Amateur Radio Astronomy: with Software Defined Radio

David Kirwan, Alan Davy, and John Ronan

Waterford Institute of Technology,
Dept of Maths and Physics,
Cork Rd, Waterford City, Ireland
dkirwan@tssg.org
http://www.wit.ie

Abstract. The abstract should summarize the contents of the paper and should contain at least 70 and at most 150 words. It should be written using the *abstract* environment.

Keywords: radio astronomy, software defined networking, signal processing

1 Introduction

1.1 Background

I use the AsyncTask in order to make a REST call [article, 1988, p. 215] to the Pacemaker HTTP API, on a separate thread than the main UI thread. When the network operation is complete the synchronous side of the AsyncTask incorporates the Mediator pattern as it informs the main PacemakerActivity.set_json(json) with the response from the asynchronous part of the operation. [Goossens, Mittelbach, and Samarin, 1993]

1.2 Research Problem Statement

Json txt returned from the API rest call, is parsed into a Ruby Hash data structure. The iterator and composite patterns are used within the application, in order to traverse through the Pacemaker Users data structure, and perform an operation. Such as the calculation of distance for all activities for a particular user.

1.3 Motivations

The application contains a background service which registers the application to receive network change events. When the Android device connects to a network, or mobile broadband, the operating system kicks off a broadcast event which is received then by the WifiReceiver BroadcastReceiver. The receiver class initiates a simple message being displayed on the main UI thread by way of a Toast message indicating whether it is a WIFI or Mobile broadband connection. The chain of responsibility pattern is used here, the OS gives the opportunity for more than one subsystem to handle a particular event in the system. While removing the costs associated with this notification.

1.4 Research Objectives

I have implemented a simple OnClickListener class which calls a function when activated. This class is an adapter between the Ruboto framework and the underlying Android OS. It allows a Ruby method within the Pacemaker Activity class to conform to the Android View.OnClickListener interface and then be executed when an end user touches a button in the UI.

1.5 Conclusions

The facade pattern is used to provide an interface between the UI and the rest of the application. In the Ruboto framework, they recommend an antipattern approach of developing the UI inside the Activity controller, it is optional however as it does provide the means to use the existing Android XML UI files.

2 Section Two

September 2014

Something

David Kirwan
00346128
dkirwan@tssg.org

3 Section Three

4 Section Four

5 Section Five

6 Section Six

Bibliography

article. some article about stuff. *some big important journal*, 5(2):9–83, 09 1988. Some note.

Michel Goossens, Frank Mittelbach, and Alexander Samarin. *The LaTeX Companion*. Addison-Wesley, Reading, Massachusetts, 1993.

A Testing McTest

Here is some content in the appendix

1..1 How I became inspired

...