Energy system modeling: public transparency, scientific reproducibility, and open development

Robbie Morrison
Schillerstraße 85, 10627 Berlin, Germany
E-mail address: robbie.morrison@posteo.de

Highlights
- growing calls for policy-based energy system models to be "opened up"
- energy system modeling projects are adopting open source development methods
- energy system database projects are co-evolving to serve datasets to open models
- source code distributed under standard copyright cannot be legally used, built, or run
- datasets distributed under standard copyright cannot be legally machine processed

Abstract
A switch to open energy system models and the establishment of open energy system databases to support these models began in earnest in 2010. As of mid-2017, such projects number about 45, up from five in 2010 and none in 2000. Three distinct yet overlapping drivers can explain this shift in paradigm: a desire for improved public transparency, the need for genuine scientific reproducibility, and a nascent experiment to see whether open source development methods can improve academic productivity and quality. This article examines these drivers, their tensions, and the central role that open software and dataset licensing plays. It also provides an audit of open energy system projects by type, license, and country.

The key message is that while public transparency can often be served by the publication of code and data under standard copyright, scientific reproducibility and open development can only be achieved through the open licensing of the associated software and datasets. These requirements necessarily extend to the public datasets from official and semi-official sources that are normally indispensable when building public policy models. Open development has the potential to improve public trust while publishing behind paywalls can hinder participation.
Introduction*

* This article does not constitute legal advice. Readers seeking such advice should consult a lawyer appropriate to their place of law.

Calls to "open up" energy system models are growing, particularly for those models used to inform public policy development (Acatech et al 2016a, Brazilian et al 2012, Cao et al 2016, DeCarolis et al 2012, Peng 2011, Pfenninger 2017, Pfenninger et al 2017, Pfenninger et al in press, Wiese et al 2014). Simultaneously, a number of energy system projects are releasing their source code under open software licenses and starting to build user and developer communities. In parallel, several open energy system database projects have been established to collect, curate, and republish the datasets needed by these models. This seismic change in practice is reviewed, together with the legal issues, mostly due to copyright, that enable and constrain these activities.

There are three distinct yet overlapping motivations for making energy system models open: improved public transparency as a reaction to sustained criticism over policy opaqueness, scientific reproducibility as a response to concerns over minimum scientific standards, and open development as an attempt to leverage the benefits that open source software development methods can offer. These three motivations can be seen as a continuum, with public transparency as the least ambitious and open development as the most. Of these, open development has the most potential to benefit science through reduced duplication of effort, better error detection and correction, and simpler collaboration within and across research fields.

While this article is aimed at energy policy models, much of what is discussed is likely to be applicable to other computational domains, including, for example, the numerical modeling of urban air quality, the simulation of economic systems, and the integrated assessment of climate protection strategies.

The legal examples provided reference either US law or German law, primarily because these two jurisdictions are responsible for most of the litigation on open licensing and consequently most of the analysis.

Some recent appeals for greater "openness" in energy system modeling (Acatech et al 2016a, Cao et al 2016, Peng 2011, Pfenninger 2017, Wiese et al 2014) have remained silent on the issue of licensing, presuming perhaps that code and data can be lawfully used once published. This is correct if and only if open licenses are provided. Otherwise standard copyright prevails and this precludes the use of both code and data beyond simple inspection (with some exceptions under German law, discussed later). This misconception is quite widespread, particularly in relation to energy datasets published on public websites in machine-readable formats. Similarly, DeCarolis et al (2012:1849) erroneously conclude that "models with open source code and data but with no license are assumed to allow redistribution without any restrictions". Code and data cannot legally be redistributed if open licenses are absent.

DeCarolis et al (2012) and Pfenninger et al (2017) do traverse the open licensing of energy system models and briefly indicate the types of license available. The latter paper additionally touches on the need to release self-authored and machine-generated datasets under open licenses to enable downstream applications, but fails to note that upstream third-party data that lacks open licensing cannot legally be read in and utilized by numerical models.
Moreover, if encumbered datasets are used by closed source projects, the input data cannot be republished in support of either public transparency or scientific reproducibility.

Indeed, only open licenses can unequivocally grant the right to study, use, improve, and distribute the associated code, data, and content — known as the four freedoms (Williams 2010:121–122). But open software licensing is as much a development model as it is a legal instrument. Open development implies that projects actively build communities by using code sharing platforms, social media channels, and other forms of engagement. Open development should be seen as aspirational, it is not a necessary condition for public transparency or scientific reproducibility.

Open data has only really become an issue for energy system studies with the advent of open modeling. Prior to that, closed source projects could purchase and use proprietary information under non-disclosure agreements (NDA). Or they could make use of publicly available copyrighted data without attracting attention. In contrast, fundamental research domains like climate modeling have long shared unencumbered code and data. But energy system models need information from official and semi-official sources, including system and market operators. These operators and their umbrella organizations have, thus far in Europe at least, been reluctant to open license their published datasets or release key system characterization information, leading to the current impasse and giving rise to crowdsourced projects to circumvent at least some of these restrictions.

Code and data are divergent in terms of reliability. Source code and documentation can be reviewed and running programs can be tested for fidelity. But assessing the quality of conventional datasets requires a knowledge of its provenance, including any cleansing and reformulation en route. Crowdsourced data brings very different challenges in terms of information integrity, mostly met though ceaseless observation and revision by the public.

There is much written about publishing code and data outputs under open licenses to advance the process of science, yet virtually nothing citable on the machine usage of published copyrighted datasets as inputs. This article seeks to clarify this latter issue as far as is possible, as well as provide literature on the use of published copyrighted source code.

Open energy system projects can be split into four distinct camps. Energy system models are modeling frameworks which cover the electricity sector and other sectors. Grid identification projects rely on crowdsourced data to create a representative model of the grid under investigation. Data portals use semantic wiki techniques or provide tailored datasets in response to custom requests. And finally energy system database projects use either relational databases or file servers to republish datasets.

This article applies three prisms to the question of open energy system modeling. The first prism looks at public transparency, scientific reproducibility, and open development. The second prism uses copyright law to examine standard copyright, open licensing, and the public domain. And the third prism considers open code and open data in light of the first two perspectives. An audit of open energy system projects then follows.

Public transparency

Public transparency is a public policy ideal which requires, at the least, that the model in question be fully documented and that the datasets used be made available for inspection, but neither necessarily under open licenses. Some authors prefer to term the headline concept comprehensibility rather than transparency (Cao et al 2016:2). The qualifier public is used to exclude other less onerous forms of transparency, such as providing peer reviewers with supplementary material.

Acatech et al (2016a:16–17) suggest that public transparency is best served with layered publishing, ranging from policymaker summaries to technical reports in sufficient detail to enable the results to be replicated. Cao et al (2016:4) “consider open source approaches to be an extreme case of transparency that does not automatically facilitate the comprehensibility of studies for policy advice”. While that may be true, open development can also improve transparency. Vibrant open source projects normally produce good documentation, if only to meet their own internal needs. Wiese et al (2014) argue that the public trust needed to underpin a rapid transition to zero carbon energy systems can only be built through the use of transparent open source energy models. Opaque policy models simply engender distrust. Strachan et al (2016:2) opine that closed energy models providing public policy support “fall far short of best practice in software development and are inconsistent with the open access movement [and] publicly funded research”. The Deep Decarbonization Pathways Project (DDPP) seeks to improve its modeling methodologies, a key motivation being “the intertwined goals of transparency, communicability and policy credibility” (Pye and Bataille 2016:527).

The oft heard call that models should publish their equations needs some examination. Mathematical programs (written in algebraic modeling languages like MathProg) can list their equations over some tens of pages because their codebase is essentially the programmatic expression of these equations (Howells et al 2011:5835–5836). But a sophisticated simulation/optimization framework (written in say C++ or Python) may need hundreds of pages to adequately record its workings. For instance, the core of deeco is documented in a 145 page PhD report (Bruckner 1997) and a 239 page user manual (Bruckner 2001), with later enhancements adding proportionately to this material.

0. MathProg is an open language that forms part of the GNU GLPK project and supports a subset of the proprietary AMPL language.

Allied to the notion of public transparency is that of market transparency. Market transparency measures include the 2013 European energy market transparency regulation 543/2013 (European Union 2013). This measure is intended
to improve market liquidity and system security and also the standing of minor players. The regulation requires transmission system operators, wholesale market operators, and their umbrella organizations to collect, aggregate in some cases, and make public energy market and system reliability data. The machine use of this data, but not its republication, is enabled by 543/2013. However, these datasets are regularly offered under legal notices that do not respect this requirement. This situation needs to be resolved. Irrespective, the inability to freely redistribute original datasets places a serious constraint on public transparency. The original datasets provided under the regulation need only be made available for five years and can then go dark.

Paywalled articles present a significant barrier to public transparency. Open access publishing can resolve this problem by assigning one of the Creative Commons licenses. Academic publishing houses will levy a substantial article processing charge (APC) for this privilege. Under the European Commission Horizon 2020 research funding programme, open access publishing should become routine (European Commission 2017). Free-of-charge provision of material under standard copyright and open access publishing under an open content license are distinct concepts with quite different objectives and attributes.

**Scientific reproducibility**

Replication is the "ultimate standard by which scientific claims are judged" (Peng 2011:1226). Replication, in the context of energy system modeling, would mean reimplementing the software and collecting the input data anew. As a consequence, replication in the computational sciences is rarely feasible, so *reproducibility* represents the attainable minimum standard. Reproducibility means taking the existing code and data and repeating the analysis. Even so, reproducibility is no guarantee that the results are correct and the conclusions valid.

The reproducibility spectrum ranges from making only the source code available, at one end, to providing the code, data, and executables, at the other (Peng 2011:1226). Ince et al (2012) argue that code must be published for reasons of reproducibility, but again remain silent on the question of open licensing. Independent researchers must be legally free to experiment with the code and data in order to examine the behavior of the models under investigation. Ayer et al (2017) describe a research data infrastructure under development to support large-scale scientific reproducibility. Some practitioners believe reproducibility requires a step change in scientific culture, covering research practices, reward structures, funding policies, publishing norms, and the release of relevant code and data (Stodden et al 2013).

In the context of energy system modeling, DeCarolis et al (2012) argue that repeatable analysis can only be achieved when the source code and datasets are jointly placed under publicly accessible version control so that independent researchers can select, run, and check specific model instances. The right to inspect, use, modify, and republish the code and data are the fundamental conditions for scientific reproducibility. Only open licensing can provide these conditions.

**Open development**

The third motivation is open development. Open development is shorthand for the use of internet-mediated open source development techniques and practices. Key attributes of open projects include: unrestricted participation, status through contribution, extreme transparency, an emphasis on consensus with voting as a last resort, and minimal but sufficient governance (Red Hat 2009). Open development has its roots in the free software movement, which has produced the GNU/Linux system, the GNU GCC compiler collection, the Apache webserver, the Firefox browser, to name but some.

Open development forms a part of the nascent collaborative commons, a term coined by Rivkin (2014) which is mostly digital in nature and enabled by internet technologies. Raworth (2017) predicts an increasing role for this new sector, placing it alongside the state and the conventional economy in terms of importance in the sustainability age.

The relationship between open development and the scientific method is an interesting one. Eric Raymond (2001) attributes the success of complex open source software projects, like the Linux kernel, to the "massive independent peer review" process that accompanies such projects (quote from Moore 2001). Notwithstanding, the degree to which open source development practices can contribute to the computational sciences remains largely unknown.

Some of the characteristics (and excitement) of open development are captured by Linux kernel developer Greg Kroah-Hartman recounting his first experiences of submitting code (Bhartiya 2016:14):

> I wrote a driver over the weekend and submitted it, and I swear within an hour people came back pointing out problems and telling me: This is wrong; this is wrong; this is wrong. It felt awesome. They were critiquing my code, and I was learning from it, so I said 'Yes, you are right. This is wrong, this is wrong, and this is wrong.' I iterated and fixed problems with it. It got accepted into the kernel. It was fun. I think feedback is very important. That feedback loop of people pointing out errors or problems with what you're doing is very traditional. I guess [that is the] scientific method. And I love it. That's how we get better.

Open development can help promote public transparency and build trust. Open source software projects have traditionally been adept at engaging newcomers (as above), whether for recruitment or to extend their userbase. The OSeMOSYS project is clearly the most advanced in this regard in the open energy modeling policy domain, using a range of channels to communicate with developers and users and with a wider energy policy audience (Howells et al 2011, Pfenninger et al in press).
Open development can offer another virtue. It is sometimes thought that closed energy system models run by particular research institutes are designed, calibrated, and run to produce certain results, particularly in relation to the future ranking of technologies. Whether true or not, open development will naturally encompass a range of views that can help to identify and reduce such bias. Energy scenarios and energy models, while useful, have clear limitations that should be discussed and debated candidly (Bruckner 2016, Dieckhoff and Leuschner 2016).

Fig 0 shows the open modeling pipeline. Data from official and semi-official sources and collected by the public enters on the left. Code development occurs in the middle. Scenarios, defined in part through public engagement and citizen sourcing, enable specific models to be formulated and run. Interpretation, followed by scientific and gray publishing and outreach, takes place on the right.

An open platform is not essential for public transparency or scientific reproducibility, but open source projects tend to avoid proprietary languages and environments. In the context of the diagram, a toolchain describes the set of programming tools and system libraries required to build software in a particular language. Open projects again tend to favor Linux, in part because of its suitability for software development. The preferred open languages for energy system projects are, in rough order of popularity: Python, Java, MathProg, C++, R, and Ruby.

The most commonly encountered closed platform for energy modeling is the GAMS language and IDE. The significant cost of GAMS (upwards of USD 3200) effectively limits participation to those who can access an institutional copy. Some open energy system modeling projects are planning to migrate from GAMS to MathProg or even Python in order to increase their potential communities. The success of OSeMOSYS can be attributed in part to its choice of MathProg (Howells et al 2011:5854).

0. The GAMS Development Corporation is planning to open source its C++ API in July 2017, which may make the language more attractive to public open source projects. update as events unfold

Future energy system scenarios should be developed using public consultation. There is a large body of law on public consultation adequacy, but the topic falls outside the scope of this article. Public outreach is increasingly seen as important activity for scientists (Woolston 2015).

While not an open licensing issue directly, the choice of distribution channel may have a significant influence on the course of a project. Common methods include public code hosting sites, institutional git servers, websites providing tar files, and email on request schemes. Some projects require users to register first while others are anonymous. Yet others require individual approval by a project administrator. The git revision control system and GitHub have certainly caught the imagination of the scientific modeling community in a way that previous code hosting sites, such as SourceForge, have not (Ram 2013). Even so, around 80% of the public repositories on GitHub fail to include a license of any description (GitHub 2015).

0. GitHub is a web-based code hosting site located in the US and built on the git distributed revision control system. Other code hosting examples include GNU Savannah, SourceForge, and GitLab.

Projects designed to be open from the outset can be better structured and documented for their eventual release. The team can agree to write to a coding standard. They can select a software license with due consideration and ensure that only compliant code and open data are used. Legacy projects wishing to open up retrospectively may find it hard to identify and locate contributors and obtain consents for the new arrangements. If permissions are not forthcoming, then the associated contributions will need to be removed or reimplemented. Commercial datasets will likewise need to be substituted by open equivalents.
Copyright law

The legal context is important because much of what can and cannot be done with code, data, and content is governed by copyright law. Copyright law grants the author of an original work a time-limited exclusive right to its use and distribution. The law is intended to support preferential exploitation and thereby incentivize creative activity (in stark contrast to the open development ethos). Copyright can only protect an original expression of ideas, not the underlying ideas themselves. Facts themselves cannot be copyrighted, but their "collection, aggregation, analysis, and interpretation" may be if these actions can be considered creative (Kitzes et al. 2017:86). Multiple authors are permitted and copyrights may be assigned to an institution or other legal person. Copyright infringement is primarily a civil matter.

To complicate matters, the interpretation of copyright law depends on whether the target is, in this context, code, data, or content. Copyright law was not developed with either source code or machine-readable data in mind. Indeed software only became eligible for copyright in the US in 1974, with subsequent judgments confirming this view. The relationship between machine-readable data and copyright is in its infancy.

The term standard copyright, also referred to as unconditional copyright, is used here to indicate that the copyright holder has not stipulated additional conditions. Standard copyright is the default under law, even when no claim for copyright is explicitly made. Open licenses add conditions that allow for downstream use (beyond simple inspection), modification, and redistribution.

There are several phrases used to describe the dispersal of copyrighted material. This article generally adopts the term distribute, other roughly equivalent language includes publish, make available, and communicate. These various terms arise from the definitions written into different national laws and EU directives. Meeker (2017:71–83) provides an extended discussion of what constitutes distribution in relation to software under copyright. New computer practices, such as web-based remote execution, also known as software as a service (SaaS), further complicate this matter.

As noted, this article draws on US law and German law. The US statute is available online (Legal Information Institute). The German version is the Urheberrechtsgesetz (UrhG), which translates as the Act on Copyright and Related Rights. An official translation is available (Juris 2017). The UrhG provides more rights for authors than does its US counterpart.

0. This particular version does not include revisions to the law made on 30 June 2017. These changes mostly address text and data mining and are not especially relevant here. check for update in due course

The question of whether the individuals who produced the material, be it code, data, or documentation, or their host institution holds the copyright is not covered here. The situation varies between country, funder policy, contributor status, and any terms of employment. See instead Morin et al. (2012) and Pfenninger et al. (in press).

Open licenses add conditions to a copyright that allow the work to remain open. All open software licenses grant the user free use of the software. Permissive licenses require attribution if the software is distributed, while copyleft licenses additionally contain measures to prevent capture (covered shortly). A license terminates if any of its conditions are violated. Open licenses and public domain dedications invariably carry a warranty disclaimer so that use of the program, data, or content is at your own risk. Morin et al. (2012) discuss the open licensing of scientific software in general, but not scientific data. Table 0 lists some commonly encountered licenses, based on family and target.

0. In relation to the open licensing of code, readers fluent in German are referred to Jaeger and Metzger (2016) regarding German and European law. Meeker (2017) provides an excellent treatment in English with a focus on US law, but also reviews the case law developing internationally, including in Germany. Neither work traverses open data.

<table>
<thead>
<tr>
<th>License family</th>
<th>Code</th>
<th>Data</th>
<th>Content</th>
</tr>
</thead>
<tbody>
<tr>
<td>Copyleft licenses</td>
<td>LGPLv3, GPLv2, GPLv3, AGPLv3</td>
<td>ODbL</td>
<td>CC BY-SA 4.0, CC BY-SA 4.0</td>
</tr>
<tr>
<td></td>
<td>Eclipse, Mozilla 2.0, CDDL, EUPL 1.2</td>
<td>GFDLv1.3, CC BY-SA 4.0</td>
<td></td>
</tr>
<tr>
<td>Permissive licenses</td>
<td>Apache 2.0, BSD (3 clause), MIT, ISC</td>
<td>ODC-By</td>
<td>CC BY 4.0, CC BY-NC 4.0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>CC BY 4.0</td>
</tr>
<tr>
<td>Public domain dedications</td>
<td>CC0 1.0</td>
<td>PDDL 1.0</td>
<td>CC0 1.0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>CC0 1.0</td>
<td></td>
</tr>
</tbody>
</table>

Abbreviations: AGPL, GNU Affero General Public License • BSD, Berkeley Software Distribution License • CC BY, Creative Commons Attribution license • CC BY-SA, Creative Commons Attribution ShareAlike license • CC0, Creative Commons Zero universal public domain dedication • CDDL, Common Development and Distribution License • EUPL, European Union Public License • GFDL, GNU Free Documentation License • GPL, GNU General Public License • ISC, Internet Systems Consortium license • LGPL, GNU Lesser General Public License • MIT, Massachusetts Institute of Technology license • Mozilla or MPL, Mozilla Public License • ODC-By, Open Data Commons Attribution license • ODbL, Open Database License • PDDL, Public Domain Dedication License

Table 0. A selection of commonly used open licenses and public domain dedications, based on family and target, with version numbers given where relevant. While projects often adopt the latest release, the 1991 GPLv2 is still in wide use, even on new projects.

0. The convention of abbreviating versions 2.0 and 3.0 of the GNU GPL as GPLv2 and GPLv3 respectively is retained here. Similarly for other GNU licenses.

Open licenses differ from their proprietary counterparts in that they are not negotiated on a case-by-case basis, nor is any license fee transacted. Indeed, no contact between the user and the copyright holder is required. Open licenses are non-discriminatory by definition, which means that no application domain, including commercial usage,
can be legally excluded.0

0. The Creative Commons suite of licenses does offer a noncommercial (NC) provision but this qualifier is not widely used or necessarily recommended. Nor is it strictly an open license. The boundary between commercial and noncommercial usage can be difficult to establish in legal terms.

Standard copyright

As noted, standard copyright is the default state if no license is specified. Standard copyright precludes both the use and further distribution of code, data, and content outside of a few narrow exceptions. That said, there is no restriction on inspecting the code or data if legally obtained, usually by anonymous download from a public code hosting site, file server, or website.

Under US law, source code which lacks an open license notice cannot be legally built and run. Meeker (2017:148) states in relation to GitHub (emphasis added): "unfortunately, if no licensing terms are applied, the default is that no rights are granted to use the software". Use, in this case, would include building (compiling and linking, interpreting, or translating) the source code and running (executing or solving) the resulting program or program instance. Use would also cover pasting source code into an existing codebase. German copyright law (UrhG § 69c) provides a definitive list on how software under standard copyright may be used and this provision prohibits the usage just indicated (Jaeger and Metzger 2016:129, Juris 2017:29-30).0

0. Jaeger and Metzger write (translated, emphasis added): "The UrhG § 69c (1) assumes a broad concept for copying which includes not only a permanent copy on a storage medium, but also the temporary loading into main (RAM) memory or processor cache. This leads to the conclusion that a copyright authorization [meaning license] is required for the mere use of a piece of software. Thus, the construction of the UrhG § 69a and following sections differs from classical copyright. Anyone who uses an analog work as intended does not require permission from the author and in particular no rights of use: reading a novel, listening to music, or viewing a work of fine art is not a process which can be prohibited by exclusive copyrights."

The machine processing of a dataset under standard copyright is less clear but follows similar reasoning. additional analysis being sought from the US and Germany

The US legal doctrine of fair use applies the United States, permits minor usage for the purposes of public review and similar. Fair use is not supported under German law, but a number of use cases are exempted in the UrhG, such as school projects and some forms of scientific research. The notion of fair use does not sensibly apply to source code, beyond the quoting of small chunks in written publications.

Copyleft licenses

Copyright law was innovatively stood on its head with the release of the GNU General Public License version 1.0 in February 1989 by programmer and software activist Richard Stallman (Casad 2017). As Meeker (2017:96) remarks, the "GPL is a kind of constitution for the free software world". The GPL classifies as a copyleft license (Kuhn et al 2015). Copyleft licenses are designed primarily to avoid code capture or enclosure. Enclosure is the practice of privatizing common property and is used here to describe source code that was originally public being incorporated into closed source programs without improvements being revealed. Copyleft licenses prevent this process, while permissive licenses permit it. Meeker (2017:8) would prefer this family of licenses be termed hereditary, but her suggestion never caught.

There are several grades of copyleft (Meeker 2017:34). Weaker copyleft (LGPL) allows open libraries to be linked to by proprietary applications, for instance. Ultra-strong copyleft (AGPL) prohibits the remote execution of open software over a network without also making the source code available. This is increasingly relevant in the context of software as a service (SaaS) architectures using browser-based thin clients. Strong copyleft (GPLv2, GPLv3) fits between these two. And weak copyleft (Eclipse, Mozilla 2.0, CDDL, EUPL 1.2) sits beneath weaker copyleft because its allows any kind of code integration as long as the copyleft code remains in separate files.

An orthogonal issue is that of the handling of software patents (Meeker 2017:153-182). Code licenses vary in their response to patent grants, although most will terminate defensively on certain types of patent claim. These issues are not traversed here because it is unlikely (in the author's view) that current or future energy system models will encounter such problems.

The copyleft software licenses were followed by similar licenses for content and then data. The Creative Commons CC BY-SA (Attribution ShareAlike) set of licenses are the best known, with version 4.0 designed for data as well. The ODbL is the most widely used copyleft license specifically crafted for data.

Most open licenses are now international and intentionally silent on the choice-of-law (in contrast to their proprietary counterparts).0 This means that litigants are free to select the country and legal system under which they seek redress. As a result, Germany has become, more or less, the jurisdiction of choice for GPL infringement claims (Jaeger 2010:37, Meeker 2017:234, 244). Such litigation is invariably aimed at enforcement and not relief (Jaeger 2010:36). Claims involving permissive software licenses are rare because the license requirements are so lax. That said, incorrect attribution in other domains, like web publishing, can result in legal action.

0. One notable exception is the EUPL, which specifies that European law and courts are to be used unless otherwise agreed.

GPL licensed code can be built and run using proprietary tools, given that the resulting program does not combine with non-GPL-compliant components. An exception is made for the system libraries that ship with proprietary operating systems (GNU 2017). GAMS code can be licensed under the GPL.
Permissive licenses

Permissive software licenses allow the user to do whatever they wish with the work, requiring only that they accept a warranty disclaimer and acknowledge the contributors if they elect to distribute their software, be it as source code or an executable at their choice. The Free Software Foundation (FSF) recommends the phrase permissive license, although the terms attribution license and non-copyleft license are also used. The Creative Commons CC BY (Attribution) set of licenses are the most common permissive licenses for content, with version 4.0 designed for data as well.

Related matters

Four further matters, not technically part of copyright law, deserve coverage: public domain dedications, database rights, contribution agreements, and server location.

Works residing in the public domain no longer carry exclusive intellectual property rights. These rights might have expired, been forfeited, been expressly waived, or were never applicable. The concept of public domain is a US legal doctrine which does not have an equivalent in Germany and other countries using civil law. Hence the PDDL 1.0 and CC0 1.0 public domain dedications fall back to maximally permissive copyright licenses in these jurisdictions (Kreutzer 2011:5).

Under US copyright law (17 USC § 105), scientific software and data (among other works) produced (as opposed to contracted) by the US federal government are public domain within the confines of the United States. The US government can and does assert copyright to these works in third countries, in accordance with local copyright legislation and established practices (Klein and Hodge 2008:3.1.7). The most visible example of US public domain energy policy software is the National Energy Modeling System (NEMS), which, while freely available and unrestricted in use, makes no attempt to create a community (Pfenninger 2017:393). The US Department of Energy OpenEI energy database project, in contrast, serves federal government datasets under an internationally recognized CC0 public domain dedication (Brodt-Giles 2012).

0. More specifically, 17 USC § 101 states "a work prepared by an officer or employee of the United States Government as part of that person’s official duties".

Although public domain dedications are often made for trivial programs and code snippets, they are rarely used by substantive open software projects. Public domain dedications are however promoted for scientific data because of the flexibility they offer in relation to reuse (Stodden 2009:42).

Another intellectual property right related to, but distinct from, copyright are database rights. A database right protects the "substantial investment" incurred in assembling a public database, but not the individual datasets, which themselves need not reside under copyright (European Commission 1996, Wu 2002). Database rights do not exist in the US because the US Constitutional prevents the protection of uncreative collections (Merges 2000). To infringe in Europe, a substantial part of a database must be downloaded, reconstructed, and used in a way that conflicts with the interests of the original database maker. Under German law (UrhG § 87c), exceptions are made for private use and for personal scientific use. Other legal jurisdictions do not offer this exemption.

0. More specifically, § 87c (2) states that a “substantial part of a database” may be reproduced and used “for personal scientific use if and insofar as the reproduction is justified for that purpose and the scientific use does not serve commercial purposes” (Juris 2017:36).

Database rights are not much considered here because energy modeling projects are unlikely to transgress. Database rights do however apply to open energy system database projects when their servers are located within Europe. But none have expressly waived this right. Notwithstanding, one project (OEP, with its server in Magdeburg, Germany) does allow an entire set of relational tables to be downloaded as a single request. Third-party database rights apply when stocking third-party databases from official and semi-official European sources. That matter is not pursued here because it is relevant to only a very few projects (OPSD).

Contribution agreements are used to grant rights from contributors to the project itself (Meeker 2017:196-197). Such agreements are normally restricted to projects under copyleft licensing and typically provide the flexibility to upgrade to a newer license or to relicense under less restrictive conditions. The FSF employs contribution agreements for all its projects, but the practice is not common. No open energy system project to date uses a contribution agreement.

The location of the primary server can be significant for copyright claims involving the illegal distribution of content (Meeker 2017:239–240). But claims concerning open license compliance may be brought in most jurisdictions.

License adoption

Table 1 shows the adoption of open software licenses by open energy modeling projects. Very little is known as to how and why scientific modeling projects choose open licenses. The breakdown between copyleft and permissive licenses is evenly split, with the GPLv3, Apache 2.0, and MIT licenses being the most popular. Creative Commons and other non-software licenses should not be used for source code, because only software licenses contain provisions covering technical matters like linking.

<table>
<thead>
<tr>
<th>License family</th>
<th>License</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>Copyleft licenses</td>
<td>Ultra-strong</td>
<td>AGPLv3</td>
</tr>
<tr>
<td>License Type</td>
<td>License</td>
<td>Count</td>
</tr>
<tr>
<td>--------------</td>
<td>---------</td>
<td>-------</td>
</tr>
<tr>
<td>Strong</td>
<td>GPLv3</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>GPLv2</td>
<td>3</td>
</tr>
<tr>
<td>Weaker</td>
<td>LGPLv3</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>LGPL2.1</td>
<td>1</td>
</tr>
<tr>
<td>Weak</td>
<td>Eclipse</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>Mozilla 2.0</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>CDDL</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>EUPL 1.1</td>
<td>1</td>
</tr>
<tr>
<td>Subtotal</td>
<td></td>
<td>13</td>
</tr>
<tr>
<td>Permissive licenses</td>
<td>Apache 2.0</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>BSD (3 clause)</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>MIT</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>ISC</td>
<td>1</td>
</tr>
<tr>
<td>Subtotal</td>
<td></td>
<td>13</td>
</tr>
<tr>
<td>Public domain dedications</td>
<td>CC0 1.0</td>
<td>0</td>
</tr>
<tr>
<td>Subtotal</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>Non-software licenses (not recommended)</td>
<td>CC BY-SA 3.0</td>
<td>2</td>
</tr>
<tr>
<td>Subtotal</td>
<td></td>
<td>2</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td></td>
<td><strong>28</strong></td>
</tr>
</tbody>
</table>

Table 0. Software license counts for open energy system modeling projects. Data processing scripts are not included in the tally. See table 0 for a list of the projects surveyed. [Source: Own assessment]

A license notice must be added to the codebase, dataset (as metadata if possible), or document in accordance with the particular license type. Permissive software licenses are simpler in this regard, requiring only a single standard text file in the top level directory. Meeker (2017:148) notes that this is not always done, even when the license type is announced on the project web page. Readers need to be alert to this possibility.

**Open code**

This section starts with definitions. The term *code*, in this article, refers to text-based source code, whether written in a compiled language (like C++), an interpreted language (like Python), or a translated language (like MathProg). The term covers simple one page scripts to complex codebases comprising tens of thousands of lines. An *executable* is a standalone file produced ahead-of-time, which can then be distributed and run on a target system without the original source. The more general term *binary* is used here to cover both executables and compiled libraries. The term *library* covers header-based libraries, interpreted language modules, both text and bytecode, and compiled libraries. The term *software* covers all of the preceding.

In terms of code, the choice between copyleft and permissive licensing may ultimately be one of capture versus adoption. Copyleft licenses prevent capture while permissive licenses encourage adoption. Casad (2017:17) cites the example of BSD Unix and Linux. Unix was able to flourish under the permissive BSD license, thereby providing the context for Linux, which, soon after its inception, swapped to the GPLv2 license in 1991. This new license helped keep the Linux project cohesive and focused, something that the BSD Unix family had lacked. BSD Unix was also the forerunner for the proprietary macOS operating system, a clear example of enclosure. Casad (2017:17) surmises that:

The GPL lends itself to large projects that keep the community working together on a single code base. Permissive licenses are better suited for smaller, collaborative projects that serve as a core or incubator for a larger ecosystem that might include proprietary implementations.

Fig 0 depicts the development and distribution architecture for a typical open energy system modeling project utilizing the git revision control system. The inbound and outbound open licensing conditions under primary and secondary distribution are indicated. The inbound conditions apply when one receives the source code and the outbound conditions apply when one further distributes the source code, executables, or both. In legal terms, a *local fork* constitutes a derived work. The inbound licensing conditions are identical for copyleft and permissive licenses upon primary distribution, an important fact. The user is responsible for ensuring that all third-party dependencies, including libraries, are met locally. This tends to be more of an issue for compiled languages (like C++) than interpreted languages (like Python) which usually provide excellent package management.

A push call by an independent developer results in local modifications being uploaded to the main repository as a
development branch. This is normally followed by a pull request, upon receipt of which the project maintainer solicits testing and discussion and then, if successful, merges the submitted changes into the mainline. When one contributes code in this manner, one implicitly consents to the current licensing arrangements and simultaneously becomes a joint copyright holder. Indeed, GitHub reinforces this arrangement under its terms of service. Downstream clusters may form, perhaps mapped to individual research groups or to sets of developers working on new functionality. The Linux kernel project uses secondary repositories to manage each of its subsystems.

![](image)

**Fig 0.** A typical development and distribution architecture for an open energy system modeling project, using GitHub or an equivalent institutional git server. The diagram assumes that the primary distribution is source code alone. The outbound open licensing conditions only trip when the source code, executables, or both are further distributed. A copyleft license requires that the source code accompany an executable, whereas a permissive license does not.

The secondary distribution of executables without source code, described earlier as capture and legal only under permissive licensing, is (in the author’s view) unlikely to be a common occurrence for open energy system projects, or at least those aimed at public policy. In all probability, the independent developer will be from a university, research institute, specialist consultancy, in-house corporate team, or public agency. Given the specialized nature of energy modeling, none are likely to have much incentive to develop and distribute software in their own right. They should be rather more inclined to push their improvements back upstream for scrutiny, testing, and adoption by the wider community. Hence, the many pros and cons concerning copyleft and permissive licensing may largely be irrelevant for typical open energy system modeling projects.

There are three provisos though. First, a project with a copyleft license can incorporate code and libraries with equal or weaker copyleft licensing, whereas a project with a permissive license cannot. The selection of a suitable GPL license can therefore present developers with a greater range of specialist libraries, including the many mature solvers and maths libraries licensed under the GPL. This consideration is more significant for compiled languages. Second, a project with a permissive license cannot simply copy in or include code from a copyleft source. The foreign code must be reimplemented, using the original design if appropriate. Neither can the project link to a library with a copyleft license. Third, the AGPL is the only license that requires that the source code be released when executing the software across a network, including the internet. Web-based execution has already been adopted by some projects as their mode of delivery to users (the Energy Transition Model uses the MIT permissive license). The AGPL should therefore be used by projects wishing to remain open under SaaS architectures (Wienholt 2017:5).

Contributors who do not wish to see their efforts later captured should select projects employing copyleft licensing.

0. Notwithstanding, GPLv2 code and libraries must contain the phrase "or any later version" before they can be used in GPLv3 projects.

Projects can assign parts of their codebase to different but compatible open licenses (say GPLv3 and Apache 2.0) to service different downstream use cases. If all contributors agree, projects are also at liberty to dual license or sell proprietary licenses for use in commercial closed source products (Meeker 2017:193).

In terms of advice, Meeker (2017:197) suggests that “if a project is unsure about which license to use, it is best to start with a more restrictive license and move to a more permissive license if it becomes evident that more permissiveness will serve the project better.” It may also be useful to solicit contribution agreements from developers who might later lose contact with the project. Ideally, a project should select its software license after reviewing the license compatibilities of the solvers, numerical libraries, and other third-party software components it plans to use or might possibly employ.
The data processing scripts used in open energy system database projects often carry permissive licenses. This is a reasonable position, given that most of these scripts are not projects in any sense. In contrast, the code developed by grid identification and data portal projects can be very substantial.

There can be legitimate misgivings when opening up an existing project, particularly one hosted by a research institute. The first concerns the intellectual and financial investment in the project to date. Whether to regard that investment as sunk or not is a matter for each team. The second concerns academic reputation. The open source mantra of "release early, release often" (Raymond 2001:28) does not readily apply in this case and research teams may instead want a degree of finish before putting their codebase and datasets on public view. The third is a belief that providing support will stretch team resources. Experience suggests that although email traffic will increase, the external contributions can easily outweigh this overhead (Pfenninger et al in press). That said, there is no formal requirement to support open source software and data once released.

Researchers from the climate and ocean modeling communities say that internal and external pressure, including public and media scrutiny, forced them to progressively open up their models and data over the last decade. Furthermore, there was a rationalization of models and a consolidation of effort. Whether this same dynamic will develop in the more diverse energy system modeling domain remains to be seen.

Energy system models can no longer be sensibly implemented and run as part of one PhD project. Models now require a level of detail and complexity that is beyond one person. If software is to be developed by masters and doctoral students, then a clear separation from the wider project may be advisable. This can be easily managed under git by creating a local research branch, while periodically merging improvements from mainline. Open development methods should also produce a better documentation trail for upcoming students than does current practice.

More generally, software developed collectively within an academic context may have to traverse issues that non-academic open source projects do not. Such issues may include internal and intergroup rivalry, the ownership and use of intellectual property such as software names and logos, and project continuity as research projects arise and expire. Academic norms will also apply. For instance, a failure to cite the author of some public domain code does not contravene any legal rights, but might class as plagiarism.

Open development also invites better software engineering practices: coding standards, code commenting, revision control, runtime logging, memory proofing, unit testing, scripted test suites, code reviews, and software and user documentation. These techniques are being taken up by at least some open energy system projects (Pfenninger et al in press).

Open data

This section again starts with definitions. The term data refers here primarily to machine-readable datasets. Such datasets may also be human readable if text encoded and suitably structured. But ultimately these datasets are intended to be machine processed, meaning read into memory by a computer program, cast to native data types, and then manipulated programmatically using integer and (IEEE 754) floating point arithmetic to derive useful output.

Energy system datasets can originate from official and semi-official bodies or be crowdsourced by the public using web-based projects. Although not common, data can also be scraped from PDF documents and websites. Open formats, whether text or binary encoded, tend to be preferred for reasons of portability. Examples of energy system datasets include asset inventories (constituted as tables) and time-series covering demand, weather, and market conditions (constituted as arrays). Locational information is central to most energy datasets and GIS-based management and interpretation is a growing activity. The crowdsourcing of data is part of the emerging open collaborative research movement, also known as citizen science (Franzoni and Sauermann 2014). Crowdsourcing and open development share a common ethos.

Open data warrants special consideration. Systems modeling today is as much about assembling data as it is about authoring code. But while code licensing and software development practices are quite well resolved, data licensing and data and metadata standards are not. The literature on energy system data is considerably thinner than that concerning the design and implementation of energy systems models. The open licensing of machine-readable data is a new and burgeoning legal field. Indeed there is little robust analysis and limited case law on which to draw.

Technical openness is also an important consideration. Public machine-readable standardized formats include: CSV (several formatting conventions coexist), ODS, XLSX, SQLite, JSON, XML, YAML, various GIS formats, and CIM (for electrical networks). UTF-8 is widespread for text encoding.

Energy system models originally employed structured text files for data interchange, but by the mid-1990s, modelers were considering relational databases for data processing (Groscurth 1995). These early efforts however remained local to a project and did not involve internet publishing or open data principles. The first energy system database project to go live was OpenEI in late-2009, followed by reegle (after restructuring) in 2011.

Crowdsourced data tends to be collected and released under the ODbL copyleft data license, because most crowdsourced database projects also leverage information from OpenStreeMap. The ODbL is particularly problematic for commercial users, due to its copyleft nature. Some commentators think copyleft may not be a suitable model for most data licensing.
The same legal considerations that prevent copyrighted code from being legally run also prohibit copyrighted datasets from being machine processed. The only citable source on the machine use of copyrighted data (to the author’s knowledge) is Acatech et al (2016b:2), in which comments by Lion Hirth are paraphrased thus:

A major obstacle to open source modeling is that many companies, trade associations, and institutions severely restrict the terms of use of the energy data they provide. This data may not be used directly/without further difficulties for computation.

0. Note the two alternative translations for ohne weiteres in the second sentence.

There are currently few data and metadata standards, formal or informal, relevant to energy system data. Some International Electrotechnical Commission (IEC) standards may apply, such as the CIM (Common Information Model) standard for electrical networks. Ludwig Hülk (Reiner Lemoine Institute) is developing a voluntary metadata standard for energy system datasets, using JSON, a hierarchical human and machine-readable format, and leveraging on existing open data protocols. The standard would record the copyright holders and any applicable license, as well as technical attributes and modification history. Metadata also needs to be open licensed to be useful, which raises legal questions too (Kreutzer 2011:6-10).

Some open energy system database projects support the creation of derived datasets using database queries: SQL for local databases and SPARQL for web databases. Such requests can lead to license compliance issues in relation to attribution, even when confined to datasets under permissive licensing (Meeker 2017:260).

Confirming data quality is altogether different from assessing code quality. Data quality requires a full knowledge of the conditions of collection and subsequent changes. All data modifications should be logged, together with explanations. Managing this provenance is no simple matter. Some database projects (OpenEI) provide forums so that their datasets can be scrutinized, discussed, and ranked. Dataset versioning is used by all projects, although just one (OEP) offers database versioning as well.

Security concerns over critical infrastructure are used to limit the publication of engineering details, particularly for electricity transmission assets (Rivera et al 2015:2). But Vaughan (2017) reports that the primary threat is poor cybersecurity. Notwithstanding, the risk of circulating systems modeling information in public needs to balanced against the benefits of improved energy policy analysis and advice. In any case, this kind of information is increasingly being crowdsourced and published on sites like Wikipedia, Enipedia, OpenStreetMap, and, more recently, by grid identification projects.

The European Commission Joint Research Centre (JRC) is planning to make part of its Integrated Database of the European Energy Sector (IDEES) public in late-2017 (Wiesenthal 2017). The database will initially span the years 2000–2018 for all member states. Dataset licensing is to be governed by the JRC policy on data, given by Doldrina et al (2015). This means that the “acquisition of data by the JRC from third parties shall, where possible and feasible, be governed by the Open Data principles, and all efforts shall be made to avoid imposition of restrictions to their access and use by the JRC and subsequent users” (Doldrina et al 2015:6). The Open Data principles however remain silent on the right of outside users to distribute original and modified works (ibid). With regard to Commission-sourced data, some kind of attribution license, perhaps the EU reuse and copyright notice (European Commission 2011), has been suggested (Zucker 2017). The Commission needs to finalize which open licenses it intends to use for these datasets. Metadata is to follow the JRC Data Policy Implementation Guidelines but, as of July 2017, these guidelines are not yet public.

Current projects

Fig 0 shows a classification of open energy projects as they currently stand. There is a strong bias towards high-resolution technical models and towards engineering and environmental information. Space considerations preclude a proper survey of these projects. In the scheme depicted, a framework is software that is later populated with data to create a scenario or, more pedantically, a framework instance. This approach respects the programming doctrine of code and data separation. The term framework is not used here in its computer science sense.
The electricity grid identification projects are a direct result of the information deficit in relation to distribution and transmission networks. These grid identification projects crowdsource their data, either directly or via OpenStreetMap, to infer a plausible internally-consistent model of the electricity grid under consideration, using techniques from statistics and graph theory. The resulting model is then subject to various forms of validation. (Medjroubi et al 2017, Rivera et al 2017).

The data portal projects comprise two camps. The first is the semantic wiki which uses crowdsourcing and semantic web protocols to assemble, organize, and publish energy data. Enipedia, which went live in March 2011, was the first such wiki for energy and related sectors (Davis 2012). The second camp serves custom on-demand datasets based on user selections, including geolocation. Renewables.ninja, launched in September 2016, provides renewable generation data (Pfenninger and Staffell 2016, Staffell and Pfenninger 2016). Both types of data portal use a rather different paradigm to the relational database or file server model which underpins energy system database projects. Only about half the live open energy database projects support structured queries, the other half simply act as file servers, albeit with an application programming interface (API) to enable programmatic access.

<table>
<thead>
<tr>
<th>Country</th>
<th>Open energy system frameworks</th>
<th>Open electricity sector frameworks</th>
<th>Open grid identification projects</th>
<th>Semantic wikis</th>
<th>On-demand datasets</th>
<th>Open energy system databases</th>
<th>Totals</th>
</tr>
</thead>
<tbody>
<tr>
<td>Australia</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2</td>
</tr>
<tr>
<td>Austria</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>Denmark</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>European Union</td>
<td></td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>France</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>Germany</td>
<td>3</td>
<td>8</td>
<td>4</td>
<td>2</td>
<td>17</td>
<td></td>
<td>16</td>
</tr>
<tr>
<td>Netherlands</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td>4</td>
</tr>
<tr>
<td>South Africa</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>Sweden</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2</td>
</tr>
<tr>
<td>Switzerland</td>
<td></td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>United Kingdom</td>
<td>2</td>
<td></td>
<td>1</td>
<td></td>
<td>3</td>
<td></td>
<td>5</td>
</tr>
<tr>
<td>USA</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>9</td>
<td></td>
<td>16</td>
</tr>
<tr>
<td>Totals</td>
<td>12</td>
<td>16</td>
<td>7</td>
<td>1</td>
<td>1</td>
<td></td>
<td>64</td>
</tr>
</tbody>
</table>

Table 0. Project counts by country of origin and type, as of July 2017. There are now 43. There were five in 2010 and none in 2000. The projects that make up this census are listed at the bottom of the table. [Source: Own assessment]

Table 0 shows that much of the open energy modeling revolution is taking place in Germany, followed by the United States. Possible reasons for the early adoption by Germany include the advanced state of the Energiewende, the absence of official government models, favorable research funding, and the presence of other vibrant open source and open knowledge projects (SUSE Linux, KDE, LibreOffice, Wikipedia DE).

The first projects to release their code (Balmorel in 2001, deeco in 2004, GnuAE in 2005, and OSeMOSYS in 2011), did so for reasons of open development not transparency or reproducibility. The Balmorel codebase contained the following comment (spelling corrected):

Efforts have been made to make a good model. However, most probably the model is incomplete and subject to errors. It is distributed with the idea that it will be useful anyway, and with the purpose of getting the essential feedback, which in turn will permit the development of improved versions to the benefit of other users. Hopefully it will be applied in that spirit.

0. Balmorel was initially released under standard copyright and belatedly added an ISC license in 2017. deeco was first distributed in 2004 with a GPLv2 license and retired in 2005 when key programming libraries lost vendor support. The remaining three projects listed are still active.

The SIREN project from Western Australian NGO Sustainable Energy Now is striking. It is the only open energy system model to be developed by a small NGO for advocacy purposes, showing that official analysis can be countered by community software development (Rose 2016).

Open energy system projects are now networking to advance common aims. One notable example is the Open Energy Modelling Initiative (openmod) which began life as a mailing list in October 2014. The initiative deals with issues of interest to open modelers, including good practice in open source projects, barriers to same, energy data and metadata standards, energy model classification and cataloging, open software and dataset licensing, open
Discussion

Energy modelers need to be crystal clear on their motivation for opening up their models, or more specifically, their code, data, and documentation. Public transparency poses the lowest threshold, met, in many cases, by publishing good documentation and the input and output datasets under standard copyright. Supporting publications should not reside behind paywalls and ideally be open access. Scientific reproducibility requires additionally that the code and data be released under open licenses, so that other researchers can run and verify the scenarios, experiment with the code and data, scrutinize the results and conclusions, and publish their own assessments.

Open development means that the core developers wish to build a community of users and contributors. Or at least allow secondary communities to form around a common codebase. It remains to be seen whether the open source development ethos can be successfully ported to an academic context. The crowdsourcing of energy system data for research purposes represents a similar experiment. Both sit at the intersection between scientific practice and internet-based collaboration.

Open development embeds and extends the requirements for both transparency and reproducibility. And it may ultimately provide a better vehicle for building public engagement and trust in computer-based public policy analysis and advice than either transparency or reproducibility.

For the reasons discussed, the choice of software license may have limited effect on the conduct of a modeling project. Notwithstanding, four cases require consideration, influenced by the choice of language (compiled, interpreted, translated), software dependencies (headers, modules, compiled libraries), and expected source code contributions if any. The use of specialist third-party libraries under GPL licensing will require a compatible GPL license for the mainline code. Projects with permissive licensing cannot directly include code from copyleft sources, nor link to GPL licensed libraries. If a project wishes to prevent web-based execution without the source code being released, it should use the AGPL license. And if executable-only distribution is of concern, then projects (or contributors) should select a copyleft license. The impact of each case, if any, on a particular project will depend on its circumstances and on the priorities and preferences of its core members.

The question of dataset licensing is more difficult. Where possible, permissive licenses should be applied to open data to provide flexibility. Public domain dedications place the least encumbrance on users but does little to assist with provenance and integrity. Crowdsourced material is often required to adopt the copyleft ODbL license because their projects also draw on data from OpenStreetMap.

The legal status of energy system datasets from official and semi-official sources in Europe needs attention and resolution, particularly in regard to its hosting by third-parties. It is essential that such data is able to be used for research without modelers having to operate in a legal gray zone.

Acknowledgments

Tom Brown, Berit Erlach, Ludwig Hülk, Tim Tröndle, Lukas Wienholt, and Frauke Wiese provided feedback on earlier versions. Traffic on Open Energy Modelling Initiative forums helped clarify key issues. Copyright and licensing were discussed in the open data focus group at the first EMP–E meeting in Brussels in May 2017. The author is grateful for these contributions and for the comments from <count> anonymous reviewers.

Conflicts of interest

None. The author does contribute voluntarily to the GENESYS2 open source project, released under a LGPLv2.1 license.

Funding

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

References


GitHub (9 March 2015). "Open source license usage on GitHub.com".

GNU (9 June 2017). "Frequently asked questions about the GNU licenses". GNU Project. Boston, Massachusetts, USA.


Red Hat (2009). TOSW 0.2.2 The open source way: creating and nurturing communities of contributors. Raleigh, North Carolina, USA: Red Hat.


Vaughan, Adam (26 June 2017). "UK energy industry cyber-attack fears are 'off the scale'". The Guardian. London, United Kingdom. ISSN 0261-3077.

Wienholt, Lukas (20 April 2017). Choosing a license for open source code and open data: experiences from the open eGo project — Presentation. Open Energy Modelling Forum, Frankfurt, Germany.


