Project Viviz – Progress Report

Jun-Cheng Chen, Shivsubramani Krishnamoorthy, Bao N. Nguyen, Ching Lik Teo, Praveen Vaddadi, November 11, 2009

CMSC 734, Fall 2009

Copyright 2009, Project Viviz, University of Maryland
Project Sponsors

Professor Atif Memon, Department of Computer Science, University of Maryland

Professor Ben Shneiderman, Department of Computer Science, University of Maryland
Agenda

• Introduction – software testing and visualization
• Viviz – status updates
• Problems
Introduction

• Increasing complexity of software.
• Comprehensive software testing to weed out bugs.
• Solution: Automated Testing
  – Enormous amount test cases to execute
  – Large amount of results to analyze
• Visualization tool aids in rapidly identifying faults and potentially fatal errors
  – Aviation, defense and medical software
  – Large scale concurrent software
• **Human factor:** Tester’s experiences and skills
The Big Picture

Reverse Engineering → Testing Model Creation → Test Case Generation → Test Case Prioritization → Test Case Execution

Application

GUI Model
Event Flow Graph
Test Pool
Execution Result

Model Enrichment
Test Case Enrichment

Visualization Analysis

Continuous Feedback
Copyright 2009, Project Viviz, University of Maryland
GUITAR - Output

Event Graph

Test Cases
Vizviz – a visualization tool for GUITAR

• Goal: create a visualization tool for GUITAR – a GUI testing environment developed by Prof. A. Memon

• Inputs, 3 XML files from GUITAR:
  – GUI file that describes the different elements in the GUI
  – EFG file that describes the events that connects the GUI elements together
  – TEST file that contains the results of a particular test run
Viviz Mockup
Viviz specifications (1)

• Viviz will provide the following visualizations:
  – GUI structure: *Collapsible Tree structure* or *Graph layout with GUI as a substrate*
  – EFG graph: *Network graph diagram*

• User interaction
  – Load multiple TEST files to view test covered not covered
    • Color coded edges and nodes
  – Overview, zoom-in and details on demand of visualizations
  – Filtering/searching based on:
    • GUI widgets
    • Particular events
    • Time of tests
Viviz specifications (2)

• Single window GUI
  – Simple, intuitive layout for user
  – That changes between various visualization states:
    • GUI Structure, EFG or both
  – Common set of control panels

• Programming Language:
  – Java 1.6 SE
  – Prefuse API for network visualization
  – Java Swing for GUI programming
Viviz software design
Viviz – timeline and status

- Literature survey – done
- Assignment of duties – done
- Met up with customer – done
- Finalize requirements with Prof. Memon – by next Wednesday
- First prototype – in progress: **70% done**
- Final reports and presentations – in progress: **40% done**
- Website – in progress: **60% done**
- Overall – **ON TIME**

Copyright 2009, Project Viviz, University of Maryland
Viviz project webpage

Viviz project diary

- https://docs.google.com/Doc?docid=0AR-FLr0Egz9zZG40bXh2cl8xMDdmcGtxOGI4Mw&hl=en

734 - Project Diary

- Contact (Click on the link to see the group contacts)

Nov 28:

Nov 21:

Nov 11: Progress Reports on Team Projects !!!

Nov 10 (meeting with Atif Memon):

Atif have clarified the following requirements with us:
1. He does not want 2 panels with coordinated views - for reasons of simplicity and from his viewpoint as the software testing expert. We accept that it is indeed easier for us as well.

2. He prefers that the GUI structure be visualized with the GUI as the substrate. This is possible, since the .gui xml file have (x,y) coordinates relative to the top left corner of the main GUI. However, the nodes that represents each widgets will often have coordinates that are almost similar... we have raised this issue with Atif and the conclusion is to attempt this method and show to him that it will be cluttered.

Action by Bang and Jun-cheng: show using prefuse the capability to read in the (x,y) coordinates of the .gui xml file, and display with a background of a spreadsheet the GUI showing that nodes in the graph layout can have a manually selected layout (it should be possible since a demo show nodes placed on their zip code over a map of the USA). Show that such a layout will/will not be cluttered to Atif in the next meeting.
First look at Viviz!
Problems (1)

• Prefuse API
  – Not easy to learn.
  – Took us more time than initially estimated
  – Sliding/removal of certain initial milestones

• Customer expectations
  – Had not met up with Prof. Memon as we were resolving issues in Prefuse.
  – Issues raised on the type of visualization used for GUI Structure: Tree or Graph layout?
  – Resumed meetings with Prof. Memon – to meet 2 times a week.
Problems (2)

• Unexpected events – Shiv’s computer gave up on him
  – Contained a large part of our latest source code
  – Luckily we had backups in sourceforge SVN – damage is minimal

• Lessons learnt:
  – Need to communicate and update sponsors more often.
  – Clarify doubts immediately, do not assume.
  – Always plan for delays and unexpected events.
  – Backup your source code!
Thank you!

Detailed report is linked in the project wiki page

Questions and comments,
email: cteo at cs dot umd dot edu