Renewable and Sustainable Energy Reviews 159 (2022) 112174

Contents lists available at ScienceDirect

Renewable and Sustainable Energy Reviews

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

Evaluating the usability of open source frameworks in energy system modelling

S. Berendes [a,*], S. Hilpert [b], S. Günther [b], C. Muschner [a], S. Candas [c], K. Hainsch [d], J. van Ouwerkerk [e,f,g], S. Buchholz [h], M. Söthe [b]

[a] Reiner Lemoine Institut gGmbH, Rudower Chaussee 12, 12489 Berlin, Germany
[b] Europa-Universität Flensburg, Auf dem Campus 1, 24943 Flensburg, Germany
[c] Chair of Renewable and Sustainable Energy Systems, Technical University of Munich, Lichtenbergstrasse 4a, 85748, Garching bei München, Germany
[d] Fachgebiet Wirtschafts- und Infrastrukturpolitik (WIP), Technische Universität Berlin, Straße des 17. Juni 135, 10623 Berlin, Germany
[e] Institute for Power Electronics and Electrical Drives (ISEA), RWTH Aachen University, Jägerstraße 17-19, 52066 Aachen, Germany
[f] Institute for Power Generation and Storage Systems (PGS), E.ON ERC, RWTH Aachen University, Mathieustraße 10, 52074 Aachen, Germany
[g] Jülich Aachen Research Alliance, JARA-Energy, Germany
[h] Technical University of Denmark, Produktionstorvet Bygning 424, 2800 Kgs. Lyngby, Denmark

ARTICLE INFO

Keywords:
Energy system modelling,
Optimisation frameworks
Usability testing method
Usability questionnaire
PSSUQ
Open Source
Open Science
Software engineering
Software usability

ABSTRACT

Usability is considered one of the key factors in determining the success of Open Source Software, but is it sufficiently addressed within the development process as of yet? Thus far, there are no studies on record that explicitly examine the usability of Open Source Software in the field of Energy System Modelling. In this paper, we publish a novel method, the Energy System Modelling Usability Testing (ESMUT), including a step-by-step guide on how to apply the method and the corresponding usability questionnaire (ESMUQ). The questionnaire is based on the Post Study System Usability Questionnaire and adopted for quantitative usability testing of Open Source Energy System Modelling frameworks. To illustrate its usage and show its applicability, we apply the method in a case study with five frameworks (Balmorel, GENeSYS-MOD, GENESYS-2, oemof, and urbs) and within a group of eight framework developers. Based on the case study results, we find that the participants were largely satisfied in working with the frameworks analysed, and identify correct handling of input data and error messages as the most frequently mentioned problems when working with the frameworks. Consequently, we find that the usability of the frameworks analysed in the case study requires further improvement. Due to the fact that only the developer perspective was taken into account, and the number of participants involved in the study was limited, further research is required to assess the usability of Open Source Energy System Modelling frameworks.

1. Introduction

In recent years, new scientific practices for the broad dissemination, accessibility and reproducibility of research methods and results have emerged. These practices, driven by modern information technologies, can be summarised under the umbrella of Open Science. According to [1], no clear definition of Open Science exists; however, transparency, accessibility and collaboration are essential aspects associated with Open Science practices.

In computational science, Open Source Software (OSS) is a prerequisite for Open Science to enable transparency, reproducibility and quality (TREQ) of research [2]. In order to achieve these goals, an OSS project must be able to be used by researchers other than the ones developing or maintaining the software, because results generated by software which is only understood by its developers, are neither transparent, nor reproducible. On top of that, closed software acts as a hindrance for collaboration, instead of encouraging it. Thus, the more accessible a piece of scientific OSS is, the more transparent and reproducible the research done using this piece of OSS becomes. A decisive success factor in this context is the usability of the software [3]. This means that software tools must not only be scientifically sound, but also designed such that they are easily accessible and useable by external researchers outside of the development process.

In the field of Energy System Modelling (ESM), many OSS frameworks have been developed in recent years, primarily for researchers

* Corresponding author.
E-mail address: sarah.berendes@rl-institut.de (S. Berendes).

https://doi.org/10.1016/j.rser.2022.112174
Received 14 July 2021; Received in revised form 18 December 2021; Accepted 19 January 2022
Available online 16 February 2022
1364-0321/

List of abbreviations

APL	Adaptive Public License
ESM	Energy System Modelling
ESMUQ	Energy System Modelling Usability Questionnaire
ESMUT	Energy System Modelling Usability Testing
GAMS	General Algebraic modelling System
GENESYS-2	Genetic Optimisation of a European Energy Supply System
GENeSYS-MOD	The Global Energy System Model
GPL	General Public License
HCI	Human Computer Interaction
IAMC	Integrated Assessment modelling Consortium
ISC	ISC License originated at the Internet Systems Consortium
LP	Linear Programming
MI/LP	Mixed Integer/ Linear Programming
MIT	MIT License originated at Massachusetts Institute of Technology
oemof	Open Energy Modelling Framework
OEO	Open Energy Ontology
OS	Open Source
OS-ESM	Open Source Energy System Modelling
OSS	Open Source Software
PSSUQ	Post-Study System Usability Questionnaire
STEM	Science, Technology, Engineering, and Mathematics
UCD	User Centred Design

and engineers to answer various questions about a climate-friendly future for energy supply. We refer to these as Open Source Energy System Modelling (OS-ESM) frameworks. We define OS-ESM frameworks as scientific, techno-economic optimisation tools for capacity planning and dispatch optimisation of energy system components using a bottom-up or hybrid analytical approach. They are sometimes also referred to as (auto) model generators and are defined as a "universal, reusable software environment that provides particular functionality as part of a larger software platform to facilitate development of software applications, products and solutions" in [4]. OS-ESM frameworks are licensed under an OS licence and can be used to develop and solve many different energy system models to answer a variety of research questions.

Looking at the usability of OS-ESM frameworks, we find that to date there is no academic literature based on empirical data. Hence, [5] identify *utilisation* as one of five main challenges of OSS in the field of ESM. The issue of usability of OS-ESM frameworks needs to be understood within the specific particularities of the field. First, both developers and users of OS-ESM frameworks usually have an educational background in Science, Technology, Engineering, and Mathematics (STEM). Second, in many cases there is an overlap between the roles of developers and users. These two specifics suggest that usability testing should be performed with OS-ESM frameworks to understand their specific usability requirements. In order to generate empirical data as well as evaluate and improve OSS usability, usability techniques are indispensable. The domain of Human Computer Interaction (HCI) has provided such methods for the development processes of OSS and scientific software. So far, however, no method exists that is adapted to the specific test case of OS-ESM frameworks.

Consequently, in this work, we publish a novel method to evaluate the usability of OS-ESM frameworks empirically, and demonstrate its applicability in a case study. The developed Energy System Modelling Usability Testing (ESMUT) method builds on existing usability evaluation methods of OSS and scientific software in general, but extends and customises them for the test case of OS-ESM frameworks. In addition to usability testing of a specific framework (single framework approach), ESMUT can also be applied to cross-evaluate several frameworks and compare the results by having a group of developers test each other's frameworks (cross-evaluation approach). The ESMUT method is thus also a systematic collaboration and comparison tool, that can contribute positively to improving the quality of OS-ESM frameworks and scientific reproducibility, respectively.

The remaining paper is structured as follows. This introduction is succeeded by Section 2 which provides an overview of literature dealing with the definition of usability and its scientific evaluation in scientific software and OSS in general, as well as in ESM. Based on this, we present the ESMUT method in Section 3, which is then applied in a case study in Section 4 to illustrate its application. The findings and the proposed method are critically evaluated in Section 5 where we give recommendations for next steps and fields of future research. Section 6 summarises the main conclusions of our work.

2. Background

2.1. Definition of usability

Usability is defined within different technical standards. According to ISO 9241-11 [6], *usability* describes "the extent to which a system, product or service can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use". Another standard – the ISO/IEC 9126-1 [7] – refers to *usability* as "the capability of the software product to be understood, learned, used and attractive to the user, when used under specified conditions". In this study, we adopt the definition provided by ISO/IEC 9126-1 [7] and defining that the software product we analyse is an OS-ESM framework.

2.2. Usability in scientific software and Open Source Software

As argued above, usability is one of the key factors for the success of scientific software [3], however, this factor is still often neglected in its development process [8]. One reason for this is that the developers of scientific software often have expertise in the domain for which the software is being developed, but do not have a good training in software engineering [9]. In addition, developers are often also users of their own software products, which makes it difficult for them to understand and integrate the specific user requirements of a broader user community. As a result, scientific software tends to have such specific usage requirements that non-developer users often have problems using the software without support from the developers [10]. In the same study, [10] identify the difficulty of obtaining funding as another barrier to improving the usability of scientific software [10].

To improve usability in scientific software, the domain of HCI has presented usability techniques ranging from design principles to evaluation methods. Macaulay et al. in [10] and Rampersad et al. in [8] showcase the integration of User Centred Design (UCD) in scientific software development for the fields of microscopy and astronomy, respectively. According to Llerena et al. [8], constant collaboration between developers and domain users leads to usability improvements. Furthermore, in [9] Queiroz et al. present a review on "good" usability design principles and recommend the integration of the UCD method while at the same time state that UCD has been poorly applied in scientific software development.

Studies on the usability of OSS, that is not explicitly used for scientific purposes, show similar challenges as presented above for scientific software.

Numerous studies, both current and less recent, acknowledge the importance of usability, and thus, indicate that usability of OSS is still not sufficiently addressed. In Raza et al. [11], 60% of the study respondents named poor usability as the main barrier to be tackled for (non-developer) users that are about to migrate from commercial software to OSS. But how can these barriers be overcome? According to [12], Llerena et al. state that OSS communities are generally uninformed about usability techniques developed by the domain of HCI and point out that OSS communities mostly do not have the resources to perform usability testing or to involve a usability expert due to the voluntary nature of the OSS development process. Furthermore the OSS development process is often organised in a decentralised way [13] and usability testing is perceived as unstructured compared to proprietary software [14]. Thus, Aberdour in [14] and Porter et al. in [15] advise OSS developers to do formal and detailed usability testing. Furthermore, Wang et al. in [16] find that most developers follow a system-centric rather then a user-centric design approach, which indicates that in practice OSS developers still prioritise functionality over usability. This is also supported by Nichols et al. in [17] as well as Viorres et al. in [18]. One reason for prioritising functionality over usability is the fact that developers overestimate their software's usability because their knowledge makes them inherently unfit to objectively assess the learn-ability or ease of understanding of the software they build. This assumption is supported by previous studies, in which researchers identified a strong overlap of developers and users leading to a developer-centred bias in software engineering such as Bødker et al. in [19] or Feller et al. in [20].

The literature on usability of scientific software and OSS provides orientation and a basis for the specific test case of OS-ESM frameworks. Since ESM is a technical domain, it is particularly susceptible to an overlap between developers and users. This makes it particularly difficult to address usability within the development process of OS-ESM frameworks.

2.3. Usability in open source software within the domain of energy system modelling

A literature review uncovered only very few studies in which the quality of software in the domain of ESM is assessed with a focus specifically on OSS. In order to still be able to provide an overview of the current state of software usability in this field, we broadened our review of related studies to ones which are not exclusively limited to OSS projects.

In a review publication, Machado et al. in [21] analyse 34 software tools and 442 studies to identify trends and research gaps in ESM. According to [4], three of the analysed tools are OS-ESM frameworks. The researchers classify the software tools and publications according to five predefined categories, none of which cover usability. Furthermore, Pfenninger et al. in [22] analyse and categorise ESM software of high relevance for policy-making on a national and international level, of which two tools are OS-ESM frameworks. The researchers identify four main challenges for future development of ESM software. Again, none of the identified challenges include usability. In [23], Groissböck assesses the maturity of OS-ESM software. The assessment is carried out by evaluating 81 ESM related functions, which do not encompass key usability factors. These examples show that no special measures are taken to tackle the issue of usability in OS-ESM, and instead the domain mirrors the OSS practice of focusing on functional aspects. However, what is also mirrored is the fact that this practice is contrasted by awareness about the issue of usability, as evidenced by Wiese et al. in [5], who identify *utilisation* as one of six categories of challenges related to OS-ESM software. Here, *utilisation* is a broad category encompassing many usability factors. The researchers also point out the possibility of an overlap between users and developers, specifically related to OS-ESM frameworks.

In order to assess whether usability in OS-ESM frameworks might be addressed via collaboration or direct exchange instead of research publications, we also evaluated experiences and contents from conferences and workshops of the *Open Energy Modelling Initiative* (*openmod*). The *openmod*, founded in 2014, is both, an international hub and a mouthpiece for energy modellers within academia.[1] Members meet in regular workshops or via a number of virtual channels to exchange, collaborate, and accumulate knowledge on topics related to Open Science and Energy System Modelling with the goal of improving the efficiency and quality of OS-ESM software [25]. By scanning the programs of all past *openmod* workshops as well as the forum, we again find that the topic of usability has not been addressed adequately considering its importance. We only found a call for improvement in a keynote talk on "Open Modelling for Academic Policy Advice" by Oei [26] at the *openmod* 2020 Workshop in Berlin; a discussion on its assessment in a forum thread by Hodencq [27]; and a do-a-thon session during the *openmod* 2019 Aarhus Workshop, where participants conducted usability testing in a way similar to the one we use later in this paper. To our knowledge, the results of this last instance were neither publicly documented nor empirically evaluated [28].

The lack of scientific studies and the few references to the topic within the *openmod* coupled with the special characteristics of the ESM domain indicate that usability of OS-ESM frameworks should be studied in greater depth in a scientific context. To this end we propose a specialisation of an existing usability evaluation method to the domain of ESM and apply this modification as a qualified proof of concept.

2.4. Usability evaluation methods

The domain of HCI offers different usability evaluation methods that can be applied at different stages of the development process of OSS to maintain and assess a certain level of usability. Referring to Assila et al. in [29], usability techniques provided by HCI can be categorised into (a) objective methods and (b) subjective methods.

(a) Objective methods are characterised by the fact that they collect analytical data without involving the users. For example, a number of automated metric calculations from the code base or the documentation of OSS are published by Gyimothy et al. in [13] and Samoladas et al. in [30].

(b) Subjective methods include the user's preferences and needs. Examples of subjective usability techniques are focus group surveys, (expert) interviews, and usability questionnaires. The subjective methods show that active communication involving developers and users improves the awareness of user needs and usability in general. Since Terry et al. in [31] find that OSS developers have a profound understanding of software usability techniques, and that, as most usability techniques are based on interactions, a well-established relationship between developers and users is a crucial condition for success in usability. Thus, focus group surveys seem like a good candidate for technique to improve the state of usability in OS-ESM frameworks. Focus group surveys are also applied to evaluate software usability in [12] and [16]. In the extensive literature review by Assila et al. in [29], the researchers identify 24 standardised usability questionnaires including the Post-Study System Usability Questionnaire (PSSUQ). The researchers analyse and classify the questionnaires according to, both, their key properties and usability criteria proposed by quality standards (ISO 9421-11 and ISO/WD 9241-112), and classical quality ergonomic criteria. The Post-Study System Usability Questionnaire (PSSUQ) is a widely known and widely used standardised questionnaire for the evaluation of the user interface of universal software systems. It consists of 16 usability items that can be ranked by a 7-point Lickert scale, from strongly agree (1) to strongly disagree (7). Its global reliability

[1] 42 out of 47 respondents of the Zurich meeting, that took place in 2018, were Ph.D. students, Postdocs, Staff scientists or Assistant Professors [24].

lies at 94% and it has been applied to evaluate the usability of research information systems [29].

As part of the ESMUT method, developed within this research work and described in the following section, the PSSUQ has been adapted to quantitatively assess the usability of OS-ESM frameworks.

3. The Energy System Modelling Usability Testing Method—ESMUT

Usability testing by means of the Energy System Modelling Usability Testing (ESMUT) method is designed for developers of OS-ESM frameworks who can derive concrete action from the results to further improve the usability of their frameworks. Usability testing with the ESMUT method is achieved by instructing a number of participants to perform a specific modelling task with one or more OS-ESM frameworks and to evaluate their subjective experience during the task by completing a usability questionnaire. The ESMUT method is therefore categorised as a subjective method.

Before explaining the method in detail, the roles of *instructors* and *participants* are briefly described, which are important in the application of the method. *Instructors* are those who organise, carry out and evaluate usability testing with the ESMUT method. Those who implement the model and complete the questionnaire based on their subjective experience are referred to as *participants*. Suitable participants are *users* of OS-ESM framework; they can be either *non-developer users* or *developer users*. In this study, *non-developer users* are characterised as users that usually work within energy systems (or a comparable STEM subject), have basic to expert knowledge in ESM and apply one or more OS-ESM frameworks to answer specific research questions. *Developer users* additionally develop one or more OS-ESM frameworks and usually have expert knowledge in ESM. From this point onward, OS-ESM frameworks are referred to as *frameworks* in order to increase readability.

There are two approaches to applying the ESMUT method: the **single framework approach** and the **cross-evaluation approach**. Table 1 provides an overview of the two approaches. With the single framework approach the usability of one framework is tested based on the subjective perspectives of a number of framework users. In contrast, the cross-evaluation approach is designed to cross-evaluate the usability of multiple frameworks from a developer's perspective. It enables a group of developers of different frameworks to test one another's frameworks. The cross-evaluation approach provides a systematic method to encourage the collaboration between the group of developers, in addition to evaluating the usability of the tested frameworks. With the cross-evaluation approach developer groups of different OS-ESM frameworks can jointly identify secured methods and synergies for future developments to increase the quality of the frameworks, and in the longer term, break down barriers that lead developers to develop new frameworks instead of building on the existing ones. In the following method description, we describe the single framework approach without highlighting it again in the text. Since the cross-evaluation approach is mostly based on the single-framework approach, we describe the additional steps of the cross-evaluation approach in a highlighted paragraph at the end of each subsection.

The ESMUT method consists of a step-by-step guide on how to use the method, as well as the Energy System Modelling Usability Questionnaire (ESMUQ). The step-by-step guide is divided into three steps: *Definition of the scenario and data collection (1)*, *Preparation and conduct of the modelling phase (2)*, and *Post-modelling evaluation applying the ESMUQ questionnaire (3)*. The steps are depicted in Fig. 1 and, in the following, are explained in more detail.

Definition of the scenario and data collection (1)

To specify the modelling task, first, the instructors have to define a scenario. According to Dieckhoff et al. in [32], a *scenario* consists of possible statements about the future and their justifications; *scenarios* can be analysed and reconstructed as statement systems. The scenario to be modelled during usability testing with ESMUT should be designed in a way that the participants are able to complete the modelling task in a reasonable amount of time and without extensive additional knowledge. Furthermore, it should be designed in such a way that participants have to understand and apply important basic functions and central modelling approaches of the tested frameworks during the modelling phase.

The defined scenario should include:

- a scenario narrative: a brief description of the scenario to guide participants in the interpretation of the input data set, objective function and constraints (What is it used for? What is the temporal, regional and sectoral scope?),
- a set of input data: all necessary data and units to complete the modelling task including temporal, regional, sectoral, technological, and economic parameters,
- a set of constraints: important boundary conditions, e.g. emission or investment limits, maximum uptime or downtime for energy generation units; and
- the objective function: the target function of the optimisation e.g. minimise total system cost or emissions.

To ensure that participants can successfully complete the task, they need a clear understanding of the scenario, including all parameter names and unit designations. Therefore, the instructors should prepare the scenario in a well-structured and understandable format such as the OEDatamodel [33] or the IAMC data model [34].

Cross-evaluation approach: Since a harmonised scenario forms the basis of the modelling phase, it becomes particularly relevant to agree on a common terminology to avoid misinterpretations of the parameters provided, especially because the cross-evaluation approach brings together developers of different frameworks, which most likely all use a different terminology. Since the harmonised scenario most certainly does not follow the framework-specific terminology, the probability of occurrence of misinterpretations is estimated as high. In a recent study by Booshehri et al. [35] the Open Energy Ontology (OEO) is introduced as a scientific attempt to define a common terminology within ESM.

Preparation and conduct of the modelling phase (2)

Preparation phase. The preparation phase is required to ensure a common ground to start from. If managed properly, it allows to derive additional feedback on installation and basic understanding of the tested framework.

First, the instructors need to define termination criteria for the modelling phase and communicate them clearly to the participants. Termination criteria are important to successfully finish the usability testing with a predefined start and end point. This assures that instructors and participants can plan the evaluation phase and the modelling phase, respectively. Participants also need to know when exactly they can start completing the post-modelling usability questionnaire. Suitable termination criteria, that follow a logical "OR" conjunction, are: a successful model run by the participant; multiple, repeated infeasible model runs that cannot be resolved by the participant; and a predefined time limit until the modelling task has to be completed or aborted by the participant. Compliance with the time limit can be ensured by participants sending the completed usability questionnaire (3) to the instructors. As the questionnaire is to be filled in by the participants directly after the modelling phase, the instructors can roughly rely on the participants' compliance with the time limit. This is considered appropriate as the method is more about testing usability within a certain time horizon and not about implementing a model under time pressure.

Second, the instructors need to provide installation guides for the framework to the participants, which could be done in the form of a link to the documentation of the framework. Installation guides should

Table 1
The ESMUT method: single framework versus cross-evaluation approach.

	Single framework approach	Cross-evaluation approach
Participants	Non-developer and developer users	Developer users
Characteristics	Users test a single framework	Developer users test each other's frameworks with harmonised data set
Goals	Evaluate and improve usability of frameworks	Evaluate and improve usability of frameworks
		Strengthen collaboration among developer groups (understanding basic functionalities and methodological approaches, identify improvements and synergies)
		Enable framework comparison
ESMUQ limitations	Omit questions 1.2, 1.3, 1.4, 3.5	Consider all questions

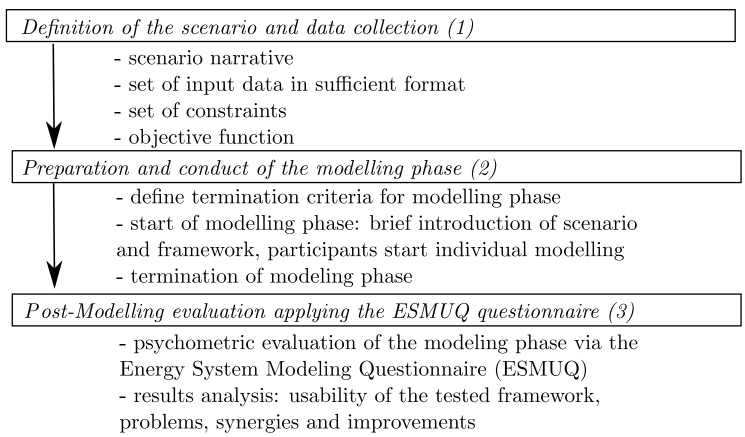

Fig. 1. Step-by-step overview of the ESMUT method.

enable the participants to install the frameworks correctly on their local machines, so that they can be set up for the modelling phase.

Third, instructors ask participants to install the framework before starting the modelling phase. If participants are not able to install the framework on their own with the provided help material, the instructors support the installation, and can directly derive necessary improvements for their help material.

Cross-evaluation approach: In case the ESMUT method is applied via the cross-evaluation approach, instructors need to have a clear scheme elaborated that allocates participants with the frameworks they are going to test. Ideally, the number of participants per tested framework is equally distributed to ensure that the exchange between the developers initiated by the cross-evaluation approach is not dominated by one or few framework(s), and further, to enable framework comparisons.

Conduct of the modelling phase. The modelling phase is the phase in which participants implement the scenario with the tested framework. The experiences that the participants make in the process form the basis for the usability evaluation in (3).

To provide initial guidance to the participants, instructors introduce the framework tested and the scenario to be implemented in a predefined amount of time. If the provided introduction and additional help material are not publicly available yet, the developers of the tested frameworks can directly derive important improvements for users that want to get started working with the framework.

Afterwards, participants start to implement the scenario defined in (1) by applying the framework. Depending on the duration and setting of the modelling phase, developers may accompany the participants during the whole session and help them by answering their questions. This can be helpful in case the participants have little to no experience with the framework itself or the programming language in which the framework has been developed. Being physically or remotely present during the modelling phase can also help the developers to identify additional assisting material and improvements to be made in the framework's documentation that would help users to get started working with the framework.

Once one of the defined termination criteria applies, participants terminate the modelling phase.

Cross-evaluation approach: In case the ESMUT method is applied via the cross-evaluation approach, the developers of framework A become participants of the usability testing of framework B or frameworks B,C,…,N, while developers of framework B become participants of the usability testing of framework A or frameworks A,C,…,N. This means, that the procedure explained for the modelling phase of a single framework is applied repeatedly by each participant for each frameworks tested in the cross-evaluation approach.

Post-modelling evaluation applying the ESMUQ questionnaire (3)

After completing the modelling phase, the participants are asked to evaluate their experiences during the modelling phase by answering the Energy System Modelling Usability Questionnaire (ESMUQ). In a next step, the results are evaluated by the instructors. In order to achieve a sound evaluation of the results, the instructors should advise the participants to formulate and explain free-text answers as precisely as possible.

The ESMUQ covers 34 questions and consists of three parts: (1) *Information about the participants*, (2) *PSSUQ adoption*, and (3) *Feedback to the framework developers*. The questionnaire is depicted in Table 2. In Part 1, the framework, which the participant has used during the

Table 2
The Energy System Modelling Usability Questionnaire (ESMUQ).

Part 1: Information about the respondent	
1.1	I have completed the modelling exercise with the following framework.
1.2	If you already have an identifier token, please enter it here!
1.3	If you do not have an identifier token yet, please choose a random 7-character string and enter it here!
1.4	You are already an expert for an energy system modelling framework. Which one?
1.5	How do you rate your programming skills? [Python]
1.6	How do you rate your programming skills? [GAMS]
1.7	How do you rate your programming skills? [C++]
1.8	Version control software [How do you rate your experience in common VCS (e.g. git)?]
1.9	How many years of modelling experience do you have?
1.10	How many years of energy system modelling experience do you have?
1.11	How have you prepared for the modelling exercise?
1.12	What is your main focus within energy system modelling?
1.13	Framework knowledge [How well did you know the specific modelling framework for energy system modelling prior to the modelling exercise?]
Part 2: PSSUQ adoption	
2.1	Overall, I am satisfied with how easy it is to use this framework.
2.2	It was simple to use this framework.
2.3	I was able to complete the task(s) and scenario(s) quickly using this framework.
2.4	I felt comfortable using this framework.
2.5	It was easy to learn to use this framework.
2.6	I believe I could become productive quickly using this framework.
2.7	The framework gave error messages that clearly told me how to fix problems.
2.8	Whenever I made a mistake using the framework, I could recover easily and quickly.
2.9	The information (such as online help, on-screen messages and other documentation) provided with this framework was clear.
2.10	It was easy to find the information needed.
2.11	The information was effective in helping me complete the task(s) and scenario(s).
2.12	The organisation of information on the system screens was clear.
2.13	The interface of this framework was pleasant.
2.14	I liked using the interface of this framework.
2.15	This framework has all the functions and capabilities I expect it to have.
2.16	Overall, I am satisfied with this framework.
Part 3: Feedback to the framework developers	
3.1	How much time in minutes did you spend for the modelling exercise until you got your final results?
3.2	Instant developer feedback (PSSUQ)
3.3	Where did you encounter problems?
3.4	How did you solve it?
3.5	Where do you see potential for synergies to your framework?

modelling phase, is specified, and (meta) information about the participant are collected. This information can be used to generate useful correlations to the usability assessment in Part 2. For example, instructors could correlate knowledge levels of ESM or a certain programming language to participant's assessment of usability and identify biases in the results. ESMUQ:Part 2 is based on the standardised PSSUQ. To be more precise with regard to our use case we have replaced all occurrences of the term *system* in the PSSUQ questionnaire with *framework*. One important thing to note is that the ESMUQ is designed with regards to the cross-evaluation approach. As a result, there are some limitations to the questionnaire when using the single framework approach. In this case, participants should omit questions 1.2, 1.3, 1.4, and 3.5. Part 3 of the ESMUQ is about giving the developers direct feedback on their experience with the framework during the modelling phase in the form of free-text answers. In this part, problems, solving strategies, and synergies are collected.

Cross-evaluation approach: After the participant has terminated the modelling phase for one of the frameworks tested, the participant is asked to complete the ESMUQ. This procedure is repeated for each framework that is tested by the participant. Identifier tokens (questions 1.2 and 1.3) are used to enable participants to answer Part 1 of the questionnaire only once and still be able to correlate information about the participant to multiple responses for multiple frameworks made by the participant, anonymously.

Regarding the evaluation of Part 2, quantitative metrics as described by Lewis [36] can be used. Results for Part 2 (questions 2.1–2.16), which map qualitative answers onto quantitative can be analysed with a quantitative method using average values and box-plots. Possible indicators can be derived by calculating the average values of questions 2.1–2.6 (System Quality), questions 2.7–2.12 (Information Quality) and 2.13–2.16 (Interface Quality). Regarding the free text answers in Part 3, visualisations such as Word-Cloud may be useful.

4. Case study: Application of the ESMUT method for five open source frameworks

4.1. Case-study design

For this case study, we applied the ESMUT method as a cross-evaluation approach to test the usability of five OS-ESM frameworks within a group of developers. The tested frameworks are Balmorel, GENeSYS-MOD, GENESYS-2, oemof, and urbs. In Table 3, a brief overview of the frameworks is provided, including information about the programming language, license, and the optimisation type. The analysed frameworks are developed in three different programming languages (GAMS, C++, Python), and are all optimisation frameworks. Balmorel and GENeSYS-MOD are based on Linear Programming (LP) while oemof and urbs are based on Mixed Integer/ Linear Programming (MI/LP). GENESYS-2 uses heuristics to solve the optimisation problem.

In applying the ESMUT method to the five chosen OS-ESM frameworks, we have pursued the following objectives:

1. understand if usability is an issue for the analysed frameworks
2. identify problems and synergies
3. intensify the collaboration across developer communities of different frameworks

Table 3
Overview of the five OS frameworks.

	Balmorel	GENeSYS-MOD	GENESYS-2	oemof	urbs
Programming language	GAMS	GAMS	C++	Python	Python
License	ISC	APL	GPL	MIT	GPL
Optimisation type	LP	LP	Heuristic	MI/LP	MI/LP

Table 4
Case study: scenario overview.

Narrative	The scenario depicts a simplified model of the electricity sector of Berlin (BE) and Brandenburg (BB) in 8784 hourly time steps. It is divided into 2 nodes, each node represents a federal state. The scenario is designed for usability testing purposes, the accuracy of the data is not validated.
Regional scope	BE, BB
Temporal scope	year: 2016, resolution: 1 h, number of time steps: 8784
Sectoral scope	electricity
Technologies	lignite, hard coal, natural gas, light oil, biomass, wind, solar, hydro
Emission limit	no emission limit
Objective	minimise total system cost
Optimisation type (s)	dispatch, investment

The developed scenario which is used for the usability testing represents a simple 2-node model of the electricity system of the German states Berlin (BE) and Brandenburg (BB) for the year 2016. Table 4 provides an overview of this scenario. The corresponding inputs are published in the OEDatamodel format and made available for further use [37].

In the case study, all participants were developer users, which means that they are the developers of one of the frameworks analysed. Accordingly, all participants have expertise in mathematical optimisation and a background in ESM. During the modelling phase, each framework developer was asked to fulfil the modelling task with two unknown frameworks. Thus, the developers of one framework became the users of the other frameworks (frameworks 1 and 4 were evaluated by three modellers, while frameworks 2, 3, and 5 were evaluated by two modellers).

We started the modelling phase during the second project meeting of the open_MODEX project. There, we first discussed the objectives, to remind the participants of the allocation between frameworks and participants, and to briefly introduce the test procedure. Afterwards, the framework developer had 30 min to introduce the framework to the participants and provide initial help. Then, the participants had 60 min to start the implementation of the scenario. This procedure was repeated once to also start the modelling phase of the second framework. After the project meeting, the participants were given around two more months to finally implement the scenarios. They were allowed to ask the developers for support at any time during the modelling phase.

After the modelling phase had finished, the participants were asked to fill in the ESMUQ which we provided as an online survey via LimeSurvey [38]. The evaluation of the questionnaires is described in the following section.

4.2. Results

Based on the average scores for a sub-set of questions of Part 2 of the ESMUQ, three indicators are calculated: (1) system quality (questions 2.1–2.6), (2) information quality (questions 2.7–2.12), and (3) interface quality (questions 2.13–2.16). Fig. 2 shows the results for the three indicators and each framework. While the values differ among the frameworks, all of them rank low to medium on the scale from one to seven. On average, system quality ranks better than interface and information quality.

The plot in Fig. 3 allows for a deeper understanding of the results. Overall, the mean value of 2.42 with a standard deviation of 0.9 for

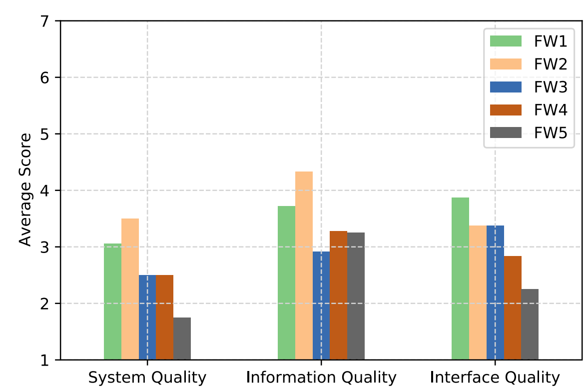

Fig. 2. Average score of each framework for the indicators system quality (2.1–2.6), information quality (2.7–2.12) and interface quality (question 2.13–2.16). Score on the *y*-axis is to be interpreted with 1 for *strongly* agree and 7 for *strongly disagree*.

question 2.16 indicates that experts were rather satisfied with using the frameworks. Data also shows that the developer users were rather comfortable using the frameworks, found it easy to learn and to use the frameworks in a productive way (questions 2.4, 2.5, 2.6). However, at the same time, answers also suggest that the interface's use and appeal was not as optimal (questions 2.13, 2.14). Most OS-ESM frameworks do not feature graphical user interfaces and modellers are therefore used to working without them. This might be the one reason why developer users found the usage of frameworks to be comfortable overall, despite comparatively poor interface ratings. The highest values with a mean of 4.6 are observed for question 2.7: "The framework gave error messages that clearly told me how to fix problems", indicating that error messages were often not helpful for the users. Similarly, mean values above 3 for questions 2.8, 2.9, an 2.10 can be observed, indicating that handling mistakes and getting useful information can be improved for some frameworks.

An overview of the developer feedback derived from ESMUQ Part 3 is provided in Table 5, which summarises the extracted keywords from the respective questions sorted by their frequency of occurrence. Problems during the application of the frameworks were associated with input-data handling, the operating system, the conversion-script, and associated debugging. Fig. 4 shows a word cloud visualisation, based on the participants' answers for encountered problems. For many participants, correct input-data processing posed the major challenge during the modelling exercise. Unclear structure, missing information

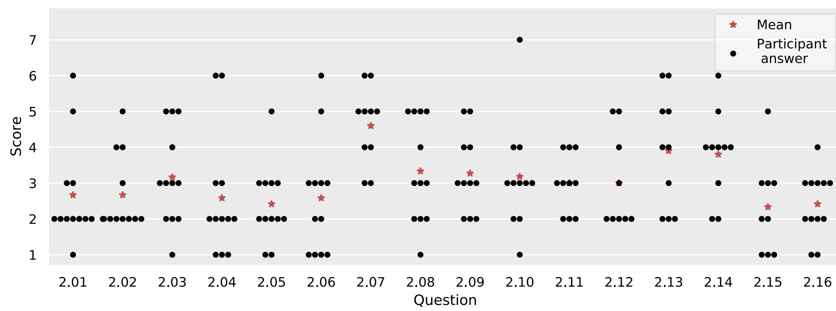

Fig. 3. Swarm plot with individual scores and mean values based on aggregated data of all frameworks displayed for each question of the ESMUQ Part 2: PSSUQ adoption. Scores are scaled from 1 (strongly agree) to 7 (strongly disagree) which means low values indicate good usability while high values represent low usability.

Table 5
Extracted keyword frequency of processed answers for questions 3.2–3.5 of ESMUQ:Part 3. An overview and definition of the keywords is provided in Table A.6.

	High (>3)	Medium (2–3)	Low (<2)
Problems	input_data(11), conversion_script(5), error_messages(4)	model_script(3), solver(2), operating_system(2), general_experience(2)	output_configuration(1)
Strengths		general_experience(3), input_data(2)	data_management(1), folder_structure(1), conversion_script(1)
Strategies	debug_error_messages(5), developer_support(4)	debug_input_data(3), trial_and_error(2), debug_conversion_script(2)	documentation(1), debug_operating_system_problems(1), user_support(1), look_at_examples(1)
Improvements		documentation(2)	error_messages(1), conversion_script(1)
Synergies		standardised_input_data_format(3), input_data(3), identify_FW_improvements(3)	cross_model_reader(1)

Fig. 4. Word cloud visualisation of processed words in answers to question 3.3: "Where did you encounter problems?".

on required parameters or unintended typing errors (e.g. white-spaces) caused errors. In addition, opaque error-messages provided little help in case of errors. Problems associated with the operating systems can be traced back to being rooted in missing knowledge of operating systems and their folder structure. However, framework cross-platform compatibility is not always given. This incompatibility can cause errors during the installation as well as during the data reading and writing processes, even if users are familiar with the operating system. This problem can emerge when all developers exclusively use one operating system and the software is not tested on other platforms.

With regard to solving strategies, in most cases, the problems were solved by debugging the error messages. Simultaneously, error messages were also listed as problems, which indicates that error messages did not supply sufficient or easily comprehensible information for the participants. In addition, developer support was sought out to solve problems. The documentation and examples were less frequently used for debugging. However, we should assume that the results are not representative of the needs of less experienced users and developer support might not be as easily accessible in standard use cases as in the usability testing procedure.

Finally, it is important to note that participants were able to identify synergies by testing the frameworks. In 4 of 8 answers on synergies, a common (standardised) input data format was named. This was described by one user as the "development of a standardised input data format and a cross model data reader, that could check for data integrity or data anomalies". The proposal of a common input data format was not restricted to frameworks of the same programming language. In three answers, no explicit synergies were named. However, participants highlighted the importance of inspirations for improvements with regard to their framework input-data structure when working with other frameworks.

4.3. Results synthesis and discussion

By conducting the case study as a cross-evaluation procedure and with eight framework developers, we find that the participants were largely satisfied with using the frameworks, and that average metrics such as system quality, information quality and interface quality were relatively positive. The results indicate good usability of the tested frameworks. However, the results show that the system quality of frameworks is higher than the information and the interface quality - a reasonable result but also important for scientific software. For the broad dissemination of OS-ESM frameworks, the latter two criteria must be improved as well. Besides finding satisfying general usability, we were able to identify problems indicating that the usability of the tested frameworks requires further improvement. *Input data, conversion script* and *error messages* are the most frequently mentioned problems the participants state when working with the frameworks. In particular, framework-specific error messages need to be considered during development with regard to more inexperienced users. Inexperience may relate not only to the modelling of the energy system, but also to the operating system or the programming language. On the upside, the results reveal that testing one-another's frameworks can benefit the tested software, as well as being essential inspiration for its own projects. *Input data* and *standardised input data* were the most frequently mentioned answers associated to synergies and show the participants‘ acknowledgement for improvement potential on the data side.

In conducting the case study, we were able to identify a few limitations when it comes to interpreting its results. First of all, all participants were developer users, which means that they are experienced in ESM, mathematical optimisation and programming. Conclusively, we should assume that the results are not representative of the needs of more inexperienced users. Second, the evaluation was carried out with an anonymisation of the frameworks analysed, as a framework-specific evaluation would not be representative due to the small number of participants per framework. If the method is carried out with a larger number of participants, a framework-specific evaluation could also be carried out to compare frameworks with each other. Third, due to the free-text answers in Part 3 of the questionnaire, we often received vague answers which made an evaluation more complicated as, for example, when providing the participants with a keyword selection of possible answers. A selection of predefined keywords to categorise the free text fields could benefit automated evaluations. For example, participants could select the predefined keyword *input_data* and give reasons or examples of why they chose this keyword in another field. Furthermore, we only examined the modelling phase for the usability, not for the fact that the scenario was implemented without errors by the participant. Theoretically, the method can be applied with an evaluation of the modelling results compared to a correct reference data set. When the evaluation is extended in this way, more information can be gained about the usability of a framework, as an essential part of ESM is model parameterisation and validation.

5. Discussion

The case study was conducted as part of the open_MODEX project, whose main goals are to improve the quality of OS-ESM frameworks and to promote collaboration between the developer groups. In addition, synergies such as a common software function or a common data management concept are to be identified and implemented. In order to achieve these goals, a profound understanding of the application and internal structure of the frameworks is crucial. The presented ESMUT method is designed to support this understanding. With the method, knowledge about the software is being generated via external feedback which can be integrated into the tested frameworks' development process and improve usability. The idea behind ESMUT is that hands-on, active engagement with software can provide a much deeper understanding than information derived from presentations and documentation. This was proven by using the ESMUT method, which allowed developers to provide informed feedback based on their experience by implementing a scenario with the tested framework. Widespread use of the ESMUT method can therefore also lead to developer communities being better able to understand and build on existing approaches. This can reduce unnecessary duplication of effort in the development process of OS-ESM frameworks in the future and strengthen the collaboration between groups of OS-ESM framework developers.

Despite the potential advantages of applying ESMUT, there are also limiting features that need to be considered. First of all, when using the method, it is essential to note that ESMUT is a subjective method. Consequently, the usability of the tested software depends on the participants‘ interpretation and how the instructors apply and moderate the method. Moreover, it depends on the participants‘ knowledge of underlying software and ESM. Hence, using this method for ranking frameworks will not be a suitable application.

Another important point with regard to the interpretation of results is the gap between the ESMUT modelling task and real-world modelling problems. Generally, as in our case study, test cases will represent a highly simplified modelling task, simply due to time restrictions. In contrast, today's real-world modelling problems are becoming increasingly complex due to the transformation of energy systems and their socio-political challenges. Nevertheless, our case study results reveal that simplified test cases already provide valuable insights. Moreover, this method can be applied to more sophisticated modelling tasks, if enough time is available.

Currently, the ESMUT method is designed for OS-ESM frameworks which leads to further limitations. In the ESMUQ Part 1, participants are asked about their knowledge of the programming languages Python, GAMS and C++. In this way, the level of knowledge of the programming languages in which the tested framework was developed can be correlated with the participant's assessment of usability. This should allow instructors to explore potential biases or differences in user requirements. If frameworks developed in a programming language other than those mentioned are tested these correlations cannot be made. However, with this limitation, the method can still be applied for usability testing of these frameworks. Furthermore, it is important to note that most of the more established OS-ESM frameworks are developed in one of the programming languages, GAMS or Python, for which the method should be directly transferable, such as PyPSA, OSeMOSys, TIMES, Calliope, etc. (see [23]).

Principally, the method is also applicable for non-OS-ESM frameworks of the same type, since the transparency of the source code is not crucial for the application of the method. However, the extent to which the usability of non-OS-ESM frameworks has already been scientifically assessed and whether ESMUT is a useful tool for such objects of study requires further research.

5.1. Future work

Although we were able to show that participants were generally satisfied with the use of the five OS-ESM frameworks, the identified problems and shortcomings suggest that their usability needs to be addressed in more detail. This leads us to the hypothesis that usability has not yet been adequately addressed for many OS-ESM frameworks. Therefore, further scientific research is required to investigate whether and how usability is considered in the development process of OS-ESM frameworks. We propose to apply the ESMUT method in the form of a comprehensive study with a larger number of participants, including developers and users, to obtain more representative results and increase heterogeneity in usability discussions.

Table A.6
Case study results: Keyword definitions.

Keyword	Definition
solver	A framework specific algorithm or external program that is used to solve the optimisation problem.
error_messages	Messages that are displayed when an error occurs during the implementation of the scenario (framework specific or programming language specific), that helps to debug the error.
general_experience	The general experience while working with the framework e.g. satisfaction, intuitiveness, etc.
model_script	The source code of the framework that describes the optimisation problem mathematically. For linear problems, this consists of set of linear equations/inequalities, and the objective function.
input_data	The data set is part of the scenario and consists of all the required parameters to successfully complete the modelling task.
conversion_script	An algorithm/script that processes the input data so that it can be read by the model script.
output_configuration	The post-processing of the optimisation results.
operating_system	The operating system of the computer e.g. Windows, Linux, Mac OS
developer_support	Help provided by the developers to solve problems during the implementation of the scenario
user_support	Help provided by other participants to solve problems during the implementation of the scenario
trial_and_error	Participants solve problems during the implementation of the scenario by trial and error.
look_at_examples	Participants solve problems during the implementation of the scenario by looking at examples/help material that is provided by the frameworks.
documentation	The written documentation of the source code and the instructions for use of the framework.
standardised_input_data_format	A data model standard that can be used by multiple frameworks.
cross_model_reader	An algorithm/script that allows a scenario to be implemented and solved by multiple frameworks.
identify_FW_improvements	Participants were able to identify improvements (modelling approaches, data processing, etc.) for their framework through the implementation of the modelling tasks with another framework.

Also, comparing the usability of OSS in ESM with scientific OSS from other domains may generate additional insights that can be used to determine which usability phenomena occur in the specific domain of ESM and which are more attributable to the OS approach.

In addition, it may be of interest to understand how large the intersection is between developers and users of each OS-ESM framework, as this may influence perceptions of usability and the factors that determine it. By drawing parallels to the usability of (scientific) OSS, it may be possible to formulate concrete usability techniques tailored to OS-ESM. These techniques may then be integrated into the development process of frameworks to systematically ensure high usability.

Finally, as part of the ESMUT method, a standard scenario for usability testing of OS-ESM frameworks can be developed, based on the Open Energy Ontology and fully complying with the Open Data principles. This would have the advantage of making the usability testing results of different OS-ESM frameworks easily accessible.

6. Conclusion

With the ESMUT method, we publish a method adapted to OS-ESM frameworks that previously did not exist. With our work, we contribute to the research community by providing a theoretical framework procedure, a questionnaire, plus datasets for usability testing of OS-ESM frameworks. Therefore, it contributes to the practical application of Open Science in energy systems research and fosters transparency, accessibility and collaboration between developer groups of OS-ESM frameworks. We show the applicability of the method through a case study which we conducted with the five OS-ESM frameworks Balmorel, GENeSYS-MOD, GENESYS-2, oemof, and urbs. The case study indicates solid usability of the tested frameworks with the potential for improvement. However, due to the limited number of participants with their relatively high level of expertise in the field of ESM, we suggest that the method has to be applied in a more extensive study to allow for a scientifically sound evaluation of the frameworks, which is beyond the scope of this work. Nevertheless, in conducting the case study, we were able to strengthen the collaboration within the developer communities of the tested frameworks. As a result, the participants developed a deeper understanding of the frameworks and their identified potential, improvements and synergies. Consequently, by using these synergies, collaboration can be further improved between the developer communities of the analysed frameworks. ESMUT proved to be a suitable method for assessing usability in OS-ESM frameworks within limits discussed above and should be further explored. Therefore, to further improve OS-ESM frameworks usability, we regard the dissemination and application of the proposed method within the ESM community as highly valuable.

CRediT authorship contribution statement

S. Berendes: Conzeptualization, Methodology, Validation, Data curation, Visualisation, Writing – original draft, Writing – review & editing, Project administration. **S. Hilpert:** Software, Validation, Formal analysis, Writing – original draft, Writing – review & editing. **S. Günther:** Methodology, Investigation, Writing – review & editing. **C. Muschner:** Data curation, Investigation, Writing – review & editing. **S. Candas:** Investigation, Data curation, Visualisation, Writing – review & editing. **K. Hainsch:** Investigation, Writing – review & editing. **J. van Ouwerkerk:** Investigation, Writing – review & editing. **S. Buchholz:** Investigation, Writing – review & editing. **M. Söthe:** Conzeptualization, Methodology, Investigation.

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.

Data availability

Datasets and source code related to this article can be found in http://dx.doi.org/10.5281/zenodo.5578291 [37].

Acknowledgements

The research for this paper was performed within the open_MODEX project supported by the German Federal Ministry of Economic Affairs and Energy under grant number 03ET4076. All figures and tables used in this manuscript were created by the authors themselves and are subject to the CC-BY4.0licence unless otherwise declared. You are free to share and adapt the figures and tables. The authors thank Ciara Dunks for proof reading, Stefanie Nguyen, Shaquille Henriques and Jonas Huber for small content and editorial contributions to the publication.

Appendix. Case study results

See Table A.6.

References

[1] Vicente-Saez R, Martinez-Fuentes C. Open science now: A systematic literature review for an integrated definition. J Bus Res 2018;88:428–36. http://dx.doi.org/10.1016/j.jbusres.2017.12.043.

[2] Huebner GM, Fell MJ, Watson NE. Improving energy research practices: guidance for transparency, reproducibility and quality. Build Cities 2021;2(1):1–20. http://dx.doi.org/10.5334/bc.67.

[3] Wilkinson MD, Dumontier M, Aalbersberg IJ, Appleton G, Axton M, Baak A, et al. The FAIR Guiding Principles for scientific data management and stewardship. Vol. 3, (no. 1). Springer Science and Business Media LLC; 2016, http://dx.doi.org/10.1038/sdata.2016.18.

[4] Open Energy Modelling Initiative. Open models, license: CC-BY 4.0. 2021, https://wiki.openmod-initiative.org/wiki/Open_Models.

[5] Wiese F, Hilpert S, Kaldemeyer C, Pleßmann G. A qualitative evaluation approach for energy system modelling frameworks. Energy, Sustain Soc 2018;8(1). http://dx.doi.org/10.1186/s13705-018-0154-3.

[6] International Standardization Organisation. ISO 9241-11 - Ergonomic requirements for office work with visual display terminals (VDTs):Part 11: guidance on usability, Berlin. 1998.

[7] International Standardization Organisation/International Electrotechnical Comission. ISO/IEC 9126. Software engineering – Product quality. ISO/IEC. 2001.

[8] Rampersad L, Blyth S, Elson E, Kuttel MM. Improving the usability of scientific software with participatory design: A new interface design for radio astronomy visualisation software. In: Proceedings of the South African Institute of Computer Scientists and Information Technologists. SAICSIT '17, New York, NY, USA: Association for Computing Machinery; 2017, http://dx.doi.org/10.1145/3129416.3129899.

[9] Queiroz F, Silva R, Miller J, Brockhauser S, Fangohr H. Good usability practices in scientific software development. CoRR 2017. arXiv:1709.00111.

[10] Macaulay C, Sloan D, Jiang X, Forbes P, Loynton S, Swedlow JR, et al. Usability and user-centered design in scientific software development. IEEE Softw 2009;26(1):96–102. http://dx.doi.org/10.1109/MS.2009.27.

[11] Raza A, Capretz LF, Ahmed F. An empirical study of open source software usability: The industrial perspective. Int J Open Source Softw Process 2011;3(1):1–16. http://dx.doi.org/10.4018/jossp.2011010101, arXiv:1511.08844.

[12] Llerena L, Rodriguez N, Llerena M, Castro JW, Acuña ST. Applying a usability technique in the open source software development process: Experiences from the trenches. In: Trends In E-Learning. InTech; 2018, http://dx.doi.org/10.5772/intechopen.74862.

[13] Gyimothy T, Ferenc R, Siket I. Empirical validation of object-oriented metrics on open source software for fault prediction. IEEE Trans Softw Eng 2005;31(10):897–910. http://dx.doi.org/10.1109/tse.2005.112.

[14] Aberdour M. Achieving quality in open-source software. IEEE Softw 2007;24(1):58–64. http://dx.doi.org/10.1109/ms.2007.2.

[15] Porter A, Yilmaz C, Memon AM, Krishna AS, Schmidt DC, Gokhale A. Techniques and processes for improving the quality and performance of open-source software. Softw Process Improv Pract 2006;11(2):163–76. http://dx.doi.org/10.1002/spip.260.

[16] Wang W, Cheng J, Guo JLC. How do open source software contributors perceive and address usability? Valued factors, practices, and challenges. CoRR 2020. arXiv:2007.06654.

[17] Nichols D, Twidale M. The usability of open source software. First Monday 2003;8(1). http://dx.doi.org/10.5210/fm.v8i1.1018.

[18] Viorres N, Xenofon P, Stavrakis M, Vlachogiannis E, Koutsabasis P, Darzentas J. Major HCI challenges for open source software adoption and development. In: Schuler D, editor. Online Communities And Social Computing. Berlin, Heidelberg: Springer Berlin Heidelberg; 2007, p. 455–64.

[19] Bødker M, Nielsen L, Orngreen RN. Enabling user centered design processes in open source communities. In: Aykin N, editor. Usability And Internationalization. HCI And Culture. Berlin, Heidelberg: Springer Berlin Heidelberg; 2007, p. 10–8.

[20] Feller J, Fitzgerald B. A Framework Analysis of the Open Source Software Development Paradigm, in: ICIS 2000 Proceedings. 7, 2000.

[21] Machado PG, Mouette D, Villanueva LD, Esparta AR, Leite BM, dos Santos EM. Energy systems modeling: Trends in research publication. Wiley Interdiscip Rev Energy Environ 2018;e333. http://dx.doi.org/10.1002/wene.333.

[22] Pfenninger S, Hawkes A, Keirstead J. Energy systems modeling for twenty-first century energy challenges. Renew Sustain Energy Rev 2014;33:74–86. http://dx.doi.org/10.1016/j.rser.2014.02.003.

[23] Groissböck M. Are open source energy system optimization tools mature enough for serious use? Renew Sustain Energy Rev 2018;102:234–48. http://dx.doi.org/10.1016/j.rser.2018.11.020.

[24] Tröndle T. Openmod-zurich-2018-feedback.csv. 2018, https://gist.github.com/timtroendle/aec2008f29b2eb4b45584c0406acd6c5.

[25] Open Energy Modelling Initiative. Manifesto, last accessed 2021-02-16, CC-BY 4.0. 2021, https://openmod-initiative.org/manifesto.html.

[26] Oei P-Y. Open modelling for academic policy advice, keynote at openmod workshop in berlin, last accessed 2022-02-14. 2020, https://next.rl-institut.de/s/9agHYX6aAgYGkL3?dir=undefined&openfile=428725.

[27] Hodencq S. Quantifying the benefits of open energy modelling? openmod forum, last accessed 2022-02-14. 2020, https://forum.openmod.org/t/quantifying-the-benefits-of-open-energy-modelling/2346.

[28] Quoilin S. Do-a-thon: running and testing open-source models, last accessed 2022-02-14. 2019.

[29] Assila A, Oliveira K, Ezzedine H. Standardized usability questionnaires: Features and quality focus. Comput Sci Inf Technol 2016;6.

[30] Samoladas I, Gousios G, Spinellis D, Stamelos I. The SQO-OSS quality model: Measurement based open source software evaluation. In: Russo B, Damiani E, Hissam S, Lundell B, Succi G, editors. Open Source Development, Communities And Quality. Boston, MA: Springer US; 2008, p. 237–48.

[31] Terry M, Kay M, Lafreniere B. Perceptions and practices of usability in the free/open source software (FoSS) community. In: Proceedings of The 28th International Conference on Human Factors in Computing Systems - CHI'10. ACM Press; 2010, http://dx.doi.org/10.1145/1753326.1753476.

[32] Dieckhoff C, Appelrat H-J, Fischedick M, Grunwald A, Höffler F, Mayer C, et al. Zur Interpretation von Energieszenarien (Schriftenreihe Energiesysteme der Zukunft). München: acatech u.a; 2014.

[33] Open Energy Platform. Open energy data model (OEDataModel). 2021, https://github.com/OpenEnergyPlatform/oedatamodel.

[34] Intergrated Assessment Modeling Consortium (IAMC). The IAMC timeseries scenario data format, website, last accessed 2021-06-23. 2021, https://pyam-iamc.readthedocs.io/en/stable/data.html.

[35] Booshehri M, Emele L, Flügel S, Förster H, Frey J, Frey U, et al. Introducing the open energy ontology: Enhancing data interpretation and interfacing in energy systems analysis. Energy and AI 2021;5:100074. http://dx.doi.org/10.1016/j.egyai.2021.100074.

[36] Lewis JR. Psychometric evaluation of the PSSUQ using data from five years of usability studies. Int. J. Hum.-Comput. Interact. 2002;14(3–4):463–88. http://dx.doi.org/10.1080/10447318.2002.9669130.

[37] Berendes S, Hilpert S, Günther S, Muschner C, Candas S, Hainsch K, et al. Evaluating the usability of open source frameworks in energy system modelling (supplementary material). 2022, http://dx.doi.org/10.5281/zenodo.5578291.

[38] Lime Survey. Lime survey website, last accessed 2022-02-14. 2020, https://www.limesurvey.org/.