Visualization workflows for level-12 HUC scales: Towards an expert system for watershed analysis in a distributed computing environment

Lorne Leonard, Chris Duffy

Abstract

Visualization workflows are important services for expert users to analyze watersheds when using our HydroTerre end-to-end workflows. Analysis is an interactive and iterative process and we demonstrate that the expert user can focus on model results, not data preparation, by using a web application to rapidly create, tune, and calibrate hydrological models anywhere in the continental USA (CONUS). The HydroTerre system captures user interaction for provenance and reproducibility to share modeling strategies with modelers. Our end-to-end workflow consists of four workflows. The first is data workflows using Essential Terrestrial Variables (ETV) datasets that we demonstrated to construct watershed models anywhere in the CONUS (Leonard and Duffy 2013). The second is data-model workflows that transform the data workflow results to model inputs. The model inputs are consumed in the third workflow, model workflows (Leonard and Duffy 2014b) that handle distribution of data and model within High Performance Computing (HPC) environments. This article focuses on our fourth workflow, visualization workflows, which consume the first three workflows to form an end-to-end system to create and share hydrological model results efficiently for analysis and peer review. We show how visualization workflows are incorporated into the HydroTerre infrastructure design and demonstrate the efficiency and robustness for an expert modeler to ***produce, analyze, and share*** new hydrological models using CONUS national datasets.

Keywords

Visualization as a service
Model as a service
Data as a service
Visualization workflows
Provenance
Reproducibility
End-to-End System

Software availability

Name

HydroTerre

Software & System Architect, Engineer, and Research Developer

Lorne Leonard, Department of Civil Engineering & Penn State Institutes of Energy and the Environment, The Pennsylvania State University.

Contact information

Lorne Leonard & Christopher J. Duffy Department of Civil & Environmental Engineering, The Pennsylvania State University, 212 Sackett Building, University Park, PA 16802, USA

Software required

Internet browser (later versions are recommended) with JavaScript and Silverlight support.

Program languages and frameworks used in HydroTerre

Client Side: HTML, JavaScript, XAML and XML

Server Side: C++, C#, Microsoft SQL, XAML and XML

Frameworks: .NET, ArcObjects, Silverlight, LINQ, COM

Availability and cost

Any user can access HydroTerre level-12 HUC ETV data-workflow web applications at no cost at: http://www.hydroterre.psu.edu

Please contact Leonard for permission to access Model and visualization workflows.

1. Introduction

We demonstrate that workflows empower modelers to rapidly produce watershed models anywhere in the continental United States (CONUS). By automatically recording user interactions with our system, we derive provenance datasets that serve as a resource for United States Geological Survey (USGS) level-12 Hydrological Unit Codes (HUC-12) (USGS 2013)(Seaber, Kapinos, and Knapp, 1987) catchment scale models. The significance of storing all user settings and cyberinfrastructure properties, within provenance datasets, is that it allows users to reproduce other's models using our system and not needing to keep petabytes of results and intermediate steps.

Four types of workflows are necessary to achieve this scale of hydrological modeling. The first is data workflows as demonstrated using national Essential Terrestrial Variable (ETV) datasets (Leonard and Duffy, 2013). The second is data-model workflows that transform ETV data for model use. The third is model workflows consuming the data-model data bundles to produce hydrological models. We have executed these workflows more than a million times, storing user provenance in our datasets for reproducibility (Leonard and Duffy 2014b). Visualization is the fourth workflow and the focus of this article. We demonstrate the feasibility of visualization workflows for level-12 HUCs to rapidly prototype and develop hydrological models anywhere in the CONUS.

The combination of these workflows constitutes an expert system that provides software as a service for hydrological modeling. The Penn State Integrated Hydrological Model (PIHM) (Qu and Duffy, 2007) is demonstrated here, but the workflows serve as a template for other models to adapt and become new services. Visualization workflow is the tail end of our end-to-end workflows (data, data-model, model, and visualization workflows) for hydrological analysis using national datasets. However, we need to clarify to the reader that the intention of this article is on the transformation process of data and model, not particular calibrated model results.

These workflows, in conjunction with the web application prototype, are necessary to capture all steps taken by the expert user to enable reproducibility and provenance for peer review and sharing of data and models. An explanation of both hardware and software architecture is required to explain how the software components operate. **Section 1** highlights the need to couple visualization with data, data-model, and model workflows for hydrological modeling. **Section 2** provides an overview map about the HydroTerre architecture and a summary of our data, data-model, and model workflows. **Section 3** explains technical details about the visualization workflows. Our prototype web application to create and evaluate end-to-end workflows is discussed in **Section 4**. **Section 5** demonstrates the feasibility of using the prototype to create visualizations using distributed computing environments with CONUS level-12 HUC catchments for rapid modeling, comparison, and debugging model results.

1.1 What is automated within the end-to-end workflows?

A user initiates the end-to-end workflows in HydroTerre by selecting a level-12 HUC. There are four phases and the reader is referred to (Leonard and Duffy 2014b) for specific details to the first three phases. Briefly, the first phase is the ETV data workflow that is responsible for selecting, projecting, clipping, and extracting data within the level-12 HUC catchment efficiently. The second phase (data-model workflow) is the transformation of ETV data generated into PIHM input file formats. The third phase (model workflow) is the web-based user interface that captures all steps by storing data-model, model, and visualization parameters as database objects for fast retrieval, provenance, and reproducibility (Silva et al. 2007) (Silva et al. 2011) (Groth and Streefkerk 2006) (Bowers 2012). This happens automatically when a user submits a task and is a critical step to sharing parameters and modelling steps with other users and stakeholders. The fourth phase, and the focus of this article, is the transformation from PIHM outputs to web-based visualizations for analysis. Visualizations are automatically created (specified by user) once the model results are available within a distributed compute environment.

1.2 Why visualization workflows?

Data workflows provide ETV spatial data (soils, land cover, etc.) at the level-12 HUC scale and time-series North American Land Data Assimilation System (NLDAS 2011) climate forcing for a period of 30 years (one climate normal) (Arguez and Vose, 2011). This is done via a web-based visualization application and is as simple as a user selecting a level-12 HUC and specifying a forcing period. Within a few minutes, an email is sent to the user with a link to where a data bundle is available to the user for download (Leonard and Duffy, 2013). This strategy, at a level-12 HUC scale, is efficient for a user to download to their personal compute environment and manipulate for their own needs. However, beyond level-12 HUC scale, data bundle sizes increase from 100s of gigabytes to terabytes and efficient visualization services to assist hydrologists for analysis are needed. Visualization tools are important to convey to the expert user, strategies to partition data to select upstream catchments appropriate for their modeling needs (Leonard and Duffy 2014a) (Leonard et al. 2015a). Finally, the hydrological modeler wants to check model results, find data issues, make alterations, and calibrate their models efficiently.

End-to-end workflows will help users to standardize the hydrological modeling processes by minimizing errors due either to the original source or with how modelers process and make decisions about data and models. Using workflows reduces the time that modelers invest in manually changing individual parameters and input data to optimize the data inputs to generate quality hydrological models. The end-to-end workflow assures that data and model provenance is not lost (Davidson and Freire 2008) (Deelman et al. 2009). Overall, the fundamental reason for these end-to-end workflows is to capture all steps for reproducibility and provenance.

To capture these steps requires capturing user interaction within the web application. The workflows described here do not restrict users from downloading data and visualization results and using the data offline. However, the emphasis of these workflows is reproducibility and rapid prototyping so users can retrieve a personal copy at the end of the process and to share their strategies for novice or expert users.

Furthermore, storing parameters to replicate the entire end-to-end workflow process is disk space efficient, a hundred gigabytes of disk is needed to store a million workflow instances. This may appear trivial for a few case studies, however, 100’s of terabytes of data storage is necessary for the ETV web-service data-workflows. Assuming 1 to 10 gigabytes of storage is required for 30 years of input data per level-12 HUC data-model and model workflows, and the same amount of storage for visualization workflows, 100’s of terabytes of disk storage would be required to keep end-to-end workflow steps for the entire CONUS.

1.3 Constraints

This article focuses on the visualization transformation process and the use of a distributed computing environment to evaluate the feasibility to share hydrological models via web services. It should be noted that there are multiple workflow versions designed and developed to be optimized for PIHM in specific HPC environments. This is due to the emphasis on performance that is constrained by various computing environments and management practices. For example, security (user and data) is different at each HPC environment. Due to management computing practices at The Pennsylvania State University, the prototype web services are restricted to the public to protect data and security of resources. Level-12 HUC ETV data workflows are publicly available with no restrictions.

In the prototype phase, the workflows presented here are restricted to one level-12 HUC selected by the web user and executing workflows are restricted to expert users. In the next phase of this prototype, we will deal with issues associated with scaling up to a network of level-12 HUC watersheds, as discussed in (Leonard and Duffy, 2014) (Leonard, et.al 2015a), where flow direction between level-12 HUC requires validation to verify the hierarchy and upstream level-12 HUC is calibrated.

1.4 Related work

In recent years, a number of hydrological web based visualization service-oriented applications have been developed that use site-specific study sites for sharing data. The Iowa flood Information System provides access to flood inundation maps with real-time flood conditions and flood forecasts for gages and sensors throughout the state of Iowa, USA, using web services and distributed services for data, map, analysis, and visualization (Demir and Krajewski 2013) (Gilles et al. 2012). Another service includes HydroShare, with similar goals to HydroTerre, to share hydrological data, models, and visualization results (Horsburgh et al. 2015), (CUASHI 2015), (Valentine et al. 2014), (Tarboton et al. 2014), (Ames et al. 2014). The AWARE project geo-portal application supports map visualization using Google Map API to show results from (Granell, Díaz, and Gould, 2010) two hydrological models, the Snowmelt Runoff Model (Martinec, Rango, and Roberts, 1994) for daily stream flow forecasts in mountain basins, and the TUW-HBV model (Parajka, Merz, and Blöschl, 2005) a semi-lumped rainfall runoff model. Goodall, (J. Goodall, Horsburgh, Whiteaker, Maidment, and Zaslavsky, 2008), (J. L. Goodall, Robinson, and Castronova, 2011) consider service oriented computing as a strategy for integrating independent water resource models and (Horsburgh, Maidment, Whiteaker, Zaslavsky, and Piasecki, 2009) have applied the concept to publishing environmental data for desktop visualization using HydroDesktop (Ames et al. 2012) (Castronova et al. 2013). With regard to integrating data, model, and visualization workflows for meteorological data, we point the reader to (Turuncoglu, Dalfes, Murphy, and DeLuca, 2013) who discusses coupling an Earth System Modeling Framework (ESMF) with the Regional Ocean Modeling System (ROMS) and Weather Research and Forecasting Model (WRF). Clearly, hydrological service oriented applications will be essential to the next generation of model applications.

2. System design

This section describes the computer hardware and software that forms the foundation for automation of HydroTerre’s end-to-end workflows. The hardware has been structured to efficiently serve the large volumes of data required to support these workflows anywhere in the CONUS. The data and data-model workflows are distributed within the data-tier of the HydroTerre system, and the hydrological modeling is distributed to other HPC systems to compute PIHM models. Visualization workflows are distributed in the visualization tier. Clearly, an efficient and robust service-oriented architecture (Section 2.2) is critical to support the rapid prototyping and delivery of visualization workflows. Furthermore, Section 2.3 summarizes the workflows that we have evaluated more than a million times that are required for our visualization workflow service.

2.1 Hardware and administration layers

The end-to-end workflows are implemented in a four-tier hardware layer system (Fig. 1). The web interface tier hosts the web applications and services. ESRI's ArcGIS server software (ESRI 2014) development kits (SDK) support GIS web applications and Microsoft SQL server (Microsoft 2014a) is used to store, create, and query spatial datasets. Microsoft SQL server is used to store and query datasets in the second and fourth tier. Components of the data workflows that retrieve the forcing data are implemented on this tier and form the first layer of the distributed computing system as data queries are executed on multiple compute nodes. All the components are executed in parallel for maximum performance that ranges from minutes to many hours depending on the catchment size. We have demonstrated that the way these services, hardware and software, are coupled is critical for performance. The reader is referred to (Leonard et al. 2015a) for further details about performance tuning on forcing datasets and to (Leonard and Duffy, 2013) for further details about compute times and data sizes of ETV datasets. The reader is referred to (Leonard and Duffy 2014b) for details about the web, data, and model tiers where we demonstrated the robustness of our system by executing these workflows millions of times. The network has been upgraded by the Penn State Network Research group from 1 Gb/s to 10 Gb/s (Miller 2015).

Here, we focus on the automated workflows between the Penn State University (PSU) HPC clusters (Tier 3), visualization support (Tier 4), and back to the web interface (Tier 1). Using a specified PIHM account, a custom PIHM dispatcher application runs continuously and uses web services to retrieve PIHM jobs. For example, the data-model workflows have been tested using Penn State Universities CyberSTAR cluster (CyberSTAR 2014). In this mode, the user does not need to login to the compute environment and the PIHM models are automatically dispatched to the compute nodes and then to the visualization server. Job management is achieved via the data tier, with all compute and visualization nodes accessing job tasks from the data tier, via user-project databases.

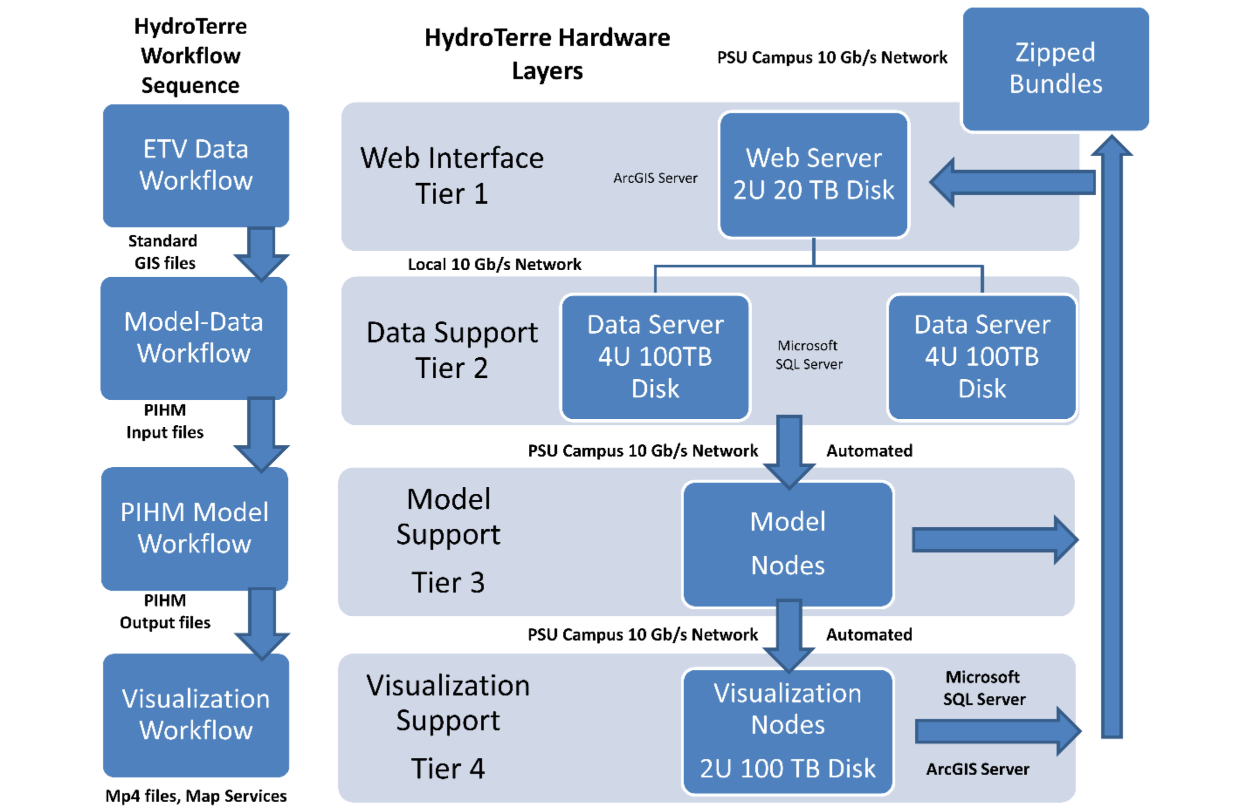


Fig. 1. Four-tier hardware layer system to support data-model, model, and visualization workflows. Tier-one supports web applications, tier-two supports the data services, tier-three supports the model development and tier four supports the visualization services. Both tiers one and two support the data-model workflows.

Statistics about data, model, and visualization performance are returned to the web tier database. Access to the model results is automatically sent back to the web interface tier that is then retrieved by the visualization nodes for processing. The exact end-to-end workflow can be replicated using the identical cyberinfrastructure. Reproducibility is extremely useful for debugging purposes, critical when 100,000s of jobs are running within different types of HPC environments. The web tier is the Graphical User Interface (GUI) to all aspects of the workflows, with the data, model, and visualization tiers operating without any user intervention. How the four tiers work together, when a web user starts the HydroTerre web application, is discussed in Section 2.2, starting with an overview map of the service-oriented software architecture.

2.2 Overview map of service-oriented architecture

In previous sections, the workflows have been presented abstractly as individual objects to represent their main functionality. In fact, the workflows are hundreds of discrete pieces of software that provide application functionality to other applications that constitute HydroTerre end-to-end workflows. The workflows are accessible as private service-oriented architecture (SOA) (Microsoft 2014b) (Bell 2008) (Bell 2010) services using common communication techniques of Simple Object Access Protocol (SOAP) (World Wide Web Consortium 2014a), Representational State Transfer (REST) (Fielding and Taylor, 2002), Web Services Description Language (WSDL) (World Wide Web Consortium 2014b), and Database Markup Language (DBML) (Microsoft 2014c). Section 3 discusses specific details about the visualization workflows and their components. However, it is important to first provide the reader with an overview map of the SOA and explain the significant paths behind the web application that occur when a user selects a level-12 HUC to execute end-to-end workflows.

When a user visits the prototype application[1] via a web browser, they are accessing internet services hosted on the web interface tier. The HydroTerre website user interface has been developed with Silverlight (Microsoft 2014d) and ArcGIS server SDK. The user interface is responsible for selecting, querying, creating, and retrieving Microsoft SQL Server datasets for display within the web application (Fig. 2A). All data displayed and used in controls reside in databases on the data-tier; the user interface is data driven. The main communication methods between the user interface and the data tier and between the data tier and workflow service layer (Fig. 2B) are SOAP, REST,

[1] http://www.hydroterre.psu.edu/Development/HydroTerre_Leonard_Models/HydroTerre_Models.aspx

WSDL, and DBML. The choice of communication technique depends on where the data resides, what tier layer, and system administration.

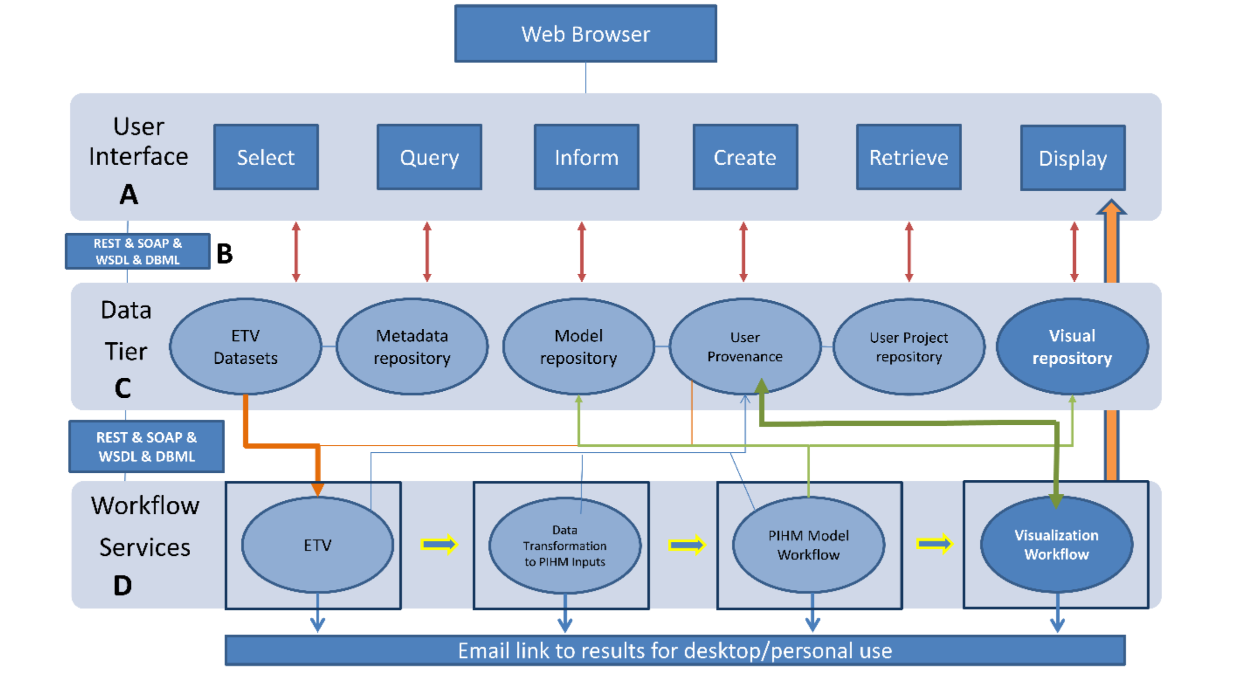


Fig. 2. Service-oriented architecture for data-model workflows consists of three layers. The first layer is the web based user interface, supported by a data tier layer, and a workflow service layer. Visualization workflows combine the results from ETV, data-model and model workflows for interaction by the user.

The data tier (Fig. 2C) has three categories. The first consists of ETV datasets and the reader is referred to (Leonard and Duffy, 2013) for further details about their function and computation complexity. The second category is databases that store data-model, model, and visualization parameters. The reader is referred to (Leonard and Duffy 2014b) for further details. The third category builds upon the second category to create, select, and inform user parameters to execute visualization workflows.

When a user selects a level-12 HUC and submits a job to execute the workflows, a new table row with fields shown in Table 1 is created with a Globally Unique Identifier (GUID) (Microsoft 2014e) primary key. Each row contains the HUC identification key and the users email address, and each workflow is stored as a separate Extensible Markup Language (XML) (World Wide Web Consortium 2014c) document. Thus, via the web user interface, queries from millions of workflow results can be searched using level-12 HUC identification, user names, or email addresses to populate the data controls and replicate the exact workflow parameters.

Table 1

The HydroTerre National Job object stored when users execute workflows. The email, name, HUC, and date objects enable SQL queries for filtering and identifying jobs. The HPC properties object stores information related to the compute nodes where the workflows are executed. HUC properties store information about the HUC catchment. The Visualization, Model, and Data Properties object store parameters returned by the workflows. UI Properties contain all the parameters used to execute the data and model workflows. Job and Workflow properties store parameters used and returned by the workflows on the compute nodes.

Column Name	Type	Description
JobID_Nat	nvarchar	Project GUID key
JobID_Data	nvarchar	Data workflow GUID key
SubmitJob	datetime	Time user submitted project job
DeleteJob	datetime	When project was deleted
Last_Accessed	datetime	When project was last accessed
Email_Address	nvarchar	User email address
Project_Name	nvarchar	Automated project name
Pretty_Name	nvarchar	User specified project name
HUC_Name	nvarchar	USGS HUC Name (not unique)
HUC_ID	nvarchar	USGS HUC identification
HPC_Properties	nvarchar (xml document)	HPC XML object
HUC_Properties	nvarchar (xml document)	HUC XML object
Model_Properties	nvarchar (xml document)	Model XML object
Data_Properties	nvarchar (xml document)	Data XML object
UI_Properties	nvarchar (xml document)	User interface XML object
Job_Properties	nvarchar (xml document)	Job Project XML object
Viz_Properties	nvarchar (xml document)	Visualization Workflow XML object (Appendix A1)
Workflow_Properties	nvarchar (xml document)	Workflow XML object
Status_DWF	int	Data workflow status
Status_MWF	int	Model workflow status
Status_VWF	int	Visualization workflow status

		(Appendix B1)

Recall that only the parameters for workflows are stored, not the results. Therefore, a user cannot simply download the ETV, data-model, or visualizations results from a previous job, due to the large amount of disk storage required to store results permanently. The task will need to be executed again, but at the level-12 HUC scale the time to re-create the end-to-end workflow is minimal and requires slight effort from the user. Sections 4 and 5 will demonstrate the simplicity of creating and reproducing end-to-end workflows.

Whether a user is creating a new hydrological study, or is cloning an existing study, none of the datasets generated by the workflows are stored. What is kept is the user inputs that control the workflows and the sequence of end-to-end workflows is the same, as shown in Fig. 2D, from data (ETV), data-model, model, and visualization. During this sequence of workflows, possible locations of errors (missing data), or failures (power or disk) happen in each of the workflows and supporting cyberinfrastructure. To empower the web user to resolve an error returned during workflow execution, either due to administration or from parameter issues, a meaningful error object is returned to the user via the web interface.

2.3 Data, data-model, and model workflows

All HydroTerre workflows are initiated via the web application. The data workflow requires the user to select a level-12 HUC and specify an email address and forcing period. The main processes query and extract datasets within the HUC catchment boundary that are compressed into a zip file and the web user is emailed a link to the file. The zip file contains standard GIS datasets including shape files, GeoTIFFs, and text files. The forcing file is produced using the HydroTerre distributed compute environment. To improve the forcing file generation process, further analysis of data encoding (XML versus Google ProtoBuffer(Google 2015)) and system tuning has been conducted that has improved forcing generation from eleven minutes to one minute for a typical level-12 HUC (Leonard et al. 2015a).

The data workflow is an independent service that provides data, downloaded via web links, for any model. Conversely, the data-model workflow is dependent and consumes the data workflow service as data inputs. With the same approach as the data workflow, the data-model workflow requires minimal inputs (four in total with default settings for novice users). These inputs control the catchment level-12 HUC boundary and stream topology, that in turn, controls the level of detail of the unstructured mesh representing the terrain topology. The data-model workflow also assigns values from ETV datasets to the mesh and generates XML files.

The PIHM-model workflow consumes the data-model workflows as data inputs. Unlike the data and data-model services, there is no goal of limiting the user inputs to control PIHM. Default values are assigned to all parameters within the Calibration, Parameter, Initialization, and HPC model categories to minimize a beginner’s requirement to start a PIHM model. However, these values are unlikely to be accurate, but the process is useful to focus whether the model parameters are causing problems or the data-model workflow setup is the cause of model failures. The web application clearly indicates for quick investigation the cause of known errors such as poor quality meshes, missing data, or numerical instability issues. Although, in (Leonard and Duffy 2014b) millions of workflows were evaluated to determine HPC requirements and investigate the main reasons why data-workflows failed (stream networks and poor meshes) at the level-12 HUC scale. The HydroTerre web application lacked the ability of encouraging the iterative and investigative process of the expert, the hydrologist, to rapidly change model parameters and visualize the results via the web services. The following sections demonstrate the importance and feasibility of visualization workflows to improve hydrological modeling processes using CONUS national datasets.

3. Visualization workflow services

Section 2 provided an overview of how the cyberinfrastructure contributed to HydroTerre's end-to-end workflows and summarized the workflows required to support the visualization workflow. Section 3 describes details about how the workflow services operate together to visualize hydrological model results. Section 3.1 summarizes the main visualization component services that consume the data and model services within a distributed computing environment to provide support for watershed analysis. Section 3.2 explains the technical details of the visualization workflow and how the prototype demonstrated in Sections 4 and 5 execute and retrieve visualization workflow results.

3.1 Visualization workflow overview

The visualization workflow consumes the data, data-model, and model workflow services as data inputs. The purpose of visualization services is to provide maps and data visualizations for the expert user to drill-down, interrogate model results, and quickly test and re-submit models for further evaluation. Assuming the data, data-model and model workflows are successful, the visualization workflow retrieves all three data-bundles to the visualization service node and unzips the contents (Fig. 3A). Unzipping is dependent on user preferences (Fig. 3B) and settings defined in the project database (Fig. 3C). Spatial and time-series data is then transformed (Fig. 3D) for map services (Fig. 3E), data-visualizations, and mp4 movie generation (Fig. 3F). When the HydroTerre system has completed the workflows, an email is sent to the user. Then, the user can return to the web application to investigate the model results using the visualization services (Fig. 3G). When the modeler is not satisfied with the model results, he or she can clone the workflow that is closest to their goals, make changes to the model or data parameters, and resubmit the end-to-end workflow. With enough cyberinfrastructure resources, this process could aid the expert user to rapidly produce a calibrated hydrological model.

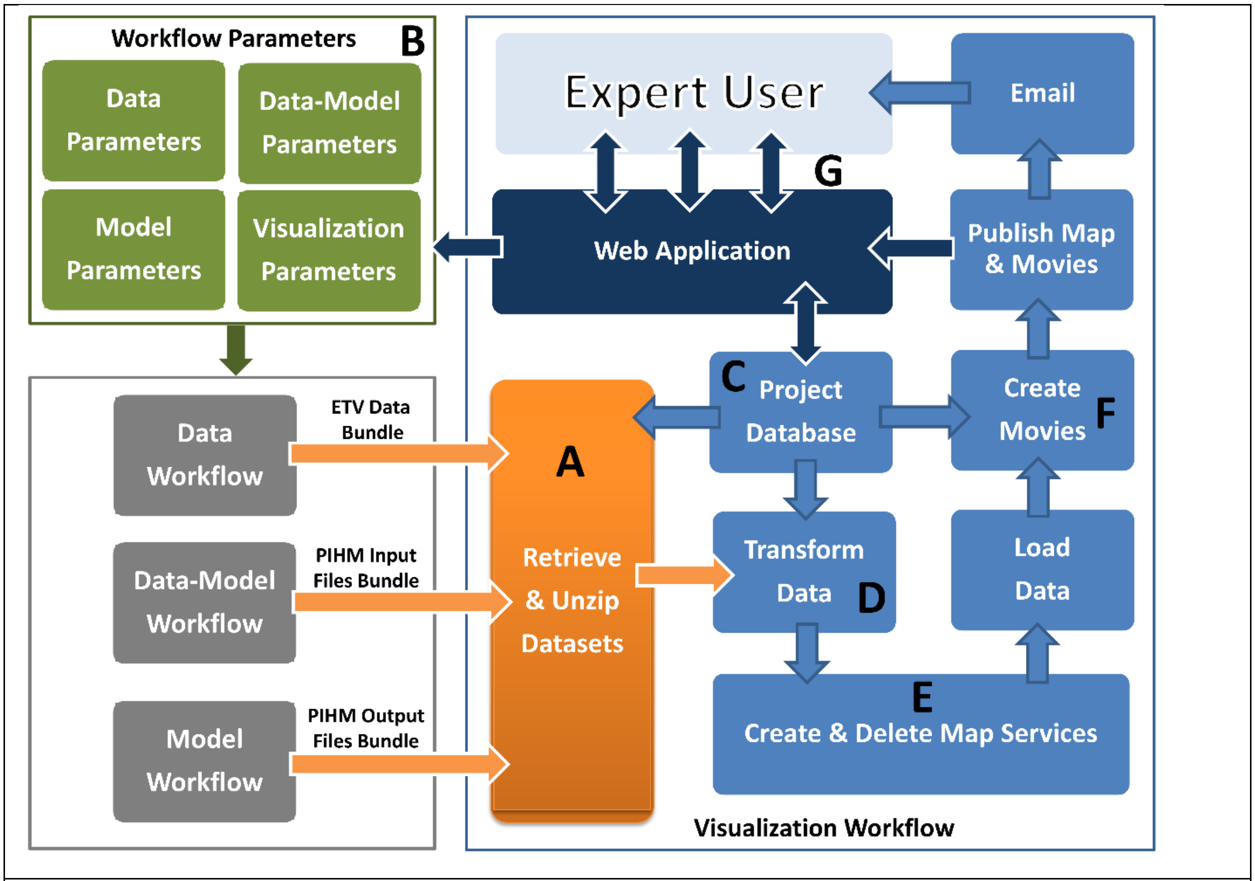


Fig. 3. The ETV data workflow handles queries and clipping of national datasets for the selected level-12 HUC. These results are then processed by the data-model workflows that transform the data into PIHM datasets. The model workflow consumes this data-model bundle and executes PIHM in a distributed compute environment. The visualization workflow consumes all three data bundles, and with user settings stored in the project database, the data bundles are transformed for visualization services. The visualization services include interactive maps that load ETV and data-model spatial datasets, as well as rendered movies. Then, these visualizations are made available to the web application for the expert user to interact with.

3.2 Executing and retrieving visualization workflow results

The visualization service is a back-end service within a dedicated visualization tier (Fig. 1) that continuously queries the data, model, and visualization workflow statuses (Table 1) using SQL from a submission database table. Once the workflow statuses are successful (Appendix B), the visualization service queries by de-serializing the data and model properties (Table 1 and Appendix C) to retrieve the locations of the data, data-model, and model bundle results from each workflow. It is necessary to query each workflow bundle location, as each workflow occurs in a distributed compute environment and does not happen on the same visualization compute tier.

A workspace folder is created based on the project job identification key (JobID_Nat in Table 1) and with wget software (Free Software Foundation 2015), the data bundles are retrieved and unzipped in the workspace folder. The workflow verifies each data bundle. For example, by checking that the data bundle files are intact, as such an error happens with the transfer of data between networks. The disk size of the three data bundles is dependent on the level-12 HUC catchment size, the number of catchment mesh elements, and the number of years simulated. The visualization workflow validates that there is enough disk space on the visualization node and predicts the amount of disk required to complete the remaining components of the visualization workflow. The prediction is based on the visualization properties specified (Table 1 and Appendix A) by the user and the number of concurrent threads. For example, more disk is required to create and process daily averaging of each model variable versus yearly averaging. If there is not enough disk space, the web user is informed with the appropriate error object (Appendix B). How this information is shared with the web user is dependent on the web application calling the visualization service.

The next component of the visualization workflow is to transform the data, data-model, and model results for web services. Transformation includes checking that all spatial projection systems are identical and pivoting tables for efficient use in SOAP, REST, WSDL, and DBML consumption (Section 2). In our modeling example, PIHM creates separate text output files for multiple hydrological fluxes, with each row in the output file representing one time interval. Within each row, each column is a mesh element value at that time step. This data structure is not appropriate for table joins with mesh shapefiles using ArcGIS server. Instead, all variables need to be grouped by mesh identification to be joined with the mesh. This can be computationally expensive when both mesh sizes are large (10,000s of elements) and the time averaging period is short (hourly, daily) for large simulation periods. If there are errors in the data, for example, Not a Number (NANs), or missing rows and columns, the web user is informed via an appropriate error object.

After the data has been transformed using C++, C#, and Python custom software tools, ArcGIS map services are created based on the unique JobID_Nat key. The transformed data is loaded into the map service. These map services are automatically deleted when passed the Delete Job (Table 1) date. Transformed data is also necessary for movie generation. Each averaged model variable is loaded and joined with the mesh. Then, each mesh variable is rendered as an image per averaged time interval. The rendered images are joined using FFmpeg (FFmpeg team 2015) to create mp4 movies for web access. Once the movies have been generated, the map services are published for the web application to access. Then, the user is emailed (if requested) his or her visualization results. Again, as many software tools have been grouped into one visualization workflow, there are many possible locations for failures and the web user is notified via an appropriate error object summarized in Appendix B.

4. Prototype to create visualization workflows

At the website www.hydroterre.psu.edu, under the services tab, a stand-alone demonstration to execute the ETV data workflow, independent of any model, is available to the reader. Here, we present a prototype[2] Silverlight web application that does not treat the data workflow as a standalone service and is coupled with both data-model, model, and visualization workflow services. This prototype consumes private web services (due to administration restrictions) based on Sections 2 and 3 that are summarized in Appendix C. Section 4.1 reintroduces the procedure to setup data-model and model workflows. Section 4.2 introduces the user interface to define a visualization workflow for hydrological analysis.

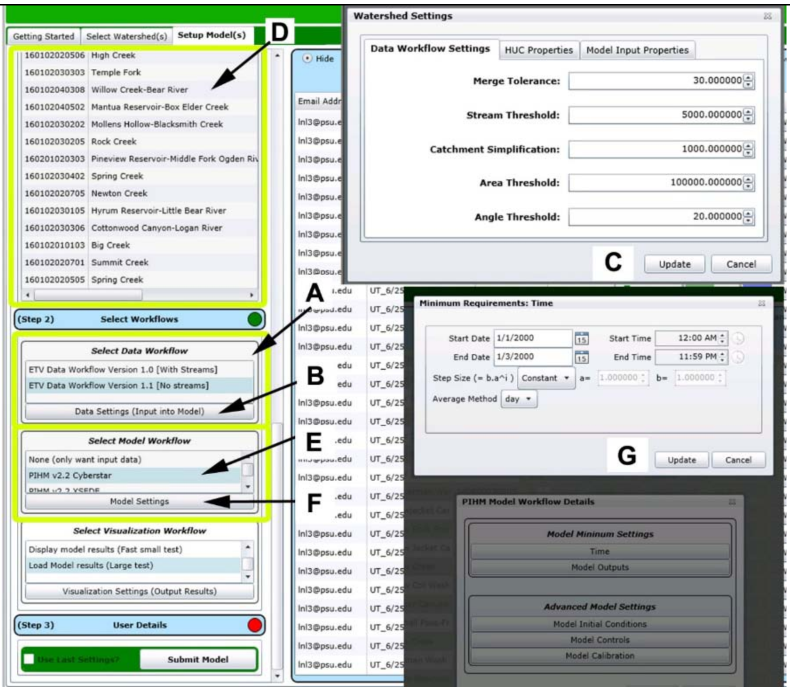
4.1 Setup data-model and model workflows

Once the user has defined a list of level-12 HUCs for hydrological modelling, the next step is to select a data-model workflow as highlighted in Fig. 4A. The user can define

[2] www.hydroterre.psu.edu/Development/HydroTerre_Leonard_Models/HydroTerre_Models.html

data-model workflow parameters by clicking on the button highlighted in Fig. 4B to reveal the user interface control (Fig. 4C). Any changes update the parameters that are applied to the selection list highlighted in Fig. 4D. These parameters are serialized into an XML data workflow object string (Appendix C) and stored in the Data_Properties cell (Table 1) for each individual HydroTerre national job selected by the user.

After defining the data-model workflow properties, the user can select which PIHM workflow version and which HPC resource they wish to use (Fig. 4E). As summarized in Section 2.3, the user can define and control PIHM by clicking on the interface button highlighted in Fig. 4F to reveal the user interface control (Fig. 4G). All the selected level-12 HUCs will use the same user defined parameters (Fig. 4D). The parameters in these controls are de-serialized (Model_Properties and HPC_Properties) objects from the stored HydroTerre national job object called using REST or SOAP web services from the Silverlight web application. Any modifications made by the user via the interface are serialized into strings and the national job object (Table 1) is updated with SQL query.



160102020506	High Creek
160102030303	Temple Fork
160102040308	Willow Creek-Bear River
160102040502	Mantua Reservoir-Box Elder Creek
160102030202	Mollens Hollow-Blacksmith Creek
160102030205	Rock Creek
160201020303	Pineview Reservoir-Middle Fork Ogden Riv
160102030402	Spring Creek
160102020705	Newton Creek
160102030105	Hyrum Reservoir-Little Bear River
160102030306	Cottonwood Canyon-Logan River
160102010103	Big Creek
160102020701	Summit Creek
160102020505	Spring Creek

ETV Data Workflow Version 1.0 [With Streams]
ETV Data Workflow Version 1.1 [No streams]

None (only want input data)
PIHM v2.2 Cyberstar
PIHM v2.2 XSEDE

Display model results (Fast small test)
Load Model results (Large test)

Merge Tolerance: 30.000000
Stream Threshold: 5000.000000
Catchment Simplification: 1000.000000
Area Threshold: 100000.000000
Angle Threshold: 20.000000

Start Date 1/1/2000
End Date 1/3/2000
Start Time 12:00 AM
End Time 11:59 PM
Step Size (= b.a^i) Constant a= 1.000000 b= 1.000000
Average Method day

Fig. 4. The user selects which data-model workflow (a) to apply. To change the data-model workflow, the user clicks on the data settings button (b) and changes variables in the interface (c). Workflow settings are then applied to the level-12 HUC selection

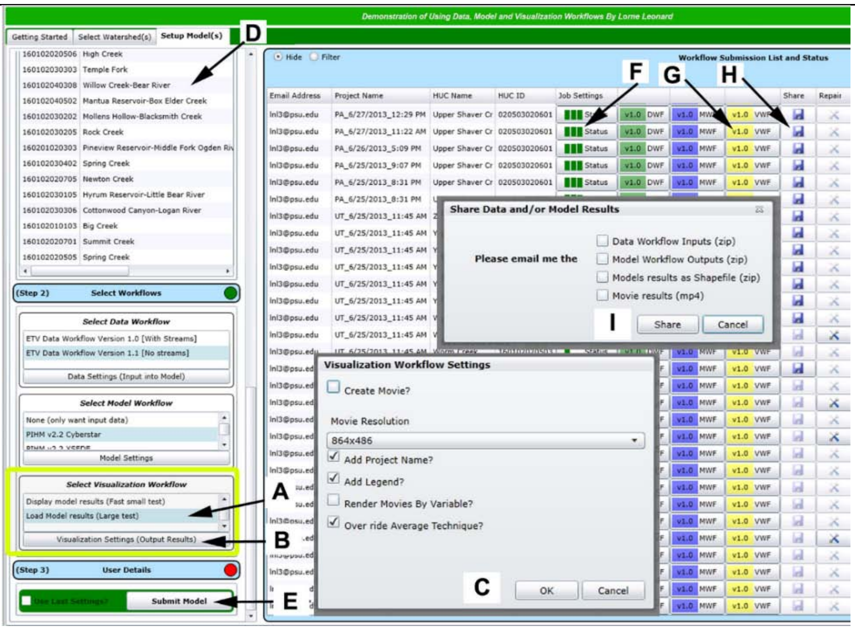
(d). The user selects which model workflow (e) they wish to apply. To change the model workflow, the user clicks on the model settings button (f) and can change variables in the interface (g). Workflow settings are then applied to the level-12 HUC selection (d).

4.2 Setup visualization workflows

The user follows the same procedure to define a visualization workflow by selecting which version and HPC resource to use (Fig. 5A). A database table stores unique properties for each available HPC resource (Appendix C) that is queried using SQL to populate the user interface. By default, movie generation is off with extensive simulations due to the potential long time the process takes. Instead, to encourage faster movie generation, the averaging method is set to long periods as shown in Fig. 4G. The user can override this behavior and change movie parameters by clicking on the settings button (Fig. 5B) and modifying properties in the modal window (Fig. 5C). These changes are applied only to the selected list in Fig. 5D. The parameters are serialized into an XML visualization workflow object string (Appendix C) that are stored in the Viz_Properties object (Table 1) for each individual HydroTerre national job in the selection list using SQL update query.

The last step is for the user to submit the workflows to the workflow submission list (first in first out queue) indicated in Fig. 5E. When the ETV, data-model, and model workflows have finished, the visualization starts as discussed in Section 3. Once the visualization has completed, there will be three green bars (Fig. 5F). A white bar indicates the workflow is either waiting or not submitted. Orange indicates “in progress” and red indicates “failure”. The user, either clicking on the refresh button or using a timer (every five minutes) to update the submission list, updates the status bars. The Silverlight web application queries each HydroTerre national job object and color codes the status bars based on the return values stored in the workflow status objects (Table 1).

When the user moves the mouse cursor over these status bars, a detailed message is displayed to inform the user of further details retrieved using SQL to query the code meaning. The other visual indication that the visualization workflow has completed is the VMF button is enabled (Fig. 5G). Clicking the VMF button changes the user interface to the model analysis tab as discussed in sections 5.2 and 5.3. If the user wishes to share results from any of the workflows, they can click on the share button (Fig. 5H) to reveal the share modal window (Fig. 5I) with various options to retrieve data and visualization results. Results from any of the workflows (if initiated by the user) are available as zipped data bundles for personal use. The movie results are available as mp4s as explained in Section 3.2. Recall, the workflows are executed in a distributed compute environment, thus, each data product will be from different locations. The Silverlight web application is responsible to query the workflow properties stored in the HydroTerre national job object to populate the user interface to direct users to the appropriate location to download and display model results.



Demonstration of Using Data, Model and Visualization Workflows By Lorne Leonard

160102020506	High Creek
160102030303	Temple Fork
160102040308	Willow Creek-Bear River
160102040502	Mantua Reservoir-Box Elder Creek
160102030202	Mollens Hollow-Blacksmith Creek
160102030205	Rock Creek
160201020303	Pineview Reservoir-Middle Fork Ogden Riv
160102030402	Spring Creek
160102020705	Newton Creek
160102030105	Hyrum Reservoir-Little Bear River
160102030306	Cottonwood Canyon-Logan River
160102010103	Big Creek
160102020701	Summit Creek
160102020505	Spring Creek

(Step 2) Select Workflows ●

Select Data Workflow
- ETV Data Workflow Version 1.0 [With Streams]
- ETV Data Workflow Version 1.1 [No streams]

Data Settings (Input into Model)

Select Model Workflow
- None (only want input data)
- PIHM v2.2 Cyberstar
- PIHM v2.2 XSEDE

Model Settings

Select Visualization Workflow
- Display model results (Fast small test)
- Load Model results (Large test)

Visualization Settings (Output Results)

(Step 3) User Details ●

□ Use Last Settings? Submit Model

○ Hide ○ Filter

Workflow Submission List and Status

Email Address	Project Name	HUC Name	HUC ID	Job Settings				Share	Repair
lnl3@psu.edu	PA_6/27/2013_12:29 PM	Upper Shaver Cr	020503020601	Status	v1.0 DWF	v1.0 MWF	v1.0 VWF		
lnl3@psu.edu	PA_6/27/2013_11:22 AM	Upper Shaver Cr	020503020601	Status	v1.0 DWF	v1.0 MWF	v1.0 VWF		
lnl3@psu.edu	PA_6/26/2013_5:09 PM	Upper Shaver Cr	020503020601	Status	v1.0 DWF	v1.0 MWF	v1.0 VWF		
lnl3@psu.edu	PA_6/25/2013_9:07 PM	Upper Shaver Cr	020503020601	Status	v1.0 DWF	v1.0 MWF	v1.0 VWF		
lnl3@psu.edu	PA_6/25/2013_8:31 PM	Upper Shaver Cr	020503020601	Status	v1.0 DWF	v1.0 MWF	v1.0 VWF		
lnl3@psu.edu	PA_6/25/2013_8:31 PM								
lnl3@psu.edu	UT_6/25/2013_11:45 AM								
lnl3@psu.edu	UT_6/25/2013_11:45 AM								
lnl3@psu.edu	UT_6/25/2013_11:45 AM								
lnl3@psu.edu	UT_6/25/2013_11:45 AM								
lnl3@psu.edu	UT_6/25/2013_11:45 AM								
lnl3@psu.edu	UT_6/25/2013_11:45 AM								
lnl3@psu.edu	UT_6/25/2013_11:45 AM								
lnl3@psu.edu	UT_6/25/2013_11:45 AM								

Share Data and/or Model Results

Please email me the
- □ Data Workflow Inputs (zip)
- □ Model Workflow Outputs (zip)
- □ Models results as Shapefile (zip)
- □ Movie results (mp4)

Share Cancel

Visualization Workflow Settings

□ Create Movie?

Movie Resolution: 864x486

☑ Add Project Name?

☑ Add Legend?

□ Render Movies By Variable?

☑ Over ride Average Technique?

OK Cancel

Fig. 5. The user selects which visualization workflow (a) they wish to apply. To change the visualization workflow, the user clicks on the settings button (b) and can change variables in the interface (c). Workflow settings are then applied to the level-12 HUC selection (d). Clicking on (e) submits ETV, data-model, model and visualization workflows. Workflow results are indicated in three color bars (f) and clicking on (g) enables the model analysis tab for the expert user to investigate

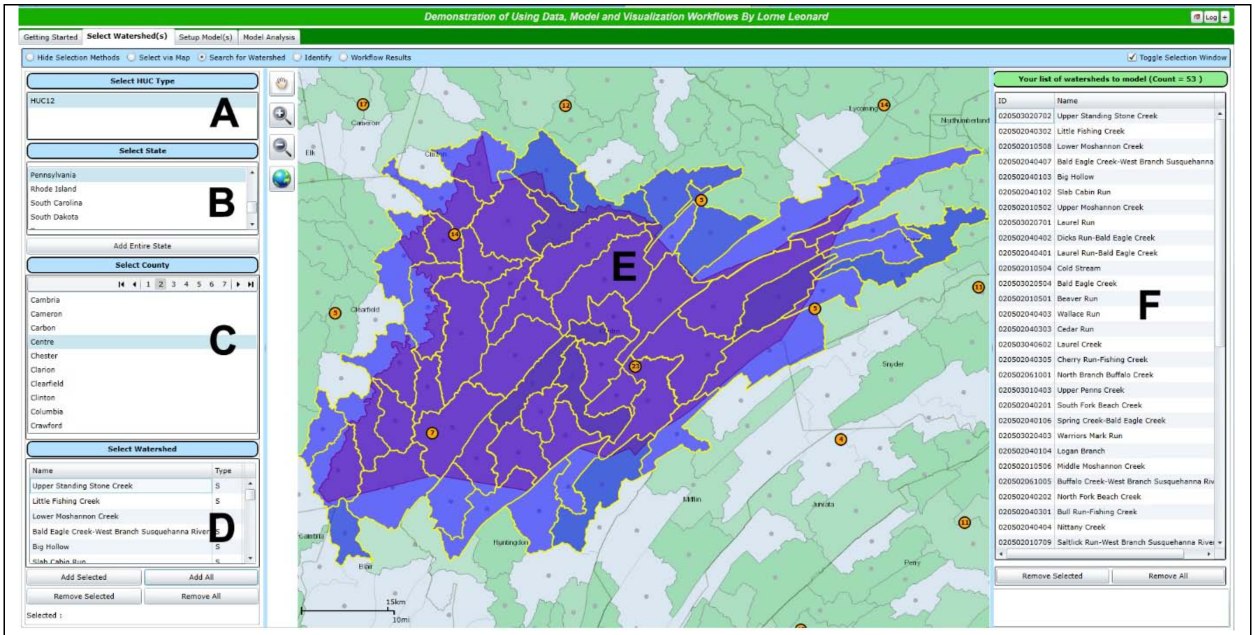
further. If the user wishes to download workflow results, clicking on (h) displays (i) for the user to indicate which datasets the user is interested in.

5. Demonstration of using an expert system to analyze and share hydrological models at level-12 HUC scales

The previous sections have discussed the system design and workflows implemented in the back-end of the web application using distributed compute resources. This section focuses on demonstrating the web application from the expert user perspective to test and share hydrological model results. Recall, as defined in section 1.3 constraints, the emphasis of this article is not the correctness of the hydrological model results, it is the design and capability for expert users to use HydroTerre to rapidly build and refine models using workflows. Section 5.1 provides an overview of how users select level-12 HUCs and assign workflows. Sections 5.2 and 5.3 explain how to use the interface for spatial and time-series data analysis.

5.1 Initiating workflows & provenance

Here, we demonstrate the process to model level-12 HUCs within the Centre County boundary located in the State of Pennsylvania, USA. There are 53 level-12 HUCs with a total area of 4740 square kilometers (Fig. 6). The no-stream data-model workflow was selected with default settings (Fig. 4A). The model workflow using one model server was chosen with a period of one year (Fig. 4F). The default value for averaging per year in this period was overridden to monthly intervals for visualization purposes (Fig. 4G). The movie generation option was enabled for the visualization workflow using one dedicated visualization server node. This was purposely done to benchmark how long the process takes in serial. Clearly, the more compute visualization resources available (distributed in parallel), the more efficiently the visualization results would be available to users by distributing the workflows. From this simple test, four data-model workflows failed due to domain decomposition. Twenty-seven failed in the modeling workflow stage due to poor meshes. Twenty-two visualization workflows succeeded using default settings and the entire process took approximately three hours on one server. The entire provenance is recorded in the workflow submission list (Fig. 5F) and is accessible to other users as a starting position for their own modeling needs. Workflows that did not succeed would require the user to change data (Fig. 4B) and model parameters (Fig. 4F) to generate successful results. As our focus is on visualization workflows, we do not demonstrate this process here, but refer the reader to article (Leonard and Duffy 2014b) for further details about these processes.



Demonstration of Using Data, Model and Visualization Workflows By Lorne Leonard

Getting Started | Select Watershed(s) | Setup Model(s) | Model Analysis

○ Hide Selection Methods ○ Select via Map ● Search for Watershed ○ Identify ○ Workflow Results

☑ Toggle Selection Window

Select HUC Type

HUC12

Select State

Pennsylvania
Rhode Island
South Carolina
South Dakota

Add Entire State

Select County

Cambria
Cameron
Carbon
Centre
Chester
Clarion
Clearfield
Clinton
Columbia
Crawford

Select Watershed

Name	Type
Upper Standing Stone Creek	S
Little Fishing Creek	S
Lower Moshannon Creek	
Bald Eagle Creek-West Branch Susquehanna River	S
Big Hollow	S
Slab Cabin Run	S

Selected :

Your list of watersheds to model (Count = 53)

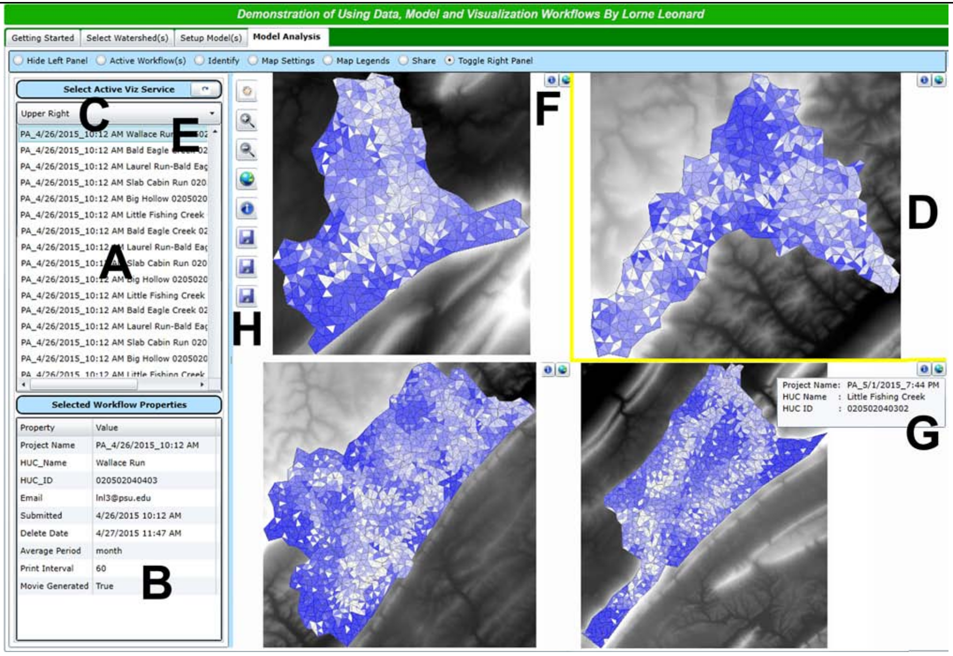
ID	Name
020503020702	Upper Standing Stone Creek
020502040302	Little Fishing Creek
020502010508	Lower Moshannon Creek
020502040407	Bald Eagle Creek-West Branch Susquehanna
020502040103	Big Hollow
020502040102	Slab Cabin Run
020502010502	Upper Moshannon Creek
020503020701	Laurel Run
020502040402	Dicks Run-Bald Eagle Creek
020502040401	Laurel Run-Bald Eagle Creek
020502010504	Cold Stream
020503020504	Bald Eagle Creek
020502010501	Beaver Run
020502040403	Wallace Run
020502040303	Cedar Run
020503040602	Laurel Creek
020502040305	Cherry Run-Fishing Creek
020502061001	North Branch Buffalo Creek
020503010403	Upper Penns Creek
020502040201	South Fork Beach Creek
020502040106	Spring Creek-Bald Eagle Creek
020503020403	Warriors Mark Run
020502040104	Logan Branch
020502010506	Middle Moshannon Creek
020502061005	Buffalo Creek-West Branch Susquehanna Riv
020502040202	North Fork Beach Creek
020502040301	Bull Run-Fishing Creek
020502040404	Nittany Creek
020502010709	Saltlick Run-West Branch Susquehanna River

Fig. 6. To initiate end-to-end workflows, the user needs to select at least one level-12 HUC. Here, we demonstrate selecting all level-12 HUCs within Centre County Pennsylvania, USA. The user selects the HUC type (a) and then selects the CONUS state of interest (b), followed by the county (c) and then selecting add all (d) to create a selection list shown as a map (e) and as a list (f).

5.2 Spatial Analysis

There are two ways for the user to analyze spatial results. The first method is to click on the VMF button in the workflow submission list (Fig. 5G) which will load the data, data-model and model workflow results into a web-based map within the model analysis tools (Fig. 7) as explained in Sections 2 and 3. This tool is extremely useful when searching millions of jobs and waiting for tasks to complete or to filter via project name, user, or level-12 HUC name. The other method is to select an active visualization service located in Fig.7A. The selected workflow properties describing the visualization properties are shown in Fig.7B.

The default layout includes four panels for side-by-side comparison of model results for the expert to visually inspect differences between model calibration parameters and the use of coordinated views to explore results (Roberts 2007). The user specifies the location of the map model results by selecting the panel name (Fig. 7C), which outlines the panel in yellow (Fig. 7D) and then choosing a model name (Fig. 7E). If the expert user wants to drill-down at one catchment, the user can expand the map service (Fig. 7F) to fill the workspace with only one map. When the user is modeling the same catchment with different parameters, to overcome any confusion, the user can scroll over the identification button (Fig. 7G) for a brief description.



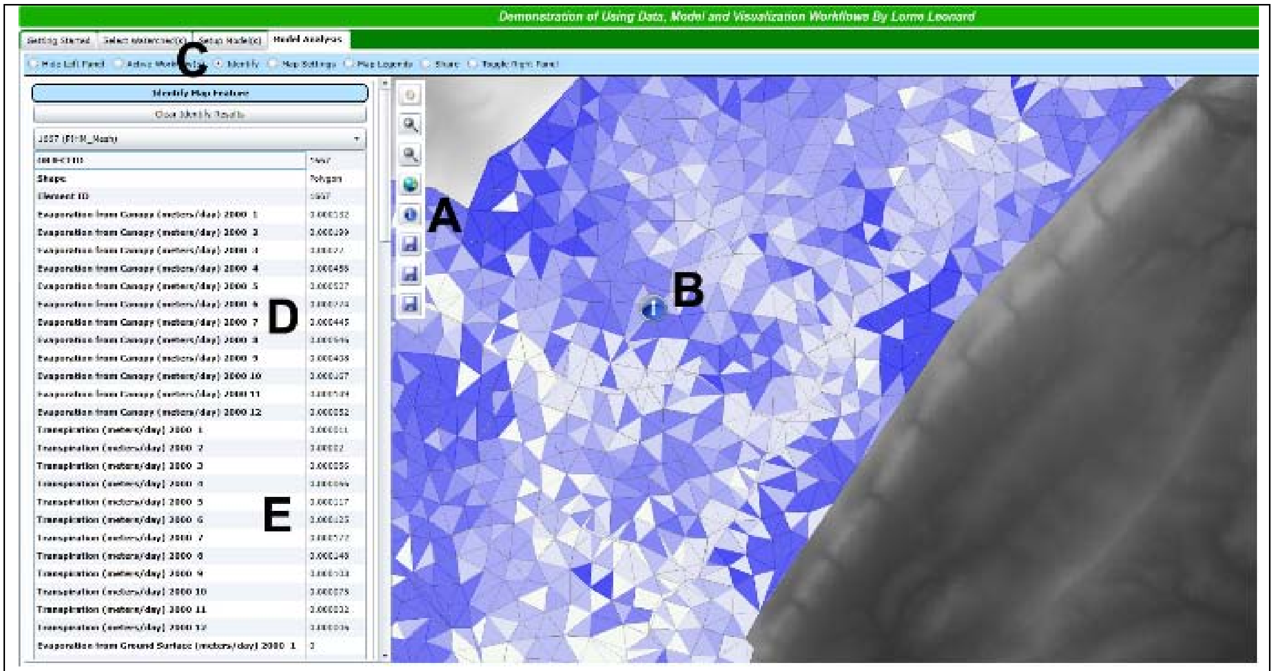
PA_4/26/2015_10:12 AM Wallace Ru 02
PA_4/26/2015_10:12 AM Bald Eagle 02
PA_4/26/2015_10:12 AM Laurel Run-Bald Eag
PA_4/26/2015_10:12 AM Slab Cabin Run 020
PA_4/26/2015_10:12 AM Big Hollow 0205020
PA_4/26/2015_10:12 AM Little Fishing Creek
PA_4/26/2015_10:12 AM Bald Eagle Creek 02
PA_4/26/2015_10:12 Laurel Run-Bald Eag
PA_4/26/2015_10:1 Slab Cabin Run 020
PA_4/26/2015_10: AM ig Hollow 0205020
PA_4/26/2015_10:12 AM Little Fishing Creek
PA_4/26/2015_10:12 AM Bald Eagle Creek 02
PA_4/26/2015_10:12 AM Laurel Run-Bald Eag
PA_4/26/2015_10:12 AM Slab Cabin Run 020
PA_4/26/2015_10:12 AM Big Hollow 0205020
PA_4/26/2015_10:12 AM Little Fishing Creek

Property	Value
Project Name	PA_4/26/2015_10:12 AM
HUC_Name	Wallace Run
HUC_ID	020502040403
Email	lnl3@psu.edu
Submitted	4/26/2015 10:12 AM
Delete Date	4/27/2015 11:47 AM
Average Period	month
Print Interval	60
Movie Generated	True

Project Name: PA_5/1/2015_7:44 PM
HUC Name : Little Fishing Creek
HUC ID : 020502040302

Fig. 7. The model analysis interface. To load a visualization service, the user selects from the active service list (a). Workflow properties about the service is shown in (b). To specify which quadrant, the user selects (c) the active corner, highlighted in yellow (d), and then clicks a visualization service (e). If the user wants to look at one map only, clicking at (f) will fill the quadrants with one map. Moving the cursor over the button at (g) reveals basic information about the map service to help the user identify which project is in the quadrant. The user controls each map with the same toolbar (h) for zooming in/out, pan, identification and save workflow results to their desktop.

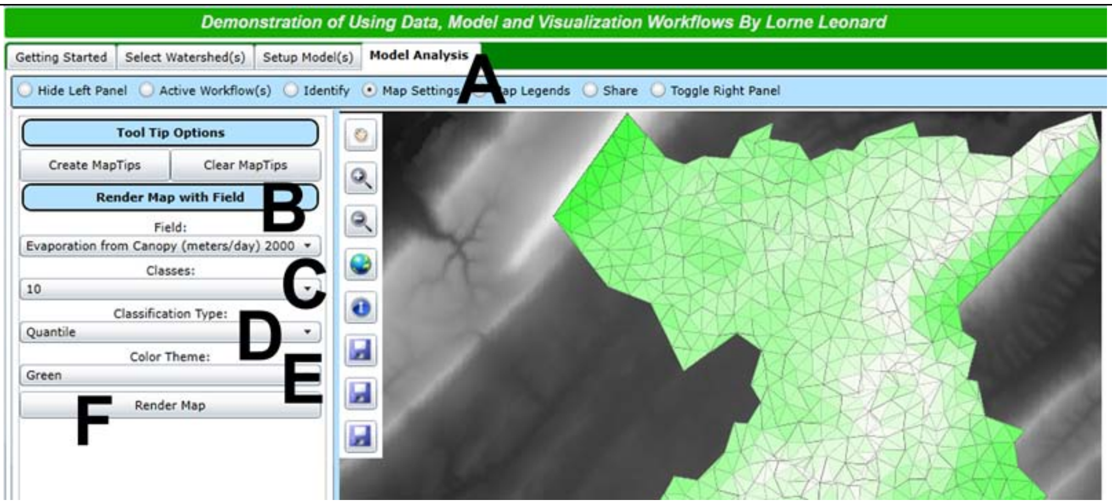
The user can interact with the selected map (outlined in yellow) with standard map tools (zoom, pan, extent) located in Fig. 7H. The tool bar contains tools to save images or shape files of the current catchment geometry to the users' desktop. The identification tool (Fig. 8A) enables users to drill-down at mesh cell values (Fig. 8B). The left panel (Fig. 8C) aids the user to investigate all model result variables for the entire simulation duration at one location. For example, all the monthly evaporation values (Fig. 8D) and transpirations (Fig. 8E) are grouped together for rapid visual inspection of seasonal trends.



OBJECTID	1557
Shape	Polygon
Element ID	1557
Evaporation from Canopy (meters/day) 2000 1	0.000152
Evaporation from Canopy (meters/day) 2000 2	0.000199
Evaporation from Canopy (meters/day) 2000 3	0.000277
Evaporation from Canopy (meters/day) 2000 4	0.000459
Evaporation from Canopy (meters/day) 2000 5	0.000527
Evaporation from Canopy (meters/day) 2000 6	0.000774
Evaporation from Canopy (meters/day) 2000 7	0.000445
Evaporation from Canopy (meters/day) 2000 8	0.000646
Evaporation from Canopy (meters/day) 2000 9	0.000438
Evaporation from Canopy (meters/day) 2000 10	0.000127
Evaporation from Canopy (meters/day) 2000 11	0.000584
Evaporation from Canopy (meters/day) 2000 12	0.000052
Transpiration (meters/day) 2000 1	0.000011
Transpiration (meters/day) 2000 2	0.00002
Transpiration (meters/day) 2000 3	0.000055
Transpiration (meters/day) 2000 4	0.000066
Transpiration (meters/day) 2000 5	0.000117
Transpiration (meters/day) 2000 6	0.000125
Transpiration (meters/day) 2000 7	0.000572
Transpiration (meters/day) 2000 8	0.000149
Transpiration (meters/day) 2000 9	0.000133
Transpiration (meters/day) 2000 10	0.000075
Transpiration (meters/day) 2000 11	0.000032
Transpiration (meters/day) 2000 12	0.000036
Evaporation from Ground Surface (meters/day) 2000 1	0

Fig. 8. The user can identify individual features by clicking on the id tool (a) and then clicking on the map (b). Doing so automatically shows the identification panel (c). Model variables have been grouped by time (d, e) to help the user inspect values for the entire duration.

Another method for the user to investigate mesh cell properties is to re-render the mesh cells by changing the map settings (Fig. 9A). The model mesh fields, for example evaporation, ground water, etc. can be selected as a field for the map rendering (Fig. 9B). There is an option to specify the number of classes (Fig. 9C) and type of classification (equal interval, quantile) at Fig. 9D. The user can specify the color theme at Fig. 9E, and then is required to click on the render map button (Fig. 9F) to re-render the model mesh.



Field: Evaporation from Canopy (meters/day) 2000

Classes: 10

Classification Type: Quantile

Color Theme: Green

Fig. 9. The user can change rendered model values by changing the map settings (a). The user selects from (b) to change the model field value, the number of classes (c), the classification type (d), and the color theme (e) to define the rendering theme. Then, the user is required to click on render map button (f) to update the mesh of the selected map quadrant.

These tools are web-based interactive for mesh cell inspection. The other method to inspect model results is to play pre-rendered movies of the model results (Fig. 10) generated during the visualization workflow as described in Section 3. If the user specified movie creation (Fig. 5C), the share button (Fig. 10A) displays the movie panel. The movie loops through all variables starting with element identifications (Fig. 10B). Each movie has a legend (Fig. 10C) and scale bar (Fig. 10D). The user controls movie playback with standard controls of play, stop, and pause (Fig. 10E). The user can zoom in and out of the movie to look for details and save images from the movie. Additionally, the user can change the movie playback speed and seek option (Fig. 10F).

The movie panel is purposely placed left of the map so the user can inspect model results interactively on the right panel at the same time. Rendered movies assist the user to quickly identify mesh cells that are not spatially continuous with its neighbors. When this occurs, the right panel can be used by the expert-user to identify and drill-down to the mesh cell. Then, the user can re-submit a new end-to-end workflow by cloning the current workflow and adjusting parameters to address these issues. As with all the workflows, the mp4 movies generated can be downloaded for personal use. Downloading is encouraged, as the visualization services are automatically deleted in 24 hours with default settings.

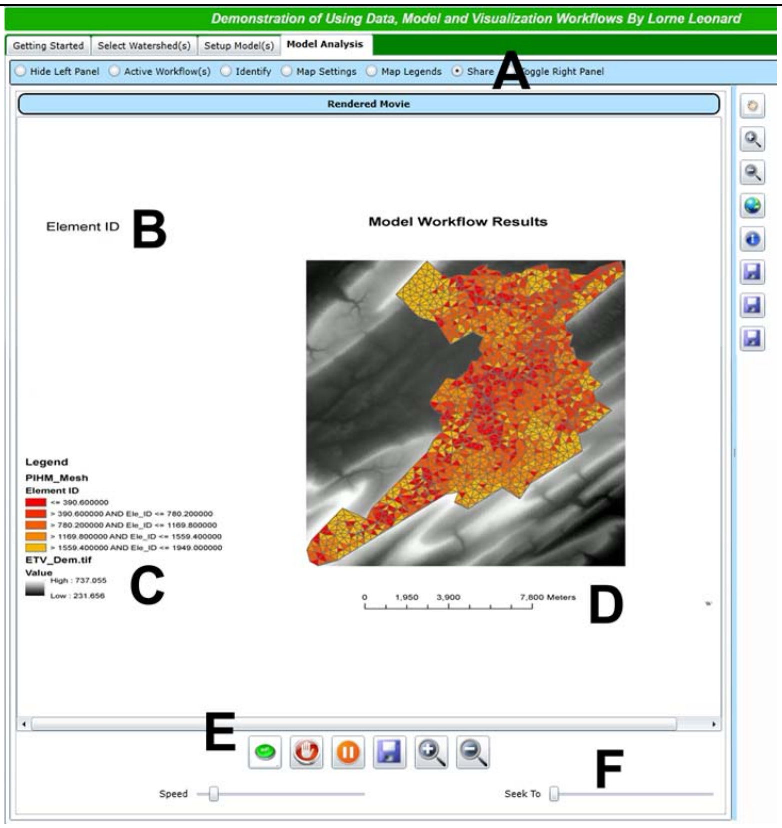
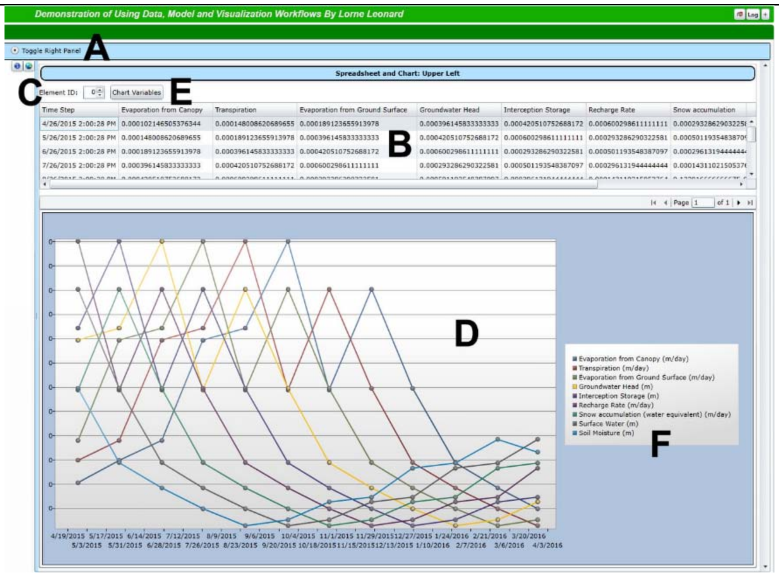


Fig. 10. If the user specified render movies option, the share button (a) displays the movie panel. The model variable is shown at location (b) while the movie is playing. The variable legend is shown (c) as well as a scale bar (d). Movie controls such as play, stop, and pause are available (e), as well as the ability to change speed and seek (f).

5.3 Time series data analysis

There are two ways to visualize time-series model results by clicking on the right panel (Fig.11 A). The first is a spreadsheet (Fig.11 B) with each column representing the model output results such as evaporation, transpiration, groundwater head, and recharge rate. The user can sort values by clicking on the column head or title. These results start with the first mesh element and the user can specify which element they are interested in by changing the numeric up/down control (Fig.11 C). The second method is by using charts (Fig.11 D). By default, all the variables are plotted at the same time, however, the user can change which variables to plot (Fig.11 E) which automatically updates the legend (Fig.11 F). As with the spatial data, the time-series data can be saved to the users' local environment. Additionally, the interface layout has been designed this way, so the user can have model movie results on the left side, the interactive map in the center, and the time-series results on the right. This provides full control to the expert user to investigate the model results with a variety of different tools at the same time for rapid prototyping using national data products.



Demonstration of Using Data, Model and Visualization Workflows By Lorne Leonard

Spreadsheet and Chart: Upper Left

Element ID: 0

Time Step	Evaporation from Canopy	Transpiration	Evaporation from Ground Surface	Groundwater Head	Interception Storage	Recharge Rate	Snow accumulation
4/26/2015 2:00:28 PM	0.000102146505376344	0.000148008620689655	0.000189123655913978	0.000396145833333333	0.000420510752688172	0.000600298611111111	0.0002932862903225
5/26/2015 2:00:28 PM	0.000148008620689655	0.000189123655913978	0.000396145833333333	0.000420510752688172	0.000600298611111111	0.000293286290322581	0.000501193548387O
6/26/2015 2:00:28 PM	0.000189123655913978	0.000396145833333333	0.000420510752688172	0.000600298611111111	0.000293286290322581	0.000501193548387097	0.000296131944444
7/26/2015 2:00:28 PM	0.000396145833333333	0.000420510752688172	0.000600298611111111	0.000293286290322581	0.000501193548387097	0.000296131944444444	0.000143110215053

Fig. 11. By clicking on toggle switch (a), a spreadsheet of model variables is available (b). Each mesh cell can be examined by changing the value at (c) that automatically updates the spreadsheet and chart (d). The user can control which variables to display in the spreadsheet and chart (e) and reflected in the legend (f).

6. Conclusion

The first HydroTerre paper (Leonard and Duffy 2013) introduced data as a service, using ETV data workflows to retrieve data at a level-12 HUC scale. The second paper

(Leonard and Duffy 2014b) explained the processes that transform ETV data with data-model and model workflows to model any level-12 HUC in the CONUS as a model service. This paper demonstrates visualization as a service that builds upon these two papers and describes how to create interactive web-based spatial and data visualizations for any level-12 HUC using end-to-end workflows. All three services have been designed using a distributed compute environment for scalability and reproducibility by storing workflow parameters within HydroTerre national job objects stored in SQL databases.

Our approach demonstrates how automated web-based data access and workflows allow seamless allocation of resources (software, data, and HPC resources) with minimal interaction from the user and without the modeler losing control of creating and analyzing hydrological models. By balancing hardware and software configurations, we demonstrated the feasibility of transforming data sources from several federal agencies that amounts to 100's of terabytes of disk storage. The automatically stored provenance for end-to-end workflows developed here assures reproducibility of model simulations from ETV data sets. Intermediate workflow results generated during data-model and model workflow are not permanently stored on our HydroTerre system. The versioned ETV datasets are kept to reproduce the initial datasets used by the workflows and the SQL national job objects store the versioned workflow parameters. Therefore, we can execute millions of end-to-end workflows without storing results that we have shown will require many petabytes of storage. By combing all three software-workflow services, we have demonstrated that a web-based expert user can rapidly create reproducible hydrological models anywhere in the CONUS. This, we hope, will engage modelers to share not only their data and model results, but also their workflows to improve hydrological science.

7. Future direction

This research focuses on the important issue of eliminating hurdles involved with using physics based models, such as PIHM, in a HPC environment using workflows in a distributed compute environment. Our research has been dedicated towards the level-12 HUC scale due to its feasibility as an online web software service. The next phase of this research is to scale end-to-end workflows from level-12 HUCs up to multiple level-2 HUCs (18 level-2 HUC regions in the CONUS). Initial analysis of our workflows indicate improvements will be required to data structures and HydroTerre hardware configurations to scale data workflows (Leonard 2015a) (Leonard et al. 2015a). These data structures will also need to adopt cross-domain naming conventions for sharing data and model results with other scientific domains (Peckham 2014).

New visual analytic tools are required to guide users to identify and replace missing data (e.g. bedrock depth, soil parameters and NHD networks) due to the large quantities and complexities identified during the end-to-end workflows with CONUS datasets (Leonard 2015a). Furthermore, new visual analytic tools are necessary to address inconsistencies with stream flow directions from National Hydrography Datasets (Leonard 2015a) (Leonard et al. 2015b). Having consistent stream flow directions is necessary to create quality meshes. Likewise, new visual analytical processes need to be incorporated within the web application to enhance the hydrologists' ability to compare patterns, knowledge generation, and annotation with model results rather than depend on the users' memory for analysis (Huang et al. 2015) (Sips et al. 2012) (Sacha et al. 2014) (Groth and Streefkerk 2006).

Finally, our future goal is to make these services available to expert users without restrictions. We plan to continue with two approaches. The first is extending the software tools developed for the Extreme Science and Engineering Discover Environment (XSEDE 2014) for users to retrieve data bundles for their models (not necessarily Hydrology) and their own HPC environment. These users will manage their own disk storage and service units (i.e. CPU time). The second approach, with support, is to provide seamless access for all services. As demonstrated in this article, we have established the feasibility of using standard HPC environments (XSEDE and The Penn State Institute for CyberScience (Penn State 2015)). However, there are constraints with these HPC environments (security, disk allocation and service units). The other HPC environment is to use cloud resources such as Amazon Web Services. The dynamic scalability of cloud services is appealing and we have purposely designed HydroTerre to be ready for cloud infrastructure by treating each service independently with strict connectivity between services. At present, the high costs associated with network and data storage with our workflows (hundreds of terabytes to petabytes) is prohibitive using commercial cloud infrastructure. However, we believe working with private cloud environments such as those with The Penn State Institute for CyberScience are competitive and important to continue our HydroTerre research.

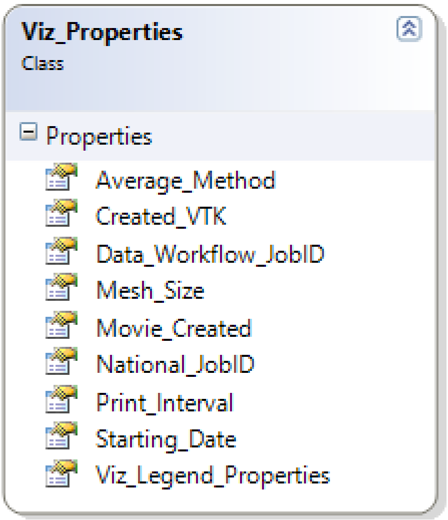
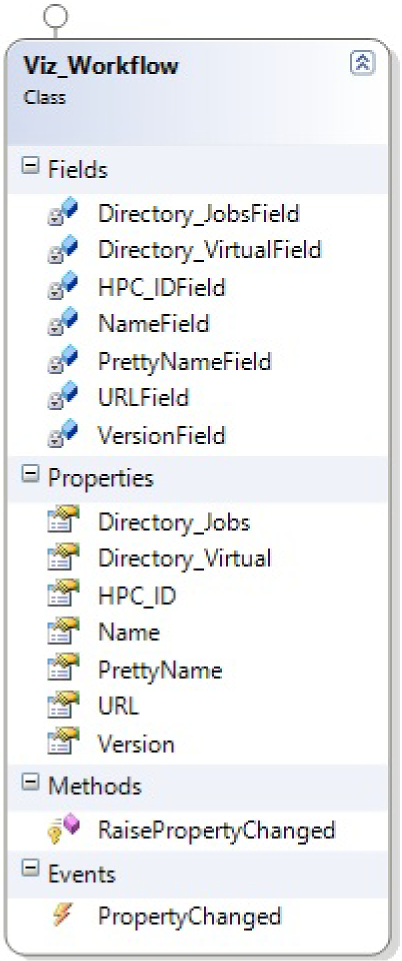
Acknowledgements

This research was supported in part by the National Science Foundation through XSEDE resources provided by the XSEDE Science Gateways program (TG-EAR120019), NSF EarthCube (GEO-44417482), NSF INSPIRE (IIS-1344272), EPA (96305901), NOAA (NA10OAR4310166), CC-NIE (1245980). The authors would like to acknowledge the support from the Institute for CyberScience Director Padma Raghavan and Penn State Institutes for Energy and the Environment Director Tom Richard at The Pennsylvania State University. Additionally, the support of Ken Miller from Penn State Research Network.

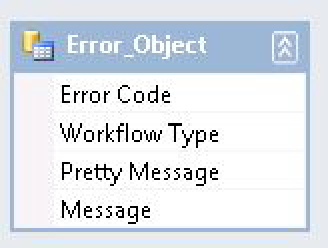
Appendix A: Project database objects

The section below describe details about the visualization workflow objects described in Section 2.2 and the processes in Section 3. All the schema diagrams are available to view at (http://www.hydroterre.psu.edu/Development/Help_Model/AppendixA.aspx). Storing data as XML objects increases the flexibility of versioning and reproducibility between data-model workflows.

Appendix A1: Visualization Properties

 Viz_Properties Class ⊟ Properties Average_Method Created_VTK Data_Workflow_JobID Mesh_Size Movie_Created National_JobID Print_Interval Starting_Date Viz_Legend_Properties	 Viz_Workflow Class ⊟ Fields Directory_JobsField Directory_VirtualField HPC_IDField NameField PrettyNameField URLField VersionField ⊟ Properties Directory_Jobs Directory_Virtual HPC_ID Name PrettyName URL Version ⊟ Methods RaisePropertyChanged ⊟ Events PropertyChanged

Appendix B: Error object

Error object used to communicate between workflows and web application interface.		
 Error_Object Error Code Workflow Type Pretty Message Message	Error Code	Unique error code (Appendix B1 to B3)
	Workflow Type	Key to workflow type and version
	Pretty Message	Meaningful message show to user
	Message	Actual message returned from software and/or hardware.

B1: Data workflow		**B2: PIHM model workflow**	
Error Code Range	**Message**	**Error Code Range**	**Message**
-100 to -1	Hardware Problems		
0 to 499	Workflow Status	-2050 to -2059	Invalid element
500 to 599	Raster Processing	-2040 to -2049	Time Job Cancelled
600 to 699	Vector Processing	-2030 to -2039	Threshold Job Cancelled
700 to 799	Topology Processing	-2000 to -2029	Sundials Errors
800 to 899	XML generation	-100 to -1999	PIHM File Errors
900 to 999	Image Processing	0 to -99	Workflow Status
1000 to 1200	Soil & Geology Processing	1 to 100	Hardware Problems

B3: Visualization workflow			
Error Code Range	**Message**	**Error Code Range**	**Message**
-100 to -1	Hardware Problems	1020 to 1800	Visualization Data Workflow Processing
0 to 499	Workflow Status	1800 to 2000	Map Services
1000 to 1010	Data Workflow Processing	2000 to 2200	Map and Image Rendering
1010 to 1020	Model Workflow Processing	2200 to 3000	Video Creation

Appendix C: Private web services used by prototype

The prototype web application discussed in Section 4 uses private web services as those described in Sections 2 and 3 that have XMD schemas available at (http://www.hydroterre.psu.edu/Development/Help_Model/ DataSources.aspx). This prototype uses three web services, (1) The Job service is used to create project jobs to handle the workflows as shown in Section 4.1; (2) Model Service used to create HUC-12 submission lists and update/submit workflow properties to job objects stored as shown in Section 2; (3) Project service used to populate the project listed discussed in Section 4.

References

Ames, D. P., Horsburgh, J. S., Cao, Y., Kadlec, J., Whiteaker, T., and Valentine, D. (2012). “HydroDesktop: Web services-based software for hydrologic data discovery, download, visualization, and analysis.” *Environmental Modelling and Software*, 37, 146–156.

Ames, D. P., Quinn, N. W. T., Eds, A. E. R., Jones, N., Nelson, J., Swain, N., Christensen, S., Tarboton, D., and Dash, P. (2014). “Tethys : A Software Framework for Web-Based Modeling and Decision Support Applications.” *Proceedings of the 7th International Congress on Environmental Modelling and Software, June 15-19, San Diego, California, USA*.

Arguez, A., and Vose, R. S. (2011). “The Definition of the Standard WMO Climate Normal: The Key to Deriving Alternative Climate Normals.” *Bulletin of the American Meteorological Society*, 92(6), 699–704.

Bell, M. (2008). “Service-oriented modeling service analysis, design, and architecture.” John Wiley & Sons, Hoboken, N.J.

Bell, M. (2010). “SOA modeling patterns for service-oriented discovery and analysis.” John Wiley & Sons, Hoboken, N.J.

Bowers, S. (2012). “Scientific Workflow, Provenance, and Data Modeling Challenges and Approaches.” *Journal on Data Semantics*, 1, 19–30.

Castronova, A. M., Goodall, J. L., and Ercan, M. B. (2013). “Integrated modeling within a hydrologic information system: An OpenMI based approach.” *Environmental Modelling and Software*, Elsevier Ltd, 39, 263–273.

CUASHI. (2015). “HydroShare.” <http://hydroshare.cuahsi.org/>.

CyberSTAR. (2014). “A Scalable Terascale Advanced Resource for Discovery through Computing.” <http://www.ics.psu.edu/infrast/>.

Davidson, S., and Freire, J. (2008). “Provenance and scientific workflows: challenges and opportunities.” *Proceedings of the 2008 ACM SIGMOD …*, 1–6.

Deelman, E., Gannon, D., Shields, M., and Taylor, I. (2009). “Workflows and e-Science: An overview of workflow system features and capabilities.” *Future Generation Computer Systems*, Elsevier B.V., 25(5), 528–540.

Demir, I., and Krajewski, W. F. (2013). “Towards an integrated Flood Information System: Centralized data access, analysis, and visualization.” *Environmental Modelling and Software*, Elsevier Ltd, 50, 77–84.

ESRI. (2014). “ArcGIS Server.” <http://www.esri.com/software/arcgis/arcgisserver>.

FFmpeg team. (2015). “FFmpeg.”

Fielding, R. T., and Taylor, R. N. (2002). “Principled design of the modern Web architecture.” *ACM Transactions on Internet Technology*, 2(2), 115–150.

Free Software Foundation. (2015). “GNU Wget.”

Gilles, D., Young, N., Schroeder, H., Piotrowski, J., and Chang, Y. J. (2012). “Inundation mapping initiatives of the iowa flood center: Statewide coverage and detailed urban flooding analysis.” *Water (Switzerland)*, 4(1), 85–106.

Goodall, J., Horsburgh, J., Whiteaker, T., Maidment, D., and Zaslavsky, I. (2008). “A first approach to web services for the National Water Information System.” *Environmental Modelling & Software*, 23(4), 404–411.

Goodall, J. L., Robinson, B. F., and Castronova, A. M. (2011). “Modeling water resource systems using a service-oriented computing paradigm.” *Environmental Modelling & Software*, Elsevier Ltd, 26(5), 573–582.

Google. (2015). “Protocol Buffers.” <https://developers.google.com/protocol-buffers/>.

Granell, C., Díaz, L., and Gould, M. (2010). “Service-oriented applications for environmental models: Reusable geospatial services.” *Environmental Modelling & Software*, Elsevier Ltd, 25(2), 182–198.

Groth, D. P., and Streefkerk, K. (2006). “Provenance and annotation for visual exploration systems.” *IEEE Transactions on Visualization and Computer Graphics*, 12(6), 1500–1510.

Horsburgh, J. S., Morsy, M. M., Castronova, A. M., Goodall, J. L., Gan, T., Yi, H., Stealey, M. J., and Tarboton, D. G. (2015). “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*, n/a–n/a.

Huang, D., Tory, M., Aseniero, B. A., Bartram, L., Bateman, S., Carpendale, S., Tang, A., and Woodbury, R. (2015). “Personal Visualization and Personal Visual Analytics.” 21(3), 420–433.

Leonard, L.N. (2015a). “HYDROTERRE: Towards An Expert System For Scaling Hydrological Data And Models From Hill-Slopes To Major-River Basins.” The Pennsylvania State University.

Leonard, L., and Duffy, C. (2014a). “HydroTerre: Selecting up-stream level-12 HUCs using depth-first graphs anywhere in the continental USA.” *11th International Conference on Hydroinformatics HIC 2014*.

Leonard, L., and Duffy, C. J. (2013). “Essential Terrestrial Variable data workflows for distributed water resources modeling.” *Environmental Modelling & Software*, Elsevier Ltd, 50, 85–96.

Leonard, L., and Duffy, C. J. (2014b). “Automating data-model workflows at a level 12 HUC scale: Watershed modeling in a distributed computing environment.” *Environmental Modelling & Software*, Elsevier Ltd, 61, 174–190.

Leonard, L., Madduri, K., and Duffy, C. (2015a). “Tuning Heterogeneous Computing Platforms for Large-scale Hydrology Data Management.” *IEEE Transactions on Parallel and Distributed Systems*, 1–12.

Leonard, L., Madduri, K., and Duffy, C. J. (2015b). “Graph-based Analysis for Large-scale Hydrological Modelling.” *IEEE Exploring Graphs at Scale*, Big Graph Visual Analytics Challenges and Opportunities IEEE VIS 2015 Workshop, Chicago.

Martinec, J., Rango, A., and Roberts, R. (1994). *The snowmelt runoff model (SRM) user’s manual*. Berne, Switzerland.

Microsoft. (2014a). “SQL Server.” <https://www.microsoft.com/en-us/sqlserver/default.aspx>.

Microsoft. (2014b). “Service-Oriented Architecture.” <http://msdn.microsoft.com/en-us/library/bb977471.aspx>.

Microsoft. (2014c). “Database Markup Language.” <http://msdn.microsoft.com/en-us/library/bb399400%28v=vs.110%29.aspx>.

Microsoft. (2014d). “Silverlight.” <http://www.microsoft.com/silverlight/>.

Microsoft. (2014e). “Globally Unique Identifier.” <http://msdn.microsoft.com/en-us/library/aa373931(VS.85).aspx>.

Miller, K. (2015). “PennState Research network.” <http://rn.psu.edu/>.

NLDAS. (2011). “North American Land Data Assimilation System.” <http://ldas.gsfc.nasa.gov/nldas/NLDAS2forcing.php>.

Parajka, J., Merz, R., and Blöschl, G. (2005). “A comparison of regionalisation methods for catchment model parameters.” *Hydrology and Earth System Sciences*, 9(3), 157–171.

Peckham, S. D. (2014). “The CSDMS Standard Names : Cross-Domain Naming Conventions for Describing Process Models , Data Sets and Their Associated Variables.” *Proceedings of the 7th International Congress on Environmental Modelling and Software, June 15-19, San Diego, California, USA*.

Penn State. (2015). “The Penn State Institute for CyberScience.” <http://ics.psu.edu/>.

Qu, Y., and Duffy, C. J. (2007). “A semidiscrete finite volume formulation for multiprocess watershed simulation.” *Water Resources Research*, 43(8), 1–18.

Roberts, J. C. (2007). “State of the art: Coordinated & multiple views in exploratory visualization.” *Proceedings - Fifth International Conference on Coordinated and Multiple Views in Exploratory Visualization, CMV 2007*, (Cmv), 61–71.

Sacha, D., Stoffel, a, Stoffel, F., Kwon, B., Ellis, G., and Keim, D. (2014). “Knowledge Generation Model for Visual Analytics.” *Visualization and Computer Graphics, IEEE Transactions on*, PP(99), 1.

Seaber, P. R., Kapinos, F. P., and Knapp, G. L. (1987). *Hydrologic unit maps*. U.S. G.P.O. ; For sale by the Books and Open-File Reports Section, U.S. Geological Survey, [Washington]; Denver, CO.

Silva, C. T., Anderson, E., Santos, E., and Freire, J. (2011). “Using VisTrails and Provenance for Teaching Scientific Visualization.” *Computer Graphics Forum*, 30(1), 75–84.

Silva, C. T., Freire, J., and Callahan, S. P. (2007). “Provenance for Visualizations: Reproducibility and Beyond.” *Computing in Science & Engineering*, 9(5), 82–89.

Sips, M., Kothur, P., Unger, A., Hege, H. C., and Dransch, D. (2012). “A visual analytics approach to multiscale exploration of environmental time series.” *IEEE Transactions on Visualization and Computer Graphics*, 18(12), 2899–2907.

Tarboton, D. G., Horsburgh, J. S., Maidment, D. R., Whiteaker, T., Zaslavsky, I., and Piasecki, M. (2009). “Development of a Community Hydrologic Information System.” *18th World IMACS Congress and MODSIM09 International Congress on Modelling and Simulation*, L. T. . Anderssen, R.S Braddock, R.D Newham, ed., Cairns, Australia, 988–994.

Tarboton, D. G., Idaszak, R., Horsburgh, J. S., Heard, J., Ames, D., Goodball, J. L., Merwade, V., Couch, A., Arrigo, J., Hooper, R., Valentine, D., and Maidment, D. R. (2014). “HydroShare: Advancing Collaboration through Hydrologic Data and Model Sharing.” *International Environmental Modelling and Software Society (iEMSs) 7th International Congress on Environmental Modelling and Software*.

Turuncoglu, U. U., Dalfes, N., Murphy, S., and DeLuca, C. (2013). “Toward self-describing and workflow integrated Earth system models: A coupled atmosphere-ocean modeling system application.” *Environmental Modelling & Software*, Elsevier Ltd, 39, 247–262.

USGS. (2013). “USGS HUC.” <http://water.usgs.gov/GIS/huc.html>.

Valentine, D., Couch, A., Ames, D. A. N., Goodall, J. L., Band, L., Merwade, V., Arrigo, J., and Hooper, R. (2014). “A RESOURCE CENTRIC APPROACH FOR ADVANCING COLLABORATION THROUGH HYDROLOGIC DATA AND MODEL SHARING.” *11th*

International Conference on Hydroinformatics, HIC 2014, New York City, USA.

World Wide Web Consortium. (2014a). “Simple Object Access Protocol.” <http://www.w3.org/TR/soap12-part1/>.

World Wide Web Consortium. (2014b). “Web Services Description Language.” <http://www.w3.org/TR/wsdl>.

World Wide Web Consortium. (2014c). “Extensible Markup Language.” <http://www.w3.org/TR/xml11/#charsets>.

XSEDE. (2014). “Extreme Science and Engineering Discovery Environment.” <https://www.xsede.org>.