International Journal of Applied Engineering Research
ISSN 0973-4562 Volume 10, Number 12 (2015) pp. 30961-30967

http://www.ripublication.com

Survey On Reusable Software Component Repository In Green Technology

***Pradeep Kumar and **Dr. Shailendra Narayan Singh**

**Ph.D. Research Scholar, Dept. of Information Technology,*
AIIT, Amity University, Noida
e-mail: bhartipradeep77@gmail.com
** Associate Professor, *Dept. of Computer Science & Engineering*
ASET, Amity University, Noida
e-mail:snsingh36@amity.edu;sns2033@gmail.com

Abstract

Space, time, cost and energy etc are all linked with each other and energy consumption is dependent upon all these factors. Processing of software contributes significantly in the heat generation. Due of scarcity of resources especially energy, lots of activities has been done towards saving of energy. World is inching towards the green technologies. There are various measures of green effect has been develop over the period of time. Software engineering is not an exception. Software scientists are contributing through implementing various efficient measures in the development of software life cycle. Global warming and its negative effect on the environment prompt us to do research based on efficiency of reusable software components with repositories on green issues parameters. Integration of previous developed software into new model is become easier and its environmental impact will be positive. Many researchers advocated Software Information Base (SIB) and similar practices in the industry to make software development process more efficient but new methods points out many lacunae in the SIB also. In this research, we are trying to critically examine the views of different researchers on the Green software engineering (GSE) and their contribution. We will also suggest outline framework to make more efficient software engineering practices.

Keywords: Green computing, Green software engineering (GSE), Efficient computing, Software information based (SIB), Information Storage and Retrieval, Information Search and Retrieval.

INTRODUCTION

Green software engineering (GSE) is the need of the hour. Green computing is the essence of software engineering. Software reuse is the sustainable approach during crisis. It gives positive impact on environment. Other related problem of environment can also be dealt with reusability of software. Especially problems related to sustainable practices. Reuse word is very promising has been implemented since inception object oriented concepts. Reuse concept itself tending towards green concepts and its contribution towards saving of cost, space, energy etc can't be marginalized. After many researchers and findings from different people time to time, it's still barren on quantitative front. There is lack of calculative measures on each and every front. Reuse word is used as synonyms in the area of code, software objects; information's and extended towards specifications, development process and decision making. Software developments process which is develop on reuse philosophy strict itself in a bounded area of object oriented systems and now it needs to be revisit concept with the new process and thinking's. Storage of software takes much space and reuse of software modules also stores the same part of software modules in many other software's. In spite of developing new codes we are using same code again and again with a single development. Use and reuse of software components are dependent upon many factors. Sometime it needs customization and many times not. Integration of modules is the biggest challenge of the software reuse process. Reuse-based development cycle highlights the importance of reuse-targeted repository, together with its associated functions/tools. Requirement of repositories can be dealt with the same pace but other assumptions or problems related to it must be tackling with the same pace. Use and reuse of software components and modules can be act similar to the once develop and many use systems but its needs to revisit the concept to make it more and more green technologies. Reuse of software components has great relevance in software engineering technologies. This study is based on the existing theories developed on the reuse of software as a first phase of towards green software engineering and tries to suggest second generation concepts of green software engineering which is more efficient computing than the existing one.

LITERATURE REVIEW

Software engineering is continuously witnessing the positive role of software development processes in green computing. Extension of reusability of components and other parts is one of the examples of the green computing. There were many researches which estimated energy consumption of software. Seo, et al. [20] and they work on different framework for estimating the energy consumption. Previously, framework designs were focuses on architecture of components and connectors and their role saving energy. In many cases it may be useful but few. This framework was based on development stages. According to the findings researchers came to the conclusion that life of system and its green contribution is directly proportional to each other. Gupta and Singh [10] did work on the issues related to energy consumptions and suggested many measures based on architecture and protocols.

According to Gupta and Singh, the most effective ways to implement green software engineering are

1. Components can be used as only when it is need to be used.
2. Components will sleep when it is not being used.

They said that it can be implemented with the suitable changes in the internet architecture and current protocol. Author further suggested new efficient models for energy conscious software people. Environmental issues will be considered as a prime agent in the same regards [11], "software will play an increasing role in the environmental impacts of pervasive computing". Some of the theorist argues that while hardware is becoming cheaper, we should more focus on the functionality of the software and its branches. Chetty, et al. [5] studied the power consumption of computers by observing how people in their homes use built-in power settings. Green computing and sustainable energy issues are very big and there are many pros and cons of these. So, implementation policies of the process need to develop with lots of care. These findings could also be applied to software energy consumption: while the decrease in energy consumption of one user switching from Firefox to IE may not be significant, the collective energy savings could be significant and thus provide a motivation to users to make the change. Additionally, Chetty, et al. [5] found that transparency is must for the software industry at all levels in efficiency front. Green software and its tracker must include more advanced and relevant parameters in the same regards. Set of keywords is the most important asset in the development of software and its development process. By analogy with what we find in the subject index of a library of books, a software asset is represented by a set of keywords [15]. While one of the key areas of software reusability is its parameters on the basis of software components will be searched. Searching of components consumes a large part of energy. Keywords fixation is the major areas where we need to be focused our study. While the state-of-the-art has advanced beyond simple keyword representations, the notion of attaching a set of words to an asset is still in use in modern reuse terminology, where the process of describing components is referred to as classifying them [14]. Keyword setting is much more difficult in cloud based data and software components because of its nature (variety, veracity, volume etc). Maintenance of software is much more difficult in this software and there is dearth of literature available in this area. The approach most widely discussed in the reuse literature is the faceted approach introduced by Prieto-Diaz [16]. With the help of roots of key words and extension of these, searching process will become more easy and transparent in an unstructured manner. An example of such facets might be [19]: function, object/item-type, medium, system-type, functional-area, and setting. In [3], Boerstler proposes with his feature-oriented classification a hierarchical refinement of the faceted approach. What he calls a view-refinement corresponds essentially to classical facets. As a given component would be classified according to various facets, it can be classified according to the various attributes one might attach to a feature (or, in the authors terms: views of a feature). However, the view categories can be refined, such that a faceted tree structure emerges. When a sufficient level of detail has been obtained, one switches from the view-refinement to is refinement.

Here, one out of a choice of conceptually disjunctive attributes is selected to characterize the component. This attribute can be further refined into (disjunctive) sub-categories. Certain knowledge based approaches can be considered as another form of elaboration of descriptive methods. The classification of knowledge based approaches in general provides some difficulties, since their degree of formalization is certainly above those of conventional keyword based or faceted descriptive approaches. However, as long as the core semantics are based on the natural-language semantic of strings, it seems fair to discuss them in this context. In addition to conventional knowledge based approaches based on semantic nets or rules, case based reasoning approaches to the reuse problem have emerged recently [9]. Their merits is on one hand, that they can deal with quite intricate patterns, while on the other hand, one does not need to perform an extensive domain analysis to come up with highly discriminative descriptors. The different aspects representing the dimensions of green software engineering are covered in the GREENSOFT Model [17]. The GREENSOFT Model is a conceptual reference model supporting IT professionals and software users in the sustainable development and usage of software. It consists of four parts: A life cycle of software products, criteria and metrics, procedure models, and tools. The life cycle of software products considers ecological, social, and environmental aspects of a software product over its whole life. According to Life Cycle Thinking, it follows the motto “from cradle to grave” [24] and comprises the following phases: Development, Distribution, Acquisition, Deployment, Usage, Maintenance, Deactivation, and Disposal [8]. Some basic criteria and metrics are presented that can be used to evaluate the effects and impacts of a software product on sustainable development. Even some more criteria and metrics are proposed in [2, 4, 22] and [12]. In order to sum up these criteria, Kern et al. [13] introduced a quality model for green and sustainable software. Especially regarding metrics, it is essential to apply measurement methods, e.g. to measure, estimate, and rate the energy consumption of ICT. Here, different approaches exist. Some examples for measuring the energy consumption of software are described in [6, 7, 18] and [25]. The procedure models included in the reference model examine the development, purchasing, administrating, and usage of software. Another approach of a Green Software Development is presented by Shenoy et al. [21]. Additionally, recommendations and tools to support the different stakeholders are integrated in the GREENSOFT Model [19]. Agarwal et al. [1] list some more environmental friendly good practices in development and implementation of software systems. The overall software process is considered by Albertao et al. [2]. In view of the fact that the common assumption states that software is in general “environmentally friendly”, they introduce metrics that can be assessed in a real software project. The approach presented shows that it is feasible to continuously improve software projects regarding sustainability issues by measuring a set of metrics repeatedly over several iterations. Another definition of Green Software is given by Taina [22] who requires green software to fulfill the following three high level requirements:

1. The required software engineering processes of software development, maintenance, and disposal must save resources and reduce waste.
2. Software execution must save resources and reduce waste.

3. Software must support sustainable development.

Software development process and purpose is not tough and it is alien with the requirements. It also support green computing issues. From his point of view, this can be true for software that has a special sustainability supporting purpose, e.g. a software that improves the effectiveness of a wind-driven power station, but it cannot be true for a general purpose software, where it is not possible to know for which secondary purposes it may be used: a CAD software may be used to design a lignite power plant but it may be also used to design a wind mill or any other product, which in turn makes it nearly impossible to determine the secondary effects of its application.

FUTURE CHALLENGES AND GUIDELINES

Since recorded history of use and reuse of software components in software engineering witness many innovative changes. History is full of literature on software engineering issues and prospective solutions but dearth of literature available on green software technologies on recent challenges of cloud based computing. All branches of computer science are inching towards the greener technologies has been working on many related issues. Now, due to emerging area of green software engineering there are many pros and cons generated recently. Below is the few recent challenges we are noticing on the green software engineering front and they are as

a) Software components database need to be generated along with alignment of more advanced technologies like cloud computing, Big Data etc.
b) Graphical methodologies are needed to be introduced to make software engineering more efficient and greener.
c) It requires to be developed more quantitative formulae to measure efficiency of cloud based software engineering green computing.
d) New platform based technologies requires to develop, so that it can easily helps in migration of existing software engineering

CONCLUSIONS

In this paper we gave broad spectrum of green software engineering issues and challenges in alignment new technologies. After review of existing literature on the issues of green technologies we come to the conclusion and below given finding –

a) Software engineering processes are passing through very crucial stage of transformation from traditional to green technologies.
b) While platform of software's are changing continuously, developing processes for new platforms in more efficient way is challenging issues and need to be address.
c) Integration of new platform with other new advanced technologies is upcoming challenge for software engineering scientists.
d) Ownership of cloud based software will become challenge in cloud based software engineering software's and process.

e) Security will become one more challenge for system designers in green software engineering software's.

REFERENCES

[1] Agarwal, S., Nath, A., and Chowdhury, D, "Sustainable Approaches and Good Practices in Green Software Engineering", IJRRCS 3, 1, pp. 1425–1428, 2012.

[2] Albertao, F., Xiao, J., Tian, C., Lu, Y., Zhang, K. Q., and Liu, C., "Measuring the Sustainability Performance of Software Projects", International Conference on e-Business Engineering, pp. 369–373, 2010.

[3] Boerstler, J., "Feature Oriented Classification for Software Reuse," International Conference on Software Engineering and Knowledge Engineering, SEKE, pp. 204–211, 1995.

[4] Capra, E., Francalanci, C., and Slaughter, S. A., "Measuring Application Software Energy Efficiency", IT Professional, pp. 54-61, 2012.

[5] Chetty, M., Brush, A.B., Meyers, B.R., and Johns, P., "It's not easy being green: Understanding Home Computer Power Management", International Conference on Human Factors in Computing Systems, ACM, pp. 1033-1042, 2009.

[6] Dick, M., Kern, E., Drangmeister, J., Naumann, S., and Johann, T., "Measurement and Rating of Software-induced Energy Consumption of Desktop PCs and Servers", International Conference (EnviroInfo), pp. 290–299, 2011.

[7] Dick, M., Kern, E., Johann, T., Naumann, S., and Gülden, C., " Green Web Engineering-Measurements and Findings", International Conference (EnviroInfo), pp. 599–606, 2012.

[8] Dick, M. and Naumann, S., "Enhancing Software Engineering Processes towards Sustainable Software Product Design", International Conference on Informatics for Environmental Protection, pp. 706–715, 2010.

[9] Gomez, H., "A Cognitive Approach to Software Reuse Applying Case-Based Reasoning to Mutant Cases," International Conference on Software Engineering and Knowledge Engineering (SEKE), pp. 512–519, 1997.

[10] Gupta, M. and Singh, S., "Greening of the internet", Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications SIGCOMM, ACM, New York, NY, pp. 19-26, 2003.

[11] Jain, R. and Wullert II, J., " Challenges: Environmental Design for Pervasive Computing Systems", International Conference on Mobile Computing and Networking, ACM,pp. 263-270, 2002.

[12] Johann, T., Dick, M., Kern, E., and Naumann, S, "How to Measure Energy-Efficiency of Software: Metrics and Measurement Results", International Workshop on Green and Sustainable Software (GREENS),pp. 51–54, 2012.

[13] Kern, E., Dick, M., Naumann, S., and Guldner, A. J. T., "Green Software and Green Software Engineering – Definitions, Measurements, and Quality

Aspects", International Conference on Information and Communication for Sustainability,pp. 14-16, 2013.

[14] Karlsson, E.-A., Software Reuse – A Holistic Approach, Wiley, Chichester, West Sussex, UK, 1995.

[15] Matsumoto, Y., "A Software Factory: An Overall Approach to Software Production", ITT Workshop on Reusability in Programming, IEEE Computer Society Press, 1993.

[16] Prieto-Diaz, R., "A Software Classification Scheme," PhD Thesis, Department of Computer and Information Science, University of California at Irvine, Irvine, CA, 1985.

[17] Naumann, S., Dick, M., Kern, E., and Johann, T., "The GREENSOFT Model: A Reference Model for Green and Sustainable Software and its Engineering", SUSCOM, vol.1, issue 4, pp. 294–304, 2011.

[18] Procaccianti, G., Vetro, A., Ardito, L., and Morisio, M., "Profiling Power Consumption on Desktop Computer Systems", Information and Communication on Technology for the Fight against Global Warming", pp. 110-123, 2011.

[19] Prieto-Diaz, R., "Classification of Reusable Modules," Software Reusability, Volume I: Concepts and Models, ACM, pp. 99-123, 1990.

[20] Seo, C., Malek, S., and Medvidovic, N., "Estimating the Energy Consumption in Pervasive Java-Based Systems", Annual IEEE International Conference on Pervasive Computing and Communications. PERCOM, IEEE Computer Society, pp. 243-247, 2008.

[21] Shenoy, S. S. and Eeratta, R., "Green software development model: An approach towards sustainable software development", India Conference, pp. 1 - 6, 2011.

[22] Taina, J., "Good, Bad, and Beautiful Software-In Search of Green Software Quality Factors", CEPIS UPGRADE XII, issue 4, pp. 22–27, 2011.

[23] TestNG testing framework documentation. http://testng.org/ doc/ documentation-main. html. Accessed, 2013.

[24] Tischner, U., Dietz, B., Mabelter, S., Schmincke, E., Prösler, M., Rubik, F., and Hirschl, B., "How to do EcoDesign: A guide for environmentally and economically sound design", 2000.

[25] Wang, S., Chen, H., and Shi, W. 2011. "SPAN: A software power analyzer for multicore computer systems." SUSCOM, pp. 23–34, 2011.