GUI Test Case Prioritization using Social Network Analysis

C Maitrikul and Y Limpiyakorn
Department of Computer Engineering, Chulalongkorn University, Bangkok 10330, Thailand
Yachai.L@chula.ac.th

Abstract. Graphical User Interfaces (GUIs) has become popular in today software products due to its user-friendly merit. However, GUI tests are costly and suffer explosive test cases. Moreover, software evolution causes the GUIs inevitably changed resulting in harder testing. Regression test has been widely used for GUIs software testing. It is considered to be the most expensive phase in the software testing process, though. Among several techniques, test case prioritization would help identify the reduced test cases in the regression testing suite, still maintain the same criteria as the original number of test cases in order to solve the limitation of time and cost in this phase. In this paper, we propose using network centrality from social network analysis to prioritize the test cases and compare each parameter to find a suitable parameter for GUI test case prioritization.

1. Introduction

Many of the software products have been developed daily and rolled out to support customer needs. The development process includes adding new features and modifying existing features. Before the new version can be launched to the market, developers need to ensure the correctness of code modification and avoid the side effects caused by code modification to other modules. A regression test is normally required for verifying the ongoing release. However, according to the size of the software and the number of features, regression tests suffer time consuming and account for a huge amount of the software maintenance cost. The measures to save time and budgets are thus necessary.

Graphical User Interface (GUI) was first developed by Xerox Corporation’s Palo Alto Center in 1970s [1] and today it has become widely applied in many software products [2] because it is easy to understand and provides friendly environment for user interactions more than command-line interface (CLI). Testing on GUIs requires verifying the screens with the controls like icons, menus, buttons, text boxes, types of toolbar, and menu bars, etc. The purpose of GUI Testing is to ensure that user interface functionality conforms to the specifications and works correctly. Testers must ensure that the GUIs of the system is correct by using testing techniques to generate test cases. Before starting to test, we should determine the criteria for knowing the sufficiency of testing software. When the criteria are committed, the next step is to create a set of test cases. During the testing process, finding and fixing the defects is an expensive and time-consuming activity, especially in GUIs test of which the test cases are explosive. Many researchers have applied various prioritization techniques to solve the problem such as Multi-objective test case prioritization [3], and test case prioritization using betweenness centrality measurement to solve the limitation of time and resource [4]. This paper aims to apply the network centrality method from social network analysis (SNA) to prioritize each action in the GUIs test case, and then apply the results to prioritize test case scenarios.
2. Background

2.1. Event Handler Tree (EHT)
An Event Handler of a GUI application is the function written by the user but invoked automatically when the corresponding event is performed. There is a mapping relationship between events and Event Handlers, and the relation is not one-to-one. Wang et al. [5] proposed an EHT model to assist the test case generation process. The approach can automatically combine the cell events into executable test cases. As a result, the test cases are easy to modify when GUIs are changed. Ren et al. [4] applied centrality measures to prioritize test cases. The authors proposed a two-layer model of GUI software, where the outer layer is an event handler tree (EHT), and the inner layer is a function call graph (FCG). The experiment found that betweenness centrality is more suitable for a network with large scale. The implementation of the proposed white box testing model can find the problems that cannot be found by the local methods.

2.2. Centrality Measures
Centrality Measures [6],[7],[8] are used in graph theory and complex networks. This research focuses on four metrics: betweenness centrality, closeness centrality, eigenvector centrality, and page rank.

2.2.1. Betweenness Centrality is defined as the proportion of all the shortest paths passing through the node in the network as in equation (1).

\[ b_i(g) = \sum_{j,k \neq i} \frac{v_g(i,j,k)}{v_g(j,k)} \]  

(1)

2.2.2. Closeness Centrality is computed based on the network distance between a node and each other node in the network as in equation (2).

\[ c^clse_i(g) = \frac{n-1}{\sum_{j \neq i} \rho(i,j)} \]  

(2)

2.2.3. Eigenvector Centrality considers nodes with more connections to be more important. However, in the real world, having more friends does not by itself guarantee that someone is important, having more important friends provides a stronger signal. The measure is computed as in equation (3).

\[ C_e(v_i) = \frac{1}{\lambda} \sum_{j=1}^{n} A_{i,j} c_e(v_j) \]  

(3)

2.2.4. Page Rank is like eigenvector centrality but not everyone known by a well-known person is well known. To mitigate this problem, one can divide the value of passed centrality by the number of outgoing links from the node such that each connected neighbor gets a fraction of the source node’s centrality. The measure is calculated as in equation (4).

\[ C_p(v_i) = \alpha \sum_{j=1}^{n} A_{i,j} \frac{c(v_i)}{d^out_{j}} + \beta \]  

(4)

3. Test Data
The experiment is carried out using the selected user interfaces collected from Wongnai Restaurant Management System (RMS) which is a popular application of restaurant recommender system in Thailand. The test dataset contains over 100 test cases which are converted to graph by the Event Handler Tree (EHT). As shown in figure 1, the undirected graph contains a total of 151 nodes and 331 edges.
4. Methodology
Figure 2 illustrates the steps of test case prioritization proposed in this work. The process starts with listing all the test cases selected from the target application, Wongnai RMS. There are around 100 primary test cases that have been tested in every test cycle. These test cases are used for demonstrating the proposed approach. Next, the EHT technique is applied to convert the test cases to graph. Each node in the graph represents an action in a test case while edges represent relationship between each action. The relation between each node in this scenario is undirected since it can go back and forth between each action. Once the graph connecting between each action has been constructed, we calculate each action consequence by using social network analysis measurement. In this paper, closeness centrality, betweenness centrality, eigen vector centrality, and page rank are used to measure the importance of each node. After all the values are computed, the results are used to prioritize the test cases by considering only the test cases containing modified actions and new actions just added in those test cycles. Further, test the test case using new sequence in each criterion and measure the error in the system, then compare the earliness of error finding between each measurement in the social network analysis to find the best measurement that suits our test cases. We recorded 10 test cycles along the experiment period. In each cycle, the modified and new features and changes are added to the network. The experiment and outcome are briefly discussed in the following section.

5. Evaluation
5.1 Cycle evaluation
Figure 3. EHT containing modified and new features in each test cycle.

The performance of SNA is evaluated in each cycle judged by the earliness of the error finding. In each cycle, the modified and new features are represented as a node in the event handler tree graph as shown in figure 3. After we applied the SNA measurement technique to the graph, the ranking of the importance of the action using each measure are summarized in table 1. The test cases will be then prioritized by each measure and used for test in sequences.

5.2 Overall evaluation
After having captured 10 test cycles, the best measurement that finds the first error in the environment in each cycle is reported in table 2. The result shows that the best network centrality measurement in the experiment system is betweenness centrality and then the eigen vector centrality.

Table 1. Summarize of importance of actions measured by four selected centrality measures in each test cycle containing modified and new features (ordered by Action ID).

| Action ID | Action Type | Closeness centrality | Closeness Rank | Betweenness centrality | Betweenness Rank | Page Rank | Page Rank | Eigen vector centrality | Eigen vector Rank |
|-----------|-------------|----------------------|----------------|------------------------|------------------|-----------|-----------|-----------------------|------------------|
| 0         | new feature modified | 0.194805 | 9 | 0 | 10 | 0.003322 | 9 | 0.00419 | 10 |
| 1         | modified feature | 0.212766 | 6 | 85.145069 | 5 | 0.007014 | 6 | 0.79141 | 2 |
| 2         | new feature | 0.207182 | 7 | 0.287926 | 9 | 0.002516 | 10 | 0.200102 | 3 |
| 3         | new feature | 0.181159 | 10 | 7 | 8 | 0.00741 | 5 | 0.799898 | 1 |
| 4         | new feature | 0.217391 | 5 | 35.5 | 6 | 0.005028 | 8 | 0.006594 | 8 |
| 5         | new feature modified | 0.241158 | 3 | 217 | 4 | 0.008075 | 3 | 0.010739 | 6 |
| 6         | modified feature | 0.248756 | 2 | 970.5 | 2 | 0.007609 | 4 | 0.014107 | 5 |
| 7         | modified feature | 0.251678 | 1 | 1785.966667 | 1 | 0.024188 | 1 | 0.032661 | 4 |
| 8         | modified feature | 0.218978 | 4 | 30 | 7 | 0.005376 | 7 | 0.006282 | 9 |
| 9         | modified feature | 0.202156 | 8 | 437.5 | 3 | 0.011049 | 2 | 0.008111 | 7 |
### Table 2. Summarize of the first network centrality able to find error earliest in RMS test environment.

| Network measurement | Betweenness | Betweenness | Eigen vector | Betweenness | Betweenness | Eigen vector | Betweenness | Eigen vector | Betweenness | Eigen vector |
|---------------------|-------------|-------------|--------------|-------------|-------------|--------------|-------------|--------------|-------------|--------------|
| 1                   | 2           | 3           | 4            | 5           | 6           | 7            | 8           | 9            | 10          |

#### 6. Conclusion

Centrality has been one of the important methods in complex network structure analysis. In this research, centrality measurement has been applied to prioritize both modified and new test cases in a large recommender system. There are over 100 test cases for both existing and new functions in Wongnai RMS that are usually tested on a weekly basis without any prioritization. The experiments were carried out to evaluate the prioritization efficiency by comparing the importance of each function in the network assessed by social network analysis measurement and using it for ranking the test cases. The measures that have been selected are betweenness centrality, closeness centrality, eigenvector centrality, and page rank. Once the results of each measure have been manipulated, we tested the test case to find the best method able to find the error earliest in the test cycle, then we captured the data of 10 test cycles to find the best measure that suits our test case environment. The findings reported that the betweenness centrality yielded the best performance, followed by the eigenvector centrality. However, there is no significant level of deviation among the four measures due to the size of data. In future work, it is suggested that the effort spending in each action should be considered and the size of test cases should be increased.

#### 7. References

[1] Mini tools, Retrieved April 11, 2020, from minitool.com website: [https://www.minitool.com/lib/graphical-user-interface.html](https://www.minitool.com/lib/graphical-user-interface.html)

[2] Memon A M 2001 Comprehensive Framework for Testing Graphical User Interfaces Pittsburgh: University of Pittsburgh

[3] Sun W, Gao Z, Yang W, Fang C and Chen Z Multi-Objective Test Case Prioritization for GUI Application

[4] Ren Y, Yin B, and Wang B 2018 Test Case Prioritization for GUI Regression Testing Based on Centrality Measures. In: IEEE 42nd Annual Computer Software and Applications Conference (COMPSAC), Tokyo, pp 454-459

[5] Wang B, Yin B and Cai K 2016 Event Handler Tree Model for GUI Test Case Generation. In: IEEE 40th Annual Computer Software and Applications Conference

[6] Borgatti S 2005 Centrality and network flow, Social Networks, 27, pp 55-77

[7] Bloch F, Jackson M and Tebaldi P 2019 Centrality Measure in Network

[8] Riquelme F, Gonzalez-Cantergiani P, Molinero X and Serna M 2017 Centrality measure in social networks based on linear threshold model