

The Path to FAIR Research Models: Lessons Learned

Albert J. Kettner¹, Leslie Hsu², Brandon S. Serna²

¹Institute of Arctic and Alpine Research (INSTAAR), University of Colorado Boulder, Boulder CO, 80309, United States

²U.S. Geological Survey, Science Analytics and Synthesis, Denver CO, 80225, United States

5 *Correspondence to:* Albert J. Kettner (kettner@colorado.edu)

Abstract. Numerical modeling of Earth surface processes emerged as an important scientific tool in the late 1960s to mid-1970s, driven by the development of finite element methods in computer science. These advancements, initially applied in civil engineering, enabled scientists to simulate complex geological phenomena. At that time, models were often only described in publications, access was limited to researchers with direct connections to the developers, and the code was rarely documented for reuse, limiting their application beyond the original research context. The FAIR principles (Findability, Accessibility, Interoperability, and Reusability) as applied to data began to take shape in the 21st century with the rise of open science, digital repositories, and standardized data sharing frameworks. In the late 2010s, grassroots movements began to apply some of the FAIRness goals to numerical models. Subsequently, more formalized FAIR model principles were developed that addressed the specific needs of the scientific modeling community, resulting in the formulation of the FAIR principles for research software (FAIR4RS).

In this study, we examine the development and implementation of strategies by two geoscience research infrastructures — the CSDMS (Community Surface Dynamics Modeling System) Model Repository and the U.S. Geological Survey Model Catalog — to enhance the FAIRness of models guided by FAIR4RS. Some of the development and implementation efforts described predate the formalization of FAIR and FAIR4RS principles, making this an ongoing and adaptive process. We evaluate the temporal progression towards increased FAIR4RS alignment across three phases of research infrastructure development: prototype, refinement, and growth & iteration. Although certain principles were more straightforward to implement early in prototypes of the catalog infrastructures, others required broader community collaboration during refinement, and some continue to pose practical challenges in the growth and iteration phase. By tracing these dynamics, our aim is to provide insights that can guide other modeling initiatives in effectively adopting FAIR4RS principles within their communities.

25 **1 Introduction**

The Earth is a highly dynamic system of interacting components from geology, hydrology, ecology, and other geo- and human sciences. Researchers from these diverse disciplines employ quantitative tools, including scientific models, to capture, understand, and predict these interactions (National Research Council, 2007). Formalization of knowledge in these scientific models foster interdisciplinary dialogue and advances scientific progress (Tucker et al., 2022). In this study, the term

30 “model(s)” refers broadly to process-based (mechanistic), empirical or statistical, stochastic, reduced-complexity (conceptual), and hybrid modeling approaches.

Before the widespread use of publishing scientific research products online, source code of developed models were often not accessible to all who could benefit from them or supporting materials. For both technical and cultural reasons, model source code and other details remained with the research groups that created them. This limited sharing of models hindered the ability
35 to build upon existing knowledge and verify findings (Barnes, 2010).

Increased sharing of scientific data was a common topic in workshops and initiatives in the 2010s (Carballo-García and Boté-Vericad, 2022). These discussions led to the development of the FAIR Data Principles, published in 2016 (Wilkinson et al., 2016), where FAIR stands for Findability, Accessibility, Interoperability, and Reusability. Similar initiatives soon after emerged to make scientific models more available and accessible (Crowston, 2008; Treloar, 2014), which over time resulted
40 in adopting the FAIR data principles for models and software, with slight modifications to better suit specific needs of the software community, resulting in FAIR4RS (Barker et al., 2022; Garijo et al., 2022, Wilkinson et al., 2025).

Government agencies, scientific journals, and funding sources have promoted or required open access to newly developed or published software and models (McKiernan et al., 2016). The availability of tools and services, such as code repository platforms (Hahnel and Valen, 2020), have facilitated collaborative coding efforts and helped move the scientific modeling
45 community toward more open and reproducible work. However, many details of FAIR implementation are still left to the model developer. As noted in Barker et al. (2022) contributions from research infrastructure groups used by the model developers can significantly increase the FAIRness of models. So, integrating the technical, disciplinary, and community perspectives through user-centric solutions can enhance effective application of FAIR principles (e.g., the [Actionable FAIR4RS Task Force](https://www.researchsoft.org/tf-actionable-fair4rs/); <https://www.researchsoft.org/tf-actionable-fair4rs/>). Although FAIR4RS offers high-level guidance, its
50 practical implementation depends on reconciling diverse stakeholder needs and contexts—standardized technical approaches alone often lead to fragmented or incompatible implementations (Jacobsen et al., 2020).

The geoscience community can be broadly organized into several major research domains, each supported by representative organizations that coordinate community efforts, develop shared infrastructure, and accelerate scientific progress. These domains include hydrology (e.g., Consortium of Universities for the Advancement of Hydrologic Science, Inc.), Earth surface
55 dynamics (e.g., Community Surface Dynamics Modeling System), geodynamics (e.g., Computational Infrastructure for Geodynamics), and atmospheric science (e.g., National Center for Atmospheric Research). Each of these domains is supported by organizations that foster collaboration, develop community standards and cyberinfrastructure, and engage researchers to advance understanding within their respective fields. This paper details the path of two research infrastructures serving the
60 geoscience user community that have implemented strategies to support model developers in practically implementing the FAIR4RS principles for their models. The Community Surface Dynamics Modeling System (CSDMS) Model Repository (https://csdms.colorado.edu/wiki/Model_download_portal) and the U.S. Geological Survey (USGS) Model Catalog (<https://data.usgs.gov/modelcatalog/>) both serve subsets of the community of Earth process modelers. Note that the CSDMS Model Repository is commonly referred to as a "model repository," but it functions primarily as a catalog, because it does not

65 host model source code directly but instead provides links to externally maintained locations of the model code. The term "repository" has been retained for historical reasons, but we will use catalog from hereon to be more precise.

Overview of the CSDMS Model Catalog. The Community Surface Dynamics Modeling System (CSDMS) team represents a community of researchers that investigate the Earth's surface - the ever-changing, dynamic interface between lithosphere, hydrosphere, cryosphere, and atmosphere. CSDMS supports the modeling of surface processes by developing and disseminating integrated software modules that predict the movement of fluids and the flux (production, erosion, transport, and deposition) of sediment and solutes in landscapes and their sedimentary basins (Tucker et al., 2022). CSDMS promotes interdisciplinary collaboration, advances model interoperability standards, and provides educational resources to support learning and research. These efforts can enhance scientific understanding and inform environmental management, conservation, and policy.

75 The CSDMS model catalog was developed in the late 2000s to serve as a central digital portal for model metadata and references to model source code, and to support broader accessibility and collaboration in Earth surface dynamics modeling. Early efforts focused on collecting metadata through web forms. Initially, more metadata than actual source code was gathered, prompting the implementation of stricter protocols in 2009 requiring open-source licenses and downloadable code. This shift aimed to ensure models were well-documented, reusable, and interoperable (Overeem et al., 2013). The model catalog
80 currently contains 451 models, tools, and components representing nine domains in the Earth and environmental sciences (Fig. 1, Supplemental Table 1).

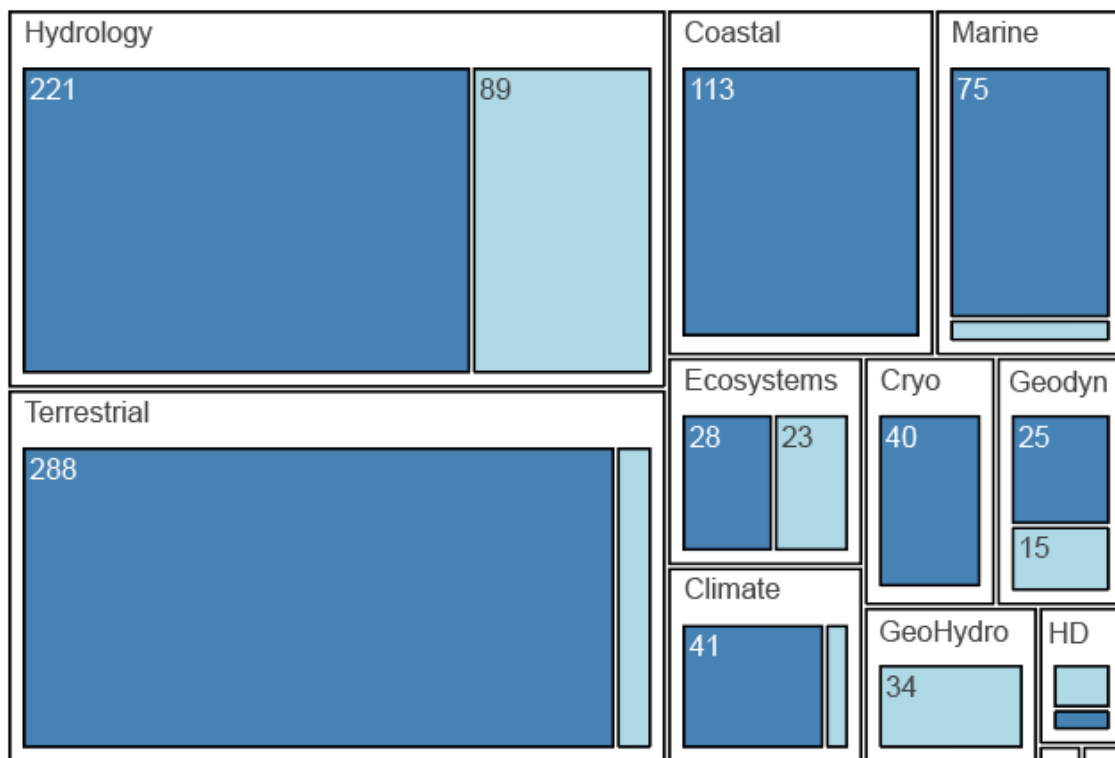


Figure 1: To illustrate the subject matter contents of the CSDMS Model Catalog and the USGS Model Catalog, this figure shows the relative proportion of domain tags (Hydrology, Terrestrial, Coastal, etc.) for the two model catalogs. A single model may have more than one domain tag. Medium-blue boxes correspond to CSDMS tags and light blue boxes correspond to USGS Model Catalog tags. Abbreviations: Cryo = Cryosphere, Geodyn = Geodynamics, GeoHydro = Geologic or Hydrogeologic, HD = Human Dimensions. The two smallest boxes represent Atmosphere and Natural Resources.

85

90 **Overview of the USGS Model Catalog.** The USGS produces and applies models in the areas of earth, water, and biological science. These models help the USGS to use earth, water, biological, and mapping data and expertise to support decision-making on environmental, resource, and public safety issues. Information about models supporting USGS work is spread over a range of websites and formats, which were created at different times and under different agency or governmental policies. In addition, all work performed by USGS Federal employees is subject to Federal Source Code Policy: Achieving Efficiency, 95 Transparency, and Innovation through Reusable and Open Source Software (Office of Management and Budget, 2016).

FAIRness and comprehension of USGS-wide models can be particularly important because of the range of disciplines covered by the USGS and the ability to use these models in combination to address transdisciplinary challenges (Jenni et al., 2017, Kiesman et al., 2021, Wilson et al., 2022). Common knowledge or terminology by one discipline can be unknown to other disciplines within the USGS. To evaluate which models are available for multi-disciplinary issues faced by decision makers for risk, fire, natural resource, safety, and more, FAIRness and model understanding can be made easier when all USGS models are provided in a catalog developed with a USGS-wide perspective.

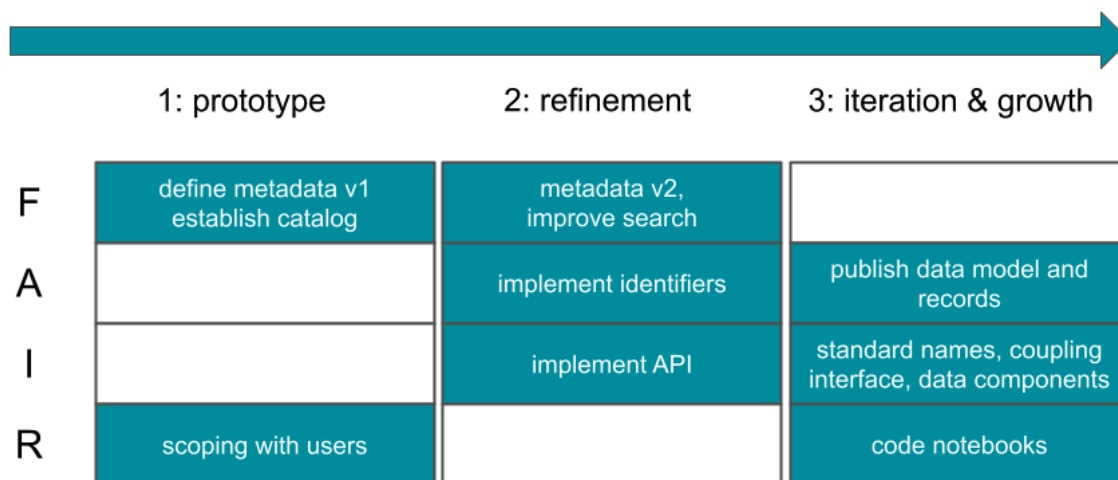
The USGS Model Catalog started in 2020 in its prototype phase as a collection of USGS model-related products in an existing USGS repository system, ScienceBase (<https://www.sciencebase.gov>). Incorporating user feedback, a second version was refined with modified metadata fields, a more customized user interface, and more transparent submission and management features. In the current growth and iteration phase, areas remain for improvement in including content for reusability. The USGS Model Catalog now exists as a searchable catalog that allows users to both contribute and find models across a wide range of earth science domains. It currently contains 191 models representing 12 domains (Fig. 1, Supplemental Table 1.)

This paper is organized into sections following the FAIR4RS principles: "Findability of Models," "Accessibility of Models," "Interoperability of Models," and "Reusability of Models." The "Discussion" details shared challenges and lessons learned in the path to FAIR4RS, some fully or partially resolved, whereas others are yet to be solved. An accounting of the FAIR principles addressed by these two systems illustrates both the simple and more complex ways that FAIR4RS can be increased. The description of features and implementation decisions over time provides two complementary stories, about how research infrastructure can help the modeling community progress down the path to FAIR4RS. The two systems, though overlapping in user audience, differ in their host organization - government versus academia - and this affected the two paths to FAIR4RS. The aim of this paper is to provide insights to guide other modeling initiatives in effectively adopting FAIR4RS principles within their communities. Moreover, the paper demonstrates that the application of these principles can enhance efficiency, transparency, and collaboration in model research and development, thereby advancing both fundamental and applied scientific domains.

2 Increasing FAIRness of models

For both the CSDMS and USGS model catalogs, specific choices were made regarding which of the FAIR4RS principles were partly or fully incorporated into each of the systems, and how they were implemented. For both systems, the implementation of the principles represents a dynamic, evolving process that can be delineated into three successive phases: prototype, refinement, and iteration & growth (Fig. 2).

A path to FAIR models supported by research infrastructure



25 **Figure 2: Schematic diagram showing the three phases on the path to FAIR models, along with specific steps related to the principles Findable (F), Accessible (A), Interoperable (I), and Resuable (R). In phases 1 and 2, metadata v1 and metadata v2 refer to metadata about the models (e.g., Supplement Tables 2 and 3). In phase 3, the data model and records refer to the data model and records for the catalog items (e.g., Serna et al., 2025). API: Application Programming Interface.**

30

In the prototype phase, initial efforts focus on developing the technical mechanisms necessary to operationalize specific FAIR principles, accompanied by active engagement with the community to solicit feedback and assess feasibility. The refinement phase involves the development of a more robust and mature implementation, incorporating community input to ensure relevance, usability, and alignment with best practices. In the iteration & growth phase, incremental updates and modifications are introduced as needed to accommodate new insights, emerging technologies, or evolving community requirements, ensuring sustained alignment with the FAIR4RS principles over time. The paper follows Barker et al. (2022) in the four sections below to detail which principles were implemented for each system and indicate where additional efforts were made to make the models more FAIR.

35

40 **2.1 Findability of models**

Findability of models refers to the discoverability of the model and its associated metadata by both humans and machine-driven systems. For software, including numerical models, Barker et al. (2022) defines the following Findability guiding

principles as essential: software is assigned a globally unique and persistent identifier, where components of the software representing levels of granularity are assigned distinct identifiers, and different versions of the software are assigned distinct identifiers; software is described with rich metadata; metadata clearly and explicitly include the identifier of the software they describe; and metadata are FAIR, searchable and indexable. By including these elements, numerical models become more discoverable and accessible to both humans and automated systems.

Findability was core to the initial development of both the CSDMS and the USGS Model Catalogs - a driving motivation was to enhance the discoverability of models for human users. Over time, the emphasis expanded to include improving machine-driven model discoverability, recognizing the growing importance of automated systems in identifying and utilizing models.

At the inception of CSDMS, it became evident to the product owner (i.e., one of the authors) that merely cataloging the various numerical models by displaying a model list, accompanied by a link to the source code and information about the developer and their affiliation, was inadequate for enhancing their discoverability because of the significant number of available models. Following the online publication of this list, CSDMS implemented a filtered-search functionality, enabling users to refine their queries and identify a more targeted subset of models that align with their specific requirements.

The USGS Model Catalog was developed for USGS researchers on transdisciplinary teams to learn about the current state of modeling and researchers involved in particular topical areas. Searches of the general internet, or of the usgs.gov site, give thousands or dozens of returns, respectively, and the results are in many different formats.

2.1.1 Software is assigned a globally unique and persistent identifier

Assigning globally unique, persistent identifiers, such as digital object identifiers (DOIs), to model source code releases can ensure their long-term findability, proper attribution, and the ability to reproduce research findings, ultimately contributing to a more robust and interconnected scholarly ecosystem (Barker et al., 2022). CSDMS uses Zenodo to assign DOIs to released versions of numerical models. For researchers and model developers, use of GitHub for version control is common (Blischak et al., 2016; Escamilla et al., 2022), and Zenodo provides a seamless interface for archiving model source code. Through this integration, each code release on GitHub can be easily preserved and assigned a DOI, facilitating citation and enhancing the reproducibility of numerical models.

The USGS Model Catalog employs a USGS model identifier that is created and served by the USGS Asset Identifier Service (<https://www1.usgs.gov/identifiers/>) and is a 36-character unique string. The USGS model identifier is not meant to replace any existing identifier but exists to link and increase findability of the many different model assets. The USGS model identifier is assigned to the model concept that encompasses different software versions, implementations in different languages, and applications of a model to different spatial and temporal boundaries. Existing DOIs for USGS software releases (code) or data releases (e.g., inputs and outputs) of the models are related to the model concept identifier and shown together on the model landing page, increasing findability. The model identifier is in the model metadata schema, the title of the metadata file, and the landing page URL (Uniform Resource Locator) in the catalog.

75 2.1.2 Software is described with rich metadata

In response to community feedback, in 2008, CSDMS started developing a set of metadata fields based on the Dublin Core Metadata Terms (<https://www.dublincore.org/specifications/dublin-core/dcmi-terms>) to provide information for users to assess the utility of a numerical model for their research, while minimizing the burden on the model developers to supply the necessary metadata. The metadata fields were designed to be comprehensive yet not overly demanding knowing that model developers
80 might be reluctant to provide metadata if it required substantial time and effort. Over the years minimal changes have been made to the metadata fields to assure continuation and uniformity of available model information (See Supplement Table 3). This rich metadata is separate from the model source code and is stored and maintained by the CSDMS facility through the CSDMS web portal front end in MySQL (<https://mysql.com>) tables.

The metadata for the USGS Model Catalog are described in Supplement Table 2 and in Serna and Hsu (2022). For establishing
85 metadata relevant for the scope of USGS models, the team drew from several existing efforts, including CSDMS, Open Modeling Foundation (OMF) (Barton et al., 2022), CodeMeta (Jones et al., 2023), and OKG-Soft (Garijo et al., 2019). These fields were then tested across the range of USGS models (numerical, geospatial, etc.) to determine a set of fields for a basic profile, and a set of fields for an advanced profile. Using an "off the shelf" metadata schema with no modifications was difficult due to USGS-specific requirements. The metadata are stored as flat JSON-schema compliant files. The metadata are also
90 defined as a Python class (Hunt, 2023), using the Pydantic library (Colvin et al., 2025), which allows automation of several of the Model Catalog components including validation and web form entry.

2.1.3 Metadata are FAIR, searchable and indexable

For making models more findable through the web, a common vocabulary as suggested by Schema.org (Ronallo, 2012) is incorporated for the USGS and the CSDMS model catalogs. [Schema.org](https://schema.org) can provide structured data formats like JSON-LD, Microdata, and RDFa, allowing markup of web pages in a way that is recognized by major search engines such as Google,
95 Bing, and Yahoo. These structured data help search engines more accurately interpret web content, and can lead to improved search result presentation and richer user experiences. By promoting consistency and interoperability across different web platforms, Schema.org enhances search results with rich snippets – more information than a standard result – which can make web search result listings more informative and visually appealing (Payne and Veryhey, 2022).

To enable searchability and indexability of metadata for machines, both the USGS and the CSDMS model catalog contain
00 Application Programming Interfaces (APIs). These APIs allow programmatic exchange of information about the models. They also enable communication, for example, between the CSDMS repository and the USGS catalog, allowing for data sharing and functional integration. Additionally, they promote modularity and reusability, enhancing code maintainability and efficiency. APIs can accelerate development and foster innovation by leveraging existing services. This enables catalog
05 developers to integrate diverse functionalities to ensure that information provided by the metadata is consistent over multiple

catalogs. APIs can also facilitate ecosystem growth by allowing third-party developers to extend platform functionality and provide standardized access to metadata (Taylor, 2016).

10 The USGS model metadata, which are publicly released and fully accessible as flat JSON files, can be queried through a REST API and GraphQL API (Serna et al., 2022). CSDMS has enabled API functionality for its website and associated databases by
15 implementing the Mediawiki extensions Cargo (<https://www.mediawiki.org/wiki/Extension:Cargo>) and Semantic MediaWiki (SMW; Krotzsch et al., 2006). Additional information on how to use the CSDMS API is provided at https://csdms.colorado.edu/wiki/Web_APIs. For example, the following API query retrieves a JSON-formatted output listing
10 models that were developed using the Python programming language:
https://csdms.colorado.edu/csdms_wiki/api.php?action=askargs&conditions=Model:%2B%7CProgramming_language::Python&printouts=Source_csdms_web_address|Source_web_address%20¶meters=limit=10&format=json.

2.2 Accessibility of models

Accessibility in the FAIR software context means being able to be reached or retrievable. It does not mean the quality of being easily understood or used (Chue Hong, 2022). Barker et al. (2020) defines the following guiding principles for accessibility:
20 software is retrievable by its identifier using a standardized communications protocol, where the protocol is open, free, and universally implementable, and the protocol allows for an authentication and authorization procedure, where necessary; and metadata are accessible, even when the software is no longer available.

Accessibility of model information supports end users, because it enables them, once a model has been identified, to determine how the model and its source code can be obtained, executed, or integrated into their work. Moreover, even if the model itself is no longer available, the associated metadata continues to provide valuable insight into the model's functionality, scope, and
25 intended applications.

2.2.1 Software is retrievable by its identifier using a standardised communications protocol

Both catalogs allow for the retrieval of models over HTTPS (Hypertext Transfer Protocol Secure) via DOIs that are associated with the model software repositories. Authentication and authorization are currently not needed to access the catalogs, because the information exposed is free and open.

30 2.2.2 Metadata are accessible, even when the software is no longer available

For both the CSDMS repository and the USGS Model Catalog, metadata remain accessible if the source code is no longer available. Model source code can become unavailable due to the programming languages, execution platforms, and delivery sites evolving such that the model source code becomes out of date, obsolete, or no longer publicly available. Repositories with model source code that were once public can be deleted or made private. For example, in 2019 the USGS changed its
35 policy such that new code could only be officially released on GitLab after new review and approval processes, and could only

continue to be publicly developed on GitHub if it went through a documentation, review, and approval process. This caused many repositories under active development to be transferred to a private site.

40 The USGS Model Catalog accomplishes the persistent accessibility of model metadata by publishing the model metadata files in a USGS software release of a git repository (Serna et al., 2022a). Metadata are stored as JSON files and follow a published metadata schema (Serna and Hsu, 2022b). The publication of the metadata files and schema has multiple benefits. One benefit is that anyone can take the JSON metadata files and build a tool around them, even if the USGS Model Catalog website ceases to exist. Another benefit is that the metadata schema can be used or built on by other efforts that want to catalog models. The persistent accessibility of model metadata through the CSDMS portal is managed differently than the USGS catalog. Source code repositories in the CSDMS catalog are primarily maintained by community members. The CSDMS team, unlike 45 government agencies, does not have the authority to require metadata in the source code repository before a software release. For that reason, model metadata is stored separately from the source code, and CSDMS maintains metadata records once they are created. Developers have the option to include a status update to the metadata that, for example, indicates that the source code is no longer available, or that the model has merged into another model.

2.3 Interoperability of models

50 Interoperability of models can have different interpretations (e.g. conceptual, operational, technical) in our user communities. Following the FAIR4RS framework, interoperable models should be able to "communicate with other software" or be coupled through exchanging data and/or metadata, and/or through interaction via e.g. APIs. Barker et al. (2022) indicate the following guiding principles for interoperability: software reads, writes and exchanges data in a way that meets domain-relevant community standards; and software includes qualified references to other objects. Interoperability is often one of the later, and 55 more advanced FAIR4RS principles to be implemented, as also is the case for the two catalogs discussed in this paper. For modeling communities like CSDMS, interoperability is especially important because scientific questions increasingly require coupling multiple models (e.g., linking hydrology, sediment transport, and coastal dynamics) or integrating models with observational datasets. Without interoperability, combining models becomes ad hoc, error-prone, and harder to reproduce (Belete et al., 2017). With interoperability, models can be more easily assembled into workflows, reused across projects, and 60 compared systematically, which accelerates scientific discovery and reduces redundant effort.

2.3.1 Software reads, writes and exchanges data in a way that meets domain-relevant community standards

A domain-relevant standard is an accepted standard that addresses the needs of a given community or communities (Chue Hong, 2022). The relevant communities for CSDMS and USGS include landscape evolution modelers, hydrologists, biologists, oceanographers, ecologists, and other Earth and biological scientists.

65 Within the CSDMS community, a consensus emerged early on that conducting simulations across diverse scientific domains may be better achieved not through the development of a few large models, but rather by advancing model-linking techniques capable of coupling existing domain-specific models. As such, over time, CSDMS has developed several domain-relevant

community standards for making models more interoperable, including the “Basic Model Interface (BMI)”, “Standard Names”, and the “Python Modeling Toolkit (PyMT)”.

70 The Basic Model Interface (<https://bmi.csdms.io/>) is a set of standard query and control functions that, when added to a model code, make that model easier to couple with other models (Peckham et al., 2013). A model with a BMI exposes the same set of functions, no matter which language the model is written in. Examples of functions to exchange data between models are `get_value` and `set_value`. These functions can be called every timestep or while running over several timesteps (for example `update_until`). Coupling can be performed between data and models, where a model can have data access through BMI plug-
75 and-play components to integrate datasets with models (Gan et al., 2024).

CSDMS developed a set of Standard Names, that provide a comprehensive set of naming rules and patterns that are modeling domain agnostic (Peckham, 2014). This ontology framework addresses the key challenge of mapping inconsistent input and output variable names, allowing one model’s output to be the input for another model. Standard names mapping means that models do not need to be rewritten to rename variables but rather map them onto the standard names through a lookup table.

80 For the CSDMS catalog, the coupling of models occurs within a framework. CSDMS developed the Python Modeling Toolkit (PyMT), a framework that allows the user to integrate and run models using the BMI Python interface and standard names (Hutton and Piper, 2020). PyMT is unique in that it enables coupling between different models that might be written in different languages (such as C, C++, Fortran, Java, Javascript, Julia, Python) by providing a standardized Python interface. It promotes interoperability between models which makes it easier to simulate complex systems that involve multiple interacting processes.

85 An example of a coupled model system is provided by Ashton et al. (2013), where a coastline evolution model (CEM) is coupled with a fluvial dynamics model (HydroTrend) to investigate how fluctuations in sediment input rates due to climate change may affect the plan-view delta morphology and evolution.

The USGS Model Catalog has not developed its own capabilities for coupling models, however, it plans to highlight examples of USGS model coupling. There are models in the USGS Model Catalog for which a BMI has been developed, and documented
90 on the model landing page, such as the PRMS ([Precipitation Runoff Modeling System](#)) model.

Both catalogs have mapped their metadata fields to the codemeta project (Jones et al., 2022). By aligning catalog metadata with CodeMeta, the catalog adopts a community-endorsed schema that facilitates exchange between different repositories, registries, and discovery platforms. Therefore, this effort increases the cross-platform compatibility of the exchange of data and/or metadata via machine-to-machine interaction. For example, it will make it easier for models in both the catalogs to be
95 integrated into other FAIR-aligned infrastructures.

2.3.2 Qualified references to other objects

The objects that are referenced in both catalogs are digital products such as data, publications, sites, and people. The software references are covered under the Reusability principle.

In the USGS Model Catalog, models are linked to other software, data, publications, web sites and people, via identifiers
00 (Table 1). For all of these entities, only the identifiers are stored in the Model Catalog metadata, and the current information

linked to those identifiers are resolved dynamically instead of being stored in the system. The related objects are organized on the data, software, publication tabs, and in a modeler or person view, allowing quick comprehension about the relationships between these assets.

05 CSDMS uses the name of a model as a unique identifier, to link to other objects (Table 1). For example, references and their citations are linked by including the name of a model, or multiple models to a reference. References that describe or apply a model are queried in real-time and displayed on a model web page.

Table 1. Qualified references of other objects

| Identifier name | System linked | Linked object | CSDMS/USGS |
|-----------------|-----------------------------|--|------------|
| DOI | USGS Publications Warehouse | Manuscript publications | USGS |
| Paper ID | Semantic Scholar | Citations and citation count | CSDMS |
| DOI | ScienceBase | Data releases, including model application data releases | USGS |
| DOI | Zenodo | Data publications or software repositories | CSDMS |
| DOI | GitHub | Software repositories | CSDMS/USGS |
| DOI / tag URL | USGS GitLab | Software repositories | CSDMS/USGS |
| ORCID | ORCID / USGS Staff Profiles | People | USGS |
| model ID | USGS Model Catalog | Models | USGS |
| metadata ID | Science Data Catalog | Metadata for data releases | USGS |

10 2.4 Reusability of models

Reusability means that the models are both usable - can be executed, and reusable - can be understood, modified, built upon, or incorporated into other software. The guiding principles for reusability include: software is described with a plurality of accurate and relevant attributes, where software is given a clear and accessible license, and software is associated with detailed provenance; software includes qualified references to other software; and software meets domain-relevant community standards (Barker et al., 2020). In addition to these practices, both of our experiences identify additional principles for reusability: use of executable notebooks and use of model frameworks.

15

Reusable models offer meaningful benefits to scientific research and engineering. A reusable model saves time by reducing the need to develop new models from scratch, improves consistency and reliability by facilitating repeatable results, fosters collaboration, and encourages standardization across teams for interdisciplinary use. An example of model reusability in action is the LandLab, a Python toolkit for modeling earth surface processes (Barnhart et al., 2020; Hutton et al., 2020a). Using model components of LandLab, Campforts et al. (2020) developed the landscape evolution model Hylands to investigate the influence of landslides and their associated sediments on landscape evolution. This effort involved creating three new LandLab components, reusing two existing components (FlowAccumulator and ChannelProfiler), and leveraging five service components, such as functionality for reading NetCDF data files. While we cannot quantify the exact time savings, the reuse of existing LandLab components and services likely reduced development time substantially by eliminating the need to reimplement and validate core functionality. This enabled the authors to focus their effort on developing new process representations rather than rebuilding standard modeling infrastructure.

2.4.1 Software is described with a plurality of accurate and relevant attributes

Relevant attributes can be determined by repositories, and by communities who create and reuse software. Plurality means that, where possible, multiple terms for the same, similar, or overlapping concepts should be provided to enable the broadest possible reuse (Chue-Hong, 2022).

Software is given a clear and accessible license. Model licenses promote reusability because they define the legal terms under which a model can be accessed, modified, and redistributed. A license provides guidelines on model usage rights, restrictions, and attribution. This ensures that developers and users can share, adapt, and build upon models within a structured and permissible framework, while avoiding unintended consequences of sharing code openly. Licensing is a complicated topic with nuances related to public domain, waiving copyright ownership, implications for patent rights for code built on originally open code, what the absence of an explicit license means, and more. A resource for learning more is the Open Source Initiative (OSI) site (opensource.org/licenses).

For models developed by USGS scientists, an appropriate open source license applied to model source code is the Creative Commons zero (CC0) license ([USGS Office of Science Quality and Integrity FAQ 137](https://www.usgs.gov/science-quality-integrity/faq-137)). Models licensed under CC0 will become part of the public domain allowing anyone to use them without restrictions from the copyright holder (<https://creativecommons.org/public-domain/cc0>). To implement this, USGS developed a default license.md for any published software repository. See the example from the MODFLOW6 (USGS, 2026). The clear and accessible license ensures the legality of re-using the USGS model code.

The source code included in the CSDMS model catalog must be licensed but given CSDMS's role in representing a diverse community from a variety of organizations, there is flexibility in the choice of license. Although CSDMS does not enforce specific licensing standards, it recommends that model developers use the GPL v3 license or a less restrictive alternative (<https://www.gnu.org/licenses/gpl-3.0.en.html>). The GPL v3 license offers several advantages, such as fostering collaboration,

50 protecting the intellectual property of developers, and promoting the free use of numerical models
(<https://csdms.colorado.edu/wiki/License>).

Software is associated with detailed provenance. To enhance model reusability, model developers should maintain detailed provenance information linked to their source code. This involves documenting the model's origin, development history, and any modifications or updates it has undergone ([NIST SP 800-53 Rev. 5](#)). The CSDMS community, rooted in academia, often addresses aspects of model provenance—such as the purpose and motivation for developing a model—through associated
55 scientific journal publications. Additionally, CSDMS advocates for the use of version control software (e.g., Git) during model development or modification to capture other critical aspects of provenance. The USGS Model Catalog does not have specific guidance for including provenance, though this characteristic is made possible for code repositories in the USGS GitLab Department of the Interior and GitHub systems.

Examples of how provenance is documented:

- 60
- CSDMS: [LandLab](#) (see changelog.md file).
 - USGS: [MODFLOW6 and related programs website](#) and is an example for a very large model project.
- USGS: [StreamMetabolizer README](#) (Example for a smaller model project).

2.4.2 Software includes qualified references to other software

Software typically relies on other software components, such as libraries, modules, or external packages, to function properly.
65 These dependencies can be cumbersome to install or integrate. It is a good practice for model developers to provide detailed documentation regarding dependencies such as up-to-date Makefiles, environment.yml, requirements.txt, and README documents. This supports potential model users that have less experience with a model to execute a model simulation on their platform of preference. Following good coding practices, this information is included in the source code, consequently, CSDMS does not include these detailed dependency specifications as part of its metadata records.

70 References to other models or tools in the USGS Model Catalog are provided in the "Related ModelCatalog item" field, which links to other pages in the catalog for models that have been coupled or that are related pre- or post-processing tools. For example the [MODPATH](#) tool is linked to the model, MODFLOW6. MODFLOW6, a groundwater flow model, produces output that can be analyzed by the MODPATH particle-tracking post-processing tool (Langevin et al., 2017). The USGS catalog also plans to pilot collection pages that group model pages according to topics that the community requests, and links software to
75 similar software for comprehension.

Although the absence of this information does not affect the ability to correctly execute a model, the CSDMS catalog provides metadata fields where developers can specify software dependencies required for pre- or post-processing. In these fields *Pre-processing software needed* and *Post-processing software needed*, developers may document additional tools required to, for example, format input data appropriately or to facilitate the visualization and analysis of model outputs.

80 2.4.3 Software meets domain-relevant community standards

The spirit behind this principle is that practitioners are aware of what others are doing and using in the community (e.g., choice of programming language, standards for testing, usage of file formats that enable reuse).

The use of committees and working groups to help identify community standards helps to achieve this especially if the community includes many scientific domains with a variety of norms and practices. For example, the identification of the best terminology used to tag and describe models - one group of modelers may prefer "numerical models" while another prefers
85 "mathematical models", or "physics-based models" and "mechanistic models."

To support this principle, both catalog systems were developed in close communication with their user groups. For the CSMDS catalog this is achieved via CSDMS Annual Meetings, webinars, forums, and newsletters. Trainings and presentations on topics like unit-testing and standard names also supports developing and sharing community standards. CSDMS is guided by
90 an Executive Committee of practicing community members. For the USGS Model Catalog, community standards are incorporated into the USGS Model Catalog through a working group that was originally sourced through the USGS modeling community, but is currently open to all through the Community for Data Integration modeling collaboration area.

In addition to implementing the FAIR4RS principles, both the CSDMS and USGS initiatives have extended upon the reusability principle to strengthen the practical usability and long-term sustainability of research software. As indicated below,
95 practical usability is advanced through mechanisms that enable the community to contribute executable notebooks, thereby promoting reproducibility and transparency. And long-term sustainability is supported by providing standardized simulation environments that facilitate integration and reuse of models (PyMT framework).

2.4.4 Additional guiding principle: Software is in an executable notebook

Executable code notebooks have become a common tool that scientists use to share and build upon different kinds of code
00 (Kluyver et al., 2016). Both CSDMS and USGS contain code notebooks and supporting infrastructure to make models more usable, executable, and reusable. These notebooks can be copied onto a local machine and run in a variety of ways, such as utilizing plug-ins to common development environments (like vscode), running on shared infrastructure run by CSDMS or USGS, or running on the commercial cloud (Binder or Google Colab).

Notebooks are a way to become familiar with running a model. Required dependencies of code can be integrated into the
05 simulation environment through the notebooks. Additionally, they provide a user interface that can make it easy for new users to execute code with accompanying documentation and visualizations. Lastly, best practices for code and documentation can be shared in notebooks in a usable manner.

Examples:

10 Pleasant Lake worked Flopy example: An example of the use of the Flopy Python Package (Bakker et al., 2018) to assemble a MODFLOW 6 groundwater flow model with external array input and advanced boundary conditions.

Leaf, A.T. and Fienen, M.N. (2022). Pleasant Lake worked Flopy example, version 0.1, U.S. Geological Survey Software Release, 2022. <https://doi.org/10.5066/P9EFHF9H>.

15 CSDMS community notebooks (includes numerous example notebooks):
https://csdms.colorado.edu/wiki/Labs_portal.

2.4.5 Additional guiding principle: Software uses a model framework

20 Similar to code notebooks, model frameworks make it easier to run and manipulate models without needing in-depth knowledge of the underlying code or languages in which the models are written. In this study, a modeling framework is defined as a systematic method for developing, running, and evaluating models within and across disciplines, often providing guidelines and tools. Frameworks can include additional advances, for example, the capability of coupling models.

CSDMS has developed the PyMT framework (Hutton and Piper, 2020) which enables users to run a model, or couple model configurations on different platforms and environments, enhancing the portability of models. There exists several frameworks, and there have been initiatives to explore the ability to run the same set of coupled models on different frameworks like OpenMI (Gregersen et al., 2007 and Harpham et al., 2019), OpenGMS (e.g. Zhang et al., 2021), and PyMT.

25 An example of a USGS-developed framework is [HyTEST](#), the Hydro-terrestrial Earth Systems Testbed project. The goal of HyTEST is to establish a long-term framework for governance, model benchmarking, evaluation, and evidence-based decision making through collaborative research. The framework provides resources, information, and tools for a computational testing environment that can accelerate and improve the process of hydrologic model development by the water-science community.

3 Discussion

30 Above is a description of how two research infrastructures, developed by CSDMS and by the USGS, can help make numerical models more FAIR, following Barker et al. (2022). Making models more FAIR has a direct effect on the scientific modeling community, as it forms the basis for continuous development (Schuwirth et al., 2019). The implementation of the FAIR guiding principles for scientific models has progressed over several years for both initiatives (Fig. 2, Table 2). The evolution of both research infrastructures from initial version to current version is significant, regarding the metadata, user experience, and capabilities. The changes were driven by user feedback and trends in technologies.

35 Table 2. List of the FAIR4RS principles (Barker et al., 2022) and a comparison of the stage at which each catalog addressed the principle.

| FAIR4RS Principle | CSDMS | USGS Model Catalog |
|---|-------|--------------------|
| F: Software, and its associated metadata, is easy for both humans and machines to find. | | |

| | | |
|--|-----------------------|-------------------------------------|
| F1. Software is assigned a globally unique and persistent identifier. | 2: refinement | 2: refinement |
| F1.1. Components of the software representing levels of granularity are assigned distinct identifiers. | 2: refinement | 3: iteration & growth |
| F1.2. Different versions of the software are assigned distinct identifiers. | 2: refinement | 3: iteration & growth |
| F2. Software is described with rich metadata. | 3: iteration & growth | 1: prototype, 3: iteration & growth |
| F3. Metadata clearly and explicitly include the identifier of the software they describe. | 2: refinement | 2: refinement |
| F4. Metadata are FAIR, searchable and indexable. | 3: iteration & growth | 3: iteration & growth |
| A: Software, and its metadata, is retrievable via standardised protocols. | | |
| A1. Software is retrievable by its identifier using a standardised communications protocol. | 3: iteration & growth | 1: prototype |
| A1.1. The protocol is open, free, and universally implementable. | 3: iteration & growth | 1: prototype |
| A1.2. The protocol allows for an authentication and authorization procedure, where necessary. | 1: prototype | 3: iteration & growth |
| A2. Metadata are accessible, even when the software is no longer available. | 3: iteration & growth | 3: iteration & growth |
| I: Software interoperates with other software by exchanging data and/or metadata, and/or through interaction via application programming interfaces (APIs), described through standards. | | |
| I1. Software reads, writes and exchanges data in a way that meets domain-relevant community standards. | 2: refinement | 2: refinement |
| I2. Software includes qualified references to other objects. | 2: refinement | 1: prototype |

| | | |
|---|-----------------------------|-------------------------------------|
| R: Software is both usable (can be executed) and reusable (can be understood, modified, built upon, or incorporated into other software). | | |
| R1. Software is described with a plurality of accurate and relevant attributes. | 1: prototype, 2: refinement | 1: prototype, 3: iteration & growth |
| R1.1. Software is given a clear and accessible license. | 3: iteration & growth | 1: prototype |
| R1.2. Software is associated with detailed provenance. | 2: refinement | 3: iteration & growth |
| R2. Software includes qualified references to other software. | 1: prototype | 2: refinement |
| R3. Software meets domain-relevant community standards. | 3: iteration & growth | 1: prototype, 3: iteration & growth |

40 In this section, we analyze the main obstacles encountered throughout the development process and evaluate their influence on our implementation strategy. For each identified challenge and lessons learned, we explicitly indicate the relevant FAIR principle(s) involved, thereby underscoring the practical implications for the development of FAIR-aligned research infrastructure. Additionally, we highlight prospective next steps.

3.1 Challenges and lessons learned

3.1.1 Balancing community effort with descriptive metadata (F)

45 Improving model findability depends on the provision of complete and well-structured descriptions for all relevant metadata fields. However, in research culture, this documentation can be perceived as an added effort with limited direct impact on research outcomes, leading to limited incentives for thorough completion (Huang et al., 2025). Striking the right balance between making models easily discoverable for potential users and minimizing the effort required from developers to describe their models may be challenging. Requiring too much information can discourage submissions, whereas too little hinders model discoverability. Our experience with developing the two cataloging efforts highlights that defining a minimal set of required metadata fields (Supplement Tables 2 and 3) is important and still can encourage developers to provide more detailed information, for example, through the capability of add-on modules. Both catalogs rely on model developer submissions, but the CSDMS and the USGS catalog have had periods where a small group of catalog team members were responsible for submitting descriptive metadata. Although this is one way to increase content metadata when resources allow, both teams found that information from the model developers themselves can be more comprehensive. In the future, using auto-generated descriptions may be part of the solution, for example through the use of large language models (LLMs, e.g., Singh et al., 2025).

55

3.1.2 Keeping the content in the catalog current and comprehensive (F, A)

New scientific models are continuously developed, and users may expect a level of completeness maintained in the catalog to support findability and accessibility. Community catalog contributions of content such as new model source code, associated publications and datasets, and up-to-date points of contact support maintaining the catalog's currency and comprehensiveness. This means that engagement with the CSDMS and USGS user communities to solicit content can lead to more complete catalogs. However, sustained catalog contributions may require the implementation of effective incentive mechanisms. Both initiatives provide examples and guidance on how to submit model information to the catalogs, while also outlining the scientific and practical benefits. Beyond increasing model visibility, submission of model information enables the integration of additional resources such as associated publications for the CSDMS repository, thereby enhancing the model's scientific impact and discoverability.

Having a method of updating catalog content that does not rely on only community contributions and manual updates is a feature that both development teams (i.e., the authors) have heard requested, but is still in the exploratory stage for both catalogs. For these more automated methods, recurring use of APIs or other feeds from relevant systems, such as the Git code repositories themselves, could retrieve information that could be considered for addition to the catalog. The USGS Model Catalog team (i.e., two of the authors) has experimented with new content feeds from internal publications, software, and data systems. From this experience, the team concluded that human review is still necessary to supplement these automated feeds.

3.1.3 Maintaining a technologically current catalog application (F, A)

Maintaining and evolving a model catalog for findability and accessibility of models is a complex challenge that requires resources to sustain data management, infrastructure, and technological advancements. A custom-built infrastructure can be challenging to maintain for a small team comprised of science subject-matter experts rather than software engineers, which is largely the case for both the CSDMS and Model Catalog teams (i.e., the authors).

One way both initiatives are working to minimize this challenge is by choosing to use open-source software tools that are supported by a large community. For the CSDMS catalog, wikimedia, and for the USGS catalog, pydantic, JSON schema, vue.js, and fastapi are tools that have been used. Using open-source tools supports adoption, sustainability, and evolution of the software to incorporate technological advances. Open-source tools that are adopted by a large and diverse community also often makes the infrastructure software more user friendly (Bonaccorsi and Rossi, 2003). For projects like these, open-source software can reduce the costs of sustainability of proprietary software into the future.

3.1.4 Using clear language to increase comprehension (F, I, R)

Certain terms possess multiple meanings that are not readily distinguishable and may lead to ambiguity, particularly for users originating from different disciplinary backgrounds. For example, the term "application" can refer to an "application of a model software to a particular spatial and temporal bound" or to a "web application site that displays the model interface." Even the

word "model" can be a source of ambiguity, because the distinction between a numerical process-based flow model, a species population model, a finite element fault model, and a mineral deposit model, and whether they belong in the same catalog, is unlikely to be clear to all catalog users.

Besides words with ambiguous meaning, disciplinary jargon can hinder the FAIRness and interoperability of numerical models (Guizzardi, 2020, Specht et al., 2024). Better understanding the meaning and differences between models can increase the number of tools available to scientists, especially when addressing interdisciplinary and team-science goals. When integrating models to enable programmatic data exchange, it is essential to ensure that one model's output is correctly identified and interpreted as a specific input variable for another model, with consistent units. To address this challenge, CSDMS has developed and implemented the CSDMS Standard Names (Peckham, 2014). This cross-discipline naming convention standardizes variable nomenclature, allowing models and datasets from diverse contributors to be seamlessly coupled in a plug-and-play manner, facilitating the creation of composite models. A glossary that can be easily updated by users to show multiple meanings of terminology used in the systems is a method that helps increase FAIRness.

3.1.5 Acknowledging researcher incentives around identifiers (A)

Seemingly simple tasks for improving FAIRness, for example by introducing new norms like assigning unique DOIs to scientific models, may meet resistance if it contradicts researcher evaluation and promotion efforts. Around 2011, the CSDMS team (including one of the authors) introduced a unique identifier in the form of a DOI for model source code. This was expected to motivate modelers to include model metadata in the CSDMS catalog. The rationale was that a DOI would make the source code permanently accessible, citable, and would provide modelers with due recognition (Mabile et al., 2025). Katz et al. (2019) highlight, among other challenges, the lack of a public mechanism for obtaining DOIs for unversioned software packages. Building on this, one of the authors observed members of the CSDMS community expressed hesitation to submit metadata under the requirement to assign a DOI to their source code. This reluctance stemmed from the fact that source code was often not valued by academic departments to the same degree as traditional journal publications (Puebla et al., 2024). As a result, modelers were concerned that a citable source code DOI might detract from citations to a formal model description or implementation paper. Consequently, to accommodate these concerns, the inclusion of a DOI for model source code versions was made optional in the CSDMS catalog. DOIs are required for official USGS software releases (USGS, 2019), and efforts are ongoing to increase the value of software and data products in the evaluation of research staff. However, the USGS catalog team (i.e., the authors) still sees instances where model documentation encourages citation of a peer-reviewed manuscript instead of a software release, presumably because the manuscript citation will have greater weight. Although there has been resistance due to implications for career evaluation, there are several initiatives working to evolve research evaluation in academia and other sectors and bring greater weight to research product citations besides manuscripts (Puebla et al., 2024, Mabile et al., 2025).

3.1.6 Keeping integrated models synced with latest development (I)

20 Increasing interoperability by incorporating a framework like the Basic Model Interface (BMI) into models requires active involvement of the model developers. Otherwise, model source code can quickly become out of sync, with an out-of-date model integrated into the framework.

The CSDMS catalog team aimed to minimize intrusiveness in implementing BMI functions, but some modifications to the model's source code are necessary (Peckham et al., 2013). Collaborating closely with the developers and securing their
25 commitment to maintain BMI compatibility during the entire process, not just at the end, can ensure the BMI remains integrated as the model evolves. Without this cooperation, divergence between the integrated BMI version and the development version of the model could result in two different versions of the model.

3.1.7 Accommodating the diversity of models (I, R)

The models hosted in the CSDMS and the USGS Model Catalogs exhibit significant variability due to the scope of their user
30 communities, which can complicate efforts to standardize their representation. These models can be categorized into various types, each tailored to specific research objectives, data availability, and scales of system understanding. Models may be process-based, conceptual, deterministic, stochastic, empirical, dynamic, static, statistical, or hybrid. Models also differ in their operational requirements such as compatibility with operating systems, reliance on specialized hardware, programming languages, and licensing restrictions. These variations further extend to data inputs and model outputs which often lack
35 standardized formats.

Both teams have initiated efforts to capture the diversity of models and filter through them by adding metadata fields that provide better resolution for the user, for example, model type (Serna and Hsu, 2022), spatial dimensions (e.g. https://csdms.colorado.edu/wiki/Property:Spatial_dimensions) and spatial extent (<https://csdms.colorado.edu/wiki/Property:Spatialscale>). These metadata fields can help potential users assess whether a model
40 aligns with their needs by understanding its unique characteristics. During the initial stages of metadata field development, CSDMS community members critically evaluated the proposed fields and provided recommendations, which were incorporated to improve the capacity of the metadata to capture informative and relevant model descriptions, thereby accommodating the breadth of modeling practices across scientific domains. The USGS data model for describing the scientific models can be extended with additional modules, which would allow different fields for different types of models.

45 3.1.8 Encouraging and informing code review (R)

Model source code reviews facilitate a host of benefits including scientific accuracy, code quality, reliability, and reproducibility (Bachelli and Bird, 2013). However, code reviews do take time, and both development teams (i.e., the authors) have heard users remark that they are not sure what a code review entails.

Initially, CSDMS aimed to establish code reviews by scientific domain, identifying key community models for evaluation, defining the appropriate scope of review, and seeking volunteers to conduct these assessments. However, this approach proved overly burdensome for the community, leading to its discontinuation. CSDMS now relies on the JOSS journal, the Journal of Open Source Software (Diehl et al., 2025), which maintains a dedicated editorial team and expert reviewers who volunteer for code review. Once a model paper is accepted by JOSS and documented in the CSDMS catalog, the model automatically receives a digital “code reviewed” certification, acknowledging its vetted status. This demonstrates an instance of leveraging different existing infrastructure and community knowledge to gain the benefits of code review. CSDMS encourages model developers to undergo code review by tagging those models as ‘code reviewed’ models and ranking them to the top in the CSDMS model catalog.

For the USGS, code review is done according to guidelines set by Instructional Memorandum OSQI 2019-01, Review and Approval of Scientific Software for Release (USGS, 2019) and further explained by the Fundamental Science Practices FAQs on Software (USGS, 2025). To help decrease the barrier of being unsure about the review process (Gomes et al., 2022), sharing of existing checklists, tips, and other resources developed by individual groups that supplement the USGS policy can increase knowledge and efficiency of the code review process. CDI-Software, a community of practice around software development, hosted by the Community for Data Integration (Hsu et al., 2022), can help educate and share templates that support the USGS review requirements.

65 **3.1.9 Measuring the effectiveness of features for FAIR (F, A, I, R)**

Both CSDMS and the USGS have taken steps to increase the FAIRness of scientific models in their communities. But, how can the impact of these actions be measured? Assessing effectiveness of these FAIR-enhancing actions can support community maintenance and user support and inform the strategic prioritization of resource-constrained teams. Funders may also be interested in metrics about the FAIRness of the products that they support.

70 Certain efforts can be quantitatively assessed, for example, CSDMS can measure the use of the BMI to facilitate model coupling by counting the models that have implemented a BMI, or by counting publications reporting coupled models that utilize the BMI. However, measuring the impact of other implemented features might be more challenging, like the implementation of schema.org tags to make the model catalogs more machine-readable. Both catalogs implemented the tags, but following the implementation of schema.org tags, neither development team (i.e., the authors) could find any existing frameworks or methodologies to assess their impact or effectiveness. Although others can now build upon the implementations, 75 there is no methodology developed to assess if others take advantage of this effort.

3.2 Catalogs in support of FAIR compliance

The two model catalogs described here play a role in helping code developers and users comply with journal FAIR requirements. By providing standardized metadata, persistent identifiers, and highlighting structured archiving workflows (for example through code notebooks), catalogs help models meet the transparency and reproducibility standards increasingly 80

required in scientific publishing. Representatives of the product teams of both model catalogs are part of the SciCodes Consortium (<https://scicodes.net>), which aims to foster collaboration and standardization among scientific software registries to improve the discoverability, interoperability, and reuse of research software across disciplines.

85 For example, the CSDMS catalog supports FAIR compliance by integrating model registration with Zenodo, assigning a DOI
for citation and requiring the selection of an explicit license. Similarly, the USGS Model Catalog strengthens FAIR compliance
by providing a structured framework for model documentation, archiving, and discovery. Each model entry is linked to a
persistent identifier, ensuring that the associated software and metadata remain findable and citable over time. By enforcing
consistent standards for versioning, provenance, and access, the catalogs support transparent reuse. These approaches not only
safeguard the longevity of scientific software but also enhance the traceability and credibility of modeling studies within the
90 broader research community.

3.3 Continuing to make models more FAIR

Both USGS and CSDMS have developed and implemented certain principles on their path to FAIR models, whereas other aspects could receive more attention to promote growth. Below we outline prospective efforts for each of the FAIR principles based on our current perspective.

95 The *Findability* of the models in the catalogs can be enhanced through richer metadata descriptions, as noted above. In the current CSDMS framework, however, this would necessitate additional time and effort from those submitting model information. For the USGS Model Catalog, user engagement to develop add-on metadata modules is an area that can increase richness of metadata. With the integration of generative artificial intelligence, many metadata fields could be automatically populated, requiring model developers only to review and refine the autogenerated content as necessary.

00 In regard to FAIR4RS *Accessibility* guiding principles, both initiatives have implemented these as outlined in Barker et al. (2022) and there are no immediate next steps.

The *Interoperability* of models in the CSDMS catalog could be further enhanced through stronger enforcement and communication of value statements of their capability to couple with other models or datasets. Achieving this would require model developers to adopt the BMI protocols and apply Standard Names for all input and output variables. It is important to
05 note however, that implementing these best practices must be carried out by the model developers themselves, which entails additional time and effort from their perspective. From the research infrastructure perspective, the action would be to make it easier to adopt the protocols and names. For the USGS Model Catalog, the interoperability principle for qualified references to other objects could be met to a fuller extent by implementing currently experimental recurring feeds of new content. These feeds from USGS publications, data releases, software releases, and web page content systems would provide a stream of new
10 objects for more efficient growth of the catalog.

Next steps for *Reusability* could include providing mechanisms to make it easier for model developers to meet the principles for documenting provenance and using executable code notebooks. Additionally, CSDMS could be more strict about the use of version control systems. CSDMS might explore the use of containerization technologies such as Apptainer and Docker,

which encapsulate code, dependencies, and runtime environments into portable, versioned units (Vanegas Ferro et al., 2022).
15 This approach supports reproducibility and interoperability across platforms while enhancing the findability, accessibility, and
reusability of numerical models through easier distribution and reuse. In parallel, the USGS Model Catalog next steps could
include working with narrower modeling communities to develop modular add-ons to the metadata fields to help accommodate
specific needs of specific disciplines, a publicly accessible glossary to help with comprehension between different disciplinary
scientists, and promotion of increased use of the existing "software: runtime_image" field to point to runtime images such as
20 container images.

Making it easier to evaluate software and identify gaps could enhance the implementation of the FAIR principles. One approach
may be to quantitatively assess the degree of FAIR compliance in community-developed models. Bogan et al. (2025) have
developed a Python-based framework that facilitates the automated evaluation of FAIR principles, systematically identifying
gaps in metadata representation. Applying such quantitative assessments into individual models can provide a twofold benefit:
25 (1) potential users gain a clear, immediate understanding of the extent to which each FAIR principle is addressed, and (2)
model developers are more likely to improve the FAIR compliance of their models if they can easily see which principles have
not yet been achieved but are easy to meet. This FAIR assessment framework could be integrated into the CSDMS and USGS
Model Catalogs to enhance transparency and promote best practices in software development.

4 Summary

30 This paper aims to advance understanding of how FAIR4RS principles can be effectively adopted within research software
communities of diverse disciplines, while offering guidance for their implementation in other modeling initiatives. The
adoption of these principles promotes sustained interdisciplinary dialogue and thereby enhances the cumulative advancement
of scientific understanding. Although numerous approaches exist for discovering related and relevant model information on
the web, the two discussed catalogs - both curated and community-supported - substantially strengthen each of the FAIRness
35 principles of model development and dissemination (Findability, Accessibility, Interoperability, and Reusability). This
improvement stems from the ability to tailor the interfaces to the scope and norms of the user community, thereby
accommodating domain-specific knowledge and reducing common sources of confusion.

In this paper, we describe the trajectory toward enhanced FAIRness of the Community Surface Dynamics Modeling System
(CSDMS) and the U.S. Geological Survey (USGS) model collections following the guiding principles outlined by Barker et
40 al. (2022). The *findability* of models in both the CSDMS and USGS Model Catalogs relies on the implementation of rich,
structured metadata aligned with community standards, and the use of web technologies like Schema.org to enhance human
and machine discoverability. Despite challenges with component-level identifiers and versioning, the integration of APIs and
structured metadata formats has substantially improved the accessibility and discoverability of numerical models, supporting
broader adoption and effective reuse in scientific research. Both catalogs ensure source code *accessibility* through HTTPS
45 retrieval of models and persistent, machine-readable metadata. We achieve this by using standardized protocols, persistent

identifiers, and separate, structured metadata files that remain accessible. Regarding *interoperability*, CSDMS enhances this through technologies such as the Basic Model Interface, Standard Names, and the Python Modeling Toolkit, whereas the USGS Model Catalog emphasizes linking models to other digital objects and highlighting BMI-enabled models, thus fostering an ecosystem where models and data can interact efficiently. *Reusability* of models is increased by the CSDMS and the USGS Model Catalog by promoting the use of version control, open-source licensing, and interactive code notebooks.

The efforts by CSDMS and the USGS to enhance the FAIRness of model catalogs encountered numerous challenges, including maintaining current and comprehensive software infrastructure, ensuring consistent and community-driven content curation, and managing the diversity of model types and disciplinary-specific jargon. At the same time, both initiatives emphasize the importance of fostering community engagement, selecting open-source tools to support sustainability and adoption, and striking a balance between the need for detailed documentation and the practical constraints on developers' time and resources. Lessons learned from these experiences underscore that adaptable, user-centered solutions that integrate technical, disciplinary, and community perspectives, can support the development of sustainable, interoperable, and effective model catalogs.

Code and Data availability

No code is necessary to produce the discussion and results presented in this paper. However, the U.S. Geological Survey (USGS) model schema and metadata files describing USGS models are available Creative Commons CC0 1.0 Universal (CC0) at: <https://doi.org/10.5281/zenodo.17408255>.

Author contributions

All authors contributed to discussions and interpretation of the presented work. AK and LH jointly drafted the paper. With input from their respective communities, BS and LH developed and implemented the U.S. Geological Survey Model Catalog, while AK did so for the CSDMS Model Repository. Both efforts are the outcome of a community-wide supported effort, with contributions from numerous community members. All authors contributed to editing the manuscript.

Competing interests

The contact author has declared that none of the authors has any competing interests.

Disclaimer

Publisher's note: Copernicus Publications remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Acknowledgements

75 The Community Surface Dynamics Modeling System (CSDMS) is supported by the US National Science Foundation (NSF), award number 1831623, and additional sources of support include OpenEarthscapes (NSF Award Number 2104102). The authors thank Alicia Rhoades and Tamar Norkin for the thoughtful comments and suggestions, which helped improve the manuscript. We also gratefully acknowledge the contributions of numerous CSDMS members and U.S. Geological Survey employees, whose sharing of codes and other efforts have created a vibrant community of practice among modelers. Any use of trade, firm, or product names is for descriptive purposes only and does not imply endorsement by the U.S. Government.

Financial support

80 The presented work has been supported by the National Science Foundation (grant nos. 1831623 and 2104102).

References

- Ashton, A. D., Hutton, E. W. H., Kettner, A. J., Xing, F., Kallumadikal, J., Nienhuis, J., and Giosan, L.: Progress in coupling models of coastline and fluvial dynamics, *Computers & Geosciences*, 53, 21–29, <https://doi.org/10.1016/j.cageo.2012.04.004>, 2013.
- 85 Bakker, M., Post, V., Langevin, C. D., Hughes, J. D., White, J. S., Starn, J., and Fienen, M. N.: FloPy: Python package for creating, running, and post-processing MODFLOW-based models, , <https://doi.org/10.5066/F7BK19FH>, 2018.
- Bacchelli, A. and Bird, C.: Expectations, outcomes, and challenges of modern code review, in: 2013 35th International Conference on Software Engineering (ICSE), 2013 35th International Conference on Software Engineering (ICSE), 712–721, <https://doi.org/10.1109/ICSE.2013.6606617>, 2013.
- 90 Barker, M., Chue Hong, N. P., Katz, D. S., Lamprecht, A.-L., Martinez-Ortiz, C., Psomopoulos, F., Harrow, J., Castro, L. J., Gruenpeter, M., Martinez, P. A., and Honeyman, T.: Introducing the FAIR Principles for research software, *Sci Data*, 9, 622, <https://doi.org/10.1038/s41597-022-01710-x>, 2022.
- Barnes, N.: Publish your computer code: it is good enough, *Nature*, 467, 753–753, <https://doi.org/10.1038/467753a>, 2010.
- 95 Barnhart, K. R., Hutton, E. W. H., Tucker, G. E., Gasparini, N. M., Istanbuluoglu, E., Hobley, D. E. J., Lyons, N. J., Mouchene, M., Nudurupati, S. S., Adams, J. M., and Bandaragoda, C.: Short communication: Landlab v2.0: a software package for Earth surface dynamics, *Earth Surf. Dynam.*, 8, 379–397, <https://doi.org/10.5194/esurf-8-379-2020>, 2020.
- Barton, C. M., Lee, A., Janssen, M. A., Van Der Leeuw, S., Tucker, G. E., Porter, C., Greenberg, J., Swantek, L., Frank, K., Chen, M., and Jagers, H. R. A.: How to make models more useful, *Proc. Natl. Acad. Sci. U.S.A.*, 119, e2202112119, <https://doi.org/10.1073/pnas.2202112119>, 2022.
- 00

- Belete, G. F., Voinov, A., and Laniak, G. F.: An overview of the model integration process: From pre-integration assessment to testing, *Environmental Modelling & Software*, 87, 49–63, <https://doi.org/10.1016/j.envsoft.2016.10.013>, 2017.
- Blischak, J. D., Davenport, E. R., and Wilson, G.: A Quick Introduction to Version Control with Git and GitHub, *PLoS Comput Biol*, 12, e1004668, <https://doi.org/10.1371/journal.pcbi.1004668>, 2016.
- 05 Bogan, A., Garousi-Nejad, I., and Castronova, A.: Advancing FAIR Research Software Practices in Hydrology Domain through Metadata Standardization and Evaluation, 2025.
- Bonaccorsi, A. and Rossi, C.: Why Open Source software can succeed, *Research Policy*, 32, 1243–1258, [https://doi.org/10.1016/S0048-7333\(03\)00051-9](https://doi.org/10.1016/S0048-7333(03)00051-9), 2003.
- Campforts, B., Shobe, C. M., Steer, P., Vanmaercke, M., Lague, D., and Braun, J.: HyLands 1.0: a hybrid landscape evolution model to simulate the impact of landslides and landslide-derived sediment on landscape evolution, *Geosci. Model Dev.*, 10 13, 3863–3886, <https://doi.org/10.5194/gmd-13-3863-2020>, 2020.
- Carballo-Garcia, A. and Boté-Vericad, J.-J.: Fair Data: History and Present Context, *CEJER*, 4, 45–53, <https://doi.org/10.37441/cej/2022/4/2/11379>, 2022.
- Chue Hong, N. P., Katz, D. S., Barker, M., Lamprecht, A.-L., Martinez, C., Psomopoulos, F. E., Harrow, J., Castro, L. J., 15 Gruenpeter, M., Martinez, P. A., and Honeyman, T.: FAIR Principles for Research Software (FAIR4RS Principles), <https://doi.org/10.15497/RDA00068>, 2021.
- Collins, N., Theurich, G., DeLuca, C., Suarez, M., Trayanov, A., Balaji, V., Li, P., Yang, W., Hill, C., and Da Silva, A.: Design and Implementation of Components in the Earth System Modeling Framework, *The International Journal of High Performance Computing Applications*, 19, 341–350, <https://doi.org/10.1177/1094342005056120>, 2005.
- 20 Colvin, S., Victorien, Runkle, S., Montague, D., Garcia Badaracco, A., Hewitt, D., Ramezani, H., Jolibois, E., Trylesinski, M., Dorsey, T., Matveenko, S., pyup. io bot, Ramirez, S., Boykov, A., Hall, A., Aono, K., Grishko, N., messense, Karabas, Y., Misra, S., Samigullin, V., Tahiri, Y., kc0506, Alae, A., Neevos, Brown II, S., and Arhancet, J.: pydantic/pydantic: v1.10.26 2025-12-18, , <https://doi.org/10.5281/ZENODO.8180180>, 2025.
- Crowston, K., Wei, K., Howison, J., and Wiggins, A.: Free/Libre open-source software development: What we know and what 25 we do not know, *ACM Comput. Surv.*, 44, 7:1-7:35, <https://doi.org/10.1145/2089125.2089127>, 2008.
- Diehl, P., Sonesson, C., Kurchin, R. C., Mounce, R., and Katz, D. S.: The Journal of Open Source Software (JOSS): Bringing Open-Source Software Practices to the Scholarly Publishing Community for Authors, Reviewers, Editors, and Publishers, *Journal of Librarianship and Scholarly Communication*, 12, <https://doi.org/10.31274/jlsc.18285>, 2025.
- Escamilla, E., Klein, M., Cooper, T., Rampin, V., Weigle, M. C., and Nelson, M. L.: The Rise of GitHub in Scholarly 30 Publications, in: *Linking Theory and Practice of Digital Libraries*, vol. 13541, edited by: Silvello, G., Corcho, O., Manghi, P., Di Nunzio, G. M., Golub, K., Ferro, N., and Poggi, A., Springer International Publishing, Cham, 187–200, https://doi.org/10.1007/978-3-031-16802-4_15, 2022.
- Fricke, S.: Semantic Scholar, *jmla*, 106, <https://doi.org/10.5195/jmla.2018.280>, 2018.

- 35 Gan, T., Tucker, G. E., Hutton, E. W. H., Piper, M. D., Overeem, I., Kettner, A. J., Campforts, B., Moriarty, J. M., Undzis, B.,
Pierce, E., and McCready, L.: CSDMS Data Components: data–model integration tools for Earth surface processes
modeling, *Geosci. Model Dev.*, 17, 2165–2185, <https://doi.org/10.5194/gmd-17-2165-2024>, 2024.
- Garijo, D., Osorio, M., Khider, D., Ratnakar, V., and Gil, Y.: OKG-Soft: An Open Knowledge Graph with Machine Readable
Scientific Software Metadata, in: 2019 15th International Conference on eScience (eScience), 2019 15th International
Conference on eScience (eScience), San Diego, CA, USA, 349–358, <https://doi.org/10.1109/eScience.2019.00046>, 2019.
- 40 Garijo, D., Ménager, H., Hwang, L., Trisovic, A., Hucka, M., Morrell, T., Allen, A., Task Force on Best Practices for Software
Registries, and SciCodes Consortium: Nine best practices for research software registries and repositories, *PeerJ Computer
Science*, 8, e1023, <https://doi.org/10.7717/peerj-cs.1023>, 2022.
- Gomes, D. G. E., Pottier, P., Crystal-Ornelas, R., Hudgins, E. J., Foroughirad, V., Sánchez-Reyes, L. L., Turba, R., Martinez,
P. A., Moreau, D., Bertram, M. G., Smout, C. A., and Gaynor, K. M.: Why don't we share data and code? Perceived
45 barriers and benefits to public archiving practices, *Proc. R. Soc. B.*, 289, 20221113,
<https://doi.org/10.1098/rspb.2022.1113>, 2022.
- Gregersen, J. B., Gijsbers, P. J. A., and Westen, S. J. P.: OpenMI: Open modelling interface, *Journal of Hydroinformatics*, 9,
175–191, <https://doi.org/10.2166/hydro.2007.023>, 2007.
- Guizzardi, G.: Ontology, Ontologies and the “I” of FAIR, *Data Intellegence*, 2, 181–191,
50 https://doi.org/10.1162/dint_a_00040, 2020.
- Hahnel, M. and Valen, D.: How to (Easily) Extend the FAIRness of Existing Repositories, *Data Intellegence*, 2, 192–198,
https://doi.org/10.1162/dint_a_00041, 2020.
- Harpham, Q. K., Hughes, A., and Moore, R. V.: Introductory overview: The OpenMI 2.0 standard for integrating numerical
models, *Environmental Modelling & Software*, 122, 104549, <https://doi.org/10.1016/j.envsoft.2019.104549>, 2019.
- 55 Hirsch, J. E.: An index to quantify an individual's scientific research output, *Proc. Natl. Acad. Sci. U.S.A.*, 102, 16569–16572,
<https://doi.org/10.1073/pnas.0507655102>, 2005.
- Hsu, L., Liford, A. N., and Donovan, G. C.: Community for Data Integration 2020 annual report, U.S. Geological Survey,
<https://doi.org/10.3133/ofr20221034>, 2022.
- Huang, Y.-N., Munteanu, V., Love, M. I., Ronkowski, C. F., Deshpande, D., Wong-Beringer, A., Corbett-Detig, R., Dimian,
60 M., Moore, J. H., Garmire, L. X., Reddy, T. B. K., Butte, A. J., Robinson, M. D., Eskin, E., Abedalthagafi, M. S., and
Mangul, S.: Perceptual and technical barriers in sharing and formatting metadata accompanying omics studies, *Cell
Genomics*, 5, 100845, <https://doi.org/10.1016/j.xgen.2025.100845>, 2025.
- Hunt, J.: Python Classes, in: *A Beginners Guide to Python 3 Programming*, Springer International Publishing, Cham, 209–
225, https://doi.org/10.1007/978-3-031-35122-8_19, 2023.
- 65 Hutton, E., Barnhart, K., Hobley, D., Tucker, G., Nudurupati, S. S., Adams, J., Gasparini, N. M., Shobe, C., Strauch, R., Knuth,
J., Margauxmouchene, Lyons, N., DavidLitwin, Glade, R., Giuseppicipolla95, Manaster, A., Alangston, Thyng, K., and
Rengers, F.: landlab/landlab, , <https://doi.org/10.5281/ZENODO.595872>, 2020a.

- Hutton, E. W., Piper, M. D., and Tucker, G. E.: The Basic Model Interface 2.0: A standard interface for coupling numerical models in the geosciences, *J. Open Source Softw.*, 5, 2317, <https://doi.org/10.21105/joss.02317>, 2020b.
- 70 Hutton, E., Piper, M., Gan, T., Kettner, A. J., and Drost, N.: *csdms/pymt: (v1.3.1)*, Zenodo, <https://doi.org/10.5281/zenodo.4985222>, 2021.
- Jacobsen, A., De Miranda Azevedo, R., Juty, N., Batista, D., Coles, S., Cornet, R., Courtot, M., Crosas, M., Dumontier, M., Evelo, C. T., Goble, C., Guizzardi, G., Hansen, K. K., Hasnain, A., Hettne, K., Heringa, J., Hooft, R. W. W., Imming, M., Jeffery, K. G., Kaliyaperumal, R., Kersloot, M. G., Kirkpatrick, C. R., Kuhn, T., Labastida, I., Magagna, B., McQuilton, P., Meyers, N., Montesanti, A., Van Reisen, M., Rocca-Serra, P., Pergl, R., Sansone, S.-A., Da Silva Santos, L. O. B., Schneider, J., Strawn, G., Thompson, M., Waagmeester, A., Weigel, T., Wilkinson, M. D., Willighagen, E. L., Wittenburg, P., Roos, M., Mons, B., and Schultes, E.: FAIR Principles: Interpretations and Implementation Considerations, *Data Intelligence*, 2, 10–29, https://doi.org/10.1162/dint_r_00024, 2020.
- 75 Jenni, K. E., Goldhaber, M. B., Betancourt, J. L., Baron, J. S., Bristol, S., Cantrill, M., Exter, P. E., Focazio, M. J., Haines, J. W., Hay, L. E., Hsu, L., Labson, V. F., Lafferty, K. D., Ludwig, K. A., Milly, P. C. D., Morelli, T. L., Morman, S. A., Nassar, N. T., Newman, T. R., Ostroff, A. C., Read, J. S., Reed, S. C., Shapiro, C. D., Smith, R. A., Sanford, W. E., Sohl, T. L., Stets, E. G., Terando, A. J., Tillitt, D. E., Tischler, M. A., Toccalino, P. L., Wald, D. J., Waldrop, M. P., Wein, A., Weltzin, J. F., and Zimmerman, C. E.: Grand challenges for integrated USGS science — A workshop report, U.S. Geological Survey, <https://doi.org/10.3133/ofr20171076>, 2017.
- 80 Jones, M. B., Boettiger, C., Mayes, A. C., Arfon Smith, Slaughter, P., Niemeyer, K., Gil, Y., Fenner, M., Nowak, K., Hahnel, M., Coy, L., Allen, A., Crosas, M., Sands, A., Hong, N. C., Cruse, P., Katz, D., and Goble, C.: CodeMeta: an exchange schema for software metadata, *KNB Data Repository*, <https://doi.org/10.5063/SCHEMA/CODEMETA-2.0>, 2017.
- Katz, D. S., Bouquin, D., Hong, N. P. C., Hausman, J., Jones, C., Chivvis, D., Clark, T., Crosas, M., Druskat, S., Fenner, M., Gillespie, T., Gonzalez-Beltran, A., Gruenpeter, M., Habermann, T., Haines, R., Harrison, M., Henneken, E., Hwang, L., Jones, M. B., Kelly, A. A., Kennedy, D. N., Leinweber, K., Rios, F., Robinson, C. B., Todorov, I., Wu, M., and Zhang, Q.: Software Citation Implementation Challenges, <https://doi.org/10.48550/ARXIV.1905.08674>, 2019.
- 90 Keisman, J. L., Bristol, S., Brown, D. S., Flickinger, A. K., Gunther, G. L., Murdoch, P. S., Musgrove, M., Nelson, J. C., Steyer, G. D., Thomas, K. A., and Waite, I. R.: Capacity assessment for Earth Monitoring, Analysis, and Prediction (EarthMAP) and future integrated monitoring and predictive science at the U.S. Geological Survey, U.S. Geological Survey, <https://doi.org/10.3133/ofr20211102>, 2021.
- 95 Kluyver, T., Ragan-Kelley, B., Pérez, F., Granger, B., Bussonnier, M., Frederic, J., Kelley, K., Hamrick, J., Grout, J., Corlay, S., Ivanov, P., Avila, D., Abdalla, S., Willing, C., and Jupyter Development Team: Jupyter Notebooks - a publishing format for reproducible computational workflows, in: *Positioning and Power in Academic Publishing: Players, Agents and Agendas*, IOS Press, <https://doi.org/10.3233/978-1-61499-649-1-87>, 2016.

- Kröttsch, M., Vrandečić, D., and Völkel, M.: Semantic MediaWiki, in: The Semantic Web - ISWC 2006, vol. 4273, edited by: Cruz, I., Decker, S., Allemang, D., Preist, C., Schwabe, D., Mika, P., Uschold, M., and Aroyo, L. M., Springer Berlin Heidelberg, Berlin, Heidelberg, 935–942, https://doi.org/10.1007/11926078_68, 2006.
- Langevin, C. D., Hughes, J. D., Banta, E. R., Provost, A., Niswonger, R., and Panday, S.: MODFLOW 6, the U.S. Geological Survey Modular Hydrologic Model, , <https://doi.org/10.5066/F76Q1VQV>, 2017.
- Leaf, A. T. and Fienen, M. N.: Pleasant Lake worked Flopy example, U.S. Geological Survey, <https://doi.org/10.5066/P9EFHF9H>, 2022.
- Mabile, L., Shmagun, H., Erdmann, C., Cambon-Thomsen, A., Thomsen, M., and Grattarola, F.: Recommendations on Open Science Rewards and Incentives, Data Science Journal, 24, 15, <https://doi.org/10.5334/dsj-2025-015>, 2025.
- McKiernan, E. C., Bourne, P. E., Brown, C. T., Buck, S., Kenall, A., Lin, J., McDougall, D., Nosek, B. A., Ram, K., Soderberg, C. K., Spies, J. R., Thaney, K., Updegrove, A., Woo, K. H., and Yarkoni, T.: How open science helps researchers succeed, eLife, 5, e16800, <https://doi.org/10.7554/eLife.16800>, 2016.
- National Research Council and Division on Engineering and Physical Sciences and Space Studies Board and Committee on Earth Science and Applications from Space and A Community Assessment and Strategy for the Future: Earth science and applications from space: national imperatives for the next decade and beyond, National Academies Press, 2007.
- Office of Management and Budget (OMB): Achieving Efficiency, Transparency, and Innovation through Reusable and Open Source Software (OMB Memorandum M-16-21), https://obamawhitehouse.archives.gov/sites/default/files/omb/memoranda/2016/m_16_21.pdf, 2016.
- Overeem, I., Berlin, M. M., and Syvitski, J. P. M.: Strategies for integrated modeling: The community surface dynamics modeling system example, Environmental Modelling & Software, 39, 314–321, <https://doi.org/10.1016/j.envsoft.2012.01.012>, 2013.
- Payne, K. and Verhey, C.: Schema.org for research data managers: a primer, IJBDM, 2, 95, <https://doi.org/10.1504/IJBDM.2022.128449>, 2022.
- Peckham, S.: The CSDMS Standard Names: Cross-Domain Naming Conventions for Describing Process Models, Data Sets and Their Associated Variables, in: International Congress on Environmental Modelling and Software, 2014.
- Peckham, S. D., Hutton, E. W. H., and Norris, B.: A component-based approach to integrated modeling in the geosciences: The design of CSDMS, Computers & Geosciences, 53, 3–12, <https://doi.org/10.1016/j.cageo.2012.04.002>, 2013.
- Puebla, I., Ascoli, G. A., Blume, J., Chodacki, J., Finnell, J., Kennedy, D. N., Mair, B., Martone, M. E., Wittenberg, J., and Poline, J.-B.: Ten simple rules for recognizing data and software contributions in hiring, promotion, and tenure, PLoS Comput Biol, 20, e1012296, <https://doi.org/10.1371/journal.pcbi.1012296>, 2024.
- Ronallo, J.: HTML5 Microdata and Schema.org, The Code4Lib Journal, 2012.
- Schuwirth, N., Borgwardt, F., Domisch, S., Friedrichs, M., Kattwinkel, M., Kneis, D., Kuemmerlen, M., Langhans, S. D., Martínez-López, J., and Vermeiren, P.: How to make ecological models useful for environmental management, Ecological Modelling, 411, 108784, <https://doi.org/10.1016/j.ecolmodel.2019.108784>, 2019.

- 35 Serna, B. S. and Hsu, L.: USGS Model Catalog Tools and Metadata, U.S. Geological Survey, <https://doi.org/10.5066/P9WU0F71>, 2022.
- Serna, B. S., Hsu, L., Donovan, G. C., and Liford, A. N.: USGS Model Catalog intake repository, U.S. Geological Survey, <https://doi.org/10.5066/P9IVG9VZ>, 2022.
- Serna, B., Hsu, L., Donovan, G., and Liford, A.: U.S. Geological Survey Model Catalog intake and tam repositories, ,
40 <https://doi.org/10.5281/ZENODO.17408254>, 2025.
- Singh, M., Kumar, A., Donaparthi, S., and Karambelkar, G.: Leveraging Retrieval Augmented Generative LLMs For Automated Metadata Description Generation to Enhance Data Catalogs, <https://doi.org/10.48550/ARXIV.2503.09003>, 2025.
- Specht, A., Stall, S., Machicao, J., Catry, T., Chaumont, M., David, R., Devillers, R., Edmunds, R., Jarry, R., Mabile, L.,
45 Miyairi, N., O'Brien, M., Correa, P. P., Santos, S., Subsol, G., and Wyborn, L.: Managing linguistic obstacles in multidisciplinary, multinational, and multilingual research projects, PLoS ONE, 19, e0311967, <https://doi.org/10.1371/journal.pone.0311967>, 2024.
- Taylor, P.: Web APIs for environmental data - state of the art investigation, <https://doi.org/10.13140/RG.2.2.17817.83047>, 2016.
- 50 Treloar, A.: The Research Data Alliance: globally co-ordinated action against barriers to data publishing and sharing, Learned Publishing, 27, <https://doi.org/10.1087/20140503>, 2014.
- Tucker, G. E., Hutton, E. W. H., Piper, M. D., Campforts, B., Gan, T., Barnhart, K. R., Kettner, A. J., Overeem, I., Peckham, S. D., McCreedy, L., and Syvitski, J.: CSDMS: a community platform for numerical modeling of Earth surface processes, Geosci. Model Dev., 15, 1413–1439, <https://doi.org/10.5194/gmd-15-1413-2022>, 2022.
- 55 USGS, IM OSQI 2019-01, Review and Approval of Scientific Software for Release: <https://www.usgs.gov/survey-manual/im-osqi-2019-01-review-and-approval-scientific-software-release>, 2019, last access: 19 November 2025.
- USGS, FSP FAQs, <https://www.usgs.gov/office-of-science-quality-and-integrity/fundamental-science-practices/faq/software>, 2025, last access: December 26, 2025.
- USGS, MODFLOW 6, the U.S. Geological Survey Modular Hydrologic Model, <https://doi.org/10.5066/F76Q1VQV>, 2026.
- 60 Vanegas Ferro, M., Lee, A., Pritchard, C., Barton, C. M., and Janssen, M. A.: Containerization for creating reusable model code, SESMO, 3, 18074, <https://doi.org/10.18174/sesmo.18074>, 2022.
- Wilkinson, M. D., Dumontier, M., Aalbersberg, Ij. J., Appleton, G., Axton, M., Baak, A., Blomberg, N., Boiten, J.-W., Da Silva Santos, L. B., Bourne, P. E., Bouwman, J., Brookes, A. J., Clark, T., Crosas, M., Dillo, I., Dumon, O., Edmunds, S., Evelo, C. T., Finkers, R., Gonzalez-Beltran, A., Gray, A. J. G., Groth, P., Goble, C., Grethe, J. S., Heringa, J., 'T Hoen,
65 P. A. C., Hooft, R., Kuhn, T., Kok, R., Kok, J., Lusher, S. J., Martone, M. E., Mons, A., Packer, A. L., Persson, B., Rocca-Serra, P., Roos, M., Van Schaik, R., Sansone, S.-A., Schultes, E., Sengstag, T., Slater, T., Strawn, G., Swertz, M. A., Thompson, M., Van Der Lei, J., Van Mulligen, E., Velterop, J., Waagmeester, A., Wittenburg, P., Wolstencroft, K., Zhao,

- J., and Mons, B.: The FAIR Guiding Principles for scientific data management and stewardship, *Sci Data*, 3, 160018, <https://doi.org/10.1038/sdata.2016.18>, 2016.
- 70 Wilkinson, S. R., Aloqalaa, M., Belhajjame, K., Crusoe, M. R., De Paula Kinoshita, B., Gadelha, L., Garijo, D., Gustafsson, O. J. R., Juty, N., Kanwal, S., Khan, F. Z., Köster, J., Peters-von Gehlen, K., Pouchard, L., Rannow, R. K., Soiland-Reyes, S., Soranzo, N., Sufi, S., Sun, Z., Vilne, B., Wouters, M. A., Yuen, D., and Goble, C.: Applying the FAIR Principles to computational workflows, *Sci Data*, 12, 328, <https://doi.org/10.1038/s41597-025-04451-9>, 2025.
- 75 Wilson, T., Wiltermuth, M. T., Jenni, K. E., Horton, R., Hunt, R. J., Williams, D. M., Nolan, V. P., Aumen, N. G., Brown, D. S., Blasch, K. W., and Murdoch, P. S.: Use case development for earth monitoring, analysis, and prediction (EarthMAP)—A road map for future integrated predictive science at the U.S. Geological Survey, U.S. Geological Survey, <https://doi.org/10.3133/ofr20211108>, 2022.
- Xu, K., Chen, M., Yue, S., Zhang, F., Wang, J., Wen, Y., and Lü, G.: The portal of OpenGMS: Bridging the contributors and users of geographic simulation resources, *Environmental Modelling & Software*, 180, 106142, <https://doi.org/10.1016/j.envsoft.2024.106142>, 2024.
- 80 Zaslavsky, I., Whitenack, T., and Valentine, D.: Exploring Environmental Model Catalogs, *International Congress on Environmental Modelling and Software*, 2014.
- Zhang, F., Chen, M., Kettner, A. J., Ames, D. P., Harpham, Q., Yue, S., Wen, Y., and Lü, G.: Interoperability engine design for model sharing and reuse among OpenMI, BMI and OpenGMS-IS model standards, *Environmental Modelling & Software*, 144, 105164, <https://doi.org/10.1016/j.envsoft.2021.105164>, 2021.
- 85