Using Search Based SE for GUI Test Data Generation

ABDUL RAUF
Department of Computer Science, National University of Computer & Emerging Sciences, Islamabad, 44000, Pakistan
a.rauf@nu.edu.pk

MUHAMMAD JAMIL AHMED
Department of Computer Science,
Iqra University,
Islamabad, Pakistan
jamilqureshee@gmail.com

SHAFIQ UR REHMAN
Department of Computer Science, National University of Computer & Emerging Sciences,
Islamabad, 44000, Pakistan
shafiq.rehman@nu.edu.pk

ARSHAD ALI SHAHID
Department of Computer Science, National University of Computer & Emerging Sciences,
Islamabad, 44000, Pakistan
arshad.ali@nu.edu.pk

Abstract:

The graphical user interfaces (GUIs) are playing a major role in the popularity of software systems. Recognizing the importance of GUI, software teams feel an immense pressure in delivering the interface according to expectations of the customer. Beside heavy focus and due attention towards GUI, software engineers are still looking for practices to ensure the thorough testing of such applications involving GUI's. One major breakthrough to automate this manual effort of GUI testing is to map GUI events with some models and graphs. Event-flow graph is relatively a fresh and useful addition to cope up with automation of GUI testing. We have used event driven nature of GUI for testing in some of previous studies. In this paper we are presenting an idea of generating test cases for GUI based on search based algorithm and manipulating ontology for GUI testing. This ontology is supposed to work on the basis of semantics of possible events and than annotations will use to generate the test cases and work as an oracle for verification of the output of testing effort.

Keywords: GUI Testing; Genetic Algorithm, Ontology, Semantic Annotations;

I. Introduction

Because of ease and flexibility provided by graphical user interfaces (GUIs), they are becoming most vital modules of software systems. On the other hand, a lot of research work is being carried out in software testing field but subfield of GUI testing is still not getting its due attention. In one of our previous study we have proposed a technique based on genetic algorithm (A search based algorithm) to exploit the event flow nature of GUI that analysis coverage based on recorded events [1]. This technique has shown very good results in analyzing coverage of GUI's. Similarly we have tried multi objective search based algorithms (NSGA-II) and MOPSO in our experimentation as well [2, 3]. These studies have also produced encouraging result from GUI test coverage prospective. Using the results of our previous experimentations, in this paper, we are presenting an approach to automate the test case generation process for GUI testing based on genetic algorithm, semantic annotation and ontology. Our approach uses the concepts from GetFollows algorithm [4], and ontology. Our proposed method can also be used to remove the manual effort required in grouping events based on functionality described in [4].

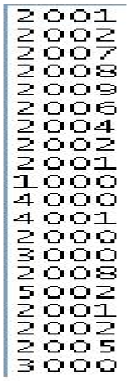
Ontology defines the basic terms and relations constituting the vocabulary of a specific domain area as well as the rules concerning that specific domain. Ontologies have been applied to describe a variety of knowledge domains [5]. An annotation is the meta tag, used to give some life to the code you are using. Annotations do not directly affect program semantics, but can affect the semantics of the running program. Semantics is the study of meaning, usually in language. In linguistics, it is the study of interpretation of signs or symbols as used by agents or communities within particular circumstances and contexts [6]. Semantic Annotation is a fundamental knowledge being used for the development and usage of intelligent contents. Genetic algorithms are inspired by Darwin's theory about evolution. Solution to a problem solved by genetic algorithms is evolved. Algorithm is started with a set of solutions (represented by chromosomes) called population. Solutions from one population are taken and used to form a new population. This is repeated until some terminating condition (for example number of populations or no improvement of the best solution after certain iterations) is satisfied.

II. Proposed Method

An imperative attribute of GUI systems is that their behavior is very much dependent on the context in which they are being used. [4]. Besides of functionality of a GUI element, response of a GUI element to an event may be different depending on the perspective established by preceding events and their execution order [4]. One basic principle in software testing is that as many components of software (widgets in case of GUI) are tested so much confidence is built regarding quality of software. Keeping in mind these concerns, we have generated the test data by interacting with the GUI of application and recording the generated events as have been shown in figure 01. We have given each event a unique number to create unique events. These events sequences are than passed to genetic algorithm (GA) for prioritizing test cases based on coverage achieved by each event sequence. Same sequence of events have also been passed to ontology as building block of domain knowledge. Using the getFollows algorithm (shown in figure 02), ontology determines the predecessors and successors of each event. Semantic aanotition, plays a role than to create the automated test cases of similar events based on ontology output and prioritization achieved through genetic algorithm. Relationship between semantic annotations with ontologies and genetic algorithm is the core of our proposed method.

The major contributions of our proposed technique are following:

- Ontology has been proposed to help automation of test case generation process.
- Genetic algorithms and Semantic annotations are being used for test case generation.



```
2001
2002
2007
2008
2009
2006
2004
2002
2001
1000
4000
4001
2000
3000
2008
5002
2001
2002
2005
3000
```

Figure 01: Sequence of Generated Events

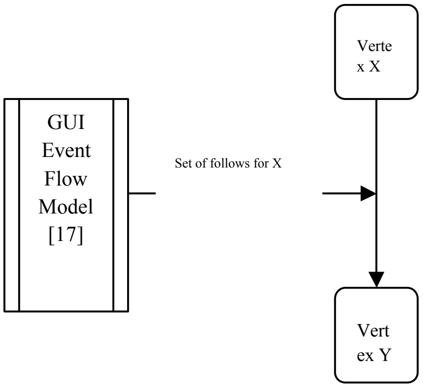


Fig 02: GetFollows from GUI Event Flow Model [7]

III. Future Work

Despite the above mentioned contributions our research effort has some limitations. As our work gets evolve, we are hopeful in overcoming these problems.
Some of the future works are:

- We have to design the process for extracting semantics from ontology.
- Complete specification of ontology needs to be provided.
- Extensive experimentation is required to verify the results.

IV. Conclusion

This paper presents a novel idea based on ontology based test case generation. Ontology has been proposed to help automation of test case generation process. And semantic annotations are being used for test case generation. The results of our study show that by increasing event combination strength and controlling starting and ending positions of events, our test cases are able to detect a large number of faults, not detected by exhaustive test suites of short tests. [4]

ACKNOWLEDGMENTS: The first author is sponsored by Higher Education Commission of Pakistan, and acknowledges HEC.

References

1. Abdul Rauf, Sajid Anwar, Arfan Jaffar, Arshad Ali Shahid, “Fully Automated GUI Testing & Coverage Analysis using Genetic Algorithms ”, International Journal of Innovative Computing, Information and Control (IJICIC) Vol.7, No.3, March 2011[To be Appeared]
2. Abdul Rauf, Sajid Anwar, Naveed Kazim Khan, Arshad Ali Shahid, “Evolutionary based Automated Coverage Analysis for GUI Testing” Communications in Computer and Information Science (Springer) ISSN: 1865-0929
3. Abdul Rauf, Naveed Kazim, Sajid Anwar, Shafiq-Ur Rehman, Arshad Ali Shahid “Maximizing GUI Test Coverage Through PSO”, 19th International Conference on Software Engineering and Data Engineering (SEDE-2010), San Francisco, California, USA, June 16 – 18, 2010.
4. X. Yuan, M.B. Cohen and A.M. Memon, GUI Interaction Testing: Incorporating Event Context, IEEE Transactions on Software Engineering, IEEE Computer Society,Los Alamitos, CA, USA,2010
5. Ellen Francine Barbosa, Elisa Yumi Nakagawa, José Carlos Maldonado: Towards the Establishment of an Ontology of Software Testing. SEKE 2006: 522-525
6. Otto Neurath (Editor), Rudolf Carnap (Editor), Charles F. W. Morris (Editor) (1955). International Encyclopedia of Unified Science. Chicago, IL: University of Chicago Press.
7. Memon, A. M. 2007. An event-flow model of GUI-based applications for testing: Research Articles. Softw. Test. Verif. Reliab. 17, 3 (Sep. 2007), 137-157. DOI= http://dx.doi.org/10.1002/stvr.v17:3