Leveraging "Energy Efficiency to Software Users"

[Summary of the Second GREENS workshop, at ICSE 2013]

Patricia Lago
VU University Amsterdam
The Netherlands
p.lago@vu.nl

Niklaus Meyer
Green IT SIG
Swiss Informatics Society,
Switzerland
niklaus.meyer@acm.org

Maurizio Morisio
Politecnico di Torino, Italy
maurizio.morisio@polito.it

Hausi A. Müller
University of Victoria, Canada
hausi@cs.uvic.ca

Giuseppe Scanniello
Universitá della Basilicata, Italy
giuseppe.scanniello@unibas.it

DOI: 10.1145/2557833.2557859 http://doi.acm.org/10.1145/2557833.2557859

ABSTRACT
The Focus of the GREENS workshop is the engineering of green and sustainable software. Our goal is to bring together academics and practitioners to discuss research initiatives, challenges, ideas, and results in this critically important area of the software industry. This second edition of the workshop was held at ICSE 2013 in San Francisco, CA, USA. The theme of GREENS 2013 is Leveraging "Energy Efficiency to Software Users." It featured a keynote talk, ten research papers and three breakout sessions that discussed topics that ranged from qualities vs. energy efficiency and environmental sustainability, to green models and views for (software) products/process and to stakeholders, relevant metrics and measurements. In this report, we present the themes of the workshop, and summarize the results of the discussions held in the breakout sessions, as well as the identified research challenges for future investigation.

Categories and Subject Descriptors
D.2 [**Software Engineering**]: Miscellaneous; D.2.1 [**Software Engineering**]: Requirements; D.2.8 [**Software Engineering**]: Metrics—*complexity, product, process*; D.2.10 [**Software Engineering**]: Design

Keywords
Energy efficiency, sustainability, software engineering, green IT

1. INTRODUCTION
Information and Communication Technology (ICT) accounts for approximately 2% of world CO2 emissions [3]. This 2% includes only the *in-use phase* of hardware: in the remaining 98% software both operationalizes the private sector in doing its business and the public sector in supporting society, as well as delivering end-user applications that permeate our personal lives. Software can contribute to decrease power consumption (i.e. become greener) in at least two ways. First, by being more energy efficient, hence using less resources and causing fewer CO2 emissions. Second, by making its supported processes more sustainable, i.e. decreasing the emissions of governments, companies and individuals. To this end, enterprise software must be completely rethought to address sustainability issues and support sustainable and innovative business models and processes.

The special theme of the second[1] edition of GREENS [2] is leveraging "Energy Efficiency to Software Users." The goal of this edition of the workshop was to bring together software engineering researchers and practitioners to discuss the state-of-the-art and state-of-the-practice in green software, as well as research challenges, novel ideas, methods, experiences, and tools to support the engineering of sustainable and energy efficient software systems.

[1]The report on the first GREENS edition is available in [1].

2. THE KEYNOTE TALK
The keynote was given by Prof. Dr. Schahram Dustdar from Technical University Vienna on Green Software Services - From Requirements to Business Models. This inspiring keynote stressed the analysis of key elements of Green Software Services (GSS) from the business perspective. Schahram provided a systematic, high-level view on the three key elements that motivate the development of GSS: stakeholders, their requirements, and business models. He also discussed how to apply GSS to various business domains, such as smart buildings and smart transportation, and business models related to cloud services and the challenges in realizing a marketplace of GSS.

3. BREAKOUT SESSIONS
The workshop accepted ten papers for inclusion in the proceedings[2]. The papers can be divided into two distinct categories: those focusing on **GSS** and those focussing on **metrics and measurements** at various levels of abstraction in the green software lifecycle. The authors of accepted papers were invited to present their ideas to the workshop in the form of position statements. The topics identified as most interesting by attendees were: (1) sustainability for software-intensive systems; (2) green metrics and measures; and (3) stakeholders interested in sustainability. All these three topics were selected for further discussion in the breakout sessions. In the following subsections, we present the discussion outcomes.

3.1 On Sustainability for Software-Intensive Systems
The discussion started with the question:

How can we frame energy efficiency and environmental sustainability as quality requirements?

The group agreed that relevant sustainability dimensions are (see also Figure 1) social sustainability, environmental sustainability, economic sustainability, and technical sustainability. While the

[2]Papers accepted for GREENS 2013 are available from the ACM Digital Library.

first three dimensions derive from consolidated definitions of sustainability, the latter is specific for IT and refers to traditional, *system* qualities.

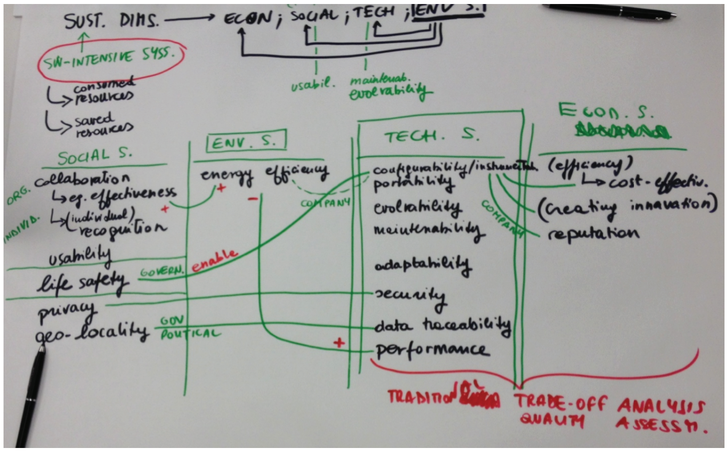

Figure 1: Sustainability dimensions for Software Quality

We think the following observations can be useful for further reflection and research:

A framework framing green software quality. It is already extremely difficult to identify, address and balance quality concerns in non-energy-aware software. In addition, when green concerns enter the picture, they are even more pervasive and potentially impact all other system qualities, hence demanding for an additional dimension that frames both types of qualities for trade-offs analysis and decision making. As sketched in Figure 1, energy efficiency classified a sub-characteristic of *environmental sustainability*, can positively or negatively influence other qualities like collaboration effectiveness (sub-characteristic of *social sustainability* or performance (sub-characteristic of *technical sustainability*). The Figure is the result of our discussion of specific case studies the participants were encountered in their past research. However, the essential message is that making explicit the quality concerns of green software involves complex impact dependencies (illustrated by the links between quality sub-characteristics), roles played on such dependencies (which can be positive or negative), and the context of the dependency. For example, in Figure 1 under column about 'social sustainability', collaboration effectiveness is relevant in the context of an organization (see label ORG.), while recognition is relevant quality for individuals (see label INDIVID.). Framing this information in a quality framework will help software engineers in making the right design decisions. Software quality models, standards, and green labels should be extended or defined, to address these aspects.

The role of stakeholders. The context of the dependencies introduced above seem to directly depend on which stakeholders are involved. The questions arising here are: what stakeholders do matter in software sustainability and what type of dependencies do they influence?

Green trade-off analysis. Traditional trade-off analysis of software systems typically addresses the technical and economic sustainability dimensions: the first concerns the qualities realized by the software solution, the second concerns the related business investments and economic benefits. We must be aware that green trade-off analysis spans all four sustainability dimensions. As such, assessment methods for energy-aware software must be revisited and extended to address them all. This is an important concern and deserves future research.

Sustainability goals and design concerns. The design concerns related to all sustainability dimensions should be made explicit in the same way as they are for (technical) design and architectural concerns. Architectural views and viewpoints for sustainability should be defined. This way, they can aid sound design decision making. We should further investigate if sustainability concerns can rely on the same mechanisms already defined for architecture descriptions, or if they require the definition of new mechanisms.

Environmental sustainability needs context. While technical sustainability is system-specific and can be assessed by analyzing the system itself and at most its execution environment (i.e. the set of technical aspects governing a software-intensive system), environmental sustainability seems to require much broader contextual information to ensure sound reasoning. Such context is multi-dimensional, multi-disciplinary and extremely complex. Environmental sustainability resembles a so-called *panarchy*, i.e. governance that would encompass all others [Wikipedia]. The implications of this meta level will be certainly subject to future research.

3.2 On Stakeholders and Metrics for Energy-Efficient Software

The discussion started with the question:

Who is or should be interested to get energy efficient software?

The group agreed that the following stakeholders should be interested in energy efficient software:

Government/Policies Makers. The government and enterprise levels (Corporate Responsibility) are interested to get energy-efficient software to achieve lower carbon footprints. They could with proper incentives and guidelines become important drivers for energy-efficient software.

Developers. Their interest is to develop energy-efficient code and report about successful techniques.

Datacenter services. Their interest is in the use of software to efficiently manage data centers from the energy perspective. Therefore, all the software providers in this segment are stakeholders, but also device manufacturers who need to provide smart energy management software included with their hardware.

Marketing. Marketing and sales persons can be considered as stakeholders because they want to use energy efficiency features as a sale driver.

Users. End-users of remote devices are stakeholders for energy efficient software. For example, they want to use their devices for a longer time without recharging the batteries.

A second part of the discussion concerned the types of metrics necessary to understand the level of greenness a software. First, we discussed the focus on sustainability. To achieve it, two perspectives on software domain were identified (or better, confirmed) to

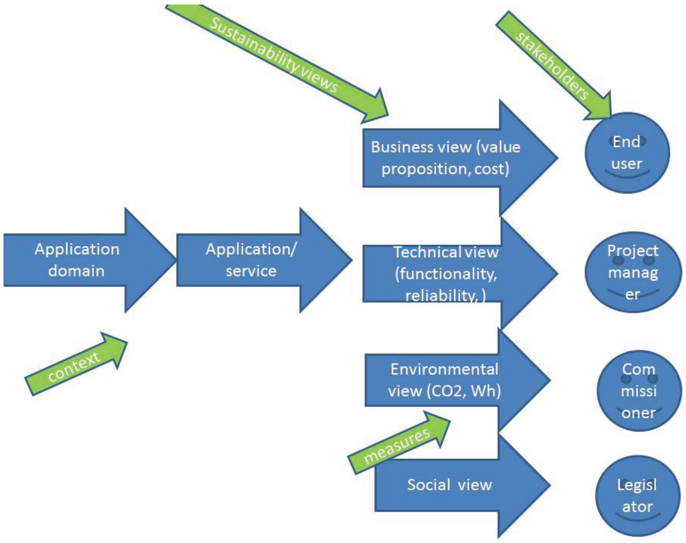

Figure 2: A proposed Sustainability model

have a direct impact on sustainability: software operation (e.g., energy consumption, energy features, the execution environment) and software development (e.g., develop for energy efficiency). Second, we discussed the following issue related to energy efficiency:

"Any research about measuring energy-efficient code is wrestling with the isolation of the respective software components from the environment it runs in. There are many influences from the hardware, software and even lighting and temperatures in the server room, which impact measurements."

The discussion group identified that the needs are stable reference models and a definition of the unit of measurement (e.g., grams of CO2). That is, before starting any major measurement effort the requirements for future comparisons should be defined including the part of sustainability to be addressed. In this concern, the group identified the following practical challenge: in many cases the software is embedded in an environment one cannot access. For example, a server where one cannot go inside the box to get the desired measuring points.

Based on the experiences of the participants in the group one broad and interesting conclusion emerged: *Identify program code that is wasting energy and include this knowledge in the software development guidelines such that non-energy-efficient code is avoided.* This will allow developing software by observing an energy saving discipline - avoiding bad habits and bad code - and will allow going somewhat around the mentioned complicated measurement challenges. Therefore, the group identified future research topics. In particular, the participants believed that future research should focus on the analysis of code (in any environment), which is not energy-efficient and on the development of guidelines, tips and tricks to avoid this kind of code.

Other results of the discussions. Research would have to go on to measure software energy footprints. Two identified possibilities are: (1) software energy labels and (2) green Service Level Agreements (SLA). One relevant challenge is to make the software developers aware that there could be an energy consumption problem.

3.3 On Green Metrics and Measures

The goal of the group was to discuss measures and metrics for green software. The group decided to organize green metrics in a general framework, that is presented in Figure 2. Interestingly, the framework is very similar to the one developed by the other group and presented in Figure 1. The core of the model uses four sustainability views (business, technical, environmental and social). Each view is a starting point to define related green measures. However, it is not possible to define measures that are suitable to all cases. The context of each case has to be taken into account to define the relevant metrics. The context is represented in the leftmost part of the figure. The context is defined in terms of application domain (i.e., industry, banks, insurances, health, etc.) and application or service (i.e., account management or investment management in the banking domain, warehouse management or order management in the industry domain, and so on). The definition of the measures depends also on the stakeholders involved. This is represented in the rightmost part of the figure. The stakeholders identified are end user, project manager, commissioner, and legislator. The set of stakeholders partially overlaps with the ones defined by the group of stakeholders, and clearly the two sets should be merged. Given a context as defined above, it is then possible to specialize a set of measures. The group proposed a generic and high level set of measures. Regarding the business view, the families of measures proposed are cost (in terms of the various stakeholders, such as end user, commissioner, etc) and value (especially for the end user and commissioner). Regarding the technical view, the families of measures proposed are the traditional ones (aka ISO 25010), such as functionality, reliability, usability etc. At this regard, some of the measures can be coupled directly with the environmental ones. For instance, energy consumption in terms of Watt-hours can be considered at application level, or at function level. Regarding the environmental view, the main measures proposed are consumption (in terms of energy, power) and polluting emissions (such as CO2). In summary, given the proposed framework, the group suggests adapting the measures proposed considering the specific context.

4. ACKNOWLEDGMENTS

GREENS is part of the dissemination activities of the Dutch Knowledge Network Green Software, and the European Fund for Regional Development under project MRA Cluster Green Software. We extend our thanks to those who have participated in the organization of the workshop, and particularly the program committee members. We also thank the participants in the discussion groups at GREENS 2013. The list of participants can be found in the GREENS website `http://greens.cs.vu.nl`.

5. REFERENCES

[1] P. Lago, R. Kazman, N. Meyer, M. Morisio, H. A. Müller, F. Paulisch, G. Scanniello, B. Penzenstadler, and O. Zimmermann. Exploring initial challenges for green software engineering: summary of the first GREENS workshop, at ICSE 2012. *SIGSOFT Softw. Eng. Notes*, 38(1):31–33, Jan. 2013.

[2] P. Lago, N. Meyer, M. Morisio, H. A. Müller, and G. Scanniello. 2nd International Workshop on Green and Sustainable Software (GREENS 2013). In *Proceedings of International Conference on Software Engineering*, pages 1523–1524. IEEE/ACM, 2013.

[3] C. Pettey. Gartner estimates ICT industry accounts for 2 percent of global CO2 emissions. http://www.gartner.com/newsroom/id/503867.