



HESS Opinions: Participatory Digital eARth Twin Hydrology systems (DARTHs) for everyone – a blueprint for hydrologists

Riccardo Rigon¹, Giuseppe Formetta², Marialaura Bancheri³, Niccolò Tubini², Concetta D'Amato¹, Olaf David⁴, and Christian Massari⁵

¹Center Agriculture Food Environment (C3A), University of Trento, Trento, Italy

²Department of Civil, Environmental and Mechanical Engineering (DICAM), Trento, Italy

³Institute for Mediterranean Agricultural and Forestry Systems (ISAFOM), National Research Council (CNR), Portici, Italy

⁴Department of Civil and Environmental Engineering, Colorado State University, Fort Collins, CO, USA

⁵National Research Council, Research Institute for Geo-Hydrological Protection, Perugia, Italy

Correspondence: Riccardo Rigon (riccardo.rigon@unitn.it)

Received: 27 December 2021 – Discussion started: 17 January 2022

Revised: 4 July 2022 – Accepted: 6 September 2022 – Published: 29 September 2022

Abstract. The “Digital Earth” (DE) metaphor is very useful for both end users and hydrological modelers (i.e., the coders). In this opinion paper, we analyze different categories of models with the view of making them part of Digital eARth Twin Hydrology systems (DARTHs). We stress the idea that DARTHs are not models, rather they are an appropriate infrastructure that hosts (certain types of) models and provides some basic services for connecting to input data. We also argue that a modeling-by-component strategy is the right one for accomplishing the requirements of the DE. Five technological steps are envisioned to move from the current state of the art of modeling. In step 1, models are decomposed into interacting modules with, for instance, the agnostic parts dealing with inputs and outputs separated from the model-specific parts that contain the algorithms. In steps 2 to 4, the appropriate software layers are added to gain transparent model execution in the cloud, independently of the hardware and the operating system of computer, without human intervention. Finally, step 5 allows models to be selected as if they were interchangeable with others without giving deceptive answers. This step includes the use of hypothesis testing, the inclusion of error of estimates, the adoption of literate programming and guidelines to obtain informative clean code.

The urgency for DARTHs to be open source is supported here in light of the open-science movement and its ideas. Therefore, it is argued that DARTHs must promote a new

participatory way of performing hydrological science, in which researchers can contribute cooperatively to characterize and control model outcomes in various territories. Finally, three enabling technologies are also discussed in the context of DARTHs – Earth observations (EOs), high-performance computing (HPC) and machine learning (ML) – as well as how these technologies can be integrated in the overall system to both boost the research activity of scientists and generate knowledge.

1 Introduction

The “Digital Earth” (DE) concept was first developed by the US Vice President Al Gore in a speech for the opening of the California Science Center in 1998. In Al Gore’s vision, the DE is meant to be “a multi-resolution, three-dimensional representation of the planet into which we can embed vast quantities of geo-referenced data”. Although the technologies available in 1998 were not at all adequate to pursue this vision, Al Gore’s speech aroused great interest within the scientific community, and the first International Symposium on Digital Earth was held in Beijing, China, in 1999. The outcomes of this symposium were summarized in the 1999 “Beijing Declaration on Digital Earth”. In 2006, the International Society for Digital Earth (ISDE) was formally established, and it founded an international peer-reviewed aca-

demic journal in 2008: the *International Journal of Digital Earth* (IJDE). Since then, another milestone document, the 2019 “11th International Symposium on Digital Earth, Florence declaration”, has been approved and position papers have been published (Goodchild et al., 2012; Craglia et al., 2012; Guo et al., 2019).

Over the years the original vision has continually evolved, following the progress of technologies and of the global community of interest. Nowadays, the DE is considered a global strategic contributor to scientific and technological developments, and it could play a strategic and sustainable role in facing the issues characterizing the Anthropocene epoch. Thanks to the advancement of technologies, it is possible to talk about digital twin (DT) Earth models. These models aim to accurately reproduce the state of the evolution of a generic physical entity through high-fidelity computer models, with the goal of understanding and simulating systems’ behaviors and of evaluating them under changing boundary and initial conditions. As such, the DT dates back to NASA’s Apollo program (Semeraro et al., 2021), and, prior to this program, they were developed in sectors where the processes being modeled are well understood, such as industry and manufacturing (Graessler and Pöhler, 2017).

In modeling natural processes, DE models use the same concepts (e.g., physical principles and equations) and data as hydrology. It is worth noting that, as early as 1986, Eagleson (1986) recognized the need to develop hydrological models to produce hydrological prognoses at the global scale. These global-scale hydrological models (Beck et al., 2017; Emerton et al., 2016; Stacke and Hagemann, 2021; Döll et al., 2003) lack the “organizing vision” of DE, as they are not an integral part of other advanced technologies, such as Earth observation, geo-information system, virtual reality and sensor webs.

Currently, the convergent forces of the space agencies and the products of their satellites have given substance to the possibility of really getting hyper-resolution models of Earth (Wood et al., 2011). These efforts have also triggered the interest of the computer science community, with its high-performance, distributed computing infrastructures and all of the related technologies. However, these same efforts often bring to light an approach where research and its related data are created and shaped by big institutional players, even private entities, and not by the large majority of researchers. At the same time, small research groups and researchers simply apply the existing deployed tools, unless they are directly involved in the development. As such, this top-down approach could limit creativity and the possibility of a vast community to contribute to the advancement of science and innovation (Oleson et al., 2013; Best et al., 2011).

The EU’s recent Destination Earth (DestinE) campaign (e.g., <https://digital-strategy.ec.europa.eu/en/policies/destination-earth> last access: 23 September 2022) runs the same abovementioned risk; therefore, it is time to restore a bottom-up approach in which the creativity of individuals and small groups can be harmonized within the big view,

with the least number of institutionalized organizations possible.

This paper looks at this goal from the perspective of those who deploy “community” hydrological models, meaning models developed by a community of researchers that freely gathers and discusses ideas about hydrological and Earth system science; produces model parts; and commits them to common, decentralized repositories. Their contribution usually encompasses theoretical achievements, implementation design, science verification from data and deployment for applications, which are seen as the natural outcome and source of the most fundamental research of hydrologic processes.

Therefore, this opinion paper aims to do the following:

- provide an up-to-date analysis of the current research in designing and developing DE components, with a particular focus on hydrological modeling;
- identify possible directions and hints, based on more than a decade of experience in deploying hydrological models (Rigon et al., 2006; Endrizzi et al., 2014; Formetta et al., 2014; Tubini and Rigon, 2022) and contributing to geographic information system (GIS) developments (e.g., JGrass, uDig and gvSIG);
- debate and identify potential ways forward to answer the following questions such as “Is it possible to have more than one Digital eARTh Hydrological (DARTH) model while avoiding fragmentation of efforts?”, “How can data become available to allow the vision to be incrementally realized?”, “Should the models be designed differently?”, “Which informatics is suitable?”, “How should high-performance computing (HPC) be envisioned and developed?”, “What efforts should be made in modeling and data representation?”, “How is the scientific reliability assessed?”, “What role can Earth observations (EOs) play?” and “Is machine learning (ML) the solution?”.

The paper is organized into four sections, with many subsections, the Appendix, and the Supplement. Section 1.1 discusses the idea that a DARTH is not a single piece of software but an ecosystem of contributions. Section 1.2 briefly discusses the questions that relate to data availability and data flow, thereby representing an introduction to the Digital eARTh Hydrological systems (DARTHS) topic. Section 2 deals with design and implementation requirements; its subsections first introduce and then explore the suitable modeling architecture for the purpose of buildings DARTH components. Reusability and the possibility of changes to scientific paradigms, with as few restrictions as possible, are presented as key DARTH concepts. As good programming practices are of fundamental importance for open science, this section also contains a discussion on clean coding, literate programming and, as a consequence, literate computing. Subsection 2.7 raises the topic of the reliability of DARTHS.

DARTHs are themselves among the “enabling technologies”, i.e., innovations that can drive radical change in the capabilities of scientists to perform new hydrology. Section 3.1 to 3.3 deal with three other enabling technologies that are pervasive in contemporary sciences and can be seen as protagonists in near-future developments – HPC, Earth observations (EOs) and the use of ML techniques; this section also provides a short review of applications, issues and perspectives for each. Section 4 discusses what could be the governance and organization of communities of DARTH developers and provides conclusions. Finally, the Supplement contains a glossary of terms (that defines and explain the acronyms and jargons used in the paper), with a specialized glossary regarding a classification of models from the point of view of DARTHs; a DARTH cheat sheet that summarizes the content of this paper; and the results of a survey done among model and model infrastructures developers that tries to capture the present state of the art. As the paper makes use of many acronyms, they are also reported in Table A1.

1.1 From models to DARTHs

As there are no real DEs in hydrology to date, our analysis refers to the most used hydrological models as a starting point. Their history (Beven, 2012) shows that there is a fragmentation of models and that “legacy more than adequacy” is the rule for researchers when choosing an application (Ador and Melsen, 2019). Models like SWAT (Arnold et al., 2012; Neitsch et al., 2011), HEC-HMS (Chu Xuefeng and Steinman Alan, 2009), SWMM (Gironás et al., 2010), and the TOPMODEL family (Peters et al., 2003) or the good old reservoir models (Knoben et al., 2019) have the lion’s share among users. These models usually provide a good balance between usability and reliability (in the sense that they produce plausible patterns and have no or few running issues). The current state of hydrological science allows us to say that all of these models work, at least for the purposes that they were requested, but they are normally closed to easy modification and tend to lag behind the state of the art of hydrological studies. The state of the art, on the other hand, often makes use of artisanal products, restricted by badly engineered codes, in the hands of a few researchers, and it is not designed for the general reuse that the DE paradigm would require. Many modular frameworks have recently been built with the aim of filling this gap, like SUMMA (Clark et al., 2011a), SuperflexPy (Dal Molin et al., 2021), GEOframe (Formetta et al., 2014) and Raven (Craig et al., 2020). However, from the points of view of both researchers and users, a further quantum leap must be made in model infrastructure design and model implementations to cope with the DARTH requirements that are addressed in the paper.

Broadly speaking, four mathematical tools dominate in hydrological modeling research (Kampf and Burges, 2007): process-based (PB) models (Paniconi and Putti, 2015; Fatichi et al., 2016a), reservoir-type models (HDSs, as hydrological

dynamical systems) (Todini, 2007; Bancheri et al., 2019b), classical statistical models (SMs) (McCuen, 2016) and current algorithmic–statistical models that make use of one form of machine learning (Shen et al., 2018; Levia et al., 2020). To these we could add a further type of model: a tight black box between Earth observations and lumped models and/or machine learning models, usually referred as EO products (Martens et al., 2017). Many references already discuss the taxonomy of models (e.g., Kampf and Burges, 2007), the strengths and weaknesses of each of these approaches (Hrachowitz and Clark, 2017), and their application at various scales of application, and we do not want to add further to those analyses. We just adopt the pragmatic view that they exist; that, notwithstanding their uncertain informatics quality (except maybe for the ML tools that are based on the use of large frameworks), they are used creatively; and that they still continue to produce insights that solve hydrological issues.

All of these models rely on parameter calibration (Duan et al., 2003) or on some type of “learning” to make their predictions realistic (Tsai et al., 2021). At present, it is impossible to disentangle the complexity of the model variety and make it simpler. The matter of hydrology and Earth system sciences is complex (in the sense of the complexity sciences) and complicated (affected by huge variability and heterogeneity); thus, whatever it takes to get a clue should be welcomed. This statement, however, implies that we cannot have a DARTH in which just one “solution” is adopted, rather we need a DARTH where many, even competing, paradigms (like PB modeling, ML tools and traditional lumped models) can be tested, compared and eventually put to work together to assess the uncertainty in forecasting. This, in turn, has consequences with respect to the code architectures, infrastructures and the informatics that have to be deployed. Some naive ideas that ML plus EOs, for instance, could do it all, like some Google Earth-like applications seem to envision, is just wishful thinking that clashes with the current view that we have of the discipline. Instead, the idea that different models (or model structures) have to be used and subjected to hypothesis testing procedures has gained ground (Clark et al., 2011b; Blöschl, 2017; Beven, 2019), even in ML (Shen et al., 2018).

Recently, in fact, it has become clear that ML and deep learning (DL) techniques can be interpreted and explained (Gharari et al., 2021); thus, they can be used as a tool for understanding (the process) (Arrieta et al., 2020) or model parameter learning (Tsai et al., 2021), instead of primarily for predictive purposes. In any case, ML growth has been mostly driven by a large variety of problems, for instance, computer vision applications, and speech and natural language processing in a way that has to be harmonized with the practices of more traditional ways of conceiving hydrological models.

If the model paradigms cannot be compressed in favor of one choice, a suitable DARTH engine should allow for the

implementation of various paradigms, and more than one competing DARTH engine should be made available in the community because the implemented technologies also come with their own legacies.

Just as life on Earth is built upon the four nucleotide bricks and presents an immense variety, it is clearly possible that DARTHS can share common protocols, standards and features and that they can evolve, picking up the best without a continuous recreation of the whole infrastructure from scratch. That is to say, promoting diversity should be accompanied by the sharing of standards for the parts. The way to do it has been traced, for instance, in Knoben et al. (2021), who argued that the whole models informatics can be separated into model-agnostic parts, which potentially can be shared, and model-specific parts, which could be differentiated among the various developers or research groups.

It should be stressed, however, that according to their own definition, DARTHS are not simply hydrological models. Models made by scientists are usually developed to test their theoretical works or to provide evidence for research publication; as such, they do not have software quality as their primary goal. DARTHS have models at their scientific core but, according to the vision we promote, they need to provide other services too, including

- being available on demand, working seamlessly on the cloud as web services;
- having automatic ways of retrieving and providing data;
- and being interoperable with other models.

In order to achieve these goals, the modeling needs to be supported by appropriate layers of software that orchestrate the whole functioning. Moreover,

- they need to be implemented robustly (see the glossary) and properly designed to be fault-tolerant.

In this paper we also support the idea that DARTHS have to serve science in its doing and evolving, harmonizing the work of the researchers that develop and use them.

A good practice in object-oriented programming is the *separation of concerns* (Gamma et al., 1995), which states that any class should have possibly only one responsibility. This approach seems reasonable in this context too, and, in the following, we will try to break down the DARTHS into their main compositional parts.

1.2 A necessary first step: making data and formats open

DARTHS, obviously, are useless if there are no data available to run them; therefore, we briefly discuss some of the data aspects relevant to envisioning their design. There is a frequent lack of institutional and political will to publish

environmental data, sometimes for reasons of national security and other times to prevent misuse or for confidentiality. This state of affairs creates many obstacles, and open spatial data infrastructures (open SDIs) are still not common (Nedovic-Budic et al., 2011; Lehmann et al., 2014). This contrasts with the literature that, for some years now, has forecast a data-rich era for hydrology, coining neologisms like datafication (Mayer-Schönberger and Cukier, 2013) with reference to the upcoming ubiquitous presence of the Internet of Things (IoT) even in natural contexts. However, environmental data are public and supported by various initiatives in the US; in other countries, i.e., in Austria (e.g., <https://ehyd.gv.at>, last access: 23 September 2022), France (e.g., <https://www.hydro.eaufrance.fr/>, last access: 23 September 2022), Germany (e.g., <https://www.giz.de/de/html/index.html>, last access: 23 September 2022), and the UK (e.g., <https://archive.ceda.ac.uk>, last access: 23 September 2022), there are projects that aim to make the largest possible amount of data open to users, but data availability sometimes requires the payment of a fee. Since the survey by Viglione et al. (2010), there have been various initiatives to fill the gaps by various institutions, such as the Global Runoff Data Center (GRDC, <http://www.bafg.de/GRDC>, last access: 23 September 2022); the Global Water Monitoring System (GWMS, <http://www.gemstat.org>, last access: 23 September 2022); the Global Earth Observation System of Systems (GEOSS, <https://earthobservations.org/geoss.php>, last access: 23 September 2022); the World Meteorological Organization (WMO) Hydrological Observing system (WHOS); and, more recently, the Destination Earth architecture. It is also worth mentioning the bottom-up effort started with the Catchment Attributes and MEteorology for Large-sample Studies (CAMELS) datasets (Addor et al., 2017). Meanwhile, the Open Geospatial Consortium (OGC) has seeded several projects (<https://www.ogc.org/node/1535>, last access: 23 September 2022) to establish standards for the delivery and deployment of data related to hydrology. A review of initiatives, topics and issues can be seen in Lehmann et al. (2014) and visionary perspectives are presented in Nativi et al. (2021), to which we cannot add more. It is clear, however, that the general situation is still largely placed in between “fragmentation and wholeness” (Ballatore, 2014); this can only be resolved with general agreements and consensus about the usefulness of sharing data and time, without forgetting the question of providing the appropriate ground-based data for the needs of developing countries.

To complicate the matter, the new developments in satellite remote sensing have been producing massive amounts of data at unprecedented spatial and temporal resolutions. According to statistical data from the Committee on Earth Observation Satellites (CEOS), over 500 EO satellites have been launched in the last half-century, and more than 150 new satellites will be in orbit in the near future (CEOS, 2019). For instance, data from missions like Sentinel of the new European Copernicus Earth observation program (<https://>

//sentinels.copernicus.eu, last access: 23 September 2022) have already exceeded petabytes of volume. More and more data are expected to come: the American, Japanese, Indian, Chinese and European space agencies have already planned new missions for observing the planet. Notable examples are the NASA Cyclone Global Navigation Satellite System (CYGNSS) (Chew and Small, 2018) and Surface Water and Ocean Topography (SWOT) missions (<https://swot.jpl.nasa.gov/mission/overview/>, last access: 23 September 2022), the US–Indian NASA-ISRO Synthetic Aperture Radar (SAR) mission (NASA-ISRO, 2018), the Radar Observing System for Europe in L-band (ROSE-L) (Davidson et al., 2019) SAR missions, and the Thermal InfraRed Imaging Satellite for High-resolution Natural resource Assessment (TRISHNA) (Lagouarde et al., 2019), in addition to a constellation of low-cost commercial CubeSats (McCabe et al., 2017a).

So the question of how these data can be easily accessed and used by DARTHS is not a trivial one. Interesting solutions to facilitate this already exist, such as the Google Earth Engine (<https://earthengine.google.com>, last access: 23 September 2022) and the newly developed European OpenEO platform (<https://openeo.cloud>, last access: 23 September 2022). These Earth observation data centers have revolutionized the way users interact with EO data and offer integrated solutions to access large volumes of a variety of EO data with relatively high speed. With this new paradigm of “bringing the users to the data”, users no longer need to download and store large volumes of EO data, and they do not have the problem of dealing with the different formats and grids made available by the range of data providers. DARTHS should, therefore, provide an easy integration with these new platforms to facilitate the use of the wealth of data made available by the various Earth observation programs.

The state of art of the matter, for what can be important for DARTHS, is that data layers have to be somewhat loosely coupled to models via a brokering approach, where some intermediate tool takes care of discovering, gathering and delivering data according to what is requested. This approach has been successfully tested in the recent past (Nativi and Bigagli, 2009; Nativi et al., 2013) and allows the binding of heterogeneous resources from different providers. It is, for instance, at the core of the Destination Earth infrastructure and data space <https://digital-strategy.ec.europa.eu/en/policies/destination-earth> (last access: 23 September 2022).

A recent conference contribution (Boldrini et al., 2020) cites a few examples of brokers: WHOSWHOS, CUAHSI HydroDesktop (through CUAHSI WaterOneFlow), National Water Institute of Argentina (INA) node.js WaterML client (through CUAHSI WaterOneFlow), DAB JS API (through DAB REST API), USGS GWIS JS API plotting library (through RDB service), R scripts (through R WaterML library), C# applications (through CUAHSI WaterOneFlow) and UCAR jOAI (through OAI-PMH/WIGOS metadata). At present, both in research and in applications, the action of matching data and models is done offline by the researchers,

but the DARTH vision would require that these data be automatically ingested and processed. Therefore, DARTHS should take care of these aspects by design and be able to abstract data from the algorithms, as also explained in Knoben et al. (2021).

Regarding the data sharing/use as well as the fact that chunks of DARTHS could be distributed for inspection and modification by third parties, two strategies could be used. In the ideal situation, where there is a universal broker that (as a provided service) collects required data from the cloud in real time, DARTHS should include a connector to this broker so that all modeling could be performed smoothly (the cloud strategy). The opposite approach is that data are acquired once and for all and are stored locally (the local strategy). The local strategy is clearly unfeasible, due to the burden of data, while the cloud strategy is clearly still immature, even if ML has recently introduced new paradigms in data treatment such as the Apache Hadoop or Apache Spark distributed platforms (Nguyen et al., 2019) to facilitate these types of operations.

At a lower level of implementation, the issue of data treatment is managed through the standardization of formats. Standards have been developed for various scopes, including GRIB (Dey and Others, 2007), NetCDF (Rew and Davis, 1990), HDF5 (Folk et al., 2011) and database formats. Other, general-purpose standards, like Apache Arrow (<https://arrow.apache.org/>, last access: 23 September 2022) for tabular data, can be an interesting choice and come with various interfaces to the most common programming languages, thereby allowing for easier compatibility with existing models.

We are not claiming here that some formats should be preferable to others. However, a format and database architecture should be chosen as a starting point, along with a set of tools to transport data from the chosen format to other formats. The data format itself should be self-explicative and not require additional information to be understood. A long experience in this direction is available from UNIDATA (<https://www.unidata.ucar.edu/>, last access: 23 September 2022), from the WMO, from the ESA and from NASA. For any DARTH builder, therefore, it should not be difficult to agree on some conventional formats to start with coherently.

2 Design and implementations requirements

In this section, we delineate the main design requirements for DARTHS. The starting point is observing the variety of users and use cases in order to have a comprehensive view of the matter.

Roles	Hard Coders	Soft Coders	Linkers	Runners	Player	Viewers	Providers
Users							
Prime						●	●
Other End Users				●	●	●	●
Technical		●	●	●			
Researchers	●	●	●	●	●		

Figure 1. Following Rizzoli et al. (2006), it can be seen that models serve diverse scopes and uses; therefore, they are a fit-for-purposes task.

2.1 Starting from the people: glimpses of information about DARTH core architecture

As Rizzoli et al. (2006) realized, there are different types of users and different types of scopes when modeling hydrology; these are summarized in Fig. 1.

Hard coders (core coders) are those who actually design and write code. Soft coders are those who just modify existing codes and, if possible, develop plug-ins using public application programming interfaces (APIs). Linkers are power users who assemble existing codes through scripting languages or procedures. This role is of particular importance in ML, as Earth scientists very often build their models on top of ML engines provided by big companies or institutions (Nguyen et al., 2019). Runners (users) are those who execute existing codes. They just modify inputs, outputs and parameters, and they create and define scenarios. Players run scenarios and make analyses. Viewers view the players' results, having a low level of interaction with the framework, and maybe use them to instantiate policies. Providers provide inputs and data to all other user roles (and what was said in the previous section applies to them). The users' names are self-explanatory. In this paper, we concentrate on the specific roles of coders, linkers and runners with respect to technical and research users. We also have to mention that, even if hard coders seem the most basic of developers, they can be classified as “professional end user developers” (Kelly and Sanders, 2008). Scientists and engineers developing software have backgrounds in the theoretical models implemented in software, but they do not usually have a strong background or formal training in computer science nor software engineering (while computer scientists, in contrast, are not usually domain experts). The coders that we are referring to here are not usually computer scientists, instead they are domain experts (i.e., hydro-, bio- and/or geo-experts) with a limited computer science background.

To serve the needs of all of those roles and users, besides the scopes inherent to the DARTH vision itself, DARTHS need to be supported by an appropriately modular software infrastructure able to manage the different needs that

each user and role has. In addition, in thinking of coders/researchers, a DARTH infrastructure should be able to accept different modeling styles and programming paradigm changes, and DARTHS have to facilitate good programming and testing practices, bridging the knowledge gap of scientists/hydrologists in computer science. Having readable code is something that cannot be renounced (Riquelme and Gjorgjieva, 2021). It is a key aspect for a successful long-term development of the code: it improves maintainability and reusability (David et al., 2013), it saves time in the future, and eases the development and growth of the research community (Riquelme and Gjorgjieva, 2021). This topic is further discussed in Sect. 2.4.

As we are aiming to gather a community of developers to build and evolve a DARTH, the code must work seamlessly on the major operating systems (MS Windows, macOS, Linux or others), and this excludes efforts limited to one platform. Besides being programming-language-agnostic, the foreseen DARTHS have to be operating-system-agnostic. It seems a secondary limitation, but, in the present state of the matter, it is certainly not, even if the DE paradigm envisions a type of infrastructure operating over the web that encompasses the characteristics of single machines.

Other consequences follow if we look at the current hydrological models from the coders' position: they are codes written in various programming languages, these days primarily in Fortran, C/C++, Python, R, Java, C#, Julia and MATLAB. The variety of programming languages used brings a legacy that cannot be avoided with a blip, excluding the great part of programming researchers that has been at the core of the evolution of the field. Therefore, DARTHS either (1) need to be based on a platform which, while not being language-agnostic per se (as it is built in a specific language), is able to link libraries in all of the languages or (2) libraries themselves have to evolve to become (web) services, as often happens in ML. Only in this way will it be able to approach the mass of researchers and fully utilize the efforts already undertaken. This would bridge the past and future of hydrological and environmental modeling and make revolutionary changes possible in an incremental way. DARTHS should not subtract, just add and evolve.

There are at least two other model design requirements of DARTHS that have to be accomplished. The first is related to the specification of the data that a model actually uses. We cannot identify the external behavior of a model by saying, for instance, that it needs “precipitation”, but more precision is necessary about the units and the spatial and temporal dimensions. To continue with the given example, we need to specify the form of the precipitation (e.g., liquid-water precipitation), the spatial aggregation (e.g., catchment average), the temporal resolution (e.g., hourly) and the units (e.g., mm). In fact, this problem was raised quite a long time ago when first there was the need to give metadata, explaining what the values in each variable represent, and their spatial and temporal properties. The CF convention

(<https://cfconventions.org/>, last access: 23 September 2022) is one of the attempts to clarify these aspects, followed by other initiatives such as the Basic Model Interface (Jiang et al., 2017; Peckham et al., 2013), which contains a “grammar” for creating new names and is more specialized than CF in the hydrological sciences. The success of a DARTH depends on these specifications in which the freedom of choice is sacrificed to the adoption of some standard that makes the models potentially interoperable and discoverable on the web by their input and output requirements. Assuming that one such convention would be adopted by all models, a user would know which models they can connect to obtain the dynamics of a process that they want to estimate or predict. A black-box characterization of models can be done by accurately specifying their inputs and outputs (I/O).

The second is that researchers and the concerned users do not just care about the models’ I/O but also about their realism, reliability, replicability and reproducibility, and robustness, characteristics whose meaning is discussed thoroughly in the glossary. This information, which, in part, can be tested a posteriori by use and comparison with real-world observations, can also be gained by inspecting the models’ codes with their implementation details.

In summary, the DARTH architecture must account for various requirements, including

- being programming-language-agnostic,
- working seamlessly under the various operating systems,
- using standard, self-explanatory data formats, and
- using standard names for the quantities that they treat in input and output.

2.2 Overly complex systems do not serve the cause

Another characteristic deemed important for DARTH infrastructure is that DARTHS should not be invasive of programming habits (i.e., not forcing the programming to adopt constructs that computer scientists appreciate but scientists and engineers cannot manage). The available environmental modeling frameworks, which are infrastructures to all effects, can be classified into two broad categories: heavyweight frameworks and lightweight frameworks (Lloyd et al., 2011). The former is characterized by large and unwieldy APIs that require a considerable effort from developers (scientists or soft coders) to become familiar them with before writing new code. Moreover, such an effort somehow creates a strict legacy within the infrastructure, and this has limited the diffusion of these systems in the recent past (David et al., 2014).

Conversely, David et al. (2013) show how lightweight frameworks have many functions for the developer due to the techniques used to reduce the overall size of the API and the developer’s dependence on it. A lightweight environmental

modeling framework (EMF) fits easily with existing models as there is no interference with complex APIs. This is very useful for environmental modelers as it allows the use of existing modeling code and libraries, integrating them into a larger framework. In a lightweight framework, model components can work and continue to evolve outside the framework; thus, adopting and using a lightweight framework is easy. This type of infrastructure is the most suited to DARTH development as each research group is not strictly bound to a specific infrastructure and it becomes easy to include new modeling solutions in existing models.

2.3 Participatory needs

If the replicability of results is guaranteed by the availability of models, data and knowledge of the simulation setups, the clearest level of knowledge can be obtained when the codes are open source and can be analyzed. Being open source is not a mandatory requirement but certainly helps the structuring of open science (Hall et al., 2022) and is highly desirable for the openness of science and code inspection by third parties. In fact, there is no such thing as “black-box science” (Stodden et al., 2013), and the peer review process alone is largely ineffective at “sniffing out” poor validation and testing of model content (Post and Votta, 2005).

We are committed to open science in this paper; therefore, in the following, we take for granted that the codes are open source, provided in an open repository, with an open license and with the building tools (i.e., the tools to compile the source codes).

If we then consider the point of view of the runners, we are interested in the reliability of the content and that runners can give their findings back to the system and enrich the knowledge base.

A new paradigm can be formulated as “participatory hydrological modeling”, indicating that it can be used by many researchers who can seamlessly cooperate to produce added value for the DARTHS. Not only do we aim to have a multitude of users, there should also be a multitude of developers to exploit science by providing bottom-up contributions. These requisites have some consequences in the functional design of a model. Take, for instance, the case of the Object Modeling System (OMS; David et al., 2013): besides a fully configurable setup, it provides a standard organization of the files used for modeling (as presented in Fig. 2). A runner can be provided with all that they need (e.g., input data, parameters values, input and output file names, and modeling solutions’ structure definition) to run their simulation (in the “dist” folder), in this case locally on their own computer, including the source code, the size of which is quite irrelevant with respect to the data (in “data” and “output” folders). The preprepared simulation is described and governed by “.sim” files in the “simulation” folder that contains the workflow of what is going to be executed with all of the required information. The runner can then add new simulations, with changed

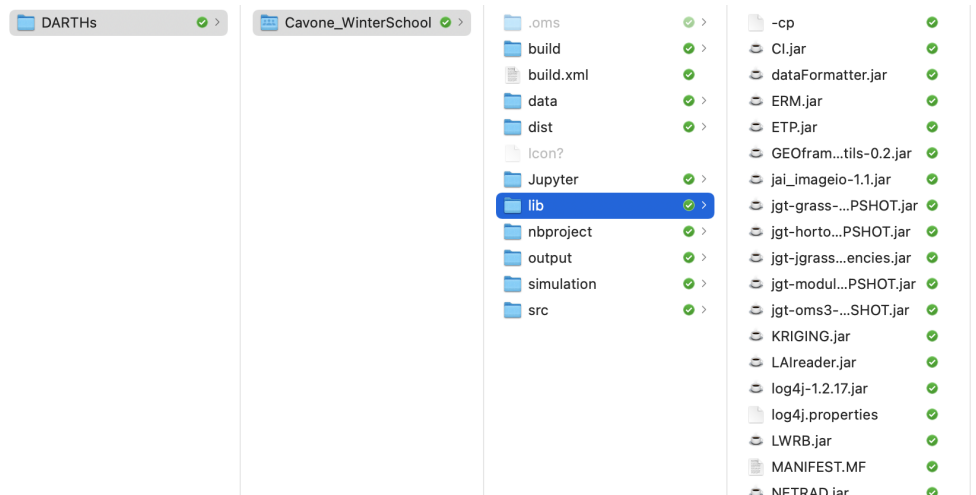


Figure 2. Standard organization of the Object Modeling System (OMS3) files, highlighting the libraries used.

parameters, modified spatial partitions and input data, and can distribute all of their improvements and results back to the community with which they are interacting. This procedure, although not yet automated, has already been widely used by the authors of this work in their projects (where collaborative efforts were necessary). The idea can no doubt be improved, and the organization provided is very simple, based on a file system organization; much more sophisticated architectures, using databases and web services, could be deployed to obtain the same functionalities with automatic procedures, similar to those used in the version control system for codes. If we then assigned a unique identifier to any catchment worldwide, the procedure could even be decentralized and certified by using, for instance, blockchain-like technologies (Serafin et al., 2021). However, as simple as it is, it is a proof of concept that such a strategy of participatory modeling is possible.

2.4 Four, plus one, steps towards DARTHs

As described in the previous sections, DARTHs can be envisioned as a distributed ecosystem of different software packages that serve multiple roles and users and in which models and data can move around the web. To get some order within the models' design, computer scientists distinguish between various levels of interoperability, according to the openness, digital portability and client-interaction style that characterize a given tool, i.e., a data-/process-driven analytical model provided as a digital software or service. Nativi et al. (2021) distinguish three levels, but we identify five with the following definitions (which are presented with more detail in the glossary):

- *Model as an Application (MaaA)* is the case for most of the existing classical hydrological models (Rizzoli et al., 2006). Traditionally engineered models tend to be

“applications”, meaning that they bundle together all of the features that are required to have an all-round modeling experience, but these are exactly the opposite of what is needed by DARTHs, where everything must be provided as a service and loosely tied. Knoben et al. (2021) give a further clear description of MaaA. Most current models fall into this category.

- *Model as a Tool (MaaT)* is an evolution of MaaA that allows interaction with the model. Actually, the runner interacts with a software tool (i.e., an interface) developed to utilize the model, not with the model itself nor a service API. The implementation of the model runs on a specific server (or locally), and it is not possible to move the model and make it run on a different machine. Just a few of the recent modularized systems fall into this category, as can be deduced from the survey that we present in the Supplement.
- *Model as a Service (MaaS)*, as for the previous case, refers to the fact that a given implementation of the model runs on a specific server; however, this time, APIs are exposed to interact with the model. Jupyter (Loizides and Schmidt, 2016) itself can be seen as an infrastructure that promotes such an approach, even if some functionalities are missing. The recent eWaterCycle (Hut et al., 2021) collector of models can also function this way. OMS3/CSIP (David et al., 2013) and Craig et al. (2020) are other examples that almost fully implement this type of infrastructure.
- *Model as a Resource (MaaR)* refers to the fact that the interoperability level follows the same patterns used for any other shared digital resource, like a dataset. This time, the analytical model itself (and not a given implementation of it) is accessed through a resource-oriented

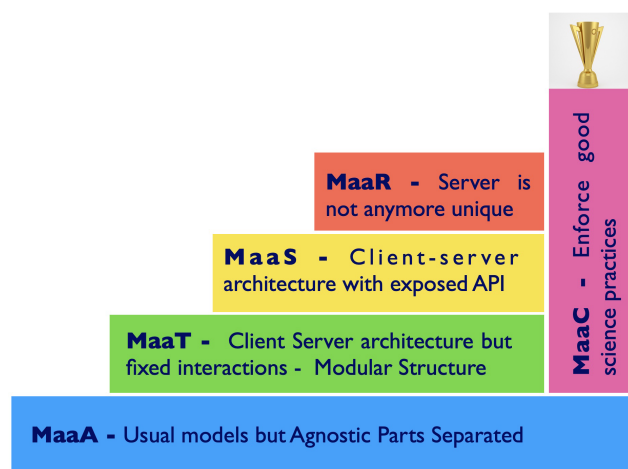


Figure 3. The convergence of present models to DARTH-aware models can be visualized using a few steps; these steps involve the introduction of some software layers, starting from present modularized models (MaaS). These layers include the structuring of client–server architectures, initially with a fixed client interface and server (MaaT); a client–server architecture where the connections can be obtained machine to machine by APIs (MaaS); and a multiplicity of client and servers where models can be dispatched at runtime according to the computational needs without user intervention (MaaR). At the same time, in order to fulfill the realism and reliability requirements, a certain number of open-science-aware models that fall under the name of MaaC should be pursued.

interface (i.e., API). Moreover, there is a software infrastructure layer, uploadable on heterogeneous hardware, that manages (not directly by the user) a set of compliant models. This software layer allows one to move models and make them run on the machine that best performs for a specific use case among a large pool of computing facilities without the need for user intervention (at the model level).

MaaA, MaaT, MaaS and MaaR can be thought of in order of suitability for DARTHS to fit the paradigm of the DE concept. MaaA models are good, for instance, for those who want to claim the intellectual property rights or for commercial environments, but they are a bad thing for science. Their code is a block, making independent revision nearly impossible. Testing new features, which is an intrinsically problematic issue (Kelly and Sanders, 2008), becomes almost unviable in MaaA. In fact, these models have been implemented as monolithic codes, and the absence of separation of concerns makes it difficult to read/debug them (Serafin, 2019). If a modeler is interested in using a particular MaaA function, this is actually not possible, and coders are re-implementing the same things over and over. MaaA models are usually defined as “silos”, in the sense that they cannot exchange data and procedures and do not favor the exchange of knowledge between related disciplines.

MaaT models are a step forward that MaaA can gain after a robust refactoring. They preserve the benefits (from MaaA) of a strong control of the model’s use and execution; however, the limitations on the usability and flexibility of the model are evident. Within MaaT, everything is controlled by the developers, who not only establish how and when the models can be used but also control the model evolution and enhancement. Clearly, if the provider is an organized community with rules for obtaining contributions, the model can be a “community model”, in the sense, for instance, that CLM5 (Lawrence et al., 2019) and JULES (Best et al., 2011) are.

MaaS interoperability consists instead of machine-to-machine interaction through a published API (e.g., for a run configuration and execution). Nevertheless, it is not possible to move the model and make it run on a different machine transparently without human intervention.

The previous three levels establish user dependence on the provider of the modeling services. What we should aim for with DARTHS is to accomplish the more flexible possibilities represented by MaaR. MaaR allows one to effectively move the model and make it run on the machine that performs best for a specific use case, with clear benefits in terms of scalability and interoperability. MaaR obviously requires a flexible infrastructure that allows models to be built rapidly and openly as well as protocols to make the parts of the system work together without side effects. While fully-fledged MaaR models do not yet exist, the Cloud Services Integration Platform (CSIP; David et al., 2014) and the Community Surface Dynamics Modeling System (CSDMS; Peckham et al., 2013) are examples of something between MaaS and MaaR. However, thus far, they have not been as widely adopted as they could be, indicating (1) some possible complexity in their use that coders and runners could not face properly or (2) some missing action in disseminating their added value among scientists. In the era of big data, various frameworks can provide executions of software tasks over the web. For instance, Kubernetes (<https://en.wikipedia.org/wiki/Kubernetes>, last access: 23 September 2022) is an orchestrator of containers that automate software deployment and management that could be used for some of these tasks. Airflow (<https://airflow.apache.org/>, last access: 23 September 2022) is a manager of workflows that could be arranged to manage these MaaS and MaaR software layers, even if DARTHS would require a more “declarative” (see the glossary) interactivity than the one that Airflow implements.

The informatics itself, however, does not guarantee the scientific content of the models that can be run on the infrastructure. Therefore, we claim that a further (fifth) step has to be made that covers both the science and technological sides:

- *Model as a Commodity (MaaC)*. In the case that models can be chosen from a pool present in the cloud, not only can they be automatically connected and made to run with minimal intervention to respond to the needs of

the users demands (as essentially required by MaaR) but they can also be contributed to, modified and expanded by the technical user to fit their purposes. Thus, the pool increases as science advances. MaaC models are also required, by construction, to embed tools for assessing their reliability; to support literate computing, as supported for instance by notebooks (Lau et al., 2020) (but not limited to them); and to allow for hypothesis testing.

MaaC models are demanding, as they should make the process of coupling models and the addition of features as automatic as possible, with the least possible input from the runner or coder. Currently, some conceptual work still needs to be done in this direction, as pointed out by the example of the infiltration model discussed in Peckham et al. (2013). As shown in Fig. 3, many of the MaaC requirements can be built in parallel to the infrastructures that implement the requirements of MaaT through to MaaR.

The listed characteristics of MaaC have implementation consequences that will be discussed in the following sections. A key feature of the most evolved modeling infrastructures is the possibility to break models down into parts that can be reassembled for a specific purpose, which depends on the ability of a modeling infrastructure to make model parts interact as well as to allow them to be modified, evolved or changed. A mature DARTH requires MaaC models.

2.5 Back to core software engineering for models

Building models differently from what was traditionally conceived (Voinov and Shugart, 2013) is the way to achieve the flexibility required by DARTHS; this spans different methods, roles, uses and the need to describe different resolutions and scales, and it requires that one considers new assumptions and paradigms and extends their scientific domain over several traditional disciplines (Savenije and Hrachowitz, 2017; Bancheri et al., 2019b).

Unlike past practices, structuring the software into composable parts, called “components”, should become the standard. Figure 4 shows the functioning of such a component, Prospero, that estimates evapotranspiration (ET). The Prospero component uses two other components: one describing the canopy and one describing the constraints (stresses) limiting evaporation. Each component exchanges information with the other through their inputs and outputs, as indicated by the arrows in Fig. 4. All internal components are intrinsically not accessible at runtime (according to the information-hiding principle – for its definition, please refer to the glossary), and the core algorithms of the components can be changed in favor of others that perform faster or more realistically without altering the overall scope of the system (i.e., estimating ET).

Components can be interchanged by the runners, chosen from a pool and linked together (in this case, illustrated using a script in a domain-specific language, DSL).

This modeling-by-component (MBC) approach has actually existed for more than 40 years (Holling, 1978), but it is only in the last 20 years that it has gained momentum in the environmental modeling community (Argent, 2004). Often, it has been referred to as “integrated environmental modeling” (IEM) because it stems from the need to study heterogeneous processes together and integrates knowledge from various disciplines (Moore and Hughes, 2017).

There are only a few examples of MBC in the more restricted hydrological and meteorological community, including TIME (Rahman et al., 2003), OpenMI (Gregersen et al., 2007), CSDMS (Peckham et al., 2013), ESMF (Collins et al., 2005) and OMS (David et al., 2013). A longer list can be found in Chen et al. (2020).

MBC concepts and the technological consequences of this method are tempting, but their real deployment can suffer from “invasiveness” in some cases (i.e., they may change the habits of a good programmer, e.g., Lloyd et al., 2011), and they require quite an adaptation of the usual programming styles. Among these abovementioned examples, OMS faced this issue, explicitly finding encouraging solutions (Lloyd et al., 2011). MBC promotes a server-oriented architecture (SOA) of the software, which is the same type of software architecture that was requested for treating heterogeneous data sources. In principle, the SOA framework can work on different machines and is scalable across various hardware architectures. The final coder or user does not have to take care of the details of the computational engines because the framework itself takes care of it. Good features of the MBC approach are as follows:

- The framework employs encapsulation, which simplifies code inspection. The components can work in a stand-alone manner (supported by a given infrastructure), and each of them can be tested separately.
- The method allows for well-established ownership of intellectual property. Each component usually has a few contributors, and different components involve a diversity of developers without being dispersed in thousands lines of code. Thus, the addition of components is always possible and does not require a recompilation of the whole system.
- The substitution of components is easy, and the use of components in hypothesis testing is favored (Beven, 2019).
- Basic services, such as implicit parallelism or tools for the calibration of model parameters, are provided under the hood, as explained in the next section.
- In a well-designed system, the composition of modeling solutions is practically unlimited, and components can accomplish a wide range of tasks, not necessarily from the same discipline (i.e., silos are avoided).

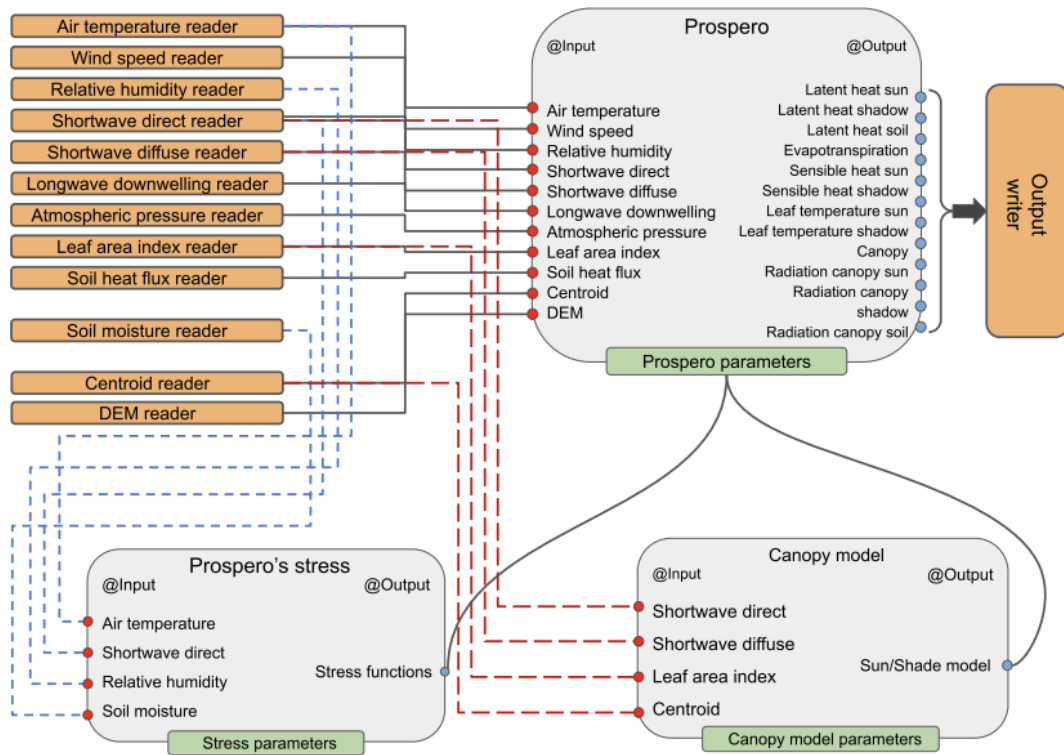


Figure 4. A visualization of models components. In this case, Prospero is a component of the GEOframe system (Bottazzi et al., 2021) that estimates evapotranspiration.

Therefore, this way of designing and organizing codes is the natural candidate for fulfilling the requirements of DARTHS and for providing the building blocks of futuristic MaaC infrastructures. However, MaaC has to be treated with care, as it can potentially lead to the spreading of unreliable or untested models. Hence, MaaC models should not be limited to simply providing the results of simulations; instead, they should be equipped by design with a collection of tools for assessing the degree of reliability of the results and for quantifying their errors (uncertainty). This aspect requires further investigation in the technicality of MaaC.

2.6 How to write and manage models

If MBC is a necessary ingredient of DARTHS, then we have to understand how to write the code inside the components. The first requirement is that the code has to be clear: in fact, in order to accomplish the open-science ideas, it can be argued that not only does the code have to be open source but it also has to be understandable. Millman and Pérez (2018) offer an overview of good practices to be followed. Years ago, Donald Knuth (Knuth, 1984) put forward the concept of literate programming. He proposed tools to enable users to integrate texts and figures with code and then, with additional tools, to prepare documents in which the code is explained. The idea, however fascinating in principle, did not really gain momentum among programmers, even though

there were programming-specific tools such as WEB (Knuth and Levy, 1983), Sweave (Leisch, 2002), Knitr (Xie, 2013) and others that would have allowed it.

It was Martin (2009), in fact, who proposed that the best documentation of the code is the code itself and its appropriate organization. With respect to the vision of Knuth (1984), some factors play a major role today, particularly the birth of high-level languages that allow a greater expressivity than was possible in the early years of high-level languages. In the early days, memory was so tight that variables and other names were usually single letters, often reused from scope to scope, and looking at a code base was more decrypting than reading. Presently, languages allow the use of more expressive names, and most of the languages have standard ways of inserting comments (to the point of making them invasive sometimes), which can be processed to produce documentation. One of the clearest examples is the Javadoc tools (Kramer, 1999).

Another issue that can arise with open-source codes is the organization of the classes, which is obscure most of the time, and often what is gained in clarity by having classes with short code contents is lost by having hundreds of them without having a clue of their use in sequence. External documentation with Unified Modeling Language (UML) diagrams can help this phase, but it is usually a neglected part of the documentation. Some languages, such as Java, have re-

cently introduced the concept of modules that serve to specify the dependence of one part of the code on others. A further step on this topic is to adopt state-of-the-art building tools, like Maven (<https://maven.apache.org/>, last access: 23 September 2022) or Gradle (<https://gradle.org/>, last access: 23 September 2022), that collect all of the dependencies needed to compile the codes and, if well written, at least clarify the dependences at the package level. These tools are much more evolved than the traditional “make” command of the C language, as they allow one to grab codes from globally diffuse repositories and keep each building process up to date, along with the latest version of the libraries that they use.

Going even more in depth into the programming, one aspect that is usually ignored by scientific programmers is that object-oriented programming is not just about using classes but also about organizing them efficiently as projects grow in dimension, as DARTH philosophy requires. These aspects are treated in “design patterns” (Gamma et al., 1995), which offer a mature understanding of what object-oriented programming is, although they are often not part of elementary books on programming. A few books indeed cover the issues for scientific computing, such as Rouson et al. (2014) for Fortran 95 and Gardner and Manduchi (2007) for Java. Another step forward can be made by using a certain degree of abstraction in software, as proposed, for instance, by Berti (2000), with examples in C++, and used in Tubini and Rigon (2022), with Java. Abstraction, one of the basic rules of design pattern (i.e., “program to interface and not to objects”), is important in that it allows the software “not to be modified” but “extended”, thereby enormously increasing the flexibility of a project and its lifetime. This software engineering technique enhances the flexibility already available with MBC and permits it to limit the code dimension and, when well engineered, a massive code reuse.

All of the above prescriptions can have different deployments in different languages, but the overall principles are as follows: reuse code and make it shorter, easy to read, and easy to extend while controlling modifications that can be disruptive to the stability of the whole.

Recently, a practice has been introduced by the *Journal of Open Source Software*, JOSS (<https://joss.theoj.org/>, last access: 22 September 2022), to review the model codes as part of the peer review process. It is a possibility that DARTH coders should not overlook. It is clear, in fact, that the credibility of the results must be accompanied by the “inspectability” of any part of the research and, when this is based on extensive computing, it must include access to the code’s internals.

To DARTH, the concept of literate computing (Rädle et al., 2017) can be more useful than literate programming itself. This is, for instance, promoted by the Mathematica notebooks (<https://www.wolfram.com/notebooks/>, last access: 23 September 2022) and, more recently, by the Jupyter notebooks (Loizides and Schmidt, 2016) and others (Lau

et al., 2020). One necessary characteristic of open science is to keep track of the setting of performed simulations and of the data manipulation used to infer results. Some frameworks, like OMS (David et al., 2013), have internal mechanisms to do this, by commanding the simulation by means of scripts that are subsequently archived in standard places with standard name extensions. However, notebooks, with their mix of scripting, visualization of data and comments, greatly enhance the documentation of a creative scientific process, even if their role should not be overemphasized. For instance, the standard operative approach for GEOframe users is to support the preparation of the analysis of inputs and outputs with Jupyter notebooks. Until more fancy ways of interfacing with models are available, these remain the best way to expose all that has been done to a scientific audience.

2.7 Assessing the reliability of DARTHs by design

An essential aspect of MaaC is the estimation of uncertainty in any type of prediction. This is essential from a practical point of view (i.e., for those activities that rely on model predictions for their planning activities) for the advancement of science, in the continuous activity between ideation, hypothesis testing and refinement. Furthermore, it is also essential for the viewer of the results, including citizens, to understand the reliability of the results. According to our definition in the glossary, reliability is a relative concept, which itself requires studies, but models do have a degree of reliability if each of their estimations always comes with an estimation of its uncertainty.

We are taught that we cannot validate models but, with the feeble light of statistics, we can try to understand how confident we can be in a certain prediction. Götzinger and Bárdossy (2008) and Beven (2018) present a very clear summary of the issues, but there is a multitude of other contributions that treat the topic. Binley et al. (1991) and Todini (2007), among others, dissected the problem related to the forecasting of discharges, but any model of any type involves errors and has its own error-producing mechanisms, as can be seen, for instance, in Yeh (1986) or Hill and Tiedeman (2007) for groundwaters, Vrugt and Neuman (2006) for vadose zone variables, Post et al. (2017) for ecohydrological modeling, Vrugt et al. (2001) for root uptake, and Yilmaz et al. (2010) for general watershed modeling.

Literature is often concerned with where the errors are. Uncertainty comes from errors in data, epistemic errors, errors in methods for parameter calibration and from heterogeneity (i.e., the variability in the domain to be described and modeled). The last example could be included in the epistemic error, but citing it separately serves to stress the point. Part of the uncertainty certainly comes from ignorance of conditions that, being unknown yet necessary to completely define the mathematical problem, must be guessed.

Errors in measurements affect the procedure of calibration and cause the inference of incorrect model parameters. Er-

Full ERM model for one HRU

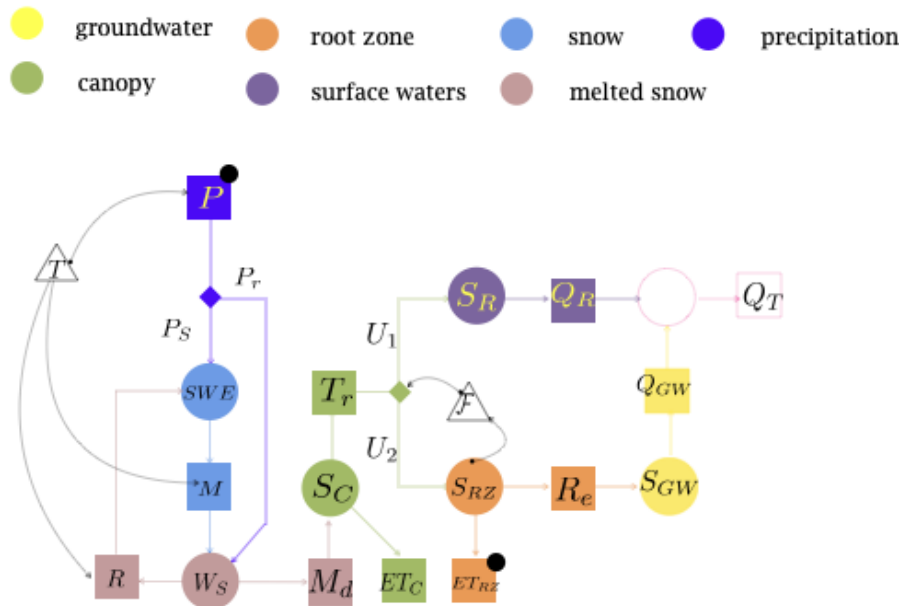


Figure 5. Representation of the Embedded Reservoir Model (Banheri et al., 2019a) that is based on a system of six ordinary differential equations that represent the hydrological fluxes at a point. HRU stands for hydrologic response unit. The reader is referred to the text for further explanations.

rors in model structure reflect in incorrect forecasts, which, in turn, cause biased comparisons between these forecasts and the models outputs.

Figure 6 portrays the life cycle of a model: circles represent data, and boxes represent actions. Input, boundary and initial conditions can contain errors that propagate to the model results, leading to different responses than the control measurement set. Model results contain bias and variability generated by errors in the model structure, which also contains parameters or hyperparameters, if the model is a machine learning one, to be determined. A phase of calibration/training follows to change the model’s parameters by evaluating a metric of the difference between predictions and measurements (goodness of fit, GoF). During this phase, the model is run with different sets of parameters until the optimal set of parameters is found, which correspond to the prescribed GoF value. When a satisfactory agreement is reached, the model can go into production or to a further level of analysis with stakeholders.

Once calibration of the model has been done to bring results up to a pre-established level of acceptability, a further evaluation of the model’s results must be done to test the adequacy of the model to solve the problem that it was intended to address (Refsgaard et al., 2007).

Usually, the decision-making process implies the existence of a group of stakeholders with a clear methodology or at least a set of beliefs on which to base their final judgments. However, this is not necessarily the case of DARTHS, where,

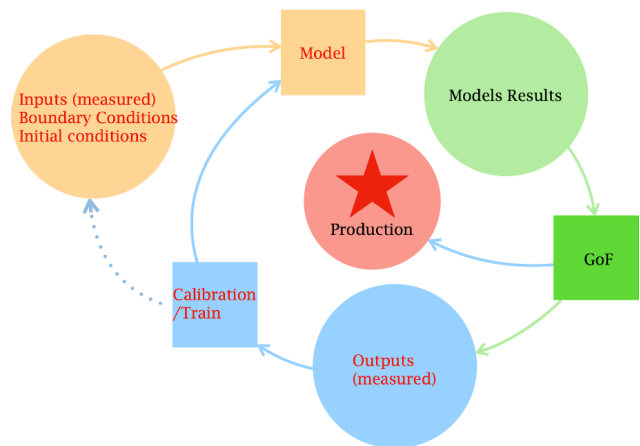


Figure 6. The life cycle of a model.

in principle, people can ask for model results without having an informed background. Therefore, even though the range of results that a model provides is restricted, given the laws of dynamics and thermodynamics, DARTHS should by design give appropriate warnings when results require discussion or interpretation. Among the myriad procedures for calibration, sensitivity analysis and data assimilation, error estimation is more art than science: while the methods are rigorous, the assumptions under which they work are of variable credibility depending on the process. For instance, Refsgaard et al. (2007) recognizes at least 14 methodologies (as well as gen-

eralized likelihood uncertainty estimation – GLUE) to obtain these estimates. Here, we do not advise the use of any particular one of these methods, but we do claim that at least one method should be chosen. If, for the sake of the advancement of science, the search for the origin of errors is paramount, we stay (for that which regards DARTHs) with the simplest fact: “purely empirically, probability and statistics can, of course, describe anything from observations to model residuals, regardless of the actual source of uncertainty” (Cox, 1946). Moreover, we want to reinforce the idea that error estimation is a practice that has to be continuously exerted and refined. If the comparison between computed and measured quantities is systematically done in DARTHs, the statistics of the performance of a certain model setup becomes more reliable with time. As a consequence, we can easily figure out the success rate of a model’s predictions and use it as a baseline for future decisions. At the same time, using experience and comparison, we can improve the reliability of the methods used to assess uncertainties. That is to say, we can test and improve both the results and the methods to determine their precision.

Model assessment occupies the space between parameter determination (Gupta et al., 1999), sensitivity analysis (Pianosi et al., 2016), data assimilation (Reichle, 2008) and decision-making (Refsgaard et al., 2007).

3 Enabling technologies and DARTHs

DARTHs are themselves an enabling technology. However, according to recent developments, they can also contain new or relatively new “technologies”. Those envisioned here are high-performance computing, which is actually a necessity for DARTHs; Earth observations, which can satisfy the data voracity; and machine learning approaches to simulations. These technologies have entangled functionalities, as we describe below.

3.1 Dealing with the computational burden

A DARTH needs to be supported with the appropriate informatics that allow it to distribute parallel simulations and search with appropriate scalability among multi-core machines and cloud infrastructures; these are further requests for HPC applied research.

The DARTH metaphor requires an extremely large use of data exchange in the background, which requires extremely high computational power.

Efficiency in a DARTH must be achieved in many ways – for instance, data gathering and ingestion: from the coder to the runner, it is important to have self-explanatory data formats that are efficient with respect to memory (Lentner, 2019). DARTH action, as it has been envisioned so far, can be represented by a graph, where the nodes are computational points, and the arrows represent potential exchanges of

data. In Fig. 4, we have outlined what a chunk of a DARTH should be (without parallel routes). A more complete view is presented in Fig. 5, where, according to the extended Petri nets (EPN) representation of hydrological dynamical systems (Bancheri et al., 2019b), the model is formed by the resolution of six ordinary differential equations (ODEs). In Fig. 5, each ODE is represented by a circle, and the model fluxes are represented by squares. As the graph shows, there are at least two independent paths that produce (as a sum) the final discharge (Q_T), and these paths can be run in parallel. If we write the informatics that solves the system of equations, each of the ODEs (i.e., the circles) is mapped to one or more DARTH component, and some other components are required to estimate the fluxes (i.e., the squares). For instance, the green evapotranspiration flux (ET_c) in Fig. 5 corresponds to components represented in Fig. 4. Therefore, stripping down the mathematics can reveal further possibilities for parallelization. In the hands of computer scientists, this can be done – for instance, it has been studied in the OMS framework (David et al., 2013, 2014). In fact, OMS/CSIP uses knowledge of the component connections to run them in parallel over multi-core machines or on cloud computing services.

Another aspect relevant to hydrologists is that the Earth’s surface can be tiled into catchments and subcatchments (Rodríguez-Iturbe and Rinaldo, 2001). This organization of the space can be utilized to better organize the computation and reduce the information that needs to be passed between the areas. In Fig. 7, 32 hydrologic response units (HRUs) and two lakes are distinguished, and each of them can be partially processed in parallel. For instance, the Net3 infrastructure (Serafin, 2019) has been deployed in the OMS core and serves this scope. It treats the external HRUs (shown in orange at the bottom of Fig. 7) independently and then moves down the computation following the channel network. It is worth noting that, if the parallelism is obvious when two paths separate, such as in the graph in Fig. 5, a pipeline type of parallelism (McCool et al., 2012) can also be implemented for the linear sequences of models represented by a tree-like graph. In a pipeline, one component gets the input from another; thus, when the graph is straight and acyclic, the work done by a process, represented by a node in the graph, can be performed by a group of processors or cores and, once one of these is freed up, it can be used again for a new job. These techniques are well known to computer scientists but usually not to hydrologists. In Fig. 7, each square node corresponds to a system that is, in most cases, equal or similar to the one presented in Fig. 5. Therefore, the underlying infrastructure has to harmonize the various levels of parallelization: the one at the component level (multiple ones for each node) with the one at the river network level. This actually happens, for example, in OMS, and we refer to it as “implicit parallelization”.

Implicit parallelization has the great advantage that its action is completely hidden to the coder who implements the

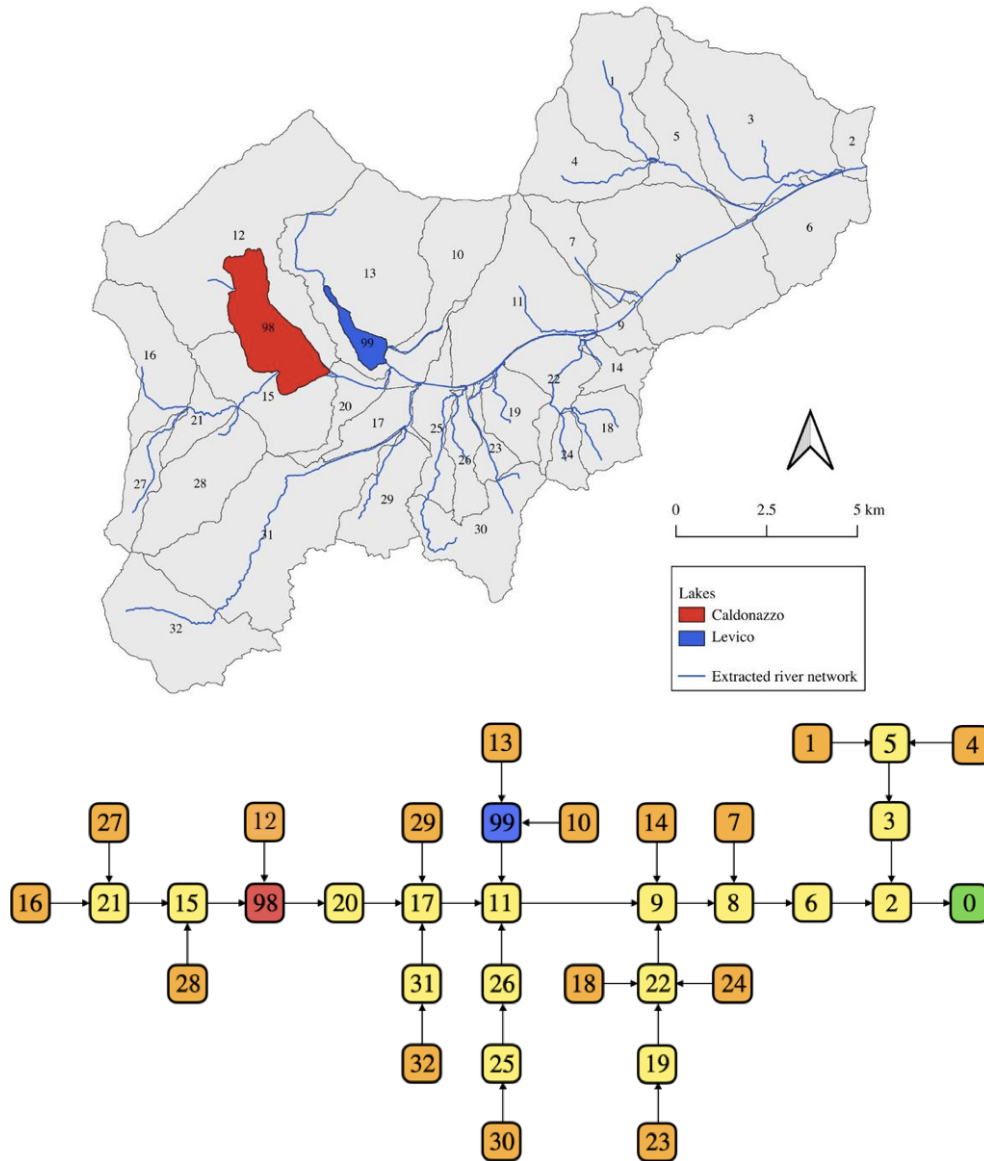


Figure 7. The upper Brenta River (Italy) is a catchment subdivided in 34 hydrologic response units (HRUs), including two lakes, Caldonazzo and Levico. The channel network gives an ordered structure to the geographic region that can be used to process the HRUs in parallel (modified from Busti, 2021).

components, and its mechanism does not affect the way the coder implements the processes. If the framework developers find more efficient ways to achieve it, they can change the engine without the hydrological code necessarily being changed.

Purists of HPC could argue that these forms of computation are far from the efficiency of codes tuned in an assembler to obtain the maximum possible performance out of any hardware. However, what matters here is the coders' and runners' ease of use and implementation, as it can lead to the greatest improvements. In well-designed DARTHS, coders can completely ignore the computational pains in paralleliz-

ing the codes and instead focus on the tasks that they have to solve (i.e., to cook their ingredients instead of taking care of building the kitchen).

It is worth mentioning at this point, with an eye on participatory computation and programming, that the spatial partitioning of catchments offers the possibility of using different types of models in different locations. For instance, the two lakes in Fig. 7 (blue and red nodes) are modeled with a different set of equations than the yellow and orange nodes. Moreover, the runs performed for the characterization of the subcatchments that are upstream of the lakes, for example, can be performed by different runners who can share just the

final results of their work. To this end, however, catchments (and their partitioning HRUs) need to have a unique identification, ultimately all over the world, although this is what latitude and longitude and their decimals were invented for.

One possible objection to what has been presented is that process-based models (Fatichi et al., 2016b), in the traditional sense like ParFlow (Kuffour et al., 2020), GEOtop (Endrizzi et al., 2013), HydroGeoSphere (Brunner and Simmons, 2012) and SHE (Refsgaard et al., 2010), which have outstanding numerics and capabilities, require a gridded partition of the Earth's surface that does not seem to fit in the nested-graph structure just envisioned. This objection is only partially valid, however, as process-based models have functional parts that can be refactored to be loosely coupled in components, while the computational grids can themselves be advantageously disjointed into spatial parts, and their computation can be organized in a graph-like structure. Certain parts of the codes, though, could remain tied to the grid structure to which other forms of parallelism methods can be applied. For these, other strategies of computation can be envisioned, in which global computations, run by institutions, can be used to drive local computations, similarly to what is done with meteorological models. The use of process-grid-based models, in fact, constitutes a third type of HPC demand that should be harmonized with the others. In summary, various levels of optimization of the computational resources can be activated in DARTHS, depending on modeling choices. However, as we have shown, some of them can remain transparent to the coders and the runners, simplifying their work so that they can be concerned with just the hydrology/biogeochemical physics of the processes, rather than with the informatics. All of this depends upon the adoption of the right framework, which appears, at this point, to be very necessary.

3.2 DARTHS as a bridge between Earth sciences and Earth observation

There is a pronounced trend in current hydrological research to use Earth observations (EOs) as the basis of the digital twin Earth (DTE). The ESA and NASA have enthusiastically embraced the idea within their programs (https://www.esa.int/Applications/Observing_the_Earth/Working_towards_a_Digital_Twin_of_Earth, last access: 23 September 2022), and, in lay people's imagination, the visualization of their resources on virtual globes is what comes closest to the idea of DE. From the standpoint of hydrological sciences, EO data, specifically those derived from space-based sensors, have provided new and independent datasets that span the range of water cycle components (the reader is referred to McCabe et al., 2017b, Lettenmaier et al., 2015, and Babaeian et al., 2019, for further details). The usefulness of Earth observations, however, lies not just in their capacity to reveal insights on the water cycle but also in their potential to benchmark Earth system models. The latter is particularly important when dealing with the representation

of human processes by models (Müller-Hansen et al., 2017), as it is undeniable that the interaction between human activities and the hydrological cycle is currently stronger than ever (Abbott et al., 2019; Wada et al., 2017).

Despite the huge availability and variety of data from EOs, the extent to which current hydrological models can efficiently and effectively ingest such massive data volumes is still poor. In meteorological and atmospheric sciences the exploitation of EO data has been supported by the development of community-based models and data assimilation (DA) systems, like OpenDA (<https://openda.org/>, last access: 23 September 2022) and the Land Information System (LIS; Kumar et al., 2006; Peters-Lidard et al., 2007). On the other hand, this has only partially happened in the hydrological community. The reasons for this range from differences in the scale of applications (hydrologic hindcasting and forecasting has been more oriented to smaller spatial and temporal scales) to aspects of the underlying physical system (i.e., atmospheric vs. hydrological systems vs. ecosystems), with hydrological systems and ecosystems characterized by different, highly nonlinear models for a variety of processes that existing DA and EO techniques may be unable to handle (Liu et al., 2012; Kumar et al., 2015).

If both communities embrace the DARTH logic explained in the previous sections, the hydrological community should learn from the meteorological and atmospheric communities and accelerate the transition from fragmented hydrologic data assimilation research towards community-supported, open-source systems that can operate an efficient ingestion of EO data. The EO community can introduce more sophisticated retrieval processes, can benefit from the granularity permitted by the MBC approach, and can use the mass and energy conservation laws to better define what EO sensors see. Besides using various sensor data, current practices in deploying EO-based products also use hydrological assumptions, such as simple water balance schemes, and different formulations for losses, such as drainage and evaporation (Martens et al., 2017; Manfreda et al., 2014; Brocca et al., 2014). Then, as often happens, these EO-based products are used by hydrologists as forcing or calibration datasets in their hydrological models with the purpose of obtaining the discharges out of a catchment. The hydrological models, however, usually have a different structure with respect to the ones that the EO products are based on, resulting in potential spurious uncertainty and not always optimal results (López López et al., 2017). Instead, the DA could be completely incorporated with the simulations without the need for the intermediate steps in which the EO community deploys its products and the hydrological community its models, and the outcomes of both are compared at a final stage. Papers like Martens et al. (2017) and Meyer et al. (2019) can give guidance on how to achieve this.

Generally speaking and with few exceptions, geophysical variables that are derived from satellite data are obtained via complex retrieval models with numerous underlying assump-

tions – assumptions that are themselves different from those used by hydrological models. This workflow could be greatly simplified if model components could directly assimilate the spectral signatures of solar and Earth-emitted radiation. To achieve this, DARTH should be equipped with model operator components (e.g. backscatter and radiative transfer models) that are able to use states, fluxes and ancillary information derived from EO data to directly ingest brightness temperature and backscatter observations (De Lannoy and Reichle, 2016; Lievens et al., 2017; Modanesi et al., 2021). In this way DARTHS would represent the bridge between the hydrological and EO communities and would facilitate the participatory science that we are envisioning here.

Other challenges emerge when trying to integrate EO data with hydrological and land surface models. These are related to the mapping of observations to model variables, to the specification of model and observation errors, and to the homogeneity and harmonization of EO data (McCabe et al., 2017b). It is of common interest to solve the mismatches that have been found between hydrological/land surface models and EOs – for instance, when observing the spatial variability of soil moisture (Cornelissen et al., 2014; Bertoldi et al., 2014) or evaporation data (Trambauer et al., 2014). As with the tighter workflow just envisioned, the structure of conceptual hydrological models, which, for instance, could be obtained by producing hybrid PB–ML models (as described in the next section), has to be improved. This, in turn, could result in a rethinking of the processes of calibration and DA, as envisioned by Tsai et al. (2021) and Geer (2021), albeit from different points of view.

In any case, we suggest that the two communities – hydrologists and remote sensing scientists – should have a stronger and closer collaboration, and DARTHS can provide a way to facilitate it. With DARTHS, we should not have hydrologists using satellite data simply as end users, without giving feedback to remote sensing scientists; moreover, remote sensing scientists should take care of the suggestions and criticism made by hydrologists.

3.3 The grand challenge of hybrid models in DARTHS

Thus far, we have been agnostic with respect to the types of models, whether PB, HDS, SM or ML; getting into the details of these, their achievements or limitations, is not the argument of this paper. However, as Nearing et al. (2021) emphasizes, there are some questions that hydrological research is struggling to find answers to. Among these, for instance, is 1 of the 23 questions posed by Blöschl et al. (2019): what are the hydrologic laws at the catchment scale and how do they change with scale? As Nearing et al. (2021) points out, the elusiveness of answers to this question in the last 30 years was not caused by the lack of data nor by the heterogeneity of catchments but more probably by some weaknesses in the physical–mathematical methods available (see also Gharari et al., 2021, for further discussions); it is necessary to work

towards a new mathematical–statistical approach (Ramadhan et al., 2020). DARTHS should be implemented to respond to this request.

The change in paradigm that is now expected will bring the possibility of merging different families of models within MBC informatics as well as the possibility of hybrid modeling solutions (Shen et al., 2018) in which MS, ML, HDS and PB solutions can be mixed. ML techniques are current practice when using PB or HDS models, for instance, in the calibration phases where genetic algorithms have been used since Vrugt et al. (2003) or in particle swarm (Kennedy and Eberhart, 1997), but these techniques are not normally used in core modeling. Conversely, the community is more oriented to assimilate EOs, and it has had an articulated approach to ML (Reichstein et al., 2019). As previously mentioned, there is cross-fertilization between the two communities, but often diversity of objectives and mismatches in spatial and temporal scales (small scales vs. large, global scales; hourly vs. daily or larger timescales) generate misunderstandings and imprecise claims regarding the performance of the respective techniques. These operative and semantic gaps can be filled with the help of DARTHS.

The hydrological community is still learning how to use ML, and there is a lot of room to incorporate statistical knowledge in PB models. As Konapala et al. (2020) writes, ML tools are used together with PB modeling, for instance, for calibration (Krapu et al., 2019), downscaling of hydrologic data (Abbaszadeh et al., 2019), rainfall–runoff modeling (Kratzert et al., 2018), data retrieval from remote sensing data (Karpatne et al., 2016; Ross et al., 2019; Cho et al., 2019) and interpreting the hydrologic process (Jiang and Kumar, 2019; Konapala and Mishra, 2020). Likewise, convolutional neural networks (CNNs) have been used extensively to learn from spatially distributed fields (Kreyenberg et al., 2019; Liu and Wu, 2016; Pan et al., 2019). In most of these trials, PB calibration is performed first and subsequently refined with ML techniques. Obviously, the same workflow is maintained in the production phase. However the granularity of the mixing of models can go deeper: as in Geer (2021), we can use solvers in which some equations are normal partial differential equations (PDEs) or ODEs, and some others, simultaneously solved, are ML-based like long short-term memory (LSTM) models.

However, there are a few technical issues that have to be resolved to satisfy the requirements of a well-engineered DARTH. The solutions of the ML community were generated with a variety of problems in mind, like computer vision applications, speech and natural language processing. As such, the modeling chains were developed somewhat differently from those traditionally used in hydrology and Earth sciences (Geer, 2021). These tools were created under the modeling assumption that ML can provide an all-purpose, non-linear, function-fitting capability that is “a universal approximator”, as said by Hornik (1991). This approach, in fact, is not very different from the idea that PDEs or ODEs

are the only tools required to solve any dynamical problem and that they are, from one perspective, like the OpenFOAM (Jasak et al., 2007) or the FEniCS (Blechta et al., 2015) libraries that provide prepackaged solvers and methods for the most common equation types. These frameworks can be assembled before runtime by means of a domain-specific language (DSL) and save time for researchers, who then act as linkers. For instance, in the case of neural networks, (Mayer and Jacobsen, 2020) ML frameworks supply libraries to define network types like CNNs or recurrent neural networks (RNNs). As with OpenFOAM and FEniCS, they provide common model architectures and interfaces via popular programming languages such as Python, C, C++ and Scala. However, unlike OpenFOAM and FEniCS, a lot of the ML supports special hardware for calculation acceleration, with libraries such as CUDA Deep Neural Network (cuDNN) library, NVIDIA Collective Communications Library (NCCL) and cuBLAS (the GPU implementation of basic linear algebra subprogram, BLAS, libraries) for GPU-accelerated deep learning training. Mayer and Jacobsen (2020) is a nice review of the ML frameworks currently available. Among these, we can mention PyTorch (<https://pytorch.org/>, last access: 23 September 2022), TensorFlow (Abadi et al., 2016), ML4J (<https://github.com/ml4jarchive>, last access: 23 September 2022) and H₂O (Cook, 2016). Moreover, these can use other frameworks, like Hadoop or Spark, to distribute calculations over heterogeneous cloud systems. The HPC computational forms of current PB and ML solutions are also different. Earth science applications tend to use supercomputers, as weather and climate models, or PB models like ParFlow (Kuffour et al., 2020), have a lot of inter-process communication (being grid based) that is optimized on supercomputers. ML approaches typically use cloud computing, taking advantage of algorithms that require less communication along with the compatibility of neural network processing with graphics or tensor processing units. One practical problem, as highlighted in the previous section, is to combine typical MaaA supercomputer-friendly models with ML software packages.

In principle, when using MBC, it would be conceptually easy to say that, in a given modeling solution, some of the components can be PB and some others can be ML based and that the workflow envisioned in the cited papers can, thus, be implemented. For instance, Serafin et al. (2021) have shown that it can be done by adopting an artificial neural network (ANN) model for runoff purposes, in the OMS/CSIP system. In this case, the ANN libraries were integrated into the framework, and this made the integration easier, although not trivial; however, in general, a more loose coupling through “bridges” should be envisioned. This can be done, and one example is the Fortran–Keras Bridge (Ott et al., 2020) that allows a two-way connection of MaaT- and MaaC-type models with the Keras framework. Fortunately, the ML types of infrastructure are often engineered as SOA software; therefore,

it is not overwhelming to obtain the interactivity required by DARTHS within the flexibility offered by MBC.

One further issue is that there is not just one ML framework, but many. In ML, computations are driven by an internal graph structure, (Geer, 2021); however, in some frameworks, this graph is static (e.g., TensorFlow and Caffe2), whereas in others (for instance PyTorch), a dynamic graph is used. Therefore, the choice of the computation model can lead to some differences in programming and runtime. As such, this requires the implementation of a “translator” of the DL structures used in one framework into the others. At least one initiative has been born to accomplish this task, the Open Neural Network Exchange (ONNX) standard (Lin et al., 2019), but it is missing an action, like the one offered by the OGC, to pursue that interoperability that DE and DARTH would require. Ultimately, because both PB and ML models require computations on direct acyclic graphs (DAGs), it is possible to think that new common techniques of calculation can be shared by ML and PB models.

For the emerging world of hydrological modeling to be fully operational within DARTHS three efforts are required:

- building the appropriate bridges;
- working for unified standards of representation of ML processes, at least at the production stage; and
- a possible reengineering of the infrastructures to allow a finer granularity of the hybrid process modeling interactions.

4 Epilogue

How can we organize the building of DARTHS and which messages can we take home?

4.1 On the organization of DARTH communities and their governance

What we have said in the paper should suggest that the goal is not to build a fully-fledged DARTH but, rather, to build DARTH services or components that can be assembled together to build a DARTH solution. Moreover, various DARTHS can share some services and develop others for themselves, such as geospatial data abstraction (GDAL) libraries that are used by most of the open-source GIS systems or the Linear Algebra Package (LAPACK) libraries that are widely present in modeling infrastructures. We can think of DARTH components as libraries, although they are more shareable on the web, and as pieces of software that are self-contained and can possibly work alone. Therefore, the organization necessary for the development, maintenance and hosting of such a system is dependent on which DARTH components we are thinking about. As, for instance, EO data are provided by institutions like the Japan Aerospace Exploration Agency (JAXA) or the Indian Space Research Or-

ganization (ISRO), it will be their natural duty to provide the data in standardized formats and make them available in appropriate ways (i.e., by exposing an API) to be linked into a DARTH solution. Other subjects can obviously provide their own elaborations or formats of the same data and make them available. For instance, the eWaterCycle application (Hut et al., 2021) uses the common format CMOR (Climate Model Output Rewriter), and the ESMValTool (Righi et al., 2020) can be used to make uniform otherwise heterogeneous data. If the core of DARTHS are DARTH components, provided by different institutions or companies, they should eventually be linked together to have a functioning DARTH solution. Deployments like those initiated with eWaterCycle or Delft-FEWS (Werner et al., 2013) and LIS (Peters-Lidard et al., 2007) are examples of integrators of resources; maybe they are not fully compliant with the DARTH architecture, but they can be a good starting point to deploy DARTHS and provide an example of DARTH providers.

Further ideas are expressed in Nativi et al. (2021), who provide a specific section with a clear title: “Effective Governance of the Independent Enterprise Systems of a Digital Ecosystem” The word ecosystem, in fact, is probably the more appropriate in this context: not applications but ecosystems of open applications.

Because DARTHS grow and develop continuously by adapting to the emerging scientific questions, in principle there is no need for a centralized director for most of the developments. However, organized communities can achieve more. In particular, any community will be effective if it chooses to adopt common standards for data and formats, while also promoting innovative solutions. The JULES project (<https://jules.jchmr.org/>, last access: 23 September 2022) is an example of governance for such communities. Examples of successful communities can also be seen in Archfield et al. (2015). An obvious and bigger model to learn from, outside the hydrology community, is provided by the GNU/Linux project where one community is responsible for the kernel of the operating system; others, including GNU, provide additional tools; and, at the end of the process, other subjects deploy variously flavored versions of GNU/Linux, called Linux distributions. The DARTH need not be either commercial or non-commercial: both strategies can be pursued (and, we guess, will be pursued). Please observe, however, that commercial is not the opposite of open source: an open-source product can be commercialized, and a commercial product can be open source. Thus, DARTHS can be both commercial or non-commercial but, either way, would require the appropriate open characteristics.

Building pieces of a DARTHS (i.e., DARTH components), as we see it, is feasible by an organized group of researchers (especially at the linker level) or by partnerships funded by international research programs. The OpenEO initiative (<https://openeo.org>, last access: 23 September 2022) in the field of Earth observation data development is an example of how a DARTH development could be funded. The en-

tire scope of this paper is to suggest a modular architecture with science parts that can, in principle, be affordable to many, if not all, scientists, as it was for JULES (Best et al., 2011), CLM (Lawrence et al., 2019) and other projects. From a lower level, SUMMA (Clark et al., 2015), GE-Oframe (Formetta et al., 2014) and OMS3 are efforts of a single research group that can provide a starting platform for a DARTH. Certainly, producing a DARTH from scratch would be a major effort that only large company associations or governments could afford.

The quality control of the final product is done by, in fact it is, the community that builds a specific part of the DARTHS and the community or the commercial entity that links the various DARTH components to form a DARTH distribution. Certainly, new techniques like the blockchain can be used to certify the steps and the chain of responsibility that a certain DARTH distribution has followed. It is expected that DARTH initiatives will also come up with test cases with specific links to (or support from) important experimental activities, sites and observatories (e.g., <https://www.hymex.fr/liaise/index.html>, last access: 23 September 2022) that link experimental measurements and observations to model development. The same observatories can also be responsible for building shared benchmarking datasets to test DARTH performance. If we refer to studies pursued with DARTHS, the ultimate responsibility is of the runners who perform the studies and, eventually, the journal that publishes them. One critical aspect though, will be the availability of data, simulation records and software to allow the testing of results by other researchers. Whoever uses a DARTH is responsible for its results, just as whoever publishes something is responsible for the published piece. Experience from the SARS-CoV-2 pandemic suggests that the reader must be educated and that keeping information hidden does not solve issues and does not cultivate the credibility of science. Therefore, in our opinion, limitations on the flow of information is a bad policy. On the other hand, the presence of error estimation, the replicability of the experiment and open code are landmarks for the good will of acting fairly.

4.2 Conclusions

In this paper, we have discussed what DARTHS are, and we found that they cannot simply be models as intended in the usual sense. They first need to be supported by an infrastructure that provides the following the following:

- the possibility to use the modeling-by-components (MBC) strategy;
- implicit parallelism for simulation that mixes various types of parallelism;
- HPC to treat the data in input and output;
- loose coupling of models and data;

- HPC to perform calibration and data assimilation (DA) of parameters and retrieved quantities.

Furthermore, we claim that this infrastructure should be able to do the following:

- dispatch data and models around the web for distributed elaboration;
- allow incremental improvements of the core programming features by a community of coders.

Detailed prescriptions were given for these infrastructures to accomplish open-science requests, including being open source and produced by open-source tools, a requirement that is, obviously, an option but that we deem necessary for the progress of science. We also gave indications about deploying the necessary code and computing literacy.

We did not discuss the content nor the structure of the models, but we claimed the following:

- DARTHs should be agnostic with respect to the model choice in order to support the progress of science;
- DARTHs should favor multiple hypothesis testing, which has been fully discussed in many recent contributions (Clark et al., 2011a; Prieto et al., 2021; Fenicia and Kavetski, 2021).

Moreover, to help progress, we categorized the models according to their characteristics with respect to being able to cope with the DARTH metaphor, spanning from MaaA to MaaC.

Those who are interested in model content can easily refer to the various commentaries, reflections and blueprints present in literature (Freeze and Harlan, 1969; Beven, 2002; Lee et al., 2005; Rigon et al., 2006; Montanari and Koutsoyiannis, 2012; Clark et al., 2015; Shen et al., 2018; Savenije and Hrachowitz, 2017), which should be thought of as complementary to the present paper.

We also remarked that DARTH has to encourage a new collaborative approach to EOs for both the hydrology (and Earth science) community and the Earth observations community: an approach in which data fusion and products are made at the coder level and not provided “as is” to researchers. Another characteristic that has been invoked is the presence of readily available error estimates with any DARTH in any forecasting and as part of the knowledge creation.

A DARTH should not be thought of as an immutable piece of software but as a dynamically growing and evolving system in which different model paradigms, data and simulations can be exchanged and/or accessed over the web. Semantic information has to be added to allow searches of the competing tools and to make the discovery of data easier. A DARTH system should be openly managed and contributed to by the good will of researchers working within

shared policies. DARTH design must favor cross-fertilization of knowledge between related sectors of science, technology and management, avoiding so-called silo-type models. The systems’ deployment concepts are, at present, themselves in development, and they are a matter of research, trial and error. Therefore, DARTHs can be designed with the fact in mind that many changes will be possible before getting the desired result.

We did not discuss the most commonly treated area of DE (i.e., the one dealing with visualization of data). It is obviously a fascinating subject, however, our previous experience in building GIS systems has taught us that visualization has to be treated as data and, therefore, loosely coupled to models because it is subject to rapid obsolescence. Many new human–computer interfaces are foreseen in the next few years, and DARTHs have to be prepared to connect to them and exploit their capabilities, including interfaces with the natural languages and body gestures that often appear in sci-fi movies.

Finally, there is an exciting area in HPC research where new methods can be envisioned for processing a chain of models and submodels along DAGs that has just started being explored and that could unify the way that the informatics and numerics of ML and MBC are actually done.

In this paper, we gave a name to the hydrological DTE and expressed optimism that the goal to build a DARTH is a feasible enterprise that can be pursued with success. Many of the aspects touched upon are, in fact, already reality in current state-of-the-art practices and just need to be evolved, systematized, and integrated into a unique framework or made interoperable. This integration is necessary because, as the authors of this work believe, great advancements and innovations in hydrology can only be achieved by treating all of the aspects discussed together.

Appendix A: Table of acronyms

Acronym	Meaning
ANN	Artificial neural network
CEOS	Committee on Earth Observation Satellites
CNN	Convolutional neural network
CSIP	Cloud Services Integration Platform
DA	Data assimilation
DAG	Direct acyclic graph
DARTH	Digital eARth Twin Hydrology
DE	Domain Earth
DSL	Domain-specific language
DT	Digital twin
DTE	Digital twin Earth
EO	Earth observation
EPN	Extending Petri net
GWMS	Global Water Monitoring System
GEOSS	Global Earth Observation System of Systems
GRDC	Global Runoff Data Centre
HDS	Reservoir-type models
IEM	Integrated environmental modeling
IJDE	<i>International Journal of Digital Earth</i>
ISDE	International Society for Digital Earth
MaaA	Model as an Application
MaaC	Model as a Commodity
MaaR	Model as a Resource
MaaS	Model as a Service
MaaT	Model as a Tool
MBC	Modeling by component
ML	Machine learning
OMS	Object Modeling System
PB	Process based
RNN	Recurrent neural network
SDI	Spatial data infrastructure
SM	Statistical model
SOA	Service-oriented architecture

Data availability. No data sets were used in this article.

Supplement. The supplement related to this article is available online at: <https://doi.org/10.5194/hess-26-4773-2022-supplement>.

Author contributions. RR was responsible for conceptualizing the study, developing the methodology, writing the original draft of the paper, and reviewing and editing the paper. GF, MB, NT, CD, OD and CM contributed to writing the original draft of the paper as well as reviewing and editing the paper.

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. We thank the two reviewers, Mark Hrachowitz and Uwe Ehret, and the editor, Erwin Zehe, whose careful review helped to greatly improve the manuscript. English language revision was carried out by Joseph Tomasi.

Financial support. This paper has been partially supported by the MIUR PRIN project (PRIN 2017) "WATER mixing in the critical ZONE: observations and predictions under environmental changes" (WATZON; project code 2017SL7ABC), the CNR "Carbon and water cycles interactions during drought and their impact on WATER and ForEst Resources in the Mediterranean region" (WAFER) project and the ESA Digital Twin Earth Hydrology project (ESA contract no. 4000129870/19/I-NB).

Review statement. This paper was edited by Erwin Zehe and reviewed by Markus Hrachowitz and Uwe Ehret.

References

- Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, s., Irving, G., Isard, M., Kudlur, M., Levenberg, J., Monga, R., Moore, S., Murray, D. G., Steiner, B., Tucker, P., Vasudevan, V., Warden, P., Wicke, M., Yu, Y., and Zheng, X.: Google Brain, A system for large-scale machine learning, in: OSDI'16: Proc. 12th USENIX Symposium on Operating Systems Design and Implementation, 265–283, USENIX Association, 2016
- Abbaszadeh, P., Moradkhani, H., and Daescu, D. N.: The quest for model uncertainty quantification: A hybrid ensemble and variational data assimilation framework, *Water Resour. Res.*, 55, 2407–2431, 2019.

- Abbott, B. W., Bishop, K., Zarnetske, J. P., Minaudo, C., Chapin, F., Krause, S., Hannah, D. M., Conner, L., Ellison, D., Godsey, S. E., Plont, S., Marçais, J., Huebner, A., Frei, R. J., Hampton, T., Gu, S., Buhman, M., Sayedi, S. S., Ursache, O., Chapin, M., Henderson, K. D., and Pinay, G.: Human domination of the global water cycle absent from depictions and perceptions, *Nat. Geosci.*, 12, 533–540, 2019.
- Addor, N. and Melsen, L. A.: Legacy, Rather Than Adequacy, Drives the Selection of Hydrological Models, *Water Resour. Res.*, 55, 378–390, 2019.
- Addor, N., Newman, A. J., Mizukami, N., and Clark, M. P.: The CAMELS data set: catchment attributes and meteorology for large-sample studies, *Hydrol. Earth Syst. Sci.*, 21, 5293–5313, <https://doi.org/10.5194/hess-21-5293-2017>, 2017.
- Archfield, S. A., Clark, M., Arheimer, B., Hay, L. E., McMillan, H., Kiang, J. E., Seibert, J., Hakala, K., Bock, A., Wagener, T., Farmer, W. H., Andréassian, V., Attinger, S., Viglione, A., Knight, R., Markstrom, S., and Over, T.: Accelerating advances in continental domain hydrologic modeling, *Water Resour. Res.*, 51, 10078–10091, 2015.
- Argent, R. M.: An overview of model integration for environmental applications – components, frameworks and semantics, *Environ. Modell. Softw.*, 19, 219–234, 2004.
- Arnold, J. G., Moriasi, D. N., Gassman, P. W., Abbaspour, K. C., White, M. J., Srinivasan, R., Santhi, C., Harmel, R. D., Van Griensven, A., Van Liew, M. W., Kannan, N., and Jha, M. K.: SWAT: Model use, calibration, and validation, *Trans. ASABE*, 55, 1491–1508, 2012.
- Arrieta, A. B., Díaz-Rodríguez, N., Del Ser, J., Bennetot, A., Tabik, S., Barbado, A., García, S., Gil-López, S., Molina, D., Benjamins, R., Chatila, R., and Herrera, F.: Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI, *Inform. Fusion*, 58, 82–115, 2020.
- Babaeian, E., Sadeghi, M., Jones, S. B., Montzka, C., Vereecken, H., and Tuller, M.: Ground, proximal, and satellite remote sensing of soil moisture, *Rev. Geophys.*, 57, 530–616, 2019.
- Ballatore, A.: The myth of the Digital Earth between fragmentation and wholeness, *Wi, J. Mobile Media*, 8, <https://doi.org/10.48550/arXiv.1412.2078>, 2014.
- Bancheri, M., Rigon, R., and Manfreda, S.: The GEOframe-NewAge Modelling System Applied in a Data Scarce Environment, *Water*, 12, 86, <https://doi.org/10.3390/w12010086>, 2019a.
- Bancheri, M., Serafin, F., and Rigon, R.: The Representation of Hydrological Dynamical Systems Using Extended Petri Nets (EPN), *Water Resour. Res.*, 55, 8895–8921, 2019b.
- Beck, H. E., van Dijk, A. I. J. M., de Roo, A., Dutra, E., Fink, G., Orth, R., and Schellekens, J.: Global evaluation of runoff from 10 state-of-the-art hydrological models, *Hydrol. Earth Syst. Sci.*, 21, 2881–2903, <https://doi.org/10.5194/hess-21-2881-2017>, 2017.
- Berti, G.: Generic software components for Scientific Computing, Ph.D. thesis, Brandenburgischen Technischen Universität Cottbus, https://www.researchgate.net/profile/Guntram-Berti/publication/239065936_Generic_software_components_for_Scientific_Computing/links/545fd2180cf295b56161c9b0/Generic-software-components-for-Scientific-Computing.pdf (last access: 23 September 2022), 2000.
- Bertoldi, G., Della Chiesa, S., Notarnicola, C., Pasolli, L., Niedrist, G., and Tappeiner, U.: Estimation of soil moisture patterns in mountain grasslands by means of SAR RADARSAT2 images and hydrological modeling, *J. Hydrol.*, 516, 245–257, 2014.
- Best, M. J., Pryor, M., Clark, D. B., Rooney, G. G., Essery, R. L. H., Ménard, C. B., Edwards, J. M., Hendry, M. A., Porson, A., Gedney, N., Mercado, L. M., Sitch, S., Blyth, E., Boucher, O., Cox, P. M., Grimmond, C. S. B., and Harding, R. J.: The Joint UK Land Environment Simulator (JULES), model description – Part 1: Energy and water fluxes, *Geosci. Model Dev.*, 4, 677–699, <https://doi.org/10.5194/gmd-4-677-2011>, 2011.
- Beven, K.: Towards an alternative blueprint for a physically based digitally simulated hydrologic response modelling system, *Hydrol. Process.*, 16, 189–206, 2002.
- Beven, K.: Environmental modelling: an uncertain future?, CRC press, 328 pp., <https://doi.org/10.1201/9781482288575>, 2018.
- Beven, K.: Towards a methodology for testing models as hypotheses in the inexact sciences, *Proc. Math. Phys. Eng. Sci.*, 475, 20180862, <https://doi.org/10.1098/rspa.2018.0862>, 2019.
- Beven, K. J.: *Rainfall-Runoff Modelling: The Primer*, The Primer, John Wiley & Sons, ISBN 9780470714591, 488 pp., 2012.
- Binley, A. M., Beven, K. J., Calver, A., and Watts, L.: Changing responses in hydrology: assessing the uncertainty in physically based model predictions, *Water Resour. Res.*, 27, 1253–1261, 1991.
- Blechta, J., Hake, J., Johansson, A., and others: The FEn-iCS project version 1.5, *Arch. Num. Softw.*, 51, 3, 9–23, <https://doi.org/10.11588/ans.2015.100.20553>, 2015.
- Blöschl, G.: Debates-Hypothesis testing in hydrology: Introduction, *Water Resour. Res.*, 53, 1767–1769, 2017.
- Blöschl, G., Bierkens, M. F., Chambel, A. et al.: Twenty-three unsolved problems in hydrology (UPH) – a community perspective, *Hydrol. Sci. J.*, 64, 1141–1158, 2019.
- Boldrini, E., Mazzetti, P., Nativi, S., Santoro, M., Papeschi, F., Roncella, R., Olivieri, M., Bordini, F., and Pecora, S.: WMO Hydrological Observing System (WHOS) broker: implementation progress and outcomes, in: European Geoscience Union General Assembly, p. 14755, Copernicus, 22nd EGU General Assembly, 4–8 May 2020, <https://doi.org/10.5194/egusphere-egu2020-14755>, 2020.
- Bottazzi, M., Bancheri, M., Mobilia, M., Bertoldi, G., Longobardi, A., and Rigon, R.: Comparing Evapotranspiration Estimates from the GEOframe-Prospero Model with Penman–Monteith and Priestley–Taylor Approaches under Different Climate Conditions, *Water*, 13, 1221, <https://doi.org/10.3390/w13091221>, 2021.
- Brocca, L., Ciabatta, L., Massari, C., Moramarco, T., Hahn, S., Hasenauer, S., Kidd, R., Dorigo, W., Wagner, W., and Levizzani, V.: Soil as a natural rain gauge: Estimating global rainfall from satellite soil moisture data, *J. Geophys. Res.-Atmos.*, 119, 5128–5141, 2014.
- Brunner, P. and Simmons, C. T.: HydroGeoSphere: A fully integrated, physically based hydrological model, *Ground Water*, 50, 170–176, 2012.
- Busti, R.: The implementation and testing of different modeling solutions to estimate water balance in mountain regions, Master's thesis, University of Trento, 2021.
- CEOS: Committee on Earth Observation Satellites (Ceos), <http://database.eohandbook.com/> (last access: 23 September 2022), 2019.

- Chen, M., Voinov, A., Ames, D. P., Kettner, A. J., Goodall, J. L., Jakeman, A. J., Barton, M. C., Harpham, Q., Cuddy, S. M., DeLuca, C., Yue, S., Wang, J., Zhang, F., Wen, Y., and Lü, G.: Position paper: Open web-distributed integrated geographic modelling and simulation to enable broader participation and applications, *Earth Sci. Rev.*, 207, 103223, <https://doi.org/10.1016/j.earscirev.2020.103223>, 2020.
- Chew, C. and Small, E.: Soil moisture sensing using spaceborne GNSS reflections: Comparison of CYGNSS reflectivity to SMAP soil moisture, *Geophys. Res. Lett.*, 45, 4049–4057, 2018.
- Cho, E., Jacobs, J. M., Jia, X., and Kraatz, S.: Identifying subsurface drainage using satellite big data and machine learning via Google earth engine, *Water Resour. Res.*, 55, 8028–8045, 2019.
- Chu X. and Steinman A.: Event and Continuous Hydrologic Modeling with HEC-HMS, *J. Irrig. Drain. Eng.*, 135, 119–124, 2009.
- Clark, M. P., Kavetski, D., and Fenicia, F.: Pursuing the method of multiple working hypotheses for hydrological modeling, *Water Resour. Res.*, 47, <https://doi.org/10.1029/2010WR009827>, 2011a.
- Clark, M. P., Kavetski, D., and Fenicia, F.: Pursuing the Method of Multiple Working Hypotheses for Hydrological Modeling, *Water Resour. Res.*, 47, <https://doi.org/10.1029/2010wr009827>, 2011b.
- Clark, M. P., Nijssen, B., Lundquist, J. D., Kavetski, D., Rupp, D. E., Woods, R. A., Freer, J. E., Gutmann, E. D., Wood, A. W., Brekke, L. D., Arnold, J. R., Gochis, D. J., and Rasmussen, R. M.: A unified approach for process-based hydrologic modeling: 1. Modeling concept, *Water Resour. Res.*, 51, 2498–2514, 2015.
- 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, *Int. J. High Perform. Comput. Appl.*, 19, 341–350, 2005.
- Cook, D.: Practical Machine Learning with H2O: Powerful, Scalable Techniques for Deep Learning and AI, “O’Reilly Media, Inc.”, ISBN 9781491964576, 300 p., 2016.
- Cornelissen, T., Diekkrüger, B., and Bogena, H. R.: Significance of scale and lower boundary condition in the 3D simulation of hydrological processes and soil moisture variability in a forested headwater catchment, *J. Hydrol.*, 516, 140–153, 2014.
- Cox, R. T.: Probability, Frequency and Reasonable Expectation, *Am. J. Phys.*, 14, 1–13, 1946.
- Craglia, M., de Bie, K., Jackson, D., Pesaresi, M., Remetey-Fülöpp, G., Wang, C., Annoni, A., Bian, L., Campbell, F., Ehlers, M., van Genderen, J., Goodchild, M., Guo, H., Lewis, A., Simpson, R., Skidmore, A., and Woodgate, P.: Digital Earth 2020: towards the vision for the next decade, *Int. J. Digital Earth*, 5, 4–21, <https://doi.org/10.1080/17538947.2011.638500>, 2012.
- Craig, J. R., Brown, G., Chlumsky, R., Jenkinson, R. W., Jost, G., Lee, K., Mai, J., Serrer, M., Sgro, N., Shafii, M., Snowdon, A. P., and Tolson, B. A.: Flexible watershed simulation with the Raven hydrological modelling framework, *Environ. Modell. Softw.*, 129, 104728, <https://doi.org/10.1016/j.envsoft.2020.104728>, 2020.
- Dal Molin, M., Kavetski, D., and Fenicia, F.: SuperflexPy 1.3.0: an open-source Python framework for building, testing, and improving conceptual hydrological models, *Geosci. Model Dev.*, 14, 7047–7072, <https://doi.org/10.5194/gmd-14-7047-2021>, 2021.
- David, O., Ascough, II, J. C., Lloyd, W., Green, T. R., Rojas, K. W., Leavesley, G. H., and Ahuja, L. R.: A software engineering perspective on environmental modeling framework design: The Object Modeling System, *Environ. Modell. Softw.*, 39, 201–213, 2013.
- David, O., Lloyd, W., Rojas, K., Arabi, M., Geter, F., Ascough, J., Green, T., Leavesley, G., and Carlson, J.: Modeling-as-a-Service (MaaS) using the Cloud Services Innovation Platform (CSIP), in: International Congress on Environmental Modelling and Software, scholarsarchive.byu.edu, 13, https://digitalcommons.tacoma.uw.edu/tech_pub/13 (last access: 23 September 2022), 2014.
- Davidson, M., Chini, M., Dierking, W., Djavidnia, S., Haarpaintner, J., Hajdich, G., Laurin, G., Lavalley, M., López-Martínez, C., Nagler, T., Pierdicca, N., and Su, B.: Copernicus L-band SAR Mission Requirements Document, https://esamultimedia.esa.int/docs/EarthObservation/Copernicus_L-band_SAR_mission_ROSE-L_MRD_v2.0_issued.pdf (last access: 23 September 2022), 2019.
- De Lannoy, G. J. and Reichle, R. H.: Global assimilation of multi-angle and multipolarization SMOS brightness temperature observations into the GEOS-5 catchment land surface model for soil moisture estimation, *J. Hydrometeorol.*, 17, 669–691, 2016.
- Dey, C., Sanders, C., Clochard, J., and Hennessy, J.: Guide to the WMO table driven code form used for the representation and exchange of regularly spaced data in binary form: FM 92 GRIB, Tech. rep., WMO Tech. Rep., 98 pp., <http://www.wmo.int/pages/prog/www/WMOCodes/Guides/GRIB/GRIB1-Contents.html> (last access: 23 September 2022), 2007.
- Döll, P., Kaspar, F., and Lehner, B.: A global hydrological model for deriving water availability indicators: model tuning and validation, *J. Hydrol.*, 270, 105–134, 2003.
- Duan, Q., Gupta, H. V., Sorooshian, S., Rousseau, A. N., and Turcotte, R.: Calibration of Watershed Models, John Wiley & Sons, edited by: Duan, Q. et al., 653 pp., AGU Washington D. C., 2003.
- Eagleson, P. S.: The emergence of global-scale hydrology, *Water Resour. Res.*, 22, 6S–14S, 1986.
- Emerton, R. E., Stephens, E. M., Pappenberger, F., Pagano, T. C., Weerts, A. H., Wood, A. W., Salamon, P., Brown, J. D., Hjerdt, N., Donnelly, C., Baugh, C. A., and Cloke, H. L.: Continental and global scale flood forecasting systems, *WIREs Water*, 3, 391–418, 2016.
- Endrizzi, S., Gruber, S., Dall’Amico, M., and Rigon, R.: GEOTop 2.0: simulating the combined energy and water balance at and below the land surface accounting for soil freezing, snow cover and terrain effects, *Geosci. Model Dev.*, 7, 2831–2857, <https://doi.org/10.5194/gmd-7-2831-2014>, 2014.
- Endrizzi, S., Gruber, S., Dall’Amico, M., and Rigon, R.: GEOTop 2.0: simulating the combined energy and water balance at and below the land surface accounting for soil freezing, snow cover and terrain effects, *Geosci. Model Dev.*, 7, 2831–2857, <https://doi.org/10.5194/gmd-7-2831-2014>, 2014.
- Faticchi, S., Vivoni, E. R., Ogden, F. L., Ivanov, V. Y., Mirus, B., Gochis, D., Downer, C. W., Camporese, M., Davison, J. H., Ebel, B., Jones, N., Kim, J., Mascaro, G., Niswonger, R., Restrepo, P., Rigon, R., Shen, C., Sulis, M., and Tarboton, D.: An overview of current applications, challenges, and future trends in

- distributed process-based models in hydrology, *J. Hydrol.*, 537, 45–60, 2016a.
- Fatichi, S., Vivoni, E. R., Ogden, F. L., Ivanov, V. Y., Mirus, B., Gochis, D., Downer, C. W., Camporese, M., Davison, J. H., Ebel, B., Jones, N., Kim, J., Mascaro, G., Niswonger, R., Restrepo, P., Rigon, R., Shen, C., Sulis, M., and Tarbotons, D.: An overview of current applications, challenges, and future trends in distributed process-based models in hydrology, *J. Hydrol.*, 537, 45–60, 2016b.
- Fenicia, F. and Kavetski, D.: Behind every robust result is a robust method: Perspectives from a case study and publication process in hydrological modelling, *Hydrol. Process.*, 35, 45–60, 2021.
- Folk, M., Heber, G., Koziol, Q., Pourmal, E., and Robinson, D.: An overview of the HDF5 technology suite and its applications, in: Proceedings of the EDBT/ICDT 2011 Workshop on Array Databases, AD '11, 36–47, Association for Computing Machinery, New York, NY, USA, 36–47, <https://doi.org/10.1145/1966895.1966900>, 2011.
- Formetta, G., Antonello, A., Franceschi, S., David, O., and Rigon, R.: Hydrological modelling with components: A GIS-based open-source framework, *Environ. Modell. Softw.*, 55, 190–200, 2014.
- Freeze, R. A. and Harlan, R. L.: Blueprint for a physically-based, digitally-simulated hydrologic response model, *J. Hydrol.*, 9, 237–258, [https://doi.org/10.1016/0022-1694\(69\)90020-1](https://doi.org/10.1016/0022-1694(69)90020-1), 1969.
- Gamma, E., Helm, R., Johnson, R., Johnson, R. E., and Vlissides, J.: Design Patterns: Elements of Reusable Object-Oriented Software, Pearson Deutschland GmbH, ISBN 9783827330437, 395 pp., 1995.
- Gardner, H. and Manduchi, G.: Design Patterns for e-Science, Springer Science & Business Media, ISBN 9783540680888, 388 pp., 2007.
- Geer, A. J.: Learning earth system models from observations: machine learning or data assimilation?, *Philos. Trans. Roy. Soc. A*, 379, 20200089, <https://doi.org/10.1098/rsta.2020.0089>, 2021.
- Gharari, S., Gupta, H. V., Clark, M. P., Hrachowitz, M., Fenicia, F., Matgen, P., and Savenije, H. H. G.: Understanding the information content in the hierarchy of model development decisions: Learning from data, *Water Resour. Res.*, 57, <https://doi.org/10.1029/2020wr027948>, 2021.
- Gironás, J., Roesner, L. A., Rossman, L. A., and Davis, J.: A new applications manual for the Storm Water Management Model (SWMM), *Environ. Modell. Softw.*, 25, 813–814, 2010.
- Goodchild, M. F., Guo, H., Annoni, A., Bian, L., De Bie, K., Campbell, F., Craglia, M., Ehlers, M., Van Genderen, J., Jackson, D., Lewis, A. J., Pesaresi, M., Remetej-Fülöpp, G., Simpson, R., Skidmore, A., Wang, C., and Woodgate, P.: Next-generation digital earth, *P. Natl. Acad. Sci. USA*, 109, 11088–11094, 2012.
- Göttinger, J. and Bárdossy, A.: Generic error model for calibration and uncertainty estimation of hydrological models, *Water Resour. Res.*, 44, W00B07, <https://doi.org/10.1029/2007wr006691>, 2008.
- Graessler, I. and Pöhler, A.: Integration of a digital twin as human representation in a scheduling procedure of a cyber-physical production system, in: 2017 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM), 289–293, <https://doi.org/10.1109/IEEM.2017.8289898>, 2017.
- Gregersen, J. B., Gijssbers, P. J. A., and Westen, S. J. P.: OpenMI: Open modelling interface, *J. Hydroinform.*, 9, 175–191, 2007.
- Guo, H., Goodchild, M. F., and Annoni, A.: Manual of Digital Earth, Springer Singapore, ISBN 9789813299146, 852 pp., 2019.
- Gupta, H. V., Sorooshian, S., and Yapo, P. O.: Status of automatic calibration for hydrologic models: Comparison with multilevel expert calibration, *J. Hydrol. Eng.*, 4, 135–143, 1999.
- Hall, C. A., Saia, S. M., Popp, A. L., Dogulu, N., Schymanski, S. J., Drost, N., van Emmerik, T., and Hut, R.: A hydrologist's guide to open science, *Hydrol. Earth Syst. Sci.*, 26, 647–664, <https://doi.org/10.5194/hess-26-647-2022>, 2022.
- Hill, M. C. and Tiedeman, C. R.: Effective Groundwater Model Calibration with Analysis of Data, Sensitivities, and Uncertainty, Hoboken, New Jersey, John Wiley and Sons, ISBN 9780471776369, 455 pp., 2007.
- Holling, C. S.: Adaptive Environmental Assessment and Management. John Wiley & Sons. <http://pure.iiasa.ac.at/id/eprint/823/> (last access: 27 September 2022), ISBN 0471996327, 402 pp., 1978.
- Hornik, K.: Approximation capabilities of multilayer feedforward networks, *Neural Netw.*, 4, 251–257, 1991.
- Hrachowitz, M. and Clark, M. P.: HESS Opinions: The complementary merits of competing modelling philosophies in hydrology, *Hydrol. Earth Syst. Sci.*, 21, 3953–3973, <https://doi.org/10.5194/hess-21-3953-2017>, 2017.
- Hut, R., Drost, N., van de Giesen, N., van Werkhoven, B., Abdollahi, B., Aerts, J., Albers, T., Alidoost, F., Andela, B., Camphuysen, J., Dzigan, Y., van Haren, R., Hutton, E., Kalverla, P., van Meersbergen, M., van den Oord, G., Pelulessy, I., Smeets, S., Verhoeven, S., de Vos, M., and Weel, B.: The eWaterCycle platform for open and FAIR hydrological collaboration, *Geosci. Model Dev.*, 15, 5371–5390, <https://doi.org/10.5194/gmd-15-5371-2022>, 2022.
- Jasak, H., Jemcov, A., and Tukovic, Z.: OpenFOAM: A C++ library for complex physics simulations, *Coupled Method. Numer. Dynam.*, Dubrovnik, Croatia, <http://csabai.web.elte.hu/http/simulationLab/jasakEtAlOpenFoam.pdf> (last access: 27 September 2022), 2007.
- Jiang, P. and Kumar, P.: Using information flow for whole system understanding from component dynamics, *Water Resour. Res.*, 55, 8305–8329, 2019.
- Jiang, P., Elag, M., Kumar, P., Peckham, S. D., Marini, L., and Rui, L.: A service-oriented architecture for coupling web service models using the Basic Model Interface (BMI), *Environ. Modell. Softw.*, 92, 107–118, 2017.
- Kampf, S. K. and Burges, S. J.: A framework for classifying and comparing distributed hillslope and catchment hydrologic models, *Water Resour. Res.*, 43, 1–24, 2007.
- Karpatne, A., Jiang, Z., Vatsavai, R. R., Shekhar, S., and Kumar, V.: Monitoring land-cover changes: A machine-learning perspective, *IEEE Geosci. Remote Sens. Magaz.*, 4, 8–21, 2016.
- Kelly, D. and Sanders, R.: The challenge of testing scientific software, in: Proceedings of the 3rd annual conference of the Association for Software Testing (CAST 2008: Beyond the Boundaries), 30–36, Citeseer, <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.464.7432&rep=rep1&type=pdf> (last access: 27 September 2022), 2008.
- Kennedy, J. and Eberhart, R. C.: A discrete binary version of the particle swarm algorithm, in: 1997 IEEE International Confer-

- ence on Systems, Man, and Cybernetics. *Computational Cybernetics and Simulation*, 4104–4108, Vol. 5, 1997.
- Knoben, W. J. M., Freer, J. E., Fowler, K. J. A., Peel, M. C., and Woods, R. A.: Modular Assessment of Rainfall–Runoff Models Toolbox (MARRMoT) v1.2: an open-source, extendable framework providing implementations of 46 conceptual hydrologic models as continuous state-space formulations, *Geosci. Model Dev.*, 12, 2463–2480, <https://doi.org/10.5194/gmd-12-2463-2019>, 2019.
- Knoben, W. J. M., Clark, M. P., Bales, J., Bennett, A., Gharari, S., Marsh, C. B., Nijssen, B., Pietroniro, A., Spiteri, R. J., Tarboton, D. G., and Wood, A. W.: Community Workflows to Advance Reproducibility in Hydrologic Modeling: Separating model-agnostic and model-specific configuration steps in applications of large-domain hydrologic models, <https://doi.org/10.1002/essoar.10509195.1>, 2021.
- Knuth, D. E.: *Literate Programming*, *Comput. J.*, 27, 97–111, 1984.
- Knuth, D. E. and Levy, S.: The WEB system of structured documentation, Tech. Rep. STAN-CS-83-980, Stanford University, <http://i.stanford.edu/pub/cstr/reports/cs/tr/83/980/CS-TR-83-980.pdf> (last access: 27 September 2022), 210 pp., 1983.
- Konapala, G. and Mishra, A.: Quantifying climate and catchment control on hydrological drought in the continental United States, *Water Resour. Res.*, 56, e2018WR024620, <https://doi.org/10.1029/2018wr024620>, 2020.
- Konapala, G., Kao, S.-C., Painter, S. L., and Lu, D.: Machine learning assisted hybrid models can improve streamflow simulation in diverse catchments across the conterminous US, *Environ. Res. Lett.*, 15, 104022, <https://doi.org/10.1088/1748-9326/aba927>, 2020.
- Kramer, D.: API documentation from source code comments: a case study of Javadoc, in: *Proceedings of the 17th annual international conference on Computer documentation, SIGDOC '99*, 147–153, Association for Computing Machinery, New York, NY, USA, 1999.
- Krapu, C., Borsuk, M., and Kumar, M.: Gradient-based inverse estimation for a rainfall-runoff model, *Water Resour. Res.*, 55, 6625–6639, 2019.
- Kratzert, F., Klotz, D., Brenner, C., Schulz, K., and Herrnegger, M.: Rainfall–runoff modelling using Long Short-Term Memory (LSTM) networks, *Hydrol. Earth Syst. Sci.*, 22, 6005–6022, <https://doi.org/10.5194/hess-22-6005-2018>, 2018.
- Kreyenberg, P. J., Bauser, H. H., and Roth, K.: Velocity field estimation on density-driven solute transport with a convolutional neural network, *Water Resour. Res.*, 55, 7275–7293, 2019.
- Kuffour, B. N. O., Engdahl, N. B., Woodward, C. S., Condon, L. E., Kollet, S., and Maxwell, R. M.: Simulating coupled surface–subsurface flows with ParFlow v3.5.0: capabilities, applications, and ongoing development of an open-source, massively parallel, integrated hydrologic model, *Geosci. Model Dev.*, 13, 1373–1397, <https://doi.org/10.5194/gmd-13-1373-2020>, 2020.
- Kumar, S. V., Peters-Lidard, C. D., Santanello, J. A., Reichle, R. H., Draper, C. S., Koster, R. D., Nearing, G., and Jasinski, M. F.: Evaluating the utility of satellite soil moisture retrievals over irrigated areas and the ability of land data assimilation methods to correct for unmodeled processes, *Hydrol. Earth Syst. Sci.*, 19, 4463–4478, <https://doi.org/10.5194/hess-19-4463-2015>, 2015.
- Kumar, S. V., Peters-Lidard, C. D., Tian, Y., Houser, P. R., Geiger, J., Olden, S., Lighty, L., Eastman, J. L., Doty, B., Dirmeyer, P., Adams, J., Mitchell, K., Wood, E. F., and Sheffield, J.: Land information system: An interoperable framework for high resolution land surface modeling, *Environ. Modell. Softw.*, 21, 1402–1415, 2006.
- Lagouarde, J.-P., Bhattacharya, B., Crébassol, P., Gamet, P., Adlakha, D., Murthy, C., Singh, S., Mishra, M., Nigam, R., Raju, P. V., Babu, S. S., Shukla, M. V., Pandya, M. R., Boulet, G., Briottet, X., Dadou, I., Dedieu, G., Gouhier, M., Hagolle, O., Irvine, M., Jacob, F., Kumar, K. K., Laignel, B., Maisongrande, P., Mallick, K., Olioso, A., Otlé, C., Roujean, J.-L., Sobrino, J., Ramakrishnan, R., Sekhar, M., and Sarkar, S. S.: Indo-French high-resolution thermal infrared space mission for earth natural resources assessment and monitoring-concept and definition of trishna, in: *ISPRS-GEOGLAM-ISRS Joint International Workshop on “Earth Observations for Agricultural Monitoring”*, Vol. 42, 403–407, <https://doi.org/10.5194/isprs-archives-XLII-3-W6-403-2019>, New Delhi, India, 2019.
- Lau, S., Drosos, I., Markel, J. M., and Guo, P. J.: The Design Space of Computational Notebooks: An Analysis of 60 Systems in Academia and Industry, in: *2020 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*, pp. 1–11, Dunedin, New Zealand, <https://doi.org/10.1109/VL/HCC50065.2020>, 2020.
- Lawrence, D. M., Fisher, R. A., Koven, C. D., Oleson, K. W., Swenson, S. C., Bonan, G., Collier, N., Ghimire, B., Kampenhout, L., Kennedy, D., Kluzek, E., Lawrence, P. J., Li, F., Li, H., Lombardozzi, D., Riley, W. J., Sacks, W. J., Shi, M., Vertenstein, M., Wieder, W. R., Xu, C., Ali, A. A., Badger, A. M., Bisht, G., Broeke, M., Brunke, M. A., Burns, S. P., Buzan, J., Clark, M., Craig, A., Dahlin, K., Drewniak, B., Fisher, J. B., Flanner, M., Fox, A. M., Gentine, P., Hoffman, F., Keppel-Aleks, G., Knox, R., Kumar, S., Lenaerts, J., Leung, L. R., Lipscomb, W. H., Lu, Y., Pandey, A., Pelletier, J. D., Perket, J., Randerson, J. T., Ricciuto, D. M., Sanderson, B. M., Slater, A., Subin, Z. M., Tang, J., Thomas, R. Q., Val Martin, M., and Zeng, X.: The community land model version 5: Description of new features, benchmarking, and impact of forcing uncertainty, *J. Adv. Model. Earth Syst.*, 11, 4245–4287, 2019.
- Lee, H., Sivapalan, M., and Zehe, E.: Representative elementary watershed (REW) approach: a new blueprint for distributed hydrological modelling at the catchment scale, *IAHS Publ.*, ISSN 0144-7815, 195 pp., 2005.
- Lehmann, A., Giuliani, G., Ray, N., Rahman, K., Abbaspour, K. C., Nativi, S., Craglia, M., Cripe, D., Quevauviller, P., and Beniston, M.: Reviewing innovative Earth observation solutions for filling science-policy gaps in hydrology, *J. Hydrol.*, 518, 267–277, 2014.
- Leisch, F.: Sweave: Dynamic Generation of Statistical Reports Using Literate Data Analysis, in: *Compstat*, pp. 575–580, Physica-Verlag HD, 575–580, https://doi.org/10.1007/978-3-642-57489-4_89, 2002.
- Lentner, G.: Shared Memory High Throughput Computing with Apache Arrow™, in: *Proceedings of the Practice and Experience in Advanced Research Computing on Rise of the Machines (learning)*, no. Article 119 in *PEARC '19*, pp. 1–2, Association for Computing Machinery, New York, NY, USA, ISBN 9781450372275, 1–2 pp., 2019.

- Lettenmaier, D. P., Alsdorf, D., Dozier, J., Huffman, G. J., Pan, M., and Wood, E. F.: Inroads of remote sensing into hydrologic science during the WRR era, *Water Resour. Res.*, 51, 7309–7342, 2015.
- Levia, D. F., Carlyle-Moses, D. E., Michalzik, B., Nanko, K., and Tischer, A.: *Forest-water interactions*, Springer, 625 p., ISBN 978-3-030-26085-9, <https://doi.org/10.1007/978-3-030-26086-6>, 2020.
- Lievens, H., Martens, B., Verhoest, N., Hahn, S., Reichle, R., and Miralles, D. G.: Assimilation of global radar backscatter and radiometer brightness temperature observations to improve soil moisture and land evaporation estimates, *Remote Sens. Environ.*, 189, 194–210, 2017.
- Lin, W.-F., Tsai, D.-Y., Tang, L., Hsieh, C.-T., Chou, C.-Y., Chang, P.-H., and Hsu, L.: ONNC: A Compilation Framework Connecting ONNX to Proprietary Deep Learning Accelerators, in: 2019 IEEE International Conference on Artificial Intelligence Circuits and Systems (AICAS), 214–218, <https://doi.org/10.1109/AICAS.2019.8771510>, 2019.
- Liu, Y. and Wu, L.: Geological Disaster Recognition on Optical Remote Sensing Images Using Deep Learning, *Proc. Comput. Sci.*, 91, 566–575, <https://doi.org/10.1016/j.procs.2016.07.144>, 2016.
- Liu, Y., Weerts, A. H., Clark, M., Hendricks Franssen, H.-J., Kumar, S., Moradkhani, H., Seo, D.-J., Schwanenberg, D., Smith, P., van Dijk, A. I. J. M., van Velzen, N., He, M., Lee, H., Noh, S. J., Rakovec, O., and Restrepo, P.: Advancing data assimilation in operational hydrologic forecasting: progresses, challenges, and emerging opportunities, *Hydrol. Earth Syst. Sci.*, 16, 3863–3887, <https://doi.org/10.5194/hess-16-3863-2012>, 2012.
- Lloyd, W., David, O., Ascough, J. C., Rojas, K. W., Carlson, J. R., Leavesley, G. H., Krause, P., Green, T. R., and Ahuja, L. R.: Environmental modeling framework invasiveness: Analysis and implications, *Environ. Modell. Softw.*, 26, 1240–1250, 2011.
- Loizides, F. and Schmidt, B.: Positioning and Power in Academic Publishing: Players, Agents and Agendas: Proceedings of the 20th International Conference on Electronic Publishing, IOS Press, 164 pp., 9781614996491, 2016.
- López López, P., Sutanudjaja, E. H., Schellekens, J., Sterk, G., and Bierkens, M. F. P.: Calibration of a large-scale hydrological model using satellite-based soil moisture and evapotranspiration products, *Hydrol. Earth Syst. Sci.*, 21, 3125–3144, <https://doi.org/10.5194/hess-21-3125-2017>, 2017.
- Manfreda, S., Brocca, L., Moramarco, T., Melone, F., and Sheffield, J.: A physically based approach for the estimation of root-zone soil moisture from surface measurements, *Hydrol. Earth Syst. Sci.*, 18, 1199–1212, <https://doi.org/10.5194/hess-18-1199-2014>, 2014.
- Martens, B., Miralles, D. G., Lievens, H., van der Schalie, R., de Jeu, R. A. M., Fernández-Prieto, D., Beck, H. E., Dorigo, W. A., and Verhoest, N. E. C.: GLEAM v3: satellite-based land evaporation and root-zone soil moisture, *Geosci. Model Dev.*, 10, 1903–1925, <https://doi.org/10.5194/gmd-10-1903-2017>, 2017.
- Martin, R. C.: *Clean Code: A Handbook of Agile Software Craftsmanship*, Prentice Hall, ISBN 9780132350884, 431 pp., 2009.
- Mayer, R. and Jacobsen, H.-A.: *Scalable Deep Learning on Distributed Infrastructures: Challenges, Techniques, and Tools*, *ACM Comput. Surv.*, 53, 1–37, 2020.
- Mayer-Schönberger, V. and Cukier, K.: *Big Data: A Revolution that Will Transform how We Live, Work, and Think*, Houghton Mifflin Harcourt, 2013.
- McCabe, M. F., Aragon, B., Houborg, R., and Mascaro, J.: CubeSats in hydrology: Ultrahigh-resolution insights into vegetation dynamics and terrestrial evaporation, *Water Resour. Res.*, 53, 10017–10024, 2017a.
- McCabe, M. F., Rodell, M., Alsdorf, D. E., Miralles, D. G., Uijlenhoet, R., Wagner, W., Lucieer, A., Houborg, R., Verhoest, N. E. C., Franz, T. E., Shi, J., Gao, H., and Wood, E. F.: The future of Earth observation in hydrology, *Hydrol. Earth Syst. Sci.*, 21, 3879–3914, <https://doi.org/10.5194/hess-21-3879-2017>, 2017b.
- McCool, M., Robison, A., and Reinders, J.: *Structured parallel programming: patterns for efficient computation*, Elsevier, ISBN 9780124159938, 432 pp., 2012.
- McCuen, R. H.: *Modeling hydrologic change: statistical methods*, CRC press, ISBN 9781566706001, 456 pp., 2016.
- Meyer, T., Jagdhuber, T., Piles, M., Fink, A., Grant, J., Vereecken, H., and Jonard, F.: Estimating gravimetric water content of a winter wheat field from L-band vegetation optical depth, *Remote Sens.*, 11, 2353, <https://doi.org/10.3390/rs11202353>, 2019.
- Millman, K. J. and Pérez, F.: Developing Open-Source Scientific Practice *, in: *Implementing Reproducible Research*, 149–83, Chapman and Hall/CRC, <https://www.jarrodmillman.com/publications/millman2014developing.pdf> (last access: 27 September 2022), 2018.
- Modanesi, S., Massari, C., Gruber, A., Lievens, H., Tarpanelli, A., Morbidelli, R., and De Lannoy, G. J. M.: Optimizing a backscatter forward operator using Sentinel-1 data over irrigated land, *Hydrol. Earth Syst. Sci.*, 25, 6283–6307, <https://doi.org/10.5194/hess-25-6283-2021>, 2021.
- Montanari, A. and Koutsoyiannis, D.: A blueprint for process-based modeling of uncertain hydrological systems, *Water Resour. Res.*, 48, <https://doi.org/10.1029/2011wr011412>, 2012.
- Moore, R. V. and Hughes, A. G.: *Integrated environmental modelling: achieving the vision*, Geological Society, London, Special Publications, 408, 17–34, 2017.
- Müller-Hansen, F., Schlüter, M., Mäs, M., Donges, J. F., Kolb, J. J., Thonicke, K., and Heitzig, J.: Towards representing human behavior and decision making in Earth system models – an overview of techniques and approaches, *Earth Syst. Dynam.*, 8, 977–1007, <https://doi.org/10.5194/esd-8-977-2017>, 2017.
- NASA-ISRO, S.: *Mission Science Users' Handbook*, Jet Propulsion Lab., California Inst. Technol., Pasadena, CA, USA, https://nisar.jpl.nasa.gov/system/documents/files/26_NISAR_FINAL_9-6-19.pdf (last access: 23 September 2022), 2018.
- Nativi, S. and Bigagli, L.: Discovery, Mediation, and Access Services for Earth Observation Data, *IEEE J. Select. Top. Appl. Earth Observ. Remote Sens.*, 2, 233–240, 2009.
- Nativi, S., Craglia, M., and Pearlman, J.: *Earth Science Infrastructures Interoperability: The Brokering Approach*, *IEEE J. Select. Top. Appl. Earth Observ. Remote Sens.*, 6, 1118–1129, 2013.
- Nativi, S., Mazzetti, P., and Craglia, M.: *Digital Ecosystems for Developing Digital Twins of the Earth: The Destination Earth Case*, *Remote Sens.*, 13, 2119, <https://doi.org/10.3390/rs13112119>, 2021.
- Nearing, G. S., Kratzert, F., Sampson, A. K., Pelissier, C. S., Klotz, D., Frame, J. M., Prieto, C., and Gupta, H. V.: What role does hydrological science play in the age of ma-

- chine learning?, *Water Resour. Res.*, 57, e2020WR028091, <https://doi.org/10.1029/2020wr028091>, 2021.
- Nedovic-Budic, Z., Cromptoets, J., and Georgiadou, Y.: Spatial Data Infrastructures in Context: North and South, CRC Press, 288 pp., ISBN 9781439828038, 2011.
- Neitsch, S. L., Arnold, J. G., Kiniry, J. R., and Williams, J. R.: Soil and water assessment tool theoretical documentation version 2009, Tech. rep., Texas Water Resources Institute, https://oaktrust.library.tamu.edu/bitstream/handle/1969.1/128050/TR-406_SoilandWaterAssessmentToolTheoreticalDocumentation.pdf?sequence=1 (last access: 27 September 2022), 2011.
- Nguyen, G., Dlugolinsky, S., Bobák, M., Tran, V., García, Á. L., Heredia, I., Malík, P., and Hluch, L.: Machine learning and deep learning frameworks and libraries for large-scale data mining: a survey, *Artific. Intellig. Rev.*, 52, 77–124, 2019.
- Oleson, K. W., Lawrence, D. M., Bonan, G. B., Drewniak, B., Huang, M., Koven, C. D., Levis, S., Li, F., Riley, W. J., Subin, Z. M., Swenson, S. C., Thornton, P. E., Bozbiyik, A., Fisher, R., Heald, C. L., Kluzek, Erik Lamarque, J., Lawrence, P. J., Leung, L. R., Lipscomb, W., Muszala, S., Ricciuto, D. M., Sacks, W., Sun, Y., Tang, J., and Yang, Z.-L.: Technical Description of version 4.5 of the Community Land Model (CLM), NCAR, 434 pp., ISSN 2153-2400, 2013.
- Ott, J., Pritchard, M., Best, N., Linstead, E., Curcic, M., and Baldi, P.: A Fortran-Keras Deep Learning Bridge for Scientific Computing, *Sci. Program.*, 2020, 8888811, <https://doi.org/10.1155/2020/8888811>, 2020.
- Pan, B., Hsu, K., AghaKouchak, A., and Sorooshian, S.: Improving precipitation estimation using convolutional neural network, *Water Resour. Res.*, 55, 2301–2321, 2019.
- Paniconi, C. and Putti, M.: Physically based modeling in catchment hydrology at 50: Survey and outlook, *Water Resour. Res.*, 51, 7090–7129, 2015.
- Peckham, S. D., Hutton, E. W. H., and Norris, B.: A component-based approach to integrated modeling in the geosciences: The design of CSDMS, *Comput. Geosci.*, 53, 3–12, 2013.
- Peters, N. E., Freer, J., and Beven, K.: Modelling hydrologic responses in a small forested catchment (Panola Mountain, Georgia, USA): a comparison of the original and a new dynamic TOP-MODEL, *Hydrol. Process.*, 17, 345–362, 2003.
- Peters-Lidard, C. D., Houser, P. R., Tian, Y., Kumar, S. V., Geiger, J., Olden, S., Lighty, L., Doty, B., Dirmeyer, P., Adams, J., Mitchell, K., Wood, E. F., and Sheffield, J.: High-performance Earth system modeling with NASA/GSFC's Land Information System, *Innov. Syst. Softw. Eng.*, 3, 157–165, 2007.
- Pianosi, F., Beven, K., Freer, J., Hall, J. W., Rougier, J., Stephenson, D. B., and Wagener, T.: Sensitivity analysis of environmental models: A systematic review with practical workflow, *Environ. Modell. Softw.*, 79, 214–232, 2016.
- Post, D. E. and Votta, L. G.: Computational Science Demands a New Paradigm, *Phys. Today*, 58, 35–41, 2005.
- Post, H., Vrugt, J. A., Fox, A., Vereecken, H., and Hendricks Franssen, H.-J.: Estimation of Community Land Model parameters for an improved assessment of net carbon fluxes at European sites, *J. Geophys. Res.-Biogeosci.*, 122, 661–689, 2017.
- Prieto, C., Kavetski, D., Le Vine, N., Álvarez, C., and Medina, R.: Identification of dominant hydrological mechanisms using Bayesian inference, multiple statistical hypothesis testing, and flexible models, *Water Resour. Res.*, 57, <https://doi.org/10.1029/2020wr028338>, 2021.
- Rädle, R., Nouwens, M., Antonsen, K., Eagan, J. R., and Klok-mose, C. N.: Codestrates: Literate Computing with Webstrates, in: Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology, UIST '17, pp. 715–725, Association for Computing Machinery, New York, NY, USA, <https://doi.org/10.1145/3126594.3126642>, 2017.
- Rahman, J. M., Seaton, S. P., Perraud, J. M., Hotham, H., Verrelli, D. I., and Coleman, J. R.: It's TIME for a new environmental modelling framework, in: MODSIM 2003 International Congress on Modelling and Simulation, vol. 4, 1727–1732, Modelling and Simulation Society of Australia and New Zealand Inc. Townsville, http://www.research.div1.com.au/RESOURCES/research/publications/conferences/20030714ff_MODSIM2003/RahmanSeatonPerraudHothamVerrelliColeman2003_1727.n.pdf (last access: 27 September 2022), 2003.
- Ramadhan, A., Marshall, J., Souza, A., Wagner, G. L., Ponnapati, M., and Rackauckas, C.: Capturing missing physics in climate model parameterizations using neural differential equations, arXiv preprint arXiv:2010.12559, <http://arxiv.org/abs/2010.12559>, 2020.
- Refsgaard, J. C., van der Sluijs, J. P., Højberg, A. L., and Vanrolleghem, P. A.: Uncertainty in the environmental modelling process—a framework and guidance, *Environ. Modell. Softw.*, 22, 1543–1556, 2007.
- Refsgaard, J. C., Storm, B., and Clausen, T.: Système Hydrologique Européen (SHE): review and perspectives after 30 years development in distributed physically-based hydrological modelling, *Hydrol. Res.*, 41, 355–377, <https://doi.org/10.2166/nh.2010.009>, 2010.
- Reichle, R. H.: Data assimilation methods in the Earth sciences, *Adv. Water Resour.*, 31, 1411–1418, 2008.
- Reichstein, M., Camps-Valls, G., Stevens, B., Jung, M., Denzler, J., Carvalhais, N., and Nuno, P.: Deep learning and process understanding for data-driven Earth system science, *Nature*, 566, 195–204, 2019.
- Rew, R. and Davis, G.: NetCDF: an interface for scientific data access, *IEEE Comput. Graph. Appl.*, 10, 76–82, 1990.
- Righi, M., Andela, B., Eyring, V., Lauer, A., Predoi, V., Schlund, M., Vegas-Regidor, J., Bock, L., Brötz, B., de Mora, L., Diblen, F., Dreyer, L., Drost, N., Earnshaw, P., Hassler, B., Koldunov, N., Little, B., Loosveldt Tomas, S., and Zimmermann, K.: Earth System Model Evaluation Tool (ESMValTool) v2.0 – technical overview, *Geosci. Model Dev.*, 13, 1179–1199, <https://doi.org/10.5194/gmd-13-1179-2020>, 2020.
- Rigon, R., Bertoldi, G., and Over, T. M.: GEOTop: A Distributed Hydrological Model with Coupled Water and Energy Budgets, *J. Hydrometeorol.*, 7, 371–388, 2006.
- Riquelme, J. L. and Gjorgjieva, J.: Towards readable code in neuroscience, *Nat. Rev. Neurosci.*, 22, 257–258, 2021.
- Rizzoli, A. E., Svensson, M. G. E., Rowe, E., Donatelli, M., Muetzelfeldt, R. M., van der Wal, T., van Evert, F. K., and Villa, F.: Modelling framework (SeamFrame) requirements, Tech. rep., SEAMLESS, ISBN 90-8585-034-7, 49 pp., 2006.
- Rodríguez-Iturbe, I. and Rinaldo, A.: Fractal River Basins: Chance and Self-Organization, Cambridge University Press, ISBN 9780521004053, 526 pp., 2001.

- Ross, M. R. V., Topp, S. N., Appling, A. P., Yang, X., Kuhn, C., Butman, D., Simard, M., and Pavelsky, T. M.: AquaSat: A data set to enable remote sensing of water quality for inland waters, *Water Resour. Res.*, 55, 10012–10025, 2019.
- Rouson, D., Xia, J., and Xu, X.: *Scientific Software Design: The Object-Oriented Way*, The object-oriented way, Cambridge University Press, Cambridge, England, Cambridge, England, Cambridge University Press, ISBN 9781107415331, 406 pp., 2014.
- Savenije, H. H. G. and Hrachowitz, M.: HESS Opinions “Catchments as meta-organisms – a new blueprint for hydrological modelling”, *Hydrol. Earth Syst. Sci.*, 21, 1107–1116, <https://doi.org/10.5194/hess-21-1107-2017>, 2017.
- Semeraro, C., Lezoche, M., Panetto, H., and Dassisti, M.: Digital twin paradigm: A systematic literature review, *Comput. Industry*, 130, 103469, <https://doi.org/10.1016/j.compind.2021.103469>, 2021.
- Serafin, F.: Enabling modeling framework with surrogate modeling capabilities and complex networks, Ph.D. thesis, University of Trento, edited by: Rigon, R. and David, O., <http://eprints-phd.biblio.unitn.it/3650/> (last access: 27 September 2022), 2019.
- Serafin, F., David, O., Carlson, J. R., Green, T. R., and Rigon, R.: Bridging technology transfer boundaries: Integrated cloud services deliver results of nonlinear process models as surrogate model ensembles, *Environ. Modell. Softw.*, 146, 105231, <https://doi.org/10.1016/j.envsoft.2021.105231>, 2021.
- Shen, C., Laloy, E., Elshorbagy, A., Albert, A., Bales, J., Chang, F.-J., Ganguly, S., Hsu, K.-L., Kifer, D., Fang, Z., Fang, K., Li, D., Li, X., and Tsai, W.-P.: HESS Opinions: Incubating deep-learning-powered hydrologic science advances as a community, *Hydrol. Earth Syst. Sci.*, 22, 5639–5656, <https://doi.org/10.5194/hess-22-5639-2018>, 2018.
- Stacke, T. and Hagemann, S.: HydroPy (v1.0): a new global hydrology model written in Python, *Geosci. Model Dev.*, 14, 7795–7816, <https://doi.org/10.5194/gmd-14-7795-2021>, 2021.
- Stodden, V., Borwein, J., and Bailey, D. H.: Setting the default to reproducible, computational science research, *SIAM News*, 46, 4–6, 2013.
- Todini, E.: Hydrological catchment modelling: past, present and future, *Hydrol. Earth Syst. Sci.*, 11, 468–482, <https://doi.org/10.5194/hess-11-468-2007>, 2007.
- Trambauer, P., Dutra, E., Maskey, S., Werner, M., Pappenberger, F., van Beek, L. P. H., and Uhlenbrook, S.: Comparison of different evaporation estimates over the African continent, *Hydrol. Earth Syst. Sci.*, 18, 193–212, <https://doi.org/10.5194/hess-18-193-2014>, 2014.
- Tsai, W.-P., Feng, D., Pan, M., Beck, H., Lawson, K., Yang, Y., Liu, J., and Shen, C.: From calibration to parameter learning: Harnessing the scaling effects of big data in geoscientific modeling, *Nat. Commun.*, 12, 1–13, 2021.
- Tubini, N. and Rigon, R.: Implementing the Water, HEat and Transport model in GEOframe (WHETGEO-1D v.1.0): algorithms, informatics, design patterns, open science features, and 1D deployment, *Geosci. Model Dev.*, 15, 75–104, <https://doi.org/10.5194/gmd-15-75-2022>, 2022.
- Viglione, A., Borga, M., Balabanis, P., and Blöschl, G.: Barriers to the exchange of hydrometeorological data in Europe: Results from a survey and implications for data policy, *J. Hydrol.*, 394, 63–77, 2010.
- Voinov, A. and Shugart, H. H.: “Integronsters”, integral and integrated modeling, *Environm. Modell. Softw.*, 39, 149–158, 2013.
- Vrugt, J., van Wijk, M. T., Hopmans, J., and Šimunek, J.: One-, two-, and three-dimensional root water uptake functions for transient modeling, *Water Resour. Res.*, 37, 2457–2470, 2001.
- Vrugt, J. A. and Neuman, S. P.: Introduction to the special section in *Vadose Zone Journal*: Parameter identification and uncertainty assessment in the unsaturated zone, *Vadose Zone J.*, 5, 915–916, 2006.
- Vrugt, J. A., Gupta, H. V., Bouten, W., and Sorooshian, S.: A Shuffled Complex Evolution Metropolis algorithm for optimization and uncertainty assessment of hydrologic model parameters, *Water Resour. Res.*, 39, 279–218, <https://doi.org/10.1029/2002WR001642>, 2003.
- Wada, Y., Bierkens, M. F. P., de Roo, A., Dirmeyer, P. A., Famiglietti, J. S., Hanasaki, N., Konar, M., Liu, J., Müller Schmied, H., Oki, T., Pokhrel, Y., Sivapalan, M., Troy, T. J., van Dijk, A. I. J. M., van Emmerik, T., Van Huijgevoort, M. H. J., Van Lanen, H. A. J., Vörösmarty, C. J., Wanders, N., and Wheeler, H.: Human–water interface in hydrological modelling: current status and future directions, *Hydrol. Earth Syst. Sci.*, 21, 4169–4193, <https://doi.org/10.5194/hess-21-4169-2017>, 2017.
- Werner, M., Schellekens, J., Gijsbers, P., van Dijk, M., van den Akker, O., and Heynert, K.: The Delft-FEWS flow forecasting system, *Environ. Modell. Softw.*, 40, 65–77, 2013.
- Wood, E. F., Roundy, J. K., Troy, T. J., van Beek, L. P. H., Bierkens, M. F. P., Blyth, E., de Roo, A., Döll, P., Ek, M., Famiglietti, J., Gochis, D., van de Giesen, N., Houser, P., Jaffé, P. R., Kollet, S., Lehner, B., Lettenmaier, D. P., Peters-Lidard, C., Sivapalan, M., Sheffield, J., Wade, A., and Whitehead, P.: Hyperresolution global land surface modeling: Meeting a grand challenge for monitoring Earth’s terrestrial water: OPINION, *Water Resour. Res.*, 47, W05301, <https://doi.org/10.1029/2010wr010090>, 2011.
- Xie, Y.: knitr: A General-Purpose Package for Dynamic Report Generation in R, R package version, <https://rdr.io/github/yihui/knitr/man/knitr-package.html> (last access: 27 September 2022), 2013.
- Yeh, W. W.-G.: Review of parameter identification procedures in groundwater hydrology: The inverse problem, *Water Resour. Res.*, 22, 95–108, 1986.
- Yilmaz, K. K., Vrugt, J. A., Gupta, H. V., and Sorooshian, S.: Model calibration in watershed hydrology, *Advances in data-based approaches for hydrologic modeling and forecasting*, pp. 53–105, edited by: Sivakumar, B. and Berndtsson, R., WORLD SCIENTIFIC, https://doi.org/10.1142/9789814307987_0003, 2010.