

A Master Program on Engineering Energy-Aware Software

Patricia Lago¹

Abstract

Software-intensive systems support most if not all aspects of modern society and in all industrial sectors. Because of software' pervasiveness and crucial role, its energy consumption and impact on sustainability cannot be neglected anymore. As computing resources are necessary to run software, and as software runs more and more globally (e.g. in data centers and by cloud providers), time has come to engineer *energy-aware software*. This means educating current professionals and future generations with the appropriate skills and competencies.

This paper describes the design of our program in "Software Engineering and Green IT" within our Computer Science curriculum and extending our Master Track in Software Engineering. We also introduce four types of modules that we defined to design our program, and for each type discuss the underlying rationale, learning objectives, risks and related mitigations, with the idea to offer them as building blocks for other educators.

1. Introduction

Software-intensive systems support most if not all aspects of modern society. Because of their pervasiveness and crucial role, their energy consumption and their impact on sustainability cannot be neglected anymore. So far modern software engineering implicitly assumes endless resources (like processing power, data storage, network speed, and energy). We know by now that computing resources, indeed, become increasingly more powerful and less expensive. However, the energy necessary to keep them on and available is becoming scarce, and is nowadays a major global problem that all major nations (as well as the European Union) aim at tackling aggressively.

As computing resources are necessary to run software, and as software runs more and more globally (e.g. in data centers and by cloud providers), time has come to engineer *energy-aware software*. This means educating current professionals and future generations with the appropriate skills and competencies. This paper describes the design of our program in "Software Engineering and Green IT" within our Computer Science curriculum.

To start providing these competencies, few professional education programs have been recently emerging worldwide: they are mostly centered on best practices and initial competencies that allow optimizing the use of (hardware) computing resources. Further, in the Netherlands some higher education institutes are starting offering specific programs related to Green IT², but most of them are at the Bachelor level (in universities of applied sciences, or Hoogschool), and no one addresses software engineering. With this proposal we offer a Software Engineering (SE) program addressing related Green IT topics, in general, and specifically of developing energy-aware software-intensive systems.

To integrate energy-awareness in our SE program we have adopted a mix of the *distributed approach* and the *centralized approach* identified in [5]. Accordingly, we both revisited pre-existing courses across the whole curriculum (distributed approach), and created few dedicated

¹ VU University Amsterdam, Department of Computer Science, p.lago@vu.nl.

² Green IT (Information Technology) is used here in its broad definition, i.e. "The study and practice of designing, manufacturing, using, and disposing of computers, servers, and associated subsystems efficiently and effectively with minimal or no impact on the environment" [8].

specific courses (blended approach). Such mixed approach is quite difficult but is said to bring a greatest impact. To ensure a sustainable implementation of the new program, which will start in academic year 2014/2015, we planned it carefully.

In this paper we describe the program and discuss the rationale, learning objectives, risks and related mitigations for both revisited modules and new ones. We also compare and contrast the program against the ones that already exist. By disseminating our plans we aim at starting a conversation within the community at large on providing higher education in the fields of energy-aware software engineering, and hopefully in creating new opportunities for our students to extend their horizon to collaboration outside the strict world of computer science, and perform their studies in collaboration with other disciplines.

In perspective, this track will open up for future potential inter-disciplinary and multi-disciplinary education: sustainability can be defined along four dimensions, these being environmental-, technical-, social-, and economic sustainability [3]. While software engineering would address technical- and environmental sustainability more centrally (e.g. system qualities like reliability and energy efficiency, respectively), both economic sustainability (e.g. returns on the investments in green software) and social sustainability (e.g. software supporting/influencing more sustainable social behaviors) are very relevant, and have natural links with topics like lifestyle informatics, big data, healthcare, smart cities, transportation (just to mention few).

2. Related Work

Green IT and the impact of technology on the environment have gained importance, and attention, in the past years thanks to the commitment of organizations (like Greenpeace) and international agreements of leading nations worldwide. However, while the key role of technology in supporting our society is undisputable, its energy- and sustainability footprint is still unclear, and highly underestimated.

Some reasons contributing to this lack of attention include, in our opinion, the relative young age of IT, the complex relation IT has with all sustainability dimensions of our society, and the intrinsic interdisciplinary nature of green IT. In addition, and sadly enough, greening their IT is never a top priority in the agenda of companies, in spite of the enormous expenditures dedicated to IT with a dramatic, steep increase every year – hence hindering progress towards a more mature sustainable society.

We lack understanding of the areas that can significantly reduce the IT energy- and sustainability footprint, and we lack mechanisms that help us gaining this understanding. To address this issue we need to equip current and future generations with socio-technical skills and competencies to understand the impact of IT on sustainability, and in particular energy efficiency, identify areas for improvements, and develop solutions to realize such improvements.

While the impact of (hardware) technologies on e.g. energy consumption and the whole lifecycle is gaining maturity, hence becoming mainstream in commercial organizations and markets, the impact of software systems and applications is almost neglected and misunderstood at best: many still believe that the impact of software is negligible, or irrelevant, in spite of the fact that computing technologies are developed and deployed with the purpose to run software on top of them; and in spite of the acknowledged exponential growth of data volumes (see big data phenomenon and the uncontrollable increase in number/size of data centres), data traffic, and software components, applications, and systems governing our society.

Industry is slowly acknowledging this lack of competencies, and starts expressing the need for higher education to provide future employees with specialities in various areas related to green IT

[7]. In this respect, we identify two types of needs: *competencies around sustainability* in broad terms and involving multiple disciplines/aspects of an organization (e.g. economically sustainable business processes, sustainable life cycle management); and *competencies around energy efficient IT* (e.g. energy-aware software engineering and resource management, energy efficient computing resources).

The survey presented in [6] surveyed higher education initiatives providing either/both such competencies within green IT (i.e. Greening by IT and Greening of IT, respectively). It identified initiatives in a total of 19 universities, 10 in Europe and 9 outside Europe (5 USA, 3 Australia, 1 Canada). Most consist in individual modules that either focus on green IT specific subjects (e.g. a module on Sustainability) or address green IT related topics within a technical subject (e.g. sustainable business processes within a module on virtual organizations). Among all, only two initiatives emerge: Leeds University is the only one offering a full Master program on Sustainable Computing, and the University of Lorraine is the only one offering a module related to Green Software.

Next to our Master track in SEGIT, another related education program is the professional certificate in Green IT (15 ECTS) offered by the Lucerne University of Applied Sciences and Arts, Switzerland, and specializing ICT professionals in competencies especially relevant to the ICT industrial sector, like energy management, certification, and data centre management.

Taking an orthogonal perspective, the work of Mann et al. [5] defined a framework meant for educators to design modules/programs addressing sustainability. As reported in the Introduction, it also classifies sustainability-focused education approaches in three types (centralized, distributed, and blended). Finally, the work in [11] studied how sustainability can be introduced in pre-existing modules in a variety of programs (among which Computer Sciences).

To summarize, the program we present in this paper seems to provide one among very few programs addressing green IT or sustainability issues, and to our knowledge the only one both adopting a blended approach for its design, and specifically focusing on green software engineering. The next Sections describe the program and discuss the blended characteristics of its modules.

3. Program Organization

The Master in Computer Science (mCS) at the VU University Amsterdam is a 2-years Master program yielding a total of 120 ECTS³. It offers students with six specialization tracks structured as: a “Master core” (48 ECTS) common to all tracks and including the Master project, module Literature Study and Seminar, and module History of digital cultures; a “Track core” (30 ECTS) of five compulsory modules characterizing the related track; and a set of constrained choice modules and electives that provide necessary background knowledge and advanced subjects, respectively. Among the six specialization tracks, from academic year 2014/2015 the track Software Engineering has been renamed into “Software Engineering and Green IT” and fully revisited to address sustainability aspects of ICT and in particular software energy efficiency. The schedule is organized in such a way that students attend a mix of all module types over the first three semesters, followed by the Master project in the fourth and last semester.

³ www.vu.nl/nl/studiegids/2014-2015/master/c-d/computer-science/

Period / Type	Software Engineering	Green IT (focus on ...)	
1 (core)	Service oriented design	← Project on energy-aware services	Advisory Board
1 (core SE)	Software asset management	← Investments	
2 (core)	Software architecture	← Concerns related to Green IT	
2 (core)	Distributed systems		
3 (elective)	Developing services for the cloud	← Energy-aware Blue Cloud (IBM)	
4 (core)	Software metrics	← Measure EE & software qualities	
4 (elective)	Software testing		
5 (elective)	Industrial internship	← Companies involved in Green IT	
5 (core G)	Green Lab	← Generalize Green IT experience	
6 (elective)	Literature study and seminar	← On Green IT topics	

Figure 1: Overview Program-specific Modules

Extending the mCS track in Software Engineering and Green IT (SEGIT) has been the result of experience built up by the author over the past 3 years in few modules combined in an incremental implicit shift in focus from traditional software engineering (focused on the socio-technical competencies required to develop large and complex software systems) to the role played by software in our society, and its impact on all four dimensions of sustainability – technical, social, economic, and environmental [3][10]. The result is illustrated in Figure 1, where we show (see first column) the modules that make up the Track core as well as advanced modules that can be selected by the students as electives.

In particular, there are 10 modules (6 ECTS each) that provide the skills and competencies specific to SEGIT. We adopted a blended approach [5] for two modules, Software metrics and Green Lab (written using a green font in the Figure) that have been developed to *provide competencies crucial for green software*. All other modules (written using a black font) were pre-existing but have been revisited to *create awareness* in the students about the role played by software engineering with respect to sustainability (and in particular energy efficiency), according to a distributed approach.

With this education offer, the program aims at targeting the professional ambition of the students to acquire one among two types of specializations:

- A “traditional” Software Engineering specialization is possible even though blended with the awareness about the role and impact of green IT and sustainability in the profession.
- A “green” Software Engineering specialization is introduced with the new modules and with courses that (if this specialization is chosen) will have a specific “green IT” focus.

Such Green IT focus is summarized in Figure 1 by the third column. Further, the last three modules in the Figure (with a green banner as background) offer the students with specific practical training: in collaboration with industry, the students specializing in green software engineering will deepen their theoretical background in green IT (in module Literature study and seminar), will put theory

into practice (in module Industrial internship) by carrying out a project at one of our industrial partners active in the field of Green IT, and finally generalize the acquired industrial experience (in module Green Lab) by experimenting with engineering energy-aware software-intensive systems, measuring and/or monitoring their energy consumption, and learning the energy impact of different software engineering practices and design decisions.

Industrial relevance and this attention to practice is emphasized by the introduction of an industrial Advisory Board (see right-hand side of Figure 1), informally already active since years and that will include key companies in the Green IT market, interested to provide research questions from an industrial perspective as well as (Master thesis) internship possibilities, guest lectures, and projects within various modules. We envisage the role of the Advisory Board especially useful for modules Green Lab and Industrial internship.

In our vision, the program presented in this paper will open up for future potential inter-disciplinary and multi-disciplinary education: sustainability can be defined along four dimensions, these being environmental-, technical-, social-, and economic sustainability; while software engineering would address technical- and environmental sustainability more centrally (e.g. system qualities like reliability and energy efficiency, respectively), both economic sustainability (e.g. returns on the investments in green software) and social sustainability (e.g. software supporting/influencing more sustainable social behaviors) are very relevant, and have natural links with topics like lifestyle informatics, big data, healthcare, smart cities, transportation (just to mention few).

4. The Building Blocks

This Section synthesizes and explains the type of modules we designed in our program: for each module type, Table 1 illustrates the design characteristics and objectives related to green IT, the potential risks and the way we planned to mitigate them. The specific modules of our program belonging to a certain type are enlisted in the second column. In our vision, these module types can provide some inspiration to educators in pursuing a similar endeavour, and ultimately represents the building blocks underlying an innovative program such as ours.

We identified four model types:

- *Project-based* and *assignment-based modules* both bring in the module active participation and involvement of industry/practice, but at different scales and complexities.
- *Centralized modules* are especially developed to provide skills and competencies on green software engineering, and as such demanded for special attention on how risks around the relative immaturity of the field can be turned into an advantage.
- *Ad-hoc modules* include pre-existing modules that did not undergo specific changes but in the focus, which can be topics within “traditional” or “green” software engineering (depending on the specialization followed by the student).

5. Conclusion

In this paper we present the design of our CS Master specialization track in “Software Engineering and Green IT, and introduce four types of modules (project-based, assignment-based, centralized and ad-hoc) that we defined to design our program: for each type we illustrate the underlying rationale, learning objectives, risks and related mitigations, with the idea to offer them as building blocks for other educators.

Module type	Program-specific modules	Design characteristics	Objectives ⁴	Risks	Risk mitigation
Project-based	Service oriented design, Software architecture, Developing services for the cloud,	<ol style="list-style-type: none"> 1. A <i>practical project</i> is carried out in teamwork to put theory into practice. 2. An <i>industrial partner</i> proposes the project and acts as customer / project owner. 3. The <i>project is common</i> for all attending students. 4. The project entails module-specific <i>technical topics</i> augmented with relevant <i>green IT topics</i>. 	<ol style="list-style-type: none"> 1. Experience <i>hands on</i> the impact of green IT in practice. 2. Experience the innovative character of green IT topics in industrial practice; create awareness and motivate the students. 3. – 4. The students <i>not</i> specializing in Green Software Engineering become aware of the impact of software and IT on the relevant sustainability issues. 	Not all educators have (sufficient) expertise in green IT.	Involvement of industry specialized in green IT topics for the company's business provides experts that help educators understand the implications for software engineering. In addition, the involvement of practitioners motivates further the students by illustrating the relevance of the program for their future carrier.
Assignment-based	Software asset management	<p><i>(Design characteristics resemble those of Project-based modules, with the difference that assignments are of smaller size and complexity than a project)</i></p> <ol style="list-style-type: none"> 1. A subset of the assignments address/cover topics related to green IT. 	<ol style="list-style-type: none"> 1. (Same as objective #4 in Project-based modules) 	(Same as above)	(Same as above)
Centralized	Software metrics, Green Lab	<ol style="list-style-type: none"> 1. The module is design to explicitly address the needs of practice in relevant green IT topics. E.g. module Software metrics allows building sound metrics as well as measure the impact of software on sustainability-related aspects. 2. The module builds upon establish traditional software engineering theory (e.g. metrics of software systems' properties like complexity or performance, or theories for empirical experimentation) and adds novel theory specific for green software engineering (e.g. energy efficiency metrics, or experimentation practices especially conceived for green software) 	<ol style="list-style-type: none"> 1. Acquire new competencies especially needed in practice to introduce sustainable innovation. 2. Extend software engineering competencies with green IT specific ones, so that new professional profiles are created for the relevant industry sectors. 	Education material (specific for green software engineering) is still subject to ongoing, pioneering research, hence immature and subject to continuous revision.	<ul style="list-style-type: none"> • By following the education philosophy of <i>Community of Learners</i> [1], our students are actively involved in research. This way this risk is turned into opportunity, and research challenges for our students. • Also, students can contribute to this emerging research with more and larger scale experimental results; attract further interest from industry. • Modules of this type are especially designed for

⁴ Here we highlight only the objectives that are specific for Green Software Engineering, or that have a specific relevance for Green Software Engineering.

					active student research involvement.
Ad-hoc	Industrial internship, Literature study and seminar	<ol style="list-style-type: none"> 1. The module is the same for students specializing in either traditional- or green software engineering; the difference is in the selected subject. 2. The students acquire <i>advanced knowledge</i> in specific SEGIT topics of interest. 3. The students acquire <i>hands-on experience</i> in industry. 	<ol style="list-style-type: none"> 1. – 2. Feed the personal passion for a SEGIT topic; shape an own professional profile in related to Green Software Engineering. 3. Learn the state of the practice in organizations that are active in the field of green IT or that want to introduce innovation towards some sustainability objectives. 	(These modules are long established, no specific risks are identified)	–

Table 1: Analysis of underlying program design

While some modules have been informally revisited over the years to focus on green IT topics, the track will start officially in academic year 2014/2015. Key organizations from various industry sectors already expressed eager interest to sponsor the program and participate in various related practical education activities.

While our past research investments have been dedicated to create a sound scientific basis of theories and knowledge to transfer to our students (e.g. [2][4][9]), we expect industrial involvement to further accelerate knowledge creation and learning, especially needed in such a novel higher education program. In our future work we will monitor the development of competencies and the dissemination and adoption of knowledge from and to practice.

Acknowledgments

This work received partial funding from the European Fund for Regional Development under project MRA Cluster Green Software.

References

[1] A. L. Brown and J. C. Campione. “Guided discovery in a community of learners”, The MIT Press, 1994.

[2] Gu, Q., Lago, P., and Bozzelli, P. A systematic literature review of green software metrics. ACM Comput. Surv. Under submission. Preliminary version as Technical Report (2013), 31.

[3] P. Lago, S. A. Kocak, I. Crnkovic, H. Femmer, H. A. Muller, and B. Penzenstadler, “Framing sustainability as quality property of green software,” Submitted to Communications of the ACM, 2014, under review.

[4] G. A. Lewis, P. Lago, G. Procaccianti, “Architecture Strategies for Cyber-Foraging: Preliminary Results from a Systematic Literature Review”, in Proceedings of the 8th European conference on Software architecture (Vienna, Austria, 2014), ECSA’14, Springer-Verlag, 16.

[5] S. Mann, L. Muller, J. Davis, C. Roda and A. Young, “Computing and sustainability: evaluating resources for educators”. ACM SIGCSE Bulletin, 2009. Vol: 41 No: 4: p.144-155.

[6] B. Merkus, “SEFLab SURF Innovatieregeling 2013”, Technical Report Amsterdam University of Applied Sciences, 20 Nov. 2013 (in Dutch).

[7] W. Faaij, H. Radstake, M. Hendriks, G. van Rije, and T. Struijk, “Wat verwachten werkgevers van het onderwijs als het gaat om duurzaamheid?”. Technical Report Groene Generatie, 24 Oct. 2013.

- [8] S. Murugesan, "Harnessing green IT: Principles and practices", in *IT Professional, IEEE*, 10 (2008), 24–33.
- [9] G. Procaccianti, P. Lago, and G. Lewis, "Green architectural tactics for the cloud," in *Proceedings of the 11th Working IEEE/IFIP Conference on Software Architecture (WICSA)*, Apr. 2014.
- [10] M. Razavian, G. Procaccianti, D.A. Tamburri, "Four-Dimensional Sustainable E-Services". *EnviroInfo*, Sep. 2014.
- [11] K. Sammalisto and T. Lindhqvist, "Integration of sustainability in higher education: A study with international perspectives", in *Innovative Higher Education*, Elsevier, Vol. 32(4), 2008, 221–233.