Conception and test of a measuring station for the analysis of the resource and energy consumption of material flow-oriented environmental management information systems (EMIS)

Dennis Junger[1], Volker Wohlgemuth[2] and Eike Kammer[3]

Abstract: The aim of this work is to design a measurement setting that enables the analysis of the resource and energy consumption of material flow oriented corporate environmental information systems (BUIS). For this purpose, the focus is put on the Life Cycle Assessment (LCA) software. LCA software supports LCA mainly in Life Cycle Inventory (LCI) and in Life Cycle Impact Analysis (LCIA). For the impact assessments, through which potential impacts on the environment are calculated, the LCA software requires many substance-specific life cycle impact data sets, which are stored in databases. In this work, we mainly refer to LCA software that runs in the form of desktop applications. In order to create a measurement configuration suitable for LCA software, the software to be tested and the necessary methodology for desktop solutions are searched. The found solution for sustainable evaluation of software products is adapted to the goal of comparing the efficiency LCA software. The determined measurement standard will be applied, modified and extended. The measurement configuration will be built and tested. Problems of a theoretical nature and problems in setting up the tests are addressed. Any problems of the comparability basis that may arise when performing tests on the selected LCA in the software in the future will be addressed, and a possible solution path will be shown. An outlook on the further course of the measurements is given.

Keywords: Green IT, Energy Consumption, Green Coding, Life Cycle Assessment Software, Desktop Software

Addresses Sustainable Development Goal 12: Responsible consumption and production

1. Introduction

The sustainable design of material and substance flows is inevitable in order to best counter the growing number of environmental crises, such as the hole in the ozone layer, acid rain or global warming. In response to previous environmental crises, the german

[1] HTW Berlin, University of Applied Sciences, Industrial Environmental Informatics Unit, Wilhelminenhofstr. 75A, 12459 Berlin, jungerdm@htw-berlin.de, https://orcid.org/0000-0002-9271-2261
[2] HTW Berlin, University of Applied Sciences, Industrial Environmental Informatics Unit, Wilhelminenhofstr. 75A, 12459 Berlin, Volker.Wohlgemuth@HTW-Berlin.de
[3] HTW Berlin, University of Applied Sciences, Electrical Engineering Unit, Wilhelminenhofstr. 75A, 12459 Berlin, eike@kammer-cloud.de

Enquête Commission adopted the Agreement on the "Protection of Man and the Environment" in 1992. Since then, there has been an obligation for governmental and company actors to carry out material flow analyses. Material flow analyses, which are prepared on a unit or product basis, are product life cycle assessments (LCA) or, more generally, life cycle assessments, which are defined under the ISO standard 14040 and 14044. Such LCAs can be used to derive product environmental declarations and environmental impact assessments (LCIA), such as a carbon footprint. The first production system to be LCA's in Germany was that of beverage packaging in the 1970s. [TU22] The ISO standard 14040 defines the principles and methods of the same, whereas the standard 14044 specifies the steps of the LCA. According to ISO 14040, a typical LCA is composed of four phases: Goal and Scope of Investigation, Life Cycle Inventory (LCI), Impact Assessment (LCIA), and the Interpretation.

The environmental impact of software depends on the hardware used. As described in Fig.1, the hardware basically consumes natural resources, primarily in the form of electrical energy during the time of usage. This is consumed mainly by the components listed under "capacities used", or operating resources. In addition to uninstallation, these are mainly required for the production and use of the software.

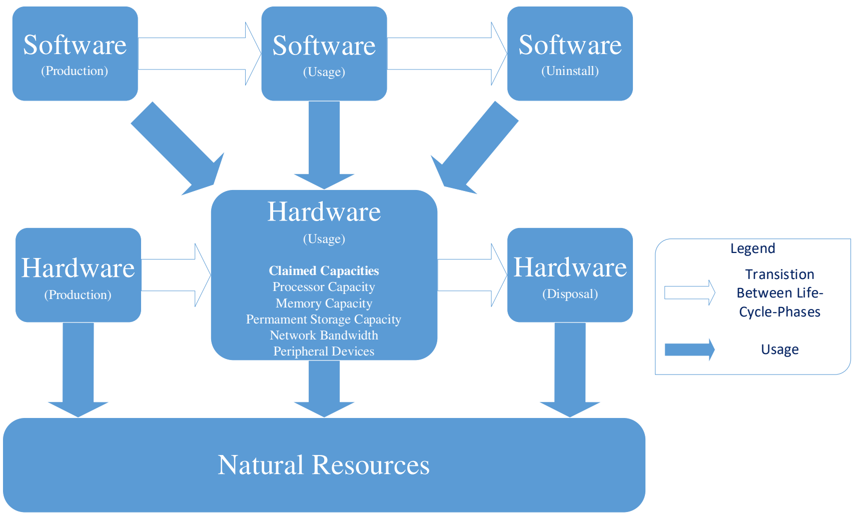

Fig. 1: Life Cycle of Hardware and Software modified [Gr18]

The chosen Software is usually used to create LCAs of products. Here, manufacturing and transport steps as well as their individual impacts of products are quantified as precisely as possible and calculated by means of software solutions that take into account as many circumstances of the entire process as possible with the help of large (impact) databases. The production paths to be balanced, which can be represented as Petri nets, are modeled and calculated in the software. Thousands of upstream and downstream data sets are usually required to map production paths, and detailed material and energy flows must be described for each one. [LC22] In order to query these LCI data sets in the LCA software in an energy-saving and high-performance way, efficient databases on the one hand and efficient query algorithms on the other hand are required. In addition to data management, data retrieval and data storage, LCA also differs in the algorithms of the mathematical procedures. [BS22] Once LCA models gain in size and number of elements examined, computation time and memory requirements can increase rapidly, with varying resource requirements also noticeable when computing linear than recursive models. Currently available LCA software is implemented in a wide variety of programming and scripting languages and their frameworks. Examples would be OpenLCA in Java [OL22], the Activity Browser with Brightway backend in Python [AB22], and Umberto LCA+ in .NET (C#) [IS22]. The framework chosen can also mean a difference in the efficiency and resource consumption of the software. In the course of this work, a measurement framework and setting is created to test already produced LCA software products in the usage phase of their life cycle. The aim is to get an accurate picture of the energy consumption and power distribution within the hardware in order to measure relevant operating parameters of LCA software and to verify the functionality of the tested in general.

2. Conception of a test setup

2.1 Categorization of the LCA software

To create a suitable test scenario, the software to be tested must be characterized according to its properties. It is categorized according to the location of the data processing and the location of the data storage. A distinction is made between the following four software classes: Local application, remote data storage application, remote processing application, and server service. [BE22] In order to choose the software and the necessary measurement setup correctly, it is necessary to classify the corresponding architecture pattern of the software.

There are currently several different LCA software on the market. In addition to purely web-based software, the main focus in this work is on locally executed software. Characteristic for this type of software is that the user interface, application logic and data management are mainly executed on the local system.

The category chosen for the selected LCA software corresponds here to that of the "local application". The concept of the measuring station is thus based on existing measuring stations for local desktop applications.

2.2 The designed measurement setup

In the measurement setup shown in Fig. 2, a distinction is made between "on-site" and "remote" settings. Here, attention was paid to extending the measurement setup presented in the Blue Angel [BE22] to include a remote measurement method. Especially in the Covid-19 pandemic, access to offices and universities was difficult due to restrictions. In the remote setting, the System Under Test (SUT) and the energy data aggregator are accessed via the control & evaluation station. Here, the Workload Generator (WG), the resource measurement, and the energy data collection are started. It is necessary to start the WG via the control and evaluation station so that there is still an active remote connection to the SUT if the connection to the test network is interrupted. This way the signal resolution of the SUT is constant. Alternatively, the WG and the resource measurement of the SUT can be started manually on site. The energy measurement is started at the control & evaluation station as with the remote measurement. This is currently necessary because of the different operating systems. An NTP server provides the current time for the entire measurement environment and is used for the subsequent assignment of the measurement data in the evaluation. The energy data aggregator queries the energy data of the energy meter via SNMP. These are stored on the aggregator in the form of a log file. After completion of the measurements and data collection, the control & evaluation station evaluates the data using the evaluation software "Open Source Software Consumption Analysis and Reporting" (OSCAR) from Trier University of Applied Sciences and creates an energy report.

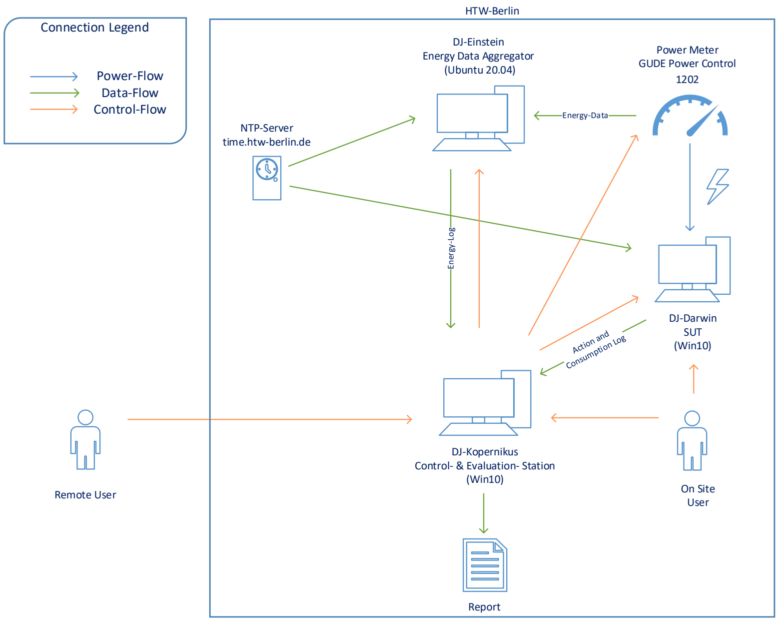

Fig. 2: Information flow chart of the measurement setup

2.3 Problem of comparability of measurements with solution approach

The Blue Angel for software products [BE22] aims to cyclically optimize measurements of a software solution in a continuous improvement process. When different software is compared, the question arises as to what extent the measurement processes are comparable. Tasks and actions to be performed are defined and executed. In order to be able to repeat and compare measurement sequences, they must be recorded and replayed using macros. The same macro cannot be used for different software because different software interfaces, dialogs or operating modes are implemented. In order to be able to play the macros require these after inputs a delay time in form of a pause, so that the software can react to the transacted control instructions. Different numbers of input and output loops now lead to a different number of pauses and thus to a different measurement duration of the software. The length of the measurement pauses is also decisive for the measurement result. The approach of ISO/IEC 14756 [UC22] is that these delays are defined for different usage types and are run through, weighted and averaged by means of a measurement plan. The intersection of the solutions presented is chosen. When building LCA, a very large part of the modeling consists of analyzing and understanding the model being built. This usually takes place with the PC turned on without user interaction and is tied to work time rather than usability. Thus, consistent measurement length is prioritized. Measurements are divided into tasks and actions with the same content and time (e.g. 60 seconds). Furthermore, the total duration of the measurements is defined (e.g. 600 seconds). The advantage is that the human component is not considered. A better measurement result is obtained, with which also conclusions can be made about the efficiency of the programming, about the efficiency of used algorithms, frameworks and databases. In order to avoid measurement outliers, a statistical mean value is formed over 30 measurement processes as specified in [BE22].

2.4 System Under Test (SUT) hardware

The hardware selection is based on the recommendations of the blue angel. In "Development and application of evaluation bases for resource-efficient software taking into account existing methodology", it is described how a reference system for software measurements is created on the basis of the ‘bwPC’, a desktop computer defined once a year for the public service in Baden-Württemberg as the state of the art. Here, the “Öko-Institut e.V.” proposed a SUT for the years 2010 to 2017. In order to select a current SUT for the year 2021, the hardware configuration was based on the hardware noted in the "bwPC Formular v6.2.7" of the computer center of the University of Freiburg at the time of procurement. An almost identical system was ordered from the current HTW-Berlin contract partner Dell.

2.5 Special features of the display unit and the display settings of the SUT

Since the energy consumption of the display can vary greatly depending on the display technology used, the display device is not measured when designing the measuring station. However, even if the display device is not part of the system to be tested, it has a considerable influence on the measurement. Among other things, the display resolution, signal resolution, color depth and also the repetition rate have an influence on the calculation effort of the display. Since the Blue Angel does not define a display device for software, the test setup assumes that the system under test runs at the standard resolution recommended for Windows 10 for the monitor used. The default refresh rate of 60 hertz and the color depth of 32 bits are selected. The default Windows resolution depends on the screen size and aspect ratio. In order to ensure traceability of future tests, the bwMonitor of the computer center of the University of Freiburg was chosen analogous to the bwPC. This is currently a 24-inch widescreen Full HD monitor (Philips 241B8QJEB) with a 16:9 resolution of 1920 x1080 pixels.

It is recommended to use this standardization also in the Blue Angel for software products.

2.6 The energy data aggregator

As recommended in the Blue Angel, a measuring device is used that aggregates the measured power consumption over one second and can save it together with a time stamp in a log file. [BE22] The "Gude Expert Power Control 1202" was chosen, which is read out via SNMP (here from the energy data aggregator). The script used was obtained from Trier University of Applied Sciences and extended by the functionality of a time synchronization. The current version is provided in [GL22].

2.7 Image of the measuring station

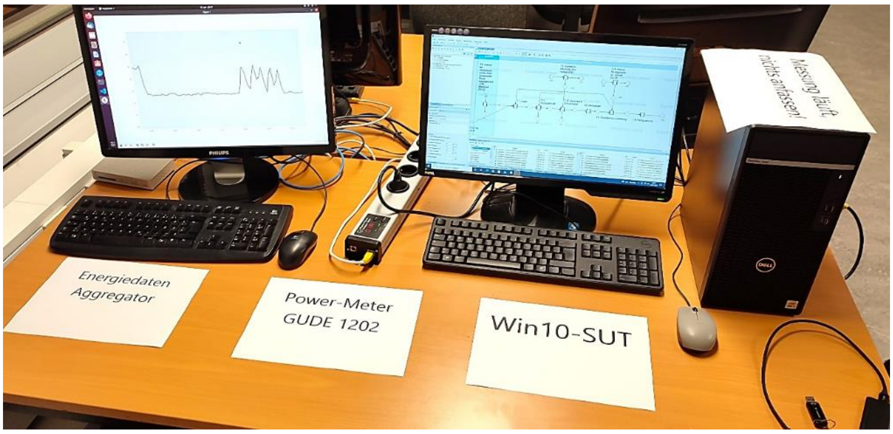

Fig.3: The measuring station with loaded Umberto LCA+ model

2.8 Selection of the SUT equipment devices to be monitored

Apart from the Blue Angel, there is yet no measurement methodology with corresponding measurement standards for carrying out resource- and energy-related software measurement. [BE22] The unique selling point of the Blue Angel software measurement methodology, compared to existing hardware measurements for energy consumption evaluation, is the inclusion of the operating equipment consideration for finding the cause of additional consumption. The choice of resources was based on the evaluation software "OSCAR" version 0.190404 published by the Environmental Campus of the University of Trier (resources: "data carrier physical" (written and read), "processor time", "working memory assured" and "swap file"). The resources are recorded like the energy data in intervals of 1 second each by means of the Windows-own and preinstalled program "performance monitoring" (perfmon.exe). The network traffic is recorded by means of "network adapter" (send and receive) in the measurement instructions of Trier University. In this test setup the "network interface" (send and receive) is also recorded to not only record the physical signal amount, but also signal amounts that take place through connections with "localhost"(loopbacks). Recording facilitates the detection of bottlenecks or energy wastage.

3. Test measurements

In order to test the functionality of the measuring station, test measurements were carried out (600 seconds, 30 repetitions, full screen mode, LCIA data sets according to Tab. 1). In each case, an idle measurement of the listed software and an idle measurement without software (baseline) were made for the evaluation. The LCIA data of Ecoinvent 3.6 was chosen as database basis. Due to the use of different data formats, import difficulties of these as well as also license problems not the same LCIA data sets could be imported into the tested software. By Tab. 1 the following conclusions are obvious:

- OpenLCA requires less power in idle mode than Umberto LCA+ or Brightway2 with Activity Browser frontend
- Software frameworks (.NET, Java, Python) can create differences
- It’s possible that different number of data sets can have an impact on RAM usage.
- Umberto LCA+ required a larger RAM usage and permanent memory usage than OpenLCA and Activity Browser (maybe due different user interface or technological stack)

For further qualitative statements, further measurements according to the measurement plan are necessary.

Test measurement with data sets Idle, On site	**Umberto LCA+**	**Activity-Browser**	**OpenLCA**

Imported LCIA data	Ecoinvent: APOS 3.6 Consequential 3.6 CutOff 3.6	Ecoinvent: APOS 3.6 Consequential 3.8 CutOff 3.6	ELCD 3.2 Agribalyse
Mean el. power	8,24 W	9,53 W	7,91 W
Mean el. work	1,37 Wh	1,59 Wh	1,32 Wh
Average CPU usage	0,55%	4,02%	0,87%
Average RAM usage	40,07%	39,81%	36,73%
Amount of data transferred via network	781,0612269 MByte	444,7875595 Mbyte	736,2177884 MByte
Permanent storage usage	7, 5396711 × 10^5 MByte	1, 9882778 × 10^5 Mbyte	2, 1451082 × 10^5 Mbyte

Tab.1: Preliminary energy and resource comparison Idle on site

4. Conclusion and Outlook

It was possible to identify the most significant criteria for the measurement site. The software to be tested was categorized as desktop software and thus a measurement concept was sought. The Blue Angel for resource and energy efficient software products was found. Based on the setting proposed here, a setup of a measurement station was designed. The possibility of remote measurements was added to the measurement concept. In order to validate this, further measurements are necessary.

Current weaknesses of the SUT configuration of the existing methodology are the non-standardized display settings. Different resolutions, color depths and refresh rates affect the power and resource consumption. Based on the specification of the bwMonitor, defined by the computer center of the University of Freiburg for public authorities, a possible standard could be found. This can now be recommended as an extension of the Blue Angel criteria to couple the screen resolution with the device information and Windows standard settings.

Basic considerations of a measuring place and measuring procedure were conceived. User-independent measurement scenarios with normalized times were chosen to compare existing software in the LCI and LCIA phases. To highlight the differences of the linear and recursive calculation, suitable usage scenarios have to be defined. Commonly used LCIA methods that determine, store and export the potential fossil energy input (KEA_{fossil}) or the potential global warming potential (GWP_{100}) are suitable.

For the exclusion of performance differences in the data set construction, the measurements are performed with LCIA data from the same databases. These should have the same data sets imported. Current LCIA data base is the Ecoinvent 3.6, because they are already integrated in Umberto LCA+ and a license is available at the university. The OpenLCA software does not import the selected LCIA data in their original form, but only pre-processed and are extra to purchase. Alternative data can be found in a variety of different formats and cannot be imported in every software. Looking ahead, standardized formats of LCIA data would simplify the import into any LCA software and therefore will improve the measurement results.

During the conception of the test phase, it was noticed that it was difficult to impossible to obtain appropriate licenses for the software to be tested without having to purchase them. The sales department of the companies could not be convinced, despite clear target communication, to allow a free comparison with their software. It would be advisable for procedures such as the Blue Angel for sustainable software products to be implemented not only as an eco-label, but also, for example, in the form of directives, e.g. within the European Ecodesign Directive.

Bibliography

[BE22] https://www.umweltbundesamt.de/sites/default/files/medien/479/publikationen/texte_119-2021_umweltzeichen_blauer_engel_fuer_ressourcenund_energieeffiziente_softwareprodukte.pdf accessed: 19/05/2022

[BS22] Dr. Bernhard Steubing, Universität Leiden - mail 2022

[GL22] https://gitlab.rlp.net/green-software-engineering/mobiles-messgerat/-/blob/master/get_energy_utilization_mit_Sync.py, accessed: 19/05/2022

[Gr18] Gröger et al. 2018 (Universität Zürich - UZH)

[TU22] https://www.semanticscholar.org/paper/Vergleich-von-Softwarel%C3%B6sungen-f%C3%BCr-die-%E2%80%93-eine-L%C3%BCdemann-Feig/cfb699d68e9e9c1f8dea22fbc92fdfd1de13fddf , accessed: 19/05/2022

[UC22] https://www.umwelt-campus.de/fileadmin/Umwelt-Campus/Greensoft/measurement_of_software-induced_energy_consumption_web.pdf, accessed: 19/05/2022