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

Riccardo Rigon[1], Giuseppe Formetta[2], Marialaura Bancheri[3], Niccolò Tubini[2], Concetta D'Amato[1], Olaf David[4], and Christian Massari[5]

[1]Center Agriculture Food Environment - C3A, University of Trento, Trento, Italy
[2]Department of Civil, Environmental and Mechanical Engineering - DICAM, Trento, Italy
[3]Institute for Mediterranean Agricultural and Forestry Systems (ISAFOM), National Research Council (CNR),Italy
[4]Department of Civil and Environmental Engineering, Colorado State University, Fort Collins, CO, USA
[5]National Research Council, Research Institute for Geo-Hydrological Protection, Perugia, Italy

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

Abstract. The Digital Earth (DE) metaphor is very useful for both end users and for hydrological modellers (i.e., the coders). ~~However, in literature it can promote the erroneous view of models as a commodity, that is a basic good used in commerce that is interchangeable with other goods of the same type, without any warning about the fact that some models work better than others, some models just work while others can be simply wrong. These distinctions are at the core of doing good science. This opinion contribution, on the one hand, tries to accept the challenge of adopting models as commodities but, on the other, it wants to show that this acceptance comes with some consequences as to how models must be structured.~~ In this opinion paper, w~~We~~ analyse different categories of models, with the view of making them part of a Digital eARth Twin Hydrology system (called DARTH). We ~~also~~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 ~~and allows for~~ a modelling-by-components 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 modelling. In the first step, models are be decomposed into interacting modules with, for instance, the agnostic parts deal with inputs and outputs, separated from the model specific ones that contain the algorithms. In the other three steps, 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. There is a fifth step that would allow models to be selected as if they were interchangeable with others without giving deceptive answers. This step includes: use of hypothesis testing, inclusion of error of estimates, adoption of literate programming, and guidelines to obtain informative clean code.

The urgency for DARTHs to be Open Source ~~and well written pieces of codes~~ is ~~also discussed~~ 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 doing hydrological science, where researchers can contribute cooperatively to characterize and control model outcomes in various territories. ~~Furthermore, this has consequences for the engineering of the DARTHs systems.~~ Finally, three enabling technologies are also discussed in the context of DARTHs, i.e., Earth Observations (EO), High Performance Computing (HPC)

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 in 1998 the technologies

30 available were not at all adequate to pursue this vision, Al Gore's speech aroused great interest in the scientific community and in 1999 the first International Symposium on Digital Earth was held in Beijing China. 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 in 2008 it gave itself an international peer-reviewed academic journal, when the International Journal of Digital Earth was founded (IJDE). Since then, another milestone document, the 2019 Florence Declaration on Digital Earth,

35 was approved and position papers 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 to face issues characterizing the Anthropocene epoch. Thanks to the advancement of technologies, it is possible to talk about Digital Twin Earth models (DT). They want to accurately reproduce the state of

40 evolution of a generic physical entity through high-fidelity computer models, with the aim of understanding, simulating systems behaviors, and evaluating them under changing boundary and initial conditions. As such, the DT dates back to NASA's Apollo program (Semeraro et al., 2021) and, before, they were developed in sectors where the processes being modelled are well understood, such as industry and manufacturing (Graessler and Pöhler, 2017).

In modelling natural processes, DE models use the same concepts (e.g., physical principles, equations) and data as Hy-

45 drology. It is worth noting that as early as 1986 Eagleson recognized the need to develop hydrological models to produce hydrological prognoses at the global scale (Eagleson, 1986). 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, for example.

50 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 the related technologies. However, these same efforts often bring to light a "top-down" 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 do applications using already ~~developed~~ deployed tools, unless they are directly involved in the development. As such, this top-down approach could ~~treacherously~~ limit creativity ~~, the transmission of knowledge, and, ultimately,~~ 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 Digital Twin Earth campaign (e.g. https://digital-strategy.ec.europa.eu/en/policies/deearth) runs the same risk: it is time therefore 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.

This opinion paper aims, therefore, to:

- provide an up-to-date analysis of the current research in designing and developing DE components, with a particular focus on hydrological modelling;

- 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, 2021) and contributing to GIS developments (JGrass, uDig, gvSIG);

- debate and identify potential ways forward to answer the following questions: 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 modelling and data representation? How is the scientific reliability assessed ?~~science certified~~ ? What role can Earth observations (EO) play? Is Machine Learning (ML) the solution?

The paper is organized in four sections, with many subsections, and an appendix, and it has a few accompanying manuscripts. 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 regard data availability and data flow. They represent an introduction to the DARTHs topic. Section 2 deals with design and implementation requirements. Its subsections first introduce and then explore the suitable modelling architecture for the purpose of buildings DARTHs components. Reusability and the possibility of changes to scientific paradigms, with as few restrictions as possible, are presented as key DARTHs concepts. Since 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 do new hydrology. Sections 3.1 to 3.3 deal with three other of these technologies that are pervasive in contemporary sciences and can be seen as protagonists in the near-future developments: HPC, Earth Observations (EO) and the use of Machine Learning (ML) techniques, providing for each a short review of applications, issues and perspectives. Section 4 discusses what could be the governance and organization of communities of DARTHs developers. Conclusions follow. Finally, the supplemental material contains: a glossary of terms (where we define and explain acronyms and jargons used in the paper), with a specialized glossary regarding a classification of models from the point of view of DARTHs; a DARTHs cheatsheet that summarizes the contents of this paper; the results of a survey done among model and model infrastructures developers that tries to capture the present state of the art. Because the paper makes use of many acronyms, all of them are reported in Table A1.

1.1 From models to DARTHs

Since 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 "legacy more than adequacy" is the rule for researchers when choosing an application (Addor 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 ~~reasonable colour maps~~ plausible patterns and have no or few running issues). The current state of the hydrological science allows us to say that all these models work, at least for the purposes 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 thought for the general reuse that the DE paradigm would require. Many modular frameworks have been recently built with the aim to filling this gap, like SUMMA (Clark et al., 2011a), SuperPyflex (Dal Molin et al., 2021), GEOframe (Formetta et al., 2014), 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 ~~building~~, to cope with the DARTHs requirements that are addressed in the paper.

Broadly speaking, four ~~great paradigms~~ mathematical tools dominate in hydrological modelling research (Kampf and Burges, 2007): the process based models (PB) (Paniconi and Putti, 2015; Fatichi et al., 2016a), the reservoir type models (HDS, as Hydrological dynamical systems) (Todini, 2007; Bancheri et al., 2019b), the classical statistical models (McCuen, 2016), and the 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 which is a tight black-box between Earth-Observations and lumped models and/or machine learning models usually referred as EO products (Martens et al., 2016). Many references already discuss the taxonomy of models (e.g., Kampf and Burges (2007)), ~~The references cited already discuss~~ and 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 ~~have~~ just ~~to~~ adopt the pragmatic view that they exist, that, notwith-

standing their uncertain informatics quality (except maybe for the ML tools that are based on the use of large frameworks), they are used creatively, and 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 ~~modelling panorama~~ 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) and 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 modelling, ML tools and traditional lumped models) can be tested and compared, and eventually put to work together to assess the uncertainty in forecasting. This, in turn, has consequences on the code architectures, infrastructures and the informatics that have to be deployed. Some naive ideas that ML plus EO, 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 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 (Gharari et al. (2021)), in fact, it has become clear that ML and Deep Learning (DL) techniques can be interpreted and explained, thus they can be used as a tool for understanding (Arrieta et al., 2020) or model parameter learning (Tsai et al., 2021), instead of primarily for predictive purposes. ~~can be investigated and its knowledge explained. It has also become clear that Deep Learning (DL) techniques can be used as tools of interpretation or model parameters learning instead of primarily for predictive purposes.~~ In any case, ML growth ~~in parallel to traditional hydrological modelling and was~~ has been mostly driven by a large variety of problems, for instance, computer vision applications, and speech and natural language processing ~~. It solved these issues in their framework~~ 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 favour of one choice, then a suitable DARTH engine should allow for the implementation of various ones, and more than one competing DARTHs 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 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) where it is 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:

- be available on demand, working seamlessly on the cloud as web services

- have automatic ways of retrieving and providing data.

- ~~but they~~ and be interoperable with other models;

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

- 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 researchers that develop and use them. ~~Therefore, there is a need for a fundamental rethinking of how models are implemented and delivered.~~

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 ~~and open data formats~~

DARTHs, obviously, are useless if there are no data available to run them; therefore, we discuss briefly 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, 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, in the U.S. environmental data are public and supported by various initiatives; in other countries, i.e., in Austria (e.g., https://ehyd.gv.at), France (e.g., https://www.hydro.eaufrance.fr/), Germany (e.g., www.ded.de), U.K. (e.g., https://archive.ceda.ac.uk) there are projects that aim to bring the largest possible number of data to be 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 Global Runoff Data Center (GRDC, http:www.bafg.de/GRDC), the Global Water Monitoring System (GWMS: http://www.gemstat.org), the Global Earth Observation System of Systems (GEOSS:https://earthobservations.org/geoss.php), the WMO Hydrological Observing system (WHOS), and, more recently, the Destination Earth architecture. It is also worth mentioning the bottom-up effort started with the CAMELS datasets, so named from the acronym from Catchment Attributes and MEteorology for Large-sample Studies (Addor et al., 2017). Meanwhile, the Open Geospatial Consortium has seeded several projects (https://www.ogc.org/node/1535) 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, wiyhout forgetting the question of providing the appropriate ground based data for the needs of developing countries .

190   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 programme (https://sentinels.copernicus.eu) have already exceeded petabytes of volume. More and more

195   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 SWOT missions (https://swot.jpl.nasa.gov/mission/overview/), the US-Indian NASA-ISRO SAR (NASA-ISRO, 2018), the Radar Observing System for Europe—L-band (ROSE-L) (Davidson et al., 2019) SAR missions and LSTM missions, and the Thermal InfraRed Imaging Satellite for High-resolution Natural resource Assessment (TRISHNA)

200   (Lagouarde et al., 2019), in addition to a constellation of low-cost commercial CubeSats satellites (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) and the new born European Open EO (https://openeo.cloud). These earth observation data centres have revolutionized the way users interact with EO and offer integrated solutions to access large volumes of a variety of EO data with relatively high speed. With this new paradigm

205   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 programmes.

The state of art of the matter, for what can be important for DARTHs, is that data layers have to be somewhat loosely coupled

210   to models through 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.

A recent conference contribution (Boldrini et al., 2020) cites a few examples of brokers: WHOSWHOS; CUAHSI Hy-

215   droDesktop (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 off-line by the researchers, but the DARTH vision would require that these data be automatically ingested and processed.

220   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, and 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 modelling could be performed smoothly (the cloud strategy). The opposite approach is that data is acquired once and for all and locally stored (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, for instance, the distributed platforms Hadoop or Apache Spark (Nguyen et al., 2019) to facilitate this type of operations. However, at present hybrid solutions could be envisioned and deployed.

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/) for tabular data, can be an interesting choice and come with various interfaces to the most common programming languages, thus allowing for easier compatibility with existing models.

We are not claiming here that some formats should be preferable to others. However, some format, and some database architecture, should be chosen as a starting point along with a set of tools to transport data from the one format to others. 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/), from meteorology WMO, from ESA and 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 which are 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.

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

We have already supported the idea that DARTHs should be agnostic to the choices of science. However, this option has a few implications. As Rizzoli et al. (2006) realized, there are different types of users and different types of scopes when modelling hydrology. They are summarized in Figure 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 (API). Linkers are power users who assemble existing codes through scripting languages or procedures. This role is of particular importance in ML where very often earth scientists 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

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

Figure 1. After Rizzoli et al. (2006) - Models serve diverse scopes and uses. Therefore they are a fit-for-purposes task.

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, and 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 back-ground and formal training in computer science or software engineering (while vice-versa, computer scientists are not usually domain experts). The coders we are referring to here are not usually computer scientists, but domain experts, i.e. hydro-bio-geo experts, with limited computer science backgrounds.

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 modelling styles and programming paradigm changes, and, in general, not being invasive of programming habits (i.e. not forcing the programming to adopt constructs that computer scientists like but scientists and engineers cannot manage). However, the legacy within the infrastructure can potentially be a critical point (bottleneck) in the development of a DARTH model. The available environmental modelling frameworks, infrastructures to all effects, can be classified in two broad categories: heavyweight frameworks and lightweight frameworks (Lloyd et al., 2011). The former is characterized by large and unwieldy API that require a considerable effort from developers (scientists or soft coders) to become familiar with before writing new code. This becomes even more demanding if one considers that often research groups maintain many legacy models still in use based on algorithms and equations developed decades ago (David et al., 2014). Moreover, such an effort somehow creates a strict legacy

~~within the infrastructure and this somehow limits the possibility of having more than one modelling solution within the same DARTH.~~

~~Conversely, David et al. (2013), show how lightweight frameworks have many functions for the developer due to the~~ ~~techniques used to reduce the API's overall size and developer dependence on the API. 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 modellers as it allows the use of existing modelling code and libraries, integrating them into a larger framework. In a lightweight framework, model components can work and continue to evolve outside the framework so that 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 modelling solutions in existing models.~~

and DARTHs have to facilitate good programming and testing practices, bridging the knowledge gap of scientist/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 section 2.4.

Since 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, whether MS Windows, Mac OS, 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 ~~programmers~~ 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. DARTHs need, therefore, 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 the languages, or 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 make treasure of the efforts already made. This would bridge the past and future of hydrological and environmental modelling and make revolutionary changes possible in an incremental way. DARTHs should not subtract, just add and evolve.

There are at least two other matters about the model design requirements of DARTHs that have to be accomplished. The first is related to the specification of the data 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 dimension. 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, the 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 which are their spatial and temporal properties. The CF convention (https://cfconventions.org/) 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., 2013a), 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 in 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 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 care just about the models' i/o but also of their realism, reliability, (replicability-)reproducibility and robustness, ~~(R^4)~~ characteristics whose meaning is discussed thoroughly in the glossary. ~~Realism is concerned with reproducing natural behaviour: a realistic model reproduces the variety and behavioural (dynamical) complexity of the (hydrological) system it is intended to describe, but we do not discussed in this paper. Reliability is re-defined here as the possible to give an error of the estimated quantities in any of the circumstance and is discussed in section 2.5. Replicability refers to the fact that multiple runs of the same model with the same inputs and the same setup must always produce the same results.~~ This information, which, in part, can be tested a *posteriori* by use and comparison with ~~reality~~ real world observations, can also be gained by inspecting the models' ~~internals~~ codes with their implementation details. ~~Therefore, if 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.~~

To sum up, the DARTH architecture must account for various requirements:

– to be programming language agnostic;

– to work seamlessly under the various operating systems;

– to use standard, self-explanatory data formats;

– to use standard names for the quantities they treat in input and output.

2.2 Too complex systems do not serve the cause

Another characteristic deemed important for DARTHs infrastructure is that they should not be invasive of programming habits (i.e. not forcing the programming to adopt constructs that computer scientists like but scientists and engineers cannot manage). ~~However, the legacy within the infrastructure can potentially be a critical point (bottleneck) in the development of a DARTH model.~~ The available environmental modelling 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 API that require a considerable effort from developers (scientists or soft coders) to become familiar with before writing new code. ~~This becomes even more demanding if one considers that often research groups maintain many legacy models still in use based on algorithms and equations developed decades ago.~~ Moreover, such an effort somehow creates a strict

legacy within the infrastructure and this ~~somehow~~ has limited the diffusion of these systems in the recent past (David et al., 2014). ~~the possibility of having more than one modelling solution within the same DARTH.~~

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 modellers as it allows the use of existing modelling code and libraries, integrating them into a larger framework. In a lightweight framework, model components can work and continue to evolve outside the framework, and so 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 modelling solutions in existing models.

2.3 Participatory needs

If 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., 2021) 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 contents (Post and Votta, 2005).

We are committed to open science in this paper and 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) as well.

If we then consider the point of view of the runners, we are interested in the reliability of the contents ~~and how to certify the providers of models and information. Besides, we are interested~~ 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 modelling", 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 be a multitude of developers to exploit science bringing bottom-up contributions. These requisites have some consequences in the functional design of a model. Take for instance the case of the Object Modelling System (David et al., 2013): besides a fully configurable setup, it provides a standard organization of the files used for modelling, as presented in Figure 2. A runner can be provided with all they need (input data, parameters values, input and output file names, modeling solutions structure definition, etc.) 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 already prepared 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 the required information. The runner can then add new simulations, with changed parameters, modified spatial partitions and input data, and then distribute all of their improvements and results back to the community with which they are interacting. This procedure, yet not automatized, has been already widely used by the Authors in their
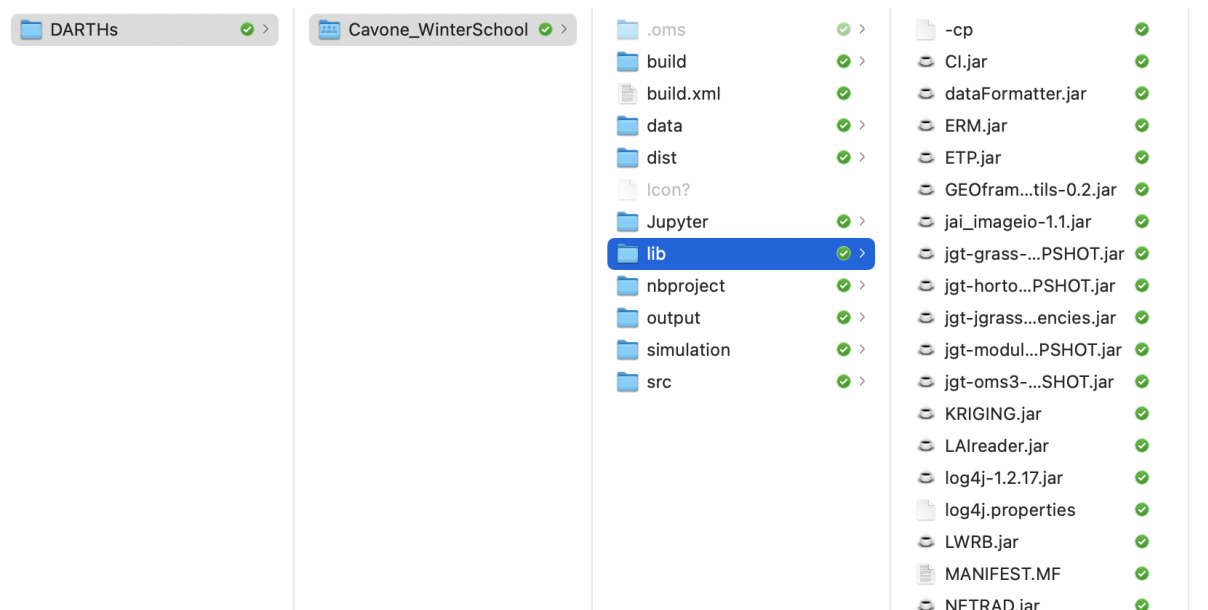
Figure 2. Standard organization of the OMS3 files with the libraries used highlighted

projects where collaborative efforts were necessary. The idea can be improved, no doubt, 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. Then, if we assign a unique identifier to any catchment worldwide, the procedure can even be decentralized and certified by using, for instance, block-chain-like technologies (Serafin et al., 2021). But, simple as it is, it is a proof of concept that such a strategy of participatory modelling is possible.

2.4 ~~The model architecture suitable for~~ Four, plus one, steps towards DARTHs

As described in the previous sections DARTHs can then be envisioned as a distributed ecosystem of softwares that serve multiple roles and users and where 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): this 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 the features that are required to have an all round modelling 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): it 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 and not with the model itself, or 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 we present in the supplemental material.

– Model-as-a-Service (MaaS): as for the previous case, a given implementation of the model runs on a specific server, but 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. Also the recent eWAterCycle (Hut et al., 2021) collector of models can 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): 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 interface, i.e., API. Besides, 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 of user intervention (at the model level).

– ~~Model-as-a-Commodity (MaaC): Models can be chosen from a pool present in the cloud. They can be automatically connected and made to run with a minimal intervention to respond to the needs of the users demands. MaaC can also be contributed to, modified and expanded by the technical user to fit their purposes and the pool increases as science advances.~~

MaaA, MaaT, MaaS and MaaR ~~and MaaC~~ can be thought of in order of suitability for DARTHs to fit the paradigm of the DE concept. MaaA 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 where independent revision is nearly impossible. Testing new features, which is per se a problematic issue (Kelly and Sanders, 2008), becomes almost unviable in MaaA. In fact, these models have been implemented as monolithic codes where the absence of separation of concerns makes it difficult to read/debug them (Serafin, 2019). If a modeller 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 are usually defined as "silos", in the sense that they cannot exchange data and procedures and do not favour the exchange of knowledge between related disciplines.

MaaT is a step forward that MaaA can gain after a robust refactoring. They preserve as benefits (from MaaA) 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 ~~provider~~ developers who not only ~~still~~ establish~~es the policies~~ how and when
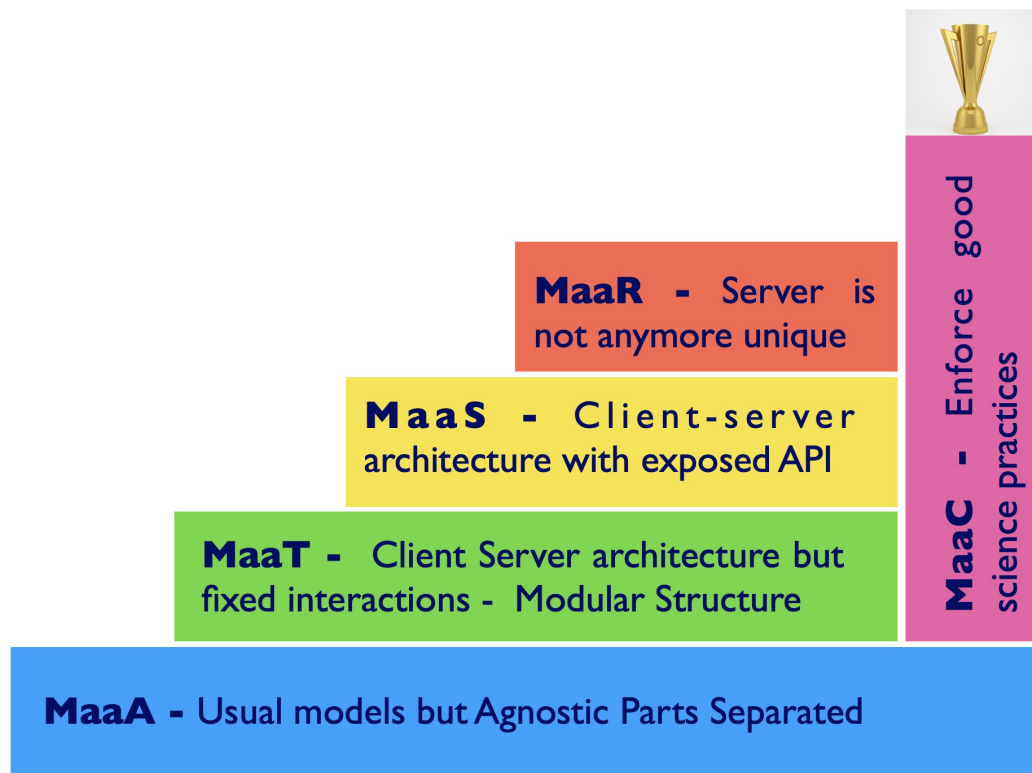
Figure 3. The convergence of present models to DARTHs aware models can be seen as a few steps that involve introducing some software layers, even starting from present modularized models (MaaS). They include the structuring of client-server architectures initially with a fixed client interface and server (MaaT), a client-server architecture where the connections can be obtained machine-to-machine by API (MaaS), and multiplicity of client and servers where models can be dispatched at run time according to the computational needs without intervention of the users (MaaR). At the same time, in order to fulfill the realistic and reliability requirements, a certain number of open science-aware models that fall under the name of MaaC should be pursued.

the models can be used ~~the models use~~ but also control the model evolution and enhancement. Clearly if the provider is an organized community with rules for getting contributions, the model can be a "community model", in the sense, for instance, that CLM 5 (Lawrence et al., 2019) and Jules (Best et al., 2011) are.

425   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 a dependence of the user on the provider of the modelling services. What we should aim for with DARTHs is to accomplish the more flexible possibilities represented by the MaaR. MaaR allows one to effectively
430   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, and protocols to make the parts of the system work together without side effects. While fully-fledged MaaRs do not yet exist, for instance the Cloud Services Innovation Platform (CSIP) (David et al., 2014) and CSDMS Peckham et al. (2013a) are something between MaaS and MaaR. However, so far, they have not been as widely adopted as they could be, indicating some possible complexity in their use that coders and runners could not face properly, or 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) 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/) 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 contents of the models that can be run on the infrastructure. Therefore, we claim that a further, fifth step, has to be made, which covers both the science and technological sides, to obtain:

- Model-as-a-Commodity (MaaC):In the case models can be chosen from a pool present in the cloud, not only can they be automatically connected and made to run with a 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 are also required, by construction, to embed tools for assessing their reliability, support literate computing, as supported for instance by Notebooks (Lau et al., 2020) (but not limited to them) and allow for hypothesis testing.

MaaC ~~, however, are even more~~ are demanding, as they should make the process of coupling models and the addition of features as automatic as possible, with the least input possible from the runner or coder. Actually, 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. (2013a). As shown in Figure 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 modelling infrastructures is the possibility of breaking down models into parts that can be reassembled for a specific purpose, which all depends on the possibility of a modelling infrastructure to make model parts interact, and to allow them to be modified, evolved or changed. A mature DARTH ~~however~~ requires ~~the MaaC architecture~~ MaaCs.

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 systems, which spans different methods, roles, uses and the needs to describe different resolutions and scales, and needs to consider new assumptions and paradigms, and extend their scientific domain over several traditional disciplines (Savenije and Hrachowitz, 2017; Bancheri et al., 2019b).
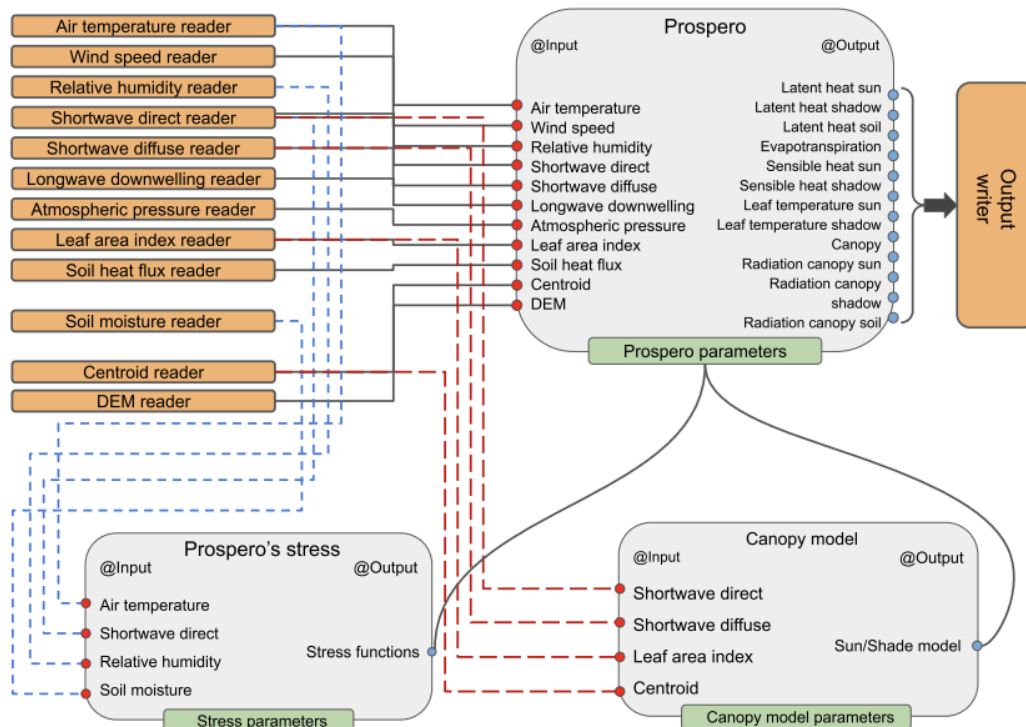
Figure 4. A visualization of models components. In this case, Prospero is a component of the GEOframe system (Bottazzi et al., 2021) that estimates evapotranspiration.

~~However,~~ Unlike past practices, structuring the software into composable parts, called "components" should become the
465 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 the other describing the constraints (stresses) limiting evaporation. Each component exchanges information with the other through their inputs and outputs, as indicated by the arrows in Figure 4. All internal components, per se, are 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 actually be changed in favour of others
470 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 the case illustrated by using a script in a Domain Specific Language, DSL).

This modelling-by-component (MBC) approach has actually existed for more than forty years (Holling, 1978) but it is only in the last twenty years that it has gained momentum in the environmental modelling community (Argent, 2004). Often, it was

475 named "Integrated Environmental Modeling" (IEM) because it responds to the request of studying heterogeneous processes together and it integrates knowledge from various disciplines (Moore and Hughes, 2017).

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

480 MBC concepts and its technological consequences 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, OMS faced this issue, explicitly finding encouraging solutions (Lloyd et al., 2011). MBC promotes server-oriented-architectures of the software (SOA), which is the same type of software architecture that was requested for treating the heterogeneous data sources. In principle, the SOA framework can work on different machines,

485 and be 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 MBC can be recognized to be:

- Encapsulation that simplifies code inspection. Components can work stand-alone (supported by a given infrastructure) and each of them can be tested separately;

- Well established intellectual property. Each component usually has a few contributors and different components involve
490 a diversity of developers without being dispersed in thousands lines of code. Adding components is always possible and does not require a recompilation of the whole system;

- Substitution of components is easy and the use of components in hypothesis testing favoured (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;

495 - In a well designed system, the composition of modelling solutions is practically unlimited and components ~~are not confined to convey science from a single discipline~~ can accomplish a wide range of tasks, not necessarily from the same discipline (i.e., silos are avoided).

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, since it

500 can potentially lead to the spreading of ~~fake~~ unreliable or untested models. Because of this, MaaC should not be limited to simply providing the results of simulations, but 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

505 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

Figure 5. Representation of the Embedded Reservoir Model, Bancheri et al. (2019a), that is based on a system of six ordinary differential equations that represent the hydrological fluxes in a point. HRU stands for Hydrologic Response Units. See the text for further explanations.

not only the code has to be open source, but also understandable. Millman et al. (2018) offers an overview of good practices to be followed. Years ago, Donald Knuth proposed the concept of literate programming (Knuth, 1984). 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 Knuth's vision, 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. Now, 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).

520   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 UML diagrams can help this phase, but it is usually a neglected part of the documentation. Some languages, such as Java, have recently 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

525   (https://maven.apache.org/) or Gradle (https://gradle.org/), that collect all the dependencies needed to compile the codes and, if well written, at least clarify the dependences at package level. These tools are much more evolved than the traditional "make" command of the C language, since they allow one to grab codes from globally diffuse repositories and keep each building process up-to-date, together with the latest version of the libraries they use.

    Going even more in depth into the programming, one aspect that is usually ignored by scientific programmers is that object

530   oriented programming is not just about using classes but 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 FORTRAN95 and Gardner and Manduchi (2007) for JAVA. Another step forward can be made by using a certain degree of abstraction in software, as

535   proposed, for instance, by Berti (2000) with examples in C++ and used in Tubini and Rigon (2021) using 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", thus enormously increasing the flexibility of a project and its lifetime. This software engineering technique enhances the flexibility already available by MBC and permits it to limit the code dimension and, when well engineered, a massive code reuse.

540   All of the above prescriptions can have different deployments in different languages but the overall principles are: reuse code, 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 LINK, 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

545   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/) and, more recently, by the Jupyter Notebooks (Loizides and Schmidt, 2016) and others (Lau et al., 2020). One necessary characteristic of open science

550   is to keep track of the setting of performed simulations and of the data manipulation used to infer results. Some frameworks, like OMS (David, 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 (Formetta et al., 2022 in preparation) users

555 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

560 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

565 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 modelling, Vrugt et al. (2001) for root uptake,

570 and Yilmaz et al. (2010) for general watershed modelling.

Literature is often concerned with where the errors are. Uncertainty comes from errors in data, epistemic errors, errors in methods for parameter calibration, and heterogeneity, i.e. the variability of the domain to be described and modelled. 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 ~~some~~ conditions that, being unknown yet necessary to completely define the mathematical

575 problem, must be guessed.

Errors in measurements affect the procedure of calibration and cause the inference of incorrect model parameters. Errors in model structure reflect in wrong forecasts, which in turn cause biased comparisons between them and the models outputs.

Figure 6 portrays the life cycle of a model: circles represent data; 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

580 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

585 further level of analysis with stakeholders.
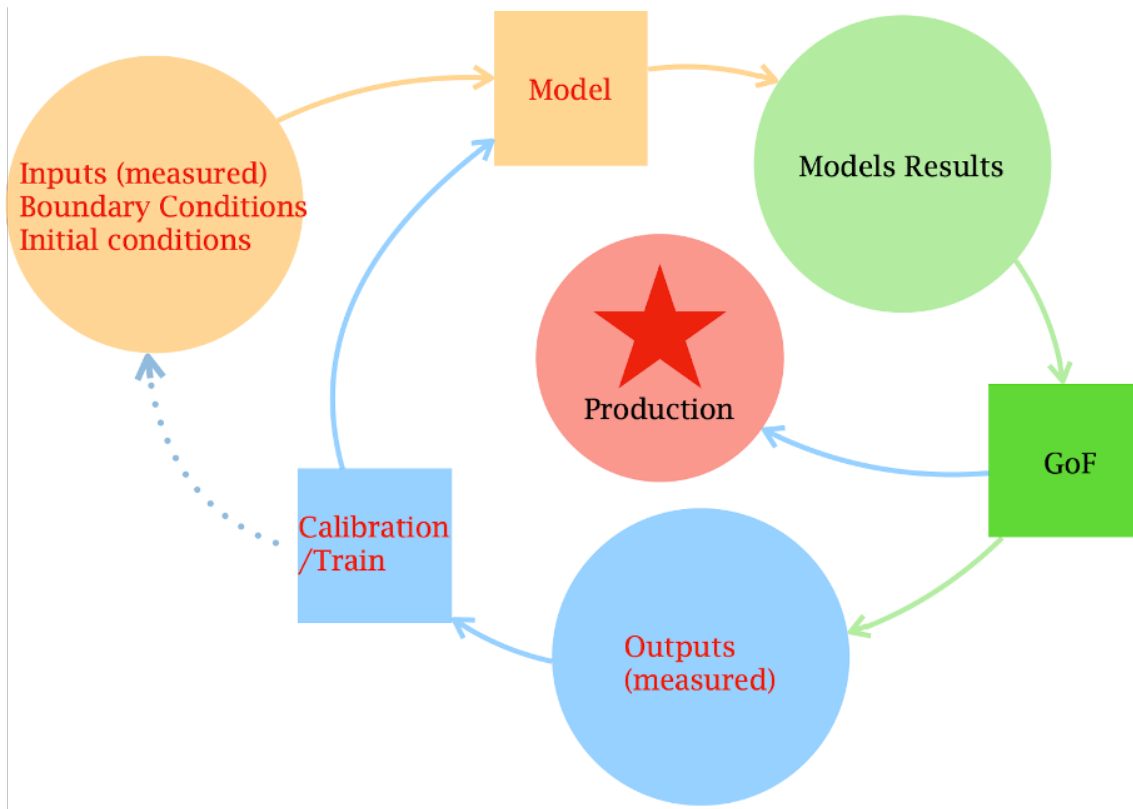
Figure 6. The life cycle of a model.

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 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 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 ~~quite~~ restricted, given the laws of dynamics and thermodynamics, DARTHs should by design give ~~some type of~~ 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 (plus 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, for that which regards DARTHs we stay with the simplest fact: "purely empirically, probability and statistics can, of course, describe anything from observations to model residuals, regardless

600 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, then statistics of the performance of a certain model setup becomes more reliable with time. As a consequence, we can easily figure out what was 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

605 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 the recent developments, they can also contain new

610 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

615 appropriate scalability among multi-core machines and cloud infrastructures; these are further requests for the HPC applied research.

The DARTH metaphor requires an extremely large use of data exchange ~~under the hood~~ 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

620 runner, it is important to have self-explanatory data formats that are efficient in memory (Lentner, 2019). DARTH action, as it has been envisioned so far, can be represented by a graph where the nodes are computational points, while the arrows represent potential exchanges of data. In Figure 4, we already presented what a chunk of a DARTH should be (without parallel routes). A more complete view is presented in Figure 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

625 equations (ODEs). In Figure 5, each ODE is represented by a circle, while 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 QT, 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 components, and some other components are required to estimate the fluxes (i.e. the squares). For instance, the green Evapotranspiration flux, ETc, in Figure 5 corresponds to components represented in Figure 4. Therefore, stripping

630 down the mathematics can reveal further possibilities for parallelization. In the hands of computer scientists this can be done

and, 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 Figure 7, 32 Hydrologic Response Units (HRUs) and two lakes are distinguished, and each one 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 Figure 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 Figure 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 Figure 7, each square node corresponds to a system that is, in most cases, equal or similar to the one presented in Figure 5. Therefore, the underlying infrastructure has to harmonize the various levels of parallelization: the one at 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 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 performances out of any hardware. However, here what matters 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 parallelizing the codes and instead focus on the tasks 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 partition of catchments offers the possibility of using different types of models in different locations. For instance, in Figure 7 the two lakes (blue and red nodes) are modelled 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 scope, however, catchments (and their partitioning HRU) need to have a unique identification, ultimately all over the world: but 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
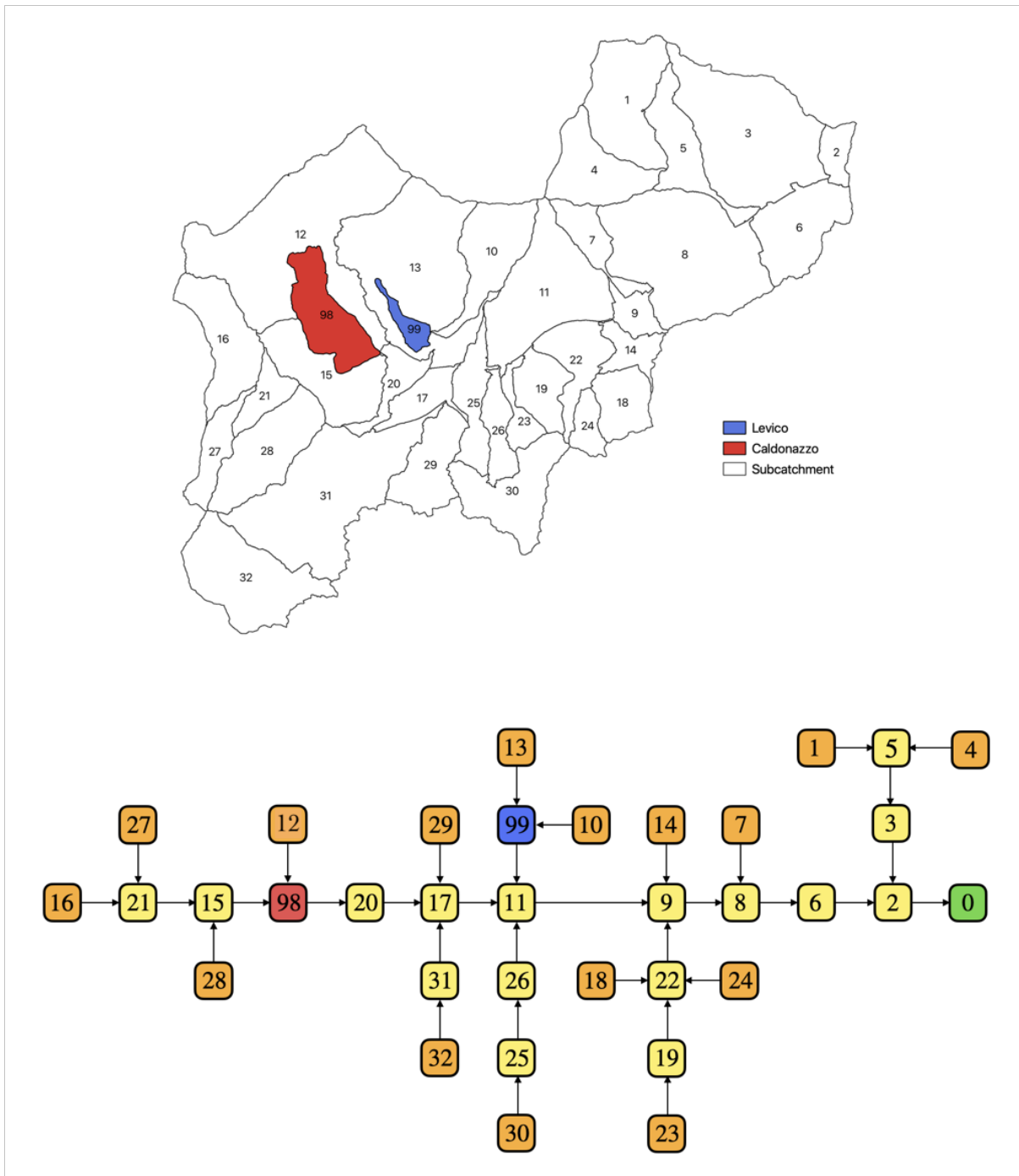
Figure 7. The upper river Brenta (Italy) is a catchment subdivided in 34 Hydrologic Response Units, including the two lakes of Caldonazzo and Levico. The channel network gives an ordered structure to the geographic region that can be used to process the HRU in parallel Modified from **?**

does not seem to fit in the nested-graph structure just envisioned. This objection is only partially valid, however, since 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 organized in 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 modelling 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 this upon the adoption of the right framework, which appears, at this point, to be twice 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 Observation (EO) as the basis of the Digital Twin Earth (DTE). ESA and NASA have enthusiastically embraced the idea within their programmes (https://www.esa.int/ Applications/Observing_the_Earth/Working_towards_a_Digital_Twin_of_Earth). 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, and 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), 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 EO, 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 (www.openda.org) and the Land Information System (LIS, Kumar et al. (2006); Peters-Lidard et al. (2007a)). On the other hand, in the hydrological community this has only partially happened. 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 which 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 and benefit from the granularity permitted by MBC and 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., 2016; 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. (2016) 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 assumptions, 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 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 EO, 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 bring to 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; and vice-versa, remote-sensing scientists should take care of the suggestions and criticism made by hydrologists.

3.3 The grand challenge of hybrid models in DARTHs

We have been agnostic, so far, with respect to type of models, whether DPB, HDS, MS 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, one 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 of 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 of paradigm that is now expected will bring the possibility of merging different families of models within the MBC informatics, and of hybrid modelling solutions (Shen et al., 2018) where MS, ML, HDS and PB can be mixed. ML techniques are current practice when using PB or HDS models, as for instance in the calibration phases where genetic algorithms have been used since Vrugt et al. (2003), or particle swarm (Kennedy and Eberhart, 1997), but these techniques are not normally used in core modelling. Conversely, the community is more oriented to assimilate EO 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 of spatial and temporal scales (small scale vs large, global ones, hourly vs daily or larger time scales) generate misunderstandings and imprecise claims on the performances 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 modelling, for instance, for calibration (Krapu et al., 2019), downscaling of hydrologic data (Abbaszadeh et al., 2019), rainfall-runoff modelling (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 (CNN) 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, in the production phase the same workflow is maintained. 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 PDE or ODE, and some others, simultaneously solved, are ML based like LSTM (Long Term Short Term 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 modelling chains were developed somewhat differently from those traditionally used in hydrology and Earth sciences (Geer, 2021). These tools were created under the modelling 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 PDE or ODE 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 equations 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 Convolution Neural Networks (CNN) or Recurrent Neural Networks (RNN). 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 library (cuDNN), NVIDIA Collective Communications Library (NCCL), and cuBLAS (the GPU implementation of 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/), TensorFlow (Abadi et al., 2016), ML4J (https://github.com/ml4jarchive), H2O (Cook, 2016). Besides, 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 are also different. Earth science applications tend to use supercomputers, as weather and climate models, or PB 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 (GPUs or TPUs). One practical problem, as highlighted in the previous section, is to combine typical MaaA supercomputer-friendly models with ML softwares.

In principle when using MBC, it would be conceptually easy to say that, in a given modelling solution, some of the components can be PB and some others ML based, and the workflow envisioned in the cited papers be implemented. For instance, (Serafin et al., 2021), has 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, though not trivial, but 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 to MaaC type of models with the Keras framework. Fortunately, the ML types of infrastructure are often engineered as SOA softwares and, 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), but in some frameworks this graph is static, e.g., TensorFlow and Caffe2, while 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 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 require computations on Direct Acyclic Graphs (DAG), it is possible to think that new common techniques of calculation can be shared by ML and PB models.

For the emerging world of HM 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;

- A possible re-engineering of the infrastructures to allow a finer granularity of the hybrid process modelling 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 DARTHs solution. Moreover, various DARTHs can share some services and develop others for themselves, such as GDAL libraries that are used by most of the Open Source GIS systems or the Lapack libraries that are widely present in modelling infrastructures. We can think of DARTH components as libraries but more shareable on the web and, in any case, as pieces of software that are self-contained and possibly can 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. Since, for instance, EO data are provided by institutions like JAXA or 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 Application Programming Interface, 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 and the ESMVARtool (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 be eventually linked together to have a functioning DARTHs solution. Deployments like those initiated with eWaterCycle or Deltares-FEWS (Werner et al., 2013) and LIS (Peters-Lidard et al., 2007b) are examples of integrators of resources; maybe they are not fully compliant with the DARTHs architecture but they can be a good starting point to deploy DARTHs and provide an example of DARTHs providers.

Further ideas are expressed in Nativi et al. (2021), whcic has 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/) is an example of governance for such communities. Examples of successful communities can be seen also 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 and others, including GNU, provide additional tools. And at the end of the process other subjects deploy variously flavoured 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. So DARTHs can be both commercial or non-commercial but, either way, would require 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 programmes. The OpenEO initiative, https://openeo.org, in the field of Earth observation data development is an example of how a DARTH development could be funded. The entire 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), GEOframe (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 DARTHs components to form a DARTH distribution. Certainly, new techniques like the block-chain can be used to certify the steps and the chain of responsibility that a certain DARTH distribution has followed. It is expected that DARTHs 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) 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, simulations 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-COVID2 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 errors 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 possibility to use the Modelling 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.

Besides, we claim that this infrastructure should be able to:

- 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 think necessary for the progress of science. We also gave indications about deploying the necessary code and computing literacy.

We did not discuss the contents or the structure of the models but claimed that:

- DARTHs should be agnostic with respect to the model choice in order to support the progress of science;

- DARTHs should favour multiple hypothesis testing, which was fully discussed in many recent contributions (Clark et al., 2011a; Prieto et al., 2021; Fenicia and Kavetski, 2021).

Besides, to help progress, we categorized the models according to their characteristics in being able to cope with the DARTHs 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 EO 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 favour 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 having 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 the 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 DAG that has just started being explored and that could unify the way 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 believe, great advancements and innovations in hydrology can only be achieved by treating all aspects discussed together.

Video supplement. TEXT

Appendix A: Table of Acronyms

Author contributions. The first Author had the idea of discussing the production of DT in Hydrology. All the Authors contributed creatively and equally likely to all the phases of work.

Competing interests. There are no competing interests

Acknowledgements. This paper has been partially supported by MIUR Project (PRIN 2017) "WATer mixing in the critical ZONe: observations and predictions under environmental changes-WATZON" (project code: 2017SL7ABC), the "Carbon and water cycles interactions during drought and their impact on WAter and ForEst Resources in the Mediterranean region" "WAFER" @CNR (Consiglio Nazionale delle Ricerche) and the ESA Digital Twin Earth Hydrology (ESA Contract 4000129870/19/I-NB)). We thank the two reviewers, Dr. Mark Hrachowitz and Dr. Uwe Ehret, and the Editor Dr. Erwin Zehe whose careful review helped to greatly improve the manuscript. English was revised by Eng. Joseph Tomasi.

References

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., et al.: Tensorflow: A system for large-scale machine learning, 12th USENIX Symposium on Operating Systems Design and Implementation, 2016.

Abbaszadeh, P., Moradkhani, H., and Daescu, D. N.: The quest for model uncertainty quantification: A hybrid ensemble and variational data assimilation framework, Water resources research, 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., et al.: Human domination of the global water cycle absent from depictions and perceptions, Nature Geoscience, 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, 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, 10 078–10 091, 2015.

Argent, R. M.: An overview of model integration for environmental applications—components, frameworks and semantics, Environmental Modelling & Software, 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., and Others: SWAT: Model use, calibration, and validation, Transactions of the 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., et al.: Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI, Information 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, Reviews of Geophysics, 57, 530–616, 2019.

Ballatore, A.: The myth of the Digital Earth between fragmentation and wholeness, Wi: Journal of Mobile Media, 8, 2014.

Bancheri, M., Rigon, R., and Manfreda, S.: The GEOframe-NewAge Modelling System Applied in a Data Scarce Environment, Water, 12, 86, 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., Roo, A. d., 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, 2017.

Berti, G.: Generic software components for Scientific Computing, Ph.D. thesis, Brandenburgischen Technischen Universitat Cottbus, 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, Journal of Hydrology, 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, Geoscientific Model Development, 4, 677–699, 2011.

Beven, K.: Towards an alternative blueprint for a physically based digitally simulated hydrologic response modelling system, Hydrol. Process., 16, 189–206, 2002.

955 Beven, K.: Environmental modelling: an uncertain future?, CRC press, 2018.

Beven, K.: Towards a methodology for testing models as hypotheses in the inexact sciences, Proc. Math. Phys. Eng. Sci., 475, 20180 862, 2019.

Beven, K. J.: Rainfall-Runoff Modelling: The Primer, The Primer, John Wiley & Sons, 2012.

Binley, A. M., Beven, K. J., Calver, A., and Watts, L.: Changing responses in hydrology: assessing the uncertainty in physically based model
960   predictions, Water Resources Research, 27, 1253–1261, 1991.

Blechta, J., Hake, J., Johansson, A., and others: The FEniCS project version 1.5, Archive of Numerical Software, 2015.

Blöschl, G.: Debates-Hypothesis testing in hydrology: Introduction, Water Resour. Res., 53, 1767–1769, 2017.

Blöschl, G., Bierkens, M. F., Chambel, A., Cudennec, C., Destouni, G., Fiori, A., Kirchner, J. W., McDonnell, J. J., Savenije, H. H., Sivapalan,
        M., et al.: Twenty-three unsolved problems in hydrology (UPH)–a community perspective, Hydrological sciences journal, 64, 1141–1158,
965   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, 2020.

Bottazzi, M., Bancheri, M., Mobilia, M., Bertoldi, G., Longobardi, A., and Rigon, R.: Comparing Evapotranspiration Estimates from the
970   GEOframe-Prospero Model with Penman–Monteith and Priestley-Taylor Approaches under Different Climate Conditions, Water, 13,
        1221, 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, Journal of Geophysical Research: Atmospheres, 119,
        5128–5141, 2014.

975 Brunner, P. and Simmons, C. T.: HydroGeoSphere: A fully integrated, physically based hydrological model, Ground Water, 50, 170–176,
        2012.

CEOS: Committee on Earth Observation Satellites (Ceos), http://database.eohandbook.com/, 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
980   to enable broader participation and applications, Earth Sci. Rev., 207, 103 223, 2020.

Chew, C. and Small, E.: Soil moisture sensing using spaceborne GNSS reflections: Comparison of CYGNSS reflectivity to SMAP soil
        moisture, Geophysical Research Letters, 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.

985 Chu Xuefeng and Steinman Alan: 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, 2011a.

Clark, M. P., Kavetski, D., and Fenicia, F.: Pursuing the method of multiple working hypotheses for hydrological modeling, Water Resour.
        Res., 47, 2011b.

990   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.

995   Cook, D.: Practical Machine Learning with H2O: Powerful, Scalable Techniques for Deep Learning and AI, "O'Reilly Media, Inc.", 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, Journal of hydrology, 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

1000   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, 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, Environmental Modelling & Software, 129, 104 728, 2020.

1005   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, 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, Environmental Modelling & Software, 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)

1010   using the Cloud Services Innovation Platform (CSIP), in: International Congress on Environmental Modelling and Software, scholarsarchive.byu.edu, 2014.

Davidson, M., Chini, M., Dierking, W., Djavidnia, S., Haarpaintner, J., Hajduch, G., Laurin, G., Lavalle, M., López-Martínez, C., Nagler, T., et al.: 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, 2019.

1015   De Lannoy, G. J. and Reichle, R. H.: Global assimilation of multiangle and multipolarization SMOS brightness temperature observations into the GEOS-5 catchment land surface model for soil moisture estimation, Journal of Hydrometeorology, 17, 669–691, 2016.

Dey, C. and Others: 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.[Available online at http://www.wmo.int/pages/prog/www/WMOCodes/Guides//GRIB/GRIB1-Contents.html, 2007.

1020   Döll, P., Kaspar, F., and Lehner, B.: A global hydrological model for deriving water availability indicators: model tuning and validation, Journal of Hydrology, 270, 105–134, 2003.

Duan, Q., Gupta, H. V., Sorooshian, S., Rousseau, A. N., and Turcotte, R.: Calibration of Watershed Models, John Wiley & Sons, 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,

1025   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, GMD, 2013.

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, Geoscientific Model Development, 7, 2831–2857, 2014.

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 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., et al.: An overview of current applications, challenges, and future trends in distributed process-based models in hydrology, Journal of Hydrology, 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, 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, pp. 36–47, Association for Computing Machinery, New York, NY, USA, 2011.

Formetta, G., Antonello, A., Franceschi, S., David, O., and Rigon, R.: Hydrological modelling with components: A GIS-based open-source framework, Environmental Modelling & Software, 55, 190–200, 2014.

Freeze, R. A. and Harlan, R. L.: Blueprint for a physically-based, digitally-simulated hydrologic response model, J. Hydrol., 1969.

Gamma, E., Helm, R., Johnson, R., . Johnson, R. E., and Vlissides, J.: Design Patterns: Elements of Reusable Object-Oriented Software, Pearson Deutschland GmbH, 1995.

Gardner, H. and Manduchi, G.: Design Patterns for e-Science, Springer Science & Business Media, 2007.

Geer, A. J.: Learning earth system models from observations: machine learning or data assimilation?, Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences, 379, 20200 089, 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, 2021.

Gironás, J., Roesner, L. A., Rossman, L. A., and Davis, J.: A new applications manual for the Storm Water Management Model(SWMM), Environmental Modelling & Software, 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., et al.: Next-generation digital earth, Proceedings of the National Academy of Sciences, 109, 11 088–11 094, 2012.

Götzinger, J. and Bárdossy, A.: Generic error model for calibration and uncertainty estimation of hydrological models, Water Resources Research, 44, 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), pp. 289–293, IEEE, 2017.

Gregersen, J. B., Gijsbers, P. J. A., and Westen, S. J. P.: OpenMI: Open modelling interface, J. hydroinformatics, 9, 175–191, 2007.

Guo, H., Goodchild, M. F., and Annoni, A.: Manual of Digital Earth, Springer Singapore, 2019.

Gupta, H. V., Sorooshian, S., and Yapo, P. O.: Status of automatic calibration for hydrologic models: Comparison with multilevel expert calibration, Journal of hydrologic engineering, 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. Discussion, pp. 1–23, 2021.

Hill, M. C. and Tiedeman, C. R.: Effective groundwater model calibration with analysis of data, sensitivities, and uncertainty, 2007.

Holling, C. S.: Adaptive Environmental Assessment and Management, International Institute for Applied Systems Analysis, 1978.

Hornik, K.: Approximation capabilities of multilayer feedforward networks, Neural Netw., 4, 251–257, 1991.

Hrachowitz and Clark, M. P.: The complementary merits of competing modelling philosophies in hydrology, M.Hydrology and Earth System
Sciences, 21, 3953–3973, 2017.

Hut, R., Drost, N., van de Giesen, N., van Werkhoven, B., Abdollahi, B., Aerts, J., Albers, T., Alidoost, F., Andela, B., Camphuijsen, J.,
Dzigan, Y., van Haren, R., Hutton, E., Kalverla, P., van Meersbergen, M., van den Oord, G., Pelupessy, I., Smeets, S., Verhoeven, S.,
de Vos, M., and Weel, B.: The eWaterCycle platform for Open and FAIR Hydrological collaboration, 2021.

Jasak, H., Jemcov, A., and Tukovic, Z.: OpenFOAM: A C++ library for complex physics simulations, Coupled Methods in Numerical
Dynamics, 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), Environmental Modelling & Software, 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
Geoscience and Remote Sensing Magazine, 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), pp. 30–36, Citeseer, 2008.

Kennedy, J. and Eberhart, R. C.: A discrete binary version of the particle swarm algorithm, in: 1997 IEEE International Conference on
Systems, Man, and Cybernetics. Computational Cybernetics and Simulation, vol. 5, pp. 4104–4108 vol.5, 1997.

Knoben, W. J., Freer, J. E., Fowler, K. J., Peel, M. C., and Woods, R. A.: Modular Assessment of Rainfall–Runoff Models Toolbox (MAR-
RMoT) v1. 2: an open-source, extendable framework providing implementations of 46 conceptual hydrologic models as continuous
state-space formulations, Geoscientific Model Development, 12, 2463–2480, 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, 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, 1983.

Konapala, G. and Mishra, A.: Quantifying climate and catchment control on hydrological drought in the continental United States, Water
Resour. Res., 56, 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, 104 022, 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, pp. 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 Resources Research, 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, Hydrology and Earth System Sciences, 22, 6005–6022, 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, Geoscientific Model Development, 13, 1373–1397, 2020.

Kumar, S., Peters-Lidard, C., Santanello, J., Reichle, R., Draper, C., Koster, R., Nearing, G., and Jasinski, M.: Evaluating the utility of satellite soil moisture retrievals over irrigated areas and the ability of land data assimilation methods to correct for unmodeled processes, Hydrology and Earth System Sciences, 19, 4463–4478, 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, Environmental Modelling & Software, 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., et al.: 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, p. 403, 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, 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., 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, 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, 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 Resources Research, 51, 7309–7342, 2015.

Levia, D. F., Carlyle-Moses, D. E., Michalzik, B., Nanko, K., and Tischer, A.: Forest-water interactions, Springer, 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 Sensing of Environment, 189, 194–210, 2017.

1145 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), pp. 214–218, 2019.

Liu, Y. and Wu, L.: Geological Disaster Recognition on Optical Remote Sensing Images Using Deep Learning, Procedia Computer Science, 91, 566–575, https://doi.org/10.1016/j.procs.2016.07.144, 2016.

1150 Liu, Y., Weerts, A., Clark, M., Hendricks Franssen, H.-J., Kumar, S., Moradkhani, H., Seo, D.-J., Schwanenberg, D., Smith, P., Van Dijk, A., et al.: Advancing data assimilation in operational hydrologic forecasting: progresses, challenges, and emerging opportunities, Hydrology and Earth System Sciences, 16, 3863–3887, 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, Environmental Modelling & Software, 26, 1240–1250, 2011.

1155 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, 2016.

López López, P., Sutanudjaja, E. H., Schellekens, J., Sterk, G., and Bierkens, M. F.: Calibration of a large-scale hydrological model using satellite-based soil moisture and evapotranspiration products, Hydrology and Earth System Sciences, 21, 3125–3144, 2017.

Manfreda, S., Brocca, L., Moramarco, T., Melone, F., and Sheffield, J.: A physically based approach for the estimation of root-zone soil
1160 moisture from surface measurements, Hydrology and Earth System Sciences, 18, 1199–1212, 2014.

Martens, B., Miralles, D. G., Lievens, H., van der Schalie, R., de Jeu, R. A. M., Férnandez-Prieto, D., Beck, H. E., Dorigo, W. A., and Verhoest, N. E. C.: GLEAM v3: satellite-based land evaporation and root-zone soil moisture, Geoscientific Model Development Discussions, pp. 1–36, 2016.

Martin, R. C.: Clean Code: A Handbook of Agile Software Craftsmanship, Prentice Hall, 2009.

1165 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
1170 and terrestrial evaporation, Water Resources Research, 53, 10 017–10 024, 2017a.

McCabe, M. F., Rodell, M., Alsdorf, D. E., Miralles, D. G., Uijlenhoet, R., Wagner, W., Lucieer, A., Houborg, R., Verhoest, N. E., Franz, T. E., et al.: The future of Earth observation in hydrology, Hydrology and earth system sciences, 21, 3879–3914, 2017b.

McCool, M., Robison, A., and Reinders, J.: Structured parallel programming: patterns for efficient computation, Elsevier, 2012.

McCuen, R. H.: Modeling hydrologic change: statistical methods, CRC press, 2016.

1175 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 sensing, 11, 2353, 2019.

Millman, K. J., Jarrod Millman, K., and Pérez, F.: Developing Open-Source Scientific Practice *, 2018.

Modanesi, S., Massari, C., Gruber, A., Lievens, H., Tarpanelli, A., Morbidelli, R., and De Lannoy, G. J.: Optimizing a backscatter forward operator using Sentinel-1 data over irrigated land, Hydrology and Earth System Sciences Discussions, pp. 1–39, 2021.

1180 Montanari, A. and Koutsoyiannis, D.: A blueprint for process-based modeling of uncertain hydrological systems, Water Resour. Res., 48, 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
1185 decision making in Earth system models–an overview of techniques and approaches, Earth System Dynamics, 8, 977–1007, 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, 2018.

Nativi, S. and Bigagli, L.: Discovery, Mediation, and Access Services for Earth Observation Data, IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 2, 233–240, 2009.

1190 Nativi, S., Craglia, M., and Pearlman, J.: Earth Science Infrastructures Interoperability: The Brokering Approach, IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 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 Sensing, 13, 2119, 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
1195 science play in the age of machine learning?, Water Resources Research, 57, e2020WR028 091, 2021.

Nedovic-Budic, Z., Crompvoets, J., and Georgiadou, Y.: Spatial Data Infrastructures in Context: North and South, CRC Press, 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, 2011.

Nguyen, G., Dlugolinsky, S., Bobák, M., Tran, V., García, Á. L., Heredia, I., Malík, P., and Hluchỳ, L.: Machine learning and deep learning
1200 frameworks and libraries for large-scale data mining: a survey, Artificial Intelligence Review, 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, 2013.

1205 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, 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,
1210 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, 2013a.

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, 2013b.

1215 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 TOPMODEL, 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, 2007a.

1220 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, 2007b.

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, Environmental Modelling & Software, 79, 214–232, 2016.

1225 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, Journal of Geophysical Research: Biogeosciences, 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, 2021.

1230 Rädle, R., Nouwens, M., Antonsen, K., Eagan, J. R., and Klokmose, 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, 2017.

Rahman, J. M., Seaton, S. P., Perraud, J. M., Hotham, H., Verrelli, D. I., Coleman, J. R., and Others: It's TIME for a new environmental modelling framework, in: MODSIM 2003 International Congress on Modelling and Simulation, vol. 4, pp. 1727–1732, 2003.

1235 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, 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, Environmental modelling & software, 22, 1543–1556, 2007.

Refsgaard, J. C., Storm, B., and Clausen, T.: Système Hydrologique Europeén (SHE): review and perspectives after 30 years development in
1240 distributed physically-based hydrological modelling, Hydrology Research, 41, 355, 2010.

Reichle, R. H.: Data assimilation methods in the Earth sciences, Advances in water resources, 31, 1411–1418, 2008.

Reichstein, M., Camps-Valls, G., Stevens, B., Jung, M., Denzler, J., Carvalhais, N., et al.: 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.

1245 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, 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.

1250 Riquelme, J. L. and Gjorgjieva, J.: Towards readable code in neuroscience, Nature Reviews Neuroscience, 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, 2006.

Rodríguez-Iturbe, I. and Rinaldo, A.: Fractal River Basins: Chance and Self-Organization, Cambridge University Press, 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, 10 012–10 025, 2019.

Rouson, D., Xia, J., and Xu, X.: Scientific Software Design: The Object-Oriented Way, The object-oriented way, Cambridge University Press, Cambridge, England, 2014.

Savenije, H. H. G. and Hrachowitz, M.: HESS Opinions Catchments as meta-organisms – a new blueprint for hydrological modelling, 2017.

Semeraro, C., Lezoche, M., Panetto, H., and Dassisti, M.: Digital twin paradigm: A systematic literature review, Computers in Industry, 130, 103 469, 2021.

Serafin, F.: Enabling modeling framework with surrogate modeling capabilities and complex networks, Ph.D. thesis, University of Trento, 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, Environmental Modelling & Software, 146, 105 231, 2021.

Shen, C., Laloy, E., Elshorbagy, A., Albert, A., Bales, J., Chang, F.-J., Ganguly, S., Hsu, K.-L., Kifer, D., Fang, Z., et al.: HESS Opinions: Incubating deep-learning-powered hydrologic science advances as a community, Hydrology and Earth System Sciences, 22, 5639–5656, 2018.

Stacke, T. and Hagemann, S.: HydroPy (v1. 0): A new global hydrology model written in Python, Geoscientific Model Development Discussions, pp. 1–28, 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, Hydrology and Earth System Sciences, 11, 468–482, 2007.

Trambauer, P., Dutra, E., Maskey, S., Werner, M., Pappenberger, F., Van Beek, L., and Uhlenbrook, S.: Comparison of different evaporation estimates over the African continent, Hydrology and Earth System Sciences, 18, 193–212, 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, GMDD, 2021.

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, Environmental Modelling & Software, 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 resources research, 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 Journal, 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 resources research, 39, 2003.

Wada, Y., Bierkens, M. F., Roo, A. d., Dirmeyer, P. A., Famiglietti, J. S., Hanasaki, N., Konar, M., Liu, J., Müller Schmied, H., Oki, T., et al.: Human–water interface in hydrological modelling: current status and future directions, Hydrology and Earth System Sciences, 21, 4169–4193, 2017.

Werner, M., Schellekens, J., Gijsbers, P., van Dijk, M., van den Akker, O., and Heynert, K.: The Delft-FEWS flow forecasting system, Environmental Modelling & Software, 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, 54, 2011.

Xie, Y.: knitr: A general-purpose package for dynamic report generation in R, R package version, 2013.

Yeh, W. W.-G.: Review of parameter identification procedures in groundwater hydrology: The inverse problem, Water resources research, 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, 2010.

Table A1. This table lists the main acronyms used in the paper.

Acronymn	Meaning
ANN	Artificial Neural Networks
CEOS	Committee on Earth Observation Satellites
CNN	Convolutional Neural Networks
CSIP	Cloud Services Innovation Platform
DA	Data assimilation
DAG	Directly Acyclic Graphs
DARTH	Digital eArth Twin Hydrology
DE	Domain Earth
DSL	Domain Specific Language
DT	Digital Twin
DTE	Digital Twin Earth
EO	Earth Observations
EPN	Extending Petri Net
GEMS	Global Water Monitoring System
GEOSS	Global Earth Observation System
GRDC	Global Runoff Data Center
HDS	Reservoir Type models
HM	Hydrological Modelling
IEM	Integrated Environmental Modeling
IJDE	International Journal of Digital Earth
ISDE	International Society for Digital Earth
MaaA	Model-as-a-Application
MaaC	Model-as-a-Commodity
MaaR	Model-as-a-Resource
MaaS	Model-as-a-Service
MaaT	Model-as-a-Tool
MBC	Modelling-by-component
ML	Machine learning
OMS	Object Modelling System
PB	Process Based Models
RNN	Recurrent Neural Networks
SDI	Spatial Data Infrastructure
SM	Statistical models
SOA	Serve-oriented-architectures