Global Journals LaTeX JournalKaleidoscope™
Artificial Intelligence formulated this projection for compatibility purposes from the original article published at Global Journals. However, this technology is currently in beta. *Therefore, kindly ignore odd layouts, missed formulae, text, tables, or figures.*

Green Computing using Perl and Python

Srivibha Vadravu

Received: 10 June 2021 Accepted: 5 July 2021 Published: 15 July 2021

Abstract

Green computing or clean computing is necessary for Software Engineering. Perl and Python are important programming languages for green computing. Perl is regular language. Perl is mainly used for server-side programming because it is a regular language. It a portable and green programming language. It can be used as object-oriented (OO) or non objectoriented (Non-O-O) programming language. Python is preprocessor. It is a portable language for software engineering. It has an import feature for Green computing.

Index terms— green computing, regular expressions objectoriented, perl, preprocessor, python

reen Computing or Clean Computing is necessary to solve different types of problem solving. The Programming is needed portability of the code, and less computation time. There are different techniques methods are used for Green Computing like recursion, parallelism, regular expression and Object-Oriented. The recursion is the calling function itself. Parallelism is computing the number of tasks at a time. The regular expression is simplifies the code. The Object-Oriented shall made program is independent.

1 II.

2 Green Computing Methods

Programming is the main component for problem solution. The Green programming has some of the main features.

3 Portability

4 c) Coding

The Programming Languages fall under different paradigms Imperative, Functional, Logical, and Object-Oriented and regular it is difficult to learn all the programming languages. It easy to learn programming languages through common principles like iteration, recursion, control statements, functions, functions, subroutines, Object-oriented, etc. All principles and techniques are not available in single programming language. The selected Programming Languages are discussed for Green Computing.

Programming languages are designed based on Automata. Context-Free Language is the recursively representation of Finite Automata.

For green computing, Recursive Algorithms and Parallel algorithms are used until recently. Programming languages are playing a main role.

We consider Perl and Python for green computing. Perl is the regular language. It simplifies the programming, and it reduces time.

Python is the preprocessing language. It simplifies the with the import feature, it simplify the code.

5 III.

6 Component Technology

All components are specialized, independently deployed and extendable for the product. These components are also extendable to multi versions of the components. The following are the characteristics of the components.

The components have an externally accessible view.

The semantics such as business rules and regulations are defined for the composition of components.

As Component software extended, the components are extendable.

The component must be relocate and replace a component for other implantations or the development of new software system.

The semantic primitives must be extendable to new components.

The composition of components is tightly coupled.

The components are substituted and integrated into the other systems. Sometimes this maybe referred to as off-the-components.

7 b) Component Implementation

The component model is translated into component ware with tools for automation and management of components and interfaces. Interface to understand system architecture with the interface specifications that implement, reuse, and replacement of components. They are two types of component ware implementation for products.

Self-development in which component were developed from the scratch.

Off-the-self components in which component ware developed by black box assembling commercially available components and such components are documented, assembled and adapted.

The following are the characteristics of the implementation enterprise model. The components of the product may represent entire system Generosity: It is stepwise instantiation and controlled processes that use specifications, inheritance, relationships and contexts.

Domain system: It represents a particular area of components.

Domain object: It represents a particular process of components.

Semantic primitives: These are rules and kinds of relationships between objects.

These domain concepts are used to compose domain components of individual components.

IV.

8 Perl Programming

The Programming Languages fall under different paradigms Imperative, Functional, Logical, and Object-Oriented and Regular. It is difficult to learn all the programming languages. It made easy to learn programming languages through common principles like iteration, recursion, control statements, functions, functions, subroutines, Object-oriented etc. All principles and techniques are not available in single programming language. The selected Programming Languages are discussed for Green Computing.

The Programming Languages are constructed mainly based on Finite Automata (FA) and Regular (RE).

The Formal Languages (FL) are simple representation of Context-Free Language) CFL). The CFL is recursion of FA. For instance, $n=<STDIN>; $factorial=fact($n); print "$factorial\n";

9 Green Computing Technology

Green computing technology mainly has two criterions fundamentals of computer science and nature of computer science.

10 a) Fundamentals of Computer Science

Fundamentals of computer science may be defined as

11 Conclusion

Perl and Python are best for Green Computing or clean computing. Perl is regular language and powerful at sever side programming. Python is preprocessor and it is portable with import feature. We try to discuss m Perl and Python programming languages for green computing. [1]

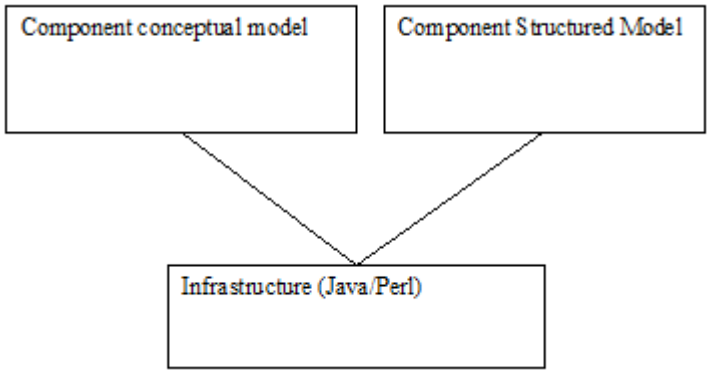
Component conceptual model
Component Structured Model
Infrastructure (Java/Perl)

Figure 1:

[Rumbaugh et al. ()] , J Rumbaugh , M Blaha , W Premerlani , F Eddy , W Lorensen , - Object . *Oriented Modeling and Design* 1991. Prentice-Hall.

[Zave and Jackson ()] 'A Component-Based Approach to Telecommunication Software'. Pamela Zave , Michael Jackson . *IEEE Software* 1988. p. .

[Ben-Shaul et al. ()] *An Integrated Network Component Architecture*, Israel Ben-Shaul , James W Gish , William Robinson . 1998. IEEE Software1998. p. .

[Digre ()] 'Business Object Component Architecture'. Tom Digre . *IEEE Software*, 1998. p. .

[Szyperski ()] *Component software: Beyond Object-Oriented Programming*, C Szyperski . 1998. Reading, Mass: Addision Wesley Longman.

[Booch ()] 'Component-Based Software Engineering'. Grady Booch . *IEEE Software* 1998. p. .

[Cox ()] *Object Oriented Programming: An Evolutionary Approach*, B J Cox . 1987. Addison Wesley Longman, and Reading, Mass.

[Booch ()] *Object-Oriented Analysis and Design with Applications, Second Edition*, G Booch . 1994. Benjamin/Cummings, Redwood city,CA.

[Venkata Subba and Reddy ()] 'Object-Oriented Software Engineering through Java and Perl'. P Venkata Subba , Reddy . *CiiT International Journal of Software Engineering and Technology* 2010. 5 p. .

[Weyuker ()] 'Testing Component-Based Software: A cautionary Tale'. Elaine Weyuker . *IEEE Software* 1998. p. .

[Brown and Wallnau ()] *The current state of CBSE*, Alan W Brown , Kurt C Wallnau . 1998. p. .

[Booch et al. ()] *The Unified Modeling Language-Use Guide*, G Booch , J Roumbaugh , I Jacobson . 1999. Reading, MA: Addison-Wesley Longman mc.