International Journal of Scientific Engineering and Technology (ISSN : 2277-1581)
Volume No.3 Issue No.7, pp : 943-945 1 July 2014

Design & Development of Automated Validation Framework for Graphics Display Driver

Shilpa Hadimani[1], DR.Padma.S.K[2], Mr.Sridharan.M[3]
[1,2]Dept. of Information Science and Technology,– Mysore, Karnataka, India
[3] Project Manager, INTEL Corporation, Bangalore, India.
E- Mail:Shilpa.hadimani23@gmail.com

Abstract- : Graphics is an unavoidable part of computer. With increase of different architectures and features of graphics, Graphics Drivers are also becoming complex. To meet this pace, driver validation also needs to be updated. The proposal is to automate validation process in systematic approach to achieve optimization, efficiency, reduce manual efforts and to increase accuracy. This automation includes objective assessment of different graphics Driver feature.

Keywords- Framework, Graphics Driver, Display Switching, Automation

I. INTRODUCTION:

Graphics Driver is an interface between Operating System and GPU Hardware. Graphics Driver plays important role for communication between Operating System and Hardware. Each Graphics Driver has written for specific Graphics Hardware .Graphics Hardware might support specific feature, if there is no Graphics Driver to support it, then the feature is useless.

Software validation helps to ensure software quality and to improve software reliability. With the volume and complexity of software applications continue to increase, people's requirements on software quality are also rising; the function of software validation in the process of software development has become more and more important, while software validation effort also appears to be increasingly difficult.

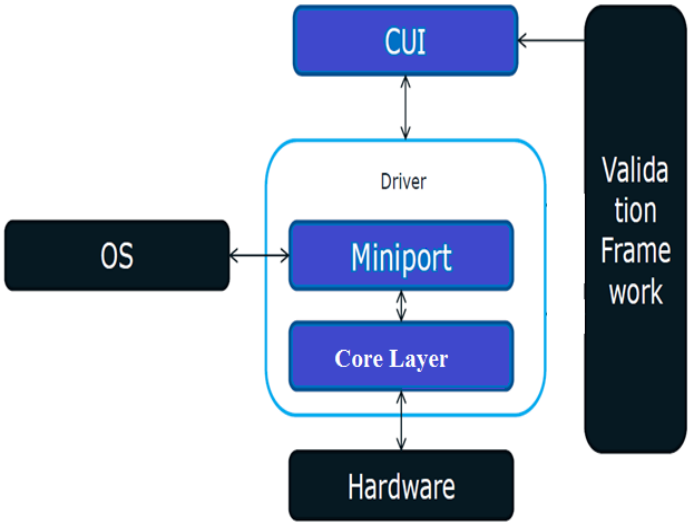
The focus is on Automation of Graphics Display Driver. Currently these types of tests are executed manually so they need more manual efforts, time and accuracy is also not guaranteed .The Proposal is to automate the validation process or test related to Graphics Display Driver using different Hardware, Software tools and Algorithms. The validation process can be automated which can reduce the manual efforts, reduce the cycle time and increase accuracy[1].

II. RELATED TECHNOLOGY :

GRAPHICS DRIVER ARCHITECTURE

Main Components of Graphics Driver

A. Intel Core Layer
B. Miniport
C. CUI



Figuare : Graphics Driver Architecture

A. Core Layer:

Core Layer is one of the important component in Graphics Driver. Intel Core Component is related to Hardware and it will do all registry setting operations . It is a Low level Display Component and it manages Display Pipe, Planes, Ports and also manages handles Mode Set, display devices etc.

B. Miniport:

Miniport component handles all OS interfaces. Primary interface for the Intel core layer. A miniport is a type of Hardware – Driver, part of the Windows Driver Model. This component of driver is responsible for adapter initialization, interrupts, MMIO mapping etc. Entry point for all IOCTL calls from OS.

C. CUI :

Common user interface(CUI) that allows users to interact with electronic devices through graphical icons and visual indicators such as secondary notation, as opposed to text-based interfaces, typed command labels or text navigation.This is the only user Interface module of Driver which is customizable and provide localization support.

III.DESIGN OF DISPLAY AUTOMATION FRAMEWORK

This framework[2] determines different quality metrics on different configuration and system and on the basis of these quality metrics the quality of Graphics Display Driver component verified. The configuration and system are selected in such a way that as it covers all types of end user experience.

➢ ***Interaction with graphics driver***

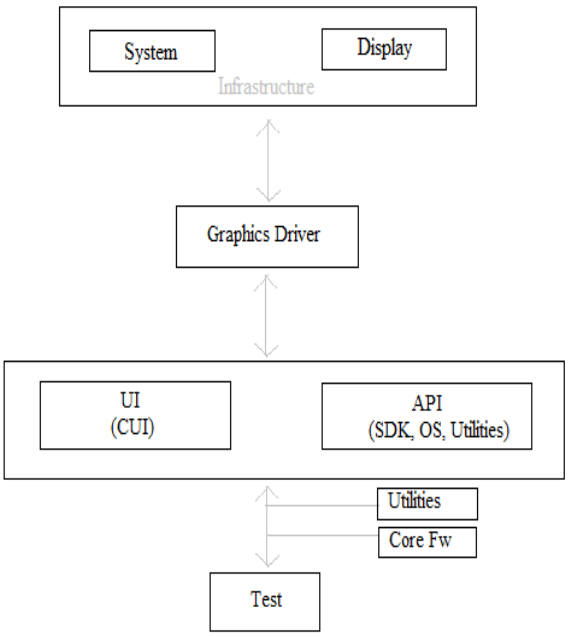


Figure : Framework interaction with Driver.

Framework interacts with graphics driver using
Driver's API's
OS API's
Using Test Players , same as user interacts with driver.

➢ ***Automation Model***

Possible Input:

Automation tests can have manual inputs or it can also be leveraged as plug and play functionality.

Expected Output:

Logs, Reports or execution time can be possible output of automation test.

IV.RESULTS AND DISCUSSSION :

Results for Display Switching:

Display Switching is a Switching between many Displays. There are many Display Configurations like

- Single Display
- Clone Display
- Extended Display

The configuration can be changed using Graphics Driver and OS property page. Changing these configurations is Display Switching[4].

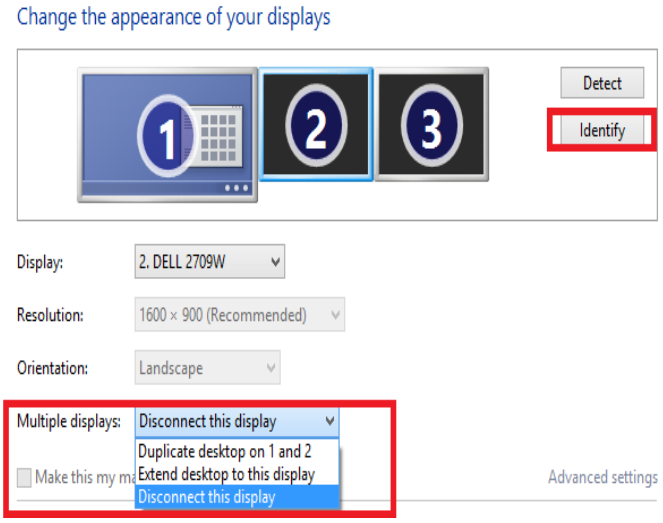


Figure: OS Screen resolution page options to switch between displays.

Expected Output :

The expectation is that driver should handle any display configuration change request from OS and CUI should immediately re
configuration if configuration change is successful.

Result :

Feature is successfully implemented in Framework. Automation framework is capable to validate this feature for any platform with any flavour of Windows OS.

Results for Deep Colour :

Deep colour is a term used to describe a gamut comprising a billion or more colours.

Types of Colour format:

- 8 bits per colour(24bit colour format)
- 10 bits per colour(30bit colour format)
- 12 bits per colour(36bit colour format)
- 16 bits per colour(48bit colour format)

Colour depths greater than 24 bits are defined as Deep Colour.

➢ Deep colour is supported for RGB 4:4:4 and YUV 4:4:4 pixel data.
➢ Panel must support Deep colour to use this feature.

8 bit colour/24 bits per pixel:

➢ 256 levels of gradation
➢ 256 x 256 x 256 = 16,777,216 possible colors:17 Million colours.

10 bit colour/30 bits per pixel:

➢ 1024 levels of gradation
➢ 1024 x 1024 x 1024 = 1,073,741,824 possible colours: 1 billion colours.

Expected Output :

Deep Colour supported panel must enable this feature after running Deep colour enable.Applicaion on that panel or in clone mode and not supported panel must be in the same state

Result:

Feature is successfully implemented in Framework. Automation framework is capable to validate this feature for any platform with any flavour of Windows OS with all Display Mode Combination.

V. ACHIEVEMENT :

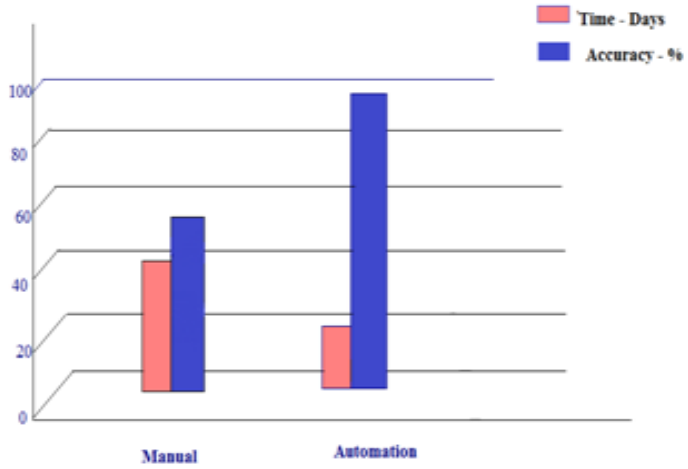


Figure: Automation Framework Achievement for Display Switching Feature.

VI .CONCLUSION:

Automation Framework makes user friendly platform to enhance Graphics Driver features. With the help of Automation Framework user can easily Automate Different Features of Graphics Driver. This reduces manual efforts and Time and also gives accurate results for different quality matrix.

ACKNOWLEDMENT:

The author wish to Thank ISE Department of SJCE – Mysore and INTEL Corporation -Bangalore, Karnataka, India for supporting this work.

REFERENCES:

i. *HTTP://WWW.INTEL.COM/P/EN_US/SUPPORT/HIGHLIGHTS/GRAPHICS /HDGRAPHICS.*

ii. *Alex Cervantes, "Exploring the Use of a Test Automation Framework"*

iii. *Jeff Bisgrove,Tom Jones ,"Integrated Test Facility (ITQ-Automation Testing to Support Intel's Manufacturing Output"*

iii. *Other Intel sources.*