Development of a English-Nepali Bilingual Dictionary for Android

(Sabdakosh)

Praveen Acharya

July 2012

1. Acknowledgements

I would like to thank Mr. Bal Krishna Bal of the Department of Computer Science and Engineering, Kathmandu University for providing his valuable mentorship and guidance for the project. I would also like to thank the team of Dobhase[1] for providing the resources for the project. Also, I would like to thank Mr. Rohitman Amatya and Mr. Chandan Gupta for their help and support.

2. ABSTRACT

Today, mobile phones provide number of advanced functions, among which possibility of running mobile applications seems to be the most interesting feature. It changes a standard mobile phone into a mini size yet still powerful computer that is able to run various applications, like map guide, dictionary, games and so on.

This paper is a report to the bilingual English – Nepali dictionary named *Sabdakosh*. *Sabdakosh* serves the purpose of a dictionary for mobile phones running on Android[2]. This report delivers the developmental process of *Sabdakosh*. It covers the framework, functional aspects followed in building *Sabdakosh*.

The project incorporates the four main components of building the dictionary.
1. An interface for the query input.
2. A Database for serving dictionary content.
3. A service for the interaction between the queries and the database.
4. A interface for displaying the result.

The main goal of this project is to develop a English – Nepali bilingual dictionary which is expected to meet the needs of wide range of users such as language learners, native speakers.

---

[2] http://www.android.com

3. INTRODUCTION

Dictionaries are commonly found in the form of books, electronic portable devices, CD-ROM or available online. A bilingual dictionary is a dictionary that translates words from one language to another. Bilingual dictionaries can be either mono-directional, meaning it translates only from language A to language B, or can be bi-directional, translating to and from both languages.

The national language of Nepal is Nepali with English being used prevalently for communicational purpose with non Nepali speakers. As these two languages are widely used, mono-directional English - Nepali translation dictionary is vital to avoid unnecessary difficulties by native speakers when traveling, studying languages, communicating with others and in many other situations. The idea about the project came up seeing the lack of English-Nepali bi-directional mobile dictionary compared to the amount of mobile dictionaries available in other 'world' languages.

Dictionary is an indispensable utility for every individual, but the convenience is however compromised when expected to carry paper dictionary or gadget around to have a quick access to words and their definitions. Besides, with paper dictionaries, user has to turn the pages of dictionary to search for a specific word. Because of the very large cost, a dictionary which is written on paper has few opportunities for revisions. For CD-ROM or online dictionaries, they are not fully portable as they require computers and Internet connection to use them.

So, a mobile dictionary seems to be a good alternative in all these cases. It is portable, has quicker access to the words and can be periodically revised. In the following sections, we will look into the development of a mobile dictionary (*Sabdakosh*).

4. Introduction to Android Framework[3]

Android is a software stack for mobile devices that includes an operating system, middleware and key applications. The Android SDK provides the tool and the APIs necessary to begin developing applications on the Android platform. Applications are written using the Java programming language and run on Dalvik, a custom virtual machine specifically designed for embedded use. Dalvik runs on top of a stripped down Linux kernel.

Android was originally developed by Google. Android was designed to be distributed based on an open source license and to compete against other mobile platforms, including those from Microsoft, Research In Motion and Apple Inc. Google also announced the founding of the Open Handset Alliance (OHA), a consortium of 47 technology companies devoted to advancing open standards for mobile devices. Android isn't currently completely open source but a large portion of it is. Google and the OHA is committed to eventually making Android 100% open source.

Architectural Overview

The Android environment is built on top of a Linux kernel and includes a set of C/C++ libraries, the Android Runtime environment, an Application Framework and a set of core applications as described in the diagram below.

Linux Kernel

At the base of the Android environment is a stripped down Linux Kernel. Linux is used to communicate with the mobile phone's hardware and Android supports multiple phone processors. Much of the Android work though happens in the layers above the OS.

---

[3] http://en.wikibooks.org/wiki/Android/Introduction

Android Runtime

The Android runtime environment consists of a set of core libraries and virtual machine. The core libraries provide most of the functionality available in the core libraries of the Java programming language.

On conventional computing devices software runs directly on the operating system kernel but an Android application runs in its own process, with its own instance of the Dalvik virtual machine. Dalvik has been written so that a mobile device can run multiple instances of it in an efficient manner. Dalvik is not your typical JVM though it's very similar. The Dalvik VM executes files in the Dalvik Executable (.dex) format which is optimized for minimal memory footprint. The VM is register-based, and runs classes compiled by a Java language compiler that have been transformed into the .dex format by the included "dx" tool.
The Dalvik VM relies on the Linux kernel for underlying functionality such as threading and low-level memory management.

Libraries

Android includes a set of C/C++ libraries used by various components of the Android system. These capabilities are exposed through the Android application framework. Some of the core libraries are listed below:
- System C library - a BSD-derived implementation of the standard C system library (libc), tuned for embedded Linux-based devices
- Media Libraries - based on PacketVideo's OpenCORE; the libraries support playback and recording of many popular audio and video formats, as well as static image files, including MPEG4, H.264, MP3, AAC, AMR, JPG, and PNG
- Surface Manager - manages access to the display subsystem and seamlessly composites 2D and 3D graphic layers from multiple applications.
- LibWebCore - a modern web browser engine which powers both the Android browser and an embeddable web view
- SGL - the underlying 2D graphics engine
- 3D libraries - an implementation based on OpenGL ES 1.0 APIs; the libraries use either hardware 3D acceleration (where available) or the

included, highly optimized 3D software rasterizer
- FreeType - bitmap and vector font rendering
- SQLite - a powerful and lightweight relational database engine available

Application Framework

Developers have full access to the same framework APIs used by the core applications. The application architecture is designed to simplify the reuse of components; any application can publish its capabilities and any other application may then make use of those capabilities (subject to security constraints enforced by the framework). This same mechanism allows components to be replaced by the user.

Underlying all applications is a set of services and systems, including:
- A rich and extensible set of Views that can be used to build an application, including lists, grids, text boxes, buttons, and even an embeddable web browser
- Content Providers that enable applications to access data from other applications (such as Contacts), or to share their own data
- A Resource Manager, providing access to non-code resources such as localized strings, graphics, and layout files
- A Notification Manager that enables all applications to display custom alerts in the status bar
- An Activity Manager that manages the life cycle of applications and provides a common navigation back-stack

Android Framework
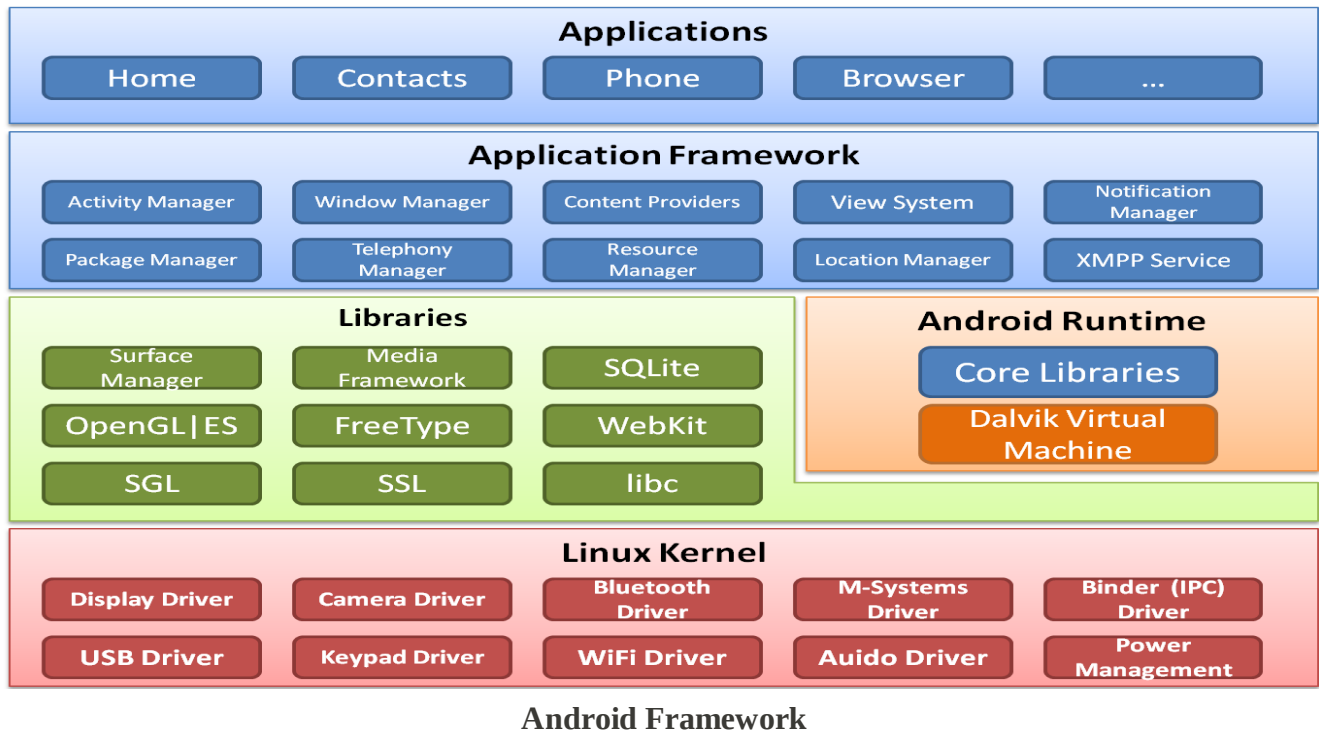
5. Development of the Dictionary (*Sabdakosh*)

Sabdakosh is a bilingual dictionary application which is intended to run on Android phones. The main components in building a dictionary can be classified as:

1. An interface for the query input.
2. A Database for serving as the dictionary content.
3. A service for the interaction between the queries and the database.
4. A interface for displaying the result.

The Android SDK provides the tool and the APIs necessary to begin developing applications on the Android platform. Applications are written using the Java programming language. The Android SDK tools compile the code along with any data and resource files into an *Android package*, an archive file with an .apk suffix. All the code in a single .apk file is considered to be one application and is the file that Android-powered devices use to install the application.

The main components in the building of the dictionary are addressed in the following section.

An interface for the query input:

A dictionary should have an interface which the user can interact with. This interface should be able to take in the query and do the processing. *Sabdakosh* has a class *Sabdakosh Activity* which handles the query input. This class is the main activity of the dictionary and is called when the application runs. It invokes a search dialogue either by pressing the search button present in the hardware or by selecting search from the menu. The query than can be typed in the search dialogue which is used for further processing.

A Database for serving as the dictionary:

Database contains all the necessary data required for the functioning of the dictionary. The dictionary database is created using SQLite database. SQLite is an Open Source Database which is embedded into Android. SQLite supports standard relational database features like SQL syntax, transactions and prepared statements. In addition it requires only little memory at runtime. Hence is ideal for mobile uses.

Sabdakosh has a class *Dictionary Database* which creates the database from a resource. The database of *Sabdakosh* contain an English word and its translation in Nepali. The final result of the corresponding query in the dictionary would be the part of the database.

A interface for displaying the result:

Once the database has been created we must be able to display the result of the query that the user wishes to know. The displaying of this result is made through and interface which is handled by the class *Result Activity*. This activity displays the output one it has been retrieved from the database through some mechanism.

A service for the interaction between the queries and the database:

Search interface, result interface and database being built there mus be a mechanism which is able to communicate between these three components in order for the proper functioning of *Sabdakosh*. The query form the search interface must be passed into the dictionary database which looks for the matching entry in the database. Once the entry has been identified it must retrieve it in order to display it on the result interface. This communication between the components is handled by the class *Dictionary Provider*. This class interconnects the components of the dictionary and provides the user with the translation of the query.

Sabdakosh
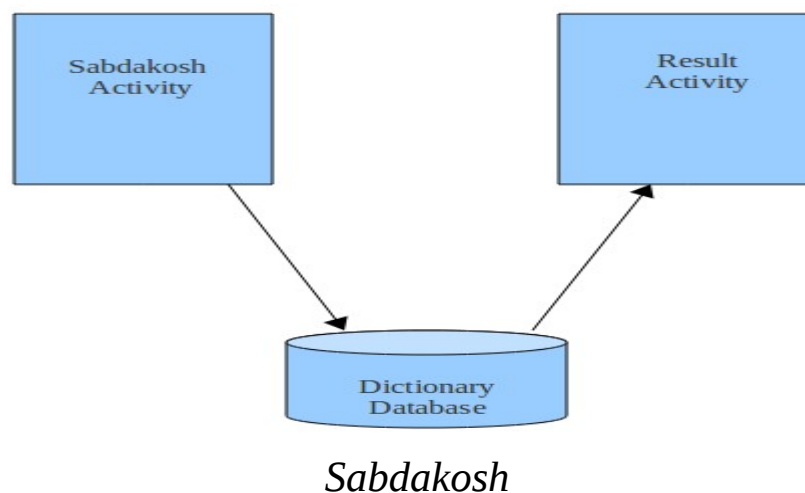
4. CONCLUSION

A bilingual dictionary being built it can be of great use to the users. It can assist them in learning language. It will also help in improving ones language skills. Foremost it will be an effective tool when traveling and communicating with people.

References:

Android Developers