UNITY 7: GUI BAESD DESIGN PLATFORM FOR UBUNTU TOUCH MOBILE OS

Asif Sardar Shaikh[1], M.U. Inamdar[2]

[1]*M.E, Second year student, Department of E&tc, Siddhant COE, Maharashtra, India*
[2]*Assistant Professor, Department of E&tc, Siddhant COE, Maharashtra, India*

Abstract

Day by day the mobile industry is growing. Along with its mass usage the applications used in these devices are also increasing. So it is of utmost importance that the GUI should be fast, fresh and user friendly. Unity 7 is the GUI which provides the platform for the Touch interface for the Ubuntu OS. It falls under the category of open source; hence it doesn't required runtime licenses. Unity is not a collection of applications but is designed to use existing programs. Traditional Linux uses command-driven interface (or text-based interface). By adopting the X-Window technology, graphical user interface (GUI) is available for Linux. It may be Menu-driven or Icon-driven. This paper shows the complete design of the Unity 7 GUI which includes the designing and implementation of Application framework, Application manager. This paper also shows the complete source code for the designing of the unity 7 GUI, its step by step building along with the different components used to build the Unity 7.

Keywords: *Unity , Unity 7 architecture, Application framework and Source code.*

----------***----------

1. INTRODUCTION

Unity is part of the Ayatana project, an initiative to improve the user experience within Ubuntu [9]. Unity is a shell interface for the GNOME desktop environment developed by Canonical Ltd. for its Ubuntu operating system. In addition to Unity, there are Application Indicators and other projects such as MeMenu, the notification system and the application Notify-OSD gathered.

Unity has a large community of developers writing applications that extend the functionality of devices, written primarily in a customized version of the Qt/C++ programming language. Ubuntu's open nature has further encouraged a large community of developers and enthusiasts to use the open source code as a foundation for community-driven projects, which add new features for advanced users.[9]

1.1 What is Unity?

As aforementioned unlike GNOME, KDE Software Compilation;

'Unity is not a collection of applications but is designed to use existing programs.' **[3]**

BAMF stands for Bash Application Matching Framework. It refers to the UNIX shell itself but they two mostly are independent. So Unity, in other words, just changes the clothes of the existing program and its appearance to the user.

2. UNITY 7 ARCHITECTURE [3]

The Ubuntu touch architecture is similar to the Linux system architecture. But the only change is in the application (or utility) layer and the GUI. Fig.1 shows the software stack

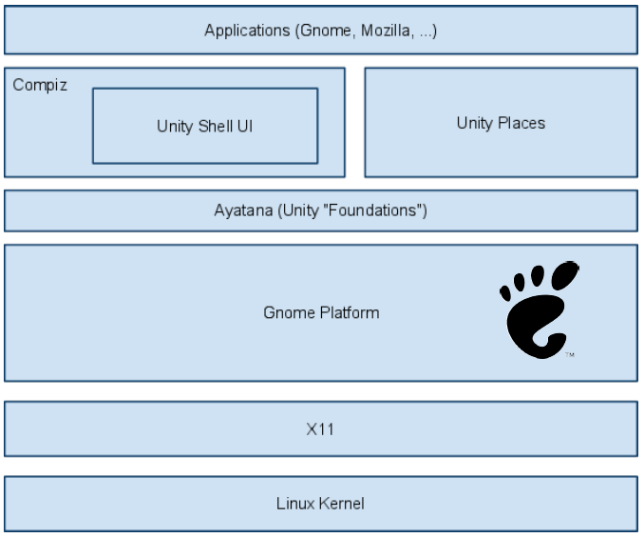

Fig -1: Unity 7 software stack [2]

It consists the layered architecture from bottom to all the way up. They are Linux kernel, X11, Gnome, Unity entities and applications etc.

2.1 Linux Kernel [1][8]

Today's operating systems provide two fundamental services for users. First, they make the mobile hardware easier to use. Second, an operating system shares hardware resources among users. One of the most important resources

is processor. The Operating systems are broadly classified according to their structuring mechanism. The structuring mechanism (or kernel) decides the execution speed, application framework, packaging style of the codes etc. The Ubuntu uses Monolithic kernel based Operating System.

The **Ubuntu touch** specifically consists Linux **kernel version 2.6.24**. It performs all the similar functions performed by the kernel in a Linux OS.

2.2 X11

In computing, the X Window System (X11, X, and sometimes informally X-Windows) is a windowing system for bitmap displays, common on UNIX-like operating systems [9].

X provides the basic framework, or primitives, for building such GUI environments: drawing and moving windows on the display and interacting with a mouse, keyboard or touch-screen. X does not mandate the user interface; individual client programs handle this. Programs may use X's graphical abilities with no user interface. As such, the visual styling of X-based environments varies greatly; different programs may present radically different interfaces [6].

2.3 Unity Foundations & Entities

Unity entities are explained in the sections B & C.

2.4 Utilities

Utility programs provide user most of the functionalities of an operating systems. Utility includes framework for the applications such as Wi-Fi, Wi-max, Bluetooth, Documents, and Camera etc. along with their network/device drivers.

2.5 Device/Network Drivers [11]

Associated with each physical device or virtual device is a piece of code, called device/network driver device, which manage the device/networking hardware. The main functions driver: Setting up hardware on initialization. It brings the associated devices into and out of services. It receives the data from the hardware and passes it back to the kernel. Further, it Sends data from the kernel to the device and Detects and handles errors.

2.6 Applications

According to the needs and requirements the applications are designed and layered in the architecture.

3. UNITY APPLICATION FRAMEWORK

It has a unique application framework; it is broadly divided into 3 parts

- Unity-home-applet
- Unity-status-bar-plugin
- Unity-navigator-plugin

Where, an **applet** is a piece of code that can be used for graphical user interface.

A **plug-in** is a piece of software that interacts with another piece of software. Any given plug-in is directly associated with a specific piece of software and extends the functionality of the original software.

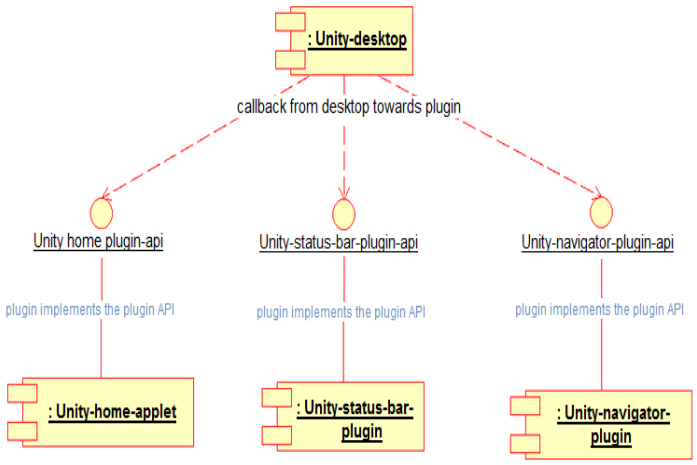

Fig – 2: Unity 7 interface architecture [2]

The **Unity 7** Application Framework is one of a few existing frameworks designed for small devices and is a good candidate for tablet use. It has strong support from Nokia and has been separated from Maemo to become part of GNOME Mobile.

This has allowed the Ubuntu community and others to contribute in a way that benefits all users. **Unity 7** mobile interface is the main UI component of the Maemo UI. It consists of two panels and the Home area.

Each panel can house plug-ins and basically function like on an ordinary desktop environment except for that this user interface was designed for touch-screen.

3.1 Unity Application Manager

Unity 7 Application Manager is the Unity 7 graphical package manager; it uses the Debian package management tools APT (Advanced Packaging Tool and dpkg) and provides a graphical interface for installing, updating and removing packages. It is a limited package manager, designed specifically for end-users, in that it doesn't directly offer the user access to system files and libraries.[7] With the Diablo release of Maemo, Unity 7 Application Manager now supports "Seamless Software Update" (SSU), which implements a variety of features to allow system upgrades to be easily performed through it.[2]

3.2 Application Framework for Different Modules

Following few diagrams show the application framework for different modules,

3.2.1 BAMF [3][11]

BAMF stands for Bash Application Matching Framework. It refers to the UNIX shell itself but they two mostly are independent.

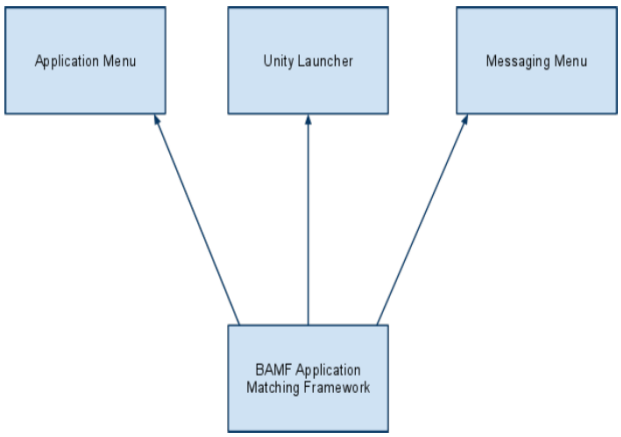

Fig – 3: BAMF application matching framework [2]

3.2.2 Application Indicator Framework

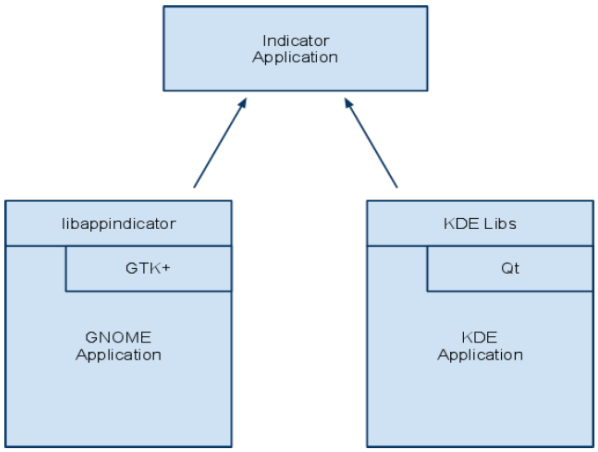

Fig – 4: Application Indicator framework [2]

3.2.3 Application Menu Indicator

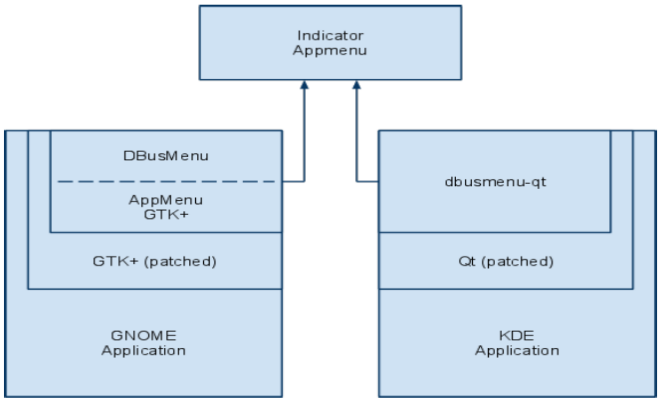

Fig – 5: Application menu Indicator framework [2]

3.2.4 Application Message Indicator

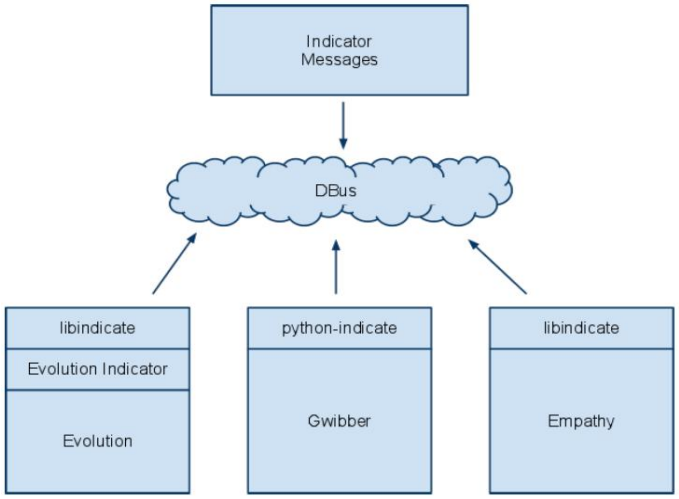

Fig – 6: Application message Indicator framework [2]

3.2.5 Top-Edge Indicator (Example)

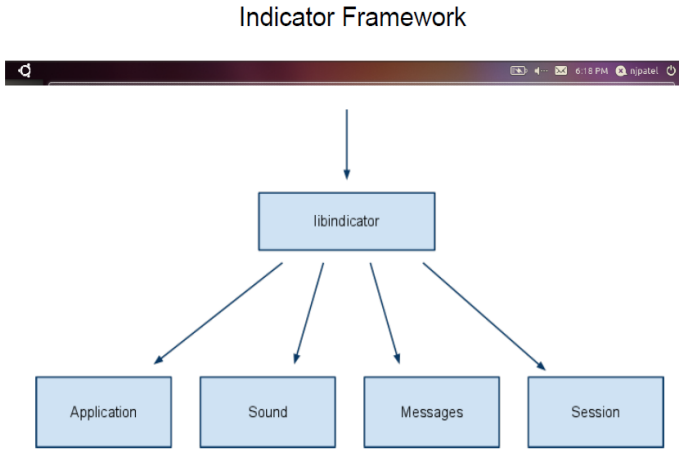

Fig – 7: Top-edge Indicator framework

3.2.6 System Sound Indicator (Example)

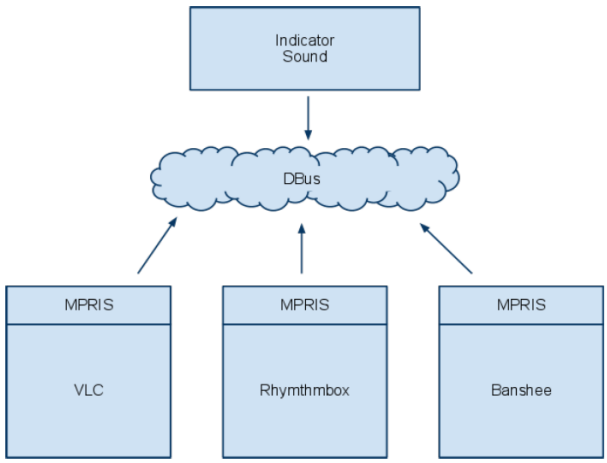

Fig – 8: System sound Indicator framework [2]

3.2.7 Application Menu Indicator (Example)

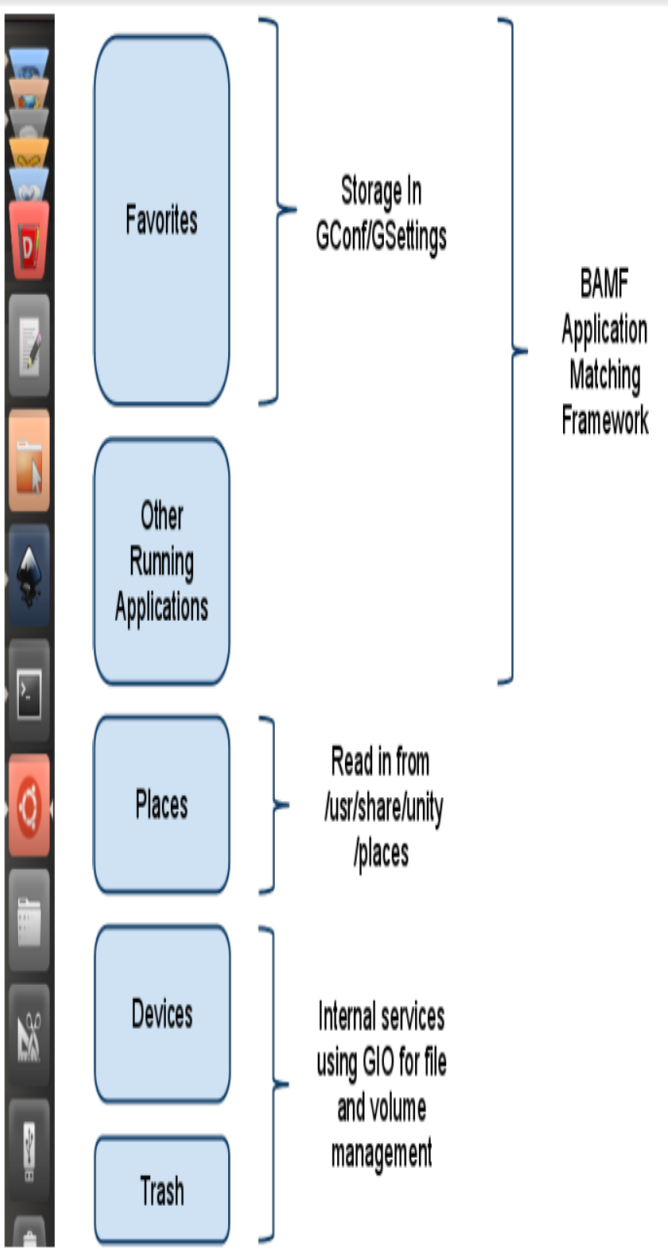

Fig – 9: Quick-list application menu Indicator framework [2]

4. PROFILING AND DESIGNING [2][10]

This section contains the pseudo codes and flowcharts for some of the important functions. The actual names of functions and arrays have been mentioned in Italian letters against tasks that they perform.

4.1 Flow Chart for the Unity 7 (the Graphical User Interface)

There are several important components needed to be designed for the Unity 7 GUI.[1] The required components are shown in the flowchart below,

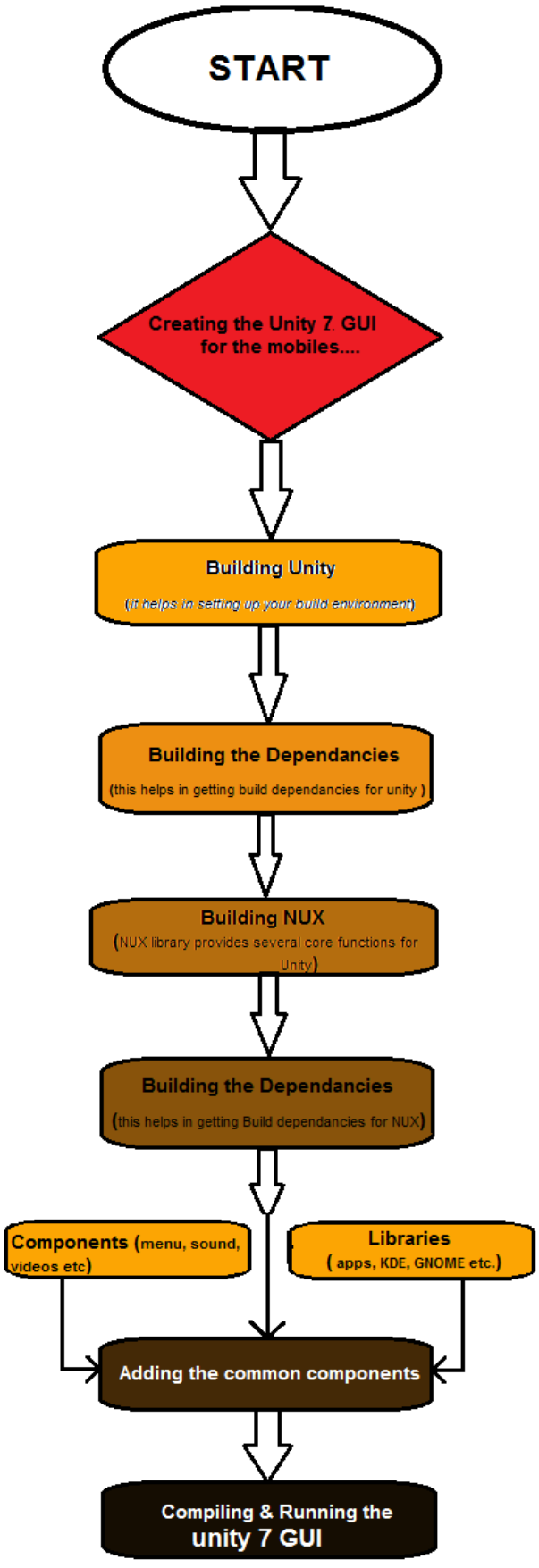

Fig – 10: Code structure for Unity 7 GUI

4.1.1 Building Unity 7 & its dependencies

1. Open terminal using command *(ALT+CNTRL+T)*
2. Go into the Super user do mode *($ sudo :)*
3. Install the Bazaar.
4. Use the trunk id
5. Make the directory for the code, using the command

   - *mkdir -p ~/code/unity*
   - *cd ~/code/unity*

6. Build-dependencies for unity, using the command

   - *sudo apt-get build-dep unity*

7. Create the bash file in your home directory called *(.bash_functions :)*
8. Compile Unity code using the file created in step no. 7
9. Now run this in a terminal:

   - *echo ". ~/.bash_functions" >> ~/.bashrc*

   Now it is possible to run:
   - remake-unity;

From the *trunk/* ; directory we created earlier.

4.1.2 Building NUX & its dependencies

At the time of writing, we can't compile unity without first building NUX from sources first.

1. Open terminal using command *(ALT+CNTRL+T)*
2. Go into the Super user do mode *($ sudo :)* & skip the step 3 from previous.
3. Use the trunk id.
4. Make the directory for the code, using the command

- *mkdir -p ~/code/nux*
- *cd ~/code/nux*
- *bzr branch lp:nux trunk*
- *cd trunk*

5. Build-dependencies for NUX, using the command

- *sudo apt-get build-dep nux*

6. Compile NUX code, using the command

- *sudo apt-get build-dep nux*

4.1.3 Adding the Components

The components are nothing but either the applications or their interface to the existing applications. Following are some of the applications which are designed for the mobile, viz. Menu Bar (which includes Message Menu, Sound Menu, Date/Time Indicator, User Menu, Device Menu etc.) and Dash (which includes Applications Lens, Applications Lens, Music Lens, Video Lens etc.)

Showing one of the examples of message the outline for the other components remains same.

1. Open terminal using command *(ALT+CNTRL+T)*
2. Go into the Super user do mode *($ sudo :)* & skip the step 3 from previous.
3. Use the trunk id
4. Compile message code, using the command

   - *bzr branch lp:indicator-messages*

4.1.4 Adding the Libraries

Interfaces for Unity shell are provided by the *libunity* client library, including classes and resources for building Lenses and Scopes for the Dash, methods for adding count and progress information to an application's icon in the Launcher, and an interface for adding media player controls and meta-data to the Sound Menu.

Library includes following components

4.1.4.1 DBusMenu

The DBusMenu library is the foundation for both the global menu and HUD in Unity. It allows an application to export a menu structure over DBus where it can be accessed by other applications and services. Toolkit bindings for Gtk and Qt will automatically add window menus for applications using them to DBusMenu, where they can be read by the Global Menu and HUD.

1. Open terminal using command *(ALT+CNTRL+T)*
2. Go into the Super user do mode *($ sudo :)* & skip the step 3 from previous.
3. Use the trunk id
4. Compile message code, using the command

   - *bzr branch lp:dbusmenu*

   - *bzr branch lp:appmenu-gtk*

   - *bzr branch lp:dbusmenu-qt*

4.1.4.2 libappinidcator

Unity displays system and application indicators in the top panel of the shell. These indicators provide a common interface to show and change the status of an application or Unity itself.

1. Open terminal using command *(ALT+CNTRL+T)*
2. Go into the Super user do mode *($ sudo :)* & skip the step 3 from previous
3. Use the trunk id
4. Compile message code, using the command

   - *bzr branch lp:libappindicator*

   - *bzr branch lp:libindicate*

   - *bzr branch lp:libindicator*

4.1.4.3 Bamf

The Bamf Application Matching Framework monitors the opening and closing of all windows on the desktop, assigning each to an "application" group that is associated with a .desktop file. This allows Unity to easily match a Windows with meta-data from its Application, such as icon, name and executable. Bamf if used by the Unity Launcher, Application Switcher and Spread.

1. Open terminal using command *(ALT+CNTRL+T)*
2. Go into the Super user do mode *($ sudo :)* & skip the step 3 from previous.
3. Use the trunk id
4. Compile message code, using the command

- *bzr branch lp:bamf*

5. SOURCE CODE LAYOUT [4][5]

This section shows the complete layout for the different source files.

5.1 Source Code for Unity [3]

The Unity source code is divided between common, reusable components and the Compiz plugins that display them. Here is an overview of where the different source files reside in the branch.

5.1.1 Top-Level Indicator

- **./dash**
  Code for the Unity Dash

- **./hud**
  Code for the Heads-Up Display (HUD)

- **./launcher**
  Code for the Unity Launcher

- **./panel**
  Code for the Unity (top) Panel

- **./plugins**
  Compiz plugins that make up the Unity Shell (See below)

- **./services**
  DBus services for the Unity panel

- **./unity-shared**
  Collection of utility classes shared by multiple Unity components

- **./unity-standalone**
Stand-alone mode for Unity testing (without compiz)

- **./UnityCore**
  Base set of classes/resources for integrating with system services (indicators, lenses, etc)

5.1.2 Compiz Plugins

- **./plugins/unityshell**
  The primary compiz plugin for the Unity Shell, provides the
  Launcher, Dash, switcher and panel

- **./plugins/unity-mt-grab-handles**
  Provides the overlay for touch-friendly window moving and
  resizing

- **./plugins/unitydialog**
  (Inactive) New style modal dialogs

5.2 Source Code for Libraries

Library includes following components

5.2.1 DBusMenu [2]

Library includes following components

- **dbusmenu/libdbusmenu-glib**
  GLib bindings to convert to and from DBusMenu

- **dbusmenu/libdbusmenu-gtk**
  Gtk bindings to convert to and from libdbusmenu-glib

- **appmenu-gtk/src**
  Automatically exports Gtk window menus using libdbudmenu-gtk

- **libdbusmenu-qt/src**
  Qt bindings to convert to and from DBusMenu

5.2.2 libappinidcator

- **libappindicator/src**
  Client library for adding an indicator for an application

- **libindicate/libindicate**

  Client library for interacting with the Message Menu indicator

- **libindicator/libindicator**
  Used by Unity to find and display Indicators in the panel

5.2.3 Bamf [3]

- **bamf/src/src**
  The bamf daemon which allows other Unity components to
  query information via dbus

- **bamf/lib/libbamf**
  Client-side library that provides an API to communicate with the bamf daemon

6. FEATURES

1. Unity is a touch-friendly interface which is based on Qt and QML. It provides the direct interface to the applications in single touch [2].

2. Since, Unity Utilizes libhybris; he system can often be used with Linux kernels used in Android, which makes it easily ported to most recent Android smartphones [2].

3. Unity is a kind of interface which just changes the appearance of user interface. Users can access the whole system by swiping from the edges of the screen. The left edge allows for instant access to applications pinned to the launcher, and swiping all the way across reveals the home, which displays applications, files and contacts. This menu is available from the home screen and any running app [3].

4. It utilizes the same core technologies as the Ubuntu Desktop, so applications designed for the latter platform run on the former and vice versa [6].

7. CONCLUSIONS

Unity is the GUI which provides the platform for the Touch interface. 'Unity is not a collection of applications but is designed to use existing programs[8][1]. It falls under the category of open source; hence it doesn't required runtime licenses. The code can be prepared using the simple C code and applications can be designed using Qt and QML soft wares. BAMF stands for Bash Application Matching Framework. It refers to the UNIX shell itself but they two mostly are independent. So Unity, in other words, just changes the clothes of the existing program and its appearance to the user.

REFERENCES

[1]. White paper on: - "Mobile OS and efforts towards open standards" By Dotcom Info-way.
[2]. Unity & Ayanta Architecture; Copyright© 2008, 09- 11; Documentation Project by; Neil patel & Ted Gould
[3]. Ubuntu Mobile Guide; Copyright© 2004,05, 06 Canonical Ltd. and members of Ubuntu Documentation Project
[4]. Linux (2008 Edition); Christopher Negus; published by:-Wiley Publishing, Inc.
[5]. Beginning Linux Programming (4th Edition); Neil Matthew; published by:-Wiley Publishing, Inc.
[6]. Buxton, W. (2002). Human Input to Computer Systems: Theories, Techniques and Technology
[7]. Jason Ansel, Kapil Arya, and Gene Cooperman. DMTCP: Transparent Checkpointing for Cluster Computations and the Desktop. In 23rd IEEE International Parallel and Distributed Processing Symposium, Rome, Italy, May 2009
[8]. "Android operating system"author Yaseer Ahmed Faraaz ahmed, in Aalim Muhammed salegh college of engineering
[9]. F. De Pellegrini, I. Carreras, D. Miorandi, I. Chlamtac, and C. Moiso. R-p2p: a data centric dtn middleware with interconnected throwboxes. In Autonomics '08, pages{10. ICST.}
[10]. A. Vahdat and D. Becker. Epidemic routing for partially connected ad hoc networks. Technical report, Duke University, 2000

BIOGRAPHIES

Asif Sardar Shaikh is the 2nd year M.E. student of Siddhant college of engineering. He is pursuing the degree in E&tc (VLSI & EMBEDDED) under the university of Pune.

Prof. M.U. Inamdar is the senior most faculty member in department of E & tc in Siddhant college of engineering. He has completed his degree in the digital electronics from Govt. College of engineering, Pune. He has the 24 years of experience in the same field