THE PASCAL SCRIPT+ LANGUAGE AND ITS INTEGRATED DEVELOPMENT ENVIRONMENT

Marek Mittmann

Gliwice, Silesian University of Technology, Faculty of Automatic, Electronic and Computer Science

Abstract. The paper has been written in order to introduce the script language, which was created by the author. It will also present the environment of designing, developing and testing the scripts which are written in this language.

Introduction

The main reason why the author has created the Pascal Script Language+ was the need to extend their applications to possibility of customizing with scripts, which would be defined by user [1]. At the very beginning it consisted of only a few basic instructions, which were taken from Pascal Language. Afterwards it turned out that the users of application intensely took advantage of the possibility of writing scripts due to the program was extended to the version that is presented.

When there was need to use it in another programs, the modules, which are responsible for interpretation of scripts, were selected and they created the DLL library. The next stage of development was creating environment in order to write and run the programs which are written in Pascal Script+ in Windows System.

The Pascal Script+ Language

Pascal Script+ has been made to do macros and small scripts and it is not very complicated. In spite of the fact that the structure is very transparent and the number of keywords and operators is limited it can be used in order to write a lot of complicated algorithms [2]. Due to simplicity and similarity to Pascal it takes only a few hours to learn rules of the language.

Despite its simplicity Pascal Script + has wide abilities which have been presented below:

- partial compatibility with Pascal language,
- exception handling,
- variety of data types (integer, real, string and others),
- possibility to use predefined objects,
- possibility to use DLL libraries,
- support for call-back functions,
- the possibility to share into modules.

Current version of the language is equipped with the library which contains 150 procedures and functions and 15 objects. They create set of 300 different methods. Contents of library is big enough to create fairly advanced scripts. We can find following groups of functions:

- arithmetic operations,
- string handling,
- files handling,
- support for displaying windows,
- basic containers.

```
begin
  Write('Input file name: '); fileName := ReadLn;
  try
    file := CFileStream.Create(fileName, fmOpenRead);
    size := file.Size;
    if size = 0 then WriteLn('File is empty')
    else begin
      WriteLn('Contents of file:');
      for i := 1 to size do begin
        ch := file.ReadChar;
        if Ord(ch) >= 32 then Write(ch)
        else Write('.');
      end;
    end;
  except
    WriteLn('Error. Cannot open file ' + fileName);
  end;
end.
```

Fig. 1. An example of the Pascal Script+ source code

Compiler and Virtual Machine

Running the scripts consists of two stages. The first stage is a translation of an source text into intermediate cod. This process is executed by independent module which name is: compiler of the Pascal Script+ language. In the second stage the program interprets the cod with a use of a virtual machine of Pascal Script+. Program, which is generated by the compiler, consists of an instructions that is similar to internal instructions of typical processor. Due to that fact they work very fast.

We can distinguish a few functional blocks in the compiler of Pascal Script+ [3, 4, 5]. An lexical analyser reads signs from the entrance sequence. The line of signs, which has been read, goes through an automat. The automat recognizes keywords, operators, constants and identifiers. This automat was generated by the TPLEX program [6, 7]. The lines of the recognized symbols are passed to the syntax analyser. The analyser is created with TPYACC generator. During syntax analyse different functions are used in order to do a semantic analyses and to generate parts of the cod. Every instruction of the language is changed into the sequence of instructions of the virtual machine. The created cod may be optimised and supplied with instruction, which protects it from making mistakes. The manager of the identifiers array is a very essential element. It collects and searches information that are connected with identifiers, which are used in the script.

The compiler has been written in Object Pascal language [8].

Operation of the virtual machine is very simple. It takes instructions of the cod, recognizes them and does proper sequences of processor's instructions. The machine was created to gain the compromise between reliability and speed.

Compiler and a virtual machine are available in a few versions.

- DLL library, which can be used by programs working in Windows, it enables these programs to use scripts.
- independent command-line compiler and an virtual machine, which enable to use Pascal Script+ scripts in Windows system.
- Integrated Development Environment

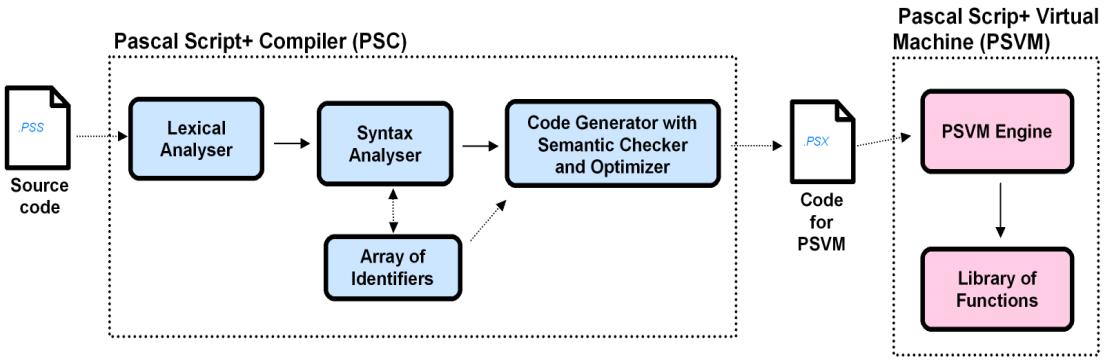

Fig. 2. Architecture of the Pascal Script+ environment

The Integrated Development Environment

Modern compilers are equipped with many tools, which help to write programs. Such elements like an editor which highlight keywords, a program wizards or an integrated help system are basic equipment and new compilers have them included. In spite of the fact that Pascal Script+ is not commercial product and it was created in very short time it is as good as other modern tools.

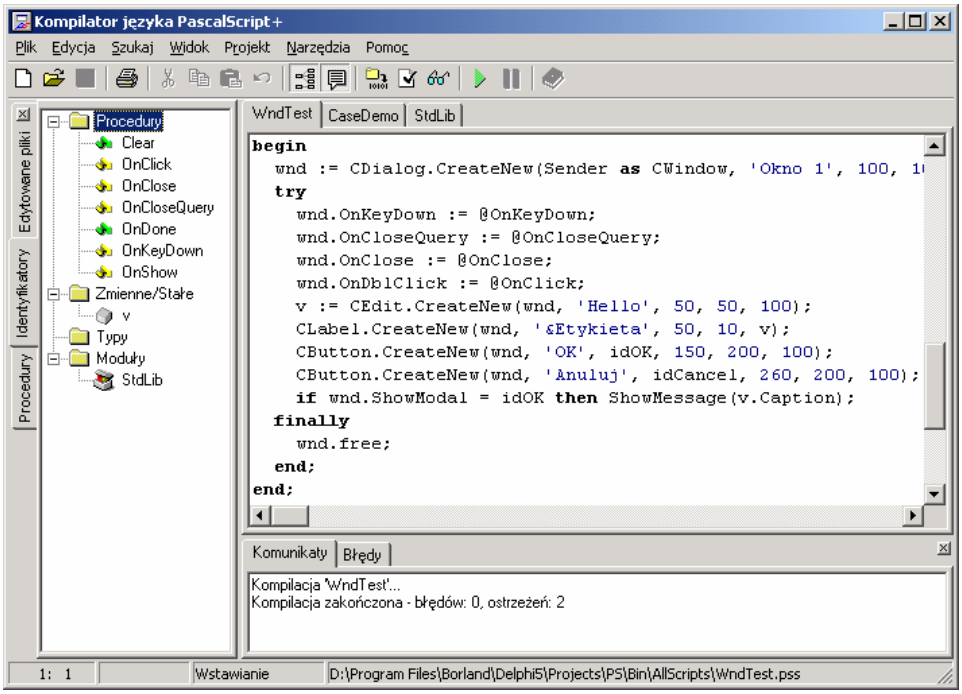

Fig. 3. The Pascal Script+ IDE

There is an editor in the main compiler. It is situated in the central part of the window. It enable to edit many modules in the same time. It is able to recognize keywords, identifiers, comments and another parts of the source text. It marks and distinguishes them with the use of different colours and fonts. A user can create its own set of colours or take advantage of prepared ones. There is also the possibility to use bookmarks in order to help with searching. Editors enable to do much operation on a text as for example: cutting, sticking and setting indentations etc. During the edition of procedure's callings there are hints, which contain the list of its parameters. Due to that fact the writer does not have to remember which arguments are taken to every single method.

Navigation panel is situated on the left side of editor. There are 3 different lists. It is possible to replace yourself among this list with the use of tabs, which is situated on the left side. The first list contains the names of editing files. If you choose one of them you can edit it. This list enables users to fast moving between modules. The second list contains the

identifiers of procedures, constants, variables and types. It helps to search declarations of them in the text quickly. The main aim of the third list is procedures and functions.

During writing text you are allowed to use different types of tools. One of them is the window with a list of all procedures and functions, which are defined in edited file, and it appears when users wish to and recall it. After choosing one of procedures or functions from the list, you are moved straightforwardly to the definition of it. You can search for one, you need, by writing the piece of the name. This toll helps to move in long scripts and it is not difficult to find the procedure in the text, which consists of thousands of lines.

Most of complex compilers have different creators, which can generate the frame of program and support its further edition. The compiler of Pascal Script+ language also has this kind of tools. Not only have users around 20 prepared wizards but also they can create their own or modify existing one.

Conclusion

The result of the researches on Pascal Script + language is modern environment, which is used in order to create and run small programs in Windows system and the library, which enables to add possibility of making macros and configuration scripts to applications. The presented method can be very useful to develop the ability of creating application without huge effort.

The presented version of the Pascal Script+ language is not the last one. It is still being developed and new tools are invented all the time. It is worth mentioning that we try to run scripts, which are written in this language, on the web pages. Moreover we intend to make tool to design graphical user's interface for scripts on the web page.

References

1. Mittmann M.: Projekt i realizacja mechanizmu kontroli dostępu do danych w programie KS-SWD. Praca dyplomowa magisterska, Instytut Informatyki, Politechnika Śląska 2001.
2. Nicholls J. E.: Struktura języka programowania. WNT, Warszawa 1980.
3. Aho A., Sethi R., Ullman J.: Compilers. Principles, Techniques, and Tools. WNT, Warszawa 2002.
4. Goos G., Waite W.: Konstrukcja kompilatorów. WNT, Warszawa 1989.
5. Gries D.: Konstrukcja translatorów dla maszyn cyfrowych. PWN, Warszawa 1984.
6. Niemann T.: A Compact Guide To Lex & Yacc.
7. Graef A.: TP Lex and Yacc – Users Manual. 1998.
8. Object Pascal Reference. Borland, 2002.