ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 2, Issue 10, October 2013

DEALING WITH INIMITABLE ISSUES FOUND IN SOFTWARE UNDER TESTING

Pardha Saradhi. N. V[1], K. Durga Prasad[2], Dr. Sujoy Bhattacharya [3]

Student of M. Tech, Specialization – Software Engineering, Department of IT, Padmasri Dr BVRIT, Medak, India [1]

Assitant Professor, Department of IT, Padmasri Dr BVRIT, Medak, India[2]

Professor, Department of IT, Padmasri Dr. BVRIT, Medak, India[3]

Abstract: Inimitable issues create a big trouble to testers and developers in resolving them because for an Application under Testing (AUT) the issue once seen may not be seen in subsequent trials of testing. These issues when found at customer site are difficult to report to a tester or developer. This paper mentions about an effective tool called Screen Recorder (SR) which records the user actions performed on a desktop. Customer to report about a non – reproducible issue can record the scenario where the issue found and can forward videos to a tester or a developer. This paper explains simple algorithm which is used to perform the recording of user workflow according to the quality selected.

Keywords: Inimitable, issue, Screen Recorder, Tester, Developer, quality.

I. INTRODUCTION

Product quality is rapidly becoming a competitive issue where differences in quality reflect differences in the behaviour of functionality of a product. In fact, the early economic research on quality focused almost exclusively on durability, since they provide a stream of services over time, increased durability implies a longer stream of services — in effect, more of the software. To achieve this there should not be any kind of problems/issues with the application that was or being designed. However some bugs may be found, which the tester usually faces so called non – reproducible issues, which are often creates a big trouble to tester/developer. These kinds of issues are seen sometimes and sometimes not, which depends on the testing environment and the tester/developer. There are situations like after delivering the product or software, the customer reverts the product to its previous state i.e., to a tester because of the issues found at the site. But when tester tries to analyse the problem he may not find the issue even after following the same steps that were performed by the customer. There may be lot of problems in this process like testing environment is different from that of customer's, communication problem in following the same steps that of customer's, configuration problems etc. To avert these problems a tool called Screen Recorder is introduced to record actions that are performed by a user on a desktop. Using this tool a customer or tester need not have to do manual work on how he used the application rather they can just record the steps were followed and the video can be used for demonstration. The common problem users generally face with the videos is the size which will be a critical part in transferring them from one device to another. This tool has got the ability to record according to the quality /resolution that was specified by the user. There is an inter-relativity among video quality, time and size i.e., for a given quality level and the duration of recording the size of the video is calculated and shown, this resembles that a user can use the tool according to his desired parameters. This tool also works in two other modes so called Demo Mode – Application training purpose and Rolling Video Mode – Bug prediction. Tool uses Virtual Network Computing (VNC) which in-turn uses Remote Frame Buffer (RFB) protocol.

II. RELATED WORK

X11VNC: In computing, x11vnc is a Virtual Network Computing (VNC) server program. It allows remote access from a remote client to a computer hosting an X Window session and the x11vnc software, continuously polling the X server's frame buffer for changes.x11vnc keeps a copy of the X server's frame buffer in RAM. The X11 programming interface XShmGetImage is used to retrieve the frame buffer pixel data. X11VNC compares the X server's frame buffer against its copy to see which pixel regions have changed (and hence need to be sent to the VNC viewers.) Reading pixel data from the physical frame buffer can be much slower than writing to it (because graphics devices are not optimized for reading) and so a sequential pixel by pixel check would often be too slow.

To improve the situation, x11vnc reads in full rows of pixels separated by 32 pixels vertically. Once it gets to the bottom of the screen it starts again near the top with a slightly different offset. After 32 passes like this it has covered the entire screen. This method enables x11vnc to detect changes on the screen roughly 32 times more quickly than a sequential check would (unless the changes are very small, say only 1 pixel tall.) If the X11 DAMAGE extension is present, x11vnc uses it to provide hints where to focus its polling, thereby finding changes even more quickly and also lowering the system load.

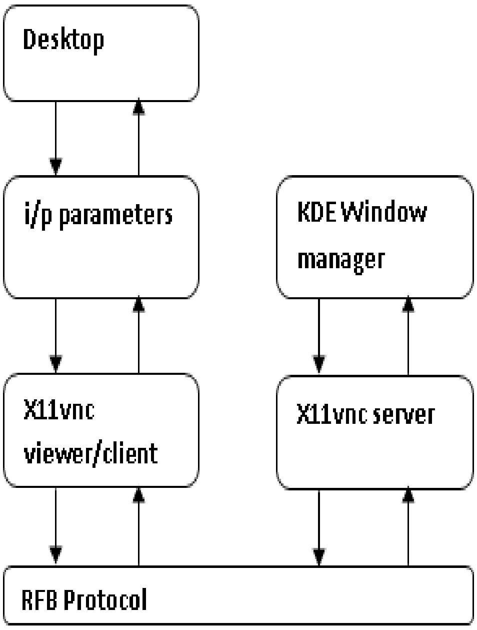

Fig: 1. Proposed architecture

III. Preliminary Sketch Of Protocol

The RFB protocol working consisting of responding to request from the client about a specific onscreen rectangle and then server responds in the form of the update consisting of an encoding the difference between the moment of the request and the last time the client requested data about this rectangle. Sending of information will lead to the high consumption of the bandwidth with the consequent delay in the process. To overcome this problem different encodings have been developed. Encoding refers to the format in which a rectangle of pixel data will be sent. Every rectangle of pixel data is prefixed by a header providing the position of the rectangle on the screen, the width and height of the rectangle, and an encoding type. This encoding type specifies the way of encoding of the pixel data. The data itself then follows using the specified encoding. These encodings are used to determine the way to transfer the graphical information. Adding new encodings developed by third parties does not compatibility with VNC applications that do not contain that new encoding. When a client wants to establishes communication with server both side must negotiate the encoding type to be use. If the client requires a non-existent encoding, the server will appropriate the next encoding available.

Following are the 4 basic types of VNC encoding can be used: Raw, RRE, Hextile, Zlib and Tight.

A. RAW:

RAW is the simplest encoding form. In this server sends all graphical pixels to the client and data consist in the form of width*height pixel values (where width and height are the width and height of the rectangle). This encoding method must be supported by clients. The process time used is minimal and the performance is very high when the server and the client are on the same machine. If the client is hosted in a remote device the performance is reduced due to the transfer of large amounts of data. This encoding is specifically designed for the low performing devices.

B. RRE:

RRE stands for Rise-and-Run-length-Encoding which consists of grouping consecutive identical pixels in order to send only the information of one pixel and the number of replications of that pixel. The basic intension behind RRE is the partitioning of rectangle of pixel data into sub-regions each of which consist of the single pixel value and union of which comprises the original rectangular region. It is a most effective method when large blocks of the same colour exist, like in patterns are to be send. There is a variant of the protocol which uses a maximum of 255x255 pixels to reduce the size of the packages. RRE rectangles arrived at the client in the form, which are easily rendered immediately and efficiently by simplest of any graphics engines.

C. Copyrect encoding (copy rectangle):

This encoding is a very simple and efficient encoding which is used when the client already has the same pixel data elsewhere in its frame buffer. The encoding on the wire simply consists of an X, Y coordinate. This gives a position in the frame buffer from which the client can copy the rectangle of pixel data. This can be used in a variety of situations, the most obvious of which are when the user moves a window across the screen, and when the contents of a window are scrolled. A less obvious use is for optimizing drawing of text or other repeating patterns.

D. Hextile:

This encoding divides the rectangles in the 16*16 tiles, allowing the dimensions of the subresctables to be specified in 4 bits each and 16 bits in total. The rectangle is split into tiles starting at the top left going in left-to-right, top-to-bottom order. This encoded data contents of the tiles simply follow one another in the predetermined order. If the width of the whole rectangle is not an exact multiple of 16 then the width of the last tile in each row will be smaller than that of previous tiles. Each tile is either encoded as raw pixel data, or as a variation on RRE. Each tile has a background pixel value, as before. No need to specify for any given tile if it is same as background of the previous tile.

IV. Working Of Screen Recorder

Tool works in three modes: Normal Mode, Rolling Video Mode and Demo Mode. All these modes are used to record the user performed workflow on the desktop but each one has got its own specific functionality.

A. Normal Mode

This mode is used to reproduce the inimitable issues. Here user can record the video using desired quality and duration, for which the size is calculated for the maximum operations that can be performed in that duration. This inter-relativity among video quality, duration and size can aware the user about the space that is going to be occupied by the video that is generated. By default an optimal resolution is selected in which the size and quality are optimized to certain level such that minimum amount of system space is occupied if recording is started by selecting the default quality level. An issue description is linked with the video generated which describes the reason to record that particular session.

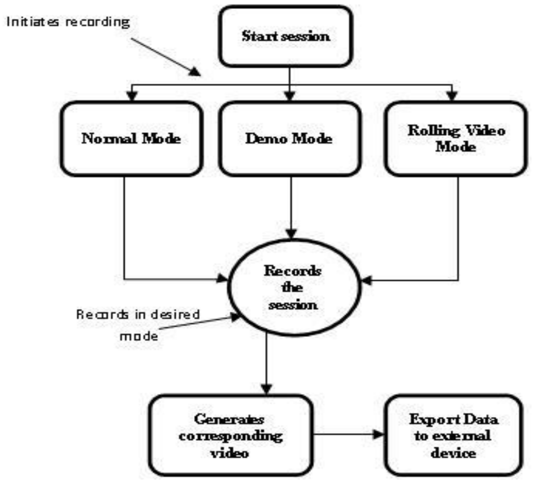

Fig: 2. Working of Screen Recorder

B. Rolling Video Mode

Rolling Video Mode: A tester/developer sometimes predicts that a bug may occur while testing a software/application which may be an inimitable bug and creates a big problem. In order to observe the functionality of Application under Testing (AUT), user can use this mode to capture its behaviour. In this mode, for a session, for an hour a package of six videos is created, each one is of 10min duration. So for a session only reasonable size of videos is created which may not be a problem for space complexity. For every hour this will be repeated and there is a limit of 5GB of size so that the videos which are created for the first hour of the session will be deleted if this limit is crossed. This will also be repeated for all the subsequent videos.

C. Demo Mode

Training on how to use an application/software can be given using this mode. In a real time environment there is a chance of demonstrating the usage of an application to several users for multiple times. To avert this situation this tool can be used in demo mode. A video is created and stored for demonstrating the usage of an application whenever needed. Otherwise, in an organization a link can be provided to the video in online training like sessions to train an individual.

This tool provides an added advantage is that the functionality of data export. A user can easily transfer the files from the local system to any other system even to a remotely located using USB, CD/DVD or FTP mechanism.

V. TESTING RESULTS

Below given is the sample screen shot taken in Normal Mode which is recorded in Normal Mode recorded with an optimal quality considered for maximum operations that can be performed for the corresponding session. The screen shot is taken in a Linux environment which is displaying the process of saving a file.

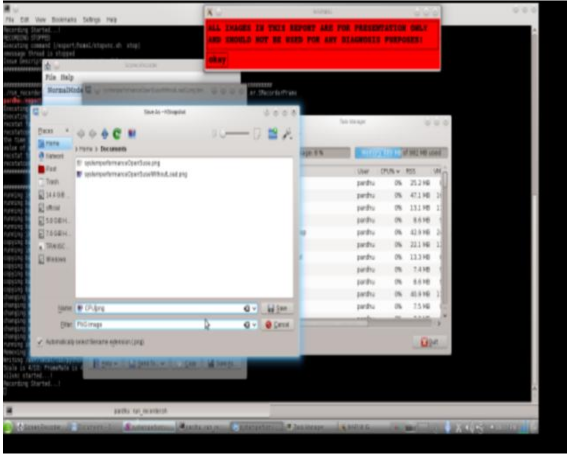
ALL IMAGES IN THIS REPORT ARE FOR PRESENTATION ONLY AND SHOULD NOT BE USED FOR ANY DIAGNOSIS PURPOSES!

Fig. 3.Sample screen shot in Normal Mode

VI. CONCLUSION

This tool is mainly focused on reproducing the inimitable issues which can be achieved by the two modes: Normal and Rolling Video Modes. Inter-relativity among video quality, duration and size will aware the user how much space is going to be occupied by the videos generated. In Rolling Video Mode chunks of videos are created which will occupy less amount of memory in the local system. Using Demo Mode training can be given on how to use an application.

VII. FUTURE ENHANCEMENTS

Using this tool, a user can remotely record a desktop with different levels of resolutions and can control or monitor it. Audio support can be enabled with possible minimum video size to avoid space complexity.

. REFERENCES

[1] Virtual Network Computing,Tristan Richardson, Quentin Stafford-Fraser,Kenneth R.Wood and Andy Hopper,Reprint from IEEE Internet Computing Volume 2, Number 1 January/February 1998.

[2] K.-J. Tan, J.-W. Gong, B.-T. Wu, D.-C. Chang, H.-Y. Li, Y.-M. Hsiao, Y.-C. Chen, S.-W. Lo, Y.-S. Chu, and J.-I. Guo, “A remote thin client system for real time multimedia streaming over VNC,” in International Conference on Multimedia and Expo (ICME), 2010, pp. 992–7.

[3] http://www.unixuser.org/~euske/python/vnc2flv/#related

[4] www.realvnc.com/docs/rfbproto.pdf

[5] www.realvnc.com/products/vnc/.../5.../VNCUser_Guide.pdf

[6] http://en.wikipedia.org/wiki/X11vnc