

Master Internship — 5 to 6 months, starting between Feb. and Apr. 2023

ENACT - ENergy efficiency through ArChitectural Tactics for the Internet of Things

Denisse Muñante (1) and Sophie Chabridon (2)

(1) SAMOVAR Lab, ENSIIE

(2) SAMOVAR Lab, Télécom SudParis, Institut Polytechnique de Paris
Évry, France

Contacts: denisse.munantearzapalo [at] ensiie.fr, Sophie.Chabridon [at] telecom-sudparis.eu

Keywords Energy-efficiency, Software engineering, Internet of Things, Distributed systems, Middleware.

Context. Energy efficiency has already been considered for many years at the hardware level. However, powerful and cheaper computing resources have led to less resource optimization in software. Considering the planet resource limits and the increasing role of software in the Internet of things, there is an urgent need to design software with energy-efficiency as a requirement. In practice, this means to consider energy-efficiency in software quality attributes at design time and then to implement architectural tactics [7] enforcing them. Moreover, usage conditions varying a lot at runtime, dynamic reconfiguration would enable additional energy savings.

Calero and Piattini explore the environmental dimension of software engineering and define Green IT as the sustainable consumption of natural resources by hardware and software components of ICT [1]. Green IT distinguishes between "Green by IT" and "Green in IT". "Green by IT" considers ICT as a means to optimize the consumption of natural resources (e.g. Smart Home,...), while "Green in IT" intends to lower the impact of ICT on natural resources. Consequently, Green in Software Engineering corresponds to the subpart of "Green in IT" which is focusing on methodologies and mechanisms to limit the environmental impact of software products.

Internship objectives. In the context of this internship, a **first objective** is to take into account environmental sustainability quality attributes, *e.g.*, saving energy consumption or keeping resources consumption low, in the design of software products and to provide recommendations to software engineers for developing software with low environmental impact. In addition to design time, such quality attributes should also be considered at runtime when usage conditions are established. A **second objective** is to explore dynamic reconfiguration solutions able to monitor energy consumption and the current execution context with the purpose of detecting and alleviating undesired scenarios, *e.g.*, rebound effects, at runtime.

Concretely the tasks that will be carried by the selected candidate:

1. Study the state-of-the-art to collect the metrics [3] used to assess green quality attributes for software products, and to explore the current green architectural tactics used in software engineering life-cycle [6, 2, 4, 8, 7].
2. Artefacts used at design-time: conceive a solution that supports architects and developers to include green quality attributes in the software development life-cycle, and select software architectures by recommending the architectures that respect the green quality attributes:
3. Artefacts used at runtime: conceive a prototype that allows to explore and select "appropriate" architectures configurations when usage conditions are established and differs from the expected ones, *e.g.*, the actual data flow frequency is not as was expected or a resource state (longevity) make system risky.
 - the solution could be formulated as a multi-criteria problem for the self-reconfiguration decision making process.
 - explore the state-of-the-art methods to solve the multi-criteria decision-making problem for self-reconfiguration, for instance a optimisation algorithm that explores solutions taking into account quality attributes is introduced in [5].
 - implement a prototype to simulate the self-reconfiguration when green quality attributes become deprecated or are violated.

This subject is part of the research works of the [DiSSEM](#) group concerning Distributed Systems, Software Engineering and Middleware, in the [ACMES](#) team of the [SAMOVAR](#) lab.

References

- [1] Coral Calero and Mario Piattini, editors. *Green in Software Engineering*. Springer, 2015.
- [2] Abdessalam Elhabbash, Maria Salama, Rami Bahsoon, and Peter Tiño. Self-awareness in software engineering: A systematic literature review. *ACM Trans. Auton. Adapt. Syst.*, 14(2):5:1–5:42, 2019.
- [3] Shokhista Ergasheva, I Khomyakov, A Kruglov, and G Succil. Metrics of energy consumption in software systems: a systematic literature review. *IOP Conference Series: Earth and Environmental Science*, 431(1):012051, 2020.
- [4] José Miguel Horcas, Mónica Pinto, and Lidia Fuentes. Context-aware energy-efficient applications for cyber-physical systems. *Ad Hoc Networks*, 82:15–30, 2019.
- [5] Fitsum Meshesha Kifetew, Denisse Muñante, Jesús Gorroñoigoitia, Alberto Siena, Angelo Susi, and Anna Perini. Grammar based genetic programming for software configuration problem. In Tim Menzies and Justyna Petke, editors, *Search Based Software Engineering - 9th International Symposium, SSBSE 2017, Paderborn, Germany, September 9-11, 2017, Proceedings*, volume 10452 of *Lecture Notes in Computer Science*, pages 130–136. Springer, 2017.
- [6] Adel Nouredine and Ajitha Rajan. Optimising energy consumption of design patterns. In Antonia Bertolino, Gerardo Canfora, and Sebastian G. Elbaum, editors, *37th IEEE/ACM International Conference on Software Engineering, ICSE 2015, Florence, Italy, May 16-24, 2015, Volume 2*, pages 623–626. IEEE Computer Society, 2015.
- [7] Carlos V. Paradis, Rick Kazman, and Damian Andrew Tamburri. Architectural tactics for energy efficiency: Review of the literature and research roadmap. In *54th Hawaii International Conference on System Sciences, HICSS*, pages 1–10. ScholarSpace, 2021.
- [8] Theresia Ratih Dewi Saputri and Seok-Won Lee. Integrated framework for incorporating sustainability design in software engineering life-cycle: An empirical study. *Inf. Softw. Technol.*, 129:106407, 2021.