

# ***Summary of “Using Dynamic Symbolic Execution to Generate Inputs in Search-Based GUI Testing”***

***Nikhil Satish Pai      Nikhil Anand***

***8th November 2015***

## ***Important Keywords:***

***ii1. EXSYST:*** *Open source automated test generation tool for GUI applications which focuses on exploring user interfaces while aiming to maximize code coverage*

***ii2. Dynamic Symbolic Execution:*** *combines symbolic execution with concrete executions of the program using concrete values. During execution a symbolic state of the system is maintained. Whenever a change in the program state occurs, the symbolic state is updated accordingly.*

***ii3. Code Coverage:*** *measure of the degree to which the source code of a program is tested.*

***ii4. Graphical User Interface:*** *Interfaces that allows the user to interact with a systems via **graphical** icons and visual indicators rather than text-based **interfaces***

## ***Brief notes:***

### ***iii1. Motivational Statement***

*Search-based testing techniques can generate complex sequences of events for graphical user interfaces, but these typically rely on simple heuristics or randomization for generation of data values. This reduces the effectiveness as the specific inputs generated may not match with the values which the users may enter. Generating such input values is possible with dynamic symbolic execution , but dynamic symbolic execution is not suited to generate complex sequence of events. The paper introduces a hybrid approach where search based testing is used to generate sequence of events and DSE is used to generate the input data.*

### ***iii2. Future Work:***

*The current extension of EXSYST is limited to a subset of widgets support will be need for more widgets Extending the existing approach to deciding whenever dynamic symbolic execution will be beneficial similar to the current implementation of EXSYST to configure how often DSE can be applied and deciding which rate is optimal.*

### **iii3. Baseline Results :**

*The authors compare the performance of the extension of the EXSYST tool created with the existing EXSYST tool.*

*The first is Tickets, a GUI application to simulate a plane ticket price calculator. Tickets uses several parameters such as passenger class, the type of customer, and the travel distance. Second is Workout Generator an application generates a workout plan for a person based on age, sex, height, and weight. The results show that the extension of EXSYST achieves higher coverage than EXSYST.*

### **iii4.Related work:**

*BARAD [15] is a GUI testing framework for applications written in Java with the Standard Widget Toolkit (SWT).*

*GAZOO [16] is a fully automated GUI testing tool for .NET applications. Both rely on DSE to find input data for event sequences. COLLIDER [18] targets Android applications. It uses concolic execution to build symbolic summaries of the GUI's event handlers.*

### **Area of Improvement:**

*Current tests only consider two standard test models, this must be expanded to include more case studies.*

### **Relation to Original paper :**

*The current paper cites the original as introducing an approach to combine Search based and dynamic symbolic execution whoever it will not be able to perform testing of GUI based applications.*

### **Reference to the Paper:**

Kevin Salvesen, Juan P. Galeotti, Florian Gross, Gordon Fraser, and Andreas Zeller. 2015. Using dynamic symbolic execution to generate inputs in search-based GUI testing. In *Proceedings of the Eighth International Workshop on Search-Based Software Testing (SBST '15)*. IEEE Press, Piscataway, NJ, USA, 32-35.