Llama

Marco Gavanelli

December 10, 2013

1 Introduction

We all have smartphones, with an enormous amount of available applications to choose from. Some of them are games, some are nice, some are just cool. Few of them are indeed *useful* on a daily base.

My favourite application is called *Llama*, which stands for *Location Aware Mobile Application*, and beside having a cute llama icon, it is indeed one of the most practically useful applications in my smartphone.

I understand llama as a rule based application, although the author possibly does not know. In fact, the user can define rules (although the author calls them *events*), that are activated when certain conditions are met; when a rule is activated, it can perform actions that change the state of the phone, or that execute programs.

As an example, these are some of the rules I wrote:

> "If I am at home, and the current time is between 2 and 4pm, activate *quiet mode*"

as my 4 years-old child is probably taking his nap.

Another rule is

> "If I am at work, and on my calendar there is a scheduled activity that contains the string 'lesson', activate *silent mode*"

Or

> "If I am at work, activate the wireless"

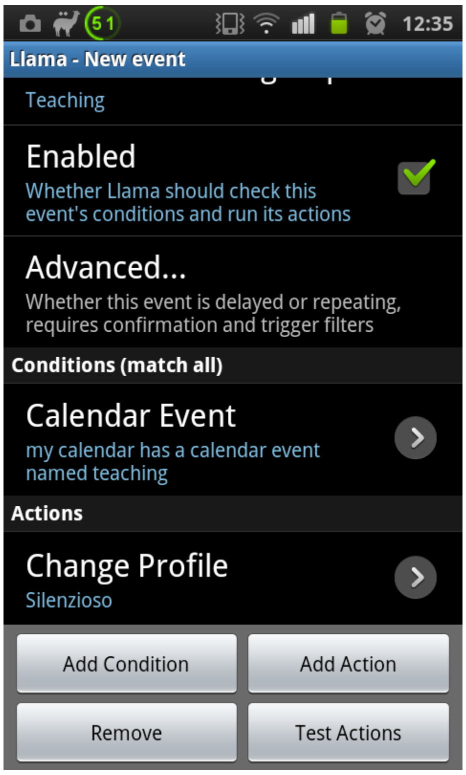
and

> "If I am leaving work, de-activate the wireless"

as during my travel back home there is no wireless connection, so I can save some battery power.

More precisely, each rule has a precondition, that is a conjunction of the events listed in Table 1 (note however that there is an 'OR' condition, that is used for disjunctions of events), and it has a conclusion, that is a conjunction of actions. The possible actions are listed in Table 2; some names refer to states one can activate/deactivate (e.g., 'bluetooth' means that you can activate or deactivate bluetooth).

Now, how does Llama know where I am? It does not use the GPS (by default), because it would require too much energy, and quickly drain my battery.


12:35

Llama - New event

Teaching

Enabled

Whether Llama should check this event's conditions and run its actions

Advanced...

Whether this event is delayed or repeating, requires confirmation and trigger filters

Conditions (match all)

Calendar Event

my calendar has a calendar event named teaching

Actions

Change Profile

Silenzioso

Figure 1: Definition of an ‘event’ (a rule) that changes the mode to ‘silent’ when the calendar contains an event named ‘teaching’

Active application
Airplane mode
Alarm time
App Notification Bar Icon
Audio becoming noisy
Battery level
Bluetooth Device connected/disconnected/not connected
calendar event
Call state
Car mode
Charging or unplugged
Day of the Week
Desk dock
Enter/in area
Leave area / not in area
Llama variable
Mobile data connection
Mobile data enabled/disabled
Mobile network ID
Music playback
NFC tag detected
OR
Phone reboot
Roaming
Screen on/off
Screen rotation
Signal strength
Time between
User is present
WiFi Hotspot
WiFi Network Connected/disconnected

Table 1: List of conditions that can trigger events

2G/3G/4G
Account sync
Airplane mode
Android intent
APN
Bluetooth
Car mode
GPS
Haptic feedback
Kill Application
Llama Android location polling
Llama Bluetooth Polling
Llama notification icon
Llama profile Changes lock
Llama variable
Llama WiFi Polling
Local plugin
Media Player
Mobile data
Music/media volume
Play a sound
Profile
Queue another event
Reboot
Reminder
Run Application
Screen Brightness
Screen lock
Screen lock password
Screen on/off
Screen rotation
Screen timeout
Speak
Speakerphone
Turn off phone
USB mass storage
Vibrate
Wallpaper
WiFi
WiFi hotspot
WiFi sleep policy

Table 2: List of actions that can be triggered by rules.

Instead, it uses the approximate localization, that is based on the cell my mobile is currently connected to. Initially, I have to train the llama to recognize my location: I can do so by defining a new location (e.g., *work*, *home*, etc.) and declaring to llama that I will be in such location for some time (e.g., I promise I will stay at home for the next hour). Llama will record all the cells identifiers the phone will be connected to in the next hour, and associate them to location *home*.

The problem is, what happens if my home is in the same cell as (or shares some cell with) my workplace? Then both locations become *active*, and all the rules that are activated in one of the two locations will trigger. As a workaround, the author proposes to remove the common cells from both locations, so we are sure that the events that trigger on entering the location are not activated. But, what about the events activated when *leaving* the location? Do we, as logic programmers, have better ideas?

Also, there are problems if we have conflicting rules. In this case, every 4-5 seconds one of the rules triggers (e.g., the phone switches from silent to loud modes and vice-versa every 4-5 seconds). Maybe a logic programmer could propose a better way to deal with these situations.

There are other, similar applications, like Tasker. I could not find one that relies on Logic Programming, or that has a clear semantics for the rules.

Also, there are applications that do the same for your computer, like Cuttlefish, that allows you to select the default printer depending on the network you are connected to, or to lock/unlock your computer when a particular USB key is inserted/removed.

As a conclusion, I think that these applications are very useful in everyday life, and too bad the LP community did not come up first in proposing these types of applications! However, maybe we can provide new ways to develop these applications, show that having rule systems having a semantics can help in situations with conflicts, and possibly provide rule engines that are more efficient, that drain less energy from the battery, or that are more stable.