

Roadmap for developing Python-openmoko stack

Main Author: Vivek Khurana

Co Author: Sudharshan S

email: mailing.vivek@gmail.com, sudharsh@gmail.com

Project url: <http://projects.openmoko.org/projects/python-openmoko/>

Introduction

Openmoko is a platform to develop a completely open mobile phone platform. As a part of effort openmoko team is developing a software stack which will provide an interface to the mobile phone hardware.

Currently to build any application you need to be versed with C and openmoko internals as well as GTK+ toolkit, if GUI applications are to be produced. As it is not feasible for a general programmer to learn all the technology stacks for developing applications, we need to have some kind of a glue language that can provide an interface to all the subsystems within. The glue language should not have any steep learning curve but at the same time should be robust enough to run under a mobile phone environment. Adding a glue also hides the unwanted details of the openmoko internals and if aided with a good documentation, developers will find it easy to code applications using the openmoko platform

Python is a general purpose programming language with proven abilities. Python is a glue language which may be used to bind to all the technologies used by openmoko technology stack. Though python is working for the openembedded stack, which actually forms the basis of openmoko stack, we still see some major performance issues in form of speed of code execution and memory leaks.

Python-openmoko project aims at fixing the problems faced while executing python programs on openmoko platform, as well as ensuring productivity for developers owing to its proven robustness. Secondly, the project aims at developing bindings for different sub-systems available within. Thirdly, we aim at developing a mechanism for generating, ipkg format packages (similar to .deb in debian derived distros) from python programs to allow ease of deployment of packages to the mobile phone.

Goals of the project

- 1) To speed up the execution of python on openmoko platform
- 2) To ensure the stability of python on openmoko project
- 3) Develop bindings for the different sub systems of openmoko project
- 4) Develop wrapper classes for libmokoui so as to ease the development of GUI apps on openmoko using python
- 5) Develop an interface for developing openmoko packages (ipk files) from python projects
- 6) Develop or find a suitable interface designer to design apps for openmoko platform.

7) Maintenance of python on openmoko project.

Speeding the code execution

This phase is concerned with speeding the code execution of the project. Main tasks for this project will be

- 1) Identify the minimum set of modules required to get python working with all the subsystems
- 2) Figure out a way (if possible) to reduce python to only an interpreter and strip compiler out of the python distribution.
- 3) Find out ways of optimizing python at the compile time.

Time required: 4-5 weeks

Expected delivery: Mid-End of October 2007

Ensure stability of python on openmoko

Main tasks for this phase are

- 1) Figure out any memory leaks that might be there in the python core or python bindings.
- 2) Figure out bottlenecks in python code execution.
- 3) Ensure overall code stability by regression testing the python code execution cycle.

Time required: 3-5 weeks

Expected delivery: Mid October 2007 to end of October 2007

Develop bindings for openmoko subsystems

Some python bindings are already in place and work is going on for others. These bindings have been developed by developers not associated with python-openmoko project.

Main task of this phase is to identify the bindings already developed, test the stability of the bindings, bug fix them and include them in python-openmoko distribution. Secondly, we will develop bindings for the subsystems which are not available.

Time required: 4-5 weeks

Expected delivery: Last week of november 2007 or First week of december 2007

Develop wrapper classes for libmokoui

Libmokoui provides the GUI interface and since almost every application on a

mobile phone will need a GUI we need to have wrapper classes around libmokoui. Libmokoui wrappers will be specified in another document. These wrapper will be built on the lines of python for symbian OS or PyKiwi framework.

Purpose of these wrappers is to develop set of classes which will reduce the amount of code required to do oft-repeated operations like creating the main window, creating a status bar etc.

Time required: 4-5 weeks

Expected delivery: End of November 2007

Develop an interface for generating ipkg packages from python programs

Openmoko platform allows deployment of third party applications using ipkg format. Main task of this phase is to develop a utility or an interface which will package python applications to ipkg packages ready for deployment. This utility should allow creating clustered packages.

Time required: One week

Expected delivery: third week of October 2007

Develop or find a suitable interface designer to design UI for openmoko apps

Main task of this phase is to find a way to design UI for openmoko applications. Glade project provides a method of developing UI for GTK+ application but it also requires libglade to be installed and the .xml file to be parsed by the same. This will have a slight overhead for the not so powerful systems like mobile phones.

We need an interface which will either generate the UI in python or generate the UI in a format which can be interpreted by python programs without requiring any special resources. Something like .ui file generated by Qt designer for Qt toolkit. By doing so, we would have significant improvements with regards to speed of execution

Time required: 5-8 weeks

Expected delivery: Last week of December 2007 or early-mid January 2008

Maintaining python on openmoko

This is an ongoing task and as and when new features are added to openmoko platform, we would add them to the python-openmoko project.

For any comments or suggestions please contact,

Vivek Khurana: mailing.vivek@gamil.com

Sudharshan S: sudharsh@gmail.com

References:

<http://www.openmoko.org>

<http://python-openmoko.projects.openmoko.org>
<http://www.python.org>