Sustainable Computing: Informatics and Systems 30 (2021) 100558

Contents lists available at ScienceDirect

Sustainable Computing: Informatics and Systems

journal homepage: www.elsevier.com/locate/suscom

FEETINGS: Framework for Energy Efficiency Testing to Improve Environmental Goal of the Software

Javier Mancebo *, Coral Calero, Felix Garcia, Mª Angeles Moraga, Ignacio Garcia-Rodriguez de Guzman

Alarcos Research Group, Institute of Techonology and Information Systems, University of Castilla-La Mancha, Ciudad Real, Spain

ARTICLE INFO

Keywords:
Software sustainability
Green software
Energy efficiency
Energy consumption

ABSTRACT

Software is a fundamental part of today's society. However, both users and software professionals need to be aware that its use impacts on the environment, due to the high energy consumption it entails. One of the main gaps to be faced is the difficulty of analyzing software energy consumption in the endeavor to know whether a particular software product is as much energetically efficient as possible, or at least more efficient than another, and to improve the environmental objectives of the software. For this reason, a Framework for Energy Efficiency Testing to Improve eNviromental Goals of the Software (FEETINGS) is presented in this paper. FEETINGS is composed of three main components: an ontology to provide precise definitions and harmonize the terminology related to software energy measurement; a process to guide researchers in carrying out the energy consumption measurements of the software, and a technological environment, which allows the capture, analysis and interpretation of software energy consumption data. This paper also presents an example of the application of the FEETINGS, which aims to raise awareness of the energy consumed by the software in activities that we perform daily, such as writing a tweet or a Facebook post. As a result, we have been able to verify that FEETINGS allows us to carry out an analysis and measurement of software energy consumption to provide users with good practices, as using an emoji or a picture rather than a GIF. © 2001 Elsevier Science. All rights reserved.

1. Introduction

In recent years, Information and Communication Technologies (ICT) are increasing very fast with the aim to make people's lives easier. In fact, the number of users of software and websites, such as social networking, Google searches, or video viewing are growing every day in the world [1].

Due to this intensive use of software, ICT energy consumption had already increased by 2018 to 1895 TW h, representing about 9% of total global energy consumption [2]. And by 2025, this could exceed 20 % of total energy and emit up to 5.5 % of the world's carbon emissions [3,4].

With these figures in mind, it is increasingly important to raise awareness among software users and developers of the ecological impact of the software applications we use daily.

Therefore, trends such as Green software have gained importance in recent years [5,6]. The purpose of the Green Software [7] is to promote the improvement of the energy efficiency of software, minimizing the impact it may have on the environment.

In order to assess whether a software product is sustainable and environmentally friendly, it is necessary to know the energy consumption that is induced by the software when it is running [8,9]. As the European Union report indicates [10], "the existence of a methodology for measuring the energy or CO_2 of the ICT infrastructure is extremely important for this sector, as it will allow the development of much more robust estimates of the impact of ICT". This information can be used to develop a plan for reducing energy consumption and improving the sustainability of the software [11].

The growing importance of Green Software and the need to know its energy efficiency, has resulted in several studies in the current literature that evaluate the software's energy consumption [6,8,12–16]. When this type of study is analyzed three type of problems arise:

- Several inconsistencies and terminological conflicts appear. That is because researchers have defined their methods of work using their

* Corresponding author.
E-mail addresses: Javier.Mancebo@uclm.es (J. Mancebo), Coral.Calero@uclm.es (C. Calero), Felix.Garcia@uclm.es (F. Garcia), MariaAngeles.Moraga@uclm.es (M.A. Moraga), Ignacio.GRodriguez@uclm.es (I. Garcia-Rodriguez de Guzman).

https://doi.org/10.1016/j.suscom.2021.100558
Received 9 June 2020; Received in revised form 1 March 2021; Accepted 4 April 2021
Available online 6 April 2021
2210-5379/

own terms or concepts, provoking a lot of examples of synonymy (same concepts with different term associated) and homonymy (different concepts with the same name). This lack of formal consensus makes it difficult to understand the main concepts involved when performing a software energy consumption assessment.
- There is a lack of a generally-agreed-on methodology that would guide software energy consumption assessments. This implies that the rigor of the studies carried out cannot be guaranteed, and that it is more complicated to replicate or compare the results obtained.
- Several measuring instruments are available for the analysis of software energy consumption. It is important to know that each measurement instrument has its own particular characteristics, and it is necessary to choose the one that best adapts to our evaluation necessities.

In order to contribute to the mitigation of the already mentioned problems, we have developed a framework to promote the reliability of capture, analysis, and interpretation of software energy consumption data, known as FEETINGS (Framework for Energy Efficiency Testing to Improve eNvironmental Goals of the Software). FEETINGS aims to provide: (1) a solution to the lack of a unique and agreed terminology; (2) a process that helps researchers to evaluate the energy efficiency of the software, allowing greater control over the measurements made, ensuring their reliability and consistency; and (3) a technological environment that supports the process and allows for realistic measurements of the energy consumed by the software and its subsequent analysis.

The measurements of energy consumption obtained when FEETINGS is applied will be useful twofold depending on the measurements done. On one hand, software professionals can be aware of the energy that the software they develop consumes when it is used. On the other hand, end users can be aware of the energy needed by the software they use.

In this paper, we present FEETINGS and an example of how to use it for end users awareness. We organize the content of the paper as follows: first, in section 2 we present the FEETINGS framework, detailing each of the components. In section 3, an example of the application of FEETINGS is presented. In section 4, we present the different studies and proposals that have served as a basis for our framework and the main contributions of FEETINGS. Finally, section 5 sets out the conclusions of this work and presents some lines of future work.

2. FEETINGS

In this section, we will describe FEETINGS, a framework to promote more reliable capture and analysis of software energy consumption data. This framework is made up of three main components, which are classified according to their nature in conceptual, methodological, and technological components, as shown in Fig. 1, which are described in the following subsections.

2.1. Conceptual component

The conceptual part of FEETINGS seeks to solve the lack of a unique and agreed terminology. For this purpose, an ontology has been elaborated which contains the concepts related to the software energy measurement. This ontology is known as Green Software Measurement Ontology (GSMO), and its purpose is to provide precise definitions of all terms and to clarify the relationships between them, removing terminological conflicts and fostering the consistent application of the framework by other researchers and practitioners with reference to a common vocabulary, related to software energy measurement.

This GSMO ontology is an extension for green software measurement of the Software Measurement Ontology (SMO) proposed by Garcia et al. [17].

To define GSMO, we have chosen REFSENO (Formal Representation for Software Engineering Ontologies) [18], which was designed explicitly for software engineering and which allows several representations for software engineering knowledge. REFSENO provides constructions to describe concepts, attributes, and relationships; these are used to represent: a table with the glossary of concepts, a table of attributes, and a table with the relationships. To simplify this explanation, we have omitted the description of the attributes of the GSMO concepts in this case.

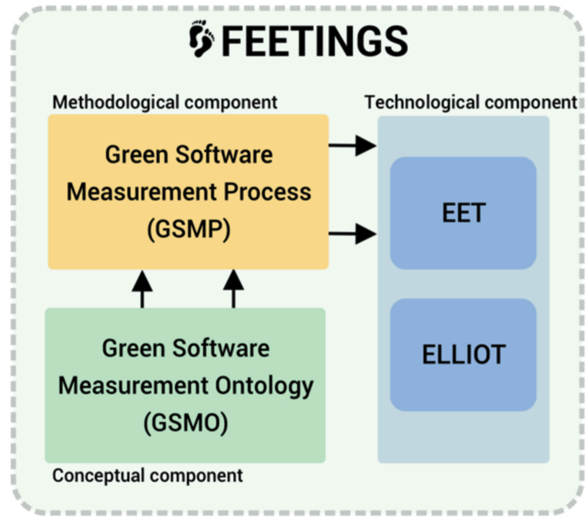

Fig. 1. Overview of FEETINGS.

In Fig. 2, the graphical representation of the terms and relationships of the GSMO is shown, using the UML (Unified Modeling Language).

The highlighted concepts are the new concepts which extend/adapt the SMO [17] for the domain of green software.

These concepts are defined in Table 1, by including the term, its description and source. The values for the fourth column can be: *Adapted*: when the definition is based on a cited document but changed to achieve the goals of this ontology; *New*: if the term used in the ontology has a new meaning in this ontology.

Table 2 shows the relationships between the terms defined above. These tables include the following columns: name of the relationship, participating concepts, and description of the relationships between the concepts.

The concepts and relationships that appear in Fig. 2, and are not defined in Tables 1 and 2, which have been extracted from the SMO proposed by Garcia et al. [17].

The conceptual component (GSMO ontology) aims to solve the problem of terminology consistency in software energy measurement, since it proposes a common vocabulary extracted from several international standards and research proposals. This ontology has, moreover, served as a basis for the development of the methodological component of FEETINGS, which is presented in the following subsection (Table 3).

2.2. Methodological component

The methodological part focuses on providing support for the activities and roles required to analyze the energy efficiency of the software. A process has thus been developed to guide researchers in carrying out the software's energy consumption measurements, from study design to analysis and reporting of results. This process is known as the Green Software Measurement Process (GSMP) and it ensures greater control over the measurements made, improving the reliability, consistency, and coherence of the measurements. It also ensures that the results obtained are comparable with other studies and facilitates the replicability of the analyses performed.

To define the GSMP, we have followed the method engineering

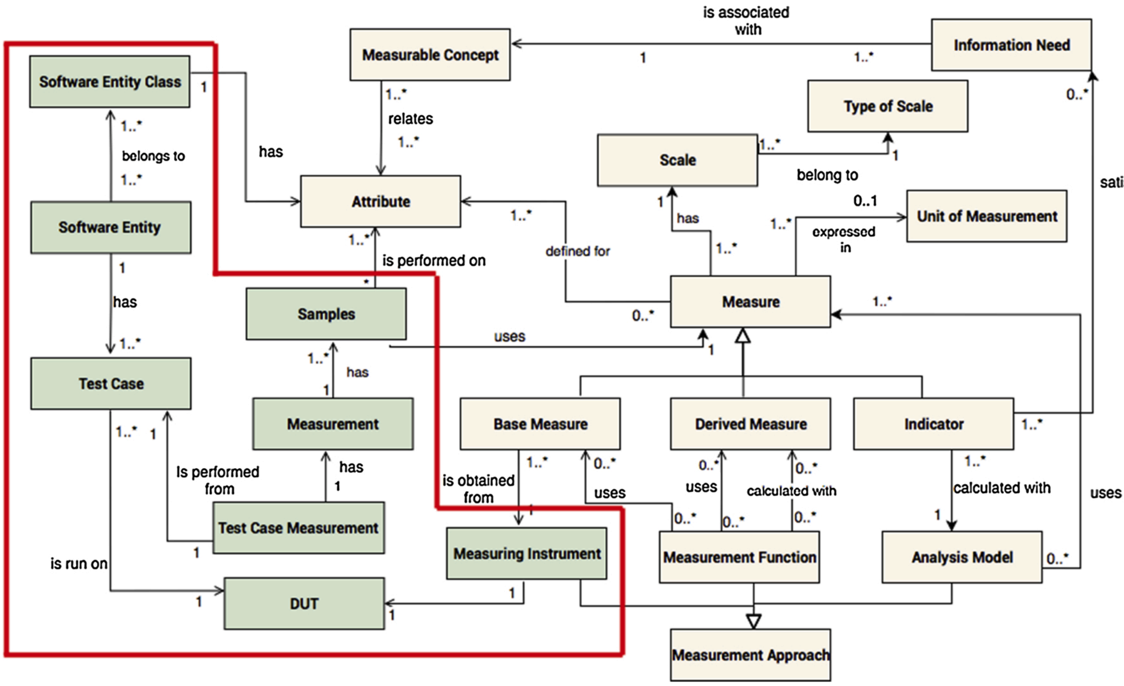

Fig. 2. UML diagram of the Green Software Measurement Ontology (GSMO).

Table 1
Definition of the terms in the GSMO.

Term	Definition	Source
Software entity	Software that is to be characterized by measuring its attributes.	Adapted from SMO
Software entity class	The collection of all the entities that satisfy the determined objective.	Adapted from SMO
Test Case	A representation of the functionality of the software entity to be measured.	New
Test Case Measurement	A set of energy consumption measurements of all the runs in a test case.	New
Measurement	A set of energy consumption samples from a single test case run.	Adapted from VIM
Samples	Each energy consumption record taken by a measuring instrument.	New
Device Under Test (DUT)	A device where the software entity to be measured is run.	New
Measuring Instrument	A method used to make energy consumption measurements.	Adapted from VIM

approach [19], and we also have taken as our basis well-known approaches to software measurement and good practices related to green software that have been proposed by other authors.

GSMP is composed of seven phases, as shown in Fig. 3, which are described below:

- *Phase I. Scope Definition:* in this phase, a complete specification of requirements for the evaluation of energy efficiency is obtained. In addition, the software subject of the study and the test cases to be analyzed must be defined.
- *Phase II. Measurement Environment Settings:* the purpose of this phase is the definition of the measurement environment that will be used in the software energy consumption assessment. As a result of this phase, the measuring instrument to be used and its measurements, the specifications of the computer where the software is to be executed (Device Under Test, DUT) are defined and the baseline energy consumption of the DUT is obtained.

Table 2
Definition of the relationship in the GSMO.

Name	Concepts	Definition
Has	Software Entity class – Attribute	A software entity class has one or more attributes. An attribute can belong to only one software entity class.
Belongs to	Software Entity – Software Entity Class	A software entity belongs to one or more entity classes. A software entity class may characterize several software entities.
Has	Software Entity – Test Case	A software entity has one or more test cases. A test case can belong to only one software entity.
Is performed from	Test Case Measurement –Test Case	Every test case measurement is performed from a test case of a software entity.
Is run on	Test Case – DUT	Every test case is run on a DUT. A DUT can run several test cases.
Has	Test Case Measurement –Measurement	A test case measurement has one or more measurements. A measurement can only belong to one test case measurement.
Has	Measurement –Samples	A measurement has one or more samples. A sample can only belong to one measurement.
Is performed from	Sample – Attribute	Every sample is performed on one or more attribute of an entity.
Is connected to	Measuring instrument – DUT	A measuring instrument is connected to a DUT where test cases are run.
Is obtained from	Base Measure – Measuring instrument	Every base measure is obtained from a measuring instrument.

Table 3
DUT specifications.

Operating systems	Windows 10 Pro
Hardware specification	Motherboard: Asus M2N-SLI Delux CPU: AMD Athom 64 × 2 5600 + 2,8GHz RAM: 4 × 1 G3 665 Hz Kingston HDD: Seagate barracuda 720 rpm 500Gb GPU: Nvidia Gforce 8600 GTS

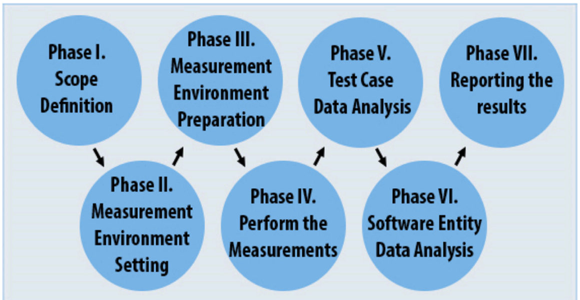

Fig. 3. Green Software Measurement Process.

- *Phase III. Measurement Environment Preparation:* focuses on the preparation of the energy consumption measurements to be performed and on the configuration of the measurement environment.
- *Phase IV. Perform the Measurements:* During this phase, energy consumption measurements are carried out and raw energy consumption data taken from the measuring instrument is collected.
- *Phase V. Test Case Data Analysis*: The processing of the raw data of energy consumption obtained by the measuring instrument. This data transformation process is known as Data Wrangling and the most important tasks to be performed are: reformatting and integrating data from different sources so that they can be analyzed correctly, the identification of possible outliers that may be present in the samples of the measurements, we recommend the use of robust parametric methods such as the median of the absolute deviations from the median (MADN), and check that each of the measurements performed is correct. To find unusual measurements, you can use the interquartile range method (IQR). and the statistical analysis of the values obtained from the measurements of the defined test cases are carried out.

Once the data is processed, the statistical analysis of the values obtained from the measurements of the defined test cases is performed. To carry out the analysis, the descriptive statistics for each test case analyzed need to be calculated. To obtain the most complete information available on energy consumption, we suggest the calculation of the following descriptive statistics: on the one hand, standard descriptive statistics (maximum and minimum value, range, mean, standard deviation, variance or interquartile range), and on the other hand, the robust descriptive statistics such as median, trimmed mean, winsorized mean or median absolute deviation. It is not compulsory to calculate all the descriptive statistics mentioned.

- *Phase VI. Software Entity Data Analysis:* in this phase, with the previous analysis, it is determined and interpreted how much energy was consumed when the software entity was executed in the DUT and some conclusions about the software energy consumption are stated.
- *Phase VII. Reporting the Results:* finally, the study carried out is documented, describing the entire process followed, together with the results on the energy consumption of the software that had been extracted.

2.3. Technological component

The technological component, as can be seen in Fig. 4, is composed of two artifacts.

On the one hand, EET (Energy Efficiency Tester), a measuring instrument, which follows a hardware-based approach and that is responsible for obtaining the software's energy consumption measurements when it is running. On the other hand, ELLIOT is a software tool in charge of processing the data collected by EET, analyzing these data, and generating an appropriate visualization of the results.

2.3.1. EET (energy efficiency tester)

EET is a measuring instrument that follows the hardware-based approach [20,21]. This measuring instrument allows the energy consumption of a set of hardware components used by the software to be captured accurately during its execution.

In addition to the total energy consumption of the DUT, where the software is running, EET supports the measurement of four different hardware components: processor, hard disk, graphic card, and monitor. Fig. 5 shows the design of the EET architecture.

As can be seen in Fig. 5, EET is connected to the DUT where the software is executed, and is composed of three main components:

- A system microcontroller, whose task is to gather the information extracted from the different sensors and store them in a MicroSD memory. It also allows the frequency with which the device performs the measurements to be adjusted.
- A set of sensors, which are responsible for taking energy consumption measurements of the hardware components (processor, hard disk, graphics card, and monitor) of the DUT connected to EET.
- A power supply, that must be connected to the device under test where the software is executed, replacing the power supply of the DUT; the sensors are connected to the energy distribution lines from the power supply to the different hardware components.

As EET produces a huge amount of data about energy consumption, it is necessary to support the processing and analysis of these data by a suitable software tool. For this reason, the ELLIOT tool was developed, which is described in the following section.

2.3.2. ELLIOT

The main objective of the ELLIOT software tool is to provide a visual environment that allows researchers to process the data collected by EET, analyze them, and generate an appropriate visualization of the results obtained. Furthermore, the ELLIOT tool is aligned with the GSMP process described in the previous section.

The technologies used to develop ELLIOT are Java for the logical layer, Java Swing and the JFreeChar library for the interface layer, and the graphics presented, and MariaDB for the data persistence layer. Thus, we can use ELLIOT on any platform to analyze energy consumption data that has been recorded on a memory card by EET.

The main functionalities supported by ELLIOT are outlined below:

- Processes all measurements carried out with the EET measuring instrument.
- Calculates different statistical variables of the energy consumption measurements according to the user's needs.
- Identifies possible outliers that may be present in the measurement samples, using robust parametric methods such as median absolute deviations from the median (MADN).
- Visualization of the results through graphs and data tables with information on the measurements of the energy consumption of the software.
- Comparison of the results obtained from different energy consumption measurements.
- Generation of reports that include all the information on the energy efficiency of the software analyzed.

The ELLIOT tool is composed of four modules that support the main functionalities, shown in Fig. 6 and described below:

- *User management.* This module allows ELLIOT administrator to manage the ELLIOT users' permission accesses depending on the role assigned to them. This module also allows users to modify their access information and data.
- *System management:* This module is used to add and modify the measuring instruments that generate the data for later analysis. It

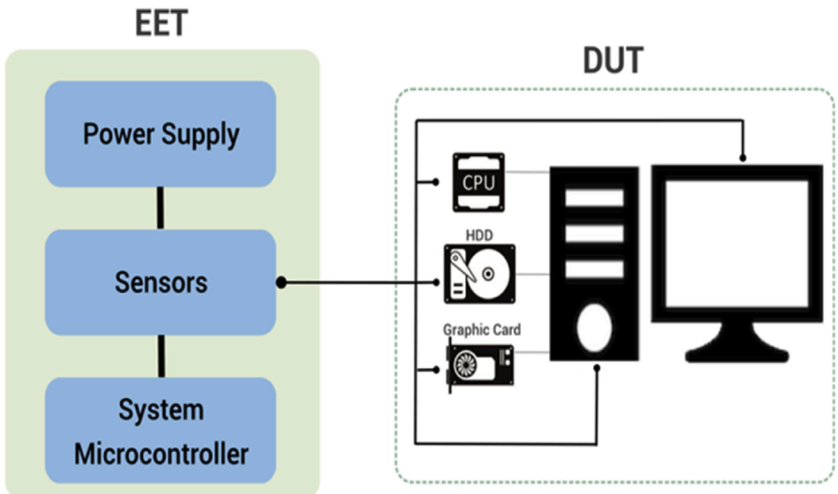

Fig. 4. Artifacts of the Technological component of FEETINGS.

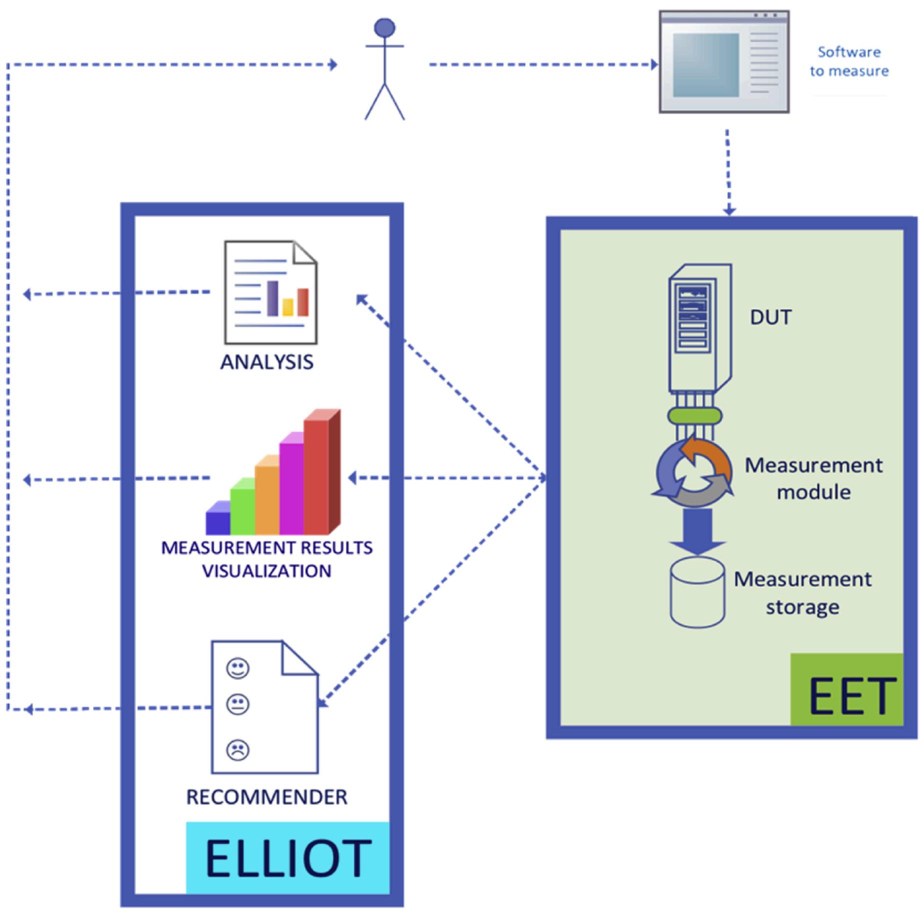

Fig. 5. Design of the EET.

also allows the configuration data of the DUT to be introduced in the tool where the software to be evaluated is running.

- *Measurement management:* This is the central module of ELLIOT, since it supports all the tasks of processing, data wrangling, measurements analysis, and visualization of the energy consumption information (see Fig. 7).
- *Report management:* This module is responsible for generating reports and making comparisons between energy consumption measurements. Moreover, a comparison of the measurements at the level of test cases or software entity is made possible.

3. Application of the FEETINGS: a case study of energy consumed by social networks

As explained in the introduction, FEETINGS can be used for several purposes. One of the main objectives of FEETINGS is to measure software energy efficiency so that researchers and software professionals

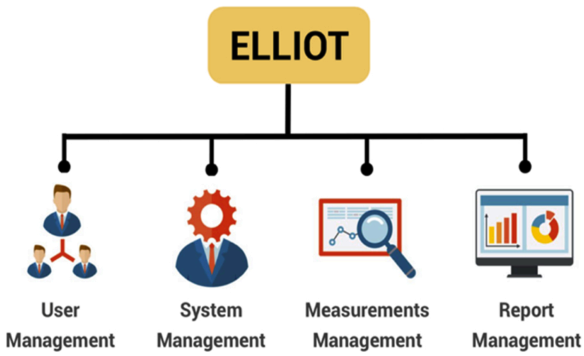

Fig. 6. ELLIOT tool modules.

can develop software that is environmentally-friendly. For this purpose, we have carried out different studies, that have allowed us to obtain information that can be useful for researchers and practitioners. Some of the most outstanding studies carried out are: an analysis of sorting algorithms [22]; a study of the interaction between the usability and the energy efficiency of a web PHR application [23]; a comparative study of data compression techniques from the perspective of the savings obtained in energy consumption [24]; and an analysis of the interaction in the software between maintainability measures (lines of code, complexity…) and energy consumption [25].

In this work, beyond the software developer perspective, we are going to focus on the other key perspective: the user's one, with the aim to show how a framework such as FEETINGS can raise the society awareness for the responsible use of software applications to take care of the environment.

Following this purpose, in this section we are going to present the result of two studies carried out, using FEETINGS, that seek to find out how tweets (Twitter) and Facebook posts should be written to reduce energy consumption and minimize the impact that they have.

Fig. 8 shows the instantiation of the concepts of this study, defined in the GSMO ontology, for this study, which also serves as a summary of the outputs obtained in the first phases of the GSMP process.

In the first phase of the GSMP, as can be seen in Fig. 8, we defined that the objective of this study was to analyze the consumption of social networks, which are the Software Entity Class *(1)*. Then, we choose the Software Entity *(2)* that will be evaluated: Twitter and Facebook. In addition, five Test Cases *(3)* were executed to measure the software entities. The test cases defined were: a tweet/post with 280 characters, a tweet/post with 1 emoji, a tweet/post with 280 emojis, a tweet/post with a picture, and a tweet/post with a GIF.

As a result of the second phase of the process, we selected EET as the measuring instrument *(4)* and defined the specification of the Device Under Test (DUT), as can be observed in Table X, where the test cases were executed *(5)*. For this study, we decided that from the measurements provided by EET, we would only take into account the energy measurements of the graphic card (GPU), to observe if the fact of using images or characters had an influence; and the total consumption of the DUT. We have not recovered and analyzed the data from the hard disk and the processor because, as both networks were executed on a web browser, the use of these two elements are minimal. Each of the samples *(6)* obtained by EET was recorded in a measurement log *(7)*.

Following the third phase of the process, we have determined that each of the test cases was run and measured (with EET) 20 times and recorded in a test case measurement log *(8)*. Being a controlled test environment, 20 measurements are usually a sufficient sample size to mitigate the impact of outliers (such as energy consumption devoted to operating system tasks).

Once the measurements and analysis have been made, Tables 4 and 5 show the power consumed in the execution of each of the test cases defined for Twitter and Facebook, respectively.

Note that in both tables, the GPU power values hardly vary, contrary to what we might think when including pictures and GIFs. However, the power consumed by the DUT does vary between test cases.

As we can observe in Fig. 9, both on Twitter and Facebook, the order of the power consumed by the DUT in the test cases is similar. In both cases, the most efficient option is to write a tweet/post with a single emoji, followed by a post with a picture. Also, in both cases, to write a publication that includes a GIF is the worst option from the point of view of energy efficiency.

In Table 6, the calculated values of the energy consumed by the DUT, when running each of the test cases in both social networks, are shown.

Analyzing the energy consumption data of the DUT on Twitter, we observe that the difference between publishing a tweet with an emoji (more efficient option) and publishing it with a GIF (more inefficient

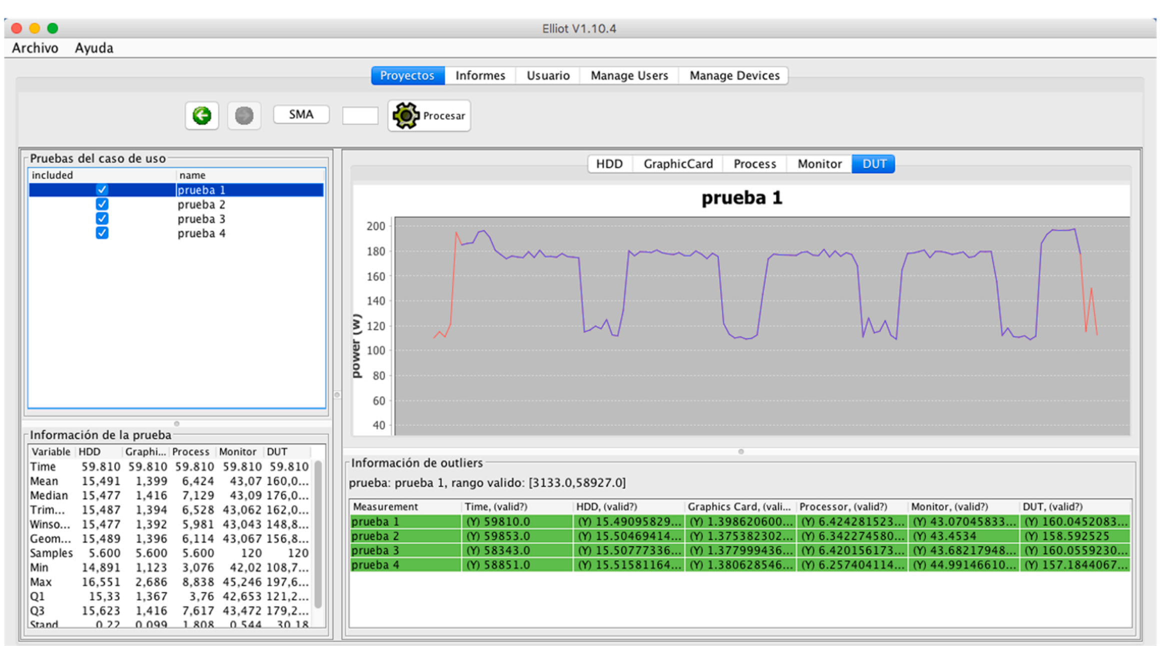

Fig. 7. ELLIOT tool interface.

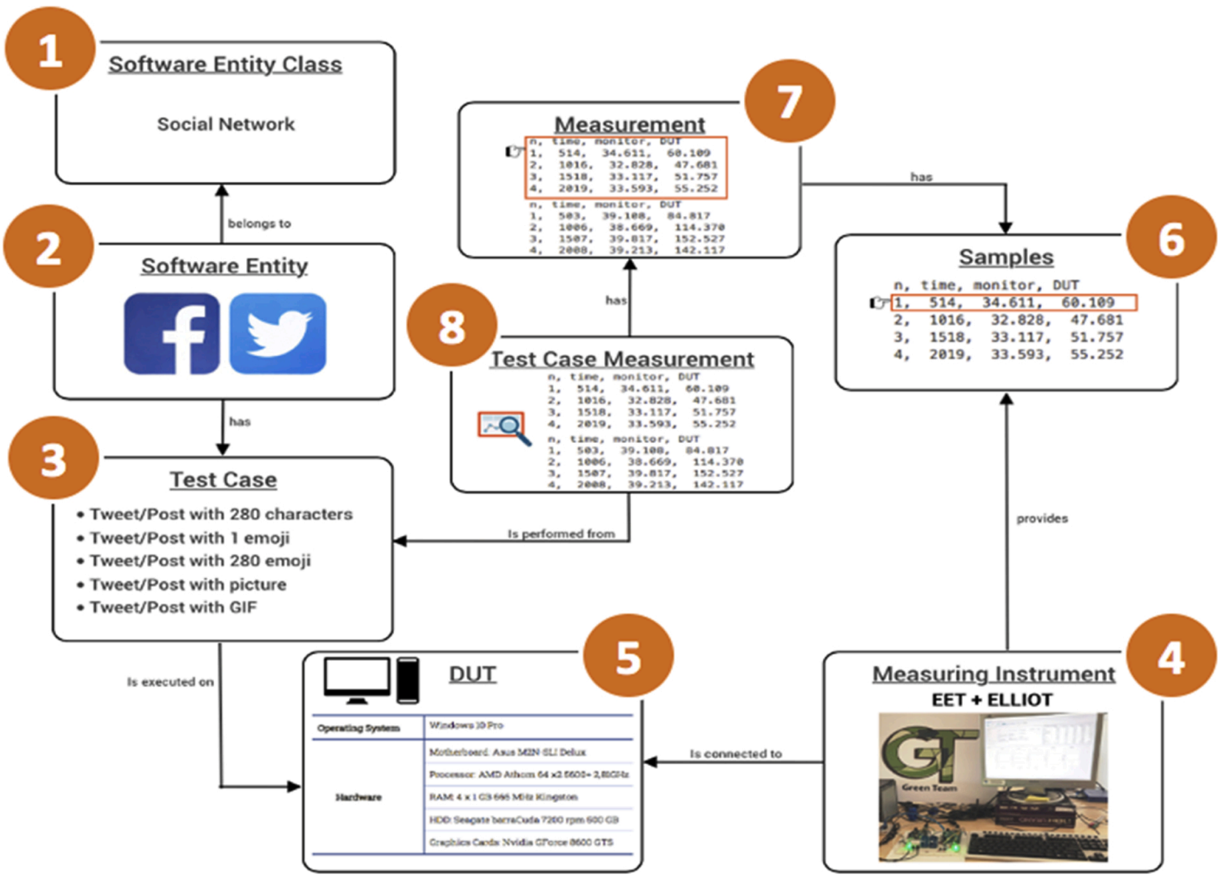

Fig. 8. GSMO instantiation.

Table 4
Power measured and execution time for each test case on Twitter.

		Power (watts)	
Test Case	Time (sec)	GPU	DUT
280 Characters	591	114	17,721
1 emoji	707	123	13,928
280 emoji	590	142	18,333
1 picture	789	118	14,353
1 GIF	678	148	19,892

Table 5
Power measured and execution time for each test case on Facebook.

		Power (watts)	
Test Case	Time (sec)	GPU	DUT
280 Characters	1194	147	8317
1 emoji	1057	136	7701
280 emoji	1090	126	8933
1 picture	1129	122	8259
1 GIF	1123	135	11,214

option) is 366,66 W per second (00,001,018 kW h). On Facebook, the difference between the two options is 434,07 W per second (00,001,205 kW h).

Considering the results obtained, the differences between the best and the worst option, from the energy point of view, seem insignificant. However, it is necessary to be aware of what these data represent. For

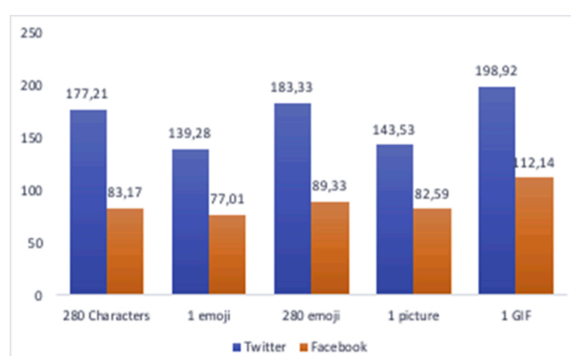

Fig. 9. Power consumed by Twitter and Facebook.

Table 6
Energy consumed for each test case on Twitter & Facebook.

	Energy Consumption (W·s)	
Test Case	Twitter	Facebook
280 Characters	104,950	98,145
1 emoji	98,574	81,420
280 emoji	108,196	99,201
1 picture	103,256	93,228
1 GIF	135,240	124,827

example, in the case of Twitter as we noted, the saving in one tweet is only 00,001,018 kW h. But we must bear in mind that in one hour, according to the information provided by the Internet Lives Stats website[1], approximately 32.400.000 tweets are published on average. If we consider that all those tweets are written with the most efficient option (with an emoji) instead of the most inefficient one (with a GIF) the savings obtained would be 329,832 kW h just in one hour. This energy saving means that 131,932 kg of CO2 are not expelled into the atmosphere. If we relativize the results obtained, with the aim of understanding the impact that our actions can have, the energy savings obtained with Twitter, in one hour, would allow us to travel more than 200.000 Kms with a Tesla 3 or light up a family home for a year.

As can be observed, small savings have a very high real impact due to the large number of users who have software applications in general and social networks.

So, the way to use these results is to recommend users that, when tweeting or posting on Facebook, they must remember that it is better to avoid the use of GIFs while emojis and characters are much better. Simple and useful clues to take into consideration when using their social networks.

4. Background and related work

This section presents a synthesis of the background and related work of this research according to: *(i)* the definition of a common terminology related to software measurement; *(ii)* the proposed software measurement methodologies and standards, including methods for measuring the energy consumed by the software; *(iii)* and the measuring instruments for software energy consumption are shown, classified according to their approach.

Furthermore, once the different existing proposals to the problems identified have been analyzed, the main contributions provided by the FEETINGS framework are presented.

4.1. Terminology related to software measurement

According to Chandrasekaran et al. [26], the unification of terms and concepts in an ontology allows knowledge to be shared, while ontological analysis clarifies the structure of knowledge.

With regard to pieces of work that present ontologies for energy measurement in software, to the best of our knowledge, there are currently none that provide terms, concepts and relationships focused on the subject of software energy measurement.

However, Garcia et al. [17] propose the Software Measurement Ontology (SMO), which aims to contribute to the harmonization of the different software measurement proposals and standards, by providing a coherent set of common concepts used in software measurement. Although the SMO ontology has a general approach, it may provide a basis to clarify the terms that can be used in any software measurement process.

4.2. Software measurement methodologies and standards

As we described in the introduction, it is important to define a process that ensures the rigor and consistency of studies using software energy consumption measurements. Different approaches that could be useful for this purpose can be found in the existing literature.

On the one hand, there are the software measurement frameworks and standards, which aim to provide guidelines for carrying out the measurement process effectively and systematically, based on the defined objectives. The following are some of the more well-known methods or standards:

- *Goal/Question/Metric (GQM),* which was proposed by Basili and Weis [27] and sets out guidelines for defining a measurement program, including information for data collection, analysis, interpretation of results and identification of possible improvements.
- *Goal Question Indicator Metric (GQ(I)M)* [28] identifies and defines software metrics that support the company's business, and improve its processes and the objectives of its projects, ensuring the relevance and traceability of the objectives with respect to the data obtained.
- *Practical Software Measurement (PSM)* [29] provides project managers and technicians with best practices and guidelines in software measurement. It is based on real experience gained from government and industry projects.
- *ISO/IEC/IEEE 15939:2017. Software Engineering–Software Measurement Process,* which identifies the activities and tasks which are required in the effort to identify successfully, define, select, apply, and improve software measurement within a generic project or the measurement organization structure.

These software measurement frameworks provide guidelines for defining and implementing software measurement programs. However, none of the methodological frameworks mentioned focuses on measuring the energy efficiency of software, so these frameworks cannot be fully adopted.

On the other hand, works that present guidelines for energy measurement in software, we can highlight the one presented by Hindle [30], where a methodology for conducting experiments using energy consumption measurements is described. This methodology is known as "Green Mining Methodology". However, this methodology is that it does not provide a protocol or good practices regarding how to carry out the measurement in a way that is valid and reliable. For this reason, Jagroep, et al. [31] present a measurement protocol, in which an extension of activity 6 of "Green Mining" is performed, detailing the specific tasks to be carried out.

After analyzing all the proposals, we consider that none of the methodologies or standards mentioned give guidance about how to carry out the entire process of measuring the energy consumption of software, from its planning through to its analysis and reporting. That is the reason why we consider it necessary to define a process for researchers to use to analyze the energy consumption of a software when it is running

4.3. Measuring instrument for software energy consumption

Managing the sustainability of software requires knowing the energy consumption of hardware when the software is running [32]. To that end, several methods have been developed that use different measuring instruments to capture software energy consumption data.

These measurement methods can be classified into two main approaches, according to their characteristics: Software-based and Hardware-based. In addition, both of these main approaches can be combined, resulting in a Hybrid approach. Each of the approaches is described below:

- *Software-based approach:* this approach uses a software tool that estimates the energy consumption of the software. In fact, mathematical formulas are established to calculate the power consumption of major components such as CPU, memory, disk, network, etc., through the measurement of other data such as CPU performance or the amount of memory used. The software tools allow data at low frequencies and with different levels of granularity to be obtained. These tools do not require a great effort for their adoption, and they are cheaper than using a hardware device. The results obtained are more inaccurate than with other approaches, however, as they are estimates of energy consumption [32].
- *Hardware-based approach*: this approach consists of measuring the energy consumption of the computer, where the software is running

[1] https://www.internetlivestats.com.

with physical power meters connected to the device. It is thereby possible to obtain highly accurate results of energy consumption. The main problem is that these devices are more expensive than measuring instruments that follow the software-based approach; they are not available to everyone, moreover. Similarly, it is not possible to measure the power consumption of a particular process or virtual machine [33].

- *Hybrid approach*: this approach is a combination of the hardware-based and software-based approaches. It uses a hardware device such as a power meter to perform the energy measurement, but in addition the energy consumption information is extended with other usage data from the computer hardware components. This approach allows for energy consumption measurements that are as accurate as the hardware-based approach, with extra information. However, the main problem with the hybrid approach is that it can be very expensive to adopt, requiring both a physical device and a software tool. Moreover, the addition of several measuring instruments may imply extra energy consumption [34].

Table 7 shows a summary of the main energy measuring instruments with their characteristics, classified according to the measurement approach that follow. As can be seen in Table 1, measuring instruments within the software-based approach and in the hybrid methods are able to provide an estimate of the energy at the lowest levels of granularity and for a larger number of components. On the other hand, measuring instruments that follow a hardware-based approach provide accurate measurements of the total power consumption of the computer. Another feature evaluated is the sampling frequency, which refers to the number of power samples collected per unit.

4.4. Main contributions of FEETINGS

Considering the proposals detailed above, the main contributions of FEETINGS to promote the capture, analysis and reliable interpretation of energy consumption data from computer programs, compared to the other proposals, are presented below.

To the best of our knowledge, there is currently no common terminology related to the evaluation of the energy consumed by the software. This causes each researcher to use their own terms or concept, making it difficult to understand. FEETINGS resolves this problem by means of the ontology developed (GSMO), which introduces the key concepts, and their relationships, used in the measurement of the energy consumed by the software.

Another problem identified is that although there are several standards and methodologies that provide the necessary information to carry out an effective and systematic measurement process, none of them are specific enough to carry out the measurement of the energy efficiency of the software in a rigorous way. For this, we have defined GSMP, that allows researchers to have greater control over the measurements made, guaranteeing their reliability and consistency. It also enables studies to be easily replicated, and the results obtained to be comparable with those of other studies.

Finally, FEETINGS presents a technological environment composed of EET, a measuring instrument that follows the hardware-based approach, and ELLIOT, a software tool that processes, analyzes, and gives an adequate visualization of the data obtained by EET. This technological environment allows to obtain more precise energy consumption measurements than measuring instruments that follow the software-based approach. Moreover, EET supports the measurement of four different hardware components (processor, hard disk, graphics card and monitor) and the total energy consumption of the DUT. Another advantage of this measuring instrument is the sampling frequency, around 100 Hz, which provides very reliable consumption information.

5. Conclusions and future work

Software plays an important role in the global energy consumption of the PC. In order to develop sustainable software, it is necessary to measure its energy impact. With this in mind, this paper has presented the FEETINGS framework, which aims to promote reliable measurement, analysis and interpretation of software energy consumption data. FEETINGS is composed of three main components: the GSMO ontology,

Table 7
Comparison of different energy measuring instrument.

Approach	Measuring Instrument	Sampling frequency	Software granularity						Hardware level of details					
			OS	Application	Process	Thread	Method	Line of Code	CPU	Memory	HDD	Network	Monitor	Whole machine
Software-based approach	SPAN [35]	–			✓		✓		✓					
	pTop [36]	1 Hz			✓				✓		✓	✓		
	Jolinar [37]	2 Hz		✓					✓	✓	✓			
	Jalen [38]	100 Hz		✓		✓	✓		✓	✓	✓			
	PowerAPI [39]	2 Hz	✓		✓				✓	✓	✓			
	PowerTop [40]	1 Hz	✓		✓				✓			✓	✓	
	Joulemeter [41]	1 Hz	✓	✓					✓	✓	✓		✓	
	IPG [42]	10 Hz	✓				✓		✓					
	TEEC [43]	2 Hz		✓					✓	✓	✓			
Hardware-Based approach	Watts UP? Pro [44]	1 Hz	✓	✓										✓
	SEFLab [45]	30 Hz	✓	✓					✓	✓	✓			✓
	GreenSoM [46]	1 Hz	✓	✓										✓
	EET	**100 Hz**	✓	✓					✓	✓	✓	✓	✓	✓
Hybrid-based approach	PowerScope [47]	100 kHz			✓		✓		✓	✓	✓	✓		
	PowerPack [48]	–		✓	✓				✓	✓	✓			
	GreenHPC [49]	500 kHz			✓				✓					✓
	Proposal by ISS [5]	1 Hz	✓	✓	✓									✓

the GSMP process, and a technological component, composed of EET and ELLIOT tool.

The information about the software's energy consumption obtained from FEETINGS can be useful from two main perspectives. On the one hand, this framework allows researchers and professionals to be aware of the energy that the software they develop consumes when used, and thus be able to develop software that is more energy efficient and environmentally friendly.

On the other hand, to show end users how much energy is needed by the software we use every day, and to make them aware of the impact that software can have on the environment.

Thus, with the use of FEETINGS, we can achieve that the software developed is more energy efficient, and that users know some good practice guidelines that allow to use it in a lesser energy consumption way.

In this paper, we have also shown the application of the FEETINGS framework to analyze the energy consumption involved in writing a tweet or a post on the social networks Twitter and Facebook. With this information, we can advise users on the best way to write a publication from the point of view of energy consumption and respect for the environment.

In addition to the case study presented in this paper, other experiments have been carried out using the FEETINGS framework in order to validate and prove the benefits of it. To demonstrate that the EET is capable of obtaining accurate energy measurements, an experiment was conducted in which the consumption data obtained by EET was compared with the results obtained by a reference measurement instrument (gold standard) [22]. For the purpose of validating the use of the GSMP, in paper [50], two different studies are presented, using different measuring instruments (one with a hardware-based approach and one with a software-based approach), proving that the process can be adapted to any experiment, regardless of the chosen measuring instrument. Moreover, other experiments have been performed where the energy consumption of the software is evaluated using the FEETINGS framework as presented in the beginning of Section 3 of this paper.

As future work, we will continue to carry out more studies to analyze software energy consumption, with the aim of providing more information to software professionals, so they can develop a more environmental respectful software; and on the other hand, continue to raise awareness among software users of the impact that the massive use of software has on energy consumption and CO2 emissions.

In addition, we will perform an in-depth analysis of the energy consumption involved in analyzing the data using the FEETINGS framework.

Author statement

Javier Mancebo: Conceptualization, Validation, Investigation, Writing - Original Draft

Coral Calero: Conceptualization, Validation, Writing - Review & Editing, Supervision

Félix Garcia: Conceptualization, Methodology, Writing - Review & Editing, Supervision

Mª Angeles Moraga: Conceptualization, Validation, Writing - Review & Editing

Ignacio García-Rodríguez de Guzmán: Conceptualization, Writing - Review & Editing, Visualization

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was part of the BIZDEVOPS-Global (RTI2018-098309-B-C31), supported by the Spanish Ministry of Economy, Industry and Competitiveness and European FEDER funds, and was also part of the SOS project (No. SBPLY/17/180501/000364) and TESTIMO project (SBPLY/17/180501/000503), both funded by the Department of Education, Culture and Sports of the Directorate General of Universities, Research and Innovation of the JCCM (Regional Government of the Autonomous Region of Castilla-La Mancha).

References

[1] H. Acar, et al., Software greenability: a case study of cloud-based business applications provisioning, in: 2018 IEEE 11th International Conference on Cloud Computing (CLOUD), IEEE, 2018, pp. 875–878.
[2] A. Andrae, Total consumer power consumption forecast, Nordic Digital Business Summit vol. 10 (2017).
[3] A. Andrae, Prediction studies of electricity use of global computing in 2030, Int. J. Sci. Eng. Invest. 8 (2019) 27–33.
[4] J. Vidal, Tsunami of Data'Could Consume One Fifth of Global Electricity by 2025, vol. 11, Climate Home News, 2017.
[5] E. Kern, et al., Sustainable software products—towards assessment criteria for resource and energy efficiency, Future Gener. Comput. Syst. 86 (2018) 199–210.
[6] G. Pinto, F. Castor, Energy efficiency: a new concern for application software developers, Commun. ACM 60 (12) (2017) 68–75.
[7] B. Penzenstadler, et al., Systematic mapping study on software engineering for sustainability (SE4S), Proceedings of the 18th International Conference on Evaluation and Assessment in Software Engineering (2014) 1–14.
[8] A. Fonseca, R. Kazman, P. Lago, A manifesto for energy-aware software, IEEE Softw. 36 (6) (2019) 79–82.
[9] M.Á. Moraga, M.F. Bertoa, Green software measurement. Green in Software Engineering, Springer, 2015, pp. 261–282.
[10] The Contribution of ICT to Energy Efficiency: Local and Regional Initiatives, European Union, 2011.
[11] P. Bozzelli, Q. Gu, P. Lago, A Systematic Literature Review on Green Software Metrics, VU University, Amsterdam, 2013.
[12] C. Calero, M. Piattini, Introduction to green in software engineering. Green in Software Engineering, Springer, 2015, pp. 3–27.
[13] C. Calero, M. Piattini, Puzzling out software sustainability, Sustain. Comput. Inform. Syst. 16 (2017) 117–124.
[14] S. Naumann, M. Dick, E. Kern, T. Johann, The greensoft model: a reference model for green and sustainable software and its engineering, Sustain. Comput. Inform. Syst. 1 (4) (2011) 294–304.
[15] C. Becker, et al., Sustainability design and software: the karlskrona manifesto, in: IEEE2015 IEEE/ACM 37th IEEE International Conference on Software Engineering, 2, 2015, pp. 467–476.
[16] C.C. Venters, et al., Software sustainability: the modern tower of babel, in: CEURCEUR Workshop Proceedings, vol. 1216, 2014, pp. 7–12.
[17] F. García, et al., Towards a consistent terminology for software measurement, Inf. Softw. Technol. 48 (8) (2006) 631–644.
[18] C. Tautz, C.G. von Wangenheim, REFSENO: a Representation Formalism for Software Engineering Ontologies, Fraunhofer-IESE, 1998.
[19] B. Henderson-Sellers, Method engineering for OO systems development, Commun. ACM 46 (10) (2003) 73–78.
[20] J. Mancebo, et al., EET: a device to support the measurement of software consumption, Proceedings of the 6th International Workshop on Green and Sustainable Software (2018) 16–22.
[21] M. Piattini, et al., Aparato para medición del consumo eléctrico de equipos informáticos (PC)," Patent ES 1199234 Y, Available:, 2018 https://bopiweb.com/elemento/2602662/.
[22] J. Mancebo, et al., Assessing the sustainability of software products—a method comparison. Advances and New Trends in Environmental Informatics, Springer, 2020, pp. 1–15.
[23] C. Calero, et al., Energy efficiency of personal health records, Multidisciplinary Digital Publishing Institute Proceedings vol. 2 (2018) 510, 19.
[24] J. Mancebo, et al., Saving energy in text search using compression, in: Presented at the GREEN 2019 : The Fourth International Conference on Green Communications, Computing and Technologies, Nice, France, 2019.
[25] J. Mancebo, C. Calero, F. García, Does maintainability relate to the energy consumption of software? A case study, Softw. Qual. J. (2021) 1–27.
[26] B. Chandrasekaran, J.R. Josephson, V.R. Benjamins, What are ontologies, and why do we need them? IEEE Intell. Syst. Appl. 14 (1) (1999) 20–26.
[27] V.R. Basili, D.M. Weiss, A methodology for collecting valid software engineering data, IEEE Trans. Softw. Eng. (6) (1984) 728–738.
[28] W. Goethert, J. Siviy, Applications of the indicator template for measurement and analysis. Carnegie-Mellon Univ Pittsburgh Pa Software Engineering Inst, 2004.
[29] PSM: Practical Software and Systems Measurement - a Foundation for Objective Project Management, 2000.
[30] A. Hindle, Green mining: a methodology of relating software change and configuration to power consumption, Empir. Softw. Eng. 20 (2) (2015) 374–409.

[31] E.A. Jagroep, et al., Software energy profiling: comparing releases of a software product, Proceedings of the 38th International Conference on Software Engineering Companion (2016) 523–532.
[32] A. Noureddine, R. Rouvoy, L. Seinturier, A review of energy measurement approaches, ACM Sigops. Oper. Syst. Rev. 47 (3) (2013) 42–49.
[33] T.A. Ghaleb, Software energy measurement at different levels of granularity, in: 2019 International Conference on Computer and Information Sciences (ICCIS), IEEE, 2019, pp. 1–6.
[34] H. Zhang, H. Hoffmann, Maximizing performance under a power cap: a comparison of hardware, software, and hybrid techniques, ACM Sigplan. Not. 51 (4) (2016) 545–559.
[35] S. Wang, H. Chen, W. Shi, SPAN: a software power analyzer for multicore computer systems, Sustain. Comput. Inform. Syst. 1 (1) (2011) 23–34.
[36] T. Do, S. Rawshdeh, W. Shi, Ptop: a Process-level Power Profiling Tool, HotPower, 2009.
[37] A. Noureddine, S. Islam, R. Bashroush, Jolinar: analysing the energy footprint of software applications, Proceedings of the 25th International Symposium on Software Testing and Analysis (2016) 445–448.
[38] A. Noureddine, A. Bourdon, R. Rouvoy, L. Seinturier, Runtime monitoring of software energy hotspots, in: 2012 Proceedings of the 27th IEEE/ACM International Conference on Automated Software Engineering, IEEE, 2012, pp. 160–169.
[39] A. Bourdon, A. Noureddine, R. Rouvoy, L. Seinturier, PowerAPI: a Software Library to Monitor the Energy Consumed at the Process-Level, vol. 2013, ERCIM News, 2013 no. 92.
[40] (25 /02/2020). PowerTop. Available: https://01.org/powertop.
[41] N. Kothari, A. Bhattacharya, Joulemeter: Virtual Machine Power Measurement and Management, MSR Tech Report, 2009.
[42] S.-W. Kim, J.J.-S. Lee, V. Dugar, J. De Vega, Intel® Power Gadget, vol. 7, Intel Corporation, 2014.
[43] T. Carçao, Measuring and visualizing energy consumption within software code, in: 2014 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC), IEEE, 2014, pp. 181–182.
[44] J.M. Hirst, J.R. Miller, B.A. Kaplan, D.D. Reed, Watts up? Pro Ac Power Meter for Automated Energy Recording, Springer, 2013.
[45] M.A. Ferreira, E. Hoekstra, B. Merkus, B. Visser, J. Visser, Seflab: a lab for measuring software energy footprints, in: 2013 2nd International Workshop on Green and Sustainable Software (GreenS), IEEE, 2013, pp. 30–37.
[46] V. Cordero, I.G.-R. de Guzmán, M. Piattini, A first approach on legacy system energy consumption measurement, in: 2015 IEEE 10th International Conference on Global Software Engineering Workshops, IEEE, 2015, pp. 35–43.
[47] J. Flinn, M. Satyanarayanan, Powerscope: a tool for profiling the energy usage of mobile applications, in: Proceedings WMCSA'99. Second IEEE Workshop on Mobile Computing Systems and Applications, IEEE, 1999, pp. 2–10.
[48] R. Ge, et al., PowerPack: energy profiling and analysis of high-performance systems and applications, IEEE Trans. Parallel Distrib. Syst. 21 (5) (2009) 658–671.
[49] G. Rostirolla, R. da Rosa Righi, V.F. Rodrigues, P. Velho, E.L. Padoin, GreenHPC: a novel framework to measure energy consumption on HPC applications. 2015 Sustainable Internet and ICT for Sustainability (SustainIT), IEEE, 2015, pp. 1–8.
[50] J. Mancebo, F. García, C. Calero, A process for analysing the energy efficiency of software, Inf. Softw. Technol. (2021) 106560.