A Framework for the Dissemination of Hydrological Models for Non-Expert Users

Timo Schaffhauser[a], Daniel Garijo[b,g], Maximiliano Osorio[b], Daniel Bittner[c,f], Suzanne Pierce[d], Hernán Vargas[b], Markus Disse[a] and Yolanda Gil[b,e]

[a] School of Engineering and Design, Technical University of Munich, Munich, Germany
[b] Information Sciences Institute, University of Southern California, Marina del Rey, California, United States
[c] Erftverband, Department for River Basin Management, Bergheim, Germany
[d] Texas Advanced Computing Center and Jackson School of Geosciences, The University of Texas, Austin, Texas, United States
[e] Department of Computer Science, University of Southern California, Los Angeles, California, United States
[f] Institute of Groundwater Management (IGW), Technical University of Dresden, Dresden, Germany
[g] Ontology Engineering Group, Universidad Politécnica de Madrid, Madrid, Spain

ARTICLE INFO

Keywords:
software metadata
model metadata
model encapsulation
model catalogs
MINT
hydrological models

ABSTRACT

Hydrological models are essential in water resources management, but the expertise required to operate them often exceeds that of potential stakeholders. We present an approach that facilitates the dissemination of hydrological models, and its implementation in the Model INTegration (MINT) framework. Our approach follows principles from software engineering to create software components that reveal only selected functionality of models which is of interest to users while abstracting from implementation complexity, and to generate metadata for the model components. This methodology makes the models more findable, accessible, interoperable, and reusable in support of FAIR principles. We showcase our methodology and its implementation in MINT using two case studies. We illustrate how the models SWAT and MODFLOW are turned into software components by hydrology experts, and how users without hydrology expertise can find, adapt, and execute them. The two models differ in terms of represented processes and in model design and structure. Our approach also benefits expert modelers, by simplifying model sharing and the execution of model ensembles. MINT is a general modeling framework that uses artificial intelligence techniques to assist users, and is released as open-source software.

ORCID(s):

Highlights

- An approach that facilitates hydrological model dissemination from expert modelers to non-experts
- Software engineering methods are proposed to simplify model complexity by creating software components
- Non-experts can easily modify selected parameters and execute models provided by experts
- Our approach makes models more findable, accessible, interoperable, and reusable in support of FAIR principles
- Various applications benefited from this approach within the MINT framework

1. Introduction

Hydrological models (HMs) are commonly used for water resources management and are mainly developed and used by expert researchers or engineers working in the water sector. The results of HMs are important and considered in decision-making processes of government agencies (Ruiz-Ortiz et al., 2019; Andreu et al., 1996). HM applications include estimation of water availability (Döll et al., 2003), development of water management strategies (Haasnoot et al., 2011), flood risk assessment (Merz et al., 2010), climate impact analysis (Krysanova and Hattermann, 2017; Lobanova et al., 2018; Hattermann et al., 2018), solute transport (Konikow, 2010; Morales et al., 2010) and spatial characterization of hydrological system variables such as soil water content (Brocca et al., 2017), desalination and industrial wastewater treatment (Panagopoulos, 2022) as well as groundwater heads (Reinecke et al., 2019). HMs vary widely in terms of their mathematical description of prevalent hydrological processes and their spatial model structure, ranging from lumped conceptual models (Bittner et al., 2018; Booij and Krol, 2010) to distributed physical models (Brunner and Simmons, 2012; Newman et al., 2017).

A fundamental understanding of hydrological processes is needed in order to reasonably set up a hydrological model for a new region or modeling problem. This may become an obstacle for the use of HMs by decision-makers and other users (Lüke and Hack, 2018). In practice, model results are presented to decision-makers as a summary focusing only on a few specific variables of interest, such as streamflow or groundwater heads. The interests and requirements of decision-makers and various stakeholders can diverge widely from what may be hydrologically interesting. Decision makers in water resources management are usually interested in the assessment of the water balance, primarily the availability of water in space and time. HMs allow a holistic view on the components of the water cycle, from which insightful information, e.g. limiting factors in space and/or time, can be derived. These variables do not necessarily be restricted to water availability, but could also refer to evapotranspiration, soil water or precipitation. Miscommunication between science and non-expert groups is therefore not a rarity (Timmerman and Langaas, 2005). This increases the *"science-policy gap"* due to differences in the level of knowledge between the information producer and receiver (Bernstein et al., 1993; Bradshaw and Borchers, 2000). Consequently, it is a challenging task for modelers to provide information that is practically usable and interpretable by a broader community of end users (Fatichi et al., 2016).

Ideally, HMs would be accessible to any potential users so that they are able to test different decisions and scenarios themselves. Potential users who are not hydrology experts can include data analysts, decision-makers, and also scientists in other disciplines who aim to incorporate water-related topics into their models. In situations where different disciplines need to work closely together, and where models from different areas such as economics, hydrology, climatology or ecology may need to be integrated, further obstacles often emerge, as HMs often need to be designed, exchanged and run by different user groups. Moreover, several models with overlapping features may be available, and selecting an appropriate model for a task can be challenging even for experienced modelers (Surfleet et al., 2012). In addition, enabling different capabilities of a model can lead to different data and input requirements.

Even for hydrology experts, it can be difficult to understand how processes are represented in different HMs, making comparison studies very time-consuming. HMs tend to have special computational requirements and use heterogenous file formats for spatio-temporal data, so that data pre-processing usually requires basic programming skills. Additional technical challenges arise when HMs require different operating systems or complex model configurations, which can limit the applicability and transferability of models even for hydrology experts. Therefore, there is a great need for new approaches to facilitate the dissemination of HMs to users who lack the expertise to develop them but are invested in using them for decision-making purposes.

Over the last few decades, efforts have been made to make HMs more accessible by integrating them into Geographic Information Systems (GIS) (Bittner et al., 2020; Rossetto et al., 2018; Refsgaard et al., 2010). In this regard, GIS-based interfaces to HMs often act as an essential component of a Decision Support System (DSS) (Lautenbach et al., 2009; Pezij et al., 2019; Zhang et al., 2014).

Executable and well-structured DSSs make HMs even applicable by non-expert groups, but DSSs usually lack transferability as they are strongly tailored to the individual conditions of a defined case study. An example of how DSSs are often developed in the course of a project to combine different stand-alone software tools can be found in (Kinzelbach et al., 2021). However, a limitation of many DSS is that they are desktop-based and therefore show limited accessibility. Moreover, they often focus on one area, such as groundwater and even on one model and are thus lacking interoperability. GIS-based interfaces have been used in the Soil Water Assessment Tool (SWAT) (Arnold et al., 1998), the Free and open source software tools WATer resources management system (FREEWAT) (Koltsida and Kallioras, 2019) or the Hydrologic Engineering Center - Hydrologic Modeling System (HEC-HMS) (US Army Corps of Engineers, 2000), but models must be set up from scratch by experienced users. Furthermore, these platforms only include a single HM, while users often want to use several HMs to compare their results. Consequently, initiatives like the Community Surface Dynamics Modeling System (CSDMS) (Peckham and Syvitski, 2007; Peckham et al., 2013), the Earth System Modeling Framework (ESMF) (Hill et al., 2004) or the HydroShare platform (Horsburgh et al., 2016) have already taken a step forward to provide and combine multiple models from different disciplines. CSDMS and ESMF include the dissemination of final and calibrated models combined with their results from a variety of disciplines in the field of Geo- and Earth Science (Overeem et al., 2013; Collins et al., 2005; Keller et al., 2014), while HydroShare is explicitly designed for the exchange, storage or management of hydrological datasets and models (Gan et al., 2020). However, these efforts are focused on users who are modeling experts pursuing science research, rather than non-expert users.

In order to ease the dissemination of expert models to non-experts, our previous work introduced the Model INTegration Framework (MINT) (Gil et al., 2018, 2021). MINT defined the components and interfaces needed to assist expert modelers when setting up pre-existing HMs for non-experts. But adding new HMs to the framework required advanced software engineering skills, making it challenging for expert users to contribute. This paper builds on our previous work, with the following novel contributions:

1. A methodology that follows principles of software engineering to create software components for HMs with a simple invocation function with pre-set inputs and parameters, capturing metadata about the model that can be used to provide guidance to non-expert users.
2. An implementation of this methodology that guides expert modelers to create model components, integrated in the Model INTegration Framework (MINT) (Gil et al., 2018, 2021).
3. Two use cases that demonstrate the use of this methodology and implementation for two models that differ in terms of hydrological processes they consider, as well as in terms of their individual code structure: SWAT (Arnold et al., 1998) and MODFLOW (Harbaugh, 2005).

This methodology makes models more findable, accessible, interoperable, and reusable in support of FAIR principles (Wilkinson et al., 2016).

The paper begins with a description of our proposed methodology for creating software components for models (Section 2). Next, in Section 4, we illustrate how the methodology is implemented in the MINT Model Insertion Checker, a standalone application designed to guide users through the proposed methodology steps. Section 5 describes two examples that follow our methodology to deliver two different HM configurations for two different regions of the globe. Section 6 shows how each of these configurations can be accessed and run in the MINT platform. Section 7 discusses the main advantages and limitations of our approach, and Section 8 presents conclusions and future work.

2. Background

HMs differ in the way they conceptualize the characteristics and flow processes in a natural system. As a result, HMs usually have dozens of parameters and input files which vary across different scenarios. For example, models like SWAT may use an input file with snowmelt observations in regions with mountains but may not take snowmelt into consideration if there are no mountains around the basin of interest. Expert hydrologists, who we will refer to here as *modelers*, need to make decisions about which hydrological processes and corresponding parameters are relevant to the intended non-expert users (e.g., decision-makers, analysts, researchers with expertise in other areas or domains, students in training or citizens who are active in non-governmental organization), who we will refer to as *users*.

2.1. Software Components

Encapsulating software into portable components allows other users to easily run software on their own machine without worrying about the environment and set up needed (Boettiger, 2015; Kurtzer et al., 2017) Following wellestablished component-based software engineering principles, we aim to create self-contained software components that only reveal functionality that is of interest to third-party users. This is important because scientific software components are often implemented in large packages or libraries that can be used for various steps such as data preparation and visualization in addition to writing software to simulate specific processes (such as atmospheric dynamics for climate models, runoff and infiltration for hydrology models, fuel density for fire modeling, etc.).

Software packages can be quite overwhelming for users, even when they are familiar with the scientific domain for which the package was written. Usability becomes even more challenging for users outside of the domain, although these users are precisely the ones who may benefit the most from the results of the respective packages.

Existing graphical user interfaces (GUIs) and GIS systems are often often difficult to reuse from other programs. User interfaces usually have a specific function to call the software with a button, using a form which users operate to define specific parameters (typically the most relevant ones). That function call (sometimes called a command line invocation) is reusable from different programs, provided that the software tool can be run from a machine with its specific execution environment. The function call uses inputs that can be provided when invoking the software component (as it is done in a user interface where the values for some input parameters are set). Other inputs can be pre-set within the component (including data files) if there are no reasons for third party users to change them given a specific use case.

A software component corresponds to a single invocation function for software. Given a sophisticated software package with multiple purposes, a software component may be created to include only certain processes and variables, a specific pre-processing step, or a specific visualization. For example, a hydrology model software may be pre-set to be applicable to hot arid regions only and ignore the processes (and therefore inputs) describing snowmelt.

2.2. Model Configurations & Encapsulation

We use the term *model encapsulation* to refer to the process of creating easy-to-use self-standing executable software components from models for a target scenario. We refer to these software components as *model configurations*. Expert modelers are responsible for designing these model configurations for a region by identifying the key parameters that non-experts should be able to modify. Model configurations declare only those relevant parameters or input files that users should be able to change so the model configuration can be easily set up and run to explore different scenarios.

The remainder of this section illustrates model configurations through a simple example, introduces key concepts in model encapsulation, and describes the main steps of our methodology.

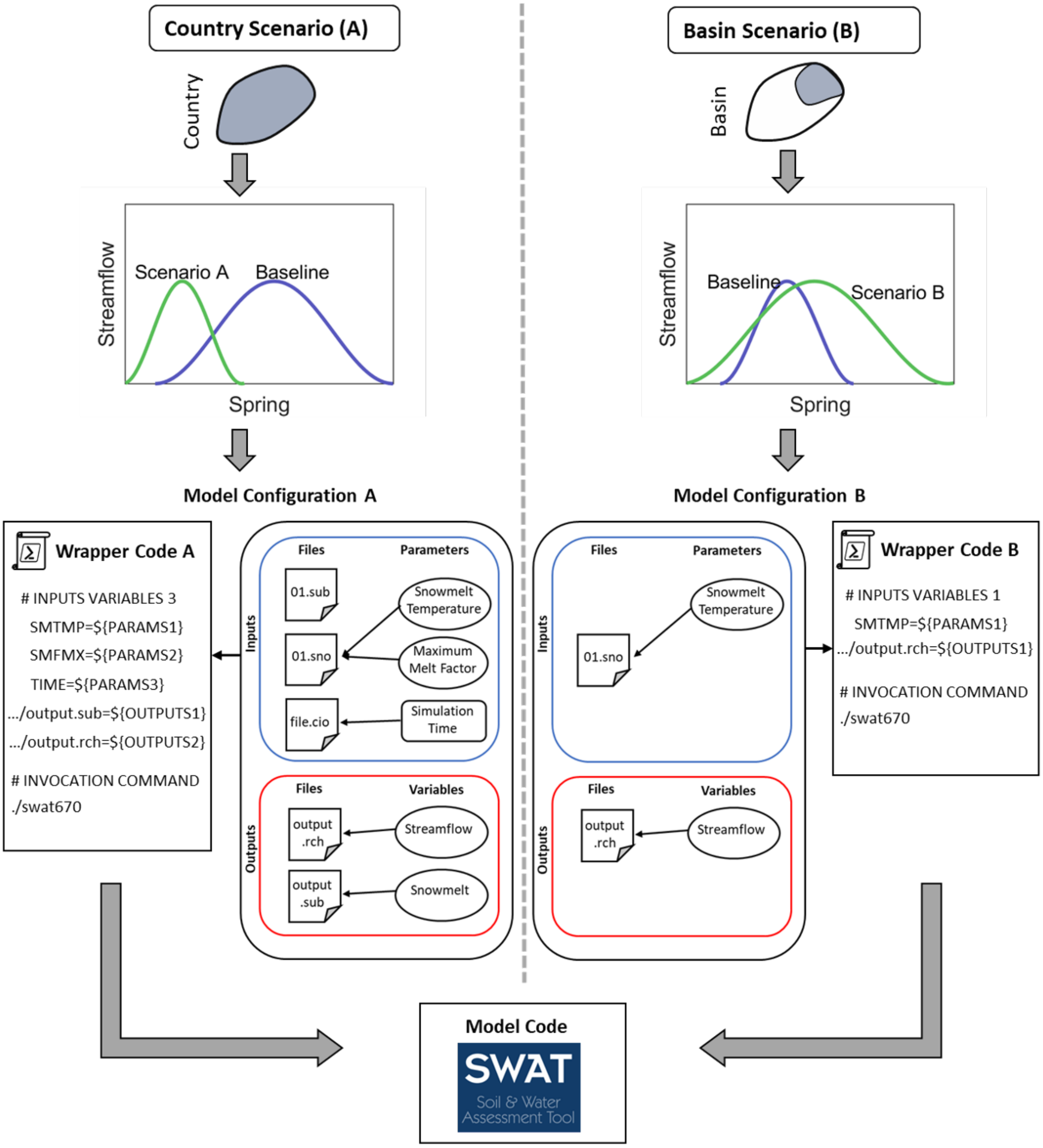

Figure 1: Overview of two model configurations. The first configuration (left) exposes snow (*01.sno*) and topographic input files (*01.sub*) associated with spring melt. Two specific parameters of the *01.sno* file are exposed, namely snowmelt temperature and maximum melt factor, as well as the simulation time as general boundary conditions (stored in *file.cio*). Additional exposed files include *output.rch* file storing streamflow, as well as the *output.sub* file to make snowmelt accessible. The second configuration (right) focuses on a smaller basin inside the country, and it is limited to discharge. The simulation time and the maximum melt factor are pre-set with a meaningful default value.

2.3. Model Configurations: An Example

For illustration purposes, let us consider Alice, an expert hydrology modeler, and Bob, a decision-maker with little hydrological expertise. Bob needs to regulate policies for the water budget at a country scale, and therefore he is interested in obtaining a rough estimate on water availability during the Spring season. In particular, Bob seeks to understand: 1) whether the water demand of specific crops can be met under different assumptions and 2) the impact of runoff for energy production, i.e., from hydro-electrical plants. Given her expertise, Bob asks Alice to provide an environment where he can run model simulations according to his requirements.

Alice anticipates that Bob may want to modify some of the simulation parameters affecting snowmelt, the dominant runoff component and source of water in Spring. A shift in the onset and duration of snowmelt usually affects the temporal water availability of the agricultural and energy sector. Alice decides to use the SWAT model and creates two model configurations to predict streamflow as a proxy representation for water availability. The first model configuration is designed at the country level, letting Bob modify the snowmelt temperature and the maximum possible snowmelt to explore the effects on agriculture (e.g., what crop yield can be achieved by different crops). The second model configuration focuses on a small basin located in the Northeastern region of the country, in order to study the conditions and effects of snowmelt for a potential small hydroelectric power plant. Both model configurations first undergo a strict and rigorous calibration and validation procedure by Alice, a necessary expert step to ensure a reliable baseline for the further usage. The calibration and validation serve as fundamental steps to provide robust and credible models.

Figure 1 shows an overview of the model configurations prepared by Alice, with the country-level configuration on the left and the basin configuration on the right. Each configuration has one or multiple *inputs* and *outputs*, representing the files accepted and produced by a configuration. We use the term *parameters* to refer to values a user may be interested in changing in a model, such as snowmelt temperature, even if these values are declared within configuration files. We consider as parameters hydrological or process-based variables, together with temporal information such as simulation length or time step, here referred to as boundary conditions (BC). A *code wrapper* captures how to invoke a model configuration by indicating how the command line should be invoked, and specifies any fixed values of inputs.

When creating a model configuration, a modeler like Alice may have to choose which of the inputs or parameters should be adjustable by the final user, among the dozens or hundreds of input files and parameters HMs have. We use the term *expose* to indicate that a parameter or input file can be adjustable by a user in a model configuration. For example, SWAT contains hundreds of files, but Alice estimates that the relevant ones for the country-level configuration are two input files with snow and elevation information. As shown in Figure 1, the input file containing snow information further includes the parameters that will be exposed to users, namely snowmelt temperature and the maximum melt factor of snow. Adapting the threshold temperature when snow begins to melt is an easy way to shift the melt season within the country. The second parameter provides information on the amount of snowmelt one could expect. Alice thus provides a meaningful range of values, within which Bob is able to increase or decrease the amount of snowmelt. In addition, Alice decides to expose a file that includes general information on BC like the time for which the model was set up or its temporal resolution, daily in this case. Thanks to this information, Alice expects Bob to be able to

Table 1
Overview of terminology used that has a specific definition in that paper, but might have an ambiguous use outside our work.

Term	Description
Model encapsulation	Process of creating easy-to-use and independent software components (e.g. from a model)
Model configuration	Abstracted version of a model which considers only relevant inputs, outputs and parameters that are adjustable. Model configurations represent software components
Boundary condition	General information of a model such as temporal information
Parameter	Hydrological or process-based variable where users might be interested in to change
Expose	Indicates that a specific parameter or file is adjustable by the user
Expert	Hydrology expert used to modelling
User	A non-expert in the field of modelling, such as for example citizens, decision makers, researchers from other fields, analysts
Wrapper	Captures how to invoke a model configuration and specifies fixed values

compare the effects of a very high and a very low value for snowmelt temperature as well as the maximum melt factor on the water availability.

As for the basin configuration, Alice is familiar with the area from her previous work. Therefore, she decides to set up all default values of the model according to her knowledge of the region. She *exposes* snowmelt temperature by making only this parameter available in the basin configuration. This configuration is more restricted, but more precisely tailored to the region at hand. Therefore, this model configuration is simplified by allowing Bob to only modify snowmelt temperature. Hence, Bob can now obtain alternative estimates with respect to the accumulation of snow during winter, which is then available as melt water. This enables the decision-maker to infer whether a small hydropower plant might be of value or not or how much energy could be produced under various snowfall conditions.

In summary, with these model configurations the modeling expert is able to hide the complexity of a general model exposing only what is relevant for a country and its hydrology, narrowing it down to a much more usable model component for other users to explore scenarios and make decisions accordingly. It should also be mentioned that Bob doesn't necessarily must be a decision maker. However, he could also be an interested member of a NGO which deals with environmental issues for example or just an interested citizen increasingly affected by hydrological events such as drought or heavy rain.

Table 1 provides an overview about the terminology we use in this paper, especially to distinguish terms which might ambiguous and are used differently in other fields.

Table 2
Overview of the main steps of our proposed model encapsulation methodology.

	Description	Result
Step 1: Start a New Environment	• Modeler indicates a working folder (it may be empty) • System prepares a basic execution environment (e.g., Unix and Python)	The system populates the component folder structure, including a setup file containing information on the target model component and creating an empty software container
Step 2: Trace Execution Dependencies	• Modeler runs a test execution • System detects dependencies to execute the model and adds them to the container	Container that includes execution dependencies for the model run
Step 3: Expose Parameters	• Modeler indicates user-adjustable parameters to be exposed • Modeler specifies default values • System stores parameter exposure and links to configuration files	File containing parameter information
Step 4a: Expose Input Files	• Modeler indicates input file types expected by the configuration	File containing the input file selection
Step 4b: Expose Output Files	• Modeler indicates output file types produced by the configuration	File containing the output file selection
Step 5: Create Wrapper Script	• Modeler reviews the execution shell script created by the system to run the new model component according to the specified settings, and does a test run of the component • System ensures that the test run completes successfully, and uses the provided input/output description, parameter settings and shell script to create the model component as a container with the required dependencies.	Creation of subfolders and files with encapsulation and execution information
Step 6: Model Upload	• System uploads the model component	Registration of model component in container and code repositories and model catalog

3. A Methodology for Model Encapsulation

We propose a methodology for creating model configurations. Our methodology requires expert modelers to determine the main parameters and input files that need to be exposed for a given executable model, including steps for guiding and testing the final model configuration so other users can use it effectively. Our methodology comprises six main steps: *Start a New Environment*, *Trace Execution Dependencies*, *Expose Parameters*, *Expose Input & Output Files*, *Wrap Execution*, and *Model Upload*. Table 2 provides a summary of all steps, which are further described here.

Step 1: Start a New Environment: Modelers start by specifying the location for the folder structure of the new model component they want to create. This should be started in a "clean" computing environment, free from other software dependencies installed on the local machine. For example, if a model is available in Python, starting in a clean environment makes it easy to isolate the model needs from other Python libraries installed in the machine for other purposes. This can be achieved by using virtual environments, that create a clean Python installation with no installed package dependencies. In our methodology we adopt software containers, a common approach to capture computational environments. Software containers enable capturing the dependencies of a software component at the operating system level (i.e., including not only the dependencies of a software component, but all the system dependencies as well), hence ensuring that it can be run in other environments.

Because containers can be complicated to set up and use for non-computer scientists, our system will be automatically creating the container and installing the dependencies and files needed to run the model. The modeler can see everything that the system is adding in the folder that they specified.

Step 2: Trace Execution Dependencies and Run Model: Once an environment has been set up, the dependencies needed to install the model must be incorporated into the environment. This includes compilers, system libraries, and other files. The modeler carries out a test run that is representative of how the model configuration will be used. During the run, the system automatically detects the model input, configuration, and output files used by the model during the run. This information is added to the container environment and used by the system in subsequent steps in order to assist the modeler to specify inputs and outputs.

(a)

```
Basin data          .bsn file 1/28/2021 12:00:00 AM ArcSWAT 2012.10_0.14
Modeling Options: Land Area
Water Balance:
           1.000    | SFTMP : Snowfall temperature [ºC]
           0.500    | SMTMP : Snow melt base temperature [ºC]
           4.500    | SMFMX : Melt factor for snow on June 21 [mm H2O/ºC-day]
           4.500    | SMFMN : Melt factor for snow on December 21 [mm H2O/ºC-day]
           1.000    | TIMP : Snow pack temperature lag factor
```

(b)

```
Basin data          .bsn file 1/28/2021 12:00:00 AM ArcSWAT 2012.10_0.14
Modeling Options: Land Area
Water Balance:
           1.000    | SFTMP : Snowfall temperature [ºC]
       ${melttemp}  | SMTMP : Snow melt base temperature [ºC]
           4.500    | SMFMX : Melt factor for snow on June 21 [mm H2O/ºC-day]
           4.500    | SMFMN : Melt factor for snow on December 21 [mm H2O/ºC-day]
           1.000    | TIMP : Snow pack temperature lag factor
```

Figure 2: An illustration of how the snowmelt temperature parameter (*SMTMP*) of a SWAT model is exposed so it is accessible for users to adjust for different scenarios: a) shows the original *.bsn* file with the default value assigned to *SMTMP*, b) shows how the default value is exposed so it can be changed by a user when running the model configuration.

Step 3: Expose Model Parameters and Define Configuration Files. Most HMs have dozens of parameters and BCs which specify constants like hydraulic conductivity, bulk density, or the general settings of the simulation. Within this step, modelers have to define which of these parameters and BCs they want to expose to users in the new model configuration. For example, the CN2 (Curve Number II) parameter of SWAT is usually one of the parameters which is typically changed during the model setup and calibration (the process of estimating relevant parameters and their corresponding values) and might be a useful parameter to expose in a model configuration.

HMs usually adjust numerical values for their simulations in two different ways: 1) with the invocation command used to run the model; or 2) through configuration files that can be edited directly or accessed via user interfaces. If a file is used, it needs to be specified by the modeler. Fig. 2 illustrates this with an example of how the snowmelt base temperature parameter is exposed for the SWAT hydrology model (SMTMP) through a configuration file.

Step 4: Expose Model Inputs and Outputs. Next, modelers have to decide which input and output files they want to expose, which depends on the intended use cases that users will want to simulate. As with parameters and BCs, expert modelers usually provide the relevant input files required by a model. Likewise, models produce all sorts of output variables, and for a given configuration only a certain subset of outputs may be relevant for the intended use cases. For instance, a modeler may expose only output files containing drought-related variables such as evapotranspiration and soil moisture.

Step 5: Create a Wrapper Script. Once the parameters, BCs and files to be exposed have been specified, the next step is to write a shell script which captures how to run the model configuration. We refer to this script as the *wrapper* script, as it *wraps* the model configuration as an executable component. The wrapper script will make sure that the component can run with the inputs and outputs selected by the modeler, and may include pre-set files or values for other inputs and parameters. In order to verify that the model works appropriately with the wrapper script, it is necessary for the modeler to provide *sample input files* which are used in a test run. If everything works successfully, the model configuration is completed and will be executable in other computational environments.

Step 6: Upload the model configuration. The final step is to deposit the model configuration in shared repositories. First, the script and test data used to wrap up the model configuration should be deposited in a code repository. Second, an archival version of the model software code must be created in a code repository, to ensure that that version can always be accessed by users in the future. Third, the container environment should be uploaded to a container registry. Finally, the model configuration should be uploaded to a model catalog, with proper *model configuration metadata* provided by the modeler to enable discovery and reuse.

4. Methodology Implementation

We implemented our methodology in the MINT Model Insertion Checker (MIC), a standalone application developed to guide users through the process of creating new model configurations. MIC performs all the steps of our

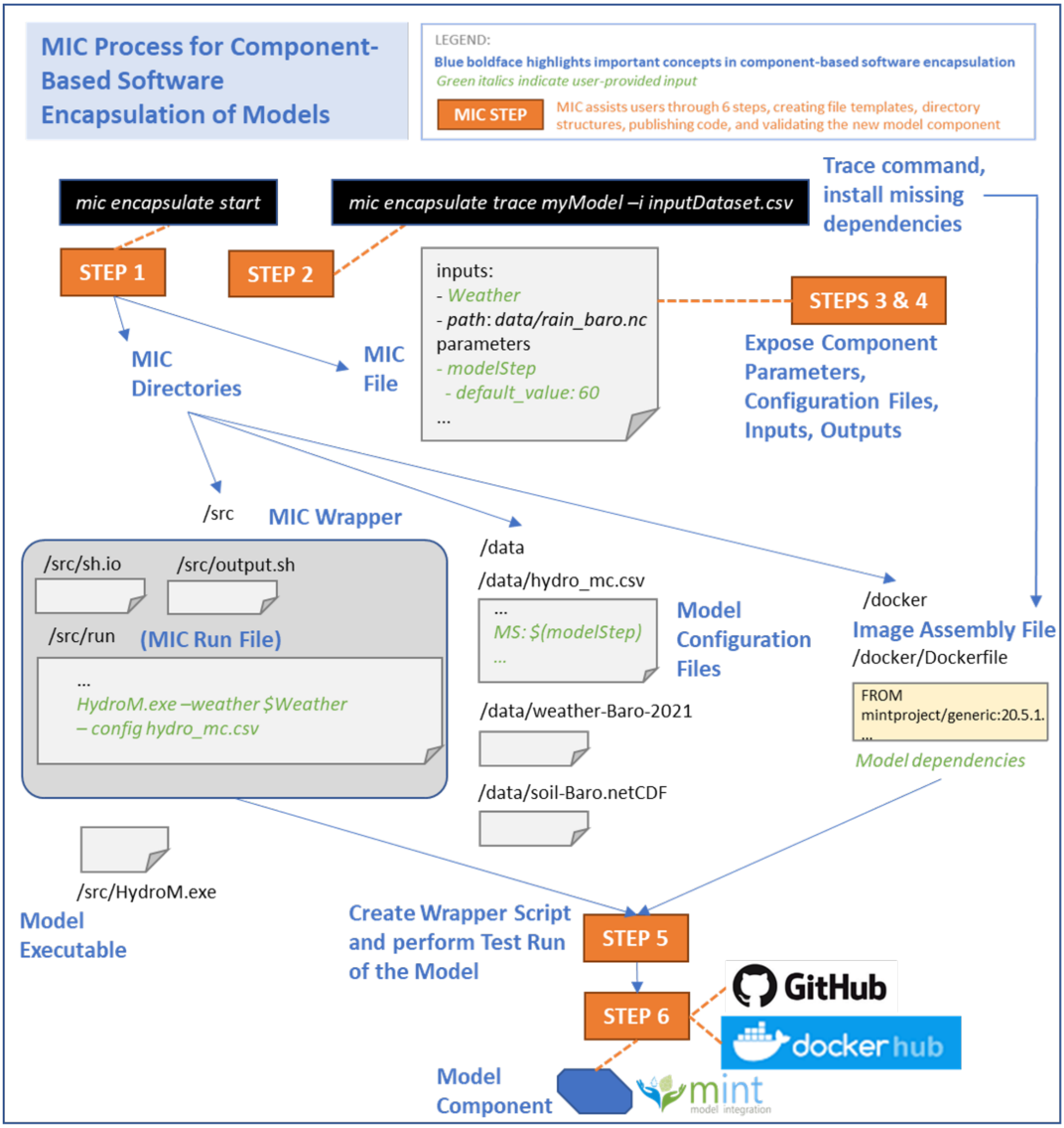

Figure 3: Overview of the methodology steps as well as the resources created by each step.

methodology in a semi-automated manner, integrating the results with commonly used software and container image repositories such as GitHub and DockerHub. MIC also integrates new model configurations and the metadata in the MINT modeling framework and its model catalog (Garijo et al., 2019). MIC is implemented as a Unix-based tool that runs in the command line, and is available as open source software (Osorio et al., 2022).[1] A step by step tutorial is

[1]Software respository is at `https://github.com/mintproject/mic/` with documentation at (`https://mic-cli.readthedocs.io/en/latest/overview/`)

available online [2] to help users and disseminate the steps of our methodology.

Figure 3 provides an overview on how MIC implements all the steps of our methodology, capturing the main software dependencies, input, parameters and generated files, and showing how the methodology steps are related to one another. MIC guides users through the six steps outlined in our methodology. It starts with a blank Unix environment, generated with a basic Docker image, where users are asked to install and run their model from scratch (*step 1*). Once a sample run is finished, MIC tracks which files have been used and generated using ReproZip (Rampin et al., 2016), an application designed to trace all dependencies and system calls of a program (*step 2*). Using the output from ReproZip, MIC drafts an initial component, based on the inputs and outputs detected in the test run. Next, MIC works with the modeler to get information about the inputs, parameters and BCs of HMs should be exposed in the model configuration, among all the candidates detected automatically (*steps 3 and 4*). The preparation of the configuration file is one of the few activities that has to be carried out manually by the modeler, as it involves information highly dependent on the use cases required by the intended users. For example, SWAT may be used to create multiple configurations depending on whether modelers need to expose snowmelt temperature, hydraulic conductivity or a factor to delay groundwater flow. The parameters and BCs exposed with MIC will be adjustable by users when running the model configuration. If one of the exposed parameters or BCs are stored in a configuration file, an additional step is required to indicate where to replace the target value in that file. An example can be seen in Figure 2, where snowmelt temperature in SWAT is exposed through the SMTMP parameter which can be provided by users at runtime. All the information provided to MIC is stored in a *MIC settings file* that can be inspected and edited by modelers at any time, e.g., to change default values for parameters or to make adjustments on what is exposed to users.

Once all the inputs, outputs, parameters and BCs form a specific model configuration are set, MIC will prompt users to perform a test run using all default values. MIC automatically creates an execution wrapper script (*step 5*) and runs the model using the local environment created earlier in the second step. If successful, the model configuration is ready to be run by others, and MIC will prompt users to double check if the results from the execution are correct.

As a final step, MIC saves the model configuration (*step 6*) including:

1. the **computational environment** used in the test run, saved as a Docker image in DockerHub[3]
2. the **wrapper script and settings file** containing the exposed inputs, outputs, parameters and BCs. MIC will store these files in a new GitHub repository, owned by the modeler who created the model configuration
3. **basic metadata** about the model configuration, including its main title, description, version of the model, geographic location, execution details and brief parameter and input descriptions. These metadata are submitted to the MINT model catalog, producing the results shown in Figure 4.

[2] https://mic-cli.readthedocs.io/en/latest/model_configuration/03a-step1/
[3] https://hub.docker.com/

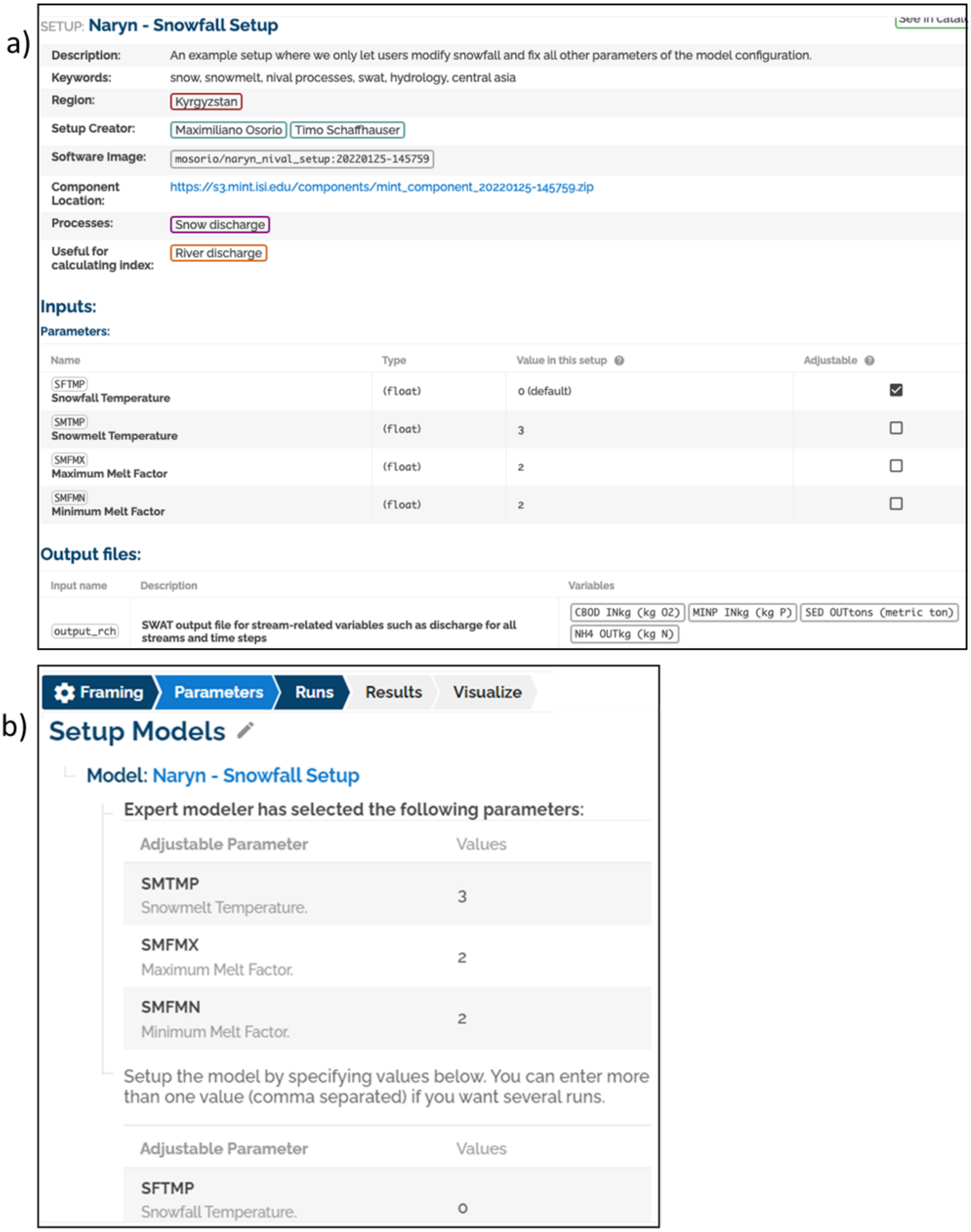

Figure 4: An example of how a model configuration is used by modelers and users: a) the modeler provides all the metadata, including the parameters and files exposed to the user, and specify default values for some of the parameters so end users only need to adjust one of them (snowfall temperature); b) end users wanting to use that model component can specify different values of the parameters and submit model runs that correspond to the scenarios they want to explore.

In the following sections we provide several screenshots of MIC to familiarize the reader and potential users with the platform. Figure 4 shows an example of how the model configuration can be accessed by a user after being created with MIC. Figure 4 a) depicts a model configuration where four parameters are exposed (i.e., minimum and maximum melt factors, snowfall temperature and snowmelt temperature) out of the dozens of parameters that are available in SWAT. Figure 4 b) shows an example where only one of the four parameters (snowfall temperature) may be changed by users when running a second, different configuration of SWAT (the other three parameters are fixed). Both configurations of the model are integrated in the MINT framework, where they can be executed through a GUI.

5. Creating Model Components: Two Practical Use Cases

In this section we showcase our methodology by encapsulating two different and widely used hydrological models, i.e, SWAT and MODFLOW, using MIC to create model components and running them in the MINT platform. By pointing out the specific differences of SWAT and MODFLOW, we illustrate the main concepts of our methodology as well as the technical features of MIC that facilitate model dissemination for any type of HM. We show model configurations for SWAT and MODFLOW for two different case studies. Each case study was defined prior to our work by a different research group working with stakeholders in different regions of the world.

5.1. SWAT: Background and Model Structure

The Soil Water Assessment Tool (SWAT) is a semi-distributed, time-continuous model developed by the Blackland Research & Extension Center of the United States Department for Agriculture (USDA) (Arnold et al., 1998). SWAT is based on the concept of the Hydrologic Response Units (HRU) and was originally developed to assess the impact of land management practices in large watersheds, while the applications nowadays range from water quality or sediment transport studies up to snow-hydrological in basins all over the world Arnold and Fohrer (2005).
HRUs are the smallest spatial unit within the model and defined on the subbasin scale, a further subdivision of the watershed. However, HRUs are not spatially located and are formed by unique combinations of land use, soil and slope within each subbasin to consider spatial heterogeneity. The HM is organized by input files grouped by different processes or characteristics, such as land management or soil inputs, for the individual spatial units. Besides, the model includes few general files where basic settings can be done. SWAT separates its calculations in a land and a water phase. It first calculates all loadings for the HRUs in each subbasin, which are then transferred to the stream. In a second step the in-stream processes, covering routing processes as well as chemical processes, are calculated.

5.2. MODFLOW: Background and Model Structure

The MODular Finite-difference FLOW model (MODFLOW), is a fully-distributed and physically-based groundwater model, developed by the United States Geological Survey (Harbaugh, 2005; Hanson et al., 2014). MODFLOW is organized in modules, which allow for user customization of specific case studies (i.e., by selecting only those modules that are relevant). For instance, a module can represent different solvers for the groundwater flow equation. Moreover, various modules exist to account for different hydrological processes in a natural system, e.g., stream flow, evapotranspiration or groundwater recharge. Given the grid-based nature of the model, several modules can be coupled by providing grid coordinates in the input files. If specific modules should be used in a model run, an input file is required for each respective module. These input files are ASCII files, either organized in a table format or grid-based. All modules to be used for a model simulation have to be included in a configuration file, i.e., a name (*.nam*) file. Depending on the interaction of different hydrological processes, MODFLOW solves the groundwater flow equation and provides water budgets for each pre-defined discrete time step in an output file, the list (*.lst*) file.

5.3. Model Implementation

In the following we describe how our methodology, described in section 3, is implemented for two different HMs, namely SWAT and MODFLOW. Most of the steps are similar for both models (and to other HMs), despite how different their software and approach are. Therefore, we focus on demonstrating how users can describe the models following our methodology using different use cases.

5.3.1. Case Studies

The location of our study areas and their geographical characteristics are illustrated in Figure 5. Our case studies focus on two very distinct hydrological systems: the Naryn River in Kyrgyzstan for SWAT and the Barton Springs segment of the Edwards aquifer in Texas for MODFLOW. For each case study we emphasize which part of the proposed methodology is similar and where differences occur, which mainly concern the exposed inputs and outputs in the respective model configurations. The BCs, such as simulation period and time step, have been set by experts for both case studies. The target user groups of both cases are non-expert analysts and decision-makers. Our intention is to grant the respective users access to the model configurations, so that they are able to run alternative scenarios on their own. A summary of the case studies can be found in Table 3.

5.3.2. Snow dynamics in the Naryn Basin - Case Study

Our first case study focuses on a part of the Naryn Basin located in Kyrgyzstan, where high flow occurs mainly in Spring and Summer due to snow and glacier melt. In contrast, low flow phases are mostly restricted to the winter season. The basin belongs to one of the headwater streams of the Syr Darya, one of the two major tributaries of the Aral Sea and drains an area of around 50,000 km^2. Our case study focuses on the headwaters of the basin, which originate in the Tianshan mountains. Snow and glacier melt are of great concern for the local population, as it provides water for energy and agriculture (Unger-Shayesteh et al., 2013; Gan et al., 2015). The parameters exposed concern

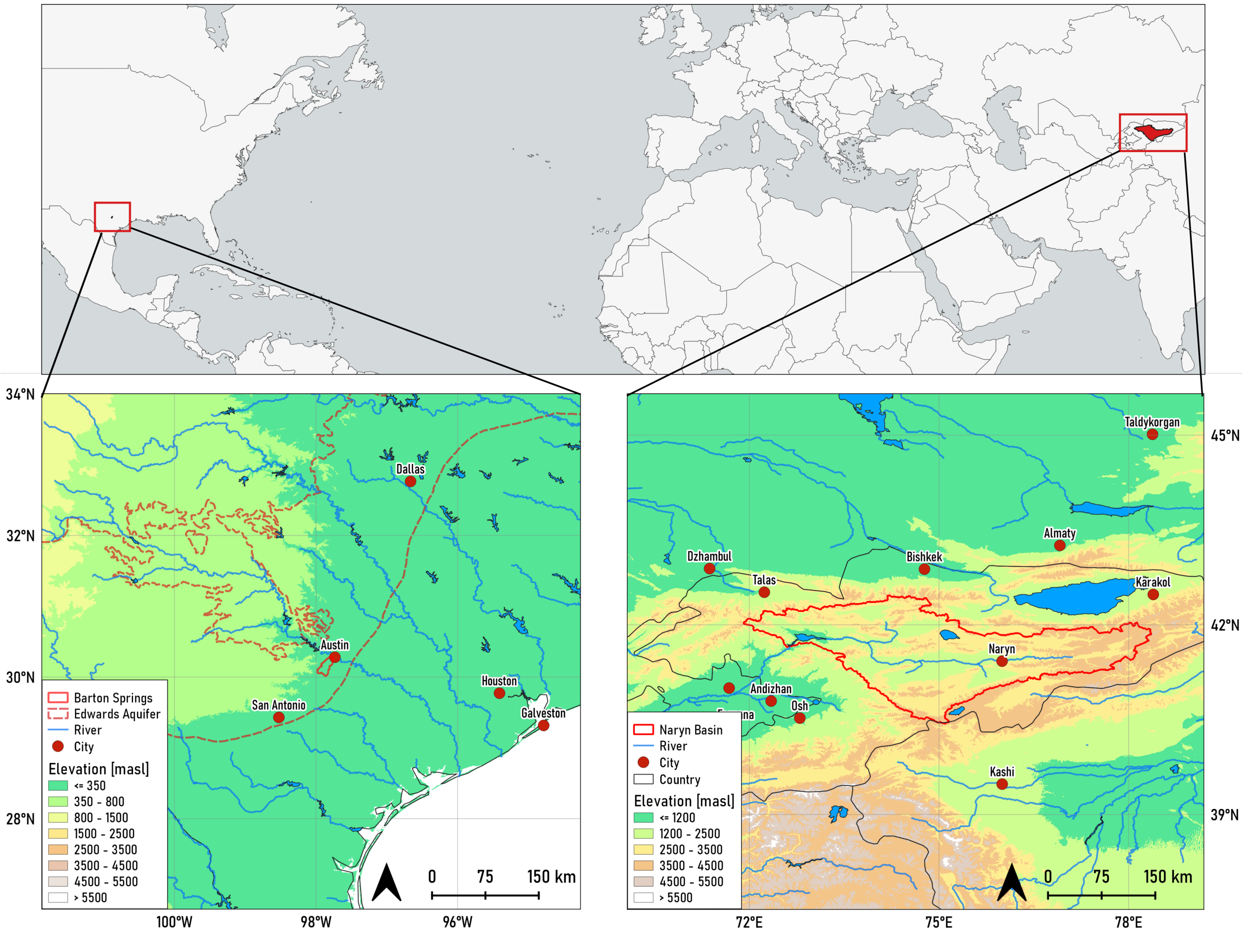

Figure 5: Regions in the two case studies: on the left the Barton Springs Segment of the Edwards Aquifer in Texas for MODFLOW, on the right the Naryn Basin in Kyrgyzstan for SWAT.

snowmelt and snowfall (a full list is provided in Table 3). The choice is based on preliminary investigations that comprised a comprehensive sensitivity analysis and calibration of our SWAT model (Schaffhauser et al., 2023). These parameters (snowfall temperature, snowmelt temperature, maximum and minimum melt rate) proved to be among the most sensitive ones providing a reasonable model performance. The case study represents an example where the model is intended to be used by local authorities. Our configuration provides an example of an abstraction that can be used by both non-experts and more experienced users. In this case, the non-experts will be decision-makers in an agricultural agency, while the more advanced users will be in the local water authority who will have a broader expertise in water-related questions. The model component shall finally be used by these decision-makers to examine the effects of changes in snow processes on streamflow. Snow processes constitute the dominant source of water and serves as a proxy of water availability in spring for the region. An exploration of the timing and amount of snowmelt provides decision-makers with valuable insights on the available water for different sectors, such as agriculture or

Table 3

Summary of the two case studies, describing the main characteristics of the two case studies. For case study 2, the abbreviations M-A, M-B and M-I refer to the three model components based on average conditions with default pumping rates, a baseline considering drought conditions and a component where the user can specify pumping rates and infiltration. Exposed parameters and input files indicate those elements highlighted by the expert modeler for each scenario which can be customized by others.

	Case Study 1	Case Study 2
Hydrological Model	SWAT	MODFLOW
Name of Model Configuration	Naryn - SWAT	Barton Springs - MODFLOW
Region	Naryn Basin, Kyrgyzstan	Edwards Aquifer, Texas, US
Region Size	10,000 km²	401 km²
Scenario Summary	Water resources management, floods, crop yield, energy production	Sustainable yield, drought assessment, evaluation of pumping rates under stress conditions
Dominant Processes	Snow and glacier related	Infiltration, pumping
Exposed Parameters	Snowmelt temperature, snowfall temperature, max. melt factor summer, min. melt factor winter	None
Exposed Input files	basin.bsn	Baseline model (M-B) & model with average conditions (M-A): None, Infiltration model (M-I): infiltration, pumping rates & recharge, wells
Exposed Output Variables	Streamflow	Hydraulic head, total water storage, total volume extracted

energy.

This information is important in many aspects. For example, authorities can deduce how much water is expected to be available for agriculture. This enables an estimate of the expected yield within the crop season, one of the major economic factors for the region. In addition, this water is required to be stored for energy production of the whole country. Besides, the period is prone to floods, frequently causing at least local threats. By having model outputs of water availability, decision-makers can allocate water to different purposes, advance or delay planting dates, and generally prepare for the specific seasonal requirements such as energy or irrigation demand.

Accordingly, we share the model configuration to enable these users to adjust the snow-related parameters, namely snowmelt temperature, snow fall temperature as well as the minimum and maximum snowmelt factors. Users can then explore their own scenarios, and monitor the actual conditions of the basin to assess which of their scenarios correspond to the actual conditions of the current season.

To provide some initial scenarios, we provide a set of default values for all snow parameters to provide users with a starting point. As the response variable of interest for the end-user is discharge, only the corresponding output file is exposed in our component. For simplicity, we decided to predefine all input files so that users cannot make any changes.

5.3.3. Drought impact on the water budget in Barton Springs - Case Study

The second study refers to the Edwards Aquifer and more precisely the Barton Springs segment in Austin, Texas, a region increasingly affected by droughts (Passarello et al., 2012, 2014). A numerical simulation, using the MODFLOW model, was developed for use as a groundwater availability model (GAM) in the state of Texas (Scanlon et al., 2001, 2003).

The MODFLOW configuration was prepared as part of a state-wide planning activity. The components underwent a scientific vetting process to assess groundwater availability. The intended end users are groundwater managers for state-designated management districts, as well as stakeholders involved in the recurring groundwater aquifer management program of the state of Texas. They are not hydrology experts necessarily, although they have expertise in groundwater. Water availability fluctuates rapidly in the region, due to normal variability in weather and climate conditions. As urban areas have expanded in the past decade, water consumption has increased and habitats for vulnerable species are at greater risk for impact during dry conditions. Table 3) shows an overview of the models. We created a model component M-B that reflects a baseline model for drought conditions with default pumping rates. We created a separate component M-A for average conditions, also with default pumping rates. M-B was explicitly designed to investigate and emphasize potential adverse effects of pumping under dry conditions. In contrast, M-A shows the impacts of similar pumping conditions under normal non-drought conditions. We also created a third component M-I where the user can specify infiltrated water (as a recharge input file) and pumping rates (as a wells input file). The components are designed to expose key model outputs concerning water table levels (hds output file representing hydraulic head levels), storage (cbb output file representing volumes), and actual pumping rates (cbb output file).

The recharge zones were developed for Barton Springs GAM because it represents a baseline interpretation of groundwater behavior, the model is readily accessible. The recharge zones were originally completed as part of a Groundwater Decision Support System developed to assess the sustainable yield (Pierce et al., 2006; Pierce, 2006).

5.3.4. Model Encapsulation

The following subsections demonstrate the model encapsulation of each case study. A summarized overview of the steps and the differences in the procedure (where users have to perform manual adaptions) for each case study, is shown in Table 4. The encapsulation process follows the model preparation steps (usually including calibration and validation) which are performed by the expert modeler.

5.3.4.1. STEP 1: Start New Environment An environment has to be created for each model configuration (see Section 2.3). For case study 1, the modeler would create a single model component focused on the snow processes of SWAT. For case study 2, the modeler chose to create three separate model components: one for baseline drought conditions, one for baseline average conditions, and a third one for analyzing different scenarios in average conditions.

Table 4
Overview of the steps conducted in MINT for the dissemination of the two case studies. We highlight where users have to incorporate manual modifications and which explicit setting we made in our example.

	SWAT	MODFLOW
1) Start New environment	no difference except name of the model configuration	no difference except name of the model configuration
2) Trace Execution Dependencies	execution command is model-specific *./swat670*	execution command is model-specific *./mf6*
3) Expose Parameters	MIC command (*mic pkg parameters*), parameters are model and case specific, here: *snowfall temperature, snowmelt temperature, maximum & minimum melt rate*	MIC command (*mic pkg parameters*), parameters are model and case specific, here: none adjustable parameter defined
4a) Expose Input Files	MIC command (*mic pkg inputs*), desired input files to share are model and case specific, in this case *basins.bsn*	MIC command (*mic pkg inputs*), desired input files to share are model and case specific
4b) Expose Output Files	MIC command (*mic pkg outputs*), desired output files to share are model and case specific, in this case *reach.rch*	MIC command (*mic pkg outputs*), desired output files to share are model and case specific, in this case *.hds, .lst*
5) Create Wrapper Script	MIC command (*mic pkg wrapper, mic pkg run*), manual & model-specific adaptions when default parameter changes are desired	MIC command (*mic pkg wrapper, mic pkg run*), manual & model-specific adaptions when default parameter changes are desired
6) Model Upload	MIC command (*mic pkg upload*), automatically uploads the model configuration to DockerHub, GitHub and MINT	MIC command (*mic pkg upload*), automatically uploads the model configuration to DockerHub, GitHub and MINT Model Catalog

The modeler starts MIC from the command line, where he provides the name of the model configuration. In our case, the names are *Naryn - SWAT* and *Barton Springs - MODFLOW 1* to *3*. MIC automatically creates the folder structure for each model configuration.

5.3.4.2. STEP 2: Trace Execution Dependencies The modeler then does a test run to check if the respective model is installed in a new environment and to trace the execution dependencies. Then, MIC is used to trace input and output dependencies (through ReproZip). Since MIC is a Unix-based tool, the invocation command for SWAT refers to the Unix-based execution file, which can be downloaded via the SWAT homepage.[4] As for MODFLOW, we used the Python-based FloPy tool for the model encapsulation.[5] FloPy serves as a tool which is used to execute existing MODFLOW-based models.

5.3.4.3. STEPS 3 & 4: Expose Parameters, Inputs and Outputs For the SWAT model configuration, several snow parameters were exposed, which were snowfall temperature, snowmelt temperature and the maximum and min-

[4] https://swat.tamu.edu/software/swat-executables/
[5] https://www.usgs.gov/software/flopy-python-package-creating-running-and-post-processing-modflow-based-models

imum melt factors. The parameter selection was based on a preliminary study done by the modeler with relevant stakeholders to identify the dominant parameters (see also Table 3). Each parameter exposed must be manually specified in MIC, as described in Section 4. Subsequently, the parameters must be indicated in the corresponding SWAT input files (as shown in Figure 2). Adjustments of default parameter values are possible during this step as well. Next, the modeler declares the input files that contain the exposed parameters as configuration files. Since all snow parameters of SWAT are stored in the basin file (*basin.bsn*), it is the only configuration file relevant to the model configuration. The users in the Naryn case study, such as authorities related to the agricultural, energy or water sector, do not need all the output files so only the *output.rch* file is exposed, as it contains all required information on streamflow within the basin.

For the configurations of the MODFLOW model in case study 2 no parameters were exposed. For the drought model component only the *.hds* and *.lst* input files were exposed, where the relevant information of the hydraulic head and the water budget can be specified by users.

5.3.4.4. STEP 5: Create Wrapper Script MIC helps wrap model configurations by taking into account the execution settings and prepares the files to test the model components. The test runs done by MIC were based on the default parameter settings defined in the previous step and double-checked manually. After the test run, the model configuration was finalized and ready for upload.

5.3.4.5. STEP 6: Model Upload Finally, MIC uploads the model configurations to relevant repositories. The Docker image of the model component was uploaded to DockerHub.[6] [7] A GitHub repository containing the input data and results was also created[8]. Finally, an entry in the MINT model catalog was created,[9] [10] and the model can be easily run from the MINT user interface.

6. Scenario Exploration by Non-Expert Users with New Model Configurations

This section describes how users can access the newly created model configurations of the two case studies. It highlights how users can easily specify simulation scenarios using the model configurations.

6.1. Accessing Model Components

Users can browse all model configurations, for example by bringing up the corresponding regions, Kyrgyzstan and Texas, or browsing entries in the MINT model catalog. Typically, a user starts in the “Use Models" tab, and

[6] https://hub.docker.com/r/mosorio/naryn_nival_setup/tags

[7] https://hub.docker.com/r/mintproject/modflow-2005/tags

[8] Components are archived in Zenodo: https://zenodo.org/record/6948339.Yue0VHZByMq

[9] https://mint.isi.edu/kyrgyzstan/models/explore/SWAT/8cc84426-d849-471b-9a5e-47bcaf094607/6a36a2e5-73bf-4098-9acd-1aaaab383d4a/14580635-c7ca-4256-935a-4ddbdacfbfe2

[10] https://mint.isi.edu/texas/models/explore/MODFLOW/modflow_2005/modflow_2005_cfg/modflow_2005_BartonSprings_avg

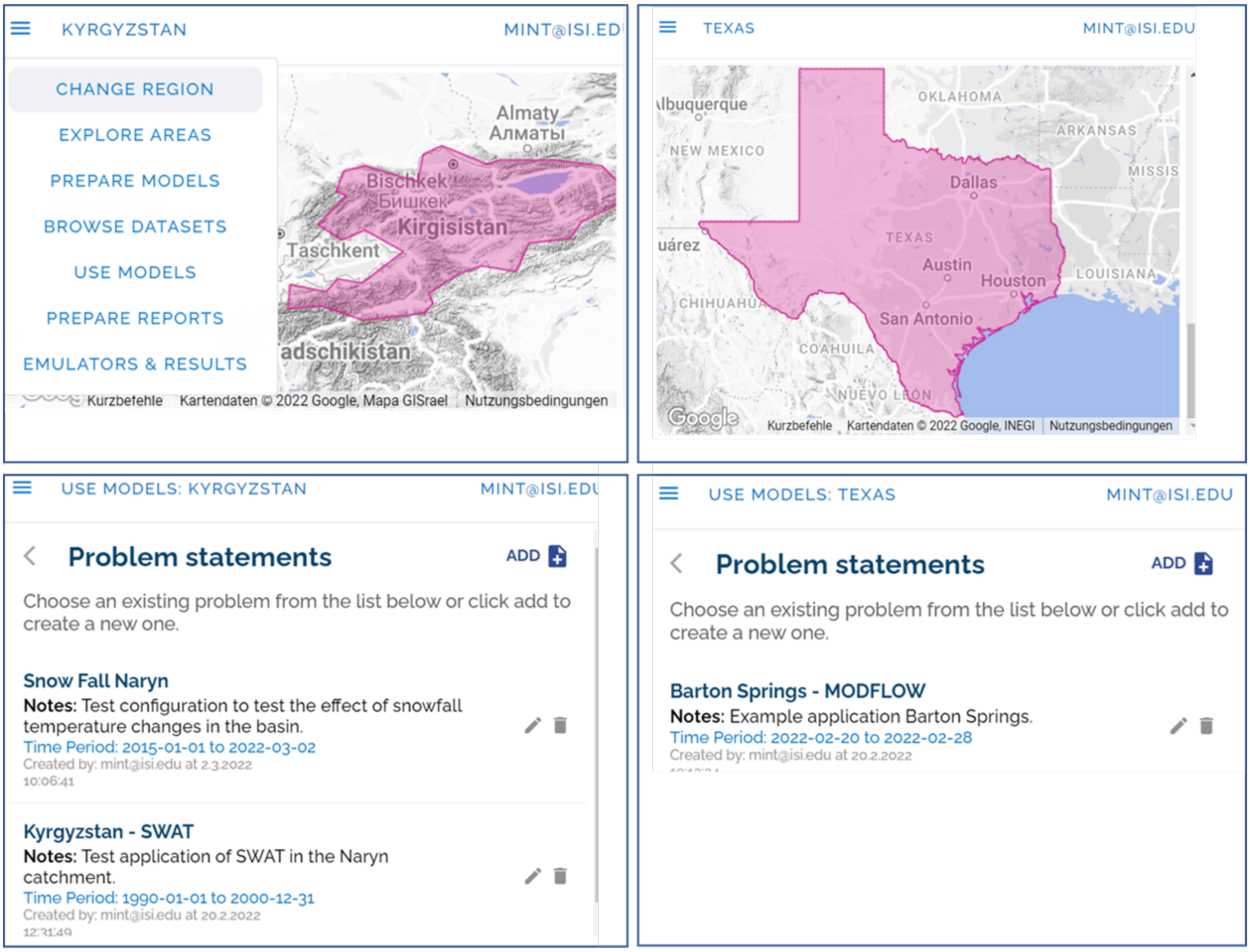

Figure 6: Illustration how the two case studies can be accessed within the MINT modeling framework, each shown on the left and right sides of the figure. The upper panels refer to the selection of the study area from a map. The lower panels show the corresponding problem statements that drive the set up and execution of the model configurations.

specifies a problem statement by selecting a time period for the simulation, a region of interest, and desired response variables (i.e., simulation outputs). Once the problem statement is specified, MINT will show the user relevant model configurations that can be run. Fig. 6 shows the MINT user interface to access model configurations in the different regions. More details are provided in the next section.

6.2. Model, Dataset & Parameter Selection

The Naryn case study aims to simulate discharge by adjusting the snow parameters that govern the predominant processes in the region. In detail, these processes involve snowmelt and snowfall and therefore the snowpack distribution in the region. These processes control discharge generation. Thus, a task was created where river discharge was used as response variable. As shown in Figure 7 a), it would also be possible to use other models to obtain discharge, such as TopoFlow (Peckham, 2009). A similar overview for the Barton Springs case study is provided in Figure 8.

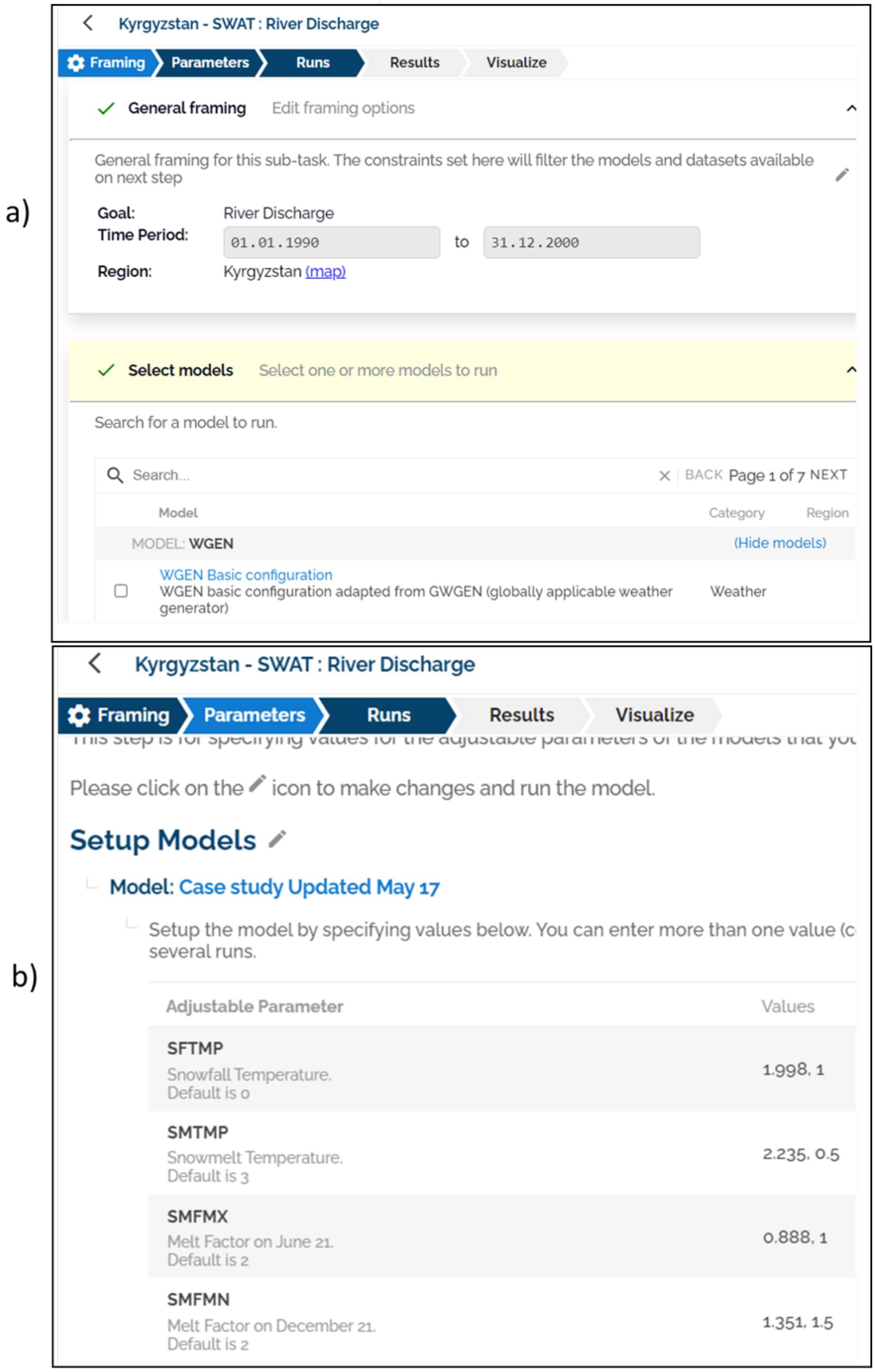

Figure 7: Illustration of: a) available model components for the simulation of discharge for the Naryn case study; b) the snow parameter modification of our model within the problem statement and task section of MINT, exemplified at the Naryn case study. It is demonstrated how the four exposed snow parameters are predefined with default values, that can be directly adjusted here.

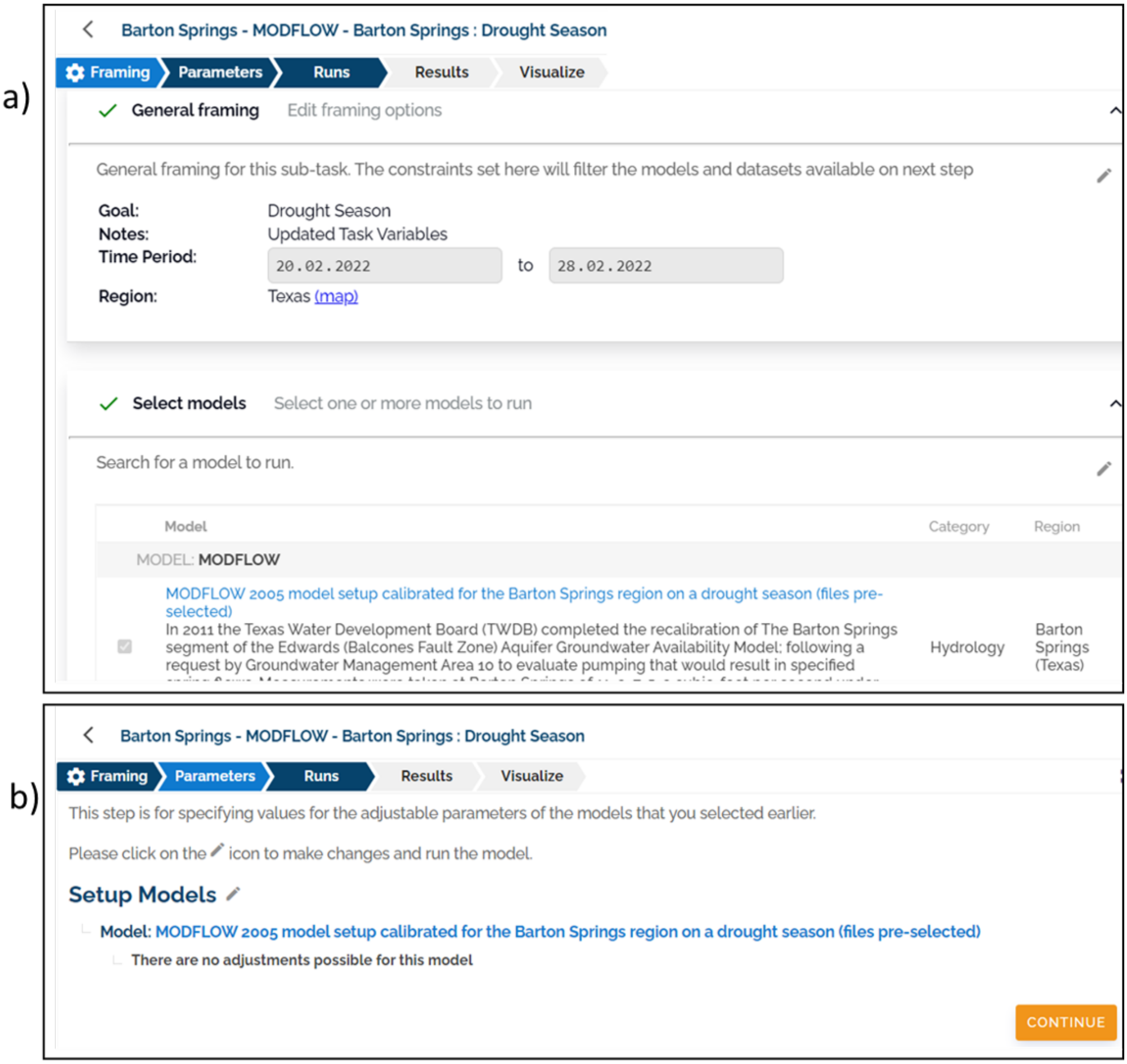

Figure 8: Illustration of: a) available model components for the drought assessment of the Barton Spring case study. b) the parameter modification of our model within the problem statement and task section of MINT, again exemplified at the case study of the Barton Springs. Due to the static design, no parameters can be adjusted, but the model can simply be run with the default parameter values.

Multiple model components may be selected, which would allow users to easily create a model ensemble to provide a more differentiated picture. At this point the user would choose among all available input datasets, such as meteorological information with precipitation and temperature. An example can be found in Appendix A1 where users can choose between two alternative well files (with different pumping rates) for the Barton Springs area, which would allow them to evaluate the effects on the groundwater levels under various pumping rates.

Users can also easily specify different parameter values to reflect different scenarios or assumptions. As described in

Section 5.3.2, four snow parameters were exposed and the user can assign them different values to explore different scenarios, as shown in Figure 7 b). This allows users to explore different dates for the onset of the snowmelt season, represented by various temperature thresholds. If users provide several values, MINT runs the model multiple times. All outputs are provided individually for each run. In the parameter tab (Figure 7 b) users can also check the default values of each parameter. This is particularly relevant if the default values are based on expert knowledge and refer to a specific baseline which can be used for the comparison with developed scenarios. The static design of the Barton Springs components does not allow for any parameter changes, thus the corresponding tab is empty (see Figure 8 b).

6.3. Execute & Analyze Model Components

When users set parameter values, the model component can be executed and the “Results" tab in MINT provides a summary of all the outputs. In the Naryn use case, the outputs were limited to the *output.rch* file where the discharge information is stored. The results represent the discharge response to the adjusted snow dynamics in the basin, which were compared with the baseline simulation results obtained with the default parameter values (no shift). For every parameter combination a model run was executed and the corresponding outputs were generated. An example is shown in Appendix A2, where the user specified two values for each of the four parameters which led to 16 combinations each resulting in an execution and each with its output files. To eliminate unnecessary computations, MINT caches the results of executions.

MINT can generate some standard visualizations, but users often generate their own custom visualizations after

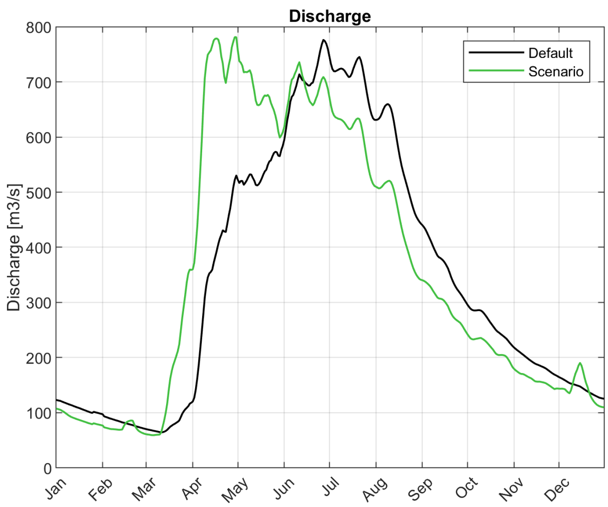

Figure 9: Results of the discharge simulation with the SWAT case study when the scenario is run (*green line*), compared to the default setup (*black line*). The figure indicates the seasonal shift towards an early melt onset under changing conditions. The example intends to show what future versions of MINT could directly visualize. The lines represent a mean over five years for each Julian Day.

downloading the results in a post-processing step. We show a visualization of the results from the SWAT case study comparing a particular scenario versus the baseline in Figure 9. Users can now directly derive the desired information, depending on how strong the average shift in river discharge would be under changing snow conditions. The results shown in the figure represent the mean of a five-year period. From the plot it becomes apparent that the conditions represented by the scenario would lead to a strong rise in discharge in early March already. Besides, the earlier onset and steep rise of snowmelt would cause higher discharge in April and May compared to the baseline, while it would be reduced during the summer months. The annual peak would already occur 1.5-2 months earlier than in the baseline. This significant change in the flow regime may have far-reaching consequences for the water sector. For example, the summer reduction may affect agricultural production as irrigation water is missing, while the strong increase after the winter months may promote the damage potential of flood events. Ultimately, water authorities may conclude to assess the potential of a reservoir to mitigate those undesired effects.

For the second case study, we declared a problem statement where we included the different Barton Springs model components. To access the model configurations, the study area in MINT has first to be changed to Texas (analogously to Fig. 6), where we can then select *Barton Springs - MODFLOW*. Users can create tasks, which reflect a specific scenario, and select an appropriate model configuration. One of the scenarios may focus on drought assessment, using the M-B configuration. Another task may be for average conditions, using the M-A configuration. A third task may focus on the impacts of specific recharge and pumping conditions, using the M-I configuration. Users have the possibility to compare the three different setups and easily analyze the differences in groundwater availability in the region. In detail, one can evaluate the effects of pumping on groundwater levels and study how the aquifer should be managed to maintain flow under specific conditions. Users might infer that, under drought conditions, pumping alone is not sufficient. In contrast to the SWAT model configuration, where only one output file is accessible, the MODFLOW model configurations offer four different output files.

It is worth noting that the application of the scenarios does not require any computing/programming skills for users. However, if users want to run encapsulated models locally, basic container skills are required. In general this is seldom the case, since MINT relies on user-friendly GUI (Fig. 6 and 8).

7. Discussion

Models created by experts are usually difficult to use by modelers in other disciplines. Despite the need by decision-makers to access sophisticated models, they remain inaccessible to non-experts (Bagstad et al., 2013). Even experts within a discipline find that it takes significant effort to setup and compare models from other modelers (Lüke and Hack, 2018; Francesconi et al., 2016). Our work shows that two very different hydrological models could be encapsulated using the same methodology to simplify model dissemination by experts for use by non-experts. Our MIC tool can be used by expert modelers without major knowledge of software engineering (e.g., using software containers, managing execution dependencies, or setting up code repositories). We demonstrated the methodology for different model domains, purposes, technical details, and model structures.

Our case studies illustrate that modelers only have to determine the parameters and input and output files to be exposed, according to the intended scenarios. Different uses of a model (e.g., snow-related analysis or studies focusing on crop yield) lead to different model configurations and are organized and easily accessible in MINT. The methodology enables expert modelers to create useful abstractions of existing models. The abstraction hides the part of the model complexity that is not necessarily required for the target users. Therefore, once a model has been encapsulated with our methodology, non-expert users are relieved from dealing with the technical details of the model execution or its structure.

Different types of non-experts may benefit from our effort, depending on their expertise and background. For example, citizens of hydrological extremes (drought and floods), who become relevant stakeholders and develop a certain level of expertise to understand their own scenarios; NGO members who are interested in model applications in the environmental sector; or decision makers who usually have a decent hydrological know-how, but may not be familiar with modeling (water authorities are often busy with administrative work, which means that there is little time for the construction and calibration of complex models). Additionally, we envision expert modelers to benefit from this effort, as it facilitates the creation of model ensembles for model comparisons or for benchmarking.

Our methodology may be used to share and use pre-agreed scenarios (as in our Barton Springs case study), and support users developing their own scenarios independently by modifying the exposed parameters. We also included the possibility of exposing input datasets in model configurations so users can select their own. For example, several meteorological data sets may be used for the execution of a model configuration. Processing all required input data is time-consuming and HMs often have different requirements. Exchanging these data often represents an obstacle (Gardner et al., 2018) that can be at least partially overcome by using MINT. Modelers are also encouraged to describe their configurations with metadata so that users can search flexibly for models and use those that are suitable for their scenarios. A region-specific search (which corresponds to Kyrgyzstan or Texas in our examples) allows users finding all available models for that region. Modelers should also provide code for output visualizations (see Section 6.3). The integration of a general visualization environment in MINT would facilitate the usability in extended scenarios, for example by integrating other datasets that may be relevant to the modeling scenarios (e.g., population density, road access, etc.).

Although the examples of this paper focus on hydrological models, our methodology has been applied to models in other domains, including agriculture and economics. We assume all encapsulated models to be open source, or have an open source executable that can be shared in a software container.

This methodology helps aligning a software component with the findable, accessible, interoperable, and reusable principles (FAIR) for data (Wilkinson et al., 2016), following current best practices for Open Science. By creating software components that have specific functionality and clear invocation and results, modelers provide self-contained and pre-prepared model components that are well characterized and become easier to reuse than the original modeling software. Model components are more accessible than the original modeling software as they are encapsulated in a software container that can be executed in any platform. Model components include extensive machine-readable software metadata that makes them more findable and interoperable.

Finally, it is worth noting that we used pre-calibrated models for our case studies. Future work will address this limitation by integrating model calibration capabilities into our framework and methodology.

8. Conclusions

This paper introduced a methodology to simplify the dissemination of expert models to non-expert users. The methodology guides modeling experts when creating software components that explore specific modeling scenarios. The methodology is applicable to any kind of model, regardless of its discipline, processes or technical details. The implementation of the methodology in the MIC tool enables a simple model encapsulation process for modelers. This does not only facilitate model dissemination and provision, but can also improve mutual work within or across disciplines and groups. In addition, the complexity of the model can be simplified by creating model configurations that suit the needs of non-expert users. Our proposed methodology thus creates new possibilities in model abstractions and promotes the satisfaction of end-user needs. This is also supported by the easy access options of model configurations in MINT, which greatly simplifies their (re)use.

We illustrated our methodology with two case studies, using two different hydrological models in two different regions of the world. The case studies provide examples how potential scenarios and use cases for the application of the methodology could look like. However, the universal applicability of the methodology within any modeling discipline enables a free design of scenarios with numerous potential use cases that can help both, the expert modeler as well as the end-user. MINT users can easily compare the effects of pumpage under different conditions on groundwater levels. Moreover, they can infer whether pumping is suitable to maintain flow under drought conditions or if additional measures should be taken into account. Additionally, we showed how a restriction of the parameter space to a useful minimum can facilitate the exploration of discharge shifts by decision-makers. The methodology encourages the possibility of independently investigating scenarios and to derive valuable insights. For example, resulting discharge shifts may lead to several consequences for the water sector, e.g., increased flood risk or decreased agricultural production to mention only two out of dozens, that call for action.

Our work supports the FAIR principles, helping model components to be more findable, accessible, interoperable and reusable. However, our methodology also presents some limitations, which are part of our future work. For example, while our methodology helps non-experts executing models created by expert modelers, some expertise is still needed to interpret the results of the simulations. In some cases this is addressed by adding documentation and metadata in the scenario, in order to provide the right context for end users. In other cases, expert modelers include ad-hoc visualizations that are executed with the model itself, helping to interpret the outputs. Extending our methodology to ensure that visualization components are described for each model output would help address this issue. We are also exploring extending MINT with general-purpose visualizations (e.g., variables obtained in tabular model results).

Another point of improvement involves expanding the supported actions for modeling experts in MINT. For example, including additional data transformations and model calibration (right now models are calibrated by experts independently).

Finally, additional case studies in other domains are part of our future work in order to further refine the applicability of our approach when disseminating models across disciplines, lowering the barrier of adoption of models by modeling experts.

Software and Data Availability

Name of the software: Model component 1 - Snow dynamics
Developer: Timo Schaffhauser (t.schaffhauser@tum.de), Maximiliano Osorio (mosorio@isi.edu)
Software availability: `https://hub.docker.com/r/mosorio/naryn_nival_setup/tags` (Docker image)
Compressed size: 286.97 MB (Docker image)

Name of the software: Model component 2 - Drought impact
Developer: Suzanne Pierce (spierce@tacc.utexas.edu), Maximiliano Osorio (mosorio@isi.edu)
Software availability: `https://hub.docker.com/r/mintproject/modflow-2005/tags` (Docker image)
Compressed size: 733.55 MB (Docker image)

Name of the software: Model Insertion Checker (MIC)
Developer: Maximiliano Osorio (mosorio@isi.edu)
Software availability: `https://zenodo.org/record/6024985}.YvPflnZByMo/`
Programming language: Python
Compressed size: 19.9 MB

Name of the dataset: SWAT & MODFLOW Model Components
Developer: Timo Schaffhauser (t.schaffhauser@tum.de), Daniel Garijo, Maximiliano Osorio, Daniel Bittner, Suzanne

Pierce, Hernan Vargas, Markus Disse, Yolanda Gil

Data availability: https://zenodo.org/record/6948339}.YvJ6V3ZByMr

Form of repository: Zenodo archive

Compressed size: 51.7 MB

Further access to the model components is possible via https://mint.isi.edu/kyrgyzstan/models/explore/SWAT/8cc84426-d849-471b-9a5e-47bcaf094607/6a36a2e5-73bf-4098-9acd-1aaaab383d4a/14580635-c7ca-4256-935a-4ddbdacfbfe2 and https://mint.isi.edu/texas/models/explore/MODFLOW/modflow_2005/modflow_2005_cfg/modflow_2005_BartonSprings_avg.

Acknowledgements

We would like to thank our collaborators in the MINT project, particularly modeling experts and users that contributed to the design of the overall MINT framework. We gratefully acknowledge support from the US Defense Advanced Research Projects Agency through award W911NF-18-1-0027 and the US Office of Naval Research through award N00014-21-1-2437. The authors from TUM also want to thank the BMBF (Bundesministerium für Bildung und Forschung) for the funding of the "OekoFlussPlan" project (grant number 01LZ1802B). The author from UPM has been supported by the Madrid Government (Comunidad de Madrid-Spain) under the Multiannual Agreement with Universidad Politécnica de Madrid in the line Support for R&D projects for Beatriz Galindo researchers, in the context of the V PRICIT (Regional Programme of Research and Technological Innovation) and the call Research Grants for Young Investigators from Universidad Politécnica de Madrid.

Appendices

A1

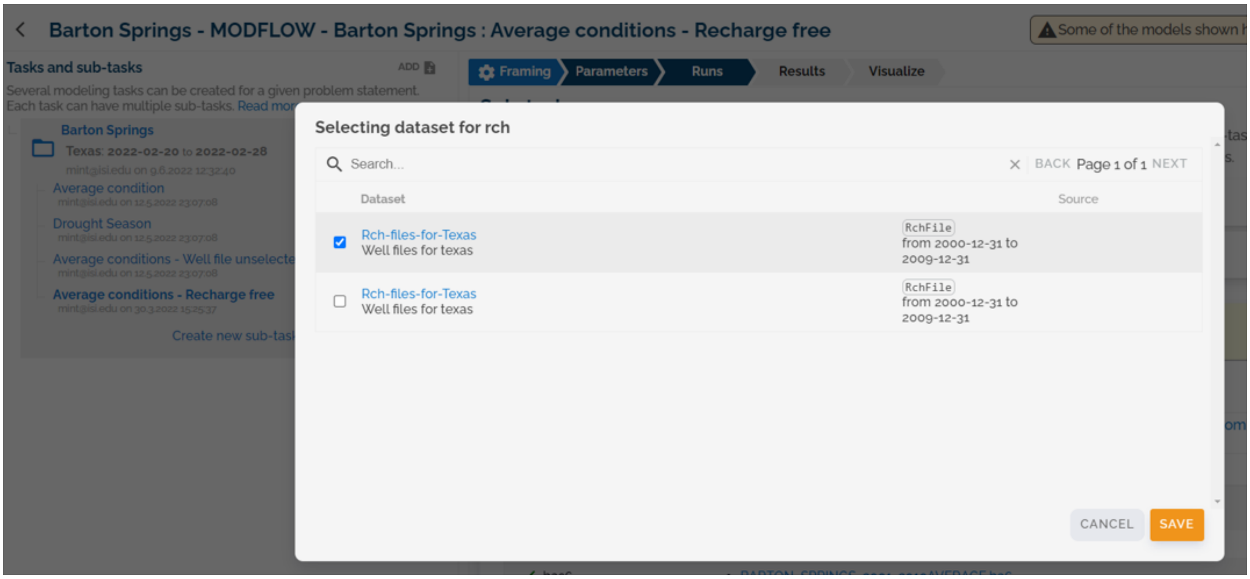

Figure A1: Illustration of the selection of different input datasets for the well files. For the Barton Springs case study users would have two possibilities in that case.

A2

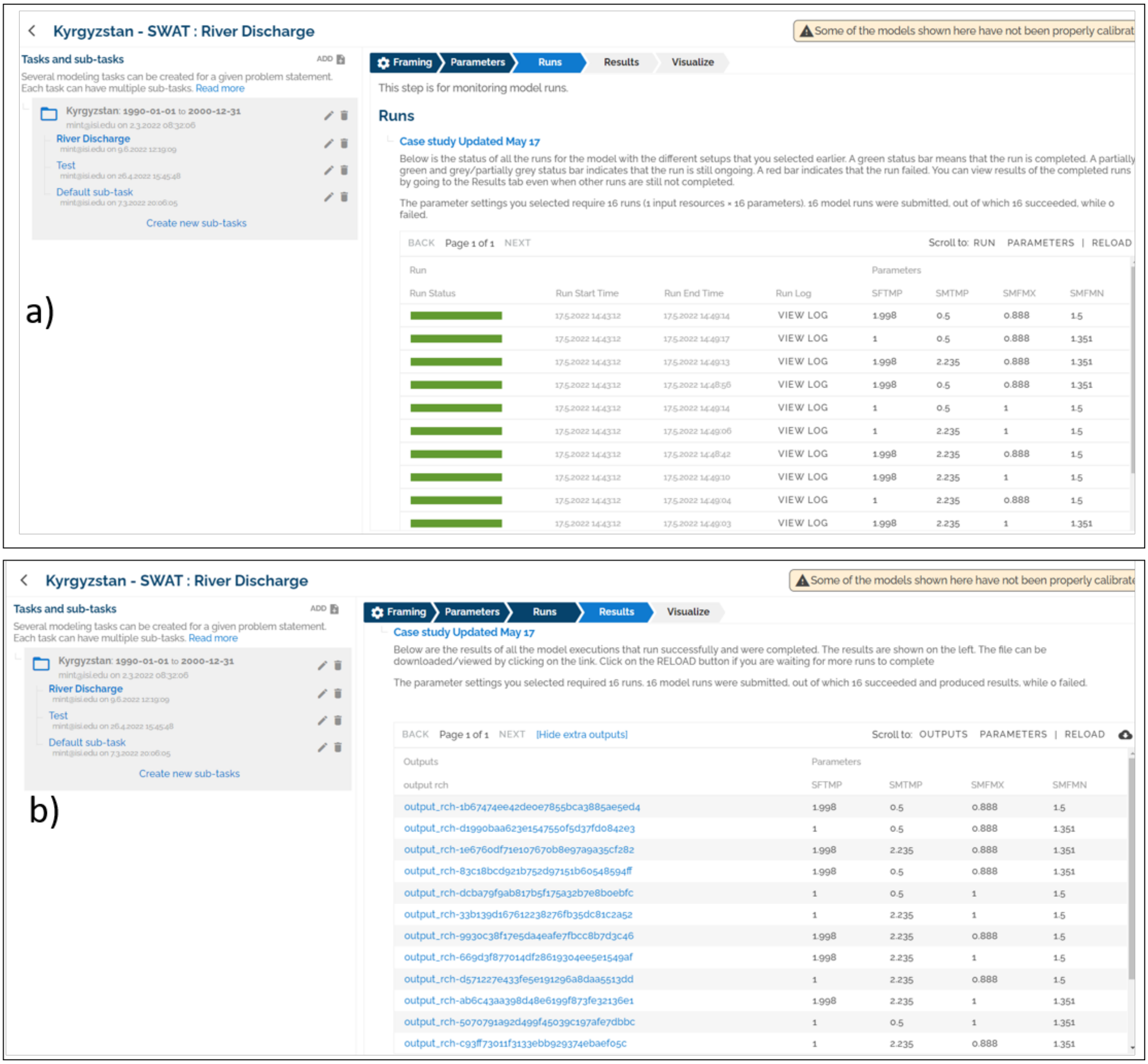

Figure A2: Example of: a) in total 16 different runs of the model component, since for all exposed parameters two different values were set; and b) the corresponding 16 output files, which were generated through running the component with all 16 potential parameter value combinations.

References

Andreu, J., Capilla, J., Sanchís, E., 1996. AQUATOOL, a generalized decision-support system for water-resources planning and operational management. Journal of Hydrology 177, 269–291. doi:10.1016/0022-1694(95/02963-x.

Arnold, J.G., Fohrer, N., 2005. SWAT2000: current capabilities and research opportunities in applied watershed modelling. Hydrological Processes 19, 563–572. doi:10.1002/hyp.5611.

Arnold, J.G., Srinivasan, R., Muttiah, R.S., Williams, J.R., 1998. LARGE AREA HYDROLOGIC MODELING AND ASSESSMENT PART i: MODEL DEVELOPMENT. Journal of the American Water Resources Association 34, 73–89. doi:10.1111/j.1752-1688.1998.tb05961.x.

Bagstad, K.J., Semmens, D.J., Waage, S., Winthrop, R., 2013. A comparative assessment of decision-support tools for ecosystem services quantification and valuation. Ecosystem Services 5, 27–39. doi:10.1016/j.ecoser.2013.07.004.

Bernstein, B.B., Thompson, B.E., Smith, R.W., 1993. A combined science and management framework for developing regional monitoring objectives. Coastal Management 21, 185–195. doi:10.1080/08920759309362202.

Bittner, D., Narany, T.S., Kohl, B., Disse, M., Chiogna, G., 2018. Modeling the hydrological impact of land use change in a dolomite-dominated karst system. Journal of Hydrology 567, 267–279.

Bittner, D., Rychlik, A., Klöffel, T., Leuteritz, A., Disse, M., Chiogna, G., 2020. A gis-based model for simulating the hydrological effects of land use changes on karst systems–the integration of the lukars model into freewat. Environmental Modelling & Software 127, 104682.

Boettiger, C., 2015. An introduction to docker for reproducible research. ACM SIGOPS Operating Systems Review 49, 71–79.

Booij, M.J., Krol, M.S., 2010. Balance between calibration objectives in a conceptual hydrological model. Hydrological Sciences Journal–Journal des Sciences Hydrologiques 55, 1017–1032.

Bradshaw, G.A., Borchers, J., 2000. Uncertainty as information: Narrowing the science-policy gap. Conservation Ecology 4.

Brocca, L., Ciabatta, L., Massari, C., Camici, S., Tarpanelli, A., 2017. Soil moisture for hydrological applications: Open questions and new opportunities. Water 9, 140.

Brunner, P., Simmons, C.T., 2012. Hydrogeosphere: a fully integrated, physically based hydrological model. Groundwater 50, 170–176.

Collins, N., Theurich, G., DeLuca, C., Suarez, M., Trayanov, A., Balaji, V., Li, P., Yang, W., Hill, C., da Silva, A., 2005. Design and implementation of components in the earth system modeling framework. The International Journal of High Performance Computing Applications 19, 341–350. doi:10.1177/1094342005056120.

Döll, P., Kaspar, F., Lehner, B., 2003. A global hydrological model for deriving water availability indicators: model tuning and validation. Journal of Hydrology 270, 105–134. doi:10.1016/s0022-1694(02/00283-4.

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., Tarboton, D., 2016. An overview of current applications, challenges, and future trends in distributed process-based models in hydrology. Journal of Hydrology 537, 45–60. doi:10.1016/j.jhydrol.2016.03.026.

Francesconi, W., Srinivasan, R., Pérez-Miñana, E., Willcock, S.P., Quintero, M., 2016. Using the soil and water assessment tool (SWAT) to model ecosystem services: A systematic review. Journal of Hydrology 535, 625–636. doi:10.1016/j.jhydrol.2016.01.034.

Gan, R., Luo, Y., Zuo, Q., Sun, L., 2015. Effects of projected climate change on the glacier and runoff generation in the naryn river basin, central asia. Journal of Hydrology 523, 240–251. URL: https://doi.org/10.1016/j.jhydrol.2015.01.057, doi:10.1016/j.jhydrol.2015.01.057.

Gan, T., Tarboton, D.G., Dash, P., Gichamo, T.Z., Horsburgh, J.S., 2020. Integrating hydrologic modeling web services with online data sharing to prepare, store, and execute hydrologic models. Environmental Modelling & Software 130, 104731. doi:10.1016/j.envsoft.2020.104731.

Gardner, M.A., Morton, C.G., Huntington, J.L., Niswonger, R.G., Henson, W.R., 2018. Input data processing tools for the integrated hydrologic model GSFLOW. Environmental Modelling & Software 109, 41–53. doi:10.1016/j.envsoft.2018.07.020.

Garijo, D., Osorio, M., Khider, D., Ratnakar, V., Gil, Y., 2019. OKG-Soft: An Open Knowledge Graph with Machine Readable Scientific Software Metadata, in: 2019 15th International Conference on eScience (eScience), IEEE, San Diego, CA, USA. pp. 349–358. doi:10.1109/eScience.2019.00046.

Gil, Y., Cobourn, K., Deelman, E., Duffy, C., da Silva, R.F., Kemanian, A., Knoblock, C.A., Kumar, V., Peckham, S., Carvalho, L., Chiang, Y.Y., Garijo, D., Khider, D., Khandelwal, A., Pham, M., Pujara, J., Ratnakar, V., Stoica, M., Vu, B., 2018. Mint: Model integration through knowledge-powered data and process composition, in: 9th International Congress on Environmental Modelling and Software.

Gil, Y., Garijo, D., Khider, D., Knoblock, C.A., Ratnakar, V., Osorio, M., Vargas, H., Pham, M., Pujara, J., Shbita, B., Vu, B., Chiang, Y.Y., Feldman, D., Lin, Y., Song, H., Kumar, V., Khandelwal, A., Steinbach, M., Tayal, K., Xu, S., Pierce, S.A., Pearson, L., Hardesty-Lewis, D., Deelman, E., Silva, R.F.D., Mayani, R., Kemanian, A.R., Shi, Y., Leonard, L., Peckham, S., Stoica, M., Cobourn, K., Zhang, Z., Duffy, C., Shu, L., 2021. Artificial intelligence for modeling complex systems: Taming the complexity of expert models to improve decision making. ACM Trans. Interact. Intell. Syst. 11. doi:10.1145/3453172.

Haasnoot, M., Middelkoop, H., Van Beek, E., Van Deursen, W., 2011. A method to develop sustainable water management strategies for an uncertain future. Sustainable Development 19, 369–381.

Hanson, R.T., Boyce, S.E., Schmid, W., Hughes, J.D., Mehl, S.W., Leake, S.A., Maddock, T., Niswonger, R.G., 2014. One-water hydrologic flow model (MODFLOW-OWHM). doi:10.3133/tm6a51.

Harbaugh, A.W., 2005. MODFLOW-2005 : the u.s. geological survey modular ground-water model–the ground-water flow process. doi:10.3133/tm6a16.

Hattermann, F.F., Vetter, T., Breuer, L., Su, B., Daggupati, P., Donnelly, C., Fekete, B., Flörke, F., Gosling, S.N., Hoffmann, P., Liersch, S., Masaki, Y., Motovilov, Y., Müller, C., Samaniego, L., Stacke, T., Wada, Y., Yang, T., Krysnaova, V., 2018. Sources of uncertainty in hydrological climate impact assessment: a cross-scale study 13, 015006. doi:10.1088/1748-9326/aa9938.

Hill, C., DeLuca, C., Balaji, Suarez, M., Silva, A.D., 2004. The architecture of the earth system modeling framework. Computing in Science & Engineering 6, 18–28. doi:10.1109/mcise.2004.1255817.

Horsburgh, J.S., Morsy, M.M., Castronova, A.M., Goodall, J.L., Gan, T., Yi, H., Stealey, M.J., Tarboton, D.G., 2016. Hydroshare: Sharing diverse environmental data types and models as social objects with application to the hydrology domain. JAWRA Journal of the American Water Resources Association 52, 873–889.

Keller, C.A., Long, M.S., Yantosca, R.M., Silva, A.M.D., Pawson, S., Jacob, D.J., 2014. HEMCO v1.0: a versatile, ESMF-compliant component for calculating emissions in atmospheric models. Geoscientific Model Development 7, 1409–1417. doi:10.5194/gmd-7-1409-2014.

Kinzelbach, W., Wang, H., Li, Y., Wang, L., Li, N., 2021. Decision support for local water authorities in guantao, in: Springer Water. Springer Singapore, pp. 77–136. URL: https://doi.org/10.1007/978-981-16-5843-3_4, doi:10.1007/978-981-16-5843-3_4.

Koltsida, E., Kallioras, A., 2019. Groundwater flow simulation through the application of the FREEWAT modeling platform. Journal of Hydroinformatics 21, 812–833. URL: https://doi.org/10.2166/hydro.2019.040, doi:10.2166/hydro.2019.040.

Konikow, L.F., 2010. The secret to successful solute-transport modeling. Ground Water 49, 144–159. URL: https://doi.org/10.1111/j.1745-6584.2010.00764.x, doi:10.1111/j.1745-6584.2010.00764.x.

Krysanova, V., Hattermann, F.F., 2017. Intercomparison of climate change impacts in 12 large river basins: overview of methods and summary of results 141, 363–379. doi:10.1007/s10584-017-1919-y.

Kurtzer, G.M., Sochat, V., Bauer, M.W., 2017. Singularity: Scientific containers for mobility of compute. PloS one 12, e0177459.

Lautenbach, S., Berlekamp, J., Graf, N., Seppelt, R., Matthies, M., 2009. Scenario analysis and management options for sustainable river basin management: Application of the elbe DSS. Environmental Modelling & Software 24, 26–43. doi:10.1016/j.envsoft.2008.05.001.

Lobanova, A., Liersch, S., Nunes, J.P., Didovets, I., Stagl, J., Huang, S., Koch, H., del Rocío Rivas López, M., Maule, C.F., Hattermann, F., Krysanova, V., 2018. Hydrological impacts of moderate and high-end climate change across european river basins 18, 15–30. doi:10.1016/j.ejrh.2018.05.003.

Lüke, A., Hack, J., 2018. Comparing the applicability of commonly used hydrological ecosystem services models for integrated decision-support. Sustainability 10, 346. doi:10.3390/su10020346.

Merz, B., Hall, J., Disse, M., Schumann, A., 2010. Fluvial flood risk management in a changing world. Natural Hazards and Earth System Sciences 10, 509–527.

Morales, T., Uriarte, J.A., Olazar, M., Antigüedad, I., Angulo, B., 2010. Solute transport modelling in karst conduits with slow zones during different hydrologic conditions. Journal of Hydrology 390, 182–189. URL: https://doi.org/10.1016/j.jhydrol.2010.06.041, doi:10.1016/j.jhydrol.2010.06.041.

Newman, A.J., Mizukami, N., Clark, M.P., Wood, A.W., Nijssen, B., Nearing, G., 2017. Benchmarking of a physically based hydrologic model. Journal of Hydrometeorology 18, 2215–2225.

Osorio, M., Garijo, D., Heidelberg, C., Khider, D., hvarg, Dhruv, Rampin, R., Maurya, R., Gil, Y., 2022. mintproject/mic: 2.2.2. doi:10.5281/zenodo.6024985.

Overeem, I., Berlin, M.M., Syvitski, J.P., 2013. Strategies for integrated modeling: The community surface dynamics modeling system example. Environmental Modelling & Software 39, 314–321. doi:10.1016/j.envsoft.2012.01.012.

Panagopoulos, A., 2022. Brine management (saline water & wastewater effluents): Sustainable utilization and resource recovery strategy through minimal and zero liquid discharge (mld & zld) desalination systems. Chemical Engineering and Processing - Process Intensification 176, 108944. doi:https://doi.org/10.1016/j.cep.2022.108944.

Passarello, M.C., Pierce, S.A., Sharp, J.M., 2014. Uncertainty and urban water recharge for managing groundwater availability using decision support. Water Science and Technology 70, 1888–1896. doi:10.2166/wst.2014.437.

Passarello, M.C., Sharp, J.M., Pierce, S.A., 2012. Estimating urban-induced artificial recharge: A case study for austin, TX. Environmental & Engineering Geoscience 18, 25–36. doi:10.2113/gseegeosci.18.1.25.

Peckham, S., 2009. Chapter 25 geomorphometry and spatial hydrologic modelling, in: Developments in Soil Science. Elsevier, pp. 579–602. URL: https://doi.org/10.1016/s0166-2481(08/00025-1, doi:10.1016/s0166-2481(08/00025-1.

Peckham, S., Syvitski, J., 2007. Evaluation of model coupling frameworks for use by the community surface dynamics modeling system (csdms).

Peckham, S.D., Hutton, E.W., Norris, B., 2013. A component-based approach to integrated modeling in the geosciences: The design of CSDMS. Computers & Geosciences 53, 3–12. doi:10.1016/j.cageo.2012.04.002.

Pezij, M., Augustijn, D.C., Hendriks, D.M., Hulscher, S.J., 2019. The role of evidence-based information in regional operational water management in the netherlands. Environmental Science & Policy 93, 75–82. doi:10.1016/j.envsci.2018.12.025.

Pierce, S.A., 2006. Groundwater Decision Support: An integrated assessment linking causal narratives, numerical models, and combinatorial search techniques to determine available yield for an aquifer system. Ph.D. thesis. Jackson School of Geosciences, Department of Geology, The University of Texas at Austin.

Pierce, S.A., Dulay, M.M., Sharp, J.M., Lowry, T.S., Tidwell, V.C., 2006. Defining tenable groundwater management: Integrating stakeholder preferences, distributed parameter models, and systems dynamics to aid groundwater resource allocation. MODFLOW and More. International Groundwater Modeling Center, Golden, Colorado .

Rampin, R., Chirigati, F., Shasha, D., Freire, J., Steeves, V., 2016. Reprozip: The reproducibility packer. Journal of Open Source Software 1, 107.

Refsgaard, J.C., Storm, B., Clausen, T., 2010. Système hydrologique europeén (SHE): review and perspectives after 30 years development in distributed physically-based hydrological modelling. Hydrology Research 41, 355–377. doi:10.2166/nh.2010.009.

Reinecke, R., Foglia, L., Mehl, S., Herman, J.D., Wachholz, A., Trautmann, T., Döll, P., 2019. Spatially distributed sensitivity of simulated global groundwater heads and flows to hydraulic conductivity, groundwater recharge, and surface water body parameterization. Hydrology and Earth System Sciences 23, 4561–4582.

Rossetto, R., Filippis, G.D., Borsi, I., Foglia, L., Cannata, M., Criollo, R., Vázquez-Suñé, E., 2018. Integrating free and open source tools and distributed modelling codes in GIS environment for data-based groundwater management. Environmental Modelling & Software 107, 210–230. doi:10.1016/j.envsoft.2018.06.007.

Ruiz-Ortiz, V., García-López, S., Solera, A., Paredes, J., 2019. Contribution of decision support systems to water management improvement in basins with high evaporation in mediterranean climates. Hydrology Research 50, 1020–1036. doi:10.2166/nh.2019.014.

Scanlon, B.R., Mace, R.E., Barrett, M.E., Smith, B.A., 2003. Can we simulate regional groundwater flow in a karst system using equivalent porous media models? case study, barton springs edwards aquifer, USA. Journal of Hydrology 276, 137–158. doi:10.1016/s0022-1694(03/00064-7.

Scanlon, B.R., Mace, R.E., Smith, B.A., Hovorka, S.D., Dutton, A.R., Reedy, R.C., 2001. Groundwater availability of the barton springs segment of the edwards aquifer, texas: Numerical simulations through 2050. The University of Texas at Austin, Bureau of Economic Geology, final report prepared for the Lower Colorado River Authority .

Schaffhauser, T., Lange, S., Tuo, Y., Disse, M., 2023. Shifted discharge and drier soils: Hydrological projections for a central asian catchment. Journal of Hydrology: Regional Studies 46, 101338. URL: https://www.sciencedirect.com/science/article/pii/S2214581823000253, doi:https://doi.org/10.1016/j.ejrh.2023.101338.

Surfleet, C.G., Tullos, D., Chang, H., Jung, I.W., 2012. Selection of hydrologic modeling approaches for climate change assessment: A comparison of model scale and structures. Journal of Hydrology 464-465, 233–248. doi:10.1016/j.jhydrol.2012.07.012.

Timmerman, J.G., Langaas, S., 2005. Water information: what is it good for? the use of information in transboundary water management. Regional Environmental Change 5, 177–187. doi:10.1007/s10113-004-0087-6.

Unger-Shayesteh, K., Vorogushyn, S., Farinotti, D., Gafurov, A., Duethmann, D., Mandychev, A., Merz, B., 2013. What do we know about past changes in the water cycle of central asian headwaters? a review. Global and Planetary Change 110, 4–25. URL: https://doi.org/10.1016/j.gloplacha.2013.02.004, doi:10.1016/j.gloplacha.2013.02.004.

US Army Corps of Engineers, 2000. Hydrologic modeling system HEC-HMS: technical reference manual. US Army Corps of Engineers, Hydrologic Engineering Center, Davis, CA.

Wilkinson, M.D., Dumontier, M., Aalbersberg, I.J., Appleton, G., Axton, M., Baak, A., Blomberg, N., Boiten, J.W., da Silva Santos, L.B., Bourne, P.E., Bouwman, J., Brookes, A.J., Clark, T., Crosas, M., Dillo, I., Dumon, O., Edmunds, S., Evelo, C.T., Finkers, R., Gonzalez-Beltran, A., Gray, A.J., Groth, P., Goble, C., Grethe, J.S., Heringa, J., 't Hoen, P.A., Hooft, R., Kuhn, T., Kok, R., Kok, J., Lusher, S.J., Martone, M.E., Mons, A., Packer, A.L., Persson, B., Rocca-Serra, P., Roos, M., van Schaik, R., Sansone, S.A., Schultes, E., Sengstag, T., Slater, T., Strawn, G., Swertz, M.A., Thompson, M., van der Lei, J., van Mulligen, E., Velterop, J., Waagmeester, A., Wittenburg, P., Wolstencroft, K., Zhao, J., Mons, B., 2016. The FAIR guiding principles for scientific data management and stewardship. Scientific Data 3. URL: https://doi.org/10.1038/sdata.2016.18, doi:10.1038/sdata.2016.18.

Zhang, K., Zargar, A., Achari, G., Islam, M.S., Sadiq, R., 2014. Application of decision support systems in water management. Environmental Reviews 22, 189–205. doi:10.1139/er-2013-0034.