RESEARCH ARTICLE

Open Science for Computer Simulation

T.Monks[a], A.Harper[a], A. Anagnostou[b] and S.J.E. Taylor[b]

[a]University of Exeter, St Luke's Campus, Exeter, Devon, UK; [b]Brunel University London, UK

ARTICLE HISTORY
Compiled July 14, 2022

ABSTRACT
This paper provides a framework for conceptualising levels of open science and open working within computer modelling and simulation. We aim to support researchers to share their models and working so that others are free to use, reproduce, adapt and build upon, and re-share their work. We introduce a six level framework of increasing complexity: not open, open access, open artefacts, open models, open environment and open infrastructure. For each we provide practical advice on what aspects of open science researchers must consider, what options are available to them, and what challenges they will need to overcome. We illustrate our open science framework using a stylised discrete-event simulation model. All code used in this paper is available, cloud executable and reusable under an MIT license.

KEYWORDS
Open Science; discrete-event simulation; agent based simulation; reproducible research;

1. Introduction

Consider the following scenario. A researcher wants to explore a problem or system using modelling and simulation (M&S). After defining the problem and/or requirements and perhaps creating some form of conceptual model, she gathers data and develops a model using some programming language or API. After verification and validation, she then conducts experiments to explore the problem/system under different conditions by simulating the model with different parameters and/or data. After obtaining results from experimentation the researcher makes deductions and conclusions, or perhaps performs more investigation. The researcher shares the outcomes of this research by publishing her experiences in a conference or a journal article. Other researchers are inspired by this researcher's work and want to understand what she has done in more detail and build on her work. They run into several possible problems:

(1) The model and data described in the paper cannot be accessed - researchers email the author for these. Unfortunately, for whatever reason, the author does not respond for some time and the researchers move on to another problem;
(2) The model and data described in the paper can be accessed through a download link - researchers use the link but it does not work anymore. See (1);

Thomas Monks. Email: t.m.w.monks@exeter.ac.uk

(3) The model and data described in the paper can be accessed through a permanent link - researchers download the model and the data - frustratingly there are no instructions as to how to run the model and the researchers cannot figure out what APIs and versions of the APIs were used, what operating system the model ran on, etc. After some time and interactions with the author the researchers make no progress and, frustrated, decide to move on to another problem.

Points 1 and 2 above are surprisingly common. Point 3 is significantly less common and does not always lead to disappointment as many researchers do document their model quite well. Standards have been developed in computer simulation to make M&S documentation more structured and their use is becoming more widespread (E.g., Rahmandad and Sterman, 2012; Grimm, Railsback, Vincenot, Berger, Gallagher, DeAngelis, Edmonds, Ge, Giske, Groeneveld, et al., 2020). Implementation problems can still exist, particularly when the work was published some time ago or there is a mismatch between computing skills or specialist APIs/computer systems have been used.

This article aims to support researchers and practitioners who wish to make their computer simulation models and working open to others. Although not our primary aim, the tools and approaches we describe also support reproducibility of results of a published M&S journal article; that is the ability to repeat all analyses, and reproduce all figures, charts, and quantitative results included in a journal article. We provide this support in six steps of increasing technical complexity.

We first briefly review open science and reproducibility within M&S. We then detail our framework that provides six levels of open working: not open; open access; open artefacts; open models; open environment; open infrastructure. We then provide a stylised example of open working. The model used is a simple discrete-event simulation model. We demonstrate how to achieve the different levels of open working within our framework and provide a set of results for readers to reproduce. We close with a discussion of the ways forward for open science in computer simulation.

2. Reproducibility and Open Science

Reproducibility in simulation research is a recognised issue (Donkin, Dennis, Ustalakov, Warren, and Clare, 2017; Uhrmacher, Brailsford, Liu, Rabe, and Tolk, 2016; Fitzpatrick, 2019; Warnke, Helms, and Uhrmacher, 2017; Ruscheinski, Warnke, and Uhrmacher, 2020). While the scientific term 'reproducibility' is often used, for research involving computer artefacts such as modelling and simulation, Feitelson (2015) suggested five levels of terminology:

- Repetition involves re-running experiments using the original artefacts.
- Replication involves recreating the artefacts.
- Variation involves recreating the artefact with an intended modification, such as changing a parameter.
- Reproduction involves recreating the essence of an experiment and artefacts in a similar, but not identical setting.
- Corroboration obtains the same result with other methodologies or procedures.

Despite widespread awareness of the importance of reproducibility, Janssen, Pritchard, and Lee (2020) found that the majority of Agent Based Simulation publications do not even mention which modelling platform or programming language were

used. A standardised approach to model description and documentation in published research is one approach to supporting validation and replication, and documentation guidance and standardised formats/protocols have been developed for this purpose (Rahmandad and Sterman, 2012; Grimm et al., 2020; Köhn and Le Novère, 2008; Monks, Currie, Onggo, Robinson, Kunc, and Taylor, 2019; Grimm, Berger, DeAngelis, Polhill, Giske, and Railsback, 2010; Moallemi, Elsawah, and Ryan, 2020). However, Janssen (2017) found that only 7% of published Agent Based Simulation studies used ODD protocol, while Alvarez, Brida, and London (2020) reported wide gaps in the use of ODD across disciplines. Further, Donkin et al. (2017); Bajracharya and Duboz (2013); Zhang and Robinson (2021) found that even the use of a clear and consistent protocol may not be sufficient for simulation replication.

Alongside full documentation, transparency can be supported by providing the model's source code or programme in an open-source software. In practice, this is rarely achieved. Janssen (2017) found that 10% of 2300 articles made source code available for publications. Janssen et al. (2020) extended their examination to 7500 articles of agent-based and individual models, reporting that only 11% of articles shared code, although there is an upward trend. Further, requests for code are frequently not honoured, even in journals demanding reproducibility (Janssen et al., 2020; Stodden, Seiler, and Ma, 2018). This finding appears to hold even in contemporary M&S Covid-19 studies. Via Scopus, we identified 200 DES, ABS and hybrid modelling papers between 2020-2021 that describe some form of COVID-19 related model. Of these 15% had "accessible" models; that were largely poorly documented code available online, or had a 'you can run the model at the vendors site' disclaimer (Taylor and Monks, 2022).

Commercial off-the-shelf software (COTS) have contributed to the diffusion of DES and ABM in academia and industry, providing advanced visual features, but bringing attached costs which restrict access. Generic models are of limited use when not made openly accessible, for example Penn, Monks, Kazmierska, and Alkoheji (2020) developed a generic healthcare DES model in a COTS package. While they found that modifying an existing model has significant advantages over creating a bespoke model, the use of proprietary software limits its flexibility and availability, and the ability of others to access, use and adapt the model. There are many free and open source software packages for modelling and simulation. Models written using these software can be shared on public repositories with a license to enable other users to reproduce, adapt or distribute the software/model; generally, to copy or modify a programme, the more permissive the licence the better (Kapitsaki and Charalambous, 2019; Dagkakis and Heavey, 2016).

Building on reproducibility and associated concepts, the concept of Open Science encourages researchers to make their scientific processes and research outputs as widely accessible as possible. Fecher and Friesike (2014) suggest that Open Science comes from five different schools of thought: measurement (of the impact of science), democratic (free access to knowledge arising from science), pragmatic (making the process of knowledge creation more efficient), public (making science available to citizens) and infrastructure (openly available platforms, tools and services for scientists). There are many benefits to openness including reduced science cost, increased transparency and quality, faster knowledge transfer, better dissemination of knowledge, increased public engagement, more coordinated international actions and inclusion of scientists in developing countries (OECD, 2015). The FOSTER project (`https://www.fosteropenscience.eu/`), a major international training initiative, recognises several key themes in Open Science (Open Access, Open Data, Open Repro-

ducible Research, Open Science Evaluation, Open Science Policies and Open Science Tools). Although many might find Open Science a new concept, others argue that Open Science encompasses a range of concepts that should be considered as normal scientific practice (Anagnostou and Taylor, 2020). However, it must be noted that not all researchers can make their models open e.g. those based on industrial or health systems where commercial or data protection issues are of concern (Taylor, Eldabi, Monks, Rabe, and Uhrmacher, 2018).

Taylor, Anagnostou, Fabiyi, Currie, Monks, Barbera, and Becker (2017) suggested guidelines for Open Science in M&S. Where possible researchers should publish openly using Green or Gold open access, document models using standardised methods (as noted above), make scientific outputs (e.g., data, results, model source code, etc. with the necessary data, parameters and random number seed) openly accessible by using Open Access Repositories that support the use of Digital Object Identifiers (DOIs), use licenses to specify how your work should be shared and used, use a Researcher Registry (e.g., ORCID to uniquely identify a Researcher and link this to associated works via DOIs, and consider using a Science Gateway or similar web-based approach to enable the widest possible access. Adopting this means that in theory the articles reporting the research can be accessed, the artefacts described in the research can be accessed, and other scientists can attempt to reproduce the results of simulation experiments reported in the articles using the same experimental artefacts. This would enable others to build on the research and to identify possible flaws in the original research, providing additional confidence in model behaviour. For example, Zhang and Robinson (2021) were able to replicate and improve upon a published model having uncovered several errors.

While the above constitutes elements of how MS researchers might make their work open, it does not really give an "organised" view. Equally, it does not capture the effort of doing this. For example, a common Open Science practice is to make the article describing the research open through the use of an institutional open access repository (i.e., Green Open Access) or to make the article open at the publisher (i.e., Gold Open Access)[1]. This takes little effort and enables others to access the article for free. However, it does not allow others to access the model etc., or to reproduce the results in the paper. Making the model, data, results etc. openly accessible takes considerably more effort but also increases the openness of the research. There are different approaches that one might take within this with varying degrees of effort balanced with degrees of openness and reproducibility. The key issue is that our M&S community does not really have a guide to do this, one that tries to show effort versus openness. The next section describes our conceptualisation of this.

3. Levels of Open Working for Modelling & Simulation

The previous section identified approaches to Open Science and reproducibility with respect to M&S. It also introduced the various issues that need to be considered when attempting to make M&S as open as is practical. None of these really capture the degree to which openness has been accomplished. This is particularly important in M&S as it is not just data, as with open data, nor is it just software, as with open source software - it is both and more. Reporting guidelines such as ODD, STRESS etc (Grimm et al., 2020; Monks et al., 2019) help to capture this. However, there is

[1]We note that issues around Green and Gold Open Access are changing rapidly – see https://www.coalition-s.org/ for more information.

Table 1. Framework for open working in computer simulation

Level	**A researcher publishes an article about a simulation study that describes the aim, background literature, approach, data, model and results**
0. Not Open	The article is behind a paywall and cannot be accessed without payment or some form of subscription.
1. Open Access	The article is published under Gold or Green Open Access. Researchers can freely access the article.
2. Open Artefacts	As Level 1 + the researcher openly publishes the data, model and results by depositing them in an Open Access Repository following Open Data practices.
3. Open Models	As Level 2 + the researcher documents the simulation study using a standard reporting process, documents the model to specify how it should be implemented and run, and potentially seeks to have some form of independent reproducibility certificate.
4. Open Environment	As Level 3 + the researcher installs the model and associated artefacts on a virtual machine or virtual environment.
5. Open Infrastructure	As Level 4 + the researcher deploys the model and associated artefacts with a Science Gateway to allow other researchers to run the model as simply as possible. This may enable simulation on the web.

[a]Higher levels represent a higher level of open working has been achieved.

[b]Higher levels of open working require additional effort to achieve

a need to also capture the ease that results can be reproduced, software used and accessed, etc. These guidelines can capture these but not at any level of transparency. We therefore propose a series of M&S Openness Levels that may help to clarify. These are summarised in Table 1. Higher levels represent a higher standard of open working. The higher the level the more effort required to achieve it by the MS team.

3.1. Level 0: Not Open

An author does not attempt to making anything open.

An author publishes an article about a simulation study that describes the aim, background literature, approach, data, model and results. This is the lowest level of openness where no attempt has been made to make anything open. The article may have a DOI that facilitates search discovery. However, the article is behind a paywall and cannot be accessed without payment or some form of subscription. Reproducibility is extremely limited as there is no access to the article or any research artefact and it is impossible to reproduce the results without contacting the authors (with the issues as described in the introduction).

3.2. Level 1: Open Access

An author makes the published article free to access.

An author publishes an article about a simulation study that describes the aim, background literature, approach, data, model and results. The article may have a DOI that facilitates search discovery. Other researchers can access the paper without payment. Reproducibility is again extremely limited as there is no access to the article or any research artefact without contacting the authors.

As discussed above, Open Access Publishing aims to make a version of an article free to access. Two main classifications exist: Gold and Green. Under Gold open access the publisher of a journal provides free open access to the articles of that journal. This is funded by the author (or institution) who pays the journal an Article Publishing Charge (APC) (charges vary considerably). Some journals only charge for

printed versions and offer free Gold open access online. Others do not charge. Articles are normally licensed under a some form of sharing licence (e.g., Creative Commons). Green open access involves an author self-archiving an article, typically in some institutional Open Access Repository. This might be the journal article, a post-print or a pre-print depending on the agreement with a publisher. The self-archived article might also be subject to an embargo period set by the journal publishers. A major issue in Green open access, however, is discoverability (i.e., when searching for the self-archived version of a paper). In the wider-field of computational modelling many authors are making use of pre-print servers. For example, many machine learning articles can be found at `arXiv.org` (hosted by Cornell). Another popular choice for pre-prints is the Open Science Foundation (OSF) `https://osf.io/preprints/`. Articles published as pre-prints are not peer reviewed, but provide researchers with a way to record what work has been done, by whom and when. Repositories such as the OSF also allow researchers to easily link to and research artefacts, such as data or software, related to the paper. This brings us to level 2.

3.3. Level 2: Open Artefacts

An author makes the article free to access and makes as many M&S artefacts (the model, data, results, software, etc.) as open as possible.

An author publishes an article about a simulation study that describes the aim, background literature, approach, data, model and results. The article will have a DOI that facilitates search discovery. The article is published under Gold or Green Open Access. The data, model and results are published openly. The M&S study is more open, as researchers can access these artefacts as described in the paper. However, researchers may still need to contact the authors to understand how to implement and run the model and may need specific software skills to do this (e.g., understanding how to use Python, Java, C++, etc. or a specific COTS simulation software package (authors may need to pay to get a licence for the latter). Researchers may also encounter problems if their computing environment is different (e.g., different operating system, different system variables, access to different APIs/runtime libraries, etc.).

Authors may choose to host research artefacts on their own website. Very few websites guarantee that links are persistent (i.e., unlike DOIs). In particular, links to webpages served by academic institutions or textbook supplementary online pages on a publisher's site tend to be highly unreliable and make the artefacts inaccessible (i.e., they change or are broken over time). Authors might take care initially to make sure things work but over time they forget and move onto other things. Journals sometimes make provisions for online supplementary material. This provides an opportunity to publish the model directly with the paper (e.g., Wood (2021)). A potential downside is that the model artefact is not directly discoverable or searchable this way; someone must find and read the paper in order to understand that the appendix contains a model.

An approach that is in line with Open Science thinking is to publish research artefacts in an Open Access Repository with a DOI. This means that there is a permanent version of the data, model and/or results that is accessible via the DOI. The DOI reference can also be placed in the published article to make finding the artefacts as easy as possible. CERN's Zenodo (`https://zenodo.org/`) is one well-known example of an Open Access Repository. This allows artefacts to be stored at CERN's data centre and guarantees to hold them "as long as CERN exists." It has excellent versioning control

and integration to other open science enabling platforms. Use of a repository for models also has the benefit that it makes them citable, via a DOI, and findable independent of an academic paper. An academic publication would them simply cite the artefact. For example, the simulation models used in Allen, Bhanji, Willemsen, Dudfield, Logan, and Monks (2020) are published on Zenodo and cited as any other publication would be cited i.e. Allen and Monks (2020) using DOI 10.5281/zenodo.3760626. Academic institutions often have their own open science repositories and *mint* (create) their own DOIs. Open Access Repositories are better than placing artefacts in a journal's supplementary appendix as are permanent, tend to be more flexible, and have supporting bibliometric tools that journals do not have.

If a researcher has developed software (e.g., a model written in a script or programming language, or supporting programs) then an alternative to depositing it in a open science repository is to make use of a dedicated software repository such as GitHub (`https://github.com/`), GitLab (`https://about.gitlab.com/`) or BitBucket (`https://bitbucket.org`). In addition to providing public repositories that make code available, findable and searchable, they provide a number of benefits for open science and software project management. The full breadth of functionality is beyond the scope of this paper (e.g., the use of a software repository to support collaborative model development), but here we briefly discuss releases, and integration to repositories. For simulation models, a big benefit is the ability to use version control, and the ability to create numbered releases of a model (e.g., version 1.0.0, version 1.1.0, etc.). Version numbering usually increases, for example, as features are added or bugs are fixed. This provides an easy way to communicate with simulation clients or users and ensure that they and you are talking about the same version of the model. A secondary benefit is that GitHub, for example, can be linked Zenodo as a permanent Zenodo artefact (effectively their repository is frozen and made available). If a developer has done this then their repository also has a DOI provided by Zenodo. If a developer then creates a new release or version of their model on GitHub, this new version is then made available through Zenodo without breaking links within a journal article (i.e., a new version but with the same DOI). This feature also allows you to keep a GitHub repository live and up to date with production code without any concerns that the reproducibility of historical publications is affected. Allen et al. (2020) gives a simple example of these two tools used together.

Many M&S researchers use commercial or proprietary simulation software in their research. These packages cannot be made available openly and could be a barrier to reproducibility as others cannot access or afford the package. However, many simulation software vendors have a free to access version of their software (sometimes a web-based version or evaluation copy). Further, some vendors will work with researchers to create a version that is free to access (e.g., a runtime version that allows others to run, but not edit a model).

It is also good practice when making artefacts open to provide an appropriate license for each artefact. The main reason for this is to make it clear how others may use, cite and reuse their work. At a minimum a license specifies the terms of use for a model, and the liabilities of the authors for reusing it. Most modern data science applications choose one of two permissive licenses: The BSD 2-Clause license and the MIT license. Both of these licenses specify that the software and documentation may be used for public, private, or commercial applications, and can be modified and distributed. The authors of the software are not liable for any reuse of the software and provide no warranty. A condition of use is that a copy of the license is included in any copy or application that include a substantial portion of the software. As an

Table 2. The MIT Permissive License

MIT License
Copyright (c) 2021 [copyright holder]

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

example, we include a copy of the full MIT license in Table 2. Authors may also use Creative Commons licences. Creative Commons provide an excellent tool to help authors to choose which is the appropriate licence (`https://creativecommons.org/share-your-work/`).

Overall, there are many opportunities to make the artefacts of M&S more open. Some researchers find developing a Data Management Plan (DMP) useful (indeed this is mandatory in some disciplines and under some funding agencies). A DMP helps to describe the data used in research, especially with respect to how it would be managed, analysed, stored and preserved (see `https://dmponline.dcc.ac.uk/` as an example of a tool that can to create a DMP). For more information on Open Data see Chauvette, Schick-Makaroff, and Molzahn (2019)

3.4. Level 3: Open Models

An author makes the article free to access and makes as many M&S artefacts (the model, data, results, software, etc.) as open as possible. The author also aims to make it easier for others to reproduce her/his results by documenting how the results were obtained and producing a guide to using their model and software.

In addition to publishing both her/his article and artefacts openly, the author fully documents their model and associated software using a recognised reporting format (e.g., STRESS, ODD, etc.) and creates a guide to using and running the model and associated software. The author optionally applies and is awarded a certificate of reproducibility from a recognised body (e.g., the RCR from ACM). Reproducibility is excellent as the study has been documented properly and independently tested. Researchers may still need specific software skills to do this and may still encounter problems if their computing environment is different.

Several approaches to documenting a M&S study exist. Many of these tend to be technique specific, for example, Agent Based Simulation in ODD (Grimm et al., 2010) or System Dynamics in MMRR (Rahmandad and Sterman, 2012), and/or domain specific (e.g., biological systems). In an attempt to unify reporting across the three most well-known M&S techniques, we created the STRESS guidelines (Monks et al., 2019). This provides a checklist for Agent Based Simulation, System Dynamics, and

Discrete-Event Simulation Models and can be used to support documentation provided as supplementary material to a journal article (e.g., Saville, Monks, Griffiths, and Ball (2021)).

In addition to reporting guidelines, more modern data science tools can be employed to improve the organisation of code and model documentation. In section 4 we provide a case study illustrating one way to create an open working environment for computer simulation models. We emphasise that the available toolkit for open science is evolving fast and this is just one way to achieve open models. Our aim is to demonstrate that it can be achieved now with relative ease.

3.5. Level 4: Open Environment

An author makes the article free to access and makes as many M&S artefacts (the model, data, results, software, etc.) as open as possible. The author also documents how the results were obtained and a guide to using their model and software. To facilitate ease of use the author develops an open environment that other researchers can use without needing to install software, set up databases, spreadsheets, etc.

In addition to the above levels, the researcher also creates a virtual machine and installs the model, data and associated software. Researchers need to understand how to run a virtual machine and the installed model but do not need to understand how to implement the model. Barriers to using the research outputs of the article are reduced as specific software skills are no longer needed. However, researchers still need to understand how to deploy a virtual machine. Further, this may not be possible if COTS simulation software is used.

It might appear in the first instance that just installing and running software is straightforward. However, particularly with software that uses multiple components and languages like JAVA, it is sometimes difficult to get all the installation and runtime environmental paths, libraries, APIs correct in terms of syntax, location and version. Version control is a major problem as, for example, the application might not work correctly if a single library is the wrong version. Correcting that version might stop another application from working. This is further compounded by differences in operating systems and computers (e.g., Windows, Mac OSX and Linux). This *installation problem* is not unique and has been a major problem in the software industry. Computing applications often require libraries, API, specialist operating systems features, etc. that in turn need careful installation.

The STRESS reporting guidelines suggest a simple solution for M&S researchers: create a set of instructions that specify the OS, software and dependencies to download and install locally. This sounds sensible, but the limitation of this approach is that it can be extremely challenging for users to get a model up and running. Carefully crafted instructions can still be ambiguous and there is also a real issue with how long a set of instructions remains valid: in reality, a complete unknown.

Virtualisation is an excellent solution to this - one creates a virtual machine that has all the software installed and a user just runs that virtual machine on their system. Admittedly, a user must learn how to do this but arguably it is a simpler route to making their study more open and accessible. There are different virtualisation technologies available. One of the most widely known is Docker (`www.docker.com`). A Docker "container" for your operating system and installed model and artefacts can be created using the excellent tutorials developed by the very large Docker user community. This can be made available online through DockerHub. This is similar to

GitHub, but instead hosts these software images as opposed to code. As an example, consider a situation where a model developed using Repast Simphony 2.8.0 on Ubuntu 20.10. How can a Windows 10 user recreate this software environment with all of its dependencies? A solution is to package up the model and OS in a Docker image made available through DockerHub. A Windows or Mac OS user installs Docker on their local machine, 'pulls' the image (downloads a remote image to a local machine), and then runs the image in order to access the model. The user need not be concerned with how the simulation model is implemented (even to the extent that it is coded in repast), the OS, or how it is installed and setup. Rather the user can focus on experimentation with the model to obtain results. Docker is part of the wider open science ecosystem and integrates, for example, with GitHub: researchers can push updates of their simulation model code to a repository and automatically create a new up-to-date docker image for others to use.

3.6. Level 5: Open Infrastructure

An author makes the article free to access and makes as many M&S artefacts (the model, data, results, software, etc.) as open as possible. The author also documents how the results were obtained and a guide to using their model and software. To facilitate ease of use the author develops an open environment that other researchers can use without needing to install software, set up databases, spreadsheets, etc. The author hosts that environment on an openly accessible infrastructure and provides access through the web.

An author publishes openly, creates a virtual machine but also wants to make access to their work as simple as possible. They deploy their work on a digital infrastructure and create a science gateway, a web-based portal or front end, that can be used to easily access and run their model. This takes more work from the researcher but is perhaps the simplest way for other to access and run the model.

In some scientific communities, scientists make use of digital infrastructures (e-Infrastructures, cyberinfrastructures or e-Science infrastructures). These are integrated collections of computers, data, applications and sensors across different organizations to support international scientific projects (Foster, Kesselman, and Tuecke, 2001; Hey and Trefethen, 2005; Bird, Jones, and Kee, 2009). Large-scale infrastructure providers have established a sustainable funding base over the long term and are supporting a range of scientific and, to some extent, industrial projects (e.g. the European Grid Initiative (EGI) (European distributed computing infrastructure), GEANT (European high performance networking infrastructure) and supporting National Research and Education Networks (NRENs) e.g. JANET in the UK (Barjak, Eccles, Meyer, Robinson, and Schroeder, 2013). For those that do not have access to these, there are alternatives that allow others to host and deploy their own infrastructures. Briefly, applications can be created from these by first deploying the application service on the e-Infrastructure and registering it in some form for service catalogue (see below) and then accessing the service via a science gateway (a web-based system that allow scientists to use e-Infrastructures with a simple front end that has been developed for their needs) or some kind of programming interface (usually some kind of RESTful interface) integrated into software that is familiar to the user. These have evolved to support workflow languages that allow tasks to be chained together (Deelman, Gannon, Shields, and Taylor, 2009; Liew, Atkinson, Galea, Ang, Martin, and Hemert, 2016).

The above allows Open Infrastructures to be developed (open still needs security authorisation) and science gateways to be created. These are front end web portals: essentially a website dedicated to running the researcher's model. This needs security (Authentication, Authorisation and Identification - AAI) so using an independent AAI service (e.g., eduGAIN) end users log on to the portal. A web form could be used to put in data and/or parameters used in the model. Another option if the researcher used a FAIR Open Access Repository is to input a DOI of the data and parameters used in a paper. End users could then use the form to run the model. This then uses the second part of the science gateway, the science gateway server. The server runs a compute engine. When a request to run a model comes through from the front end, the server downloads the model in its container from the container repository and runs the container on the compute engine. It passes the DOI URLs to the model and the model runs. Access to local or remote computing is provided depending on the requirements of the model. On completion the results are passed back to the front end and are downloaded by the end user. These might then be compared to the results described in the paper (and stored in the FAIR Open Access Repository). Researchers then continue to use the model for their own experiments and published further work inspired by the original article. An example of how can be found in (Anagnostou, Taylor, Groen, Suleimenova, Anokye, Bruno, and Barbera, 2019) where a model has been developed and deployed in a Science Gateway.

In addition to creating an open science gateway, we note that for simulation tools written in general purpose programming languages there are free tools to enable open environments i.e. free simulation on the web; albeit with very limited computational resources, flexibilty and security relative to a full open science gateway. For example, in R users might consider a Shiny App or in Python users might consider Streamlit or BinderHub (for example, Currie and Monks (2021) ; Monks (2020)). The suitability of these tools will vary on the application of modelling and simulation, but offer a simple way for others to access an executable model remotely.

4. Case study: An Open Treatment Centre Model

To illustrate possible ways to achieve up to level 5 of open working in computer simulation, we provide an online case study. We have created this as a separate artefact that can be cited. This has been deposited and assigned a DOI via Zenodo (Monks, Harper, Taylor, and Anagnostou, 2022). The use of Zenodo and a DOI means that we have a permanent record and link to the work and avoid long term archival issues of standard URLS. We can also update the deposit if our paper or model needs to be revised creating a new DOI (i.e. our old DOIs including in publications or presentations are still valid and link to the most recent update). An online interactive version of the case study can be found at `https://tommonks.github.io/treatment-centre-sim`. We will now explain our case study setting, open working methods, and provide a reproducible result. We summarise our open working across our levels in table 3.

4.1. Case study setting

We adapt a textbook example from Nelson (2013, p.170): a discrete-event simulation model of a U.S based treatment centre. In the model, patients arrive to the health centre between 6am and 12am following a non-stationary poisson process. On arrival, all patients sign-in and are triaged into two classes: trauma and non-trauma. Trauma

Table 3. Case study: tools used to make the project open.

	Level	Achieved via
1	Open Access	Open Science Foundation Pre-print; Gold open access
2	Open Artefacts	Zenodo+DOI; Github
3	Open Model	Free and open sim software; MIT licensed model; Jupyter Book; STRESS-DES
4	Open Environment	Docker image; dockerhub
5	Open Infrastructure	BinderHub

patients include impact injuries, broken bones, strains or cuts etc. Non-trauma include acute sickness, pain, and general feelings of being unwell etc. Trauma patients must first be stablised in a trauma room. These patients then undergo treatment in a cubicle before being discharged. Non-trauma patients go through registration and examination activities. A proportion of non-trauma patients require treatment in a cubicle before being discharged. The model predicts waiting time and resource utilisation statistics for the treatment centre. The model allows managers to ask question about the physical design and layout of the treatment centre, the order in which patients are seen, the diagnostic equipment needed by patients, and the speed of treatments. For example: "what if we converted a doctors examination room into a room where nurses assess the urgency of the patients needs."; or "what if the number of patients we treat in the afternoon doubled".

4.2. Open working

We have made the artefacts and model discoverable and open (levels 1, 2 and 3) by publishing this paper as a pre-print on the Open Science Foundation and creating an online book documenting the model using the Jupyter Book framework (Executable Books Community, 2020). Jupyter books aim to bring together jupyter notebooks, code, images, markdown and other media (e.g. Videos) describing computational material in order to produce publication quality work. As such it is ideal for open working in computer simulation. All the artefacts used to create the online material are included in the Zenodo archive. Production code for the simulation is available on Github `https://github.com/TomMonks/treatment-centre-sim`. All artefacts are published under permissive licenses (CC-BY for the book and MIT for code). The book includes both the model code and experiments in a notebook to help reader understanding. The model coding uses free and open tooling from python 3 and Simpy 4 (Team SimPy, 2020). The code is supplemented by a section on STRESS reporting for discrete-event simulation, non-technical explanations of the code, and step by step instructions to execute the model and reproduce results. Our open model also includes instructions to potential contributors. We have used GitHub Issues and pull requests, to enable anyone to report, typographical errors, mistakes in our code or methodology, suggested improvements and general feedback on the study.

We developed our version of Nelson's treatment centre model on Ubuntu Linux using a variety of python tools and versions. To achieve level 4's open environment we have created a docker image that can be used to run the model without requiring a machine running linux or python. We invite interested readers to install docker (`https://docs.docker.com/`), pull our image, and follow our instructions to run the model. The image and instructions can be found at `https://hub.docker.com/r/tommonks01/treat_sim`. We also provide a simpler, but less reliable, solution using a conda virtual environment that will attempt to install the correct version of python packages directly on a machine regardless of the operating system.

Table 4. Simulation results that can be verified by the example reproducible pipeline.

Mean waiting time (mins)	base	triage+1	exam+1	treat+1	triage+exam
Triage	28.86	1.09	36.46	39.36	1.14
Registation	113.56	138.79	106.27	108.46	140.14
Examination	24.57	24.15	0.13	23.03	0.15
Non-trauma treatment	137.72	139.98	151.82	2.26	152.09
Trauma stabilisation	144.46	154.00	132.94	146.09	165.43
Trauma treatment	168.27	209.12	195.16	150.37	193.69

Lastly, we provide a simple example of achieving some of level 5's open infrastructure aims. Our model or docker image does not need to be installed locally. Instead the model can be executed online (without installation locally) using BinderHub directly from the Jupyter Book.

4.3. Example of reproducible results

As a simple demonstration of the reproducible pipeline we have constructed for this example, consider Table 4. This contains a subset of results for several mean waiting time performance measures from the model, across multiple experiments. We invite readers to attempt to recreate these results using our online interactive Jupyter Book and model that accompanies the article. The final step in analysis of the model will both generate a comparable table and the LaTeX used to create the table within this paper. If reproducible results (e.g. charts, tables, and statistics) are an aim for a study we would encourage researchers to take a similar pipeline approach.

Alternatives to the data science tools we list here for improving ease of use do of course exist. Interested readers may have their own methods and we encourage them to publish their models and research openly in order to share their best practice.

5. Discussion

The academic M&S community is advancing an active interest towards open models and methods in line with principles of Open Science. However, current knowledge and applications tend to be disparate, and there is an increasingly urgent need to bring resources together into a coherent, supportive framework for M&S researchers. Other computational modelling fields that also fall underneath the umbrella term of 'data science' are progressing their open science practice considerably faster than M&S. For example, in the UK the Alan Turing Institute has developed the Turing Way (The Turing Way Community and Scriberia, 2021): a handbook to support reproducible, ethical and collaborative data science. We note that this is a large community-driven effort for open science. Our framework aims to begin coordinating similar open research and work within the M&S community.

An immediate specific challenge for applied open research in M&S is how to manage the benefits and downsides of commercial software. While COTS packages offer substantive benefits to users, our framework illustrates that they inherently limit who else can use, adapt, and benefit from the research. Potential model users might include researchers from developing nations, early career researchers, M&S students, governments, citizens, and front line services such as fire, police, and health care. While we expect COTS packages to remain the dominant tool within the M&S literature, we argue that researchers should consider the impact of their choice of software on

their ability to share their work, on day-one of their M&S study. High quality free and open source software for M&S are available (Dagkakis and Heavey, 2016) and, depending on the application, may be a better choice for research. In some cases this is an ethical decision, for example when a study is funded using public money from a research council, or when simulation-supported decisions affect the lives of people. The use of free and open source simulation may also be a more inclusive decision that allows diverse groups of researchers and professionals to collaborate, or directly reuse research material regardless of their circumstances and budget.

As has been said elsewhere, openly published parameters, models, results, and full documentation using a recognised reporting format such as STRESS (Monks et al., 2019) or ODD (Grimm et al., 2020) supports transparency and reproducibility. We urge M&S researchers and authors to consider this as their default mode for science and not as additional work. However, our framework illustrates that reporting guidelines do not support all aspects of open science for simulation. We have outlined the issues of open working that researchers will need to consider, such as model installation and execution, operating system differences, software dependencies, distributed version control, licensing, and discoverability and maintenance of links. We have also demonstrated the application of modern data science tools to improve the organisation of code and model documentation through open infrastructure and open environments. Our stylised case study demonstrates one possible way of achieving this with relative ease, enabling reuse and adaptation by both researchers and industry users.

A major barrier to moving to open models is the lack of high quality examples available, or open workflow guidance and methodology in the literature for researchers, journal editors and article reviewers to draw on and critique research. Additionally, research to support open science in simulation is required to remove the complexity and learning barrier that many researchers face when adopting contemporary computer science knowledge to future-proof materials and methods to enable better open research. An inevitable limitation of reproducible research guides such as the Turing Way is that they come from a research software engineering (RSE) perspective. Open science within M&S community would benefit from more tailored guidance that tackles open models. This is an exciting opportunity for M&S where new methods and infrastructure could be developed to support the diverse community and skills within it.

Acknowledgement(s)

TM and AH are funded by the National Institute for Health Research Applied Research Collaboration South West Peninsula. The views expressed in this publication are those of the author(s) and not necessarily those of the National Institute for Health Research or the Department of Health and Social Care.

References

Hazhir Rahmandad and John D Sterman. Reporting guidelines for simulation-based research in social sciences. 2012.

Volker Grimm, Steven F Railsback, Christian E Vincenot, Uta Berger, Cara Gallagher, Donald L DeAngelis, Bruce Edmonds, Jiaqi Ge, Jarl Giske, Juergen Groeneveld, et al. The odd protocol for describing agent-based and other simulation models: A second update to

improve clarity, replication, and structural realism. *Journal of Artificial Societies and Social Simulation*, 23(2), 2020.

Elizabeth Donkin, Peter Dennis, Andrey Ustalakov, John Warren, and Amanda Clare. Replicating complex agent based models, a formidable task. *Environmental Modelling & Software*, 92:142–151, 2017.

Adelinde M Uhrmacher, Sally Brailsford, Jason Liu, Markus Rabe, and Andreas Tolk. Panel—reproducible research in discrete event simulation—a must or rather a maybe? In *2016 Winter Simulation Conference (WSC)*, pages 1301–1315. IEEE, 2016.

Ben G Fitzpatrick. Issues in reproducible simulation research. *Bulletin of mathematical biology*, 81(1):1–6, 2019.

T Warnke, T Helms, and A M Uhrmacher. Reproducible and flexible simulation experiments with ML-Rules and SESSL. *Bioinformatics*, 34(8):1424–1427, 11 2017. ISSN 1367-4803. . URL `https://doi.org/10.1093/bioinformatics/btx741`.

Andreas Ruscheinski, Tom Warnke, and Adelinde M. Uhrmacher. Artifact-based workflows for supporting simulation studies. *IEEE Transactions on Knowledge and Data Engineering*, 32(6):1064–1078, 2020. .

Dror G Feitelson. From repeatability to reproducibility and corroboration. *ACM SIGOPS Operating Systems Review*, 49(1):3–11, 2015.

Marco A Janssen, Calvin Pritchard, and Allen Lee. On code sharing and model documentation of published individual and agent-based models. *Environmental Modelling & Software*, 134: 104873, 2020.

Dagmar Köhn and Nicolas Le Novère. Sed-ml–an xml format for the implementation of the miase guidelines. In *International Conference on Computational Methods in Systems Biology*, pages 176–190. Springer, 2008.

Thomas Monks, Christine SM Currie, Bhakti Stephan Onggo, Stewart Robinson, Martin Kunc, and Simon JE Taylor. Strengthening the reporting of empirical simulation studies: Introducing the stress guidelines. *Journal of Simulation*, 13(1):55–67, 2019.

Volker Grimm, Uta Berger, Donald L DeAngelis, J Gary Polhill, Jarl Giske, and Steven F Railsback. The odd protocol: a review and first update. *Ecological modelling*, 221(23): 2760–2768, 2010.

Enayat A Moallemi, Sondoss Elsawah, and Michael J Ryan. Strengthening 'good'modelling practices in robust decision support: A reporting guideline for combining multiple model-based methods. *Mathematics and Computers in Simulation*, 175:3–24, 2020.

Marco A Janssen. The practice of archiving model code of agent-based models. *Journal of Artificial Societies and Social Simulation*, 20(1), 2017.

Emiliano Alvarez, Gabriel Brida, and Silvia London. Agent based models and simulation in social sciences: A bibliometric review. *DOCUMENTO DE TRABAJO*, 2020:26, 2020.

Kishoj Bajracharya and Raphael Duboz. Comparison of three agent-based platforms on the basis of a simple epidemiological model (wip). In *Proceedings of the Symposium on Theory of Modeling & Simulation-DEVS Integrative M&S Symposium*, pages 1–6, 2013.

Jiaxin Zhang and Derek T Robinson. Replication of an agent-based model using the replication standard. *Environmental Modelling & Software*, 139:105016, 2021.

Victoria Stodden, Jennifer Seiler, and Zhaokun Ma. An empirical analysis of journal policy effectiveness for computational reproducibility. *Proceedings of the National Academy of Sciences*, 115(11):2584–2589, 2018.

Simon Taylor, J.E and Thomas Monks. Covid-19 Computer Simulation Models - Search Strategy 2020/21, June 2022. URL `https://doi.org/10.5281/zenodo.6628395`.

Marion L Penn, Thomas Monks, Anna A Kazmierska, and MRAR Alkoheji. Towards generic modelling of hospital wards: Reuse and redevelopment of simple models. *Journal of Simulation*, 14(2):107–118, 2020.

Georgia M Kapitsaki and Georgia Charalambous. Modeling and recommending open source licenses with findosslicense. *IEEE Transactions on Software Engineering*, 47(5):919–935, 2019.

Georgios Dagkakis and Cathal Heavey. A review of open source discrete event simulation

software for operations research. *Journal of Simulation*, 10(3):193–206, 2016.

Benedikt Fecher and Sascha Friesike. *Open Science: One Term, Five Schools of Thought*, pages 17–47. Springer International Publishing, Cham, 2014. ISBN 978-3-319-00026-8. . URL https://doi.org/10.1007/978-3-319-00026-8_2.

OECD. Making open science a reality. (25), 2015. . URL https://www.oecd-ilibrary.org/content/paper/5jrs2f963zs1-en.

Anastasia Anagnostou and Simon J. E. Taylor. Can open science change the world? *Computer*, 53(10):13–22, 2020. .

Simon J. E. Taylor, Tillal Eldabi, Thomas Monks, Markus Rabe, and Adelinde M. Uhrmacher. Crisis, what crisis – does reproducibility in modeling simulation really matter? In *2018 Winter Simulation Conference (WSC)*, pages 749–762, 2018. .

Simon J. E. Taylor, Anastasia Anagnostou, Adedeji Fabiyi, Christine Currie, Thomas Monks, Roberto Barbera, and Bruce Becker. Open science: Approaches and benefits for modeling simulation. In *2017 Winter Simulation Conference (WSC)*, pages 535–549, 2017. .

Richard M Wood. Unravelling the dynamics of referral-to-treatment in the nhs. *Health Systems*, 10(2):131–137, 2021. . URL https://doi.org/10.1080/20476965.2019.1700764.

Michael Allen, Amir Bhanji, Jonas Willemsen, Steven Dudfield, Stuart Logan, and Thomas Monks. A simulation modelling toolkit for organising outpatient dialysis services during the covid-19 pandemic. *PLOS ONE*, 15(8):1–13, 08 2020. . URL https://doi.org/10.1371/journal.pone.0237628.

Michael Allen and Thomas Monks. COVID19 dialysis service delivery simulation models, April 2020. URL https://doi.org/10.5281/zenodo.3760626.

Amelia Chauvette, Kara Schick-Makaroff, and Anita E. Molzahn. Open data in qualitative research. *International Journal of Qualitative Methods*, 18:1609406918823863, 2019. . URL https://doi.org/10.1177/1609406918823863.

Christina Saville, Thomas Monks, Peter Griffiths, and Jane Elisabeth Ball. Costs and consequences of using average demand to plan baseline nurse staffing levels: a computer simulation study. *BMJ Quality & Safety*, 30(1):7–16, 2021. ISSN 2044-5415. . URL https://qualitysafety.bmj.com/content/30/1/7.

Ian Foster, Carl Kesselman, and Steven Tuecke. The anatomy of the grid: Enabling scalable virtual organizations. *The International Journal of High Performance Computing Applications*, 15(3):200–222, 2001. . URL https://doi.org/10.1177/109434200101500302.

Tony Hey and Anne E. Trefethen. Cyberinfrastructure for e-science. *Science*, 308(5723): 817–821, 2005. . URL https://www.science.org/doi/abs/10.1126/science.1110410.

Ian Bird, Bob Jones, and Kerk F. Kee. The organization and management of grid infrastructures. *Computer*, 42(1):36–46, 2009. .

Franz Barjak, Kathryn Eccles, Eric T. Meyer, Simon Robinson, and Ralph Schroeder. The emerging governance of e-infrastructure. *Journal of Computer-Mediated Communication*, 18(2):1–24, 2013. . URL https://onlinelibrary.wiley.com/doi/abs/10.1111/jcc4.12000.

Ewa Deelman, Dennis Gannon, Matthew Shields, and Ian Taylor. Workflows and e-science: An overview of workflow system features and capabilities. *Future Generation Computer Systems*, 25(5):528–540, 2009. ISSN 0167-739X. . URL https://www.sciencedirect.com/science/article/pii/S0167739X08000861.

Chee Sun Liew, Malcolm P. Atkinson, Michelle Galea, Tan Fong Ang, Paul Martin, and Jano I. Van Hemert. Scientific workflows: Moving across paradigms. *ACM Comput. Surv.*, 49(4), dec 2016. ISSN 0360-0300. . URL https://doi.org/10.1145/3012429.

Anastasia Anagnostou, Simon J. E. Taylor, Derek Groen, Diana Suleimenova, Nana Anokye, Riccardo Bruno, and Roberto Barbera. Building global research capacity in public health: The case of a science gateway for physical activity lifelong modelling and simulation. In *2019 Winter Simulation Conference (WSC)*, pages 1067–1078, 2019. .

Christine S. M. Currie and Thomas Monks. A practical approach to subset selection for multi-objective optimization via simulation. *ACM Trans. Model. Comput. Simul.*, 31(4), aug 2021. ISSN 1049-3301. . URL https://doi.org/10.1145/3462187.

Thomas Monks. Clahrcwessex/bootcomp: v1.0.1, June 2020. URL https://doi.org/10.5281/zenodo.3901489.

Thomas Monks, Alison Harper, Simon Taylor, J.E, and Anastasia Anagnostou. Tommonks/treatment-centre-sim: v0.4.0, June 2022. URL https://doi.org/10.5281/zenodo.6772475.

Barry Nelson. *Foundations and methods of stochastic simulation: a first course*. Spinger, London, 2013.

Executable Books Community. Jupyter book, February 2020. URL https://doi.org/10.5281/zenodo.4539666.

Team SimPy. Simpy 3.0.11. https://simpy.readthedocs.io/en/latest/index.html, 2020.

The Turing Way Community and Scriberia. Illustrations from The Turing Way: Shared under CC-BY 4.0 for reuse, May 2021. URL https://doi.org/10.5281/zenodo.6560477. This work was supported by The UKRI Strategic Priorities Fund under the EPSRC Grant EP/T001569/1, particularly the "Tools, Practices and Systems" theme within that grant, and by The Alan Turing Institute under the EPSRC grant EP/N510129/1.