ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 6, June 2016

GCBSEEUC: A New Technique to Compute Energy Consumption in CBSE Software

Vijay Kumar Sinha [1], Mohita Narang[2]

Research Scholar, Dept. of Applied Science, Chandigarh Group of Colleges, Mohali, Punjab, India[1]

Assistant Professor, Dept. of Computer Science, Chandigarh Group of Colleges, Mohali, Punjab, India[2]

ABSTRACT: Green computing is newer research area where the researcher tries to develop energy efficient computing to save the environment and energy wastage. In software industry repetitive writing the same codes, developing same application again and again and serving unnecessarily useless components are the great hurdles in the field of achieving green computing targets. In this research we present a model of green software development by means of using component based software . As the components once developed for some client can be used for another client without recoding .This reduce the redevelopment time. The ready components can be used by another software suppliers as a reedy product, thus again saving of significant energy. The components can be reused with minor customizing for different client. This save the time and money as well and enhance profitability. In this paper we developed a model for CBSE softwares to achieve the energy efficiency goals. We proposed 4 levels of model in which first model includes the design and development phase,. The second level includes the user's level customization for energy efficiency. In the third level we focus on energy saving by auto saving and data back up as data loss wastes a lot of energy to regain the data. In the fourth stage we focus on disposal, reengineering, reuse of software components. We successfully demonstrated that our proposed component based software development model is highly efficient to achieve the environment friendly energy efficient green computing targets. We propose an energy consumption calculation formula GCBSEEU for CBSE software.

KEYWORDS : Green Computing , Energy Efficiency , CBSE , Code reuse Green ICT, Green and Sustainable Software Engineering, Green and Sustainable Software, Sustainable Development, GPIs, ASD, GORE ,GCBSEEU

I. INTRODUCTION

The Component based software engineering is probably best suited software development techniques available so far that perfectly suited for energy efficiency green computing model.

Recently many efforts have been done in obtaining green software. Some efforts are focused on building green and sustainable software, some design software processes to aid all stakeholders in building green software products. Others efforts are focused on building software tools that measure the effect of software on the environment and the effect of application development environments on the software in terms of energy efficiency. There are efforts that emphasize on the operating system to help control the power consumption of applications.

A variety of research work on Green ICT has mainly focused on environmental sustainability in terms of computer hardware. But revealing the issues related to energy consumption in software can be a great help in achieving green computing. Software features are responsible for CO_2 emissions as are hardware components. Software has an indirect effect on the environment by operating and managing the underlying hardware running it. Some software based solutions can monitor and utilize resources efficiently and others can be sustainable enough to limit the need of adding more hardware due to updates. Unfortunately there is a lack of models and work in the area of computer software and software development processes.

General software solutions found in [4, 5] include virtualization, closing applications no longer in use, efficient algorithms by writing a compact design of codes and data structures, reduction of parallelism overhead by developing efficient load balancing algorithms, fine grained green computing, and creating energy allocation algorithms for routing

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 6, June 2016

data. Naumannet al., [1] came up with a conceptual reference model named GREENSOFT model for sustainable software. Their four part model supports software developers, administrators, and software users in creating, maintaining, and using software in a green manner. The four parts cover a life cycle model, metrics, procedure models, and recommendations and tools for different stakeholders. Shenoy and Earatta in [38] also provide a green development model in which they suggest steps that may lead to lower carbon emissions in the software life cycle stages. Mahaux and Canon [2] argue that requirements engineering is critical to the whole software life cycle primarily in the usage phase where customers are delivered the system and expect it to conform to their requirements. They claim that correct requirements engineering can help software last longer thus reducing the energy consumption. Capra et al., [9] focus on developing a measure of energy efficiency for software applications and illustrate how application development environments can have a detrimental effect due to the additional lines of code they add. Gupta and Singh in [8] present a framework for creating an intelligent power profile that implement three methods at the time of login into the system. These methods continuously measure the power consumption of running software in a given period of time and can be incorporated in operating systems.

With the growing demand of more complex software applications, Information and Communication Technology (ICT) has had a huge negative impact on the environment due to its increasing resource and power consumption. The effect of ICT on sustainable development [1, 12] especially on software is the hot topic now-a-days in Green Computing. Sustainable development refers to resource use for meeting the needs of humans while taking into account the ecological, economic, and societal impacts. Although ICT recently has been trying to find efficient solutions for the environment, it is not clear whether energy and resource savings by ICT will exceed its resource consumption.

Research that focuses on the design of code and how it may cause bloating are found in [13]. An effort spent on energy efficiency through adding more cores on a single CPU can be found in [14]. Works that focus on the importance of requirements engineering for sustainable software are found in [22, 23, 24, 30, 33]. In [22] the software requirements engineering process focusing on sustainability requirements for a company named 'The Yellow Project' is reported.

In [10] an approach based on periodic measurements of GPIs and QoS(Quality of service) and adoption of Service Oriented architecture is used to optimize energy efficiency at the Software-as-Service layer. Efforts spent on integrating sustainability in Service-Oriented software are found in [29, 31]. In [11, 12] efforts are spent on defining general good practices in green software engineering such as collecting requirements through electronic means and deploying the concept of virtualization. An approach to greenify service based applications is achieved through integrating eco-aware requirements based on energy goals are found in [23]. In [24] it is argued that green ICT concepts related to software requirement engineering should be added to undergraduate software courses. In [30] a requirements. Engineering approach is developed that allows engineers to handle sustainability as a first class quality objective. . Addressing sustainability of software processes is found in [32]. In [34] efforts are spent in having clear metrics for measuring the carbon footprint of software development, the amount of resources used by software, and how much damage it does to the environment.

Work found in [33, 39] focus on quality engineering based on the measurements of software in terms of quality metrics. Works that are dedicated for sustainable development in computer science are found in [3, 37].

This paper focuses on achieving green and sustainable software by building a green and sustainable software model that will aid software engineers in the development process of software and include the recent approaches taken by software to ensure the safety of the environment.

II.OBJECTIVE OF RESEARCH

The principal objective of this research includes:

a) To Study and examine the existing Green Computing Techniques and Models
b) Development of a model to reduction of energy wasting by using component Based Software .
c) Comparison of existing green computing models with CBSE computing models.

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 6, June 2016

III.PROBLEM STATEMENT

As the increasing threats of global warming and the strict guidelines of environment regulatory bodies it is necessary to build energy efficient software which saves energy wastage on various levels and thus save energy finally .Which in turn contributes its part to the environment friendly eco system to reduce the energy wastage?

Suitability aspects of CBSE software towards green computing:

The Component based software have several unique features which make it highly efficient for green computing such as:

1) The component based software are need not to be programmed again and again for different applications as it can be directly attached as a separate component and then integrated with the systems as detachable component. Thus saves a lot of human efforts which in turn saves the energy wastage in development stage.
2) The component developed for a system can be easily debugged, recycled, reengineered and reused. This saves a lot of software development time. Thus contributes in energy saving.
3) The component based software can be easily customized for different client's need and budget as it can be easily added or removed from the system. This saves the customization time for clients and thus saves energy.
4) There are many freely available open source components are available for developers, by use of these freely available components we further reduce the development cost as well as time , which in turn contributes significantly in achieving green computing targets.
5) At end user level it can be further customized as per the requirements of the user's needs by enabling or disabling the software sub-components. This further reduces the processor cycle and memory consumption. This in turn saves energy and attributes in green computing targets.

   The above mentioned featured clears that the CBSE based software are highly adaptable for achieving green computing targets.

IV.RESEARCH METHODOLOGY: CASE STUDY

We use Case study method, modeling and statistical analysis for implementing the proposed model.
We use ERP based software which is component based and having vast components for this case study.

V.PROPOSED MODEL OF GCBSE

Green computing is newer research area where the researcher tries to develop energy efficient computing to save the environment and energy wastage. In software industry repetitive writing the same codes, developing same application again and again and serving unnecessarily useless components are the great hurdles in the field of achieving green computing targets. In this research we present a model of green software development by means of using component based softwares. As the components once developed for some client can be used for another client without recoding .This reduce the redevelopment time. The ready components can be used by another software suppliers as a reedy product, thus again saving of significant energy. The components can be reused with minor customizing for different client. This save the time and money as well and enhance profitability. In this paper we developed a model for CBSE software to achieve the energy efficiency goals. We proposed 4 levels of model in which first model includes the design and development phase,. The second level includes the user's level customization for energy efficiency. In the third level we focus on energy saving by auto saving and data back up as data loss wastes a lot of energy to regain the data. In the fourth stage we focus on disposal, reengineering, reuse of software components. We successfully demonstrated that our proposed component based software development model is highly efficient to achieve the environment friendly energy efficient green computing targets.

GCBSE model in Development phase: Stage-I

Energy Units Consumptions (EUC) in GCBSE:

GCBSEEUC = (∑Time spent + ∑ Line of Codes in all components)*∑ Human Involved

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 6, June 2016

Here GCBSEEU = Green Component Based Software Engineering Energy Units Consumptions

In the development phase of green computing we reduce the energy wastage in every stage of SDLC life cycle. Requirement Design Unit Testing Implementations Increment System Testing Release of System, Uses, Green analysis

VI. IFW ERP EDUTECH CBSE SOFTWARE

Sl No.	SDLC Stages	Lines of Codes	Time Spent (Days) 1st Client	Number of Human Individuals	GCBSE EUC (I)	Time Spent (Days) 2nd Client	Number of Human Individuals 2nd client	GCBSE EUC (II)	Energy savings GCBSEEU (I-II)
1	Requirement Gathering	0	5	1	5	3	1	3	2
2	Designing	0	25	5	125	12	10	120	5
3	Coding / Customization	546000	315	24	13111560	25	9	4914225	8197335
4	Integration	0	25	1	25	5	4	20	5
5	Testing	0	65	5	325	12	10	120	205
6	Delivery and Installation	0	12	8	96	9	5	45	51
7	Maintenance		10	7	70	8	2	16	54
8	Disposal	0							
Total		546000	457	51	13112206	74	41	4914549	8197657
Energy Savings = 62.519 %									

Stage –II : Energy Saving By Customization of Components

VII. ENERGY SAVING BY REDUCTION OF OPTIONAL SUB COMPONENTS

CBSE software are highly adaptive towards customization. The customization of enables reduce the unnecessary sub components and make it as per user's need.

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 6, June 2016

Components of IFW Campus ERP Modules:
Table -I

IFW Campus ERP Solutions Major Software Components				
	Administrative Login Components :	Student Login Software Components :	Parent Login Components :	Employee Module Components
1	Student listing	Fees module	Campus news	Subject specialization report
2	Faculty listing	Due fees report	Due fees report	D.R.R. (daily report register)
3	Faculty + student logs	Fees invoice	Fees module	Milestone management
4	Download manager	Fees deposited	Fees invoice	Access exam time table
5	D.R.R. (daily report register)	Attendance record	Fees deposited	Attendance of students
6	Academic calendar	Academic time table	Student guidelines	AttendanceReport
7	Milestone management	Exam time table	Attendance record	Book submission reminder
8	Milestone status of faculties	Exam sitting arrangement	Book submission reminder	Access salary information
9		Performance in exam	Academic time table	Student guidelines
10		Library book searching & reservation	Performance in exam	Campus news
11		Teacher evaluation test	Exam sitting arrangement	Input exam date
12		Student guidelines	Training & placement	Access time table
13		Book submission reminder	Exam time table	Downloads
14	Library book searching & reservation	Downloads	Academic calendar	Self-attendance
15	Training & placement management	Academic calendar	News manager	Training & placement
16		Training & placement	Downloads	Library book searching & reservation
		Campus news		Academic calendar

Here the Red marked components are Essential Components and Blue marked components are Optional components. Reduction of optional components reduces the development time significantly as well as reduce the memory consumption and CPU cycles.
On a average a CBSE software contains ¼ fraction of optional sub components.
In a typical condition we can achieve nearly 25% of energy savings by reduction of optional subcomponents .

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 6, June 2016

VIII.THE GRAPHICS AND ANIMATION COMPONENTS

Graphics and animations are the sub-components of CBSE software.
It consumes significant computer memory as well as CPU cycle. This result a slower computing experience. Which in turn consume time and energy.
The animator sub components are:

1. Various Effects (Smoothing, fading, Shining, Rotating etc.)
2. Moving Animations (Video , Picture)
3. Graphics
   For GCBSE model we reduce these components as they enhance only appearance but reduce the system performance.

Fig.1 CPU and Memory Uses with full animation

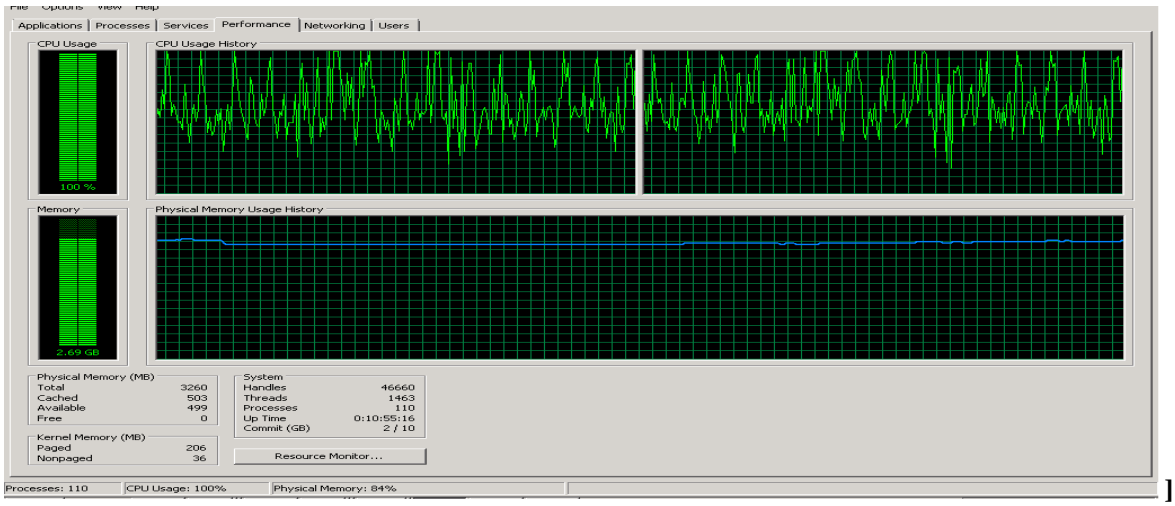
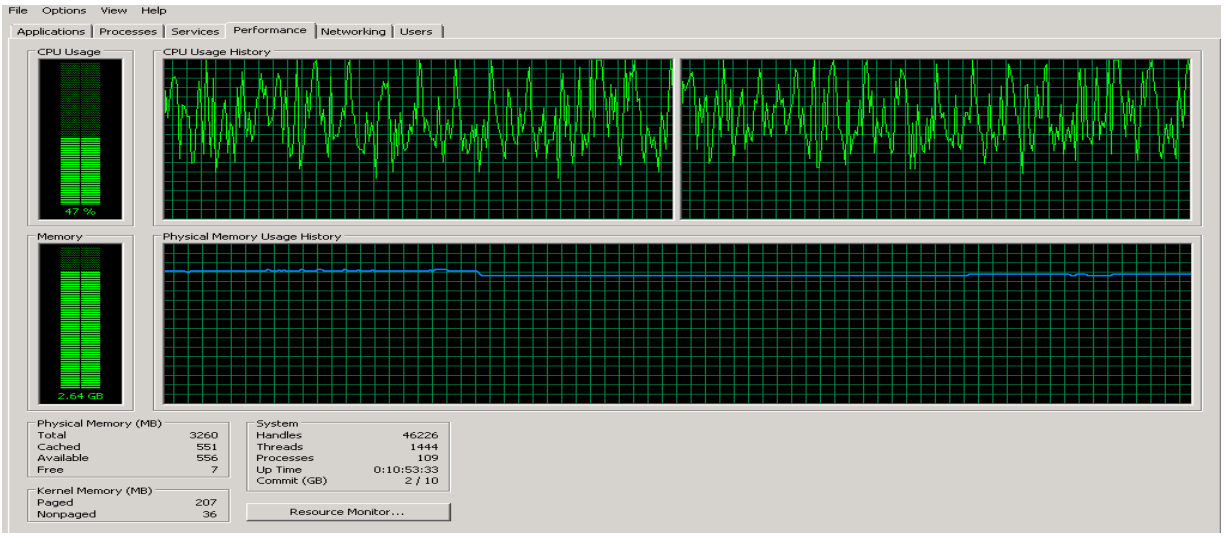


Fig:2 CPU and Memory Uses with Reduced Animation



ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 6, June 2016

Energy Saving Model of Data Protection in GCBSE:

Data losses affects the business badly as well as spoils the precious human energy . We propose a model The designing of auto saving GCBSA as well as a backup module to keep the file and work save .

IX. AUTO SAVE MODEL OF GCBSE

In this auto save model of GCBSA software we design an auto saved duplicate copy of the documents which keep saving the work continuously as temporary file. Whenever the work completed it replace those temporary files with the permanent file.
This keep our work when power failure in desktop or Laptop battery down cases. This saves human efforts and thus contributes the Green Computing Goals.

X. GCBSA DATA BACKUP DESIGN

The data loss might be occurs due to viruses , system crash or any physical damage to the system . When the data is lost then it again spoils precious human energy to recover and recreate the same data. In commercial organization it may cause severe business losses.
We design the GCBSA data backup in 3 levels:
Data backup within the stand alone system. In a stand alone system we keep the data in admin protected separate disk drive. By this our data remains safe when operating system fails. Or the OS containing drive goes formatted.

1. Alternatively the GCBSE keep the data on a removal storage like flash
2. drive or portable HD.
3. In network environment we design to keep the data on a data backup server connected with local LAN.
4. In public network environment we keep the data on cloud repository like Google Drive or on a remote server .

By means of this model we achieve the green computing goals up to a significant level and minimize the energy wastage on each level of component Based Software computing.

XI. RESULT AND DISCUSSIONS

We achieved significantly ¬62.5% in energy savings during the software development stages. We further achieve almost 25% energy reduction by means of reduction of optional sub-components. We further minimize the energy wastage by deactivating the animation and graphics components.
By means of data auto saving and data backup plan we protect the system from data losses and thus saving of energy spoiling. Thus we succeed to achieve the maximum energy savings without compromising the system performance.

XII. CONCLUSION

In this paper we successfully demonstrated the various energy saving techniques with help of developing a GCBSE model. We further developed an energy measuring formulae as GCBSEEUC = (∑Time spent + ∑ Line of Codes in all components)* ∑ Human Involved .We achieved significantly ¬62.5% in energy savings during the software development stages. We further achieve almost 25% energy reduction by means of reduction of optional sub-components. We further minimize the energy wastage by deactivating the animation and graphics components.

XIII. FUTURE SCOPE

This GCBSE model further can be expanded to hardware level, Operating System level and Network level. The effects of band width on network in case of cloud computing like 2G 3G, 4 g can also be optimized for GCBSE.

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 6, June 2016

ACKNOWLEDGEMENT

We are thankful for the IKG Punjab Technical University, Kapurthala (India) for the opportunity provided us to conduct current this research work on Reduction of Cyclomatic Complexity by use of Mutation Testing . We also grateful to the Chandigarh Engineering College, Landran , Mohali (Punjab) affiliated to IKG PTU , Kapurthala for supporting and providing environment for our research work . The active and constant support of our department and research supervisors being the main source of inspiration led this research a success.

REFERENCES

[1] S. Naumann, M. Dick, E. Kern and T. Johann, "The GREENSOFT Model: A reference model for green and sustainable software and its engineering", Sustainable Computing: Informatics and Systems, vol. 1, no. 4, pp. 294-304 ,2011.
[2] M. Mahaux and C. Canon, "Integrating the Complexity of Sustainability in Requirements Engineering", 1st international workshop on Requirements for Sustainable Systems, pp. 1-5,2012.
[3] B. Penzenstadler and V. Bauer, "Jumpstart Sustainability in Seminars: Hands-on Experiences in Class", 2nd Computer Science Education Research Conference, pp. 25-32,2012.
[4] A. Govindasamy and S. Joseph, "Optimization of Operating Systems towards Green Computing", International Journal of Combinatorial Optimization Problems and Informatics, vol. 2, no. 3, pp. 39-51.2011.
[5] C. T. D. Lo and K. Qian, "Green computing methodology for next generation computing scientists", IEEE 34th Annual Computer Software and Applications Conference (COMPSAC), pp. 250-251,2010.
[6] S. Wang, H. Chen and W. Shi, "SPAN: A software power analyzer for multicore computer systems", Sustainable Computing: Informatics and Systems, vol. 1, no. 1, pp. 23-34,2011.
[7] A. Noureddine, A. Bourdon, R. Rouvoy and L. Seinturier, "A preliminary study of the impact of software engineering on GreenIT", 2012 IEEE First International Workshop on Green and Sustainable Software (GREENS), pp. 21-27,2012.
[8] P. K. Gupta and G. Singh, "A Framework of Creating Intelligent Power Profiles in Operating Systems to Minimize Power Consumption and Greenhouse Effect Caused by Computer Systems", Journal of Green Engineering, vol. 1, no. 02, pp. 145-163,2011.
[9] E. Capra, C. Francalanci and S. A. Slaughter, "Is software "green"? Application development environments and energy efficiency in open source applications", Information and Software Technology, vol. 54, no., pp. 60-71,2012.
[10] F. Oliveira and T. Ledoux, "Self-optimization of the energy footprint in Service-Oriented Architectures", ACM International Workshop on Green Computing Middleware, pp. 1-6,2010.
[11] G. Sissa, "Green Software", UPGRADE, vol. X1, no. 3, pp. 53-63,2010.
[12] S. Agarwal, N. Asoke and C. Dipayan, "Sustainable Approaches and Good Practices in Green Software Engineering", International Journal of Research and Reviews in Computer Science (IJRRCS), vol. 3, no. 1,pp. 1425-1428,2012.
[13] S. Bhattacharya, K. Gopinath, K. Rajamani and M. Gupta, "Software Bloat and Wasted Joules: Is Modularity a Hurdle to Green Software?", IEEE Computer, vol. 44, no. 9, pp. 97-101.2011.
[14] A. Marowka, "Back to thin-core massively parallel processors", IEEE Computer, vol. 44, no. 12, pp. 49-54.
2011
[15] A. Kipp, T. Jiang, M. Fugini and I. Salomie, "Layered green performance indicators", Future Generation Computer Systems, vol. 28, no. 2, pp. 478-489,2012.
[16] F. Berkhout and J. Hertin, "Impacts of Information and Communication Technologies on Environmental Sustainability: Speculations and Evidence", SPRU (Science and Technology Policy Research,pp. 1-21,2001.
[17] N. Amsel, Z. Ibrahim, A. Malik and B. Tomlinson, "Toward sustainable software engineering: NIER track", 2011 IEEE 33rd International Conference on Software Engineering (ICSE), pp. 976-979,2011.
[18] C. Sahin, F. Cayci, J. Clause, F. Kiamilev, L. Pollock and K. Winbladh, "Towards power reduction through improved software design", IEEE Energytech,, pp. 1-6,2012.
[19] T. Johann, M. Dick, S. Naumann and E. Kern, "How to measure energy-efficiency of software: Metrics and measurement results", 2012 IEEE First International Workshop on Green and Sustainable Software (GREENS), pp. 51-54,2012.
[20] S. S. Mahmoud and I. Ahmad, "Green Performance Indicators for Energy Aware IT Systems: Survey and Assessment", Journal of Green Engineering, vol. 3, no. 1, pp. 33-69.
[21] F. Albertao, J. Xiao, C. Tian, Y. Lu, K. Q. Zhang and C. Liu, "Measuring the sustainability performance of software project", 2010 IEEE 7th International Conference on e-Business Engineering (ICEBE), pp. 369-373,2010.
[22] M. Mahaux, H. Patrick and G. Saval, "Discovering sustainability requirements: An experience report", Requirements Engineering: Foundation for Software Quality, pp. 19-33,2011.
[23] J. C. Deprez, R. Ramdoyal and C. Ponsard, "Integrating Energy and Eco-Aware Requirements Engineering in the Development of Services-Based Applications on Virtual Clouds", CETIC, First International Workshop on Requirements Engineering for Sustainable Systems, pp. 1-7,2012.
[24] F. Ahmed and K. Shuaib, "Incorporating Green IT concepts in undergraduate software requirements engineering course: An experience report", 2012 IEEE 7th Iberian Conference on Information Systems and Technologies (CISTI), pp. 1-4,2012.
[25] S. Misra, V. Kumar, U. Kumar, K. Fantazy and M. Akhter, "Agile Software Development Practices: Evolution, Principles, and Criticisms", International Journal of Quality & Reliability Management, vol. 29, no. 9, pp. 973-979,2012.
[26] C. Manteuffel and S. Ioakeimidis, "A systematic mapping study on sustainable software engineering: A research preview", 9th Student Colloquium@RUG 2011-2012, pp. 35-39,2012
[27] D. M. Raffo, W. Harrison and J. Vandeville, "Software Process Decision support: making process tradeoffs using a hybrid metrics, modeling

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 6, June 2016

and utility framework", Proceedings of the 14th ACM international conference on Software engineering and knowledge engineering, pp. 803-809,2002.

[28] D. Stefan, E. Letier, M. Barrett and M. S. Sawicki, "Goal-oriented system modeling for managing environmental sustainability", 3rd Workshop on Software Research and Climate Change,pp. 1-4,2011.

[29] M. Razavian, D. A. Tamburri, Q. Gu and P. Lago, "Modeling to support communication and engineering of service-oriented software", 2012 IEEE Workshop on European Software Services and Systems Research-Results and Challenges (S-Cube),pp. 8-9,2012.

[30] B. Penzenstadler, B. Tomlinson and D. Richardson, "RE4ES: Support Environmental Sustainability by Requirements Engineering", 1ST International Workshop on Requirements Engineering for Sustainable Systems, pp. 1-6,2012.

[31] P. Lago, T. Jansen and M. Jansen, "The Service Greenery-Integrating Sustainability in Service Oriented Software", International Workshop on Software Research and Climate Change (WSRCC), pp. 1-2,2010.

[32] G. Lami, F. Fabbrini and M. Fusani, "Software Sustainability from a Process-Centric Perspective", Systems, Software and Services Process Improvement, vol. 301, no. 1, **(2012)**, pp. 97-108.

[33] B. Penzenstadler, "Supporting Sustainability Aspects in Software Engineering", 3rd International Conference on Computational Sustainability, pp. 1-4,2012.

[34] J. Taina, "Good, Bad, and Beautiful Software–In Search of Green Software Quality Factors", Green ICT: Trends and Challenges, vol. XII, no. 4, pp. 22-27,2011.

[35] T. Dyba and T. Dingsoyr, "Empirical studies of Agile Software Development: A Systematic Review", Information and Software Technology, vol. 50, no. 9, pp. 833-859,2008.

[36] T. Dingsoyr, S. Nerur, V. G. Balijepally and N. B. Moe, "A Decade of Agile Methodologies: Towards Explaining Agile Software Development", The Journal of Systems and Software, vol. 85, pp. 1213-1221,2012.

[38] T. Johann, M. Dick, E. Kern and S. Naumann, "Sustainable Development, Sustainable Software and Sustainable Software Engineering", IEEE (eds.): SHUSER International Symposium on Humanities, Science and Engineering Research, pp. 34-39,2011.

[39] S. Shenoy and R. Eeratta, "Green Software Development Model: An Approach towards Sustainable Software Development", 2011 IEEE Annual India Conference (INDICON), pp. 1-6,2011.

[40] B. Penzenstadler, "Towards a Definition of Sustainability in and for Software Engineering", 28th ACM Annual Symposium on Applied Computing (SAC), pp. 1-3,2013.

BIOGRPHY

	Vijay Kumar Sinha received his M.Tech. degree in Computer Science and Engineering from Punjabi University , Patiala. Currently he is persuing his Ph.d. degree in Computer Science & Engineering from IKG Punjab Technical University , Kapurthala (Punjab) India.His research interests include Image Processing , Software engineering , Software components, Software Reuse, Software architecture and software metrics.
	Ms. Mohita Narang is working as Assistant Professor at Chandigarh Engineering college ,Landran , Mohali (Punjab).She received her M.tech degree from Punjabi university. Her area of research interest includes software engineering , Software reuse , Green Computing.