Environmental Modelling & Software 75 (2016) 498–512

Contents lists available at ScienceDirect

Environmental Modelling & Software

journal homepage: www.elsevier.com/locate/envsoft

SWATShare – A web platform for collaborative research and education through online sharing, simulation and visualization of SWAT models

Mohammad Adnan Rajib [a], Venkatesh Merwade [a,*], I Luk Kim [b], Lan Zhao [b], Carol Song [b], Shandian Zhe [b]

[a] Lyles School of Civil Engineering, Purdue University, West Lafayette, IN 47907, USA
[b] Rosen Center for Advanced Computing, Purdue University, West Lafayette, IN 47907, USA

ARTICLE INFO

Article history:
Received 24 April 2015
Received in revised form
27 October 2015
Accepted 30 October 2015
Available online 11 November 2015

Keywords:
Cyberinfrastructure
WaterHUB
SWAT
SWATShare
XSEDE
Hydrology

ABSTRACT

Hydrologic models for a particular watershed or a region are created for addressing a specific research or management problem, and most of the models do not get reused after the project is completed. Similarly, multiple models may exist for a particular geographic location from different researchers or organizations. To avoid the duplication of efforts, and enable model reuse and enhancement through collaborative efforts, a prototype cyberinfrastructure, called SWATShare, is developed for sharing, execution and visualization of Soil and Water Assessment Tool (SWAT). The objective of this paper is to present the software architecture, functional capabilities and implementation of SWATShare as a collaborative environment for hydrology research and education using the models published and shared in the system. Besides the capability of publishing, sharing, discovery and downloading of SWAT models, some of the functions in SWATShare such as model calibration are supported by providing access to high performance computing resources including the XSEDE and cloud. Additionally, SWATShare can create dynamic spatial and temporal plots of model outputs at different scales. SWATShare can also be used as an educational tool within a classroom setting for comparing the hydrologic processes under different geographic and climatic settings. The utility of SWATShare for collaborative research and education is demonstrated by using three case studies. Even though this paper focuses on the SWAT model, the system's architecture can be replicated for other models as well.

Software availability

Product Name: SWATShare
Product Type: Web-based
Year first available: 2013
High performance computational resources: Clusters at the Purdue University and the Extreme Science and Engineering Discovery Environment (XSEDE) resources at the San Diego Super Computer Center (SDSC)
Software/web services/programming languages being used: Flex, MySQL, GeoServer, Tomcat, Apache server, PHP, Python
Deployment: WaterHUB, Purdue University, USA
Availability: publicly available at https://mygeohub.org/groups/water-hub/swatshare

1. Introduction

Hydrologic models simulate the hydrologic cycle and are used to understand the cause–effect relationships at various spatial and temporal scales. Besides helping to simulate the hydrologic cycle, hydrologic models have also been successfully used in decision support systems for integrated water resources management (e.g. Black et al., 2014; Leenhardt et al., 2012; Mahmoud et al., 2009; Rekolainen et al., 2003; Voinov and Bousquet, 2010; Wasson et al., 2003). However, the direct transfer of research results into practice is still not considered full-fledged (Borowski and Hare, 2007; De Kok et al., 2008; Delipetrev et al., 2014; Dong et al., 2013). A few factors have hindered the application of hydrologic models as educational or policy tools. First, most hydrologic models have been developed as research tools without considering their application as learning or policy tools by students, policy makers and stakeholders (Janssen et al., 2008; Olsson and Andersson, 2006). When a model is applied on a watershed for answering

* Corresponding author. 550 Stadium Mall Dr., West Lafayette, 47907, USA.
E-mail address: vmerwade@purdue.edu (V. Merwade).

http://dx.doi.org/10.1016/j.envsoft.2015.10.032
1364-8152/

research questions, the results from the study get published, but the model stays with the researchers for perpetuity. This has led to limited collaboration among model users and lack of support services to enable the reuse of already created watershed models for further research and/or educational training of students and stakeholders. Second, models have steep learning curves, intensive input data needs and demand for computational resources that are usually not available within a class room environment (Merwade and Ruddell, 2012; Seibert et al., 2013). Third, most legacy models have been developed for independent use without support for interoperability among different models and/or software platform.

Some of the issues listed above can be overcome by the creation of a platform-independent and easy-to-use modeling environment that provides access to existing models, their input/output datasets and a mechanism to perform simultaneous simulations. A web-based modeling environment is desirable for integrated water resources research and management, because it is open, interactive, fast, hierarchical and therefore, flexible (Voinov and Costanza, 1999). Many concurrent developments related to modeling reflect a trend of cyber-enabled solutions. These include the development of web-based decision support systems for creating watershed models by accessing customized databases (e.g., GISHydro@Maryland (MSHA-MDE, 2010), Rangeland Hydrology and Erosion Model (Nearing et al., 2011), Long Term Hydrologic Impact Analysis (Engel et al., 2003)), development of web services for supporting high performance parallel computing for model execution, optimization and output visualization (e.g. Bürger et al. (2012) and Delipetrev et al. (2014)), development of web-based tools for accessing and processing geospatial and remote sensing datasets for feeding into hydrologic models (e.g., Mathiyalagan et al. (2005), Kanwar et al. (2010)), and development of web-based systems for integrating data and models (e.g., Huang (2003), Rao et al. (2007)). In addition to these individual efforts, many community and governmental organizations are leading the development of cyberinfrastructure (CI) for sharing and publishing of data, development of metadata standards, and linking of hydrologic models.

CUAHSI (Consortium of Universities for the Advancement of Hydrologic Science, Inc., https://www.cuahsi.org/) in the United States launched the Hydrologic Information System (HIS, Tarboton et al. (2009)) incorporating HydroServer (Horsburgh et al., 2010) and HydroDesktop (Ames et al., 2012) which provide inter-operable web services, datasets, standards and tools for enhanced hydrologic modeling and analysis (e.g. Castronova et al., 2013; Peckham and Goodall, 2013). Terrestrial Ecosystem Research Network (TERN, http://www.tern.org.au/) is a similar major national-level project to create HIS for Australia. Efforts are also underway within the European Union, including MyWater (http://mywater-fp7.eu/), the Distributed Research Infrastructure for Hydro-Meteorology (DRIHM, http://www.drihm.eu/), Environmental Information System for Planners (EISP, Culshaw et al. (2006)) and Climate Induced Changes on the Hydrology of Mediterranean Basins (CLIMB, http://lgi-climbsrv.geographie.uni-kiel.de/). While publishing, sharing and visualization of data and/or model outputs through a web interface are common features in almost all these CI projects, only a few of them support cyber-enabled execution of distributed hydrologic models using high performance computing (HPC) resources. For example, the Swedish Meteorological and Hydrological Institute (SMHI) has recently launched European Hyrological Predictions for the Environment (E-HYPE, http://e-hypeweb.smhi.se/) that delivers real-time hydrologic and nutrient data from a calibrated hydrologic model for 35,000 sub-basins across entire Europe. As a part of EU's Trans-boundary Catchments (TRANSCAT) project, Horak et al. (2008) proposed T-DSS, a distributed prototype framework linking HEC-HMS (USACE, 2013) and MODFLOW (Harbaugh, 2005) through remote modeling and data processing servers.

Most current web-based modeling environments and tools offer remote multi-user activity, but these do not offer tools for sharing information and engaging multiple investigators around a particular modeling study. Specifically, users cannot publish and share their hydrologic models in a way that can enable collaboration among different users with diverse end objectives over a common geographic area of interest. Access to multiple models for one or more regions can also benefit students who, without putting much effort, can visualize model outputs to study the effect of climate change or land cover on hydrologic cycle. While some online tools exist for educational purposes (e.g. Aghakouchak and Habib, 2010; Habib et al., 2012), a cyber-tool that will enable students to publish and run models can led to collective effort of model validation and refinement, thus leading to advancement of both learning and research.

Considering the background information presented in the previous paragraphs, there is a need to create a CI for sharing of data and models for collaborative research and education. This need is being addressed by an ongoing project, called HydroShare (Tarboton et al., 2014), that is developing an online collaborative system for open sharing of hydrologic data and models. The goal of this paper is to describe a web-based platform, which is being developed in collaboration with HydroShare, that enables researchers and educators to increase the impact of their hydrologic modeling efforts by publishing and sharing watershed specific hydrologic models with the broader community. In return, this platform provides access to HPC resources for simulation, calibration and visualization utilities that the researchers and educators can use without relying on any local modeling or computational resources. While numerous computer models of varying complexity exist in hydro-climatology literature, the prototype cyberinfrastructure presented in this paper, called SWATShare, is created for Soil and Water Assessment Tool (SWAT). SWAT is a continuous-time, semi-distributed, process-based basin model to simulate hydrology and various water quality constituents (Arnold et al., 2012; Gassman et al., 2007; Neitsch et al., 2011). This paper provides basic description of SWATShare's functionality and the architecture behind its cyberinfrastructure, including test cases of how this cyberinfrastructure can be used for collaborative research and education.

2. Methods

2.1. Conceptual design of the collaborative framework

Development of SWATShare is motivated by the idea of creating a global repository of simulation models that can be shared with the community for enhancing hydrology education and research. When models that are meticulously created are shared with the broader community, the same models can be applied for various purposes, thus broadening their applicability and impact. As a prototype, SWATShare is developed for:

(i) publishing and sharing of SWAT models on the web including input data and related output files;
(ii) performing model simulation, sensitivity analysis and auto-calibration using Purdue Condor Pool and XSEDE distributed high performance computing resources (http://xsede.org);
(iii) visualizing SWAT outputs dynamically in both space and time at reach, sub-basin and watershed scale.

In addition to research, SWATShare can also be used as an online educational tool for students to understand hydrologic systems under different geographic and climatic settings by using the shared SWAT models. Successful application of web tool like

SWATShare as a collaborative environment for research and education is dependent on proper handling of model communication, access and persistence (Borowski and Hare, 2007; Loucks et al., 1985). A brief discussion of what this means within the context of SWATShare is provided below.

2.1.1. Model Communication

'Model Communication' refers to a suitable user interface and input/output visualization which makes a model easier to understand. SWAT users vary in levels of expertise and training, and have different end objectives. Creating a SWAT model not only requires the knowledge of all the technical details of the model processes (e.g. Neitsch et al., 2011), but it also requires expertise to correctly delineate and discretize a watershed, create input datasets, run and calibrate the model. Moreover, a successful model simulation creates large numbers of output files, in the range of thousands, which can be challenging to handle and visualize at different spatial and temporal scales. Many non-technical users generally take the model's pre-fabricated "black box" data pre-processing and modeling approach that is not easy to understand and appreciate (Ramsey, 2009; Voinov and Bousquet, 2010). This is when most distributed hydrologic models become ineffective as a policy generating tool (Borowski and Hare, 2007). SWATShare focuses on major hydrologic outputs that are easy to manage and comprehend over the web through an easily understandable interface. Without undergoing any of SWAT's modeling steps, a student or a policy maker can locate the target watershed over a map and view the outputs for that watershed. In this case, a user is accessing a model that is created by another user with the assumption that the shared model is perfect. However, successful implementation of model communication will also depend on tools for quality control, including tools providing comments, likes and dislikes to indicate the credibility of shared models.

2.1.2. Model Persistence

'Model Persistence' refers to continuous usage of a model through sharing and storage. In watershed management practices and research, multiple models get created for one geographic location at different times focusing on different purposes (Delipetrev et al., 2014; Horak et al., 2008). In most cases, a model remains confined within the scope of a particular project, without its further use for addressing other issues within the watershed by other researchers or stakeholders. Alternatively, various clones of a similar model for the same watershed can get proliferated without much difference in the model set-up. In some cases, the models developed by multiple users might actually be doing the same thing (Voinov and Bousquet, 2010). From this perspective, SWATShare is designed to enable a SWAT model persist on the internet and continue to grow by input from other users through its sharing and storing capabilities. This way, the "memory" of the project is maintained, which is essential for future learning and reuse of results, especially when the model's results are related to broader policy implications (Loucks et al., 1985). Enabling a CI with model sharing capability needs robust backend architecture capable of (i) capturing model information and mapping it to a metadata standard; (ii) tracking and storing different versions of the same model; and (iii) discovery of shared models and downloading options. It also needs a user interface that supports seamless flow of operations by minimizing intermediate steps of finding and using shared models. Implementation of these attributes is even more challenging for a model such as SWAT that is open source and involves hundreds of input files.

Other cyber-enabled SWAT modeling platforms such as the Basin Scale Hydrological Toolkit (BASHYT, Cau et al. (2013)) is significantly different from SWATShare in terms of sharing functionality. For example, BASHYT users can evaluate the outputs of an existing SWAT model, but the access to the model input files is limited only to the creator of the model. On the contrary, a user can not only store the model in SWATShare but can also make it public along with its input/output directory for other users to download. This approach potentially saves computational efforts and minimizes duplication. The creator and user of a particular SWAT model can collaborate to accomplish new research or educational objectives, thereby maximizing the model persistence. To ease the process of model discovery within SWATShare itself as well as among cross-platforms, the metadata structure in SWATShare is made more comprehensive compared to BASHYT. Consequently, SWATShare is being made interoperable with other cyber-enabled hydrologic information systems (or science gateways) such as HydroShare.

2.1.3. Model Access

'Model Access' does not only mean access to model outputs on the internet, but it also means platform independence, including the flexibility of remote modification of the model's spatial and temporal inputs. In the conventional desktop-based modeling environment, a policy maker is only superficially involved with the results (Janssen et al., 2008; Loucks et al., 1985), usually by extracting information from the models or the reports created by other experts and researchers. This approach limits an effective human-computer-model interaction and intervention (De Kok et al., 2008; Loucks and Fedra, 1987). Importantly, research as well as policy making is more about problem solving, and less about modeling. To make a platform effective as a decision support system and/or a scalable educational tool, the users should not be driven through a predefined loop of input pre-processing and modeling protocol (Black et al., 2014; McIntosh et al., 2011). Instead, in such a framework, a model should always evolve around the "access – refine – re-run" pathway (Loucks et al., 1985). This way, a single model for a particular geographic location can be manipulated and analyzed differently by an educator, a researcher or a watershed manager. In a web-based application, one particular model, if shared by the owner, can undergo such modifications simultaneously by multiple other users (Delipetrev et al., 2014).

SWATShare is platform-independent because only a web browser is needed to run simulations and view model outputs. In addition, SWATShare users can download and remotely modify any shared model. This modification does not necessarily alter the original model shared within the system, but it enables storing of different versions of the same watershed model so that a hierarchical history of model execution and/or evaluation can be maintained. For example, a policy maker can edit an existing SWAT model to test new policy alternatives and re-calibrate it to produce new outputs. Students can modify and re-run existing SWAT models to examine the effect of different management practices, landuse or climate change scenarios over a particular watershed or a region. This makes SWATShare an effective education tool for water resources research and decision support system in watershed management practices.

2.2. Software architecture

The software architecture for SWATShare is presented in Fig. 1. SWATShare is deployed on WaterHUB, which is built using the HUBzero technology (Mclennan and Kennell, 2010). HUBzero is an open source software framework for developing online scientific web sites. It provides various tools for scientific computing and collaboration out of the box, such as Group, Project, Publication, ticket tracking, forum, and automated process for users to contribute contents and online interactive tools via self-service. The

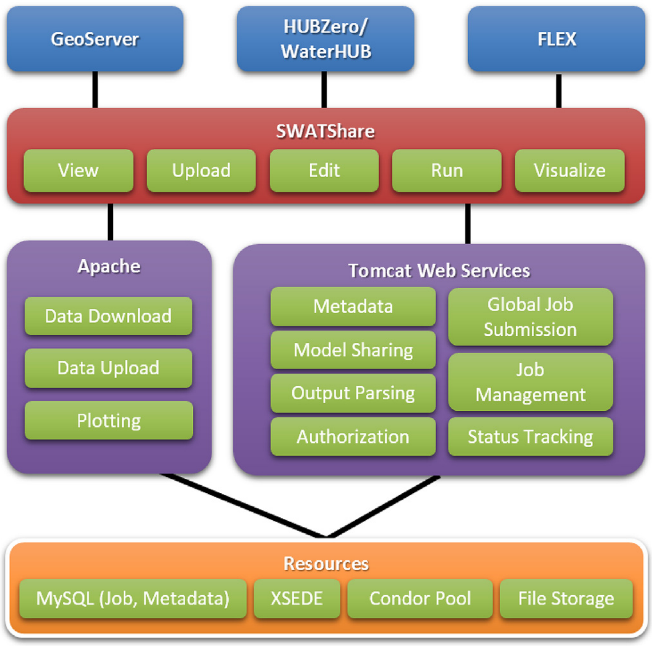

Fig. 1. SWATShare architecture.

SWATShare graphical user interface is developed using Flex software development kit (http://flex.apache.org/) that provides rich interactive user interface which is portable across different web browsers. Software components to develop SWATShare, which began in 2009 with limited funding, were selected based on in-house expertise, availability (open source), reliability, sustainability, portability and compatibility across different platforms. Some of the software tools used in SWATShare, for example, Flex, may not be the best options available today, but many existing systems use these tools. Specifically, Flex has been used in other recent web-based hydrologic applications, such as gSWAT (Bacu et al., 2011) and Great Lakes Water Level Dashboard (GLWD, Gronewold et al. (2013)). SWATShare's access control is managed through the HUBzero framework, i.e., a user needs to log in through HUBZero in order to upload, modify, or run SWAT models. This is necessary for managing each user's models and simulations in a private workspace, and for tracking resource usage, data sharing permission and website security. SWATShare's user interface supports five key functions: View, Upload, Edit, Run and Visualize.

GeoServer (http://geoserver.org/) is used for rendering an interactive map so that users can discover existing models based on geographic location and related metadata. GeoServer and MapServer are commonly used open source servers for displaying geospatial layers. For example, Brooking and Hunter (2013), Delipetrev et al. (2014) and Feng et al. (2011) applied GeoServer for managing geospatial data layers in their prototype web applications; while Cau et al. (2013) and Ames et al. (2012) used MapServer (http://mapserver.org/) and DotSpatial (http://dotspatial.codeplex.com/) for BASHYT and HydroDesktop, respectively. GeoServer was used in SWATShare for the following reasons: (i) in-house expertise from other projects; (ii) it is newer compared to MapServer, and is supported by larger user and developer community; (iii) it is easier to administrate; and (iv) has an easy-to-use REST API.

Behind the web front end, the SWATShare tool invokes SWATWS which includes a set of Tomcat web services (http://tomcat.apache.org/) to perform tasks based on user input, such as checking authorization for all user operations, saving new metadata to the backend MySQL database, composing and submitting simulations to an HPC resource via Globus, tracking the status of simulations, sharing a model, and parsing the model output into a common internal format appropriate for visualization. Currently, SWATShare submits simulations to the Purdue Condor pool and Purdue Carter cluster with options of submission to the NSF XSEDE HPC resources at the San Diego Supercomputer Center (SDSC) as well. To visualize the simulation output, SWATShare extracts data from the model outputs using Tomcat web services and then generates spatial and temporal graph plots using programs written in PHP and Python.

Model upload, download and visualization tasks are implemented as AMFPHP services (http://www.silexlabs.org/amfphp/) running in an Apache server. SWATShare currently supports only HTTP upload and download of model input/output with a size limit of 2 GB, which is the maximum size of a SWAT model's geodatabase. Alternatives are being considered to support upload of larger data/model using either secure FTP or Globus Online (https://www.globus.org/).

The model metadata structure in SWATShare follows the Dublin Core standard (http://dublincore.org/) of generic resource descriptions. SWATShare includes most major Dublin Core descriptors related to resource identification such as model name, simulation type, and model version, among others. In the current version, SWATShare automatically performs geo-referencing and identifies the approximate center point of the watershed, without asking any input from the user while uploading; whereas temporal coverage is collected from the duration of model simulation. In addition to these major metadata descriptors, a more structured way of collecting and storing model related metadata, including formalization of nomenclature such as model creator, contributor and publisher has been adopted from the HydroShare project.

SWATShare stores the metadata in a MySQL (http://www.mysql.com/) database (Fig. 1) and the metadata for available models gets extracted from the database using Tomcat web services. Many similar systems use some form of SQL databases for storing metadata and related information. For example, SQLite (http://www.sqlite.org/) is used for maintaining database in BASHYT (Cau et al., 2013) and in HydroDesktop (Ames et al., 2012), while PostgreSQL (http://www.postgresql.org/) and PostGIS (http://postgis.net/) have been the common choices in many other web-based platforms (e.g. Brooking and Hunter, 2013; Castrogiovanni et al., 2005; Delipetrev et al., 2014; Demir and Krajewski, 2013; Horak et al., 2008; Sun, 2013). MySQL is chosen for SWATShare because of in-house expertise for similar applications on HUBZero.

2.3. *Job submission and execution workflow*

The workflow for running SWAT simulations in SWATShare is shown in Fig. 2a. Once models (jobs) are uploaded, SWATWS dispatches the jobs to an available HPC resource, depending on the type of computation required. A SWAT simulation job goes through three phases, including pending, running, and cleanup (Fig. 2b). In the pending phase, the Globus credential and resource specification language (RSL) are created, which are then used to submit the SWAT job using Globus application program interface (API). In the running phase, a SWAT job manager is run to decompress the input, process the data according to the simulation type selected (normal, sensitivity analysis, or auto-calibration), and then invoke the SWAT executable for the simulation. After the simulation is successful, SWATWS collects all the output files and compress into a tar ball. In the final cleanup phase, the output file archive and log files are created and stored on the storage server.

Usually, normal simulation of a SWAT model is not computationally intensive even for a considerably large watershed. Use of HPC resources is more advantageous to run model calibrations, because one single calibration job for a SWAT model usually involves more

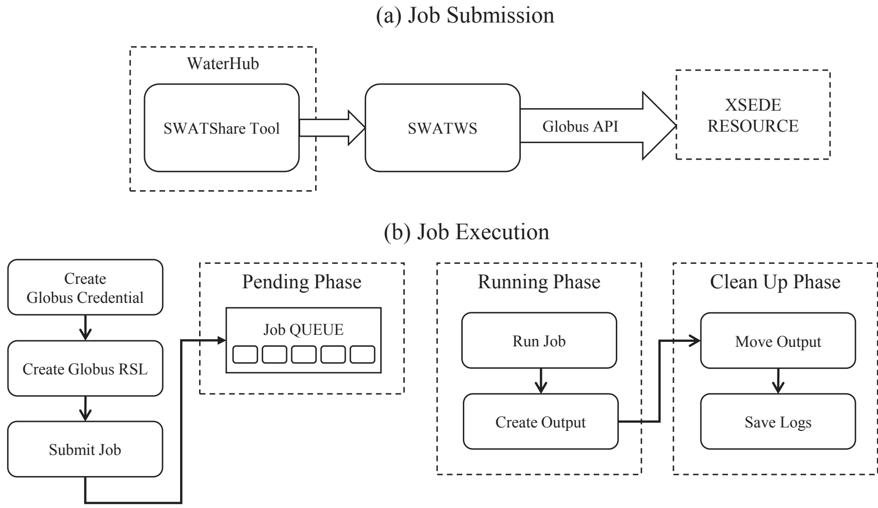

Fig. 2. Design of SWATShare job submission and execution workflow.

than 500 iterative simulations. In SWATShare, all the iterations of a job are submitted into one worker node of the HPC resources, which virtually works as a single machine. In contrast, gSWAT (Bacu et al., 2011) distributes the iterations of one job into multiple available nodes on their HPC resource (http://www.envirogrids.net/), hence speeding up the computation; then a post processing module collects the outputs from all the worker nodes and produces the final calibration result. This difference is mainly associated with the parameter optimization algorithm being employed in the respective systems. Regardless, both of these systems allow multi-user activity at any instant of time. gSWAT also allows online modification of calibration parameter list which is quite convenient to reach a desired level of optimization with more control on the user side. The ongoing efforts of incorporating new calibration algorithms in SWATShare will eventually enable parallel execution of a single job as well as online parameter selection/modification, leading toward more efficient and flexible work environment.

2.4. SWATShare interface

2.4.1. View

The view interface (Fig. 3) enables the discovery of all SWAT models in the system by using a global map interface and related metadata information. All models in SWATShare are categorized into three groups, including My Models, Shared Models and Other Models. My models include all the models that are owned by a user. A user can edit, modify, run and visualize all models in the My Models category. Shared Models lists those models in the system that are made public by their owners. A user can view metadata associated with a shared model and download it, but cannot directly edit or run a shared model. However, a shared model can be easily copied and owned to create another instance of the same model. A user can then make changes to the new instance of the shared model. Other Models lists those models that are published in the system, but not being made public. A user can only see limited metadata associated with these models including the owner's username, geographic location, and temporal frame, among others. If needed, a request can be sent to the owner to share the model offline for collaborative research and educational endeavors.

2.4.2. Upload

The Upload interface is used for submitting a SWAT model and its associated files in a compressed format (.zip or .tar). A model uploaded as a zip or tar file can include following items from the model's corresponding ArcSWAT project (Fig. 4): (i) ESRI ArcMap document that is used for creating the input files using ArcSWAT; (ii) project geodatabase, which is a Microsoft Access Database (.mdb) file with the same name as the ArcMap document but with an .mdb extension; (iii) Watershed folder containing the shapefiles; (iv) Scenarios folder with TxtInOut inside, which is the primary folder for SWAT simulations and output storage; (v) RasterStore folder and (vi) RasterStore geodatabase. These six items are created automatically when a SWAT model is created by using ArcSWAT (Neitsch et al., 2011). While the items listed above are desirable to be uploaded in the system for publishing a SWAT model, there are no restrictions if a user wants to include less or more items except the Watershed and Scenarios folders as these two are necessary for simulation and visualization. Even though the description presented here refers to the ArcSWAT structure, a SWAT model created by any other pre-processing software such as QSWAT or MWSWAT can also be uploaded as a zip file and used for HPC simulation and temporal visualization as long as it contains the TxtInOut folder. Other items besides TxtInOut are useful for users who may download the entire project and modify model inputs through the pre-processing software.

The .zip compressed format is used so that all files can be uploaded or download in a single step. This also enables cross-platform discovery of SWAT models between multiple systems such as SWATShare and HydroShare, both of which considers a model instance as a single resource containing model input/output and other contents in one compressed file. Additionally, the zip format allows the flexibility of including additional files such as project notes or related publication list that any other user may find useful for using or extending a particular SWAT model. If the model (zip folder) size is greater than 2 gigabytes, a user can remove the excess files from the model and provide instructions on getting the files through other means such as from an ftp location or from a shared drive.

Once the zip or tar file is uploaded, SWATShare automatically extracts the primary metadata related to model simulation processes, such as runoff/routing calculation methods, number of subbasins/HRUs in the model, simulation time step/duration, SWAT version etc. (step 1 in Fig. 4). The model creator also needs to provide some secondary information (step 2) that cannot be automatically extracted including model name and description, creator/contributor of the model, DEM source and resolution, landuse and soil data source, simulation type as well as the

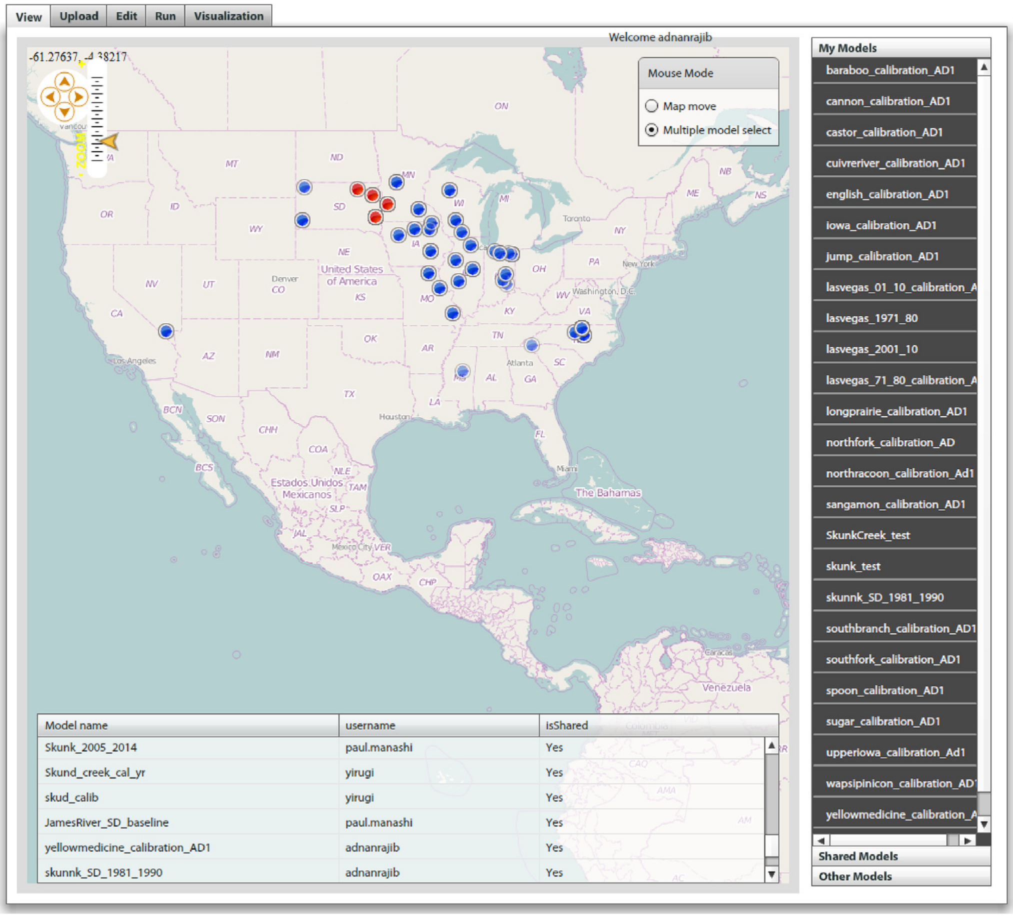

Fig. 3. SWATShare View interface. Although models available only within the USA are displayed here, SWAT models for any watershed around the world that has been shared into the system can be selected for download/visualization.

management practices (crop rotation, tillage, tile drainage etc.) being incorporated in the model. Others can find these metadata useful in selecting a model for their application and necessary offline modification after downloading the model. Information of the creators/contributors associated with a particular model ensures provenance as the model might continue to grow inside the system by ingesting inputs from multiple users over time.

2.4.3. *Edit*

The Edit interface is similar to the Upload interface, but it is primarily used to edit metadata or change the input data for an existing model. This interface also enables copying any shared model to user's personal account for the purpose of performing simulations and visualizations. A shared model cannot be used for simulation and visualization unless it is owned by the new user. This is an "indirect security measure" for maintaining the integrity of the original model. Copying of shared models leads to existence of multiple variations of SWAT models for the same watershed. Users can identify all the variations of a particular model that resulted from the modifications done by different users by looking at the metadata in the View interface.

2.4.4. *Run*

The Run interface (Fig. 5) can be used to perform a regular SWAT simulation, sensitivity analysis or auto-calibration. The run interface allows users to track the status of their jobs, access log files for debugging, and download the model output after a successful simulation. After a simulation job is submitted, it enters the simulation workflow described in Section 2.3, and the Status column in the Run interface displays the current status of the simulation. The meaning of each status is explained in Table 1. When the job status of a model changes to DONE, its outputs can be visualized by using the Visualization interface.

2.4.5. *Visualization*

The Visualization interface interactively generates temporal plots and spatial maps based on model outputs. Figs. 6 and 7 show sample spatial and temporal visualizations, respectively. The present version of SWATShare supports three visualization plots, including: (i) time series plots of watershed-average outputs; (ii) time series plots of reach (stream) outputs at the watershed outlet; and (iii) spatial maps of average values at the sub-basin level. The visualization of model outputs is enabled at the time-step at which the model is executed in the system and also at larger time-steps if the model is simulated at daily or monthly time-steps. For example, if the model is executed at daily time-step, users can also view monthly or annual plots. If the model is executed at monthly time-step, users can also view annual plots.

Fig. 6 shows an example of spatial maps for simulated surface runoff at monthly time-step. In all spatial visualization maps, a corresponding time series plot of precipitation, which is the major driving variable for all hydrologic processes, is also created. A plot of precipitation helps to correlate the spatial variation of different hydrologic variables with incident rainfall amount through time. Fig. 7 shows temporal plots of various variables, and as seen on the left hand side, users have the option of selecting multiple variables,

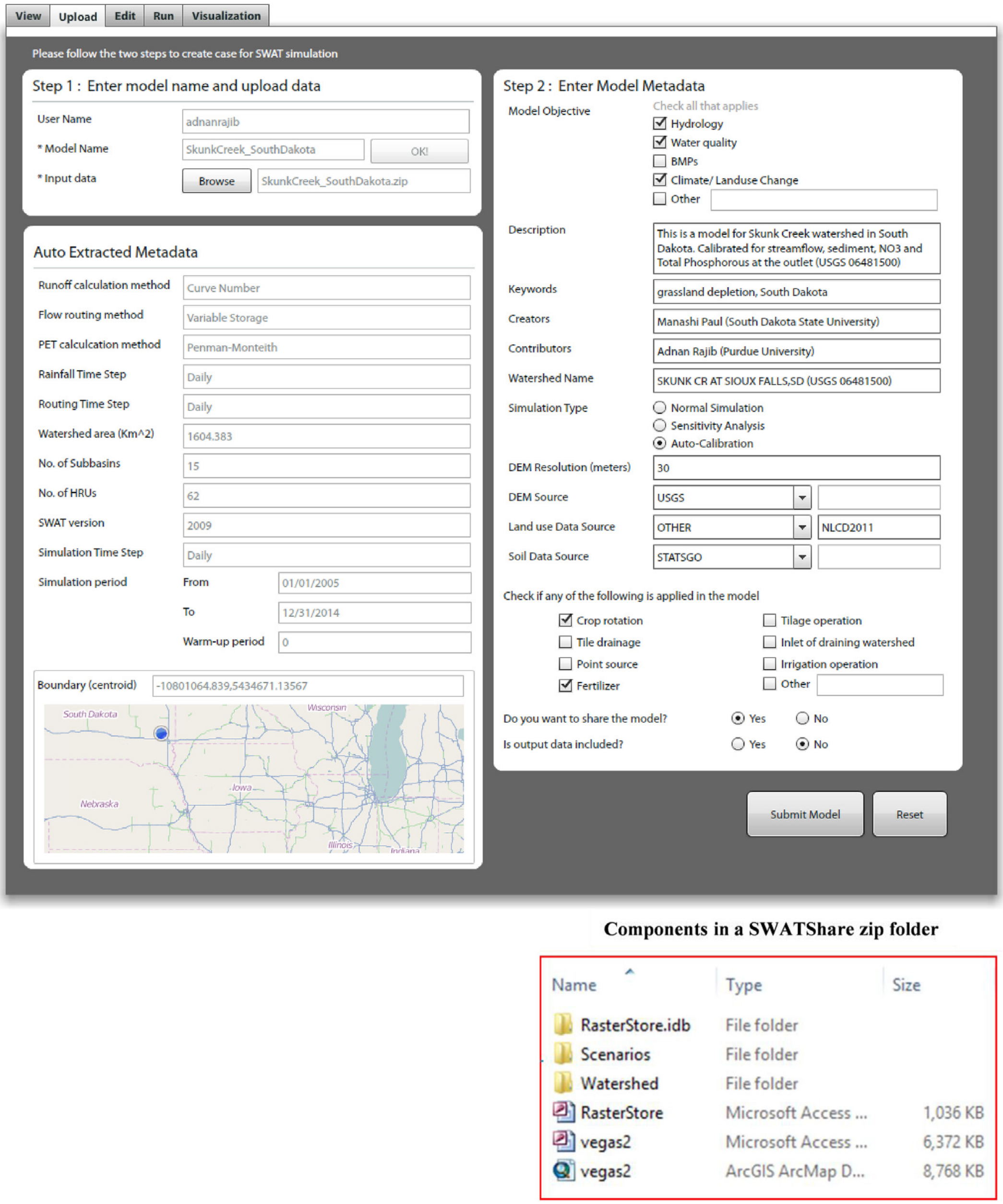

Fig. 4. SWATShare Upload interface along with the typical components in a SWATShare model zip folder. Information filled in step 2 can be considered as a standard example. Edit interface is apparently similar to this Upload interface. Detail instructions for using the Edit interface are provided in SWATShare user manual.

and all plots are displayed in one single interface. Having plots of multiple variables in one interface helps understanding the mutual variability of different hydrological processes through time. The visualization interface also allows users to download the spatial maps and time-series values of the selected output variables. The current spatial visualization interface is based on the shapefiles created using ArcSWAT pre-processing software, but work is ongoing to support spatial file formats created by other tools such as QSWAT and MWSWAT. Temporal visualization is compatible regardless of the software used in model creation, as long as there is a TxtInOut folder in the uploaded zip file.

3. Software implementation and case studies

SWATShare is designed to achieve communication, persistence and accessibility merits in terms of platform independence, as well as model sharing, high performance computing and dynamic visualization capabilities. In order to validate the fact that all these attributes of SWATShare work effectively when deployed in practical scenarios, three case studies are presented in this section. The first case study involves SWATShare implementation in a class room setting, and the second and third case studies involve the use of SWATShare for accomplishing research objectives. As a whole, these case studies demonstrate SWAThare's potential as a CI for collaborative hydrology research and education.

3.1. Case study 1 — educational application through model sharing

In this case study SWATShare's functionality as an educational tool is evaluated through performing parallel operations by multiple students in a class room setting. This case study involved 19 students in CE-54900: Computational Watershed Hydrology course

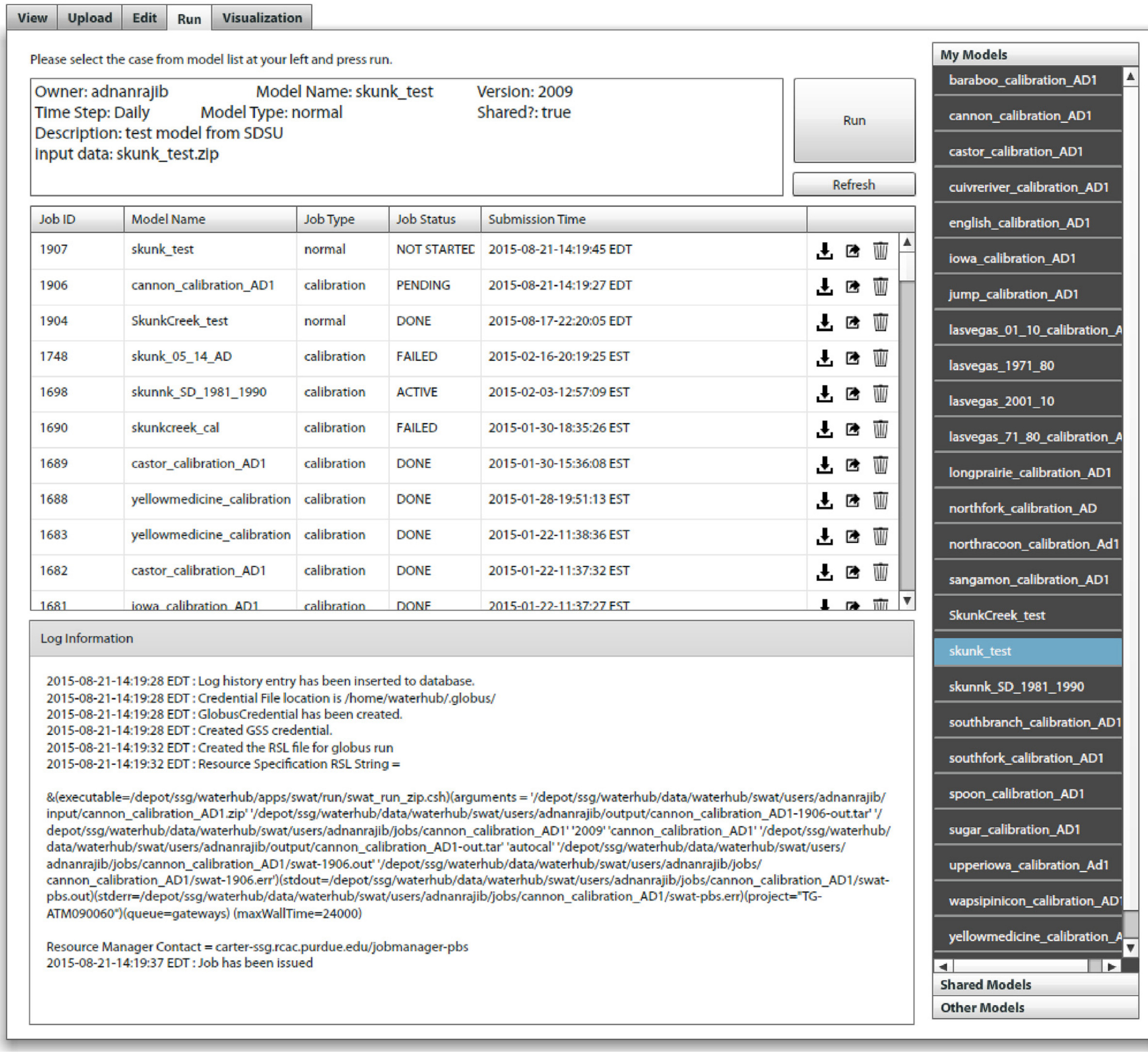

Fig. 5. SWATShare Run interface with examples of different job status.

taught at Purdue in Spring 2014. The objective of this course, which is primarily offered to graduate and upper level undergraduate students, is to teach students how to use computational tools for hydrologic data management and simulation. Using the skills learned in the class, each student was required to create a SWAT model for a different watershed in the Upper Mississippi River Basin (UMRB), thus creating a total of 19 different SWAT models with same pre-defined stream network thresholds. The watersheds given to the students (shown in Fig. 8) had variable drainage areas ranging from 600 to 3500 km^2.

After creating a SWAT model, all students published their models in SWATShare for parameter sensitivity analysis, auto-calibration, and visualization. Publishing a SWAT model included uploading the model in the .zip form and sharing it with other students in the class. The calibration jobs were executed in three batches. In the first batch, only one student performed the auto-calibration in SWATShare. Upon completion of the job from the first batch, six students simultaneously ran auto-calibration in the second batch. Finally, after jobs from second batch were finished, the remaining twelve students simultaneously ran auto-calibration in the third batch. The details about the watersheds including their drainage areas and outlet locations are presented in Table 2. The average watershed area in all three batches was around 1600 km^2, with a common simulation period of 2004–2010.

Table 1
SWATShare job status descriptions.

Status	Description
Not started	The job has been submitted but the tool has not been able to get a status update from the backend computation resource
Failed	There is some error with the job. It could be because the job failed to be submitted or the execution failed at run time
Pending	The job has been submitted to the backend computation resource, and is waiting in the queue
Active	The job is running
Done	The job is done

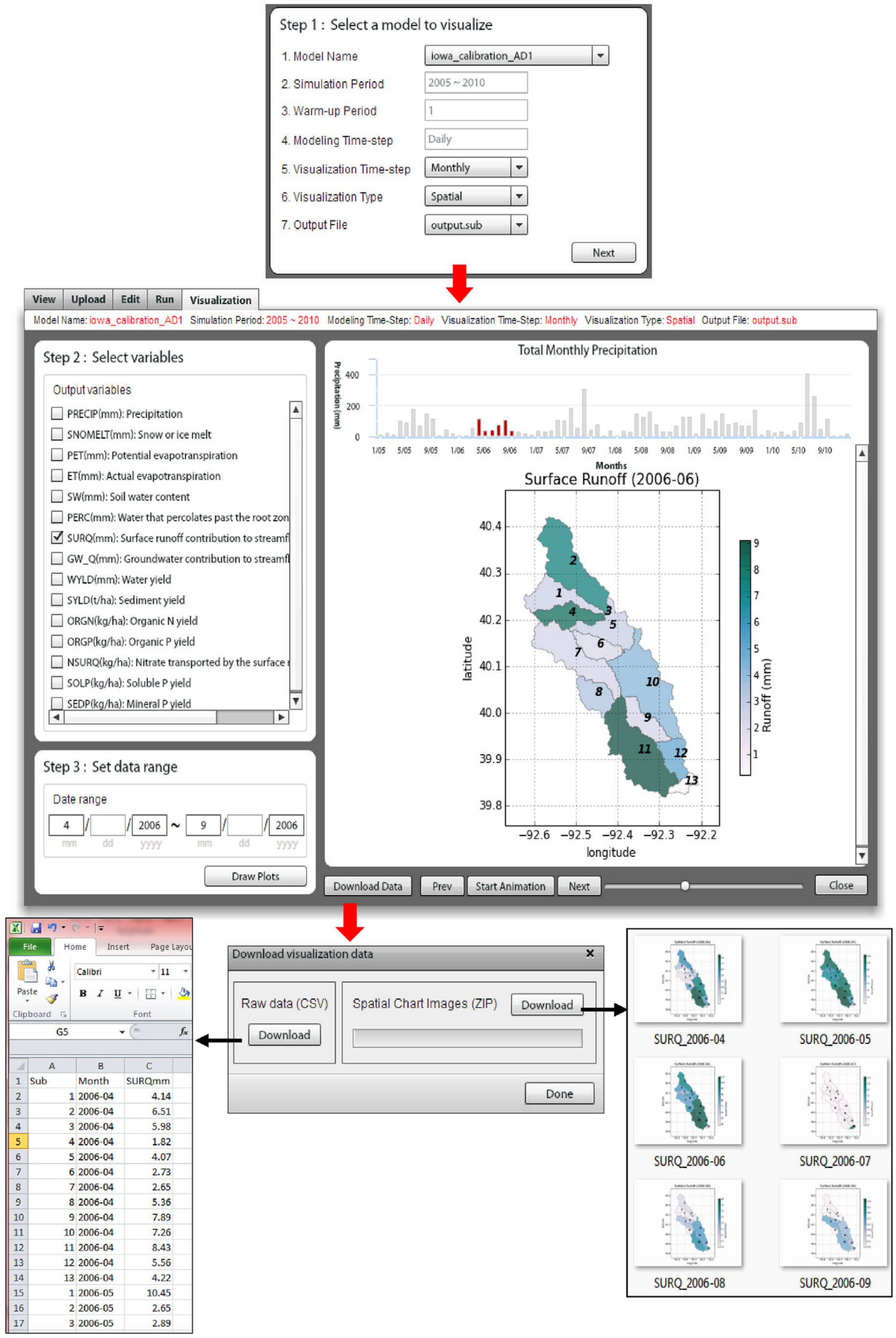

Fig. 6. SWATShare Visualization interface (creating spatial maps). This interface needs to be activated by selecting a model (from user's personal account), visualization type ("spatial" for this figure) and visualization time-step. Output images and corresponding simulated results can be downloaded to local computer directory.

The objective of this batch execution in a classroom setting was to study SWATShare's performance under concurrent multi-user activity and increasing workload. Table 3 compares the average submission queue (pending) time and average response (pending, running and cleanup) time as the two performance indicators, when jobs in each batch were submitted to Purdue clusters only. The differences in pending time were negligible irrespective of the number of models being submitted together (Table 3). The pending time in a traditional cluster system varies depending on a number of factors, including the number of jobs in the queue, jobs running on the cluster at that instance of time, wall clock limit and the scheduling algorithms of the queue, to name a few. Therefore, it is hard to compare pending times between single and multi-user scenarios. The running time is mostly effected by the I/O speed of the worker node, because the SWAT simulation involves huge file read operations across thousands of small input files (typically around 4000 files of 4 KB each). A simple experiment showed that it takes about 2.5 s to read a 4 KB file for 1000 times on the distributed parallel file system of Purdue clusters. So, an auto-calibration job with 20,000 iterations would require approximately 50–60 h, which is comparable to the result for batch 1 as shown in Table 3. With the expectation that multiple jobs will increase file I/O

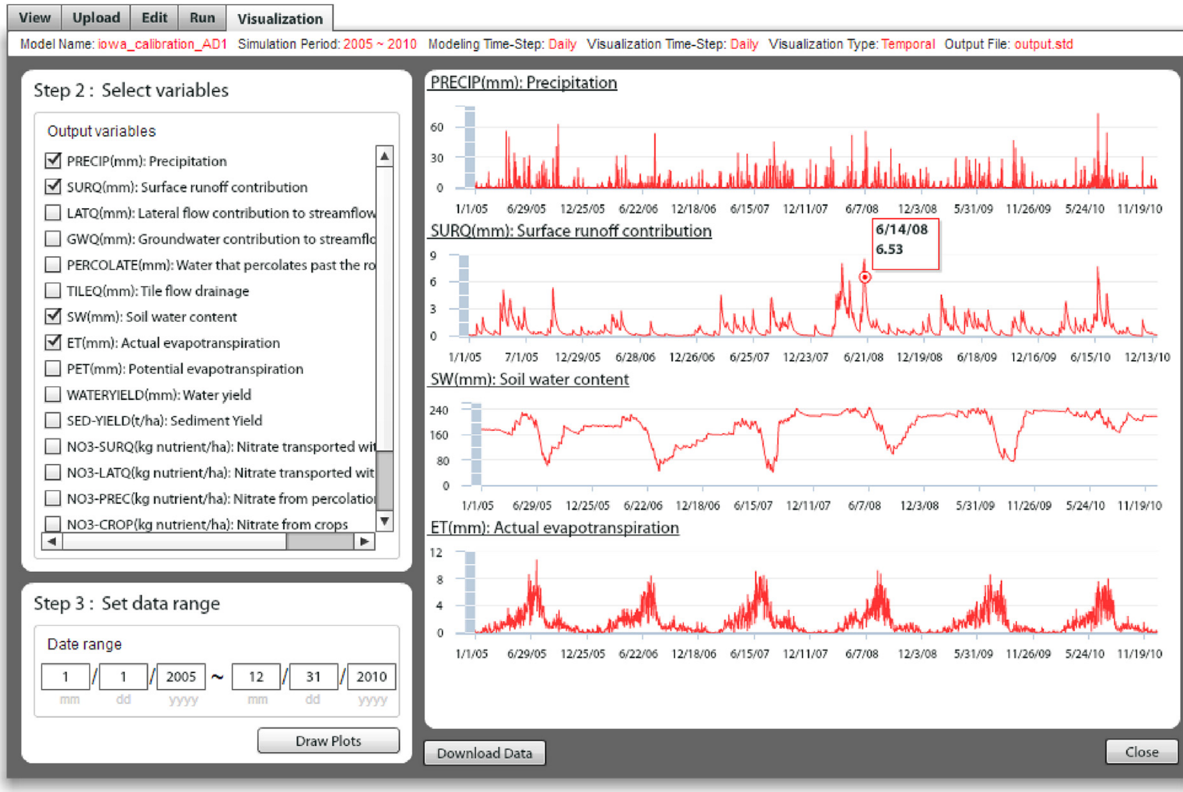

Fig. 7. SWATShare Visualization interface (creating temporal plots). Similar to spatial plots, multiple variables can be selected (from the left side panel) and seen in one window by scrolling down. Detail instructions for using the Visualization interface are provided in SWATShare user manual.

overhead, the running time (and hence, the total response time) increased for the second (6 models) and third (12 models) batches, when multiple students submitted jobs nearly at the same time and ran auto-calibration simultaneously. Similar average response times for the second and third batches indicate that the workload was evenly distributed, thereby validating SWATShare's ability to provide consistent performance under variable workload situations.

This particular classroom application of SWATShare also validated its potential as an effective toolkit for hydrology education.

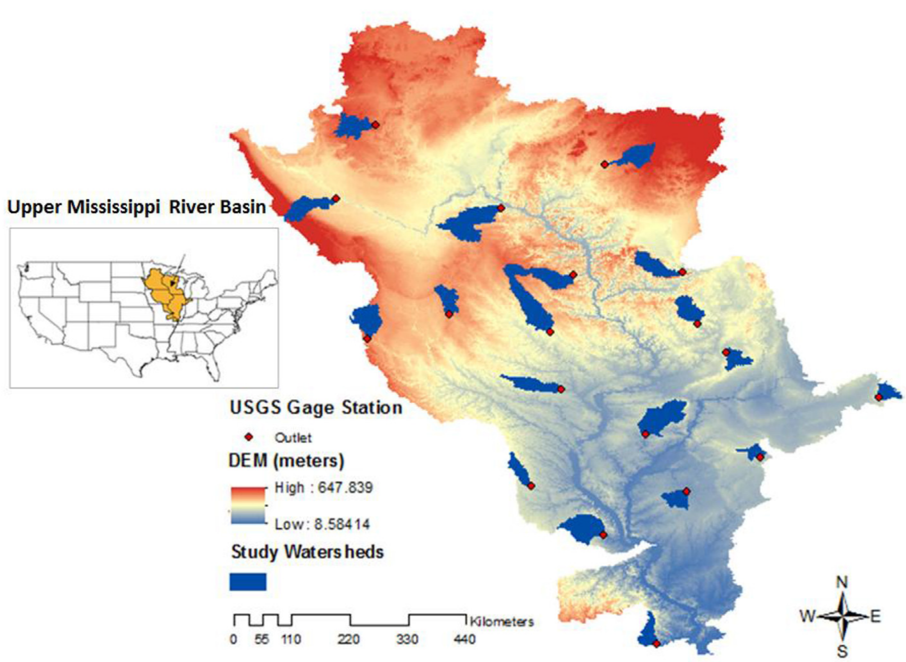

Fig. 8. Study watersheds for multi-user testing of SWATShare in real-time class room environment. All the watersheds belong to the Upper Mississippi River basin in USA.

Table 2
Watersheds in Upper Mississippi River basin for testing SWATShare's multi-user performance in a clasroom.[a]

No.	Batch[b]	USGS ID	HUC ID	Name	State	Outlet location	Area (km^2)
1	1	05405000	07070004	Baraboo River near Baraboo	Wisconsin	Lat 43.48°, Long 89.64°	1577
2	2	05502300	07110005	North Fork Salt River at Hagers Grove	Missouri	Lat 39.83°, Long 92.23°	945
3		07021000	07140107	Castor River at Zalma	Missouri	Lat 37.15°, Long 90.08°	1096
4		05449500	07080207	Iowa River near Rowan	Iowa	Lat 42.75°, Long 93.62°	1111
5		05436500	07090004	Sugar River near Brodhead	Wisconsin	Lat 42.6°, Long 89.4°	1355
6		05421000	07080102	Wapsipinicon River at Independence	Iowa	Lat 42.46°, Long 91.9°	2714
7		05569500	07130005	Spoon River at London Mills	Illinois	Lat 40.71°, Long 90.28°	2776
8	3	05570910	07130006	Sangamon River at Fisher	Illinois	Lat 40.31°, Long 88.32°	622
9		05516500	07120001	Yellow River at Plymouth	Indiana	Lat 41.34°, Long 86.3°	761
10		05439500	07090006	South Branch Kishwaukee River near Fairdale	Illinois	Lat 42.1°, Long 88.9°	1002
11		05245100	07010108	Long Prairie River at Long Prairie	Minnesota	Lat 45.98°, Long 94.87°	1124
12		05455500	07080209	English River at Kalona	Iowa	Lat 41.47°, Long 91.71°	1487
13		05362000	07050004	Jump River at Sheldon	Wisconsin	Lat 45.31°, Long 90.96°	1492
14		05313500	07020004	Yellow Medicine River near Granite Falls	Minnesota	Lat 44.72°, Long 95.54°	1725
15		05482300	07100006	North Raccoon River near Sac City	Iowa	Lat 42.35°, Long 95°	1813
16		05388250	07060002	Upper Iowa River near Dorchester	Iowa	Lat 43.42°, Long 91.51°	1994
17		05576000	07130007	South Fork Sangamon River near Rochester	Illinois	Lat 39.74°, Long 89.57°	2246
18		05514500	07110008	Cuivre River near Troy	Missouri	Lat 39.53°, Long 90.98°	2339
19		05355200	07040002	Cannon River at Welch	Minnesota	Lat 44.56°, Long 92.73°	3471

[a] Corresponding SWAT model for each of these watersheds was created by students in the CE549 course during the Spring 2014 semester at Purdue University.
[b] Set of model(s) ran simultaneously in SWATShare.

Students in the class utilized SWATShare's multi-user multi-tasking capacity, and were able to calibrate their full-scale SWAT models even staying within the time-constraints of a classroom environment. Further, students used the outputs from their own model as well as from other shared models to study the effects of drainage area, land use type and climate conditions on various hydrologic components at watershed scale. Within the context of SWATShare, students were able to see the changes in hydrologic response and relate these changes to watershed conditions in the Upper Mississippi region just by analyzing the model results shared in SWATShare by their fellow class-mates, without spending time in creating additional models. These types of exercises through a cyber platform such as SWATShare can help improve students' scientific understanding and analytical ability on various cause-and-effect relationships in hydrologic cycle, thereby closing the gap between textbook and real world applications in hydrology education (Aghakouchak and Habib, 2010; Merwade and Ruddell, 2012).

3.2. Case study 2 – collaborative research through model sharing

This case study shows how shared SWAT models in SWATShare can be reused to address new research questions. Fig. 9 shows a scenario, where a model created by user A is extended and/or reused by Users B and C. In this scenario, User A created a SWAT model for the Las Vegas Wash valley for the period of 1971–1980 with an intended purpose of estimating ground water yield. Las Vegas Wash valley is a semi-arid watershed situated within the states of Nevada and California, USA having a drainage area of 5500 km^2. This area has been undergoing massive urbanization since 1970s. Chambers et al. (2013) report 230 km^2 of new urban development in just six years from 2001 to 2007 in Clark County, Nevada. Major input data sets associated with User A's model were (i) landuse for 1970–1980 (Price et al., 2006); (ii) 1:250,000 scale State Soil Geographic Data (STATSGO) included within SWAT2009 database; (iii) 30 m resolution digitation elevation model (DEM) from USGS National Elevation Dataset in pre-packaged arc-grid format (USGS-NED, 2013); (iv) average daily precipitation, as well as the maximum and minimum daily temperature data covering the study period from a National Climatic Data Center (NCDC) climate station at the McCarran International Airport; and (v) observed mean daily streamflow data for the USGS 09419700 gage station.

User A created the SWAT model using ArcSWAT2009 in the desktop environment, and uploaded it on SWATShare for calibration using XSEDE resources. After calibrating the model and analyzing the results, User A then published the model in SWATShare. User B is another researcher who is interested in looking at recent sediment yield between years 2001–2010 in the same watershed. User B discovers User A's model in SWATShare, downloads the model and performs necessary modification to the model offline without duplicating the SWAT modeling process. The modification involved incorporating a different landuse raster NLCD 2006 (USGS-NLCD, 2013), and using the streamflow and weather data for the 2001–2010 time period. User B uploaded the new version of the model in SWATShare for calibration using this new set of data. After calibrating the model and analyzing the results, User B published and shared the model in SWATShare.

Table 3
Comparison of pending time and response time of SWATShare in a real time multi-user scenario.[a]

Number of models submitted simultaneously	Average pending time[b] (min)	Average response time[c] (h)
1	1.2	63.2
6	1.1	81.5
12	0.9	75

[a] For this particular case, jobs were submitted to Purdue clusters only.
[b] Pending time = submission queue time.
[c] Response time = pending, running and clean up time.

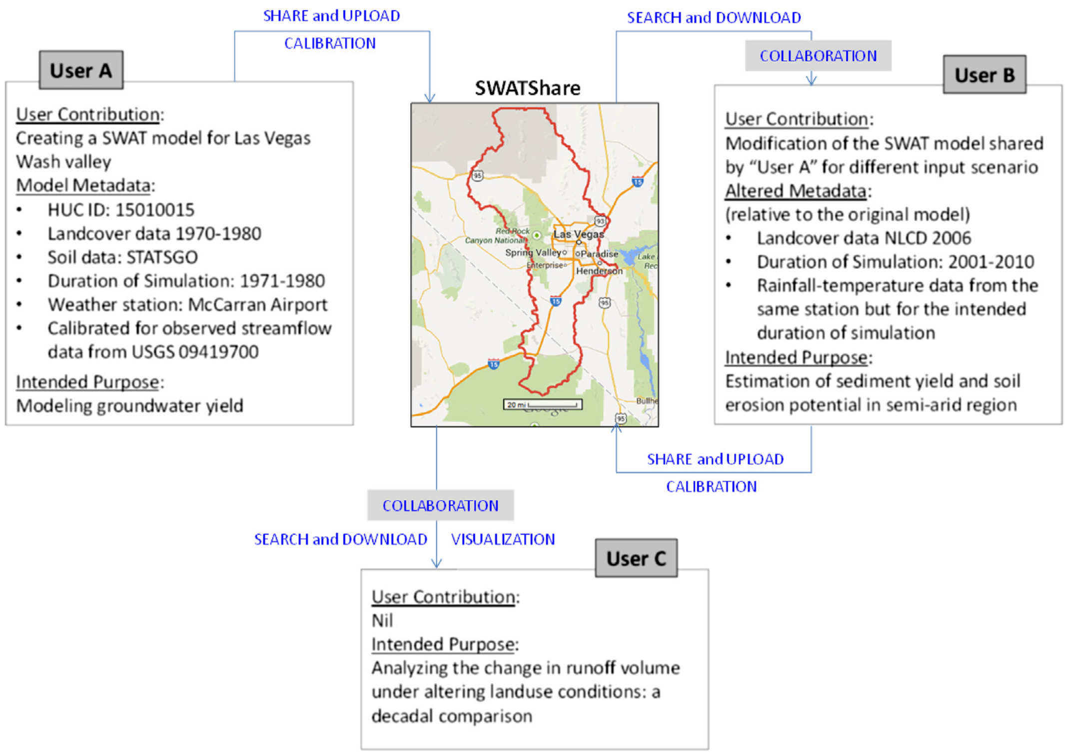

Fig. 9. Example of collaborative research utilizing SWATShare functionalities.

User C is interested in investigating the change in runoff volume under the rapidly altering landuse conditions in the Las Vegas Wash valley using a SWAT model. User C discovers both the SWAT models contributed by Users A and B in SWATShare. The metadata, such as land use source and simulation time period associated with SWAT models from both users A and B show that these are ideal for answering User C's research questions. User C copies the models from Users A and B and uses SWATShare visualization tool to create spatial maps of runoff volume for those two models (Fig. 10). The spatial maps in Fig. 10 indicate significant amplification of runoff volume in recent times, particularly in the vicinity of the Las Vegas city area. Detailed discussion of the results shown in Fig. 10 is beyond the scope of this paper.

This case study shows that how someone, User C in this case, can derive quantitative information, or get answers to research questions without even getting involved in any sort of online/offline modeling activity. User C is able to create new information from the existing models without any extensive modeling efforts, which are time consuming and cumbersome for models such as SWAT. Overall, the example discussed herein resembles an ideal case for SWATShare possessing its model communication, model persistence and model access properties as described in Section 2.

3.3. Case study 3 — simultaneous simulations of multiple SWAT models

SWATShare can be used as a computationally efficient platform for performing parallel calibration of multiple SWAT models. Calibration of SWAT models is a time consuming task which can take anywhere from days to weeks on a personal computer depending on the size of the watershed, division of the watershed in terms of number of sub-basins and hydrologic response units in SWAT, the number of parameters to be optimized and of course, the available computational resources. This makes simultaneous running of multiple calibration jobs in a standard desktop computer nearly impractical. Alternatively, any researcher, who wants to calibrate multiple SWAT configurations for the same watershed or calibrate SWAT models for multiple watersheds, can simply upload all the models in SWATShare and run the calibration jobs online. Even though the HPC/XSEDE resource used by SWATShare does not use any parallel version of the SWAT program, it can perform simulations of multiple SWAT models simultaneously because each submitted model (job) is run from a separate processor (Epema et al., 1996). Hence, the system does not wait for each of the submitted jobs from a particular user to finish in order to initiate the next simulation. Obviously, there can be several users active in SWATShare at any instance of time, each of them running multiple sensitivity/calibration jobs in the system. Such simultaneous execution of multiple jobs are documented in Kumar and Merwade (2009) and Gitau et al. (2011), who used an earlier version of SWATShare to study parameter uncertainty and management alternatives by creating different SWAT model configurations. Especially, Gitau et al. (2011) performed 43,000 SWAT simulations to evaluate the impacts of 172 different watershed management decisions combined with weather uncertainty on water quality. This activity would have taken 30 months on a desktop, but it took only 18 days using the SWATShare CI.

4. Summary and discussion

The work presented in this paper proposes a web-based, flexible and easy-to-use platform in which existing SWAT models get archived in a way that would make them more accessible. These shared models can be modified, improved and re-calibrated, if required. The three case studies presented in the paper demonstrate the potential of SWATShare as an educational tool, research tool and collaborative platform. SWATShare can also be used as a decision-support system where policy makers and watershed

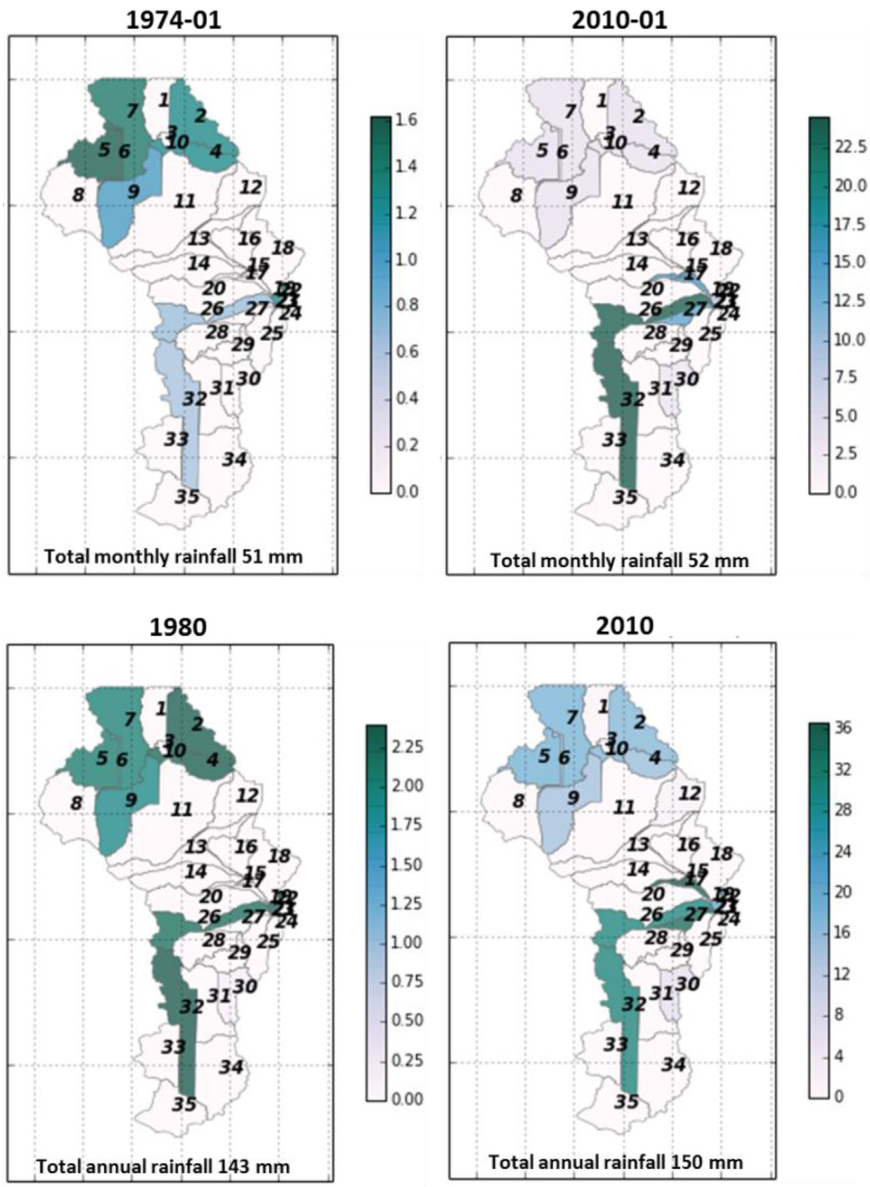

Fig. 10. Comparison of surface runoff volume (mm) in the Las Vegas Wash valley watershed at two different periods (past and recent). Maps at the top and bottom show monthly (January) and yearly outputs, respectively. Specific periods are chosen such that they have almost equal rainfall amount. The corresponding SWAT models can be found in SWATShare. Numbers within the maps indicate SWAT sub-basin IDs.

managers can access existing models, view their outputs for different spatial and temporal domains to get an assessment of different policy/management alternatives without looping through the repetitive modeling efforts.

Through the ongoing HydroShare project, more structured way of naming the models and metadata collection are currently under development to enable enhanced interoperability of SWAT models with other generic resources within the hydrology field. Similarly, collecting and storing of information related to models' journey within the system from one user to another and related modifications is needed to trace the models through provenance. Having a system such as SWATShare also involves addressing of issues related to different versions of a model. For example, SWAT model has undergone three major version changes in the last five years from SWAT2005 to SWAT2009 to SWAT2012. While the source code is relatively more stable, the pre- and post-processing tools such as ArcSWAT and auto-calibration routines are more affected through these version changes. Currently, SWATShare hosts the SWAT executable for all versions, but a suite of high performance parallel computation-based optimization methods will be added in the future which can handle any of the SWAT versions and performs model calibration within a very short period of time. This will also enable faster output generation for large-scale models, which happens to be a major concern for the SWAT user community in recent times.

While the prototype CI proposed in this paper is focused on SWAT model, the overall framework of sharing, publishing and execution of models can be extended to any other model. In fact, extending this framework to other commonly used public domain models such as HEC-HMS or HEC-RAS is relatively easier because these models are less complicated compared to SWAT. Conceptually, any model can be published or shared using the current CI without any changes because all the models are stored as individual zip files. The system will allow sharing of any zip file. However, for submitting simulations and creating visualizations, the system does check for specific folders, files and associated metadata stored in the zipped folder. Therefore, in order to extend the system to other models, a few changes or modifications need to be made to the system. The current metadata stores only the model's version, and to accept new models, the name of the model must also be stored. As a part of the HydroShare project, there are efforts to develop specific metadata templates for other commonly used models. Once the name of the model is specified, the system can display a form or invoke a service to extract the metadata associated with that specific model.

When a user downloads a shared model, makes offline modification and again uploads the model in SWATShare, it becomes a separate model instance. SWATShare stores/documents all the versions of a SWAT model, being created over the same watershed. Besides automatically extracting major metadata associated with the model, SWATShare allows user to input additional information on what modifications have been made on the original model and also list the names of the original model creators/contributors. For a selected geographic location in the View interface, a user has to browse over these metadata to manually filter through various models or various versions of the same model to select a suitable one. Despite this capability, the system is not fully capturing the provenance in the modeling efforts. Therefore, explicit documentation and/or linking of each model with its subsequent modified versions will be more beneficial. From the quality control point of view, the HUBZero platform, which hosts SWATShare, has a forum for SWATShare users where users can provide comments, likes and dislikes to indicate the credibility of shared models. Any user who has a login is able to access the forum.

Currently the view interface enables discovery of SWAT models primarily through geographic location. To search for different types of models in the system, advanced search tools that perform both spatial and textual queries will have to be used. Similarly, the execution of models and visualization of output is customized for SWAT. To make the visualization interface interoperable, it is being restructured to work on a standard database. When additional models are included, tools need to be developed to map their output files to this standard database.

The current system allows any user to look at limited metadata of any shared model, and download the model including its output files without requiring a login. However, most other functions related to metadata editing, model execution and visualization requires a login. Some of the functions such as visualization of shared model outputs could potentially be made public without requiring a login. This will make the system more accessible and provide opportunity to evaluate existing shared models without requiring a new user to login.

While there is additional work involved in making the proposed CI work with different models, several ongoing efforts including HydroShare will make adoptability of SWATShare easier than it seems at this point. For example, development of standards for model metadata, input and output files and model will make a system such as SWATShare to interact with any model in the future. In addition, many models are becoming OpenMI compliant, which in turn will enable integration of models much easier in the future. A system such as SWATShare can serve as a repository of models and serve as a catalyst in integrating multiple models at one place by exploiting the metadata and input/output files that reside in the same system. Acknowledging that some of these efforts are down the road, the development and implementation of model sharing system using a relatively complex model such as SWAT demonstrates the versatility and applicability of such a system to advance hydrologic research and education.

This paper provides the technical background on SWATShare and how it can be used for education and research through three case studies. Readers are advised to follow the SWATShare user manual that describes the interface and its functionalities in details, also providing instructions on model pre-processing steps to perform different types of simulation. SWATShare and its user manual can be accessed online at: https://mygeohub.org/groups/water-hub/swatshare.

Acknowledgment

This work is funded by the National Science Foundation (#OCI-1041379 and #OCI-1148090). Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation. The authors greatly appreciate the patience and efforts by the students in Spring 2014 CE 54900 class for using SWATShare for their term project.

References

Aghakouchak, A., Habib, E., 2010. Application of a conceptual hydrologic model in teaching hydrologic processes. Int. J. Eng. Ed. 26, 963–973.

Ames, D.P., Horsburgh, J.S., Cao, Y., Kadlec, J., Whiteaker, T., Valentine, D., 2012. HydroDesktop: web services-based software for hydrologic data discovery, download, visualization, and analysis. Environ. Model. Softw. 37, 146–156. http://dx.doi.org/10.1016/j.envsoft.2012.03.013.

Arnold, J., Moriasi, D., Gassman, P., Abbaspour, K., White, M., Srinivasan, R., Santhi, C., Harmel, R.D., Griensven, A. Van, 2012. SWAT: model use, calibration, and validation. Trans. ASABE 55, 1491–1508.

Bacu, V., Mihon, D., Rodila, D., Stefanut, T., Gorgan, D., 2011. Grid based architectural components for SWAT model calibration. In: HPCS 2011 – International Conference on High Performance Computing and Simulation, 4–8 July, Istanbul, Turkey, pp. 193–199. http://dx.doi.org/10.1109/HPCSim.2011.5999824.

Black, D.C., Wallbrink, P.J., Jordan, P.W., 2014. Towards best practice implementation and application of models for analysis of water resources management scenarios. Environ. Model. Softw. 52, 136–148. http://dx.doi.org/10.1016/j.envsoft.2013.10.023.

Borowski, I., Hare, M., 2007. Exploring the gap between water managers and researchers: difficulties of model-based tools to support practical water management. Water Resour. Manag. 21, 1049–1074. http://dx.doi.org/10.1007/s11269-006-9098-z.

Brooking, C., Hunter, J., 2013. Providing online access to hydrological model simulations through interactive geospatial animations. Environ. Model. Softw. 43, 163–168. http://dx.doi.org/10.1016/j.envsoft.2013.01.011.

Bürger, C.M., Kollet, S., Schumacher, J., Bösel, D., 2012. Introduction of a web service for cloud computing with the integrated hydrologic simulation platform ParFlow. Comput. Geosci. 48, 334–336. http://dx.doi.org/10.1016/j.cageo.2012.01.007.

Castrogiovanni, E.M., La Loggia, G., Noto, L.V., 2005. Design storm prediction and hydrologic modeling using a web-GIS approach on a free-software platform. Atmos. Res. 77, 367–377. http://dx.doi.org/10.1016/j.atmosres.2004.12.015.

Castronova, A.M., Goodall, J.L., Ercan, M.B., 2013. Integrated modeling within a hydrologic information system: an OpenMI based approach. Environ. Model. Softw. 39, 263–273. http://dx.doi.org/10.1016/j.envsoft.2012.02.011.

Cau, P., Bacu, V., Manca, S., Lehman, A., Muroni, D., Gorgan, D., 2013. An interoperable GIS oriented information and support system for water resources management. Int. J. Adv. Comput. Sci. Appl. 75–82. http://dx.doi.org/10.14569/SpecialIssue.2013.030309.

Chambers, J.C., Brooks, M.L., Pendleton, B.K., Raish, C.B. (Eds.), 2013. The Southern Nevada Agency Partnership Science and Research Synthesis: Science to Support Land Management in Southern Nevada. U.S. Department of Agriculture, Forest Service, Rocky Mountain Research Station. RMRS-GTR-3rd ed.

Culshaw, M.G., Nathanail, C.P., Leeks, G.J.L., Alker, S., Bridge, D., Duffy, T., Fowler, D., Packman, J.C., Swetnam, R., Wadsworth, R., Wyatt, B., 2006. The role of web-based environmental information in urban planning – the environmental information system for planners. Sci. Total Environ. 360, 233–245. http://dx.doi.org/10.1016/j.scitotenv.2005.08.037.

De Kok, J.-L., Kofalk, S., Berlekamp, J., Hahn, B., Wind, H., 2008. From design to application of a decision-support system for integrated river-basin management. Water Resour. Manag. http://dx.doi.org/10.1007/s11269-008-9352-7.

Delipetrev, B., Jonoski, A., Solomatine, D.P., 2014. Development of a web application for water resources based on open source software. Comput. Geosci. 62, 35–42. http://dx.doi.org/10.1016/j.cageo.2013.09.012.

Demir, I., Krajewski, W.F., 2013. Towards an integrated flood information system: centralized data access, analysis, and visualization. Environ. Model. Softw. 50, 77–84. http://dx.doi.org/10.1016/j.envsoft.2013.08.009.

Dong, C., Schoups, G., van de Giesen, N., 2013. Scenario development for water resource planning and management: a review. Technol. Forecast. Soc. Change 80, 749–761. http://dx.doi.org/10.1016/j.techfore.2012.09.015.

Engel, B., Choi, J.-Y., Harbor, J., Pandey, S., 2003. Web-based DSS for hydrologic impact evaluation of small watershed land use changes. Comput. Electron. Agric. 39, 241–249. http://dx.doi.org/10.1016/S0168-1699(03)00078-4.
Epema, D.H.J., Livny, M., van Dantzig, R., Evers, X., Pruyne, J., 1996. A worldwide flock of condors: load sharing among workstation clusters. Future Gener. Comput. Syst. 12, 53–65.
Feng, M., Liu, S., Euliss Jr., N.H., Young, C., Mushet, D.M., 2011. Prototyping an online wetland ecosystem services model using open model sharing standards. Environ. Model. Softw. 26 (4), 458–468. http://dx.doi.org/10.1016/j.envsoft.2010.10.008.
Gassman, P.W., Reyes, M.R., Green, C.H., Arnold, J.G., 2007. The soil and water assessment tool: historical development, applications, and future research directions. Trans. ASABE 50, 1211–1250.
Gitau, M.W., Chiang, L.C., Sayeed, M., Cahubey, I., 2011. Watershed modeling using large-scale distributed computing in condor and the soil and water assessment tool model. Simulation. http://dx.doi.org/10.1177/0037549711402524.
Gronewold, A.D., Clites, A.H., Smith, J.P., Hunter, T.S., 2013. A dynamic graphical interface for visualizing projected, measured, and reconstructed surface water elevations on the earth's largest lakes. Environ. Model. Softw. 49, 34–39. http://dx.doi.org/10.1016/j.envsoft.2013.07.003.
Habib, E., Ma, Y., Williams, D., Sharif, H.O., Hossain, F., 2012. HydroViz: design and evaluation of a web-based tool for improving hydrology education. Hydrol. Earth Syst. Sci. 16, 3767–3781. http://dx.doi.org/10.5194/hess-16-3767-2012.
Harbaugh, A.W., 2005. MODFLOW-2005, the U.S. Geological Survey Modular Ground-water Model – the Ground-water Flow Process: U.S. Geological Survey Techniques and Methods 6-A16. Available online at: http://pubs.usgs.gov/tm/2005/tm6A16/.
Horak, J., Orlik, A., Stromsky, J., 2008. Web services for distributed and interoperable hydro-information systems. Hydrol. Earth Syst. Sci. 12, 635–644.
Horsburgh, J.S., Tarboton, D.G., Schreuders, K.A.T., Maidment, D.R., Zaslavsky, I., Valentine, D., 2010. HydroServer: a platform for publishing space-time hydrologic datasets. In: AWRA Spring Speciality Conference, Florida, USA, 29–31 March.
Huang, B., 2003. Web-based dynamic and interactive environmental visualization. Comput. Environ. Urban Syst. 27, 623–636.
Janssen, J.A.E.B., Hoekstra, A.Y., de Kok, J.-L., Schielen, R.M.J., 2008. Delineating the model-stakeholder gap: framing perceptions to analyse the information requirement in river management. Water Resour. Manag. 23, 1423–1445. http://dx.doi.org/10.1007/s11269-008-9334-9.
Kanwar, R., Narayan, U., Lakshmi, V., 2010. Web service based hydrologic data distribution system. Comput. Geosci. 36, 819–826. http://dx.doi.org/10.1016/j.cageo.2007.10.017.
Kumar, S., Merwade, V., 2009. Impact of watershed subdivision and soil data resolution on SWAT model calibration and parameter uncertainty. J. Am. Water Resour. Assoc. 45, 1179–1196. http://dx.doi.org/10.1111/j.1752-1688.2009.00353.x.
Leenhardt, D., Therond, O., Cordier, M.-O., Gascuel-Odoux, C., Reynaud, A., Durand, P., Bergez, J.-E., Clavel, L., Masson, V., Moreau, P., 2012. A generic framework for scenario exercises using models applied to water-resource management. Environ. Model. Softw. 37, 125–133. http://dx.doi.org/10.1016/j.envsoft.2012.03.010.
Loucks, D.P., Fedra, K., 1987. Impact of changing computer technology on hydrologic and water resource modeling. Rev. Geophys. 25, 107–112.
Loucks, D.P., Kindler, J., Fedra, K., 1985. Interactive water resources modeling and model use: an overview. Water Resour. Res. 21, 95–102.
Mahmoud, M., Liu, Y., Hartmann, H., Stewart, S., Wagener, T., Semmens, D., Stewart, R., Gupta, H., Dominguez, D., Dominguez, F., Hulse, D., Letcher, R., Rashleigh, B., Smith, C., Street, R., Ticehurst, J., Twery, M., van Delden, H., Waldick, R., White, D., Winter, L., 2009. A formal framework for scenario development in support of environmental decision-making. Environ. Model. Softw. 24, 798–808. http://dx.doi.org/10.1016/j.envsoft.2008.11.010.
Mathiyalagan, V., Grunwald, S., Reddy, K.R., Bloom, S. a, 2005. A WebGIS and geodatabase for Florida's wetlands. Comput. Electron. Agric. 47, 69–75. http://dx.doi.org/10.1016/j.compag.2004.08.003.
McIntosh, B.S., Ascough, J.C., Twery, M., Chew, J., Elmahdi, A., Haase, D., Harou, J.J., Hepting, D., Cuddy, S., Jakeman, A.J., Chen, S., Kassahun, A., Lautenbach, S., Matthews, K., Merritt, W., Quinn, N.W.T., Rodriguez-Roda, I., Sieber, S., Stavenga, M., Sulis, A., Ticehurst, J., Volk, M., Wrobel, M., van Delden, H., El-Sawah, S., Rizzoli, A., Voinov, A., 2011. Environmental decision support systems (EDSS) development – challenges and best practices. Environ. Model. Softw. 26, 1389–1402. http://dx.doi.org/10.1016/j.envsoft.2011.09.009.
Mclennan, M., Kennell, R., 2010. HUBzero: a platform for dissemination and collaboration in computational science and Engineering. Comput. Sci. Eng. 2, 48–53. http://dx.doi.org/10.1109/MCSE.2010.41.
Merwade, V., Ruddell, B.L., 2012. Moving university hydrology education forward with community-based geoinformatics, data and modeling resources. Hydrol. Earth Syst. Sci. 16, 2393–2404. http://dx.doi.org/10.5194/hess-16-2393-2012.
MSHA-MDE, 2010. Application of Hydrologic Methods in Maryland, third ed. Maryland State Highway Administration and Maryland Department of Environment, MD, USA. Available online at: http://www.gishydro.eng.umd.edu/HydroPanel/hydrology_panel_report_3rd_edition_final.pdf.
Nearing, M.A., Wei, H., Stone, J.J., Pierson, F.B., Spaeth, K.E., Weltz, M.A., Flanagan, D.C., Hernandez, M., 2011. A rangeland hydrology and erosion model. Trans. ASABE 54, 1–9.
Neitsch, S.L., Arnold, J., Kiniry, J., Williams, J., 2011. Soil & Water Assessment Tool. Theoretical Documentation Version 2009. Texas A&M University System, College Station, TX, USA.
Olsson, J.A., Andersson, L., 2006. Possibilities and problems with the use of models as a communication tool in water resource management. Water Resour. Manag. 21, 97–110. http://dx.doi.org/10.1007/s11269-006-9043-1.
Peckham, S.D., Goodall, J.L., 2013. Driving plug-and-play models with data from web services: a demonstration of interoperability between CSDMS and CUAHSI-HIS. Comput. Geosci. 53, 154–161. http://dx.doi.org/10.1016/j.cageo.2012.04.019.
Price, C.V., Nakagaki, N., Hitt, K.J., Clawges, R.C., 2006. Enhanced Historical Land-use and Land-cover Data Sets of the U.S. Geological Survey. U.S. Geological Survey Digital Data Series 240. [digital data set]. http://pubs.usgs.gov/ds/2006/240.
Ramsey, K., 2009. GIS, modeling, and politics: on the tensions of collaborative decision support. J. Environ. Manag. 90, 1972–1980. http://dx.doi.org/10.1016/j.jenvman.2007.08.029.
Rao, M., Fan, G., Thomas, J., Cherian, G., Chudiwale, V., Awawdeh, M., 2007. A web-based GIS decision support system for managing and planning USDA's Conservation Reserve Program (CRP). Environ. Model. Softw. 22, 1270–1280. http://dx.doi.org/10.1016/j.envsoft.2006.08.003.
Rekolainen, S., Kämäri, J., Hiltunen, M., Saloranta, T.M., 2003. A conceptual framework for identifying the need and role of models in the implementation of the water framework directive. Int. J. River Basin Manag. 1, 347–352. http://dx.doi.org/10.1080/15715124.2003.9635217.
Seibert, J., Uhlenbrook, S., Wagener, T., 2013. Hydrology education in a changing world. Hydrol. Earth Syst. Sci. 17, 1393–1399. http://dx.doi.org/10.5194/hess-17-1393-2013.
Sun, A., 2013. Enabling collaborative decision-making in watershed management using cloud-computing services. Environ. Model. Softw. 41, 93–97. http://dx.doi.org/10.1016/j.envsoft.2012.11.008.
Tarboton, D.G., Idaszak, R., Horsburgh, J.S., Heard, J., Ames, D., Goodall, J.L., Band, L., Merwade, V., Couch, A., Arrigo, J., Hooper, R., Valentine, D., Maidment, D., 2014. HydroShare: advancing collaboration through hydrologic data and model sharing. In: 7th International Congress on Environmental Modelling and Software, San Diego, California, USA, 15–19 June.
Tarboton, D.G., Horsburgh, J.S., Maidment, D.R., Whiteaker, T., Zaslavsky, I., Piasecki, M., 2009. Development of a community hydrologic information system. In: 18th World IMACS/MODSIM Congress, Cairns, Australia, 13–17 July, pp. 988–994.
USACE, 2013. Hydrologic Modeling System HEC-HMS User's Manual. Army Corps of Engineers, Hydrologic Engineering Center-HEC, Davis, CA, USA.
USGS-NED, 2013. National Elevation Dataset: United States Geological Survey National Map Viewer. Available at: http://viewer.nationalmap.gov/viewer/ (accessed 10.03.13).
USGS-NLCD, 2013. National Land Cover Data Set, 2006: United States Geological Survey National Map Viewer. Available at: http://viewer.nationalmap.gov/viewer/ (accessed 10.03.13).
Voinov, A., Bousquet, F., 2010. Modelling with stakeholders. Environ. Model. Softw. 25, 1268–1281. http://dx.doi.org/10.1016/j.envsoft.2010.03.007.
Voinov, A., Costanza, R., 1999. Watershed management and the web. J. Environ. Manag. 56, 231–245.
Wasson, J., Tusseau-Vuillemin, M., Andréassian, V., Perrin, C., Faure, J., Barreteau, O., Bousquet, M., Chastan, B., 2003. What kind of water models are needed for the implementation of the European Water Framework Directive? Examples from France. Int. J. River Basin Manag. 1, 125–135. http://dx.doi.org/10.1080/15715124.2003.9635199.