

A Java Based 4th Generation Multi-Targeted User Interface Compiler (JUICE)

Amruta Mukund Talathi

*Department of Information Technology Engineering
Satara College of Engineering and Management, Satara, Maharashtra, India*

Sarita V. Balshetwar

*Department of Information Technology Engineering
Satara College of Engineering and Management, Satara, Maharashtra, India*

Dhiraj Vijay Motghare

*TCS
Pune, Maharashtra, India*

Abstract- Today designing a good GUI in a widely acceptable language is not an easy task. We plan to design a simple user language with easy to understand constructs for designing a user interface. But of course the user will not like only the GUI to be in a language other than the language in which he is developing an application. To overcome this issue we plan to implement a compiler, to be written in Java, which will combine this new language to a target language such as Java. Thus a user will get the code for the GUI he is designing in a high level language. We also plan to provide an IDE for writing the new language and for compiling it to the target language. The system consists of a compiler for compiling and translating JUICE SCRIPT into specified target language, which may be Java Swing, Java AWT, XUL or HTML. The system must be expandable for inclusion of new languages also. An IDE for developing the JUICE SCRIPT is to be provided for easier development of the JUICE SCRIPT.

Keywords – JUICE, Multitargeted compiler, JavaCC, XUL

I. INTRODUCTION

To work with a system, the users need to be able to control the system and access the state of the system. User can interact with system with the help of graphical user interfaces (GUI) which accept input via devices such as computer keyboard and mouse and provide graphical output on the computer monitor. The graphical user interface is a computer interface that uses graphic icons and controls in addition to text. The user of the computer utilizes a pointing device, like a mouse, to manipulate these icons and controls. This is considerably different from the command line interface (CLI) in which the user types a series of text commands to the computer. Today designing a good GUI in a widely acceptable language is not an easy task.

Our system consists of a compiler for compiling and translating JUICE script into specified target language, which are Java Swing, Java AWT, XUL or HTML.[1] The system must be expandable for inclusion of new languages also. An IDE for developing the JUICE script is to be provided for easier development of the JUICE script.

II. PROPOSED ALGORITHM

- Step1- The user has to open the IDE provided.
- Step2- The user has to write input script and save the file with .mgk extension.
- Step3- The file is passed as an input to the parser.
- Step4- The parser parse's the tokens one by one and pass it to switching mechanism.
- Step5-The user has to select the target through command switch.
- Step6-The tokens and command switch is passed as an input to the switching mechanism.
- Step7-Switching mechanism generates the class file name by formula:-Command switch + token + Data type.
- Step8- According to the command switch selected the class will be passed towards the Code Generator.
- Step9- In the code generator the component will be selected as per the class name generated by the switching mechanism.

Step10- As per the sub tokens generated by the parser the methods in the component are selected.

Step11-The methods invoked will write the GUI code in the output file.

Step12- The output file is compiled to get the GUI in desired language.

III. SYSTEM WORKING, EXPERIMENT AND RESULT

- *SYSTEM WORKING*

All the code for this system is segregated into different Class files. An IDE is developed where user can enter code in JUICE script and choose an option for target language.

The compiler has been implemented using JavaCC parser generator. It detects all the tokens described in JUICE Script. It also checks for parsing errors. Each GUI designing language has some common GUI components. For each component class files are created. All the Classes pertaining to a given target language are similar to Classes of other target language.

The compiler takes input file with .juc extension and converts it to target language based on the switch entered. It references the class related to a token for a given target language by adding the switch for language before the common class name for that token.

For example, on getting token 'button' and switch AWT the class referenced is AWTButtonDataType. Then a call is made to a function written in the referenced class based on which property or event is mentioned for that control. The code for the same is written in the output file by the called function. This Design Strategy makes JUICE scalable i.e. adding new language is easier. Error detection and Maintenance is also easier.

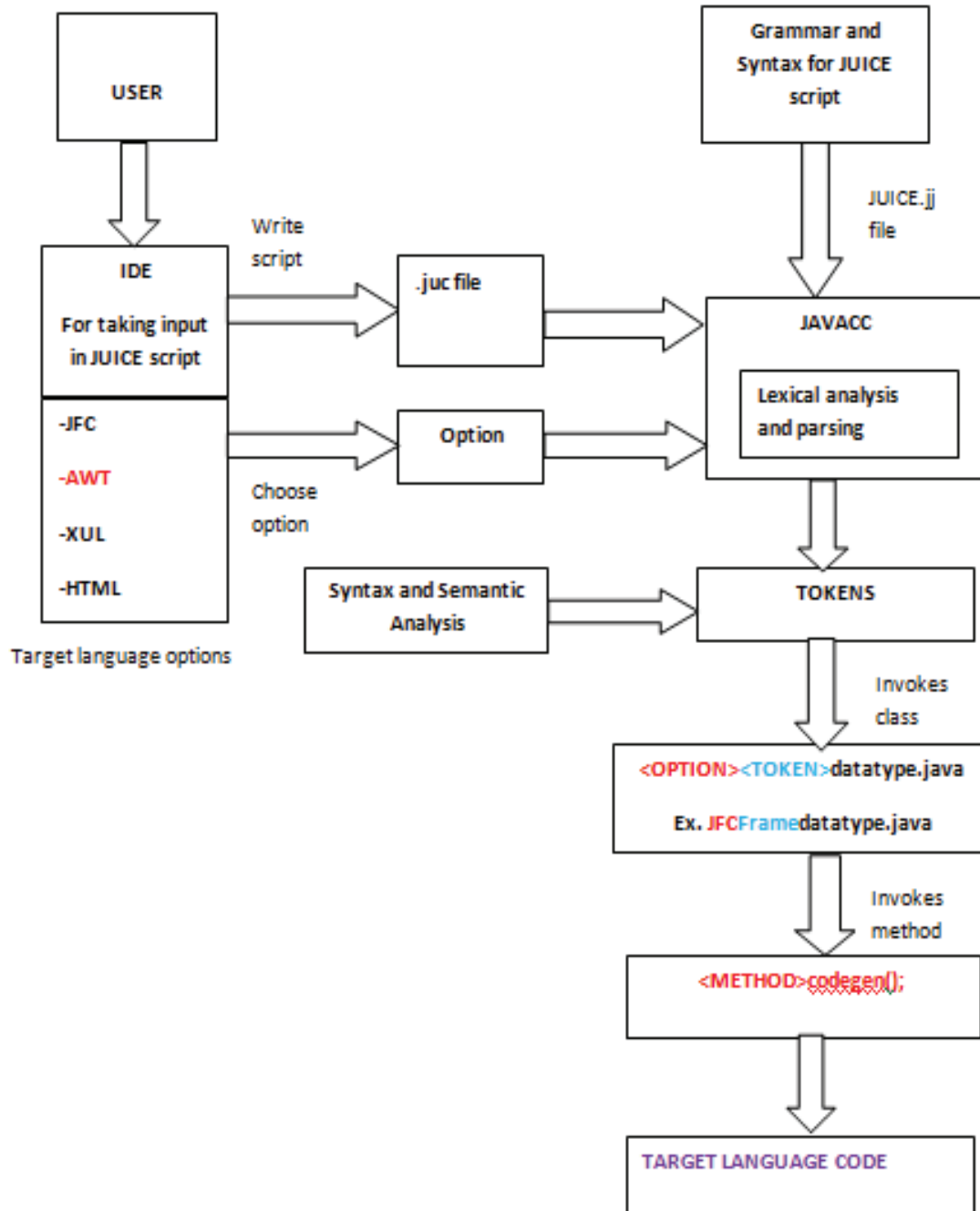


fig. system working

IV. EXPERIMENT AND RESULT

JUICE Editor look like this:

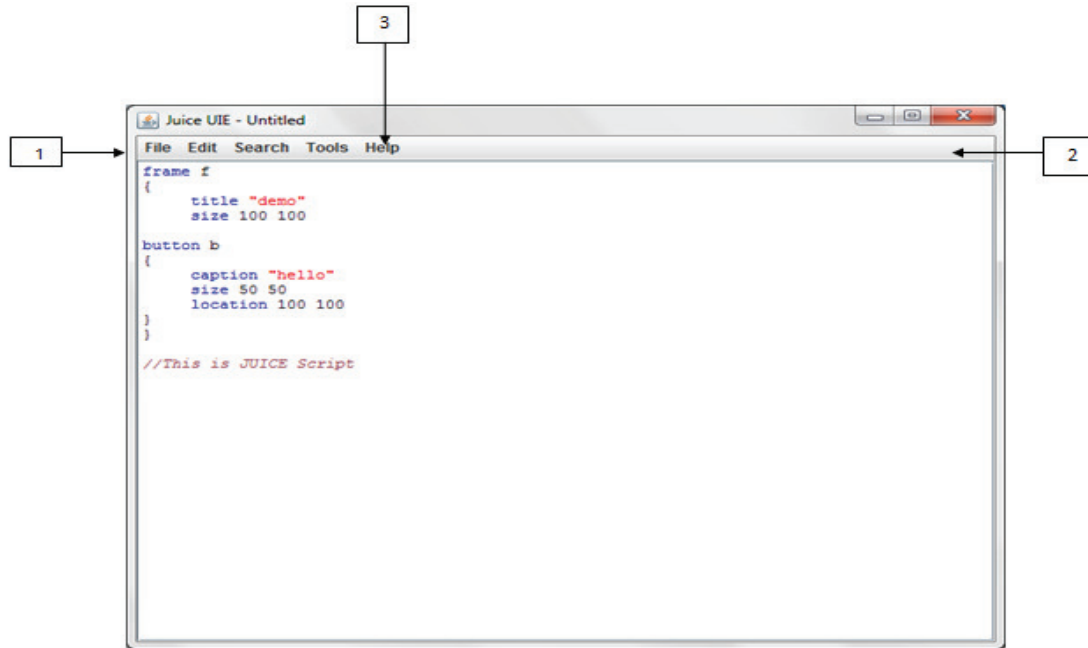
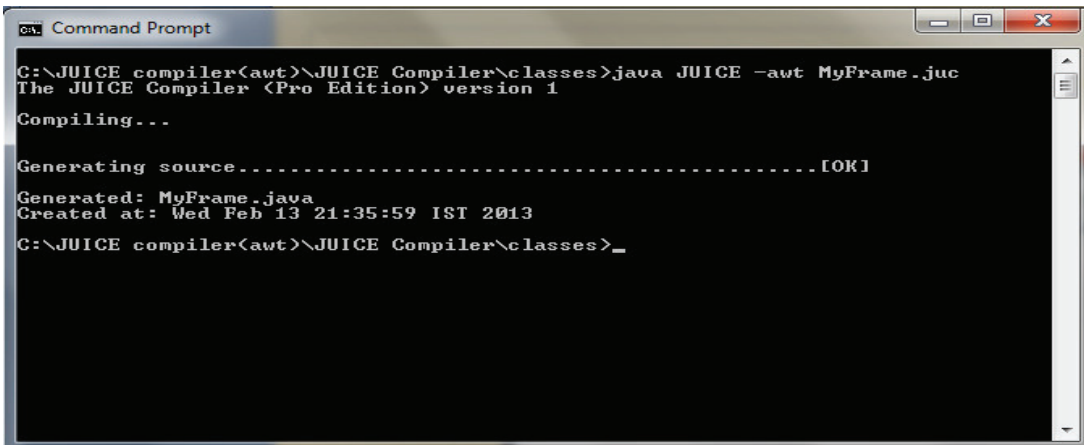


fig.JUICE EDITOR

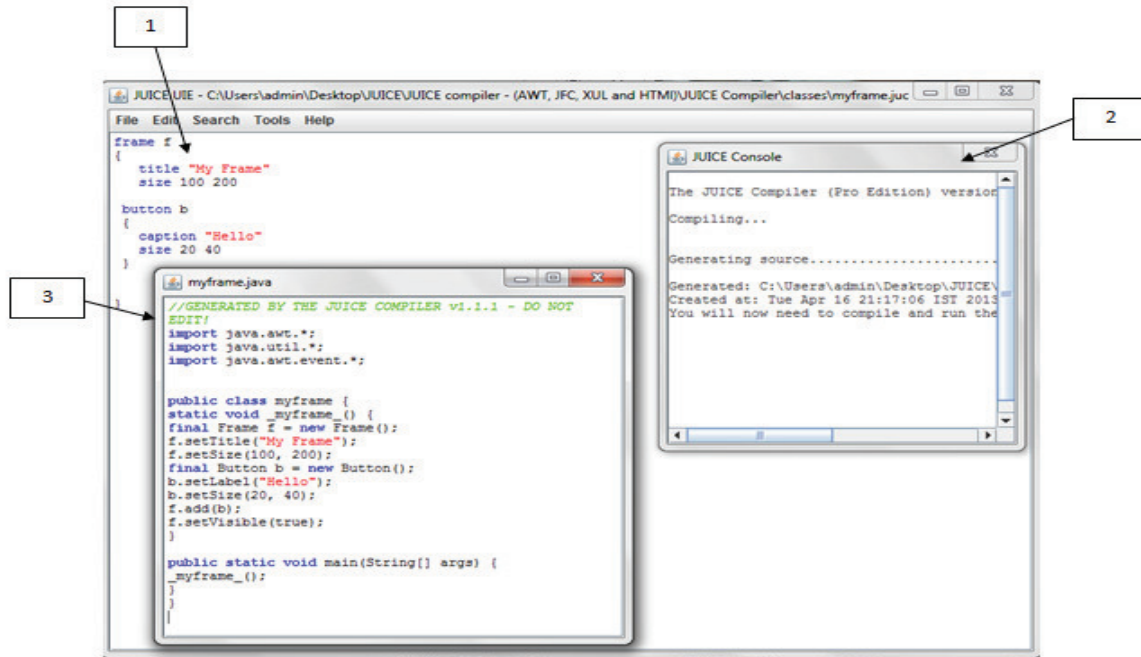
1.JUICE Editor 2.MenuBar 3.Menu

COMPILE AND RUN JUICE SCRIPT FOR AWT :-



Fig(B)

OUTPUT OF JUICE SCRIPT FOR AWT:-



Fig(C)

Figure (C) shows following points:

- 1] Shows JUICE code.
- 2] Shows Compiling and Running converted AWT code.
- 3] Shows converted AWT code.

V. CONCLUSION

This paper shows that JUICE is a rapidly evolving area of research and development. We discussed only the key problems in this area and presented some known solutions. One key research problem that we still face today is the development of truly easy, less complex, and time saving techniques for generating GUI in different languages.

REFERENCES

- [1] <https://javacc.dev.java.net/>
- [2] <http://www.scifac.ru.ac.za/compilers/conts.html>
- [3] <http://java.sun.com/products/jfc/download.html>
- [4] <http://download-llnw.oracle.com/javase/1.4.2/docs/api/java/awt/package-summary.html>
- [5] https://developer.mozilla.org/en/introduction_to_xul
- [6] <http://download.oracle.com/javase/tutorial/reflect/index.html>
- [7] <http://java.net>: JavaCC [tm]: Grammar Files
- [8] Alfred Aho and Jeffrey D. Ullman,(1986) Compilers: Principles of Compiler Design.
- [9] Advance programming in java NIIT, Prentice Hall of India, ISBN-81-203-2415-3
- [10] Raphael A. Finkel, Advanced Programming Language Design.
- [11] Andrew Appel, Jens Palsberg: Modern Compiler Implementation in Java. Cambridge University Press, 2nd edition, 2003.
- [12] Software engineering.
- [13] Benjamin Michotte, Jean Vanderdonckt, "GrafXML,AMultitarget User Interface Builder based on UsiXML".