ABSTRACT

Open-Source Data Catalog for Managing Hydrologic Data

Sarva Teja Pulla
Department of Civil & Environmental Engineering, BYU
Master of Science

The SERVIR Water Observations Data Integrator (SWODI) has been designed to promote better hydrologic and earth observations data sharing across institutions. The purpose of this application is to create a tool that will allow for easy access from multiple data sources, such as the Consortium of Universities for the Advancement of Hydrologic Science, Inc (CUAHSI), HydroServers, NASA Global Land Data Assimilation System (GLDAS), and NASA ClimateServ. The online application has been developed using the Tethys Platform, a software development framework created at Brigham Young University (BYU). The web app supports managing and discovering observational or modeled time series data. It also allows users to catalog data from existing HydroServers. The app incorporates several features for data discovery and management from data.cuahsi.org as well as HydroDesktop. However, it is unique seeing as it does not require a HydroServer to be registered with the CUAHSI Hydrologic Information System. Due to the open-source nature of the application, it is highly customizable, thus making it possible to add newer datasets with ease. This paper will explain the software design, development, and functionality of SWODI in greater detail.

Keywords: Open-source, Decision support, CUAHSI, HydroServer
ACKNOWLEDGEMENTS

I would like to thank all of the professors who have supported me throughout my time at Brigham Young University. Thank you to Dr. Dan Ames for taking me in as a graduate research assistant and for being a great advisor and mentor during my graduate studies. I would also like to thank my committee members Dr. Jim Nelson and Dr. Wood Miller, along with Dr. Norm Jones, for their continued guidance and feedback.

Additionally, I would like to thank members of the BYU Civil Engineering Office, specifically Kim Glade, for all of her help with administrative intricacies. Thank you to all the members of the BYU Hydroinformatics lab for their help, specifically Shawn Crawley and Zhiyu (Drew) Li. I am forever grateful for all open-source developers and their contributions, without which it would have been impossible to create this application. I would also like to thank my parents for all of the sacrifices they have made so that I could pursue my education and career. Finally, thank you to my wonderful wife, Melanie, for all of her love and encouragement and without whom this paper never would have been completed.
# TABLE OF CONTENTS

LIST OF TABLES ........................................................................................................................................ vi

LIST OF FIGURES ....................................................................................................................................... viii

1. Introduction ................................................................................................................................................. 1

2. Components of the web application ........................................................................................................... 5
   2.1 Tethys Framework ................................................................................................................................. 5
   2.2 Spatial Database Storage ......................................................................................................................... 6
   2.3 Map Publishing .................................................................................................................................... 7

3. Methods ..................................................................................................................................................... 9
   3.1 Software Organization ........................................................................................................................... 9
      3.1.1 Tethys App Folder Structure ......................................................................................................... 10
      3.1.2 Model ........................................................................................................................................... 11
      3.1.3 View ............................................................................................................................................ 12
      3.1.4 Controllers ................................................................................................................................... 13
         3.1.4.1 Home Controller .................................................................................................................. 14
         3.1.4.2 Catalog Controller ............................................................................................................... 14
         3.1.4.3 Add Server Controller ......................................................................................................... 14
         3.1.4.4 Details Controller ............................................................................................................... 14
         3.1.4.5 ClimateServ Controller ........................................................................................................ 15
         3.1.4.6 Update Controller ............................................................................................................... 16
         3.1.4.7 Data Rods Controller ........................................................................................................... 16
   3.2 Application User Interface Overview .................................................................................................... 16
3.2.1 Accessing the Application ................................................................. 17
3.2.2 Viewing Data from HydroServers ....................................................... 17
3.2.3 Exploring GLDAS Data ...................................................................... 19
3.2.4 Exploring Climate Serv Data ............................................................... 21
3.2.5 Managing the Admin Interface ......................................................... 24

4. Results ....................................................................................................... 27
   4.1 Limitations ............................................................................................ 28
   4.2 Future Developments .......................................................................... 28

5. Conclusion .................................................................................................. 29

References .................................................................................................... 31
LIST OF TABLES

Table 3-1: Description of HTML Pages .............................................................................................................. 13
LIST OF FIGURES

Figure 1-1: Components of HIS ........................................................................................................... 2

Figure 2-1: Tethys Platform Software Architecture .............................................................................. 6

Figure 3-1: Software Architecture ........................................................................................................ 10

Figure 3-2: Tethys App Directory Structure ......................................................................................... 11

Figure 3-3: Details Controller Sequence .............................................................................................. 15

Figure 3-4: Tethys.byu.edu Apps Library ............................................................................................. 17

Figure 3-5: List of Available HydroServers ......................................................................................... 18

Figure 3-6: HydroServer Site Pop-Up ................................................................................................... 18

Figure 3-7: Site Details Page with Time Series for Majagual ................................................................. 19

Figure 3-8: Selecting a Point Interaction ............................................................................................. 19

Figure 3-9: GLDAS Data Form .............................................................................................................. 20

Figure 3-10: GLDAS Time Series ........................................................................................................ 21

Figure 3-11: Selecting a Polygon Interaction ....................................................................................... 21

Figure 3-12: Digitized Polygon ............................................................................................................ 22

Figure 3-13: Climate Serv Console ....................................................................................................... 22

Figure 3-14: Climate Serv Time Series ............................................................................................... 23

Figure 3-15: Nepal Shapefile ................................................................................................................. 23

Figure 3-16: HydroCatalog Settings Interface ..................................................................................... 24

viii
Figure 3-17: Adding a HydroServer ........................................................................................................................................ 24

Figure 3-18: Adding a HydroServer from HIS Central Servers .................................................................................. 25

Figure 3-19: List of Current HydroServers in Catalog .................................................................................................. 25
1. INTRODUCTION

It is critical to have access to historical and/or real-time water observations data to model and forecast changing trends in hydrologic data [Stewart, 2014]. The introduction of OGC (Open Geospatial Consortium) WaterML format has standardized the format for storing and exchanging water observations, thus reducing the barrier for interoperability of water observations data (http://www.opengeospatial.org/standards). The Consortium of Universities for the Advancement of Hydrologic Science, Inc. (CUAHSI) has developed several tools such as Hydrologic Information System (HIS) [Maidment, 2008] and HydroDesktop [Ames, 2009] to further improve the accessibility of hydrologic data.

HIS has several components that work together to seamlessly share and manage hydrologic data. The three components of HIS are HIS Central, HydroServer [Horsburgh et al. 2009] and HIS Client. HIS Central acts as a metadata catalog that stores relevant information about data sources. The HydroServer is a tool that publishes and stores the data sources. Finally, these data sources can be discovered or searched through HIS Clients such as the desktop based HydroDesktop and web based HydroClient (data.cuahsi.org). The following figure shows the core components that make up the CUAHSI HIS.
Figure 1-1: Components of HIS

HydroDesktop is a desktop-based application that allows users to discover, download and visualize data that has been stored in the CUAHSI HIS. The integration of DotSpatial GIS Library (https://github.com/DotSpatial/DotSpatial) within HydroDesktop allows developers to add custom plug-ins to the existing interface, which then work automatically within HydroDesktop. This proved to be one of the stronger points of HydroDesktop allowing for a distributed software development paradigm [Baldwin & Clark, 2000]. Furthermore, HydroDesktop is incredibly modular, allowing for it to be divided into simpler modules. For example, within the app, users can add HydroServers directly and build a local catalog on their own computer.

Although HydroDesktop is a unique and helpful tool, there are limitations within this app. As of now, it can only be run on Windows, which may limit certain users. It requires admin privileges to install the app itself, so it cannot be installed in a lab or secure office setting. The app also requires regular installations of new updates that are unavoidable.
These limitations led CUAHSI to create a new web interface that mimics the features of HydroDesktop at data.cuahsi.org; however, the web interface is limited and is not uniquely extensible like HydroDesktop. For example, it focuses on one aspect of information, namely data measured at fixed-point locations, and research on other classes of data is not fully addressed. It does not allow users to add HydroServers outside the CUAHSI HIS Central system. Although the code is open source, replicating data.cuahsi.org for an outside organization is not a straightforward process; it takes a great deal of skill and expertise to be able to replicate it.

Thus to improve the extensibility and to provide more flexibility to developers and decision makers working with hydrologic data, I developed a new web-application called SERVIR Water Observations Data Integrator (SWODI) in the Tethys web-development framework. This application acts as a portal for data management and discovery for multiple hydrologic datasets. Currently, a user can catalog data from existing HydroServers and access data from NASA Global Land Data Assimilation System (GLDAS), which generates products simulated by land surface models [Fang, 2009], and NASA ClimateServ, a tool that puts three datasets into one system to allow for better evaluation of current water and agricultural situations (https://climateserv.servirglobal.net/). The end goal of this application is to ultimately have a more modular catalog and to be extensible so that newer datasets or models can be added onto the existing capability. It will also address the issue of not being able to catalog HydroServers that are not in the CUAHSI HIS Central system.

The remainder of this paper is organized as follows: section 2 goes into greater detail about the platform that was used to build the app (Tethys); section 3 provides an overview of the software architecture of the app and a user guide for the app itself; and section 4 presents the results of the application and the potential for future development.
2. COMPONENTS OF THE WEB APPLICATION

2.1 Tethys Framework

Tethys [Swain et al., 2015; Swain 2015] is a free and open-source web-development framework developed at Brigham Young University, Provo, UT. Its unique features have significantly lowered the barrier for developing hydrologic web-application. I decided to use the Tethys Platform for developing SWODI because it already comes with several in-built tools that can be used for the app, which will be explained in greater detail in section 3. On the contrary, developing this app from the ground up without the use of Tethys would require that I install all of the app components separately and then compile the website myself. Furthermore, it would have made it harder for other developers to easily replicate the process if I had compiled it all independently. Using the Tethys Platform allowed me to streamline the process for developing the app.

The Tethys Platform architecture can be broken down into three major components: Tethys SDK, Tethys Portal, and Tethys Software Suite. Tethys web apps are developed with the Python programming language and a Software Development Kit (SDK). The SDK provides Python module links to each software component of the Tethys Platform, making the functionality of each component easy to incorporate in your web apps. In addition, users can access all of the Python modules that they are accustomed to using in their scientific Python scripts to power their web apps.
The Tethys Portal is where developed apps can be published; it provides an app library page for the users to be able to access installed apps. It also includes several tools and functionalities to be able to customize features such as user permissions, portal design, etc.

The Software Suite contains several tools and packages, such as GeoServer, PostgreSQL, OpenLayers, etc., that are used commonly when developing geospatial web apps. Figure 2-1 is a visual representation of the core components of the Tethys Platform.

**Figure 2-1: Tethys Platform Software Architecture** (reproduced with permission from the Tethys Platform Documentation - [http://docs.tethysplatform.org](http://docs.tethysplatform.org))

### 2.2 Spatial Database Storage

Tethys Software Suite includes a PostgreSQL database for data storage. PostgreSQL is a powerful, open source object-relational database system. With the help of SQLAlchemy (A
Python SQL Toolkit), it is fairly easy to setup and manage new databases in the Tethys Platform. The SWODI uses one database for cataloging the HydroServers in one central location.

2.3 Map Publishing

Tethys Software Suite has a built-in GeoServer for publishing and storing geospatial datasets as web services (geoserver.org). GeoServer is a Java-based software server that allows users to view and edit geospatial data. Using open standards set forth by the OGC, GeoServer allows for great flexibility in map creation and data sharing. Sites in each HydroServer are published as a layer on the local GeoServer; once they are published, they are accessible as a Web Mapping Service (WMS) Layer. Any WMS Layer published on the GeoServer allows developers to retrieve the layer attributes and can be easily customized and styled on the client’s side. There are many different URLs used to define a WMS, such as GetCapabilities, GetMap, and GetFeatureInfo. The following is an example of a WMS request for catalog:hydroserver_sites layer to be output in Spatial Reference System (SRS) EPSG:4326 map projection and using default styling:

3. METHODS

3.1 Software Organization

The Tethys application structure follows the Model-View-Controller (MVC) software architecture, as shown in Figure 3-1. This allows for simple, readable and reusable code. The model is responsible for initializing the database and managing the database structure. The controllers handle the logic in the web application and connect the database to the front end. Any sort of data retrieval and presentation is done through the controllers. The views represent the HTML pages that are rendered for the user to see. For example, the functionality of adding new HydroServers is provided to the user in a “View” which is comprised of a template file named home.html. The front end code triggers a controller function in controllers.py called “Add Server.” This function retrieves data from the HydroServer and publishes the HydroServer metadata to the “Model.” It simultaneously publishes the data to the GeoServer as a WMS layer. Once these functions are executed the controller returns data to the front end indicating that the process has been executed successfully.
3.1.1 Tethys App Folder Structure

The Tethys App folder structure is set up so that it can be easily deployed and replicated. One component within the folder, the app package, contains the source code for executing the Tethys app. Within the app package, there are several different files that come together to create the app. The app directory structure is also shown in Figure 3-2. The model.py file initializes the database, and the controllers.py file contains the controller functions that are used throughout the app. The utilities.py file contains common functions that are used throughout the app, and the persistentstore.py file is the database itself. The templates directory contains the HTML pages that are rendered to the front end. The public directory contains resources that are responsible for rendering the HTML content, such as JavaScript, Cascading Style Sheets (CSS), and images; it also contains any external libraries. The original Django app structure has several moving parts, and it is not straightforward for novice developers. However, the Tethys project app structure has
the MVC components in one central location, making it easier for first-time web developers to leverage the MVC structure.

Figure 3-2: Tethys App Directory Structure

3.1.2 Model

The model is the database design document for the application. The model is responsible for creating the database that stores the information about the HydroServers. Any HydroServer that is added in the user-interface is stored as a database record in the Catalog database. This allows for basic retrieval and display of all the available HydroServers. The database for storing the HydroServer metadata is simple; it contains only one table with six columns. The database columns are as follows: id, title, url, geoserver_url, layer_name, and extents.

The id column is the primary key; every time there is a new record, the id number increases by one. This is generated automatically by the database to ensure that each record is unique. The title column stores the display name as entered by the user. The URL column stores the SOAP endpoint URL as entered by the admin. The geoserver_url stores the GeoServer endpoint for the application. Since Tethys apps can be installed on multiple Tethys portals, it is necessary to have a function that checks the GeoServer of the Tethys Portal that the app is installed on. The
geoserver_url and the layer name are stored so that they can be added as layers on the OpenLayers map. Finally, the extents of the GeoServer layer are stored so that the layer can be retrieved and added to the OpenLayers map. The format for adding a WMS URL in OpenLayers is as follows:

```javascript
wmsSource = new ol.source.TileWMS({url: geoserver_url,
   params: {'LAYERS':layer_name,
     'SLD_BODY':sld_string},
   serverType: 'geoserver',
   crossOrigin: 'Anonymous'});

wmsLayer = new ol.layer.Tile({
   extent:ol.proj.transformExtent([extents['minx'],extents['miny'],extents['maxx'],extents['maxy']],'EPSG:4326','EPSG:3857'),
   source: wmsSource
});
```

### 3.1.3 View

The views render information from the controllers to the front end for the users to see. The data from the controllers is passed in the form of context variables, meaning variables that are created every time the app is initiated, thus ensuring that the data is dynamic. These variables can be rendered directly through the Django HTML template. The following table gives a description of each HTML page and their respective purposes:
Table 3-1: Description of HTML Pages

<table>
<thead>
<tr>
<th>Page Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>addsite.html</td>
<td>Initial page that allows the admin to add a new HydroServer</td>
</tr>
<tr>
<td>base.html</td>
<td>The base template that contains app styling and other html elements that are constant throughout the app</td>
</tr>
<tr>
<td>cserv.html</td>
<td>Renders the time series plot from ClimateServ</td>
</tr>
<tr>
<td>datarods.html</td>
<td>Displays a time series plot for the GLDAS data</td>
</tr>
<tr>
<td>details.html</td>
<td>View the site details and time series for a selected variable and date range for a clicked point</td>
</tr>
<tr>
<td>error.html</td>
<td>Displays an error message if any requests were not executed properly</td>
</tr>
<tr>
<td>his.html</td>
<td>Shows a list of existing HydroServers in the CUAHSI HIS Catalog</td>
</tr>
<tr>
<td>home.html</td>
<td>The first page that the user sees; contains the HTML code that renders several modals</td>
</tr>
</tbody>
</table>

3.1.4 Controllers

The controllers handle the logic in the application; they are the link between the user interface and the database. Any request submitted by the user through the interface is mapped through app.py and is ultimately executed through controllers.py. The controller returns data through the HTML page or as a JSON object, which can be rendered as an HTML object.

Special controllers are required for different services because each of the services provided in the app follow different standards. The workflow for adding a new service is as simple as creating a new controller and creating the relevant HTML template and JavaScript to interact with that controller. There is no need to rebuild the app each time a new service is added, as it simply becomes an add-on to the existing interface.
3.1.4.1 Home Controller

The primary function of the home controller is to retrieve information from different data sources to set up a few parts of the user interface on the homepage. The home controller parses metadata from the CUAHSI HIS central server to generate a list of available HydroServers. It also makes requests to two NASA Application Programming Interfaces (APIs) to retrieve a list of the available variables, forecast start and end dates. This information is used to dynamically generate dropdowns that the user can then select to view specific information in regards to GLDAS and ClimateServ respectively.

3.1.4.2 Catalog Controller

The catalog controller is responsible for checking the database for any records of existing HydroServers. If there are any existing records, it retrieves that list and updates the catalog on the home page with a link to those HydroServers.

3.1.4.3 Add Server Controller

The add server controller is responsible for parsing and publishing the HydroServer data to the GeoServer. Once the request to add server is initiated, the controller requests data from the HydroServer and parses that data for a list of sites. The sites and their corresponding latitude and longitude values are stored as an array. The information in the array is then used to generate a shapefile of the locations with the help of pyshp python library. After creating the shapefile, the controller then publishes this shapefile to the local instance of GeoServer. Finally, the HydroServer metadata and the shapefile metadata are stored in the Catalog database.

3.1.4.4 Details Controller

The details controller is the most complex and intricate function in the entire application. This controller relies on several other functions and controllers to display the time series data of
a selected variable. The functionality of this controller can be broken down to three steps. First, the controller retrieves all the available variables and their date range for the selected variable. This information is passed on to the HTML template in the form of a dropdown and a date picker. Second, once the user selects a variable and the date range, that information is stored and passed to a controller. Lastly, the third controller retrieves the information from the second controller and parses the HydroServer for the relevant data. The time series data is stored in JSON format and sent to the front end for rendering into a user-viewable chart. These steps are displayed in Figure 3-3.

![Figure 3-3: Details Controller Sequence](image)

3.1.4.5 ClimateServ Controller

The ClimateServ controller is triggered when the user submits the ClimateServ form. This controller creates a dynamic URL that then requests the data through the ClimateServ API. The
data from the server is parsed and rendered through the Tethys TimeSeries gizmo, which is used to create a time series plot visualization. Using this template gizmo allows developers to add features such as plots and maps to their app with little coding. The gizmo essentially converts Python code into HTML and JavaScript elements. Since many of the results are retrieved in Python, using a gizmo reduces the amount of code needed on the front end.

3.1.4.6 Update Controller

The update controller is responsible for deleting a selected HydroServer record from the database. The controller connects to the database and deletes the requested HydroServer. Once the record is deleted, the database is forced to refresh, thus updating the catalog instantaneously. Currently, there is no mechanism to allow users to edit the name or metadata of the HydroServers in the Catalog.

3.1.4.7 Data Rods Controller

The Data Rods controller is triggered when the user selects the point option in the Map Interaction Console and submits the GLDAS form. The information from the form is used to generate a dynamic URL that requests data through the NASA GLDAS API. The data from the API is parsed and displayed through the Tethys TimeSeries gizmo.

3.2 Application User Interface Overview

The SWODI application user interface is designed to be intuitive and extensible. This overview will explain more about the different roles, functionality, and workflow of SWODI. The workflow for a user without admin privileges includes selecting a dataset and retrieving the relevant metadata from that dataset, whereas the admin workflow involves adding and updating the HydroServer catalog. The following sections will describe these workflows in greater detail.
3.2.1 Accessing the Application

The SWODI application is currently deployed on the Tethys Portal at tethys.byu.edu. The source code is available on GitHub at https://github.com/SarvaPulla/servirdi. The app is currently developed to work with Tethys 1.4 version; it is licensed under the MIT licensing convention. Thus any individual or organization that wishes to use it can download the source code and deploy it on to their own portals without any restrictions. The apps on the portal are hosted on the apps page on this website. A user can access the application by clicking on the SWODI app icon on the apps homepage as shown below.

![Figure 3-4: Tethys.byu.edu Apps Library](image)

3.2.2 Viewing Data from HydroServers

A list of the available HydroServers is shown in the catalog in the navigation bar on the left. To view a specific HydroServer, the user clicks on the button to the right of the HydroServer of interest. The user then clicks “Zoom To” to zoom in on the available site of the server.
Figure 3-5: List of Available HydroServers

After zooming to an area of interest, each of the colored dots represents a site with data. Clicking on the dots will reveal a pop-up with the site name and a button to view details of that particular site.

Figure 3-6: HydroServer Site Pop-Up
The user may click on site details to view more information on a specific location. This will then allow for selecting a variable and a date range for the data. Then, the user clicks generate graph to view the time series for that variable and date range.

![Site Details for Majagual](image)

**Figure 3-7: Site Details Page with Time Series for Majagual**

### 3.2.3 Exploring GLDAS Data

The user clicks “Explore Map Console,” and then select “Point” to activate the point interaction on the map.

![Map Console](image)

**Figure 3-8: Selecting a Point Interaction**
The user may zoom to an area of interest and click to add a point on that spot. A modal with options to select a variable and a date range will open with the ability to select specific parameters.

![Data Rods Explorer](image)

**Figure 3-9: GLDAS Data Form**

Once the variable and date range are selected, the user clicks “Get Data” to view the time series.
3.2.4 Exploring Climate Serv Data

The user clicks “Explore Map Console,” and then selects “Polygon” to activate the draw polygon tool.

Figure 3-11: Selecting a Polygon Interaction

The user may zoom to an area of interest and select points to digitize the polygon.
Figure 3-12: Digitized Polygon

A modal will then appear with options to select data and operation types as well as a date interval and range.

![Figure 3-12: Digitized Polygon](image)

Figure 3-13: Climate Serv Console

The user inputs the information and then clicks “Get Data” to view the time series.
Finally, the “Upload Shapefile” functionality follows the same process and allows users to upload a shapefile for an area of interest and view a Climate Serv time series. An example is shown below.

![Nepal Shapefile](image)

**Figure 3-15: Nepal Shapefile**
3.2.5 Managing the Admin Interface

The HydroCatalog Settings Interface is used for adding or deleting servers from the catalog. This interface is only available to those with administrative privileges. To access the interface, the admin clicks “Settings” on the bottom, left corner of the home page. A modal will open with options to either add a server or update the catalog.

![HydroCatalog Settings Interface](image)

**Figure 3-16: HydroCatalog Settings Interface**

To add a server, the user clicks “Add SOAP URL.” From here, users can enter a title and SOAP Endpoint URL for a HydroServer of their choice.

![Add WaterOneFlow Service Info](image)

**Figure 3-17: Adding a HydroServer**

Once the server is successfully added, a message will appear on the screen. The current HydroServers list will then be updated. Users can also choose a server from the CUAHSI HIS server database by clicking “Select HIS Central Server.”
Users with admin privileges can update the catalog to delete servers by clicking “Update HS Catalog.” A modal will appear where the admin can select a HydroServer to remove, and click “Delete.” This will update the current HydroServers list accordingly.

<table>
<thead>
<tr>
<th>Title</th>
<th>URL</th>
</tr>
</thead>
<tbody>
<tr>
<td>DR</td>
<td><a href="http://worldwater.byu.edu/app/index.php/dr/services/cuahsi_1_1.asmx?WSDL">http://worldwater.byu.edu/app/index.php/dr/services/cuahsi_1_1.asmx?WSDL</a></td>
</tr>
<tr>
<td>ProvoGAMUT</td>
<td><a href="http://data.istahepscor.org/ProvoRiver/WOF/cuahsi_1_1.asmx?WSDL">http://data.istahepscor.org/ProvoRiver/WOF/cuahsi_1_1.asmx?WSDL</a></td>
</tr>
<tr>
<td>ICIMODNepal</td>
<td><a href="http://worldwater.byu.edu/app/index.php/icimod/services/cuahsi_1_1.asmx?WSDL">http://worldwater.byu.edu/app/index.php/icimod/services/cuahsi_1_1.asmx?WSDL</a></td>
</tr>
<tr>
<td>DRIHM</td>
<td><a href="http://hydro10.sdsc.edu/MeteoFR/cuahsi_1_1.asmx?WSDL">http://hydro10.sdsc.edu/MeteoFR/cuahsi_1_1.asmx?WSDL</a></td>
</tr>
<tr>
<td>Sediment</td>
<td><a href="http://worldwater.byu.edu/app/index.php/sediment/services/cuahsi_1_1.asmx?WSDL">http://worldwater.byu.edu/app/index.php/sediment/services/cuahsi_1_1.asmx?WSDL</a></td>
</tr>
</tbody>
</table>
4. RESULTS

The SERVIR Water Observations Data Integrator (SWODI) application was primarily built to increase the capacity for managing and accessing hydrologic information at The International Centre for Integrated Mountain Development (ICIMOD) (http://www.icimod.org/). ICIMOD is a regional intergovernmental agency in Nepal that aims to provide earth science research to its stakeholders across eight countries. Prior to the development of SWODI, ICIMOD did not have an integrative platform where they could view all of their hydrologic data along with other relevant NASA datasets.

I built this web application so that ICIMOD can take advantage of the CUAHSI Hydrologic Information System and the SERVIR ClimateServ. By closely mimicking a version of the CUAHSI HydroServer system in a Tethys app, I was able to create a simple app that enables users to view time series observations from a HydroServer directly alongside earth observation data products. This will open up the data sharing capability in ICIMOD and allow time series data to be viewed on international data catalogs. When a regional CUAHSI HydroServer catalog is deployed, it will allow various agencies within the Himalaya region to easily access critical observations data at their fingertips.

ICIMOD works with many stakeholders that have monitoring programs which produce observations of precipitation, streamflow, water surface levels and other time series measurements critical for understanding water systems. As of now, they do not have a standardized way of storing their observations data. Eventually, once the data is standardized and
uploaded to their relevant HydroServers, these HydroServers can then be cataloged in SWODI for convenience. Furthermore, since HIS supports storing data points using the Observations Data Model (ODM), it makes it possible to access ICIMOD’s HydroServers’ data as a WaterML standard web service for anyone to use.

4.1 Limitations

As of now, the biggest limitation is keeping up to date with Tethys and other changing standards. For instance, WaterML is now at 2.0 as opposed to 1.1 as it is currently configured for the app. Since the launching of the app, Tethys also underwent a major upgrade, thus requiring major reconfigurations within the app in order for it to be compliant with Tethys 2.0. Also, the app relies on several external components causing the developer to constantly watch to make sure that those APIs are working properly. Unfortunately, this cannot be avoided, as the external components are a necessary part of the application, making it flexible and extensible.

4.2 Future Developments

There are a few potential developments that could be added to the app in the future to further increase the app’s functionality. First, a feature could be added to compare time series details between different sites. This will allow users to compare a variable at different sites for a set time period. Second, the app needs to be reconfigured for Tethys 2.0, which will allow users to be up to date with the current version of Tethys. Third, an API could be added within the app to query the list of available HydroServers and their respective sites and time series.
5. CONCLUSION

The SWODI provides a simple way to quickly access data from three different data sources, namely a CUAHSI HIS HydroServer, NASA GLDAS, and NASA ClimateServ. Unlike any other applications out there, this app catalogs the HydroServers in one central location. This improves the ease of accessibility of the data and allows for the standardization of that data.

Any organization with hydrological information can set up a HydroServer with ease and catalog the HydroServer through the provided framework. Traditionally, researchers have been required to register their HydroServers with CUAHSI for it to show up within the CUAHSI catalog. This app now circumvents the need for users to register with CUAHSI to be able to view the HydroServer data or make it discoverable in an interface.

Since the app was developed in the Tethys framework, it makes it extensible and allows developers to add extra datasets with ease. It is open-source and thus can be easily replicated and modified as desired. Ultimately, this improved data accessibility will be a great addition to any institution’s data management needs.
REFERENCES


Stewart, Bruce (2015). Measuring what we manage-the importance of hydrological data to water resources management. Proceedings of the International Association of Hydrological Sciences, 366, 80.


