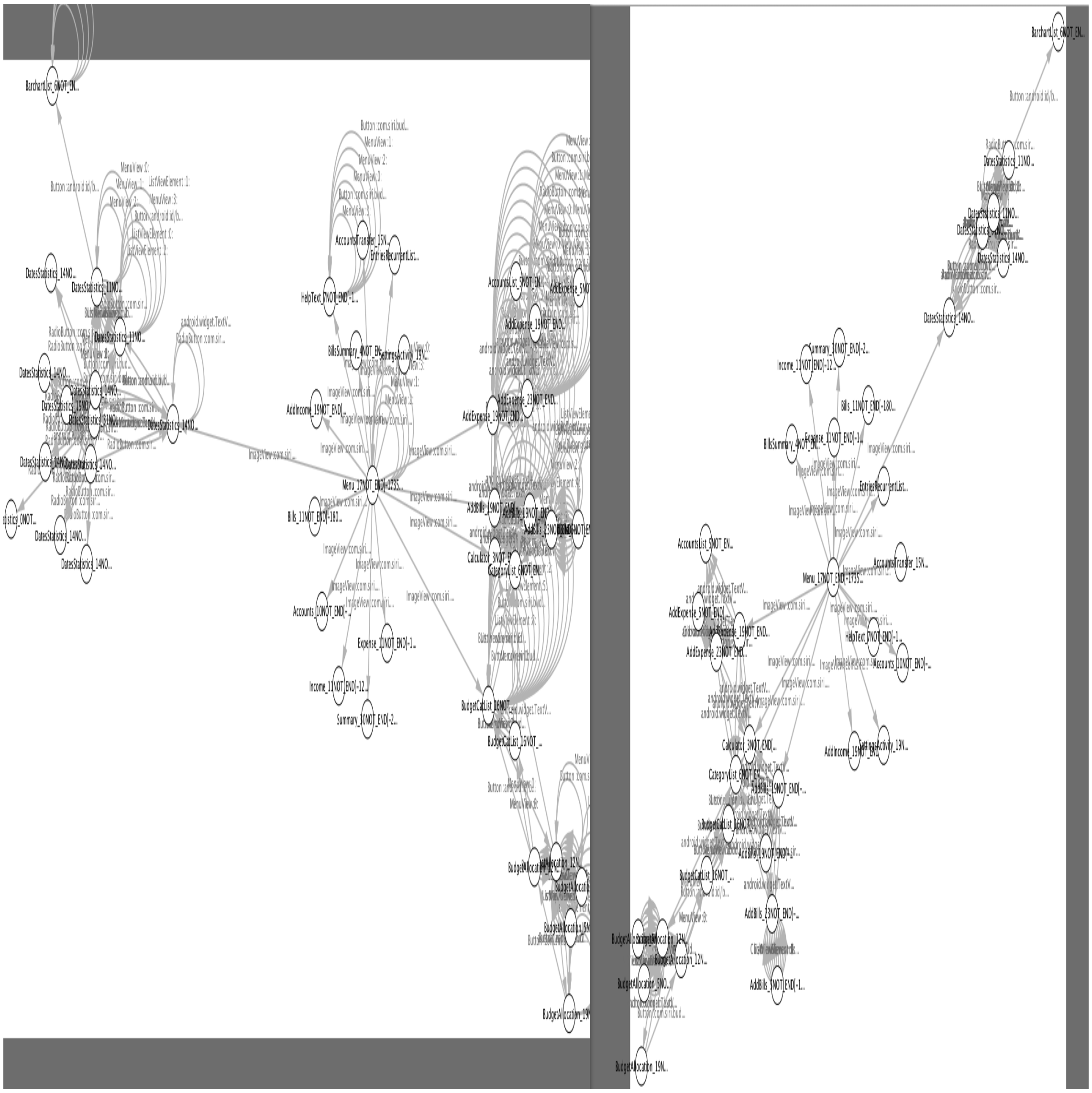

Figure 7 – The extrapolated STSs models of Android Budget application