Design of a Virtual Watershed Client for the WC-WAVE Project

Chase D. Carthen, Thomas J. Rushton, Christine M. Johnson,
Aaron Hesson, Daniel Nielson, Bryan Worrell
Computer Science and Engineering
University of Nevada
Reno, NV, USA
chase@nevada.unr.edu

John W. Anderson, Roger Lew,
Nicholas R. Wood, Matthew Ziegler
Virtual Technology and Design
University of Idaho
Moscow, ID 83844
jwa@uidaho.edu

Donna M. Delparte, W. Joel Johansen
Geosciences
Idaho State University
Pocatello, ID, 83209
delparte@isu.edu

Sergiu M. Dascalu, Frederick C Harris, Jr.
Computer Science and Engineering
University of Nevada
Reno, NV, USA
Fred.Harris@cse.unr.edu

***Abstract*—The Virtual Watershed Client, built using the Unity game engine, provides a tool that allows researchers, students, and others to observe and analyze geospatial datasets and theoretical model data acquired from the Virtual Watershed Platform. The Virtual Watershed Platform is used as a service to acquire data that is used by the Virtual Watershed Client to create terrain, satellite imagery, shapes, and model data. A detailed design and the current implementation of this Virtual Watershed Client will be discussed in this paper.**

Keywords—Visualization; Geospatial; Model Data; Virtual Watershed Platform; Virtual Watershed Client; Terrain

I. Introduction

Today there is a need to visualize and analyze datasets captured by sensors in a watershed. These sensors provide information about the surrounding environment of the watershed and allow for researchers to run models to analyze the data captured from these sensors. Researchers use this acquired data to develop models that effectively reflect the environment of the watershed. These models can then be used to predict water runoff, precipitation, snow melt, underground water flow, and other environmental factors within a watershed.

Models consist of many input and output variables that require visualization and tuning. There are several existing tools used to visualize this data and provide a researcher sufficient information to modify variables in their models. These tools include QGIS [2], ArcMap [3], and other Geographical Information System (GIS) and non-GIS programs. However, most of these tools lack the ability to analyze data in an interactive first person perspective.

The Virtual Watershed Client is currently being implemented and has been designed to provide researchers a 3D analytical tool to visualize model data. Researchers will be able to provide data to the Virtual Watershed Platform, and the Virtual Watershed Client can then be used to visualize this data. The Virtual Watershed Platform uses the Open Geospatial Consortium (OGC) services for serving data [4]. The client will use libraries such as the Geospatial Data Abstraction Library (GDAL) to parse any data served to the Virtual Watershed Client [5]. Researchers can then observe this visualization to evaluate the outcomes of their model runs.

The current implementation of the Virtual Watershed Client was built using the Unity game engine [1]. The Unity game Engine allows our platform to be published on many potential platforms, such as Windows, Mac, Linux, and the web browser.

There is not a lot of related work in the area of virtual watersheds and their visualization. There are two major projects that relate to the work that we have done. In [6] the authors discuss methods of determining the optimal locations for dam placement in order to harvest water. The authors of [7] describe a toolkit built for hydrologic modelling using a geographic information system such as ArcView GIS. Their system, the Automated Geospatial Watershed Assessment tool (AGWA) is capable of executing and then visualizing results inside of ArcView GIS from models such as the Soil and Water Assessment Tool (SWAT) and Kinematic Runoff and Erosion Model (KINEROS2).

The rest of the paper presents the design of the Virtual Watershed Client. Section II discusses the overall scope of the WC-WAVE project. We provide specifics of the design of the Virtual Watershed Client in Section III. Section IV describes an example of our current implementation of the Virtual Watershed Client. Finally, in Section V we give our closing remarks and present ideas for future work.

II. WC-WAVE Project Overview

The Western Consortium for Watershed Analysis, Visualization, and Exploration (WC-WAVE) [8] is a tri-state consortium composed of researchers and students in Idaho, Nevada, and New Mexico. Its overall aim is to study the localized impact of climate change on high-mountain catchments. The WC-WAVE project is comprised of three components: Watershed Science, Visualization and Data Cyberinfrastructure (CI), and Workforce Development. Participants in these three components are collaborating to better understand interactions between precipitation, snow-pack, groundwater flow, and other watershed properties within mountain catchments.

To enhance the collaborative nature of this project and future work, as well as to promote data exploration and analysis, one of the main outcomes of the WC-WAVE project is a Virtual Watershed Platform (VWP). The VWP framework will provide data access and visualization through individual workstations (desktop), web-based environments, and advanced interactive 3D environments, including stereo projection, and immersive CAVE and Virtual Reality environments. The VWP will also allow a user to run specific hydrologic models and visualize the results.

During the preliminary meetings for this proposal, project team members put together the design for the Virtual Watershed Platform. A figure illustrating this design can be seen in Fig. 1.

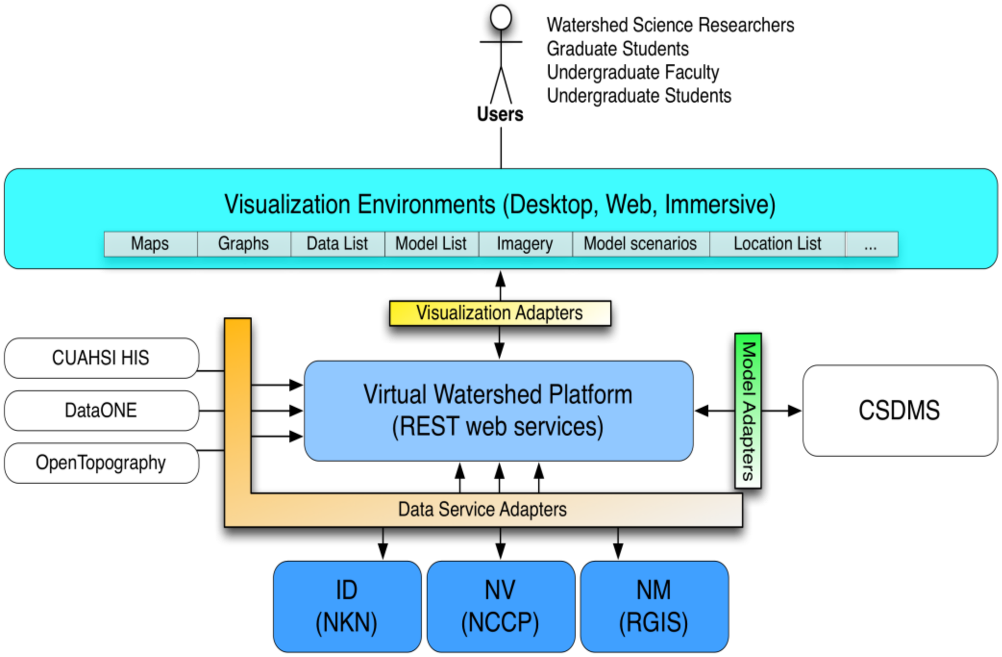

Fig. 1. Diagram illustrating the relationships of proposed project components; virtual watershed platform, data management sevices, and portals.

To implement and test the VW platform, project participants selected four watersheds throughout the tri-state region to focus on: Dry Creek (as shown in Fig. 2); Reynolds Creek Experimental Watershed in Idaho; Jemez watershed in New Mexico; and Lehman Creek in Nevada. These watersheds were chosen partly for their unique characteristics. For example, Dry Creek and Reynolds Creek provide an ideal environment for studying and developing tools related to snow pack, while the Jemez watershed is more suitable for vegetation studies. Another reason they were chosen was for their extensive data networks. Since the models within the VW platform call for several different forms of data, watersheds with adequate datasets and instrumentation were required.

Fig. 2. A picture of Dry Creek.

Members of the Watershed Science component of the WC-WAVE project selected four main models to include in the VW platform: RIPFLOW, SRH-2D [9], ParFlow [10], and iSNOBAL [11]. The SRH-2D (Sedimentation and River Hydraulics- Two-Dimensional) model can be used for simulating river channel flow and hydraulics; the ParFlow (PARallel FLOW) model is useful for modeling soil moisture and the interaction between surface and sub-surface flow; finally, the iSNOBAL (image SNOw mass and energy BALance) model is used to predict snowmelt and runoff.

The iSNOBAL model is part of our current implementation, as well as the Image Processing Workbench (IPW) suite of software tools [12, 13]. It is effective at predicting snowmelt and runoff when applied to a range of watershed sizes (1 to 2500 km^2) as well as temporal ranges (one week to an entire season). When run through IPW, iSNOBAL requires three different types of input: an initial conditions image, precipitation images, and input forcing data images. The same initial condition image can be used for an entire model run and is comprised of seven bands: elevation, roughness length, total snowcover depth, average snowcover density, active snow layer temperature, average snowcover temperature, and percentage of liquid H_2O saturation. Precipitation images consist of four bands (total precipitation mass, percentage of precipitation mass that was snow, density of snow portion of precipitation, average precipitation temperature) and are only included for times of precipitation events (i.e. storms).

Individual input forcing data images are required for each time step (typically three hours) throughout the model run, and are made up of six bands: incoming thermal (long-wave) radiation, air temperature, vapor pressure, wind speed, soil temperature, and net solar (short-wave) radiation. All of these images can be visualized in the 3D environment of the VW platform. This allows the user to check for errors, anomalies, and patterns before an actual model run. Using these images as input, iSNOBAL uses a two layer snow model to calculate energy and mass balance terms and produces two images as output. The first is a ten-band energy and mass flux image which includes predicted evaporation, snowmelt, and runoff layers. The second is a nine-band snow conditions image which contains layers for predicted thickness of snowcover, snow density, and mass of the snowcover. Similar to the input images, these output layers can be visualized with the VW platform for further analysis. A visualization of several of the iSNOBAL variables can be seen in Fig. 3.

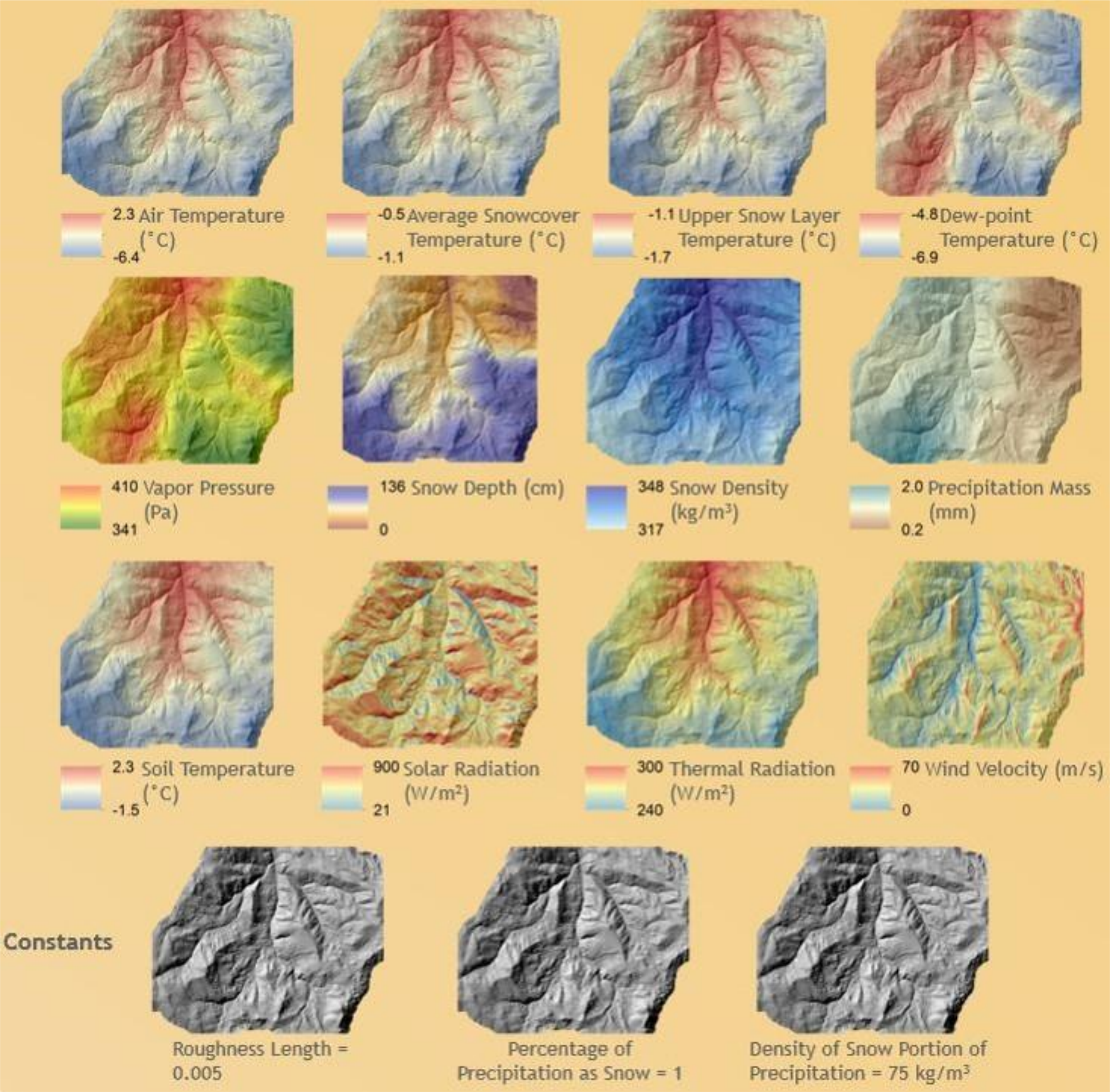
2.3 Air Temperature (°C) -6.4
-0.5 Average Snowcover Temperature (°C) -1.1
-1.1 Upper Snow Layer Temperature (°C) -1.7
-4.8 Dew-point Temperature (°C) -6.9
410 Vapor Pressure (Pa) 341
136 Snow Depth (cm) 0
348 Snow Density (kg/m³) 317
2.0 Precipitation Mass (mm) 0.2
2.3 Soil Temperature (°C) -1.5
900 Solar Radiation (W/m²) 21
300 Thermal Radiation (W/m²) 240
70 Wind Velocity (m/s) 0
Constants
Roughness Length = 0.005
Percentage of Precipitation as Snow = 1
Density of Snow Portion of Precipitation = 75 kg/m³

Fig. 3. A visualization of different Isnobal Variables.

III. Design of the Virtual Watershed Client

The primary intention for the Virtual Watershed Client is to aid in the visualization of data collected by environmental scientists for better understanding of the watershed environment and the models (iSNOBAL, Parflow, etc.). Several dependencies are needed for the Virtual Watershed Client to be functional. The Virtual Watershed Client requires the Unity game engine, libraries similar to GDAL, and the OGC services. The Virtual Watershed Client's architecture, functional requirements, and main use case are explained in detail in this section.

A. *Model View Controller*

The Virtual Watershed Client utilizes a Model View Controller (MVC) architectural pattern for the user interface (Fig. 4). The Model component is responsible for querying the Virtual Watershed Platform for data, the View component is responsible for displaying the data with the currently applied settings, and the Controller is responsible for handling any configuration changes and movement in the simulation world.

The Model component retrieves data through the OGC services that can be displayed in the form of terrains, shapes (rivers, streams, and roads), model data, and imagery. Fig. 5 shows a detailed view of the Model component. Libraries such as GDAL are used to parse data acquired from the OGC services into an acceptable form for the Virtual Watershed Client.

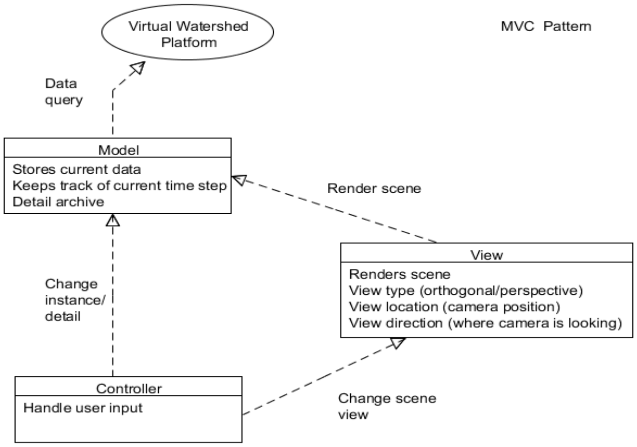
Virtual Watershed Platform
MVC Pattern
Data query
Model
Stores current data
Keeps track of current time step
Detail archive
Render scene
View
Renders scene
View type (orthogonal/perspective)
View location (camera position)
View direction (where camera is looking)
Change instance/ detail
Controller
Handle user input
Change scene view

Fig. 4. A high level diagram of the Virtual Watershed client representing its major connections.

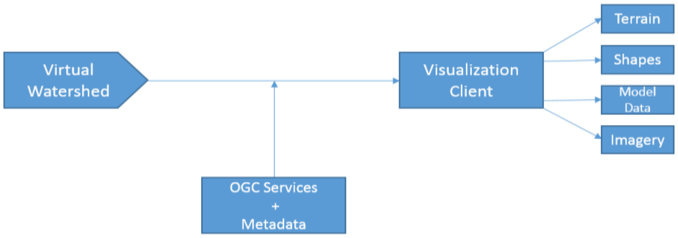

Fig. 5. A block diagram representing the Model component of the Virtual Watershed Client

B. *Functional Requirements*

The Virtual Watershed Client architecture requires certain graphical user interface (GUI), data visualization, statistical analysis, and configuration capabilities [14], [15]. The following is a brief list of functional requirements of the application:

- The Virtual Watershed Client shall have an intuitive GUI for configuring simulation settings such as selecting watershed regions, terrain quality, player movement speed, and download bandwidth. The GUI shall have the option to be enabled or disabled while the simulation is running to adjust settings.
- The Virtual Watershed Client shall use Digital Elevation Maps (DEM) and data corresponding to trees, vegetation, water, etc. to display the watershed as a 3D simulation. The user has the ability to apply various overlays such as contours, snow height, vegetation, satellite imagery, and geological maps to the watershed terrain to view specific data.
- The Virtual Watershed Client shall provide the ability to run statistic metrics (mean, standard deviation, etc.) on the data. Temporal data can also be viewed in the form of either 2D or 3D graphs, allowing users to view any trends in the data over a period of time.
- The Virtual Watershed Client shall provide the flexibility of retrieving data for multiple watershed regions from the Virtual Watershed Platform. The Virtual Watershed Client will be used to visualize information from that retrieved data. The user can easily reconfigure the simulation with a different dataset, as well as filter datasets by specific fields.

C. *Use Case*

The users of the Virtual Watershed Client are able to create watershed simulations using data retrieved from the Virtual Watershed Platform. In addition to selecting the data for the simulation, users also have the option of configuring various settings to analyze desired variables of the data in more detail. One of the approaches taken to enable this level of analysis is to allow the user to immerse themselves into the simulation in the sense that he or she can interact with various objects throughout the environment such as trees and sensors. Users also have the option to configure settings for data download speeds, image quality, etc., in order to give greater flexibility in viewing aspects of the data that are most important to the specific users' analyses. Fig. 6 demonstrates a general use case model.

The system being referred to in Fig. 6 is the running application of the Virtual Watershed Client [16]. Within the use case model, the server is any outside service that the system utilizes to acquire data. When the application launches, the user can select a dataset to be loaded from file or downloaded from the server, as well as configure the watershed. The system will use the selected dataset and configurations to start and generate a simulation. Then the

Virtual Watershed Client will display the terrain and user/model selected weather conditions. Once the visualization has begun, the user can configure a model run, and have it prepared for simulation by the system. Any data required for the generation of the model run will be acquired from the server at this time. While the model is running, information corresponding to water runoff, snow melt, underground moisture, etc. will be displayed.

IV. Application Example: the Dry Creek Watershed

The Unity engine is a platform for building interactive virtual environments. It provides an integrated development environment (IDE) consisting of a GUI and scripting to define the appearance and behavior of objects in the virtual environment. The Unity engine allows developers to combine 2D and 3D visual assets, audio, and network access to data. The Unity engine is built with Mono, which is a cross-platform open source implementation of Microsoft's .NET framework [17, 18]. Using Mono allows for procedural control, creation, and manipulation of objects in UnityScript (Javascript), C#, and boo. The engine provides a well-optimized code base capable of displaying high-fidelity interactive 3D virtual environments with real-time physics and shading. Although Unity offers great potential to geoscience visualization and education outreach, the IDE has a steep learning curve and is not natively equipped with essential features like geo-referencing, Gregorian time, or the ability to handle geospatial datasets (GeoTiffs, ShapeFiles, etc.) common to GIS Desktop solutions.

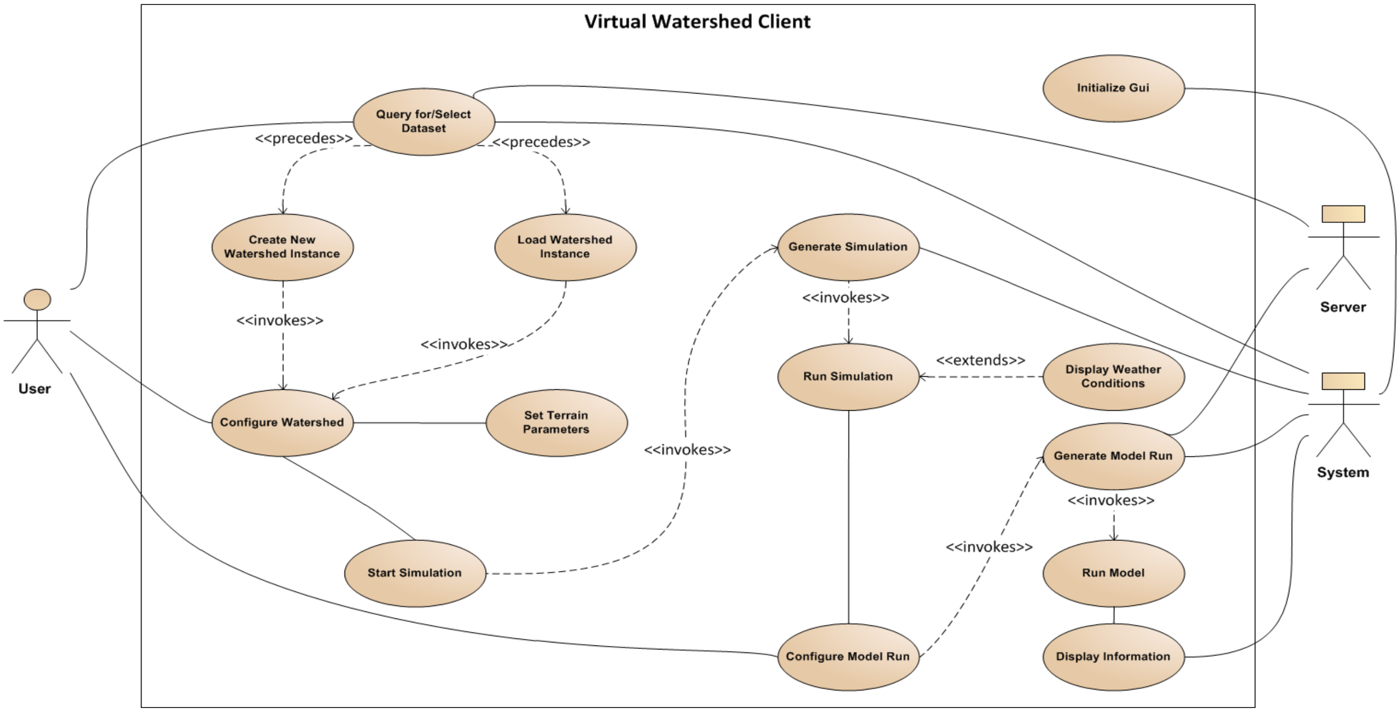

Fig. 6. A use case model of the Virtual Watershed Client.

The Virtual Watershed Client will provide a simple GUI for communicating to a centralized Virtual Watershed Platform. However, instead of communicating with the Virtual Watershed Platform, datasets are acquired and preprocessed from a local file system.

Several visualizations have been developed for the Virtual Watershed Client. The Virtual Watershed Client provides a first-person-perspective of a watershed. When available, high-resolution LiDAR can be represented by the terrain. High-resolution orthoimagery is then placed on the terrain, and procedurally splatted with detail textures and vegetation (as shown in Figure 1). Once a base map has been constructed, scientific data can be incorporated. A *Projector* with a customized shader allows for choropleths (thematic maps) to be overlaid onto the terrain (as shown in Fig. 7). These maps could represent characteristics of terrain topology like slope or spatial-temporal variables associated with hydrology models such as iSNOBAL, Parflow, or PRMS.

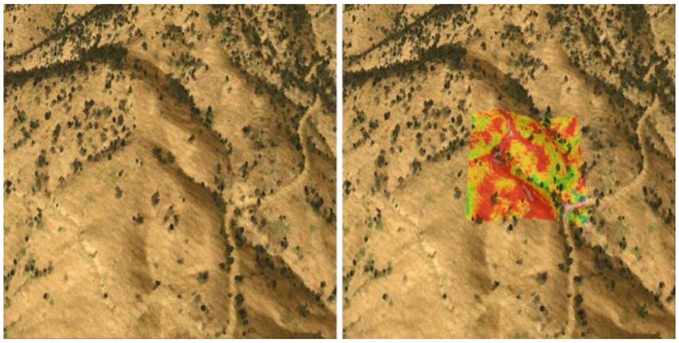

Fig. 7. A picture of the Dry Creek sub-catchment with choropleth (right) and without choropleth (left).

The first-person perspective provides a unique method of viewing such datasets, or for comparison of the virtual world with the real world environment being visualized (as shown in Fig. 9 and Fig. 11). Representations of 3D objects can be incorporated using industry standard 3D design tools such as 3dsMax [19] and Blender [20]. These 3D objects can help guide users while they are visiting the virtual watershed. Weather stations built in such a fashion are shown in Fig. 8.

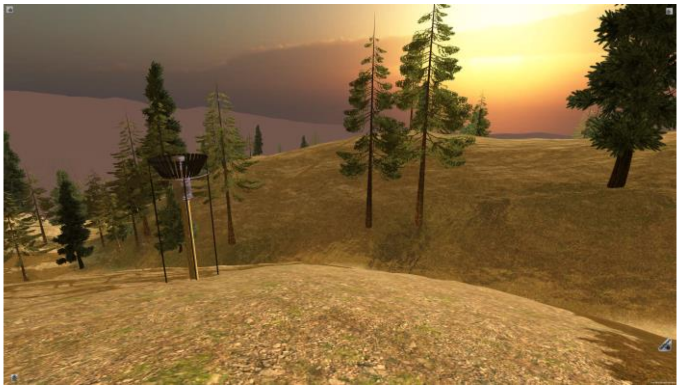

Fig. 8. The sunset and weatherstation.

The Virtual Watershed Client will incorporate a heads-up-display (HUD) for visualizing temporal-spatial variables (as shown in Fig. 10). To the top left of the figure, climate data is temporally depicted (the spatial variability at a single point in time tends to be low relative to the temporal variability). Below the climate data iSNOBAL outputs or aggregated outputs are depicted both spatially and temporally. The top row represents Surface Water Equivalents (SWE).

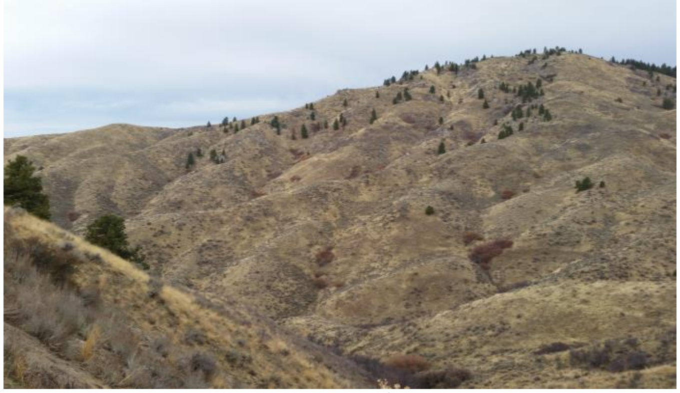

Fig. 9. A photograph of Dry Creek.

The second row contains the total energy gain or loss (Delta Q). The Melt panel depicts snow or ice melt. Lastly, "10-day SWI" is cumulative Surface Water Inputs over 240 simulation steps. The animated choropleth representations are alpha blended images produced for each time step. This eliminates perceptually unnatural discontinuities. Each of the time series panels has 5 separate lines depicting gradient weighted time series trends. The weights are derived by constructing an orthonormal basis of the terrain's gradient. In the center of the HUD is a 3D cross-section representation of a sub catchment and the current snow depth (exaggerated 10-fold). The user can rotate the object about the x and y axes by clicking and dragging. In the top-right corner is a control that allows the user to move the plane of the cross-section north and south.

The panel at the bottom of the HUD provides time and date controls as well as trends depicting precipitation.

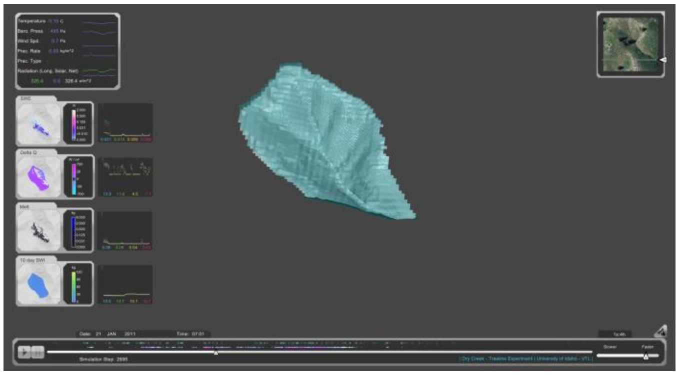

Fig. 10. A cross sectional view of the Dry Creek sub-catchment. Featuring a HUD with time series graphs.

V. Conclusions & Future Work

The Virtual Watershed Client is a tool currently being implemented that can be utilized by researchers to analyze datasets. The Dry Creek example clearly demonstrates the effectiveness of the Virtual Watershed Client by giving the user the ability to visualize time series graphs and choropleths. However, the current implementation still lacks the ability to interface with the Virtual Watershed Platform.

Integration with the Virtual Watershed Platform will allow researchers to download datasets and can be used to create the Virtual Watershed Client for Unity's Web Player in the web browser. The Unity Web Player has a security sandbox that does not allow for access to locally stored files. The integration of the Virtual Watershed Platform and overall design of the project will allow us to build the application for other platforms and to download datasets for the Unity Web Player.

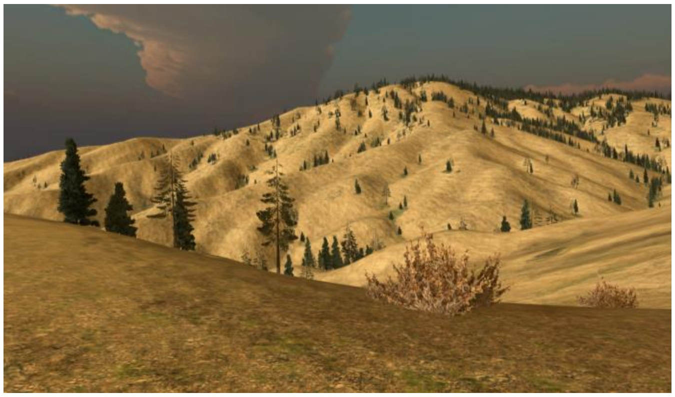

Fig. 11. The Virtual Watershed Client image of Dry Creek.

We have plans to implement our design in OpenGL and to incorporate Head Mounted Displays (HMD) such as the Oculus Rift for an immersive 3D Virtual Reality (VR) experience [21]. OpenGL would be used to create software based on the design for a six-sided CAVE (Cave automatic virtual environment) or University of Idaho's IQ station [22]. The Oculus Rift HMD hardware would be used to provide an immersive experience with the current and updated Unity implementation of the Virtual Watershed Client. The CAVE will allow users to view the world as if they were standing at the real world location. The IQ Station employs a 3D TV, allowing the user to view the world in stereoscopic 3D. All three environments will take advantage of sensors such as the OptiTrack [23] and the Microsoft Kinect sensor [24] to spatially track users within the physical environment and allow them to interact with the world presented by the Virtual Watershed Client.

Acknowledgments

This material is based in part upon work supported by the National Science Foundation under grant numbers IIA-1329469, IIA-1329470 and IIA-1329513. 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.

References

[1] Unity3d.com, 'Unity - Game Engine', 2015. [Online]. Available: http://unity3d.com/. [Accessed: 26- Feb- 2015].http://unity3d.com/

[2] Www2.qgis.org, 'Welcome to the QGIS project!', 2015. [Online]. Available: http://www2.qgis.org/en/site/. [Accessed: 28- Feb- 2015].

[3] Esri.com, 'ArcGIS Platform', 2015. [Online]. Available: http://www.esri.com/software/arcgis. [Accessed: 28- Feb- 2015].

[4] Gdal.org, 'GDAL: GDAL - Geospatial Data Abstraction Library', 2015. [Online]. Available: http://www.gdal.org/. [Accessed: 26- Feb- 2015].

[5] Opengeospatial.org, 'Open Geospatial Consortium | OGC', 2015. [Online]. Available: http://www.opengeospatial.org/. [Accessed: 26- Feb- 2015].

[6] D. Patel, M. Dholakia, N. Naresh and P. Srivastava, 'Water Harvesting Structure Positioning by Using Geo-Visualization Concept and Prioritization of Mini-Watersheds Through Morphometric Analysis in the Lower Tapi Basin', *J Indian Soc Remote Sens*, vol. 40, no. 2, pp. 299-312, 2011.

[7] S. MILLER, D. SEMMENS, D. GOODRICH, M. HERNANDEZ, R. MILLER, W. KEPNER and D. GUERTIN, 'The Automated Geospatial Watershed Assessment tool', *Environmental Modelling & Software*, vol. 22, no. 3, pp. 365-377, 2007.

[8] Westernconsortium.org, 'Western Tri-State Consortium', 2015. [Online]. Available: http://westernconsortium.org/. [Accessed: 28- Feb- 2015].

[9] Y. Lai, "Two-Dimensional Depth-Averaged Flow Modeling with an Unstructured Hybrid Mesh," J. Hydraul. Eng., 136(1), 2010, pp. 12-23.

[10] R. Maxwell, *et al*., Parflow User's Manual "ParFlow user's manual." *International Ground Water Modeling Center Report GWMI* 1.2009 (2009): 129p. [Online]. http://inside.mines.edu/~rmaxwell/ParFlow.manual.9-15-09.pdf. [Accessed: 26- Feb- 2015].

[11] D. Marks, J. Domingo, D. Susong, T. Link, and D. Garen, "A spatially distributed energy balance snowmelt model for application in mountain basins". *Hydrological Processes*, *13*(1213), 1999, 1935-1959.

[12] Zero.eng.ucmerced.edu, 'IPW command: isnobal', 2015. [Online]. Available: http://zero.eng.ucmerced.edu/snow/whitney/ly/www/man1/isnobal.html. [Accessed: 26- Feb- 2015].

[13] IPW Website, 'Software Tools for Hydro-Climatic Modeling and Analysis', 2015. [Online]. Available: http://ftp://ftp.nwrc.ars.usda.gov/ipw/intro.html. [Accessed: 26- Feb- 2015]

[14] Sommerville, *Software engineering*. Boston: Pearson, 2011.

[15] N. Juristo, A. Moreno and M. Sanchez-Segura, 'Guidelines for Eliciting Usability Functionalities', *IIEEE Trans. Software Eng.*, vol. 33, no. 11, pp. 744-758, 2007.

[16] S. Lauesen, 'Task descriptions as functional requirements', *IEEE Softw.*, vol. 20, no. 2, pp. 58-65, 2003.

[17] Mono-project.com, 'Home | Mono', 2015. [Online]. Available: http://www.mono-project.com/. [Accessed: 28- Feb- 2015].

[18] Msdn.microsoft.com, '.NET Framework and .NET SDK Downloads', 2015. [Online]. Available: https://msdn.microsoft.com/en-us/vstudio/aa496123.aspx. [Accessed: 26- Feb- 2015].

[19] Autodesk.com, '3d Modeling and Rendering Software | 3ds Max | Autodesk', 2015. [Online]. Available: http://www.autodesk.com/products/3ds-max/overview. [Accessed: 28- Feb- 2015].

[20] Blender.org, 'blender.org - Home of the Blender project - Free and Open 3D Creation Software', 2015. [Online]. Available: http://www.blender.org/. [Accessed: 28- Feb- 2015].

[21] Oculus VR, 'Oculus VR | Oculus Rift - Virtual Reality Headset for Immersive 3D Gaming', 2015. [Online]. Available: https://www.oculus.com/. [Accessed: 26- Feb- 2015].

[22] Lib.uidaho.edu, 'IQ Station - University of Idaho Library', 2015. [Online]. Available: http://www.lib.uidaho.edu/IQ/. [Accessed: 28- Feb- 2015].

[23] Optitrack.com, 'Motion Capture Systems - OptiTrack', 2015. [Online]. Available: http://www.optitrack.com/. [Accessed: 26- Feb- 2015].

[24] Microsoft.com, 'Kinect for Windows', 2015. [Online]. Available: http://www.microsoft.com/en-us/kinectforwindows/. [Accessed: 26- Feb- 2015].