
Multi Targeted Graphical User Interface Compiler

Sayali Ahire¹, Shubhangi Keskar¹, Shubhangi Devikar¹, Kalyani Gavade¹, Prof. H. H. Patel, D. V. Motghare

Computer Engineering, DCOER, Pune, India

Abstract: *Today it is difficult to design a good GUI in a acceptable language .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 in which he is developing an application. To overcome this problem we paln 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 IDE for writing the new language and for compiling it into the target language. The system consist of a compiler for compiling and translating MAGIC SCRIPT into target language which may be Java Swing, Java AWT, XULor HTML. The system must be expandable for inclusion of new language also.*

Keywords: *MAGIC, Multi-targeted compiler, JavaCC, XUL.*

1. INTRODUCTION

Using JavaCC parser generator the compiler has been implemented. It detects the tokens which described in MAGIC Script. As per the need of user the script has to be converted to the target language for making the compiler Multi-Targeted. The parsing errors will also checks the compiler takes input file with .mgk extension and convert it into specified target language. For the given target language it reference the class related to a token by adding the switch for language before the common class name for that token. An IDE for developing the MAGIC Script is to be provided for easier development of the MAGIC Scripts.

1.1. Proposed Algorithem

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 the switching mechanism.

Step5: The user has to select the through command switch.

Step6: The tokens and command switch is passed as an input to the switching mechanism.

Step7: Switching Mechanism is generators the class file name by following formula

Command Switch + token + Datatype

Step8: According to the command switch the class will be passed towards the code generator.

Step9: In the code generated the component will be selected as per the class name generated by

Switching mechanism.

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

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

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

1.2. Data Used

1.2.1. Jvider

It is the GUI builder tool for java swing applications. For the java applications and applets you can easily design the graphical user interface by jvider. It provide us drag table, understandable interface and the component which is resized. It will generate the source code in java language. It has ability to export the source code as frame or applet. It is also platform independent and runs on all environment that should be support by Java Virtual Machine.

Features of Jvider

- Standard Java Swing components are used.
- GridBagLayout is used for layout.
- Ability to save/load your designs.
- View and export produced java source code as frame or applet.
- Easy to get started with tutorial and samples provided.
- Powerful Undo/Redo.

1.2.2. GrafiXML

It is the software used to generate the graphical user interface(GUI) and save them in a UsiXML format language. It is basically work as other GUI builder but also contains tools to localize your applications.

It is similar to any user interface builder except that it manipulates more widget properties than physical ones and that it saves any user interface in UsiXML instead of particular code format. GrafiXML released under the terms of apache license, version 2.0. It is explained how the UI resulting from the design can support one or many levels of independence with respect to underlying context of use.

2. MATHEMATICAL MODEL

Consider a system S which can be defined as,

$$S=\{U,I,P,O\}.$$

Where I as input $I=\{CS,SC,syntax\}$

$$CS=\{L1,L2,L3,L4,L5\}$$

Where $L1, \dots, L5$ are target language. Since input is provided by user, user is the actor.

Where user $\in U$

Where P is processes in the system and represented as $P=\{P1,P2,P3,P4\}$

Where P1 is process of generating the tokens represented as $T=\{T1,T2,T3, \dots Tn\}$

Where P2 is process of generating the classname.

Where P3 is process of selecting the code generator.

Where P4 is the process of generating code for the sub-tokens.

3. SYSTEM STRUCTURE

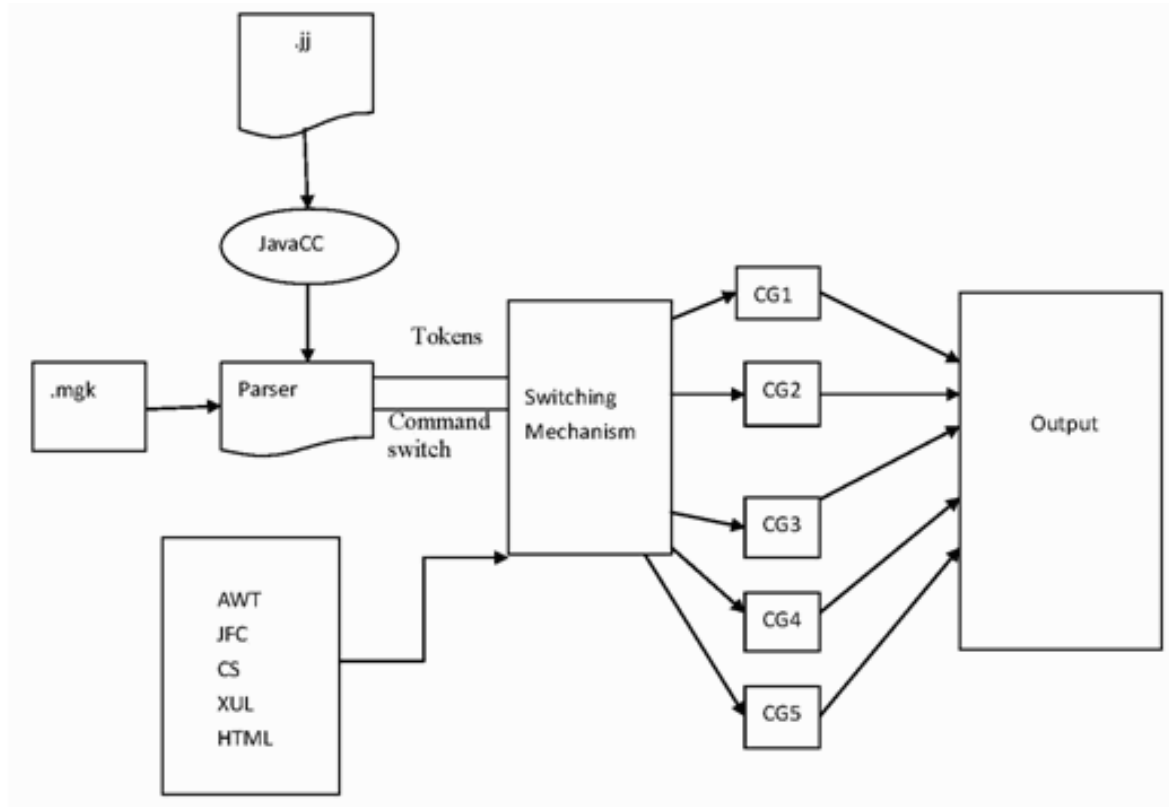


Fig1. System Architecture

4. CONCLUSION

MAGIC is rapidly evolving area of research and and development. We discussed the problems and presented some known answers. We found the less complex, easy and time saving technique for generating GUI in different languages.

REFERENCES

- [1] Dhiraj Motghare, A Java Based 4th Generation Multi-Targeted User Interface Compiler. /IJIET/2003.
- [2] <http://javacc.dev.java.net/>
- [3] <http://www.scifac.ru.ac.za./compilers/conts.html>
- [4] <http://java.sun.com/products/ifc/download.html>
- [5] <http://download-llnw.oracle.com/javase/1.4.2/docs/api/java/awt/package-summary.html>
- [6] http://developer.mozilla.org/en/introduction_to_xul.
- [7] Benjamin Michotte, Jean Vanderdonckt, "GrafiXML, AMultitarget User Interface Builder based on UsiXML".
- [8] Raphael A.Finkel, Advanced Programming Language Design.
- [9] Andrew Appel .lens Palsberg: Mosern Compiler Implementation in java. Cambridge University Press, 2nd edition, 2003.