****

Student ID: C00132795

Student Name: Shengjie Yuan

Supervisor: Christophe Meudec

Date of submission: 19.12.2010

**<** **Another Bugs Contraption Detector>**

Content

[Abstract 1](#_Toc280264378)

[1. Vision 1](#_Toc280264379)

[1.1 Project Introduction 1](#_Toc280264380)

[1.2 Project benefits analysis 1](#_Toc280264381)

[1.3 Stakeholder Descriptions 2](#_Toc280264382)

[1.4 Other information 2](#_Toc280264383)

[2. Functions description 3](#_Toc280264384)

[2.1 Functions presentation 3](#_Toc280264385)

[2.2 Use Case diagrams 5](#_Toc280264386)

[2.3 Brief Use Case 5](#_Toc280264387)

[2.4 Detail Use Case 5](#_Toc280264388)

[3. Supplementary Specifications 7](#_Toc280264389)

[3.1 Functionalities across use case 7](#_Toc280264390)

3.2 Usability……………………………………………………………………………………………………………..7

3.3 Reliability……………………………………………………………………………………………………………7

3.4 Performance……………………………………………………………………………………………………….7

3.5 Supportability……………………………………………………………………………………………………..8

3.6 Others…………………………………………………………………………………………………………………8

[4 Conclusion 8](#_Toc280264391)

# Abstract

The main purpose of this documentation is introduces the core functions of the project, in the following sections, it will make a brief description on project purpose and its design goals, in the section of supplementary specifications, the information about Functionalities, Usability, Reliability and Other also will be contained. This documentation also will analyses function use case in detail.

# Vision

# Project Introduction

The use area of this Bugs Contraption Detector is about software testing, and the Code Parsing, Code Instrumentation technologies would be used. This project allows automatic test inputs generation, it tries to generate input automatically. The ability to instrument, for test inputs generation purposes, during normal execution, information that are sent to an external constraints solver for automatic test data generation. For the additional function, it will try to make the target programming language useful and/or find problems in real code.

# Project benefits analysis

In the process of software development, test code is very important part of the whole project. Analysis code`s output by different type of input can cause developer more efficient debug. Based on this premise, this project tries to achieve the following goal: Given a method with parameters, automatically generate a set of input values that, upon execution, will exercise as many statements as possible in the method. In other words, this project seeks to generate a test suite with high code coverage.

# Stakeholder Descriptions

* + 1. **User Summary**

This program is use to analyse and test code written by C language, so the main user ground should be software developer who wants to test their code and analyses the different output result by different type input value.

* + 1. **Key High-Level Goals and Problems of the Stakeholder:**

|  |  |  |  |
| --- | --- | --- | --- |
| **High-Level Goal** | **Priority** | **Problems and Concerns** | **Current Solutions** |
| Test target code | high | Cannot find out and fix the Grammar and logic errors yet. | Check the code make sure that no error before test |
| Make more efficient in the part of test code in the process of development. | mid | It is difficult to mix and line to existing complier | Set up the path of target code, and execute program run that file. |
| High code branches coverage | high | In very rare cases will miss some branches. | Executed the program test the target code again, Report the error to developer |

# Other information

* **Target platform:**

This program is preliminary design supports for Multi-Platform, The main support platform should be WINDOW.

* **Target language:**

Programming language C should be the test target language.

* **Develop programming language:**

Programming language C/C++ is the develop language, it has good support by Microsoft Visual Studio 2008, and it has simple syntax structure and more functions developable.

# Functions description

# Functions presentation

This section will provide an example code for test our project; we can know what result of our project can be outputted. Base on the main purposes of our project development. when user use this program to execute the target code. Program will create numbers of different type random input; the target code will execute those random inputs and then output the different result.

The following code is use to test our program:

|  |
| --- |
| void DoWork( int[] v ){ if (v != null && sizeof(v) > 0 && v[3] == 12345) printf("hidden bug!"); else printf(“error”);} |

* Program starts by getting some inputs. It always begins with the simplest inputs possible. In this example, the input would be null (v = null).
* Then program executes the code along the execution path that is taken for the chosen inputs. For the example above, the method returns because v == null.
* Program negates this condition and queries a constraint solver to determine whether there is a solution for the negated condition.
* If a solution exists, then this solution represents another test input, which would cause the code to take a different path.

Program will repeats this process. Internally, it represents all conditions that the program checks as a tree as follow:  **

Every time program executes the code, the execution tree might grow, after a few iterations, program finishes for the following example.

|  |  |  |
| --- | --- | --- |
| Constraints to solve | Input Date | Observed constraints |
|  | Null | a == null |
| a != null | {0} | a != null && !(a.Length>0) |
| a != null && a.Length>0 | {0,0,0,0} | a != null && a.Length>0 && a[3] != 12345 |
| a != null && a.Length>0 && a[3] == 12345 | {0,0,0,123} | a != null && a.Length>0 && a[0] == 12345 |

# Use Case diagrams



**Use cases:** Test target code

**Actor**: user

# Brief Use Case

**Use Case**: Test target code

**Actor**: user

**Description**: This use case begins when the user wants to test the target code, the path of the target file and the file name of the target code must be provided. The test input will be created randomly and automaticity by this program. The result of the execution will be display and automaticity create result file. The command line is available to use, the operations of command line should include: set the path of the target code, executed test and set the path of test result.

# Detail Use Case

**Use case:** test target Code

**Actor:** user

**Description:** This use case begins when the user wants to test the target code, the path of the target file and the file name of the target code must be provided. The test input will be created randomly and automaticity by this program. The result of the execution will be display and automaticity create result file. The command line is available to use, the operations of command line should include: set the path of the target code, executed test and set the path of test result.

**Main Scenario about Input target code:**

1. This use case begins when the user wants to test the target code.
2. User use command line to execute this program.

3a. program executed successful, it will display successful executed information in command line.

3b. program executed failed, it will display error executed information in command line.

1. program display information that need user to set the path of the output result, this path is use to save the output file.

5. The path is set successed, it will display successful setting information in command line.

1. Program display information that need user to set the path of the target file, the target code which is want to test should in this file.

7a. The target file is exist, it will display successful loading information in command line.

7b. The target file is not exist, it will display failed loading information in command line.

8. System display the Confirm instructions ( test the target code. yes / no? ).

9a. User input yes for execute the test.

9b. User input no for exit.

10. Program executes the target file in order to test the target code.

11. When the execution finished, program display the result in command line.

12. Program automaticity creates output file and save in particular output path.

# Supplementary Specifications

# Functionalities across use case

In the use case Test target code. Program is operated by command line, once user set up the output result path, it will automatic save this path as default path, user do not need to set the output result path anymore. If user wants to change this default path, program also provides related instructions to user complete this operation.

* 1. **Usability**

This program can automaticity create different type input value to execute the target code in order to output different result, it is useful for developer test and analyses code.

This program has not any complex operations and functions, I tend to make no User Interface for user, instead that most of the operations will complete in command line, so users need to know how to operate command line for base.

* 1. **Reliability**

This application can help user test their code and check the output result, high code coverage and reliability, however, if in the very rare cases that would output incorret result for user, in order to minimize that probability, user should strictly enforce the operation correctly, user also could report the incorret output result to developer.

* 1. **Performance**

For the reason of no User Interface in this program, our goal: high efficiency low consumption, fast and efficient operation response time. It can test several target file at the same time, in order to improve code test efficiency.

* 1. **Supportability**

This application written by programming language C/C++, it can support running on Multi-Platform and developing more additional functions by other developer. It also support various levels of programmers, it is easy operate on command line.

* 1. **Others**

This program only support test programming language C code, it can test target code output result, however cannot fix grammar and logic errors in the code, therefore before execute test, make sure the code correctness.

# Conclusion

This project Another Bugs Contraption Detector can automatically generate a set of input values that will exercise as many statements as possible in the method. In other words, this project seeks to generate a test suite with high code coverage. Those functions are useful to developer test their code and check the output result base on different type input value, help developer high efficiency in the process of software development. In this documentation, it briefness descript project functionality and use case in detail, analyse project benefits and design goals.