

AUTOMATIC GRAPHICAL DESIGN GENERATOR

Bhavesh Mehra, Alpesh Purohit, Piyush Solanki, Silviya D'monte

*Department of Computer Engineering
Universal College of Engineering, Vasai, Maharashtra, India*

-----***-----

Abstract - It is great significance to automatically generate a graphical design from the input text file as program code. There are some deficiencies in existing system also there are very few full-featured integrated development platforms that can generate graphical design automatically based on program code. In existing system we need to drag and drop the symbols. So user should be having knowledge about the program. In proposed System no need of having knowledge about the program to draw a graphical design it simply accept the program as input text file and according to that graphical design is generated. Program should be logically and syntactically correct. Previously system used Drag and drops the particular symbols according to the certain conditions. So it is basically time consuming and tedious. In proposed system it generates the graphical design automatically according to the input text file. At a time one program is taken to generate the graphical design. The program that the system accepts must be error free and it must follow the certain rules and logic. It accepts the program as the input file for only selected language such as (Java).The program should be saved in text file. The text file should be loaded in system to generate the graphical design.

The project can be used in institutes wherein the graphical designs can be used to make the students understand complex programs and lastly, to make a program more understandable

Key Words: Graphical Design, Programs, Automatic, Window Application

1.INTRODUCTION (Size 11 , cambria font)

Much of the study of computer science is dedicated to discovering efficient algorithms and representing them so that they can be understood by computers. [1]According to the Schneider and Gersting 1995, An algorithm is a well-ordered collection of unambiguous and effectively computable operations that when executed, produces a result and halts in a finite amount of time . While programming languages avoid the problems of being wordy and ambiguous, they have some other disadvantages that make them undesirable for writing algorithms. Before writing a

complex program, a software Developer usually plans it out by creating a graphical design. A graphical design shows different steps a program takes, using different shapes to describe the various actions the program carries out . Graphical design serve as important pieces of documentation for individual programs and for systems of programs in a larger system. Because a graphical design accurately summarizes a program's functions, and because it uses clear English and easy-to-understand, standard symbols, it takes much less effort to read it than it does the program itself . If the program develops problems, a programmer may save time by checking the graphical design before examining the program. Though graphical design are technically-oriented, non-technical users can easily follow them, making them useful documents for project management and auditing. In this paper, we choose two reverse engineering graphical design tools on the basis of their capabilities and features that captures the functionality of the system and the design at an abstraction level that removes the need of language implementation. Our current research effort focuses on a analysis of these tools that will be useful in maintaining the consistency of design document with its code and for understanding large systems.

1.1 Objectives of the system

A graphical design is a type of diagram that represents a process, showing the steps as boxes of various kinds, and their order by connecting them with arrows. This diagrammatic representation illustrates a solution to a given problem. Process operations are represented in these boxes, and arrows; rather, they are implied by the sequencing of operations. Graphical design are used in analyzing, designing, documenting or managing a process or program in various fields. Graphical Design are used in designing and documenting complex processes or programs. Our project automatic graphical design generator is basically accept the program as the input file and generates the graphical design for

program. The system allows user to select whether the input text file be a program. The user selects the text file that contains the input file as program and it generates the graphical design as the output. It basically accepts the program as the input file and generates the Graphical design for the program. The program must be error free and it follows the certain rules and logic. The user need not have the understanding of the program; he simply has to provide the input program and the graphical design is generated. The logic and flow of the program is automatically understood by the system, likewise the graphical design is drawn.

1.2 Proposed System

The proposed system takes a program a text file as input and produces a graphical design of the same. The system allows the user to either import an already existing java program or write the program in the given workspace. Based on the program, the system automatically generates the graphical design. In multinational companies and universities, there is a requirement of graphical design so as to easily understand the project implementations. Also, in case of reverse engineering a software program wherein the user has the code but is need of graphical design for the same. Hence, there is need of something that quickly overcomes the user from this trouble.

This project is implemented so as to reduce the load of manually creating the graphical design. Using our proposed system, this drawback is overcome by automatically generating the graphical design. Conditional statements are easily analyzed because of the use of graphical design. Graphical design generated from a code is useful in case of reverse engineering a software program wherein the user has the code but is need of graphical design for the same. An automatically generated graphical design is also helpful for software testers that are in need of to understand the code; as they have very little time to understand the code by reading it and analyzing each and every statement.

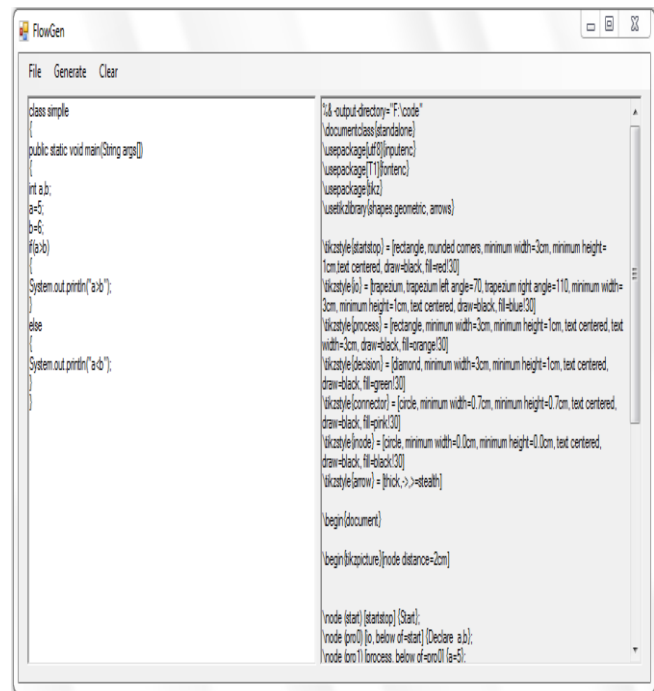


Fig -1: Windows application

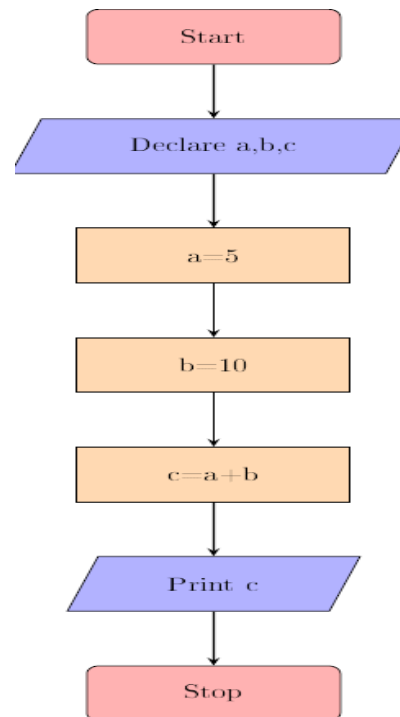


Fig -1: Graphical design

2. DISCUSSION

1. Examine the source code given as an input to the tool. Sometimes source code is available for a reverse engineer and sometimes it is not. Reverse engineering approaches involves examine source code. Reverse engineering approaches is typically very effective in finding programming errors and implementation errors in software. There are two types of analysis tools, those that require source code and those that automatically decompile the binary code and continue from there. These tools include sourcecode-related problems and issues commonly encountered in Java, does require source code.

2. Verification of the source code & Identify the Programming Language.

Reverse engineering approaches involves analyzing and understanding given source code. Program failures can be analysed to determine whether errors exist in the target software. Some other reverse engineering approaches refers to analysing a running program by probing it with various inputs. This kind of approaches requires

only a running program and does not make use of source code analysis of any kind.

3. If the input is in textual format perform compilation i.e., software engineering.

A compiler is a tool that converts machine-readable code into assembly language. Assembly language is a human-readable form of machine. Compiler reveals which machine instructions are being used in the code. Machine code is usually specific to a given hardware architecture. Thus, Compilers are written expressly for the target hardware architecture.

3. CONCLUSIONS AND FUTURE WORK

More Program Structures will be added to create graphical design for very complex programs. As in this project graphical design is viewed through latex, hence a system will be advanced so that it will have graphical design viewer in the system or export it to images for using images for other purpose. As this is a standalone application, which needs to explicitly install in system. So with the advent of HTML 5 we can even embed flow charts creating a web service

The graphical design generator helps generating simple graphical design. It implements graphical design generation for basic programs covering simple instructions, control structures and loop structures. The powerful features of Latex makes it

system independent and can still have intermediate language generation.

REFERENCES

- [1] Schneider, M. and J. Gersting (1995), *An Invitation to Computer Science*, West Publishing Company, New York, NY, p. 9
- [2] International Journal of Computer Science & Information Technology (IJCSIT) Vol 6, No 1, February 2014 **Shape-Based Plagiarism Detection for Flowchart Figures in Texts** Senosy Arrish, Fadhil Noer Afif, Ahmadu Maidorawa and Naomie Salim
- [3] B. G. Vasudevan, et al., "Flowchart knowledge extraction on image processing," in IEEE International Joint Conference on Neural Networks, 2008. IJCNN 2008. (IEEE World Computational Intelligence). Hong Kong, 2008, pp. 4075-4082.
- [4] FATESOFT (2009), Code Visual to Flowchart, FateSoft, Eden Prairie - NM - USA, [Accessed 29/10/2009] <http://www.fatesoft.com/s2f/>.
- [5] Andrew Scott, Mike Watkins and Duncan McPhee., (2010), ELearning For Novice Programmers, A Dynamic Visualisation and Problem Solving Tool, <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4529966>
- [6] I. Nassi and B. Shneiderman, "Flowchart Techniques for Structured Programming," *ACM SIGPLAN Notices*, Vol. 8, No. 8, 1973, pp. 12-26. doi:10.1145/953349.953350
- [7] J. F. Gimpel, "Contour: A Method of Preparing Structured Flowcharts," *ACM SIGPLAN Notices*, Vol. 15, No. 10, 1980, pp. 35-41.
- [8] Aivosto, (2003). Visustin Flow Chart Generator [online]. Aivosto.com. Available from: <http://www.aivosto.com/visustin.html> [Accessed 21-5-2005].
- [9] Carlisle M, Wilson T, Humphries, J and Hadfield, M, 2004, RAPTOR: Introducing Programming To Non-Majors With Flowcharts, *Journal of Computing Sciences in Colleges, Consortium for Computing Sciences in Colleges, University of Central Missouri, USA* pp: 52 – 60.

[10] Nivedita Tiwari, Lalji Prasad, " A Comparative Study: Reverse Engineering Flowcharting tools" IJITE, Vol. 7, No. 1, 2015.

BIOGRAPHIES



Mr. Bhavesh Mehra is currently completing **B.E** in Computer Engineering from the Universal College of Engineering, Vasai, Mumbai University, Mumbai, India.



Mr. Alpesh Purohit is currently completing **B.E** in Computer Engineering from the Universal College of Engineering, Vasai, Mumbai University, Mumbai, India.



Mr. Piyush Solanki is currently completing **B.E** in Computer Engineering from the Universal College of Engineering, Vasai, Mumbai University, Mumbai, India.