2016-03-01

Design, Development and Testing of Web Services for Multi-Sensor Snow Cover Mapping

Jiri Kadlec
Brigham Young University - Provo

Design, Development and Testing of Web Services for Multi-Sensor Snow Cover Mapping

Jiri Kadlec

A dissertation submitted to the faculty of
Brigham Young University
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

Daniel P. Ames, Chair
Norman L. Jones
Woodruff A. Miller
E. James Nelson
Gustavious P. Williams

Department of Civil and Environmental Engineering

Brigham Young University

March 2016

ABSTRACT

Design, Development and Testing of Web Services
for Multi-Sensor Snow Cover Mapping

Jiri Kadlec
Department of Civil and Environmental Engineering, BYU
Doctor of Philosophy

This dissertation presents the design, development and validation of new data integration methods for mapping the extent of snow cover based on open access ground station measurements, remote sensing images, volunteer observer snow reports, and cross country ski track recordings from location-enabled mobile devices.

The first step of the data integration procedure includes data discovery, data retrieval, and data quality control of snow observations at ground stations. The WaterML R package developed in this work enables hydrologists to retrieve and analyze data from multiple organizations that are listed in the Consortium of Universities for the Advancement of Hydrologic Sciences Inc (CUAHSI) Water Data Center catalog directly within the R statistical software environment. Using the WaterML R package is demonstrated by running an energy balance snowpack model in R with data inputs from CUAHSI, and by automating uploads of real time sensor observations to CUAHSI HydroServer.

The second step of the procedure requires efficient access to multi-temporal remote sensing snow images. The Snow Inspector web application developed in this research enables the users to retrieve a time series of fractional snow cover from the Moderate Resolution Imaging Spectroradiometer (MODIS) for any point on Earth. The time series retrieval method is based on automated data extraction from tile images provided by a Web Map Tile Service (WMTS). The average required time for retrieving 100 days of data using this technique is 5.4 seconds, which is significantly faster than other methods that require the download of large satellite image files. The presented data extraction technique and space-time visualization user interface can be used as a model for working with other multi-temporal hydrologic or climate data WMTS services.

The third, final step of the data integration procedure is generating continuous daily snow cover maps. A custom inverse distance weighting method has been developed to combine volunteer snow reports, cross-country ski track reports and station measurements to fill cloud gaps in the MODIS snow cover product. The method is demonstrated by producing a continuous daily time step snow presence probability map dataset for the Czech Republic region. The ability of the presented methodology to reconstruct MODIS snow cover under cloud is validated by simulating cloud cover datasets and comparing estimated snow cover to actual MODIS snow cover. The percent correctly classified indicator showed accuracy between 80 and 90% using this method. Using crowdsourcing data (volunteer snow reports and ski tracks) improves the map accuracy by 0.7 – 1.2 %. The output snow probability map data sets are published online using web applications and web services.

Keywords: crowdsourcing, image analysis, interpolation, MODIS, R statistical software, snow cover, snowpack probability, Tethys platform, time series, WaterML, web services, winter sports

ACKNOWLEDGEMENTS

I would like to acknowledge the financial support of EarthCube grant EAR1343785 and of the Consortium of Universities for Advancement of Hydrologic Sciences, Inc. (CUAHSI) who provided funding for my research. I am thankful to Wood Miller, Jim Nelson, Norm Jones, and Gus Williams for their guidance as members of my advisory committee at the Brigham Young University (BYU), and to Daniel P. Ames who worked hard as my advisor both at the Idaho State University (ISU) and at BYU and edited the whole dissertation manuscript.

I am thankful to my colleagues at ISU, Aalto University and BYU who worked together with me: Especially Tevaganthan Veluppillai, Yang Cao, Rohit Khattar, Matthew Klein, Dinesh Grover, Carlos Osorio, Rex Burch, Tiffany White, Nora Silanpää, Pasi Haverinen, Raila Lindström, Tuija Laakso, Jeff Sadler, Dustin Woodbury, Sarva Pulla, Matthew Bayles, Xiaohui Qiao, Stephen Duncan, and Bryce Anderson, Nathan Swain and Scott Christensen. I would also like to thank Ari Jolma for his support and mentoring during my work experience in Finland, Pavel Treml for introducing me to the R programming language, and Anas Altartouri for teaching me key R concepts. I appreciate my parents for their patience during my writing process. Big thanks belong to Sahar Mokhtari for her powerful encouragement and motivation.

The following acknowledgments relate to the specific chapters of this dissertation: For Chapter 2, I gratefully acknowledge the assistance of the Bureau of Land Management Salt Lake City Field Office who conducted NEPA and carried out the experimental burn treatments. The experimental project described in Chapter 2 was funded by USDA NIFA grant: 2010-38415-21908 and the wireless sensor network was generously provided by Decagon Devices, Inc. Additional funding was provided by the Charles Redd Center for Western Studies. The World Water Project at BYU was funded by a Mentored Environment Grant from the BYU Office of

Research and Creative Activities. Bryn St Clair designed the HydroServer Lite database schema and website in Chapter 2 and contributed key ideas for the WaterML R package. Richard A. Gill provided the detailed description of the wireless sensor installation. Tevaganthan Veluppillai helped with testing and development of the HydroServer Lite data uploading API. Two anonymous reviewers provided additional suggestions to improve the manuscript. Bryn StClair, Daniel P. Ames, and Richard Gill edited and reviewed the whole text of Chapter 2.

The research presented in Chapter 3 was funded by The National Science Foundation under the EarthCube grant EAR1343785. Woodruff A. Miller conducted the field validation survey in the Yellowstone National Park. Zhiyu Li helped with setting up the Tethys server. Nathan Swain helped with fixing bugs and adding extra features to the system. Shawn Crawley also contributed to the source code by adding extra user interface features. Dustin A. Woodbury, Mary Peterson and Julia Larsen helped with English language editing of Chapter 3 Three anonymous reviewers provided helpful comments to improve the Chapter 3 manuscript. Woodruff A. Miller and Daniel P. Ames edited and reviewed the whole text of Chapter 3.

The research presented in Chapter 4 was funded by the National Science Foundation funded this research under the EarthCube grant EAR1343785. Daniel P. Ames provided key ideas for the interpolation and validation methods, and edited the whole text of Chapter 4. Zhiyu Li developed the HydroShare geographic raster resource and the HydroShare raster viewer application. Scott Christensen and Nathan Swain helped with setting up the Tethys web processing service. The HydroShare team provided free disk space to store the snow probability map data. I also acknowledge the kind assistance of InMeteo Ltd, Strava Inc., and Garmin Inc. for permission to use their volunteer snow report and cross country ski track data in this research.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1 INTRODUCTION

Snowpack data are important in climatology (Groisman et al. 2006; Karl et al. 1993), hydrology (Barnett et al. 2005), and recreation (Braunisch et al. 2011; Ghaderi et al. 2014). Snowpack is a spatio-temporal field that changes in place and time primarily as a result of the interaction of several variables including: air temperature and precipitation (Henderson and Leathers 2010), topography (Lapena and Martz 1996), and vegetation cover (Veatch et al. 2009). Global, regional and local databases of snow depth, snow water equivalent and spatial extent of snow covered area exist in the form of ground measurements (Pohl et al. 2014), remote sensing images (Rees 2005), and models (Koivusalo et al. 2001). Many studies highlight the importance of open standards for sharing spatial and temporal climate data (Bai et al. 2012; Bambacus et al. 2008) and hydrological data (Salas et al. 2012). However, outside North America, much of the snow data is not yet easily accessible to the public (Henderson and Leathers 2010) or the access is very restricted (Nativi et al. 2014; Triebnig et al. 2011). The available snow datasets with global coverage are relatively low detail (Tedesco et al. 2015), incomplete (Callaghan et al. 2011), in many heterogeneous formats (Karl et al. 1993), coordinate systems, and time resolutions. Some of the remote sensing-based snow datasets require specialized expert tools to find (Delucchi 2014) and visualize (Blower et al. 2013). This dissertation explores several techniques to improve the global availability of high quality snow cover data. Specifically of interest are ground station data, remote sensing data, and crowdsourced data.

Crowdsourcing is a process of taking a task traditionally performed by a designated agent, and outsourcing it to an undefined, large group of people (Howe 2008) often using online technologies (Schenk and Guittard 2011). Mobile phone users send thousands of reports, photographs, messages and other information about snow conditions on social networks and community web sites (Wang et al. 2013; Yang et al. 2012). When linked to geographic location, these snow reports become part of volunteered geographic information (VGI) (Goodchild 2007). There are multiple open research issues in using VGI in snow cover or other environmental mapping: The volunteer-contributed data are not a representative random sample (Havlik et al. 2013), have a fuzzy character (Amintoosi et al. 2015) and high uncertainty (Yang et al. 2012). In spite of the data quality issues, collaborative projects like OpenStreetMap (Haklay 2010) and OpenWeatherMap (Ramos et al. 2014) have shown that participatory mapping is a successful method for making environmental data available online (D'Hondt et al. 2013).

Data integration is defined by Lenzerini (2002) as "the problem of combining data residing at different sources, and providing the user with a unified view of these data". In meteorology and hydrology, a key element of data integration is data assimilation. Data assimilation is the process by which observations of the actual system are incorporated into the model state of a numerical model of that system (Houser et al. 2012; Walker and Houser 2005). In spite of growing volume of public VGI data, only a few studies have tested data integration of VGI for snow data assimilation and snow mapping. Examples of using crowdsourcing for snow cover mapping are documented by Wang et al. (2013) using photograph sharing networks, Muller (2013) using the Twitter network, and Muller et al. (2015) using multiple social networks. These studies indicate that VGI from social networks represents a potentially highly informative dataset for updating continuous snow cover maps.

This dissertation builds up on the work of Salas (2012) who proposes a distributed, interoperable network of web services for sharing and integrating hydrological and climate data from multiple sensors. The research contributes towards bridging the "digital divide" that exists between the grid-based remote sensing datasets, and the point-based time series observation networks. The aim to lower the learning curve that is required to view, access, analyze and re-use snow maps and snow time-series from multiple sensors and sources. As a test platform for testing the data retrieval and integration procedures, I have selected the R statistical computation environment (R Core Team 2015). I have selected the R statistical software environment because it is multi-platform, open source, and widely used by hydrologists and environmental scientists studying the snowpack. The first hypothesis driving this work is that using interoperable web services will reduce the time and effort required to access snow data from ground stations, remote sensing platforms, and social networks in the R environment.

This dissertation also builds up on the work of Muller (2013) and explores the potential of combining open data from ground stations, remote sensing datasets, and user-contributed reports from social networks to create a new snow probability map dataset. The dataset is published as a free, online snow cover web mapping service. The second main hypothesis driving this work is that the integration of user contributed data and/or social-network derived snow data together with other open access data sources results in more accurate and higher resolution – and hence more useful snow cover maps than government agency produced data by itself.

The remainder of this dissertation is structured as follows: Chapter 2, entitled "WaterML R Package for Managing Observation Data through CUAHSI Web Services" addresses the issue of connecting scripting environments with online sensor data management systems. Through a new WaterML R package (Kadlec et al. 2015), it enables access to point-based snow measurements

from the global Consortium of Universities for Advancement of Hydrologic Science (CUAHSI) Hydrologic Information System (HIS) from inside R statistical software. The R statistical software (R Development Core Team 2015) then can be used for spatial analysis and time series analysis of the point observations, and for comparing the observations with other sources.

Chapter 3, entitled "Extracting Snow Cover Time Series Data from Open Access Web Mapping Tile Services" presents the design, development, and testing of a new open source script and web application for snow cover probability time series extraction from map images. The script is deployed as a web app using the Tethys framework (Jones et al. 2014) making it accessible to novice users through a user interface. A WaterML web-API gives access to third party applications for automation and embedding in modeling tools. The full design of the script is presented such that it can serve as a model for similar or extended tools that may be developed by others. A set of use case experiments is presented demonstrating the full functionality of the script and its limitations, and an example application for ground validation of the MODIS snow cover dataset is discussed.

Chapter 4, entitled "Using Crowdsourced and Station Data to Fill Cloud Gaps in MODIS Snow Datasets" builds up on the previous two chapters by including crowdsourcing data in the snow mapping solution. Using a custom inverse distance weighting method, measurements from meteorological stations, volunteer snow reports, and cross-country ski track reports are combined to fill cloud gaps in remotely sensed snow cover products based on Moderate Resolution Imaging Spectroradiometer (MODIS) satellite data. The method is demonstrated by producing a continuous daily time step snow probability map dataset for the Czech Republic region. For validation, the ability of the method to reconstruct MODIS snow cover under cloud by simulating cloud cover datasets and comparing estimated snow cover to actual MODIS snow

cover is tested. The output data sets are available on the HydroShare website (Horsburgh et al. 2015) for download and through a web map service API for re-use in third-party interactive map applications.

1 WATERML R PACKAGE FOR MANAGING OBSERVATION DATA THROUGH CUAHSI WEB SERVICES

Essential to the successful execution of any data intensive research study is the management of data and data collection in a formal, organized and reproducible manner. Ecological data collection and storage has evolved in many instances into a large, complex task that demands automation for accuracy in acquiring, managing, analyzing, and long-term verification of data by the researchers themselves (Michener and Jones 2012). Small independent ecology lab groups, and scientists who lead those labs focus not only on their own unique scientific questions and procedures, but now must learn data processing techniques and database development so that their work is not thwarted long term by poor storage or access (Conner et al. 2013). Those who are performing an experiment and those who hope to understand the data coming from the experiment may not have the financial means necessary to develop their own data management system. Shared data hosting websites using international standards (WaterML) and open-source software solutions for data management (HydroServer for Windows or HydroServer Lite for Linux), archiving (ODM database), and publication (WaterOneFlow web service) can be an effective way for independent researchers to remain competitive in a future world of data deluge in scientific research. The Consortium of Universities for the Advancement of Hydrologic Science (CUAHSI) provides support for scientists and independent ecology lab groups and helps them to manage and organize their experimental data using open-source technology.

A particular problem faced by ecological researchers is integrating a data management system with a computational analysis environment such as Matlab, Stata, or R. A common feature of these computational analysis environments is that they provide capabilities for exploratory analysis (plots, graphs), and statistical inference (hypothesis testing). Typically data analysis steps are all recorded in a script, making the steps reproducible (Gentleman and Lang 2007). A system that links computational analysis software with standards-based cloud data management would allow researchers to automate the retrieval of raw sensor data or previously processed data directly from the data management system into their analytical environment. The system would also allow researchers to post data and analysis results back to the system for both archival and sharing purposes.

A number of existing tools have been constructed that meet various parts of the overarching goals stated above. For example, two R packages for retrieving water quantity and quality data from USGS National Water Information System (NWIS) have recently been introduced in the R Comprehensive R Archive Network (CRAN) package repository including the "dataRetrieval" and "waterData" packages (De Cicco and Hirsch 2013; Ryberg and Vecchia 2012). These R packages provide useful data download functions that could support ecological research in terms of the U.S. national water information systems, however they are not intended for data upload or for managing data associated with laboratory research.

For laboratory research, it is possible to use, database drivers that link the R statistical software package to major relational database platforms. The "RObsDat" package (Reusser 2014) is one such driver that is specifically designed for connecting to any environmental observations database compliant with the Observations Data Model (ODM) schema using the Structured Query Language (SQL) mechanism. Other more general-purpose examples of

packages that link R to a relational database using SQL are “RMySQL” (James and DebRoy 2012) and “RSQLite” (James and Falcon 2011).

The drivers noted above require a direct SQL connection to an associated ODM database using an IP address and port number. The problem with a direct SQL connection using IP address and port number is that institutional firewalls block the necessary ports in most cases, making the connection only possible inside of the institution’s local network. In the common case of multi-institutional collaborations, such firewalls can restrict direct database access – hence another approach is required. Also, not all HydroServer instances use the ODM database schema. A solution to these problems is to abstract the physical database by only exposing a layer of web services (also called web application programming interface or web API). The web API usually uses the Hyper Text Transfer Protocol (HTTP) to pass information between a client tool and a database using JavaScript Object Notation (JSON) or Extensible Markup Language (XML) encoded text. Such a web service solves the firewall issue and allows access to the data across institutions, though compared to the expressive power of SQL, the web service typically only enables a limited set of pre-defined queries. If well defined (i.e. as in the case of the CUAHSI HIS web services), this limited subset of queries can readily satisfy the requirements of most database management use cases.

Within the environmental sciences, the most widely-used standards for communicating with a database using web services include Sensor Observation Service (SOS), and WaterOneFlow web service (Ames et al. 2012; Tarboton et al. 2009; Valentine et al. 2007). SOS has received widespread adoption in ocean and marine sciences. In hydrology, the WaterOneFlow service is widely used, with around 100 public database servers registered worldwide at http://HISCentral.cuahsi.org/ including ecological research labs (Conner et al.

2013; Whitenack 2010). The "sos4r" package (Nüst et al. 2011) facilitates connecting to the SOS web service from R. Another tool, HydroR, is a plugin for the open source HydroDesktop software that can analyze data retrieved via WaterML (Horsburgh and Reeder 2014). This tool requires installation of a separate software, HydroDesktop (Ames et al. 2012) to perform the search, discovery, and download of data before it can be analyzed in R. HydroDesktop and HydroR require the Windows operating system, which can be a disadvantage for users of other operating systems. No software tools presently exist to push analytical result from R directly to the CUAHSI HIS via web services. The "RObsDat" package overcomes several of these challenges since it is cross-platform and can both read and write data in an ODM formatted database using SQL database connection. The key limitation of the "RObsDat" package is that it is not suitable for situations where multi-institution access to a single HydroServer is required and where institutional firewalls block direct connections to SQL databases.

It is useful to note that out of the 98 data management systems registered on the HIS Central catalogue, none provides a public open direct back-end connection to the entire database but all provide a method to query their database through the WaterML web services API. In short, although WaterML is a widely used international standard, there is currently no easy method of accessing the WaterML web service from the R environment.

The remainder of this chapter presents the design, development, and testing of a new WaterML R package that addresses these problems by supporting download of data directly from any HydroServer instance and upload of data to a special version of HydroServer Lite using R and a web service interface. The data values from multiple sites or variables are retrieved as an R "data frame" that can be directly used in R. Because it is integrated directly into the R statistical software, our package can be installed and used on any operating system with an internet

connection. To test the usability of the package, three case studies are presented: (1) uploading observations from wireless sensors to HydroServer Lite, (2) exploratory and statistical analysis of continuous observation data from a large scale ecological manipulation experiment, and (3) Snowpack model setup and validation using online data from the CUAHSI HIS WaterOneFlow web services.

The work presented here is a significant extension of previous work that presented the PHP-based HydroServer Lite software tool as a system for managing ecological experiment data (Conner et al. 2013). The first issue addressed by this work is simplifying access to HydroServer data through the WaterOneFlow web service from within the R environment. Using the WaterML R package, any HydroServer that implements the WaterOneFlow web service can be accessed from R. The second issue addressed by this work is enabling the upload of analytical results from R to the HydroServer Lite through a web service application program interface (API). While the data editing capabilities currently only work with instances of the HydroServer Lite software, the work presented here serves as a model for future efforts for creating connections between analytical software tools and distributed, web services based data management systems. Also, the HydroServer Lite data upload capabilities are expected to be added to the general HydroServer software stack in the near future, which will immediately enable use of this WaterML R package for both data download and upload on any CUAHSI HydroServer.

1.1 Material and Methods

The following section describes the changes made in the HydroServer Lite online database to facilitate management of ecological experiment data from wireless field sensors. The design,

development and testing of a new R package for downloading observations from CUAHSI Web services is also presented. Finally, this section introduces three use cases for testing the newly designed software.

1.1.1 Software Design and Development

In my design I chose an open source solution using the MySQL relational database and HydroServer Lite software. The HydroServer Lite software installation package and source code are available on the website: http://hydroserverlite.codeplex.com. It can be installed and hosted on any web server or shared webhosting account that supports PHP (version 5.4 or higher) and MySQL. I hosted the database and web server on the shared webhosting site http://worldwater.byu.edu, which provides web space and database space for any independent research group to publish their open access data. A centralized web-based database such as this has the benefit of allowing ecologists to both store data and easily share it with coworkers on the project. As a primary method for statistical analysis of experimental results I chose the R computational environment because it is also open source, cross platform, and widely used in ecological research. In R, every step of data processing is stored as a script, making it possible for coworkers to reproduce the analysis. Its script-based nature also makes R a good candidate for building an integrated database and analysis system since the database web-services will also need to be used through scripted code.

For the centralized, web-based database and data model the Observations Data Model (ODM) (Horsburgh et al. 2008) was selected because it is designed specifically for observational data, it is well-integrated with the HydroServer software to make the data accessible on the Internet by multiple users, and it is well-documented.. Previous studies (Izurieta et al. 2010;

Mason et al. 2014) demonstrated that the ODM can be successfully used to represent an ecological site. Besides storing data observations, the ODM organizes metadata information about the units, geographic locations, measurement methods, variables, time, and sources. The complete ODM database with all its tables was used. Figure 2-1 shows the ODM database schema and highlights specific tables that were important for the experiment.

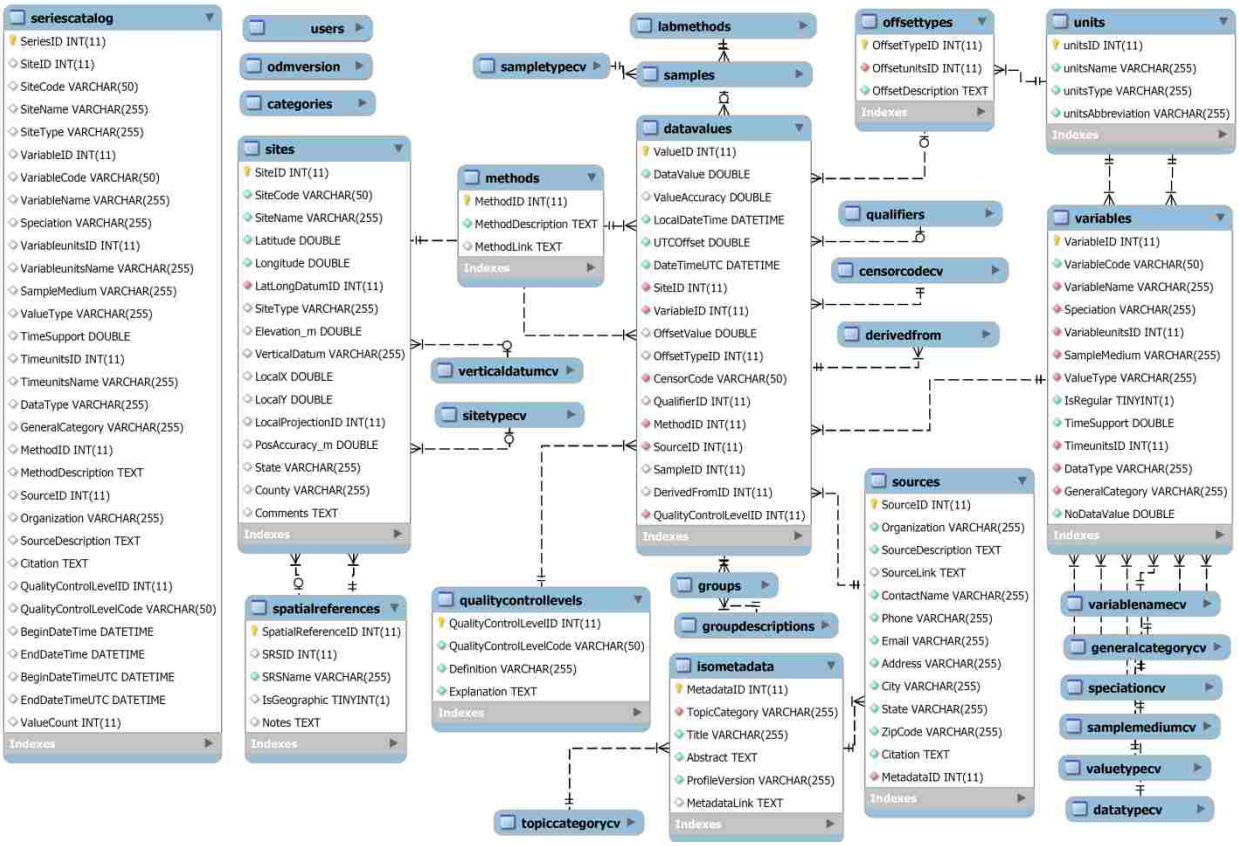

Figure 2-1: Key Tables of the ODM Data Model Used in the Experiment with Field Details Listed for the Tables that were Important for the Study.

1.1.2 HydroServer

HydroServer is an open-source software stack developed by CUAHSI that includes both a Windows-server (Horsburgh et al. 2009) implemetation and an open source Linux-server implementation named HydroServer Lite (Conner et al. 2013; Kadlec and Ames 2012). The

functionality, complete source code and installation files of HydroServer Lite have been described by Conner et al. (2013) and published on the website http://hydroserverlite.codeplex.com.

All versions of HydroServer come with a back end WaterOneFlow web services API that allows other websites and software to query the data and get results in the WaterML format. WaterML is an international Open GIS Consortium (OGC) standard for publishing hydrologic observations. There are three widely used versions of WaterML: WaterML 1.0, WaterML 1.1 (Valentine et al. 2007) and WaterML 2.0. (OGC 2012a). All WaterOneFlow web services implement WaterML 1.0 or 1.1. WaterOneFlow web services can be registered with the HIS Central Catalog (http://hiscentral.cuahsi.org). Once registered in the catalog, the data from the experimental sites becomes available for searching and query by other researchers on the catalog website (data.cuahsi.org). Other software that can be used for visualizing data from the WaterOneFlow web services include HydroDesktop (Ames et al. 2012) and HydroExcel (Whiteaker et al. 2009).

For the purpose of this experiment, I extended the original HydroServer Lite software (Conner et al. 2013) by adding a web service API for data uploading. I included these changes in the open-source release and code of HydroServer Lite version 3.0. The changes are documented and available on the website (http://hydroserverlite.codeplex.com). I have hosted the HydroServer instance on a cloud computing web space with other HydroServers at the URL (http://worldwater.byu.edu/app/index.php/rushvalley).

1.1.3 HydroServer Data Upload API

To facilitate data upload from the WaterML R package into HydroServer, I extended the existing HydroServer Lite codebase by creating a new JSON data upload web API that supplements the existing web-form based data entry tools in HydroServer Lite. If the HydroServer Lite server hosts multiple ODM databases, there is a separate upload service for every ODM database. I implemented the JSON API using the open-source CodeIgniter Model-View-Controller (MVC) PHP web framework. For each table in the ODM database, the MVC framework defines a model class with query functions. For example the Sites table has a Sites model with functions GetAllSites, GetSitesByBox, AddSite, EditSite, and DeleteSite. The API controller defines the new data upload API for HydroServer works as follows: The client sends a hypertext transfer protocol (HTTP) POST request with the user name, password, site, variable, method, and an array of (time, value) pairs in the JSON format in the POST request body. The HydroServer first checks if the user is authenticated by verifying the user name and password. A limitation of the API is that it requires sending the user name and password in plain text over an HTTP POST request. To keep the user credentials secure during transport over the network, I recommend securing the HydroServer Lite data upload by the HTTPS protocol where the user name and password are encrypted. If the user name and password belong to an authorized user, then it checks if the posted data is in the JSON format and looks for the valid site code, valid variable code, valid method id and valid source id. For example, a site code is valid if the site with the code already exists in the database. The combination of site, variable, method source and quality control level is known as a "Time Series" in the ODM. The Time Series information in the SeriesCatalog table also stores the begin time, end time and value count of the series. Finally, the data values are checked. By default, only data values whose site, variable, method,

source, quality control level or time are different than in the existing time series, are inserted to the DataValues table. This check prevents the insertion of duplicate data values. I have provided examples how to access the API from different programing languages: Python, C# and R. The examples are available on the website http://worldwater.byu.edu/app/index.php/rushvalley/services/api/ and they are also documented on http://hydroserverlite.codeplex.com. The JSON API has been added to HydroServer Lite version 3. The installation package and the complete source code of HydroServer Lite has been published as open-source on the website (http://hydroserverlite.codeplex.com) under the New BSD License which allows for liberal reuse in both commercial and non-commercial settings.

1.1.4 WaterML R Package

I designed a new WaterML R package with three goals: (1) to enable data discovery in R by connecting to the CUAHSI HIS Central search catalog, (2) to simplify the connection to any WaterOneFlow web service on any HydroServer through the R statistical software interface and (3) to automate the uploading of data from R to HydroServer Lite through the new HydroServer Lite JSON data upload API. The first functional requirement is supported by the functions `GetServices`, `HISCentral_GetSites`, and `HISCentral_GetSeriesCatalog`. These functions allow discovering public HydroServers, sites, and time series in the user's study area. The search can be refined by specifying a geographic bounding box, keywords, and time range.

The second functional requirement of the WaterML R package was to support connecting to five web methods defined in the WaterOneFlow web service protocol that are important for retrieving data from the HydroServer: `GetSites`, `GetVariables`, `GetSiteInfo` and

`GetValues` by including an R function that corresponded to each WaterOneFlow web method (Table 2-1).

Table 2-1 Functions of the WaterML R Package and Their Parameters

Function Name	*Parameters*	*Usage*
GetSites	Server	Get a table of all sites from the HydroServer
GetSiteInfo	Server, site code	Get a table of all variables, methods and quality control levels at a specific site
GetVariables	Server	Get a table of all variables from the HydroServer
GetValues	Server, site code, variable code, begin time, end time, method (optional), quality control level (optional), source (optional)	Given a site, variable and time range, get a table of the time series of data values.
AddSites	Server, table of sites, user credentials	Add new sites to the HydroServer
AddVariables	Server, user credentials, table of variables	Add new variables to the HydroServer
AddValues	Server, user credentials, table of times and values, site code, variable code, method code, source code	Add new data values to the HydroServer
AddMethod	Server, user credentials, table of methods	Add new methods to the HydroServer
AddSource	Server, user credentials, table of sources	Add new sources to the HydroServer
GetServices	No parameters	Find the URL's of all public HydroServers
HISCentral_GetSites	North latitude, west longitude, south latitude, east longitude	Find the sites in a given geographic region from all registered HydroServers
HISCentral_GetSeriesCatalog	North latitude, west longitude, south latitude, east longitude, keyword (optional), start time (optional), end time (optional)	Find a list of sites, variables and time-series in a given geographic region, time period and keyword

Some WaterOneFlow web service methods such as GetSiteInfoMultpleObject, GetVariableInfo and GetValuesForASiteObject, were not supported through new R package methods, but rather were supported adding optional parameters to the four core methods. Each of the data retrieval methods creates a table in a data.frame format. The data.frame structure can be used as input for data analysis functions in R. For example, if researchers need to compare the mean soil water potential at multiple sites, they would first call the `GetVariables` function to

get the table of all available variables. From this table, they would find details about the variable that represents "soil water potential". Next, they would call the `GetSites` function to get a table of sites. To find out which sites measure the selected "soil water potential" variable, the `GetSiteInfo` function can be used. Next, they would call the `GetValues` function for the selected time range, selected variable, and selected sites. The return value of the `GetValues` function is a data.frame object, which can be used as input for various statistical analysis operations in R. For example, the data values in the data frame can be grouped by different attributes, and the differences between group means can be analyzed using the one-way ANOVA method.

The third functional requirement of the WaterML R package was to support the uploading of data from R to HydroServer Lite through the new HydroServer Lite JSON data upload web service API. To facilitate this requirement, I included five functions in the package: `AddSites`, `AddVariables`, `AddMethods`, `AddSources`, and `AddValues`. These functions have the following inputs: The server URL, the HydroServer user name, the HydroServer user password, and a data frame with the data to be uploaded. The documentation specifies the required and optional columns in the data frame for each function. For example the `AddMethods` function requires the mandatory "MethodDescription" column and the optional "MethodLink" and "MethodID" columns. Additionally, the `AddValues` function requires a data frame with "Time" and "DataValue" mandatory columns, and extra input parameters that identify the site, variable, method, source and quality control level associated with the time series data in the data frame.

The WaterML R package requires the packages XML, RCurl and httr. It can be installed by any user of R in the "install packages" menu. The package has been accepted for publication

on CRAN, the R online package repository (https://cran.r-project.org/web/packages/WaterML/). When a package is on CRAN, it can be installed by any R user from inside the R environment. The WaterML R package is open source under the MIT license. I maintain the source code on the github repository (http://github.com/jirikadlec2/waterml). The source code can be readily retrieved by any interested individuals.

1.1.5 Great Basin Experimental Case Study

In 2011, a factorial experiment was installed in Rush Valley, Utah (40.1 N, -112.3 W). The experiment tests the interactions between fire history, small mammal reductions, and changes in precipitation amount in the Great Basin Desert. Half of the research plots were burned in September 2011. Fences were constructed that restrict the movement of small mammals, and on half of the plots small mammals were removed, creating a factorial burn X small mammal treatment combination that was replicated 5 times. In addition, 2-m X 2.5-m passive rainout shelters were constructed that created 30% reduction, 30% addition, and control precipitation treatments. Of particular interest was the establishment of invasive species in the post-fire treatments. In summer 2014 the Brigham Young University Gill Lab team (Richard Gill, Bryn StClair) installed a wireless sensor network that measures soil water potential, soil water content, soil and air temperature, precipitation, wind speed and direction, solar radiation, and canopy greenness using normalized difference vegetation index (NDVI)(Decagon, Inc., Pullman, WA). The case study experimental site is equipped with 20 EM50g data loggers (Decagon, Inc., Pullman, WA). Each data logger has 5 ports used for receiving data via the SDI-12 connector from the sensors in the form of a time stamp and a 32-bit integer that contains up to 3 responses. All the transmitted data must be acquired, organized, and stored in a relational database that is accessible to numerous scientists on the project. The long term goal is to

monitor and use the data in the database to statistically validate hypotheses from the project and to store and publish the data, and make it accessible to those who want to reproduce my hypothesis tests.

1.1.6 Data Transmission and Storage

Once connected with the mobile network, the EM50g sends its data to the Decagon's internet server twice daily. On the Decagon's internet server, the most recent data from each logger can be queried using the Decagon EC2Data API. To connect to the API, a user specifies the user name, user password, device ID, device password, and time. The response is a .dxd file which contains the metadata about the data logger, ports and sensors in the <metadata> section and the measured raw data values in the <data> section. Connection of the data loggers to the HydroServer via the Decagon API and the HydroServer Lite JSON API is shown in Figure 2-2.

1.1.7 Data Analysis

Once the data has been acquired and uploaded to the ODM database using the WaterML R package and the HydroServer Lite JSON data upload API, the goal was to analyze the data collected so as to test the hypothesis that soil water potential controls the productivity of invasive annual plants in Great Basin ecosystems after a fire. NDVI can be used as a proxy for productivity in this invaded ecosystem. In the experiment, the investigators began to look at the site and compare the treatments imposed on the study. Having the data grouped by treatment, allowed to test the hypothesis and learn about the interactions between soil processes and plant productivity. The process started with exploratory data analysis, followed by statistical inference. For exploratory data analysis, there was a need to develop a way to graphically display the data that are being stored in the database.

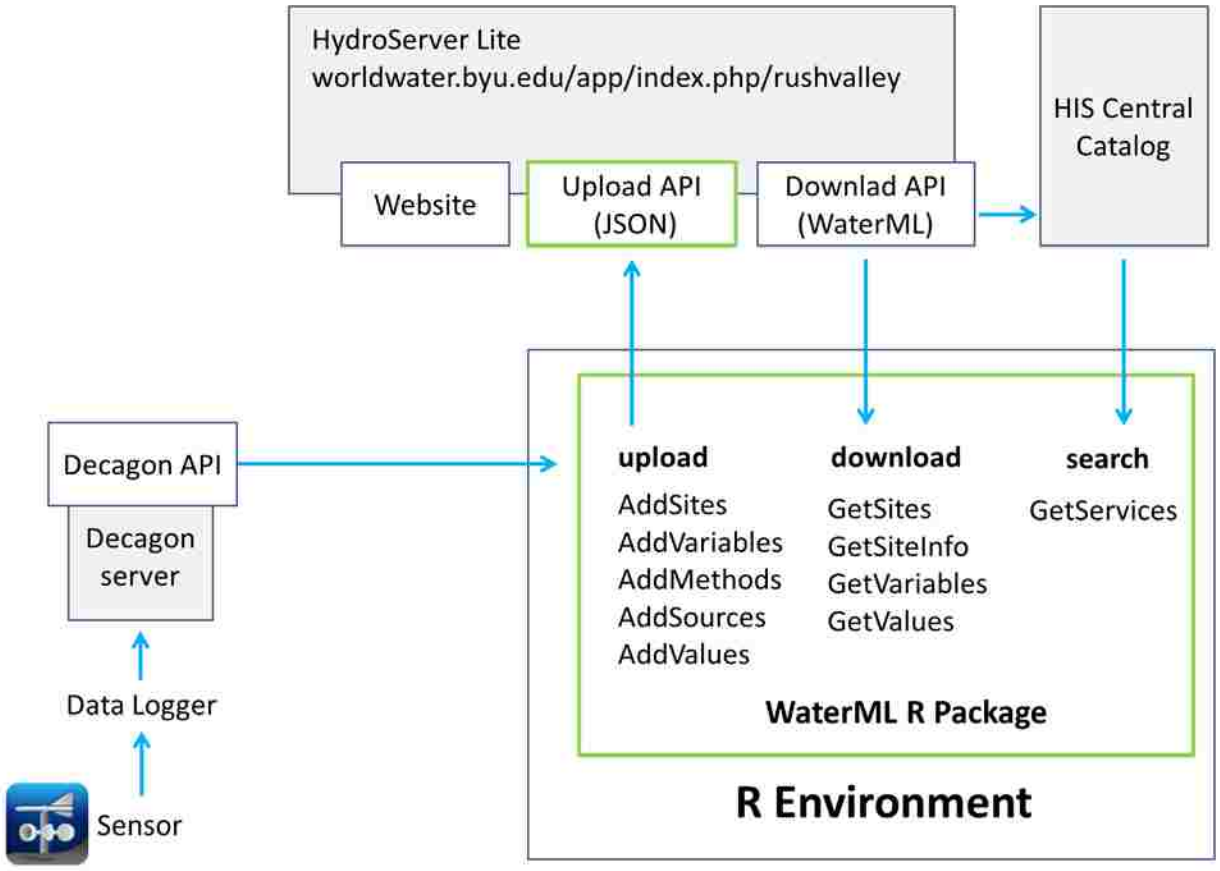

Figure 2-2 Data Acquisition Flowchart from Data Loggers to HydroServer

A useful graphical display of the data was to compare the maximum daily NDVI over a given period using a time series plot with error bars, and include a box plot to compare the distributions of the two groups. A useful statistical test was the T-test for testing the null hypothesis "the difference between the mean NDVI at sites with small mammals and at sites without small mammals is equal to zero". The R computational environment provides functions for both exploratory analysis and statistical inference.

Because the experiment is a randomized controlled block experiment, used the paired t-test to test the null hypothesis: the difference between the mean NDVI at sites with mammals and

sites without mammals is equal to zero. For setting up the paired t-test, I identified the sites by block and by treatment. If two sites are in the same block and they have a different mammal treatment, I considered the two sites as a matched pair. The table has six rows (one per paired study plot) and the following columns: block, NDVI-mammals, NDVI-no-mammals, and difference. This table, as encoded in R, allowed can be used with the standard R t-test command:

```
Result = t.test(NDVI-mammals,NDVI-no-mammals, paired=TRUE)
```

The expectation of this analysis was to indicate whether there is a statistically significant difference between the mammal and no-mammal study plots.

1.1.8 Snowpack Modelling Case Study

Accurate information about snow depth and snow water content is important for water management, transportation, and recreation activities in snow-dominated regions. The accumulation of seasonal snowpack depends on the energy balance (incoming and outgoing solar radiation, air and soil temperature) and the water balance (solid or liquid precipitation, snowmelt, snow sublimation). Examples of conceptual and physically based snowpack models have been presented by Anderson (1973), Koivusalo et al. (2001), Walter et al. (2005), Lehning et al. (2006) and many other authors. The EcoHydRology R software package (Fuka et al. 2013) includes various process based hydrological modelling procedures, including the function `SnowMelt` for calculation of snowpack accumulation and melt. The `SnowMelt` procedure uses a simplified energy balance model (Walter et al. 2005). Unlike other commonly used energy balance models, it estimates the required energy balance parameters (net incident solar radiation, atmospheric long wave radiation, terrestrial long wave radiation, sensible heat exchange, energy flux associated with the latent heats of vaporization and condensation at the surface, ground heat

conduction to the bottom of the snowpack, heat added by rainfall, and change of snowpack heat storage) using only the day of the year, daily minimum and maximum temperature, and geographic latitude. This makes it an ideal model for initial testing of snow depth and snow water equivalent prediction using real-time data from distributed sensors. The goal of this case study was to demonstrate the use of online data from the CUAHSI HIS web services to setup a snowpack energy balance model within the R environment. The steps tested in this case study included data discovery, data quality control and gap filling, model simulation, and model validation.

1.2 Results

The following section introduces the deployment and testing of the WaterML R package. Three use cases demonstrating applications of the WaterML R package and the HydroServer Lite API for managing ecological experiment data are also presented: (1) Using the WaterML R package to upload real-time observations from field sensor to the HydroServer Lite online database, (2) Data analysis of experimental observations, and (3) Snowpack energy balance modeling in R using online data from CUAHSI web services.

1.2.1 Software Design and Development Results

The WaterML R package has been published on the CRAN (http://cran.r-project.org/web/packages/WaterML/). Any R user can find and install the WaterML R package in the "Install Packages" menu of the R environment. Another method of installation is using the R command line interface:

```
install.packages("WaterML")
```

Before being published on CRAN, the code had to be approved by running the automatic check on each function, checking the documentation of all function parameters, checking the code examples in the documentation, and a review by the CRAN team. The package has two dependencies: XML and RCurl. Both dependencies will be automatically installed when installing the WaterML R package. I successfully tested the installation on three different operating systems: Windows, Mac and Linux. When the user installs the package, the documentation and examples become available by typing the text: `??WaterML` in R.

For testing the ability of the WaterML R package to access data from different HydroServers I randomly selected 10 HydroServers registered at the HIS Central catalog. For each HydroServer I ran the GetSites and GetVariables functions to retrieve the table (data frame) of all sites and variables at the HydroServer. Then I chose 10 random site-variable combinations and called the GetValues function to get the observational data. Testing of the WaterML R package using existing CUAHSI HydroServers had the following results. The GetValues function returned the table (data frame) of observation times and values in 9 out of 10 HydroServers. The one failed case was due to server timeout on the HydroServer. It is not clear exactly why the 10th HydroServer had a timeout failure, though I expect it was due to server side problems on the individual HydroServer. Of course this illustrates one of the known challenges with implementing any distributed server based system: if any one server becomes inaccessible it can make the rest of the system less useful.

Overall the user experience proved functional in most aspects of the new WaterML R package. Two of the scientists on the project downloaded and ran the package in R. They were able to access the data that was desired. They were able to load it into R and analyze it. As the

researchers become more familiar with the functionality of R they will be able to use their data in other research questions and analyses.

1.2.2 Case Study Results – Adding Data to HydroServer

The HydroServer used in the case study is hosted on the MySQL and PHP cloud web hosting site http://worldwater.byu.edu/app/index.php/rushvalley. For the Rush Valley experiment, I had to decide how to represent the experimental design and measurements within the predefined ODM database schema. While I used the complete ODM database schema, the following tables in the ODM were the most important to the presented use case: Sites, Variables, Units, Methods, Sources, QualityControlLevels, SeriesCatalog and DataValues. To identify the block, treatment and depth of each site, I assigned a unique site code for each site (Table 2-2). For example the code Ru1BMP5 means: Ru - Rush Valley, 1 - Block 1, B - Burned, M - Mammals, P - Plus water, 5 - 5 cm depth. I also used the "Comments" field in the Sites table to explain the meaning of the abbreviations in the code of each site.

Table 2-2 Code of Site by Block, Treatment and Depth

	Area	**Block**	**Burned treatment**	**Mammal exclusion treatment**	**Water treatment**	**depth**
Possible values	Ru (Rush Valley)	1-5	B – Burned N – Not burned	M – mammals N – no mammals	P – plus water M – minus water	5 – 5 cm 30 – 30 cm

After setting up the ODM database and installed the HydroServer Lite software, the next step was to set up the uploading of observations to the ODM database via the web service API. Two types of data loggers deployed on the site had to be considered: online data loggers and offline data loggers. New data from the online data loggers becomes available via the Decagon

API every hour. Data from the offline loggers can be retrieved by visiting the experimental site and connecting the logger to a PC. Special 3rd party software "echo2o utility" can be used to convert the raw data from the data loggers to a spreadsheet file in .Microsoft Excel (xls) or tab-delimited text (txt) format. An example converted data table is shown in Table 2-3.

Table 2-3 Example of First Three Rows of the Data File from the Logger after Conversion of the Raw Data with Echo2utility

Measurement.Time	Port.1.MPS.2. Water.Potential.Temp. kPa.Potential	Port.1.MPS.2. Water.Potential.Temp. .C.Temp	Port.2.MPS.6. Water.Potential.Temp. kPa.Potential
9/4/2014 18:00	-340.6	26.6	-387.5
9/5/2014 0:00	-565.8	23.7	-480.4
9/5/2014 6:00	-962.6	19.1	-630.1

As a prerequisite for automated transfer of observations from the data files to the ODM on the HydroServer I also needed to make a lookup table (Table 2-4).

Table 2-4 Example Lookup Table to Associate the Decagon Logger and Response to the Time Series

Logger ID	Response	SiteCode	Variable Code	Method ID	Source ID	Quality ID
5G0E3559	Port.3.SRS.Nr.NDVI..630.nm	Ru1BMP5	SRS-NDVI	72	1	1
5G0E3559	Port.1.MPS.2.Water.Potential	Ru1BMP5	MPS6-WP	60	1	1
5G0E3562	Port.4.GS3.Moisture.VWC	Ru5BMM5	GS3-RH	62	1	1

The lookup table associates each (logger, response) pair with an ODM time series identified by the Site, Variable, Method, Source, and Quality Control Level attributes. With the data file and the lookup table in place, R can be used to automatically upload the data to the server. In the first step the data file is read into an R data frame. Because the time date/time column was not recognized by R automatically, I needed to specify the exact format of the

“MeasurementTime” column. The lookup entries corresponding to the logger ID are read from the lookup table

```
data = read.table("5G0E3559-processed.txt", sep="\t",
header=TRUE, stringsAsFactors=FALSE)
data$time = strptime(data$Measurement.Time,"%m/%d/%y %I:%M %p",
tz="GMT")
Lookup = read.table("lookup.csv", header=TRUE)
LoggerInfo = Lookup[Lookup$LoggerID=="5G0E3559",]
```

Using a loop, I cycle through all columns in the logger data file. The column header in the logger data file contains the response name. For each data column, I found an entry in the lookup table corresponding to the logger id, and the response name to find out the relevant ODM SiteCode, VariableCode, MethodID, SourceID and QualityControlLevelID. Then I prepared a table with “time” and “DataValue” columns, and I called the AddValues function from the WaterML R package, passing in DataValues table together with the user name, password, SiteCode, VariableCode, MethodID, SourceID and QualityControlLevelID:

```
library(WaterML) # load the WaterML R package
Username = "admin" # specify user name
Password = "my-Password-6031" # specify password
serverURL =
"http://worldwater.byu.edu/app/index.php/rushvalley/api/
columns = names(data) # get the data column names from the data
table
for (i in 1:length(columns)-1) {
    ColumnName = columns[i]
    DataColumn = data$ColumnName
    timeColumn = data$time
    DataToUpload = cbind(DataColumn, timeColumn)
    #find entry in the lookup table data logger info
    info = LoggerInfo[LoggerInfo$response == ColumnName]
    #send data to HydroServer Lite
    Status = AddValues(server=ServerURL, user=Username,
password=Password,
                                        siteCode=info$SiteCode,
variableCode=info$VariableCode,
```

```
                                                    methodID=info$MethodID,
sourceID=info$SourceID,
                                                    qcID=info$QualityID)
} # end of the loop
```

The near real-time data from online data loggers are automatically updated to the HydroServer ODM database on a daily basis. A researcher also visits the site every few weeks to transfer data from the offline loggers to a laptop and to HydroServer Lite using the data upload R script. Once the data has been uploaded to HydroServer Lite, the locations of the sites can be viewed in the HydroServer Lite web user interface in an interactive map (Figure 2-3).

Figure 2-3 HydroServer Map Page

Each site has a details page for viewing and editing the measured data values (Figure 2-4). The WaterOneFlow web service of the HydroServer has also been registered in the HIS Central catalog, making it possible to discover sites and display them alongside sites from other research organizations.

Figure 2-4 HydroServer Site Details Page Showing Soil Moisture at Selected Sensor

Because the data uploading functions in the WaterML R package relied on the HydroServer Lite data upload API, it could not be tested it with any randomly selected HydroServers from the HIS Central catalog. As an alternative method to test the data uploading functions, I created three new HydroServer Lite instances hosted on different web servers and I used the WaterML R package to create up to 10 random sites, variables, methods, sources and quality control levels. Then I chose 10 random combinations of site, variable, method, source and quality control level, and uploaded between 1 and 100 data values using the AddDataValues function. After each upload, I used the GetValues function to retrieve the data and check if the values are equal. The data upload was successful for all test cases.

1.2.3 Case Study Results – Data Analysis of Experimental Observations

For exploratory data analysis I examined the effect of small mammal exclusion treatment on the NDVI. I compared the time series of maximum daily NDVI values at sites with and without the mammal exclusion treatment. In the first step I used the `GetVariables` function to check all available variables and to find which variable name stands for the "NDVI":

```
library(WaterML)
server =
"http://worldwater.byu.edu/app/index.php/rushvalley/services/cua
hsi_1_1.asmx"
variables = GetVariables(server)
```

From the table of available variables, I find that the name for "NDVI" is "SRS_Nr_NDVI". In the second step I used the `GetSites` function of the WaterML R package to obtain a table of all available sites:

```
#get the sites
all_sites = GetSites(server)
```

In the next step I use the `GetSiteInfo` function in a loop to find out which sites have measurements of the SRS_Nr_NDVI variable.

```
all_site_info = NULL
for (i in 1:nrow(all_sites)){
  all_site_info = rbind(all_site_info, GetSiteInfo(server,
all_sites$FullSiteCode[i]))
}
NDVI_sites =
all_site_info[all_site_info$VariableCode=="SRS_Nr_NDVI",]
```

Once the sites are selected, I call `GetValues` function for each selected site, passing in the SiteCode, VariableCode, StartDate and EndDate parameters. The result of this step is a data.frame table in the "long" format with SiteCode, Time and DataValue columns.

```
#get the values from all sites that measure NDVI
variable = "SRS_Nr_NDVI"
startDate = "2014-10-07"
endDate = "2014-10-15"
data = NULL
for (i in 1:nrow(NDVI_sites)){
  site = NDVI_sites$SiteCode[i] #the site code
  var = NDVI_sites$VariableCode[i] #the variable code
  values = GetValues(server, site, var, startDate,  endDate,
daily="max")
  #check for missing data
  if (is.null(values)) next
  #add the siteCode colum
  values$siteCode = site
  data = rbind(data, values)
}
```

The treatment code (mammal or no mammal) can be identified from the 5th position of the SiteCode according to Table 2-2 and I added it as an extra column to the downloaded data values table.

```
data$treatment = substring(data$siteCode, 5, 5)
```

In the final step I summarized the data by day and by treatment using the ddply function from the plyr package.

```
library(plyr)
data.summarized = ddply(data, ~time+treatment, summarise,
mean=mean(DataValue), sd=sd(DataValue))
names(data.summarized)[3] = "daily.max.NDVI"
```

For visualization I used the ggplot function from the ggplot2 package to create the error bar plot (Figure 2-5). The error bars show plus or minus one standard deviation of the daily NDVI at all sensors in the group. The group (mammal treatment or no mammal treatment) is indicated by the color of each NDVI error bar.

```
library(ggplot2)
# Plot: errorbars using position_dodge to move overlapping bars
```

```
pd = position_dodge(0.25)
ggplot(data.summarized, aes(x=time, y=daily.max.NDVI,
ymax=max(daily.max.NDVI), colour=treatment)) +
geom_errorbar(aes(ymin=daily.max.NDVI-sd,
ymax=daily.max.NDVI+sd),      width=.5, position=pd) +
  geom_line(position=pd) +
  geom_point(position=pd)
```

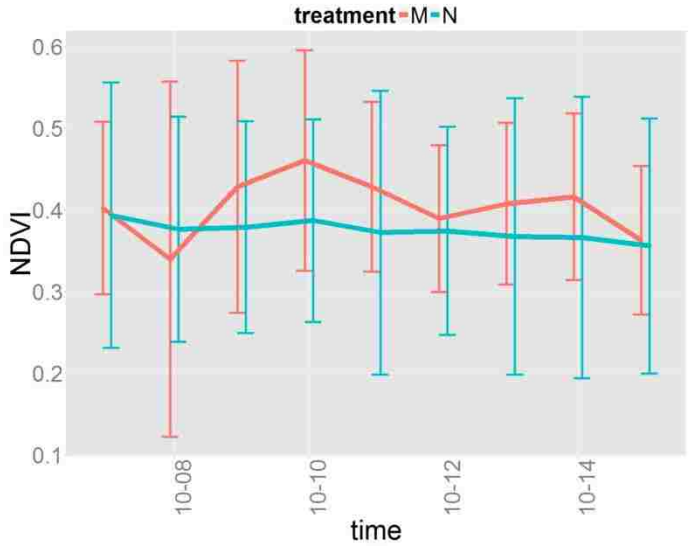

Figure 2-5 Daily Time Series Plot with Error Bars Showing the Maximum Daily NDVI at Sites with and without Mammal Exclusion Treatment

For testing the hypothesis "there is no difference in the NDVI value at sites with and without mammal treatment" I used the two-sample t-test in R. Figure 2-6 shows a side-by-side box plot and the p-value. The p-value is 0.193, meaning there is no significant difference between the means of the two groups. The following R-code creates the box-plot and the t-test:

```
boxplot(DataValue~treatment, data=data)
data.mammal = data[data$treatment == "M",]
data.nonmammal = data[data$treatment == "N",]
t.test(data.mammal$DataValue, data.nonmammal$DataValue,
var.equal=TRUE)
```

The test gives the following results:

```
data:  data.mammal$DataValue and data.nonmammal$DataValue
t = 1.3078, df = 142, p-value = 0.193
alternative hypothesis: difference in means is not equal to 0
95 percent confidence interval:
 -0.0149902  0.0735992
sample estimates:
mean of x mean of y
0.4037878 0.3744833
```

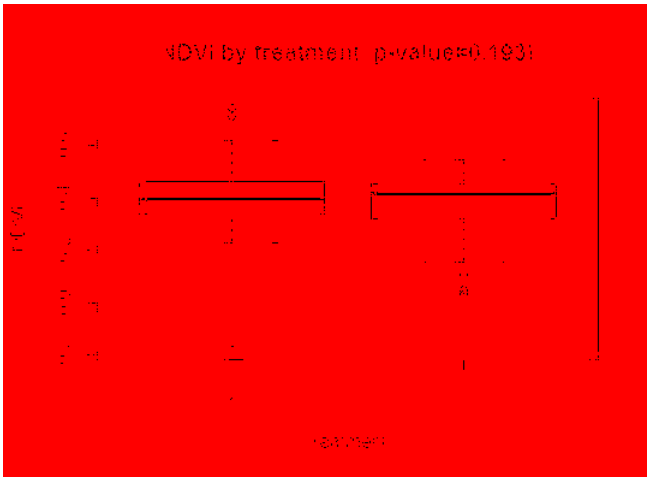

Figure 2-6 Box Plot and P-Value for Comparing Mean Differences in NDVI between Groups

By transferring data from the HydroServer WaterOneFlow web service to R, the WaterML R package makes it possible to use the built-in functions of R or third-party R packages to check for outliers or to detect data errors. Once the data errors have been detected, a new quality-controlled version of the data can be sent back to HydroServer Lite using the AddValues function in of the WaterML R package. The ODM database schema allows to associate a "quality control level" attribute with each data value. Both the raw data and the quality controlled data can be kept in the same ODM database, preserving the quality control process history.

1.2.4 Case Study Results: Snowpack Modelling in R

To test the SnowMelt snow energy balance model from the EcoHydRology R package with data inputs from WaterOneFlow web services, I chose a study area in the Jizera Mountains near the border of the Czech Republic and Poland. The watersheds in this area are largely snow-dominated, and provide water supply for the nearby city of Liberec. The proximity to large urban areas makes the Jizera Mountains a popular region for winter recreation. At the same time, a relatively low elevation (800 – 1100 m above sea level) makes the existence of the seasonal snowpack vulnerable to the effects of global warming.

Before setting up the model, I need to find sites with available input data. At minimum the SnowMelt model requires three daily time series: precipitation sum, maximum air temperature, and minimum air temperature. I define the geographic extent of the study area (longitude between 15°E and 15.4°E, latitude between 50.7°N and 50.9°N), and use the `HISCentral_GetSeriesCatalog` function from the WaterML R package to search for time series with keywords “snow depth”, “precipitation”, and “temperature” in the time period of October 2014 – May 2015.

```
# specify geographic extents of the study area
lonR=c(15.0, 15.4)
latR=c(50.7, 50.9)

# search snow depth sites
snowSites = HISCentral_GetSeriesCatalog(lonR[1], latR[1],
lonR[2], latR[2],beginDate="2014-10-01", endDate="2015-05-01",
keyword="snow depth")

# search precipitation and temperature sites
precipSites = HISCentral_GetSeriesCatalog(lonR[1], latR[1],
lonR[2], latR[2],beginDate="2014-10-01", endDate="2015-05-01",
keyword="precipitation")
tempSites = HISCentral_GetSeriesCatalog(lonR[1], latR[1],
lonR[2], latR[2],beginDate="2014-10-01", endDate="2015-05-01",
keyword="temperature")
```

The results show that 3 time series of snow depth, 18 time series of air temperature, and 23 time series of precipitation were found in the study area. In the next step I need to select sites that have observations of all three variables. To check the locations of the sites and the available variables at each site, I can use the RGoogleMaps package (Loecher and Ropkins 2015) for simple map visualization. The `GetMap.bbox` function retrieves a background map image from the Google Maps satellite layer, the `PlotOnStaticMap` displays point symbols for sites with different variables, and the `TextOnStaticMap` shows the labels of snow measuring sites. The cex, pch and col parameters control the size, shape, and color of each point symbol. The resulting map plot is shown in Figure 2-7.

```
mymap = GetMap.bbox(lonR, latR, maptype="satellite")
PlotOnStaticMap(mymap, lat=snowSites$Latitude,
lon=snowSites$Longitude, cex=2.1, pch=19, col="red", FUN=points)

PlotOnStaticMap(mymap,lat=tempSites$Latitude,
lon=tempSites$Longitude, cex=1.4, pch=19, col="yellow",
FUN=points, add=TRUE)

PlotOnStaticMap(mymap, lat=precipSites$Latitude,
lon=precipSites$Longitude, cex=0.8, pch=19, col="blue",
FUN=points, add=TRUE)

TextOnStaticMap(mymap, lat=snowSites$Latitude+0.01,
lon=snowSites$Longitude, labels=snowSites$SiteName, col="white",
add=TRUE)
```

As seen in Figure 2-7, there are two sites in the study area that measure air temperature, precipitation and snow depth at the same site (Bedřichov and Desná – Souš). Having snow depth observations will allow to verify the results of the model simulation run. For this case study I selected the site Bedřichov (820 m above sea level). After filtering the discovered snow depth, precipitation and air temperature time series to entries with SiteName="Bedrichov", the ServiceURL, FullSiteCode and FullVariableCode fields can be used as parameters of the

WaterML `GetValues` function to download the time series data from Bedrichov from the 2014/2015 winter season.

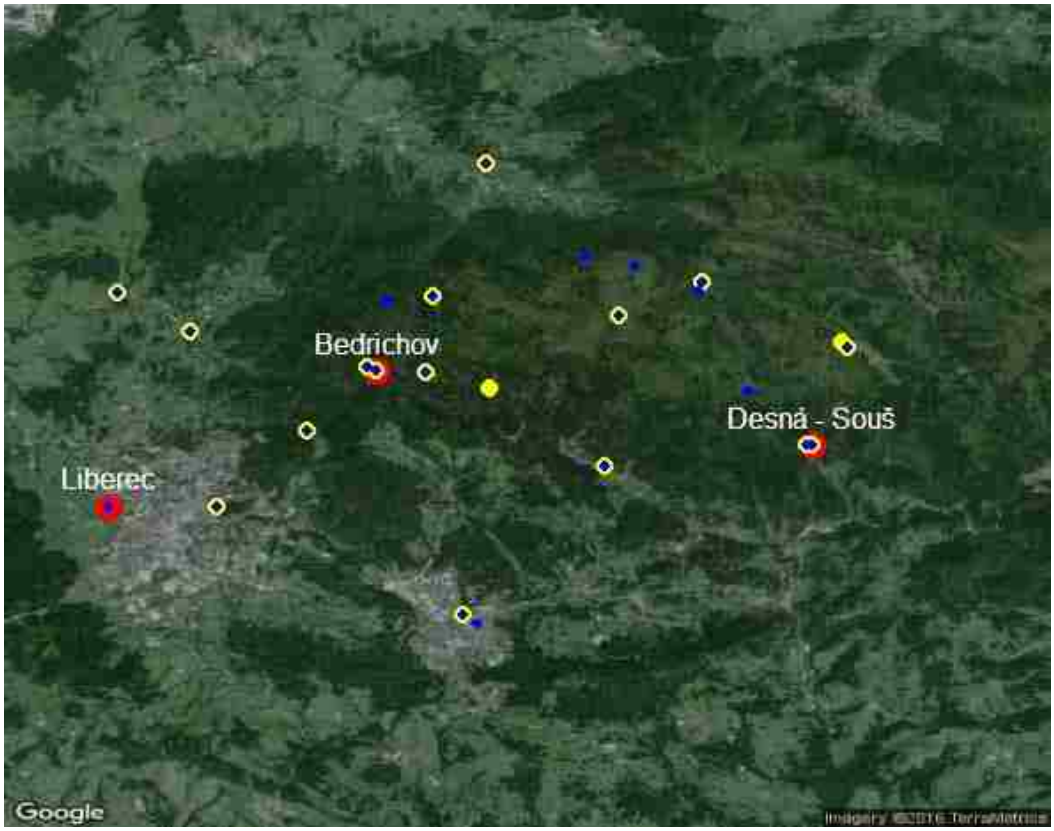

Figure 2-7 Google Map Plot Showing Variables Measured at Sites (Red: Snow, Yellow: Temperature, Blue: Precipitation)

```
startDate = "2014-11-01"
endDate = "2015-04-30"

snowSite = snowSites[snowSites$SiteName=="Bedrichov",]
precSite = precipSites[precipSites$SiteName=="Bedrichov" &
precipSites$TimeUnits=="hour",]
tempSite = tempSites[tempSites$SiteName=="Bedrichov" &
tempSites$TimeUnits=="hour",]

snowData = GetValues(snowSite$ServiceURL[1],
snowSite$FullSiteCode[1], snowSite$FullVariableCode[1],
startDate, endDate)
```

```
precData = GetValues(precSite$ServiceURL[1],
precSite$FullSiteCode[1], precSite$FullVariableCode[1],
startDate, endDate)

tempData = GetValues(tempSite$ServiceURL[1],
tempSite$FullSiteCode[1], tempSite$FullVariableCode[1],
startDate, endDate)
```

The resulting snow, precipitation and temperature data frames contain 180, 4320 and 4320 rows, respectively. This is because the snow data are daily, and the precipitation and temperature data are hourly. After downloading the observations, the next step is to check the time series for possible errors and missing data. This step is necessary because the SnowMelt model requires a continuous input time series for all three parameters. A widely used R library for time series operations is the eXtensible Time Series (xts) package (Ryan and Ulrich 2011). To convert the data frames to xts objects, I use the `xts` function with the time column as the ordered index. The plot function can then be used to display the time series chart and visually inspect the time series plots for data gaps (Figure 2-8).

```
snowTS = xts(snowData$DataValue, order.by=snowData$time)
precTS = xts(precData$DataValue, order.by=precData$time)
tempTS = xts(tempData$DataValue, order.by=tempData$time)

layout(matrix(1:3, 3, 1))
plot(tempTS, main="temperature (C)")
plot(precTS, main="precipitation (mm/h)")
plot(snowTS, main="snow depth (cm)")
```

As shown in Figure 2-8, there is a period of missing data in November – December 2014. It also appears that the snowpack only started to accumulate by end of December 2014, and completely melted in the first half of April 2015.

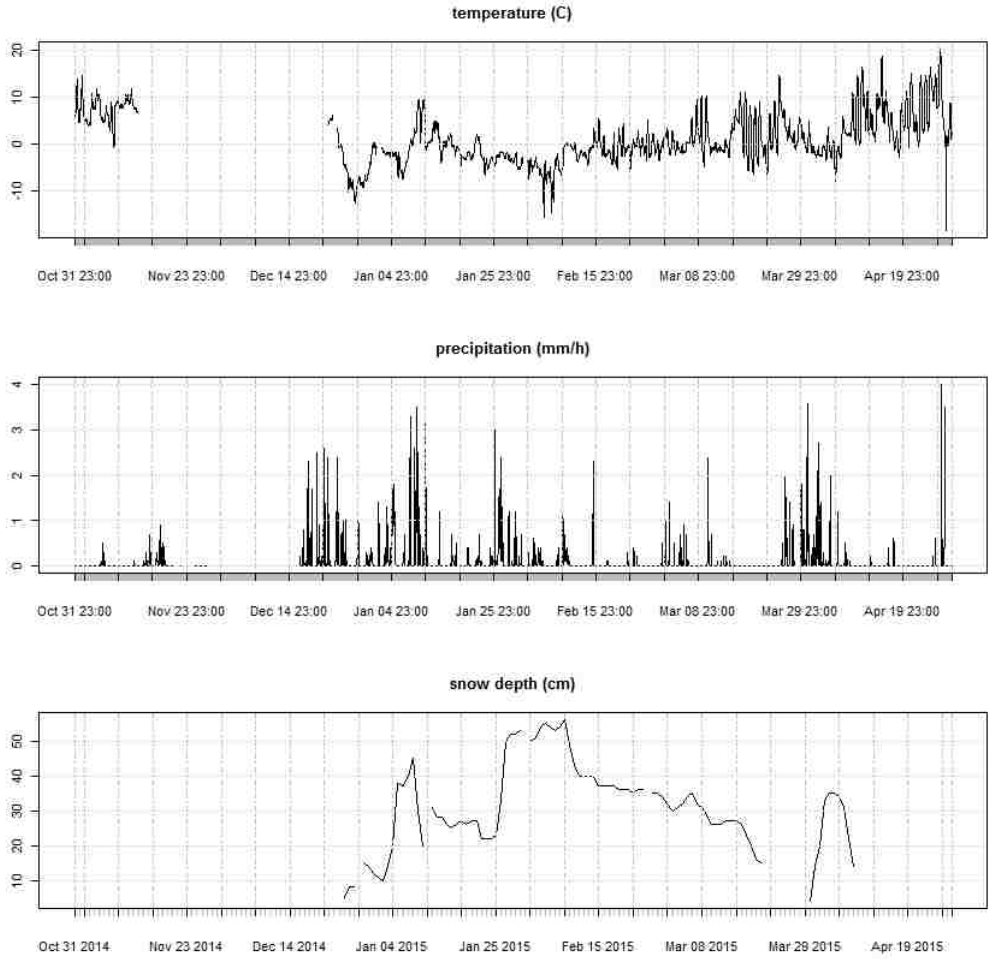

Figure 2-8 Temperature, Precipitation and Snow Depth at Bedřichov in Winter 2014/2015

Therefore I limited the model simulation time period to 25th December 2014 – 15th April 2015 by using the xts time series subset operation. To check if there is any missing data in the limited time period, I used the `is.na` function that selects the observations with value labeled as `NA` (`NA` is the symbol for missing or unavailable data in R).

```
snowTS = snowTS["20141225/20150415"]
precTS = precTS["20141225/20150415"]
tempTS = tempTS["20141225/20150415"]
precMissing = precTS[is.na(precTS)]
tempMissing = tempTS[is.na(tempTS)]
```

This shows that there are 38 missing precipitation values, and 46 missing temperature values. Therefore it is necessary to apply a gap filling function. Several interpolation and gap filling functions are available in R. To interpolate missing temperature values, I chose a polynomial interpolation method using the `na.spline` function. The polynomial interpolation is appropriate for time series with a frequent periodic cycle, such as air temperature with daily oscillation (Baltazar and Claridge 2006). The interpolated hourly air temperature values are shown in Figure 2-9. For precipitation a linear or spline interpolation would only be appropriate during a continuous precipitation event. As I did not have other data to suggest it was raining continuously before, during or after the missing hours, I replaced the missing precipitation values by zeros.

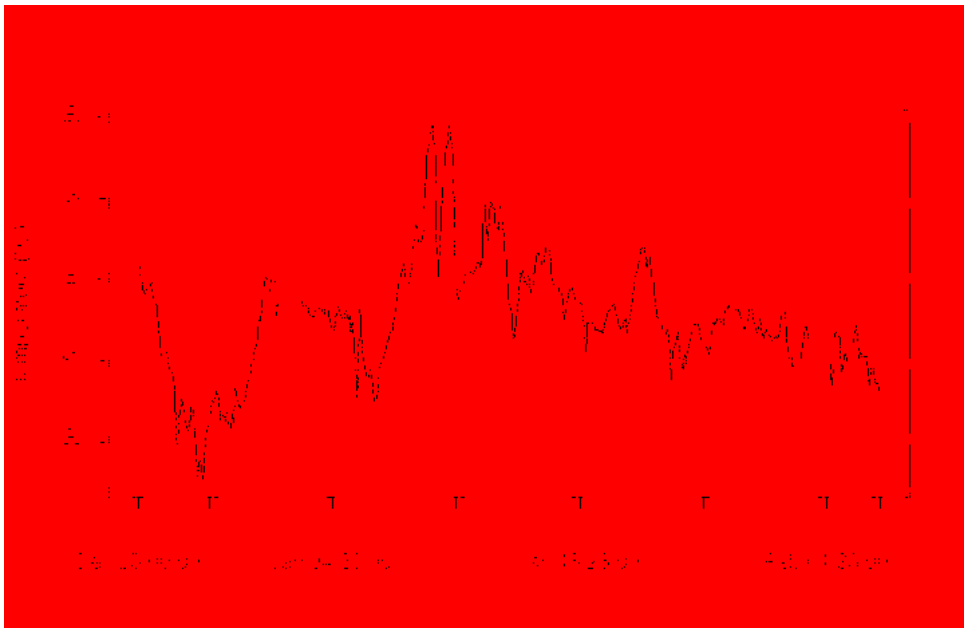

Figure 2-9 Using Spline Interpolation to Replace Missing Temperature Values

```
# replace missing values by zeros for precipitation
prec <- precTS
prec[is.na(prec)] <- 0
# use spline polynomial interpolation for temperature
temp <- na.spline(tempTS)
```

```
plot(tempTS["20141225/20150204"], ylab="temperature", main="")
points(temp[is.na(tempTS)], col="red")
```

The final step before running the model is to convert the hourly precipitation and temperature values to daily precipitation sum, daily maximum temperature, and daily minimum temperature using the xts `apply.daily` function.

```
precDaily <- apply.daily(prec, sum)
tmaxDaily <- apply.daily(temp, max)
tminDaily <- apply.daily(temp, min)
```

For launching the SnowMelt simulation I load the EcoHydRology package. The SnowMelt function requires the following five mandatory input parameters: *Date*, *precip_mm*, *Tmax_C*, *Tmin_C* and *lat_deg*. The *lat_deg* is the latitude of the site in decimal degrees, which is contained in the output from the `HISCentral_GetSeriesCatalog` function call. The *precip_mm*, *Tmax_C* and *Tmin_C* parameters need to be numeric vectors and they can be easily obtained from the daily time series using the `as.numeric` function. Other optional parameters of the SnowMelt function include wind speed, ground albedo, slope and aspect. In this case study I kept the default values of all optional parameters.

```
dates = strftime(as.Date(index(precDaily)), "%Y-%m-%d")
latitude = precBedrichov$Latitude[1]
tmax = as.numeric(tmaxDaily)
tmin = as.numeric(tminDaily)
precip = as.numeric(precDaily)
modeledSnow = SnowMelt(Date=dates, precip_mm=precip,
tmax_C=tmax, tmin_C=tmin,lat_deg=latitude)
```

The output of the SnowMelt function is a data frame with the following state variables: rainfall, snowfall water equivalent, snow melt, new snow depth, total snow depth, and total snow

water equivalent. For comparing the simulation result with observations, I can plot a time series of the modelled and observed snow depth (Figure 2-10).

```
modeledDepths = xts(modeledSnow$SnowDepth_m * 100,
order.by=as.Date(dates))
plot(modeledDepths, ylim=c(0, 60), ylab="snow depth (cm)")
points(snowTS, col="red")
legend("topright",
       legend=c("observed", "simulated"),
       pch=c(1, NA),
       col=c("red", "black"),
       lty=c(NA, 1))
```

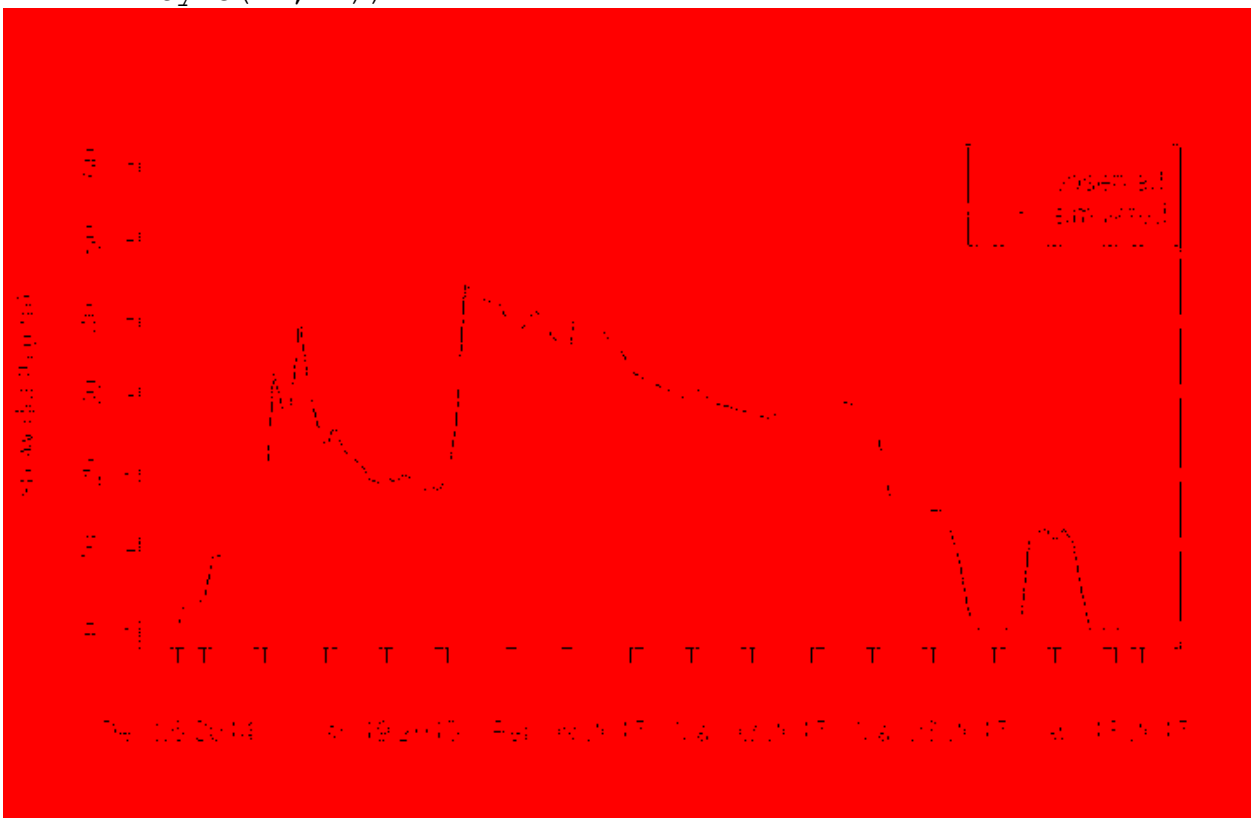

Figure 2-10 Comparison of Observed and Simulated Snow Depth using the SnowMelt Model

Figure 2-10 shows that the energy balance model provides a good prediction of the timing of snow accumulation and snow melt periods. The model underestimates the overall snow depth, which may be due to the missing precipitation values that were set to zero. The model also

underestimates duration and snow depth of the late season snow event in April 2015. The regression plot of observed and simulated values is shown in Figure 2 11. The coefficient of determination (R2) is 0.76, suggesting a good fit.

```
# regression analysis for validating model
m = lm(observed~simulated, data=comparison)
rsquared = round(summary(m)$r.squared, digits=4)
title1 <- paste("R²", "=", rsquared)

library(ggplot2)
ggplot(comparison, aes(y=simulated, x=observed)) +geom_point() +
  geom_abline(intercept=0, slope=1) +
  xlab("Observed snow depth (cm)") +
  ylab("Simulated snow depth (cm)") +
  xlim(0, 60) +
  ylim(0, 60) +
  ggtitle(title1) + theme_grey(base_size = 16)
```

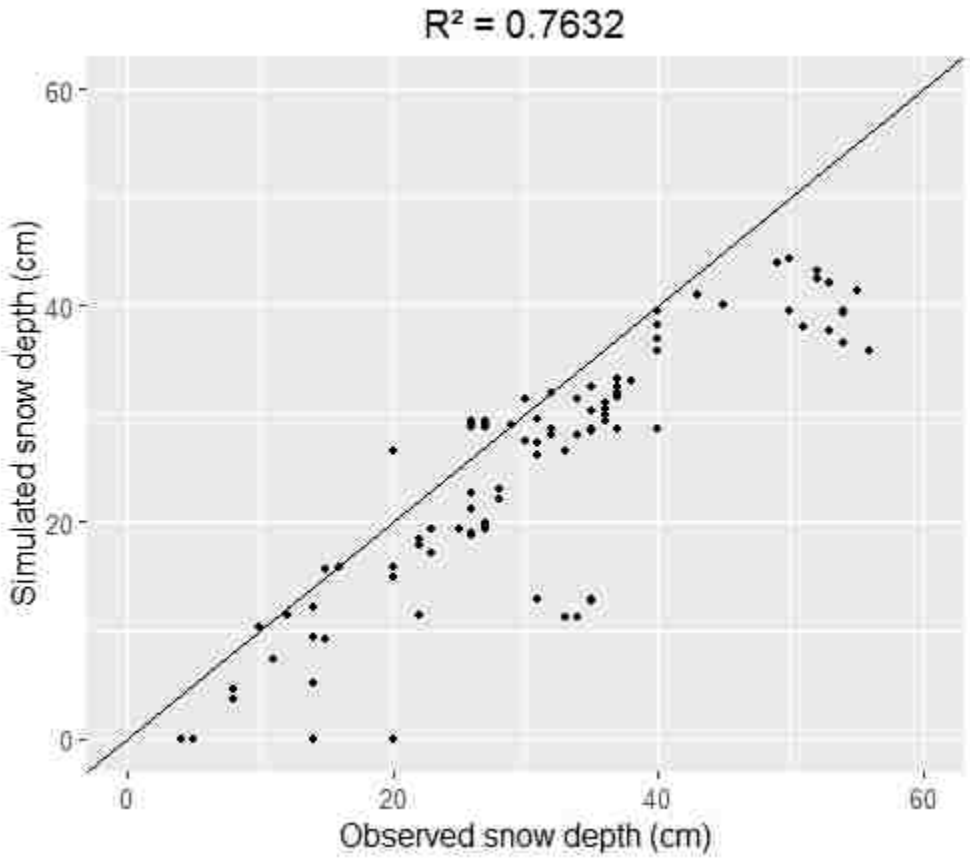

Figure 2-11 Regression Plot of Simulated and Observed Snow Depth at Bedřichov Using the R SnowMelt Model

In the study area there are only three sites with snow depth observations, but 14 sites with simultaneous precipitation and temperature observations (see Figure 2 7). The WaterML R package together with the R SnowMelt model could be used to simulate snow depth for these 14 sites to improve the spatial resolution and accuracy of snow depth maps of the study area.

1.3 Discussion and Conclusions

Ecological data management combined with data analysis presents a number of challenges for practicing researchers. In particular, there is a need for tools that link commonly used desktop analytical software tools with distributed cloud based data sharing networks. This paper presents a new WaterML R package that crosses the divide between local computation and network based data sharing. The new package, described here, has been successfully deployed in the context of a case study for a large scale ecological manipulation project.

The case study used to test the new WaterML R package required the ability to acquire, store, manage, and analyze data from new wireless Decagon sensors that were installed in the Rush Valley experiment in central Utah, and make it available to all researchers on the project. Another requirement was to integrate R with the data management system through a web service interface because researchers and external collaborators use R for statistical analysis and graphical display of the experimental data.

Every time the logger sends in new data, it becomes available for viewing on the HydroServer. Anyone interested in the data can find the HydroServer WaterOneFlow web service on the HIS Central Catalog (hiscentral.cuahsi.org), discover the sites and variables on the hydrologic data search website (data.cuahsi.org), and access the data for analysis in R.

The HydroServer instance also provides a backup of the data. In a small lab such as ours, there is limited support for transferring data from real-time sensors to HydroServer. For the initial model creation I entered manually the sites, variables, source and methods.

For uploading data values, I initially considered several methods: (1) add values using the HydroServer Lite web user interface, (2) add values using database connection, or (3) add values using HydroServer web API. Method (1) was not practical due to the large quantity of data because new observations are taken every hour. Method (2) required a remote connection directly to the database which was not accessible to collaborators outside the university campus. Therefore I chose method (3): Add values using the newly constructed HydroServer Lite web API.

The complete source code and documentation of the HydroServer Lite including the upload API is available on the website (http://hydroserverlite.codeplex.com). Because it is open-source, the API can be customized for other implementations of the ODM database and the HydroServer. The documentation includes an example R script showing how to use the upload API. This script can be easily customized for connecting other data loggers to HydroServer Lite.

One limitation encountered while creating the ODM database for the experimental site was describing the treatment method. Because the ODM has established tables designed primarily for observational data, I had to be creative in organizing the experimental results. I included the code of the experimental treatment in the name of the site in the sites table, because I determined it was important to preserve the "Methods" table strictly for data collection method, not treatment method. Two different treatments are represented as two different sites, although the sites might be in the same physical location. It was necessary for to carefully label sites so that I would know that they are not sites, but sensors.

With HydroServer and the WaterML R package, I was able to access and analyze HydroServer data directly in R. The R script can be shared with others, making the analysis easier to reproduce. Without the R package, it would be necessary to download the files from each data logger and sensor, convert the data logger files, link them to the treatment method and import them to statistical software. The R environment provides numerous tools for ecological and hydrological modelling. Using a simple snow melt example demonstrated how to find model input data, replace missing values, and perform model validation using online data from CUAHSI WaterOneFlow web services within the R environment. All steps of the analysis are saved in a script, making it easy to share the script with other users. For example, another user may be interested in testing the SnowMelt model in a different study area. Provided that CUAHSI HIS temperature, precipitation and snow depth data is available in the new study area, the only required modification is changing the study area bounding box and the selected site name.

I encountered a few limitations in developing the WaterML R package. For example, the WaterOneFlow web service does not have a query function for finding all sites that measure a specific variable. Therefore, the WaterML R package had to issue multiple web requests to the GetSiteInfo WaterOneFlow web method for each site to find out which sites have data for the given variable. The user of the WaterML R package must know how I described each treatment method in the site name to accurately group the data for the analysis. Therefore it is important to publish a documentation of the experimental design setup including detailed description of the experimental treatment methods on the HydroServer website.

Another limitation of the data upload functions (`AddSites`, `AddVariables`, `AddMethods`, `AddSources`, `AddValues`) in the WaterML R package is that these functions

only work with the HydroServer Lite (hydroserverlite.codeplex.com) PHP and MySQL implementation of the CUAHSI HydroServer. The Windows version of the CUAHSI HydroServer (hydroserver.codeplex.com) does not presently have a data upload API, and therefore the WaterML R package cannot be used with the CUAHSI HydroServer. Until a data upload web API is developed for the Windows HydroServer, other R packages like RObsDat must be used instead (requiring direct database access). The CUAHSI Water Data Center is presently considering the development of a data upload web service API for the Windows / SQL Server version of the HydroServer. This implementation of the HydroServer Lite data upload API in connection with the WaterML R package can serve as a prototype for creating two-way connections between analytical software tools like R and web services based data management systems like HydroServer.

The open source R package presented here is expected to support a wide variety of data sets, sites, and experiments, allowing scientists in small research labs to acquire, store, manage, and analyze data in a centralized location that is accessible to all scientists involved. At the same time, by integrating with HydroServer, these data will become discoverable by other researchers and potential collaborators globally.

2 EXTRACTING SNOW COVER TIME SERIES FROM OPEN ACCESS WEB MAPPING TILE SERVICES

On average between 4 mil km^2 and 47 mil km^2 of Earth's northern hemisphere's land surface is covered by snow (Brown and Robinson 2011). Information about the extent of the snowpack is important in climatology (Groisman et al. 2006; Karl et al. 1993), hydrology (Barnett et al. 2005), and recreation (Braunisch et al. 2011). Snowpack is a spatial and temporal field that rapidly changes in both place and time. Accurate information regarding the presence or absence of snow can improve the quality of hydrologic model forecasts (Barnett et al. 2005).

A snow monitoring sensor can be installed at any site. However, if the site is remote or the cost of the sensor is high, then external snow data sources need to be used. These sources are typically remote sensing images or outputs of snowpack models (Rees 2005). The external data sources differ in type of sensor, scale and time support. Therefore it is important to be able to compare how the output of the different sensors changes in time. Ideally, for any point (location) or sub-area (watershed) in the study area should have a time series of snow probability at the highest available time support from all available sensors. Many studies highlight the importance of web-based tools (Bambacus et al. 2008), open standards (Bai et al. 2012), and web services (Salas et al. 2012) for making climate data products, including snow data, accessible to the users. Having accessible information about data origins and history is also necessary in ecological research, especially when the data go through multistep processes of aggregation, modeling and analysis (Reichman et al. 2011).

Maps of present and historical snow cover have been published for many regions of the world. The sources of these maps are: classified satellite images, interpolated ground observations, predictions of snowpack simulation models, or a combination of the above. The U.S. National Snow and Ice Data Center (NSIDC) provides open access to global daily, 8-day and monthly snow cover datasets created by classifying images from the Moderate Resolution Imaging Spectroradiometer (MODIS) at 500 m resolution (Hall et al. 2006c). The NSDIC also publishes the Interactive Multi-Sensor Snow and Ice Mapping System (IMS) with daily Northern Hemisphere snow and ice analyses at 4 km and 24 km resolution (Helfrich et al. 2007; Ramsay 1998).

A variety of file formats and web service interfaces are being used for distribution of spatio-temporal snow cover data. For interoperability, standardized web services recommended by the Open Geospatial Consortium (OGC) include the Web Coverage Service (WCS), Web Map Service (WMS), and Web Map Tile Service (WMTS). The WCS (OGC 2012b) is a web service protocol specially designed for web-based retrieval of digital geospatial information representing space/time varying phenomena. Using a `GetCoverage` request, the WCS defines access to a subset of the digital dataset in the space or time dimension. The WMS (OGC 2006) is a web service protocol for delivering map images over the Internet that are generated by a map server. It is mainly used for visualization purposes. For more efficient delivery of multi-temporal earth observation data, several extensions of the WCS and WMS standard have been proposed, such as EO-WCS (Schiller et al. 2011) and EO-WMS (Baumann et al. 2015).

The Cryoland Consortium (Triebnig et al. 2011) provides access to snow extent and snow water equivalent data products from Europe. These include daily fractional snow cover from MODIS optical satellite data with uncertainty estimation and daily snow water equivalent data

based on observations of the SSM/I microwave radiometer and selected ground-based weather stations. Cryoland uses the EO-WMS web service standard for data visualization and the EO-WCS standard for data download. A limitation of the Cryoland interface is that it only provides data for the European continent. While the web map interface of Cryoland enables the user to select a polygon and download data from the polygon in GeoTiff format, the underlying EO-WCS web service does not allow it to retrieve a multi-temporal time series of fractional snow cover or snow water equivalent for a specific location in one request (Triebnig et al. 2011).

The Web Map Tile Service (WMTS) has become popular for multi-dimensional geographic data visualization on the Internet (Sample and Ioup 2010). In the tile system, the data are organized at a number of pre-defined scales. For each scale, the mapped area is divided into many square tiles with a size of 256 by 256 pixels. Each tile is stored as an image on the internet server (Batty et al. 2010). Using the tiles to enable rapid visualization in interactive maps at multiple scales is demonstrated by the National Aeronautics and Space Administration (NASA) Global Imagery Browse Service (GIBS) documented by (Cechini et al. 2013) at http://earthdata.nasa.gov/labs/worldview/. The GIBS provides highly responsive and scalable access to near real-time imagery. The wide adoption of the WMTS standard for publishing snow data is illustrated by the ArcGIS Online (arcgis.com) search catalog, where more than 50 different snow-related WMTS datasets are discoverable. The WMTS can be accessed using the web browser and in geographic information systems (GIS).

Extracting time series from multi-temporal snow cover satellite datasets has been applied in many studies (Dozier et al. 2008; Gascoin et al. 2014; Parajka and Blöschl 2006; Rittger et al. 2013; Şorman et al. 2007). These studies examine the accuracy of one or more snow cover

products for specific study areas, different seasons, type of vegetation, climate zones, or terrain. A major challenge is filling the gaps in the snow data on cloudy days (Hall et al. 2010).

Several applications have been developed to simplify the distribution of MODIS snow cover data. One application that has the goal of simplifying access to MODIS snow data is Cryoland (www.cryoland.eu). They provide an interactive web map interface with a time slider to view and download European snow cover datasets from multiple remote sensing data sources.

Another web based application is LifeWatch (http://maps.elie.ucl.ac.be/lifewatch). The LifeWatch application uses the eight-day maximum snow extent product of NASA as input data that has a spatial resolution of 500 m (MOD10A2). These data were filtered in the frame of the ESA Land Cover CCI project to derive weekly snow probability products (Basset and Los 2012). However, LifeWatch is limited in that it only provides data for Europe and it only supports extraction of long-term mean seasonal snow cover percentage. The National Aeronautics and Space Administration (NASA) have developed the Reverb ECHO interactive map (reverb.echo.nasa.gov) for search, discovery, and download of various Earth observation datasets including MODIS snow cover data. The user selects the geographic area and a time period to search and download all data files that overlap it. Typically each time step is downloaded as a separate file covering the area of interest. While the reverb allows the user to download multiple files, the user still needs to manually process each downloaded file for every time step to get a time series for any selected pixels.

This paper presents the design and development of a new web-based automated script for the extraction of snow cover probability directly from WMTS published by numerous open access data sources. The script is deployed as a web application using the Tethys framework (Jones et al. 2014) and includes both a user interface and an application programmer interface

(API) for external third party access. The script is developed such that it can extract snow cover probability time series based on the specification of a point location. The user can access an interactive map to select the point of interest, select the time range, and display and download the time series. An external application can use a web API to send the feature of interest and retrieve the time series.

The remainder of this paper is organized as follows. A methods section describes the design of the script as well as the design of a case study experiment for testing and validating the script using the MODIS fractional snow cover data retrieved through the NASA GIBS WMTS web service. A results section presents the finalized tool in user interface and web-API form as well as the results of the case study. Finally, I present conclusions and discussion regarding the application of the script to hydrologic and climate modeling scenarios. Opportunities for creating new tools following a similar approach are also presented, highlighting potential broader implications of this work.

2.1 Material and Methods

The following section starts with an overview of the MODIS snow cover data acquisition process. The algorithm used for automated WMTS time series data extraction is described. The software design and testing methods are outlined, and an application use case of the software for ground validation of the MODIS snow dataset is presented.

2.1.1 MODIS Snow Cover Data Acquisition

The original MODIS data are acquired by the Terra and Aqua satellite sensors and go through a number of processing steps and intermediate products (Riggs et al. 2006):

1. The raw dataset (Level-1A) is received from the satellite sensor with radiance counts for 36 MODIS channels, along with instrument status and spacecraft ancillary data.

2. The calibrated radiance data products (MOD02HKM and MOD021KM), the geolocation product (MOD03) and the cloud mask product (MOD35_L2) are created.

3. The Normalized Difference Snow Index (NDSI) is used to identify pixels as snow, snow-covered water body, land, water, cloud or other condition, and to calculate fractional snow cover in each pixel. The Normalized Difference Vegetation Index (NDVI) is used to improve snow detection in forested areas (Klein et al. 1998). The swath product (MOD10_L2) has a 500 m spatial resolution and it is archived in a Hierarchical Data Format – Earth Observing System (HDF-EOS). These datasets are archived online at: ftp://n5eil01u.ecs.nsidc.org/SAN/MOST/MOD10_L2.005/ with one file for each day, observation time, and swath region (Hall et al. 2006a). They can also be downloaded using the National Aeronautics and Space Administration (NASA) Reverb ECHO interactive download website (http://reverb.echo.nasa.gov)

4. The MODIS swath source data are continuously updated due to ongoing missions and the latest data become available within 3 hours of observation. The source data have also been updated after reprocessing campaigns and they are labeled by a version number. The version number used in this study is version 5.

5. Web applications (NASA Worldview; https://earthdata.nasa.gov/labs/worldview/) and web services such as the Global Imagery Browse Service (GIBS; https://wiki.earthdata.nasa.gov/display/GIBS/GIBS+API+for+Developers) managed by NASA's

Earth Observing System Data and Information System (EOSDIS) have been developed for visualization of the MOD10_L2 dataset together with other Earth observation datasets.

6. Additional corrections are applied to create the daily (MOD10A1) and eight-day (MOD10A2) composite snow cover products from the MOD10_L2 dataset. These datasets consist of 1200 km by 1200 km tiles of 500 m resolution data gridded in a sinusoidal map projection (Hall et al. 2006b). The MOD10A1 and MOD10A2 datasets are available online for download in the HDF-EOS format through FTP (ftp://n5eil01u.ecs.nsidc.org/SAN/MOST/MOD10A1.005). Similar to other MODIS datasets, they can also be downloaded through the NASA Reverb ECHO interactive download website. Several stand-alone software tools such as the MODIS reprojection tool (Dwyer and Schmidt 2006) and software developer libraries such as PyMODIS (Delucchi 2014) have been developed to simplify and automate data access to the MOD10A1 and MOD10A2 data products. However, these datasets are not currently available through a WCS, WMS or WMTS web service interface.

This study specifically uses the processed MODIS Terra Snow Cover imagery layer from the NASA EOSDIS GIBS web service that is based on the reprojected MOD10_L2 data product. This web service has available data in 500 m resolution with temporal range from May 2012 until present, published and described at: https://wiki.earthdata.nasa.gov/display/GIBS/GIBS+Available+Imagery+Products.

2.1.2 WMTS Data Extraction Design

Extracting time series data from the WMTS consists of four main steps: (1) construct the uniform resource locator (URL) address of a tile image that overlaps the point of interest at a given time step, (2) download the tile image, (3) find the coordinates of the pixel corresponding

to the point of interest inside the downloaded image, and (4) convert the pixel color to the observation units. These four steps are repeated for each time step in the time series (Figure 3-1).

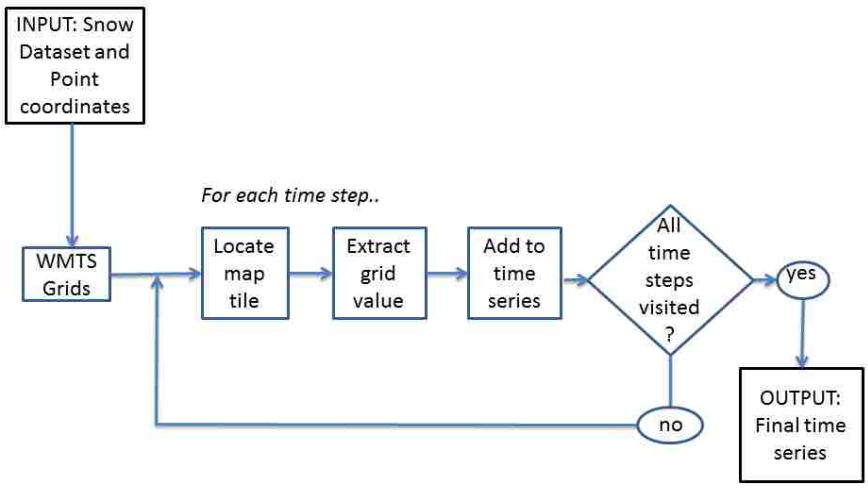

Figure 3-1 Snow Data Retrieval Function Loop for a Point Location

The WMTS specification, also known under the name "TMS" (tile map service), is a web service protocol for retrieving tiled map images which has been defined by the Open Geospatial Consortium (Masó et al. 2010). The WMTS request is in the format:

[server]/[layer]/[time]/[x]/[y]/[zoom].png **(3-1)**

where x is the Tile x (horizontal) index and y is the Tile y (vertical) index.

The zoom number (typically between 1 and 20) indicates the scale of the map. It must not be higher than the maximum supported zoom level of the WMTS web service. Given the latitude, longitude and zoom number, the tile x index and tile y index can be calculated as:

$$x_{tile} = \frac{(lon+180)}{360} * 2^{zoom} \quad \textbf{(3-2)}$$

$$y_{tile} = \frac{1-log\left(\tan(lat\_rad)+\frac{1}{\cos(lat\_rad)}\right)}{2\pi} * 2^{zoom} \quad \textbf{(3-3)}$$

where lon represents longitude in decimal degrees, lat_rad represents latitude in radians, zoom is the zoom number, xtile is the x column (x) index, and ytile is the tile row (y) index. The integer part of the xtile or ytile number is the tile x or tile y index. The fractional part is the position of the pixel inside the tile.

Once the pixel value is retrieved, it needs to be converted to the snow cover category. The category is represented as a unique color in the image. If the image is in the .png or .gif file format, the color information is often stored as "indexed color" to reduce the file storage. In indexed color the raw value of each pixel is a number (typically in the range 0-255), and the color legend is stored in a color table section of the file. Knowing the color legend of the WMTS web service and the color table, a lookup mapping the raw pixel values to the snow categories can be created. Table 3-1 shows an example lookup table for the IMS Snow Mapping web services. There are 5 categories: Snow-Free, Snow, Ice, Cloud, and No Data represented by the raw values 0, 1, 2, 3, and 255. Other WMTS services define a larger number of categories. For example the MODIS - Terra Daily Snow Cover has the categories: "cloud", "no data", and a raw value between 0 and 100 to represent the fractional snow cover percent inside the pixel.

Table 3-1 Example Lookup Table for Converting Raw Pixel Values to Snow Categories

Raw Pixel Value	Color (R, G, B)	Transparency	Snow Category
0	Transparent (255, 255, 0)	0	Snow-free
1	White (255, 255, 255, 255)	255	Snow
2	Blue (102, 255, 255, 255)	255	Ice
3	Gray (224, 224, 224)	255	Cloud
255	Transparent (0, 0, 0)	0	No Data

The lookup table, the latitude and longitude coordinates, the list of times, and the URL template of the WMTS web service are the required inputs of the WMTS snow retrieval function. If the WMTS data have a regular time step, then the begin time and the end time can be supplied instead of the list of times. The snow retrieval function uses a loop. For each time step, the URL of the tile is constructed. A web request is issued to the WMTS server, which returns an image in response. The raw pixel value is extracted from the image, and the lookup table is used to convert the raw pixel value to the snow category (percent snow cover in pixel). Finally, the snow category value is saved to an ordered list of (time, value) pairs. The output list can be saved to a file or database, or passed to other functions for time series visualization.

The script can be implemented as a function in the Python programming language. By using Python, the script can be executed on multiple operating systems for both desktop and server. The script requires a library for efficiently extracting a pixel value from an image. In Python I used the open source library pypng (http://pythonhosted.org/pypng; for reading png format).

2.1.3 Tethys Framework

To make the Python data extraction functionality available to end users, I designed an interactive web application in the Tethys framework. Tethys (Jones et al. 2014) is a platform that can be used to develop and host water resources web applications. It includes a suite of free and open source software: PostgreSQL, GeoServer, and Web Processing Service. One advantage of Tethys over a custom web application is that it provides a plug-in architecture (Heineman and Councill 2001) for relatively easy implementation of new applications. The web page graphical style, layout, navigation bar, user management, and data storage management are all handled by

Tethys. A Tethys plugin is known as “app” and it has four components: Persistent Storage, Controller, Template, and Gizmo. The persistent storage is used for storing user settings, for example the favorite location. The controller handles computations and application business logic. Tethys provides a base controller with built-in functionality for connecting to the model. The template defines the position of text, buttons, charts, tables and maps in the user interface. The Gizmo is a customizable user interface component. It can be a button, date time picker, drop-down list, chart, table, or map. The map and chart template are responsible for display of an interactive map and chart on the app’s landing page.

2.1.4 Snow Inspector User Interface Design

The Snow Inspector web user interface consists of two views: “snow map” and “snow graph”. The map uses the OpenLayers version 3 (openlayers.org) interactive map control. The user can add a point anywhere on the map by mouse click. When the user finishes working with the map, the coordinates of the user defined shapes are passed to the map controller. The map controller passes the coordinates and other input parameters (time range, WMTS service) to the snow controller. The snow controller launches the data retrieval script. As the data retrieval script is working, it has the option to save the downloaded data in a persistent storage time series cache. Using the Asynchronous JavaScript (AJAX) method, the map and chart template periodically checks the snow controller for progress status and updates the chart and a progress bar on the web page. Once the processing is finished, the controller retrieves the time series and, passes it to the chart template where the interactive chart is updated (Figure 3-2). The snow controller also includes a function to retrieve the pixel boundaries and pixel values for a specified area and date in the GeoJSON format.

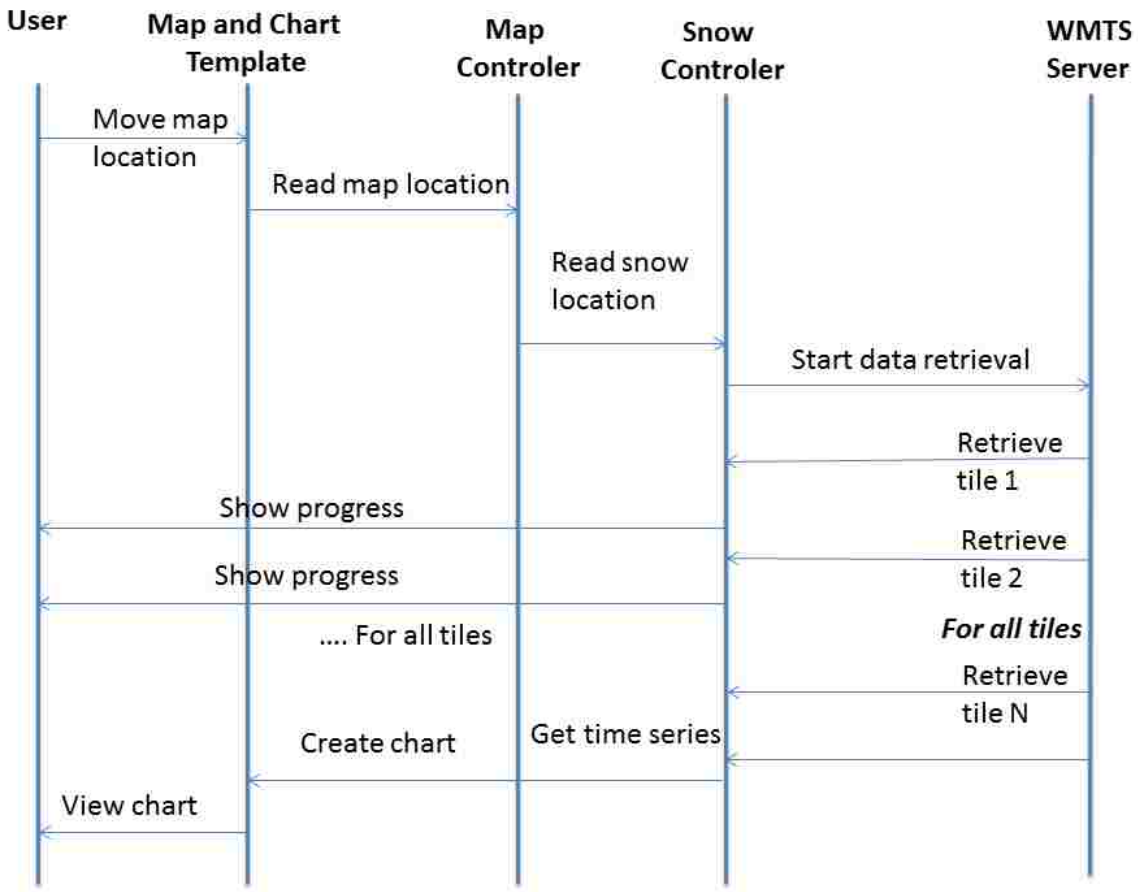

Figure 3-2 Interaction of User, Template and Controller in the Snow Inspector Application

2.1.5 Snow Inspector Data API Design

Four other templates are used for showing the result time series in different formats: A comma-separated values (CSV), JavaScript Object Notification System (JSON), WaterML 1.1 and WaterML 2.0 template. WaterML is an Open Geospatial Consortium (OGC) internationally approved standard for exchanging hydrological time series data (OGC 2012a; Valentine et al. 2012). It contains not only the data values, but also the associated metadata including the site location, data source organization, and measurement units. The overall design of the Snow Inspector web application including the controller and the templates is shown in Figure 3-3.

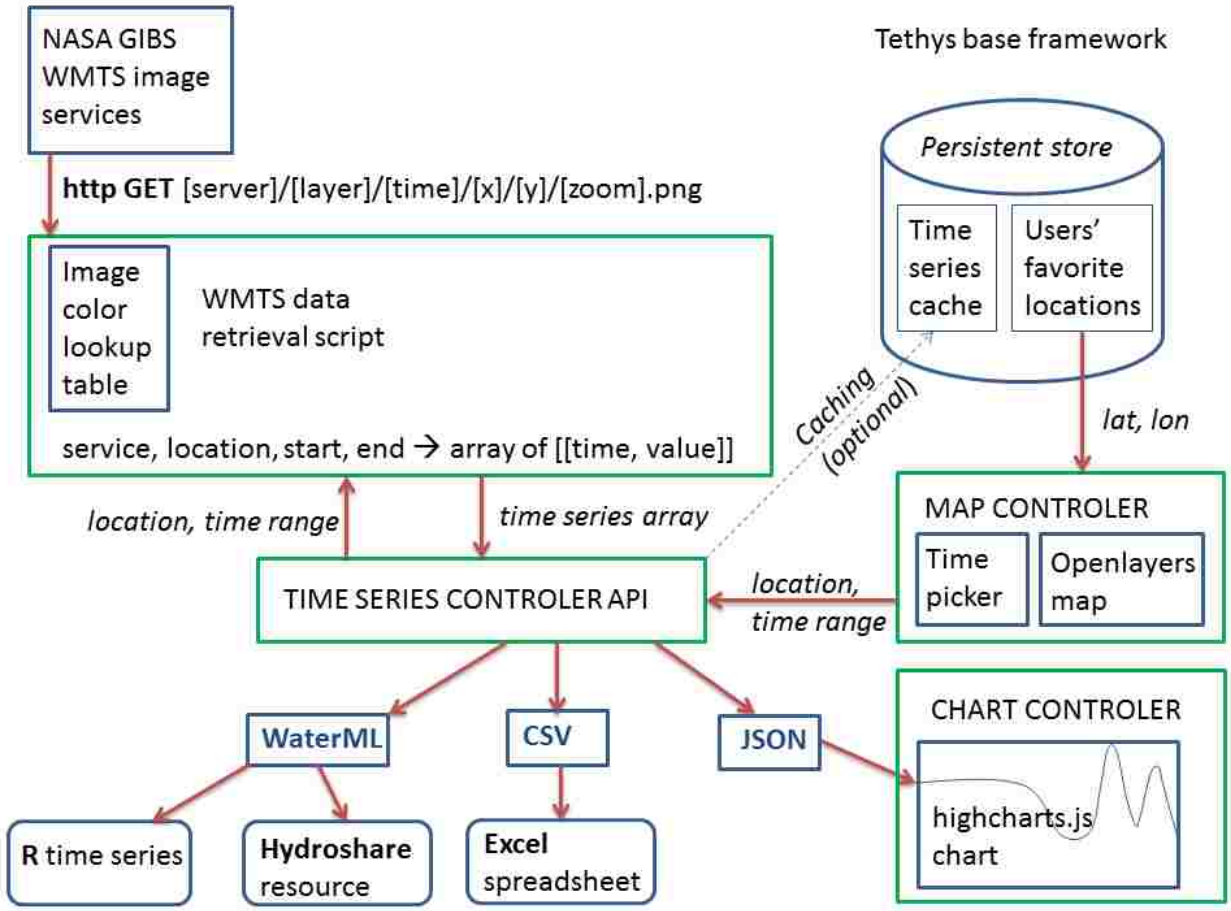

Figure 3-3 Architecture of the Snow Inspector Web Application Showing Exchange of Data Objects between the Components

The time series API can be accessed using the following query string parameters added to the URL (Table 3-2).

Table 3-2 Parameters of the Snow Inspector API for Retrieving Time Series

Parameter	Explanation
latitude	Latitude in decimal degrees
longitude	Longitude in decimal degrees
start	Start date in yyyy-mm-dd format
end	End date in yyyy-mm-dd format
format	The output format: csv, WaterML1, WaterML2, or json

For example, requesting the URL: http:/apps.hydroshare.org/apps/snow-inspector/waterml/?lat=49.69095&lon=15.98980&start=2014-10-01&end=2015-05-31 retrieves a time series of percent snow cover from the MODIS Terra WMTS pixel at the location (49.69095 N, 15.98980 E) from October 2014 until May 2015. This web request can be used in scripting software such as R or Python to automate the retrieval of snow cover time series for multiple points of interest.

2.1.6 Performance and Usability Testing

For the usability of web application, the loading speed of the web page is an important factor. I tested the time required to complete loading of the snow cover time series using the following steps:

(1) I selected 200 random time period lengths between 1 day and 1000 days. Although the MODIS images go back until the year 2000, the GIBS WMTS service for snow cover has only been available since May 2012 and the older data are not yet available through this web service, therefore I could not use a longer time period.

(2) I used spatial random sampling to choose 200 random sample sites located on land in the Northern hemisphere (between 30°North and 70°North), and I assigned one of the random time period lengths to each random site. For each selected site and time period, I issued a web request to retrieve the snow data and create the time series graph. For each sample I also registered the number of cloudy days and number of snow-free days, to check if this factor has any effect on the data retrieval speed.

(3) To test possible effects of caching on performance, I examined the differences in response time seen between initial requests and three subsequent requests.

2.1.7 Comparison of Estimated Snow to Ground Truth Observations

To test the validity of the results of the Snow Inspector application and script, I conducted a ground truth experiment involving the collection of over 100 ground observations in and around Yellowstone National Park. This study area was chosen because of the relative ease of access to a wide variety of terrain and land cover regions (i.e. no private land barriers to access). Data were collected in the following manner: 1) Potential data collection locations were identified in the office using a combination of GIS data and fore-knowledge about accessible locations; 2) An individual traveled to each location previously selected and visually surveyed the immediate vicinity; 3) An area roughly 600m by 600m was examined and an estimate of percent snow cover was made; and 4) This information was recorded together with any other unique or distinguishing characteristics of the observation location such as tree cover information and tree canopy thickness Finally, once all manual data observations were made, I determined the Snow Inspector value for percentage coverage at each of the data collection locations and computed metrics indicative of the Snow Inspector accuracy.

2.2 Results

The following section describes the results of performance and usability testing of the Snow Inspector web application. This application can be used for retrieving MODIS snow cover time series from any point on Earth. Results of the ground validation experiment using the Snow Inspector maps and time series data in the Yellowstone National Park are also presented.

2.2.1 Web Application Deployment Results

The name of the web application is “Snow Inspector”. It is available for registered users at the website http://apps.hydroshare.org. The source code is distributed under the MIT license on

the GitHub repository (http://github.com/jirikadlec2/snow-inspector). When a change is made in the source code, the administrator of the server http://apps.hydroshare.org can log-in to the server and run a command that retrieves the source code from the repository and updates the app.

The Snow Inspector user interface has two parts: Snow Map and Snow Graph. In the Snow Map, the user can define a point location on the map (Figure 3-4) by clicking on the map or by entering the coordinates User can also define the date and the length of the time period. When the map is shown at a large scale, the boundaries of the reprojected MODIS satellite pixels for the selected date are shown, and a number showing the MODIS fractional snow cover inside the pixel (%) can be displayed (Figure 3-4).

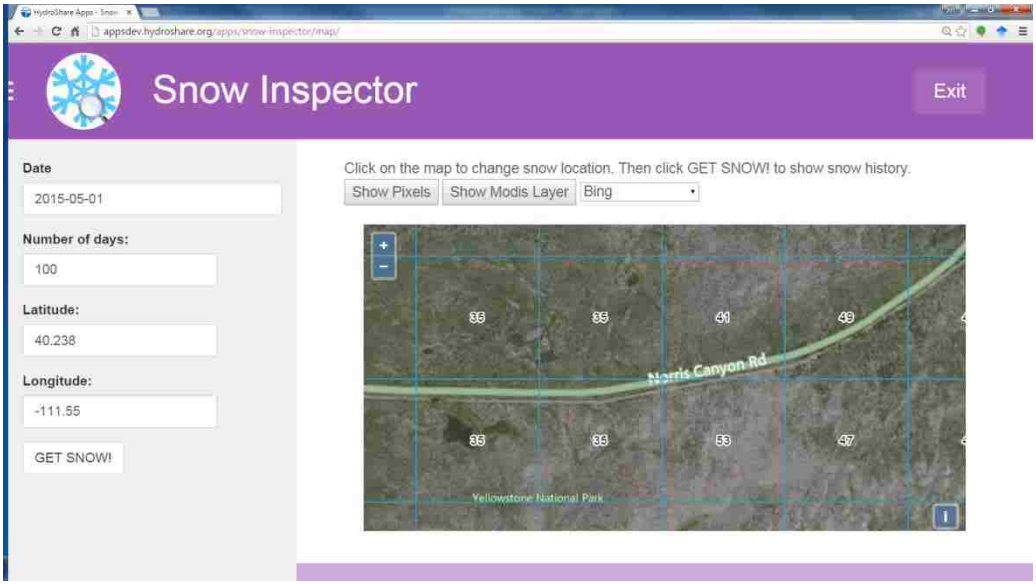

Figure 3-4 Snow Map Input Page with User Interface for Selecting the Point of Interest or Entering Latitude and Longitude Coordinates

The satellite pixel overlay corresponds to the selected date. Once the locations have been selected, the program sends multiple requests to the WMTS snow cover web service. The time series chart is shown in the Snow Graph screen. While the data are being retrieved and

processed, a progress indicator is shown in the snow graph area. Selecting a data point in the time series chart displays the original processed MODIS Terra snow cover image, allowing the user to inspect the data in more detail (Figure 3-5).

Figure 3-5 Snow Coverage Graph at a Selected Point

2.2.2 Performance and Usability Testing Results

In web-based application, the users expect a response in less than one second. The retrieval of multiple images, depending on network connectivity, may require more time. As a first step I added a "loading" animated image indicator that appears while the process is running. As a second step, I implemented a progress bar. The snow retrieval script saves the retrieved values from every time step to the application's database table. By counting the rows in the database table, I know how many images have been processed and how many are remaining. Every five seconds, the web client user interface can post an AJAX request to a controller. The controller

checks the number of complete table rows and returns it to the client following which the client updates the progress bar. The second challenge is handling response errors in the WMTS service. Occasionally I found that there was a 5-second or longer delay in retrieving one single image. To eliminate this delay, I calculated the average image retrieval time and the standard deviation of that time. If one request exceeds the average retrieval time by more than three standard deviations, the retrieval of data for this time step is skipped and the user is notified that there is an error on the MODIS Terra WMTS server or an error in internet connection and therefore data for the time step is unavailable.

To test the response time for different scenarios and to find the average retrieval time, I used spatial random sampling to choose 200 random sites on the land in the northern hemisphere (Figure 3-6). I have found that there is a linear relationship between the number of days requested and the total retrieval time (Figure 3 7). The linear regression equation is:

$$Retrieval\_time = -0.3523 + 0.0578 * number\_of\_days \quad (3\ 4)$$

On average, it took 5.4 seconds to retrieve 100 days of data

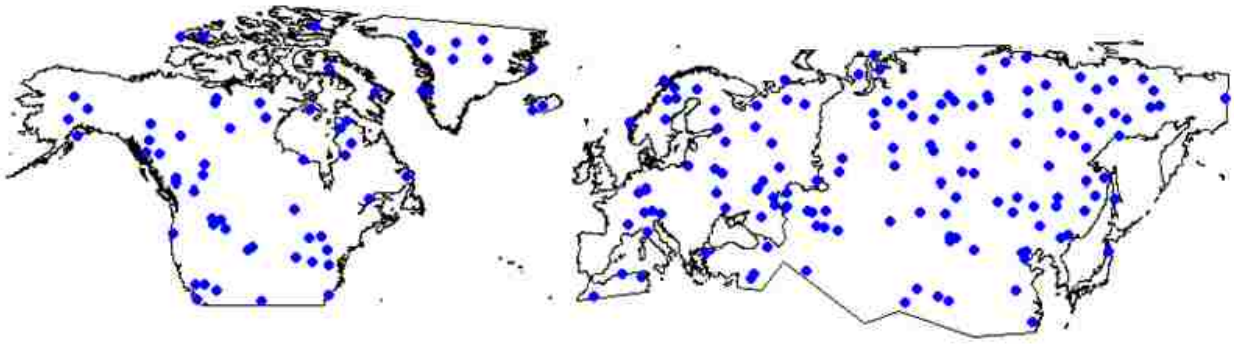

Figure 3-6 Spatial Random Sample of 200 Locations for Testing Response Time

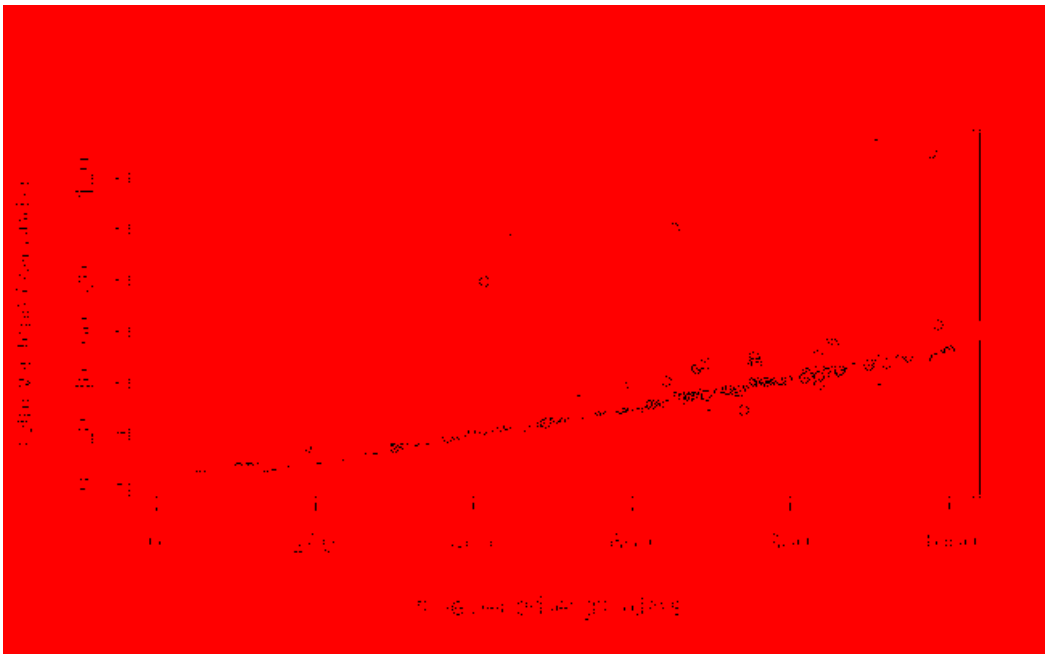

Figure 3-7 Linear Regression Scatter Plot for Number of Requested Days versus Retrieval Time

To examine the possible effect of server-side caching, I also performed four subsequent requests at all randomly selected sites using a 1000 day time series. Figure 3-8 shows the presence of a caching effect.

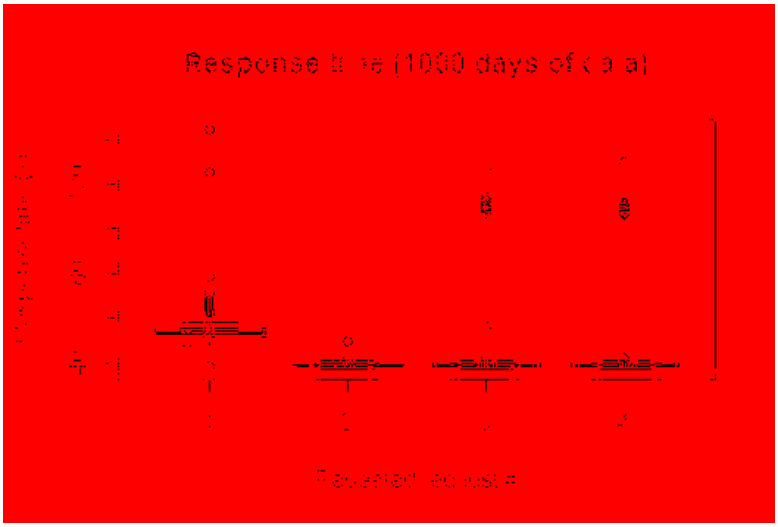

Figure 3-8 Response Time for Repeated Requests

The mean response time of the first request was 56 seconds, which was greater than the mean response time of the second subsequent request (39 seconds). I also noticed a slight increase in the response times of the third and fourth subsequent request, indicating a possible expiration of the server side cache. Considering that the download of a 90 day time series of original MODIS Terra MOD10_L2 raster data takes several minutes, using the Snow Inspector presents a significant speed improvement in the access to multi-temporal MODIS Terra percent snow cover time series.

2.2.3 Ground Validation Results

To verify the results of the Snow Inspector tool, a ground survey was conducted. On May 2 and May 9, 2015, Woodruff A. Miller visited and surveyed 36 sites in the Yellowstone National Park area (Figure 3-9). The Snow Inspector was used to identify the MODIS satellite pixel boundaries at a 1:10,000 scale (see example on Figure 3-10). For each location, between 1 and 8 pixels were identified and the percentage of snow-covered area on the ground was estimated. Locations with open ground, partially tree-covered ground, and ice-covered lakes were selected.

To examine the effect of trees on fractional snow cover estimates on partially tree-covered ground, tree canopy thickness was identified as “dense” or “sparse”. The total number of ground validation pixels was: 102 pixels on May 2, 2015, and 82 pixels on May 9, 2015. Out of the total 184 pixels, 28 pixels were inspected both on May 2 and on May 9. The complete ground validation dataset is available online in the public HydroShare repository at: https://www.hydroshare.org/resource/05a29e4ddffd42b88f482bd5be46e88d/.

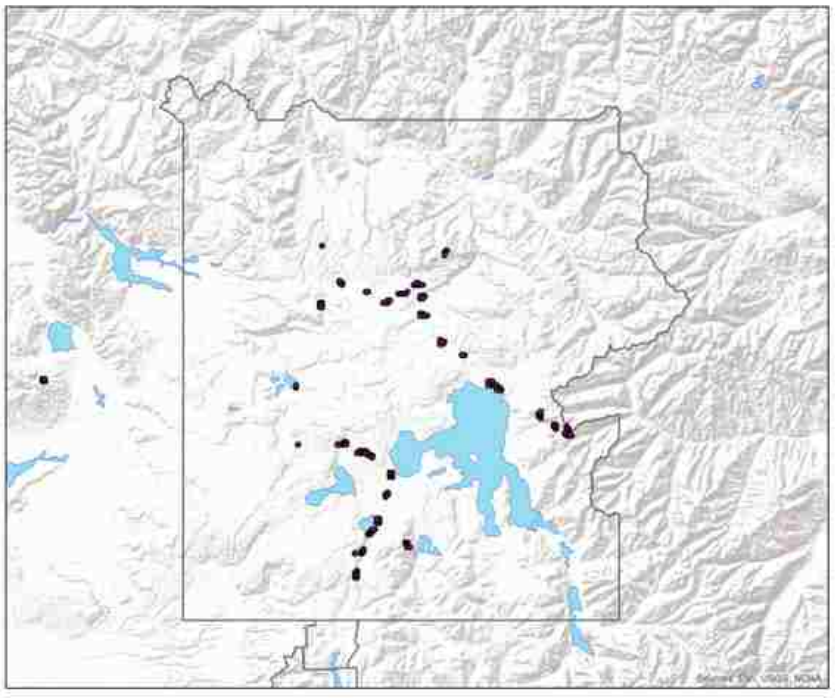

Figure 3-9 Map of Visited Sites in Yellowstone National Park

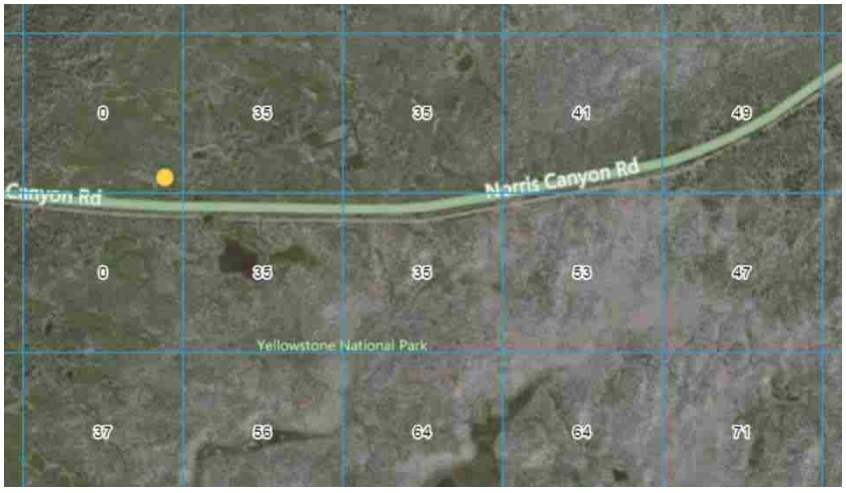

Figure 3-10 Example Validation Site: MODIS Gridded Pixel Boundaries with Aerial Photo Map Background

On May 2, 2015, 94% of the inspected locations were cloud-free. On the previous day, May 1, 2015, 100% of the locations were cloud-free. Figure 3-11 shows the comparison of the May 2, 2015 ground observations with the May 1 and May 2 satellite observations and highlights differences in type of land cover.

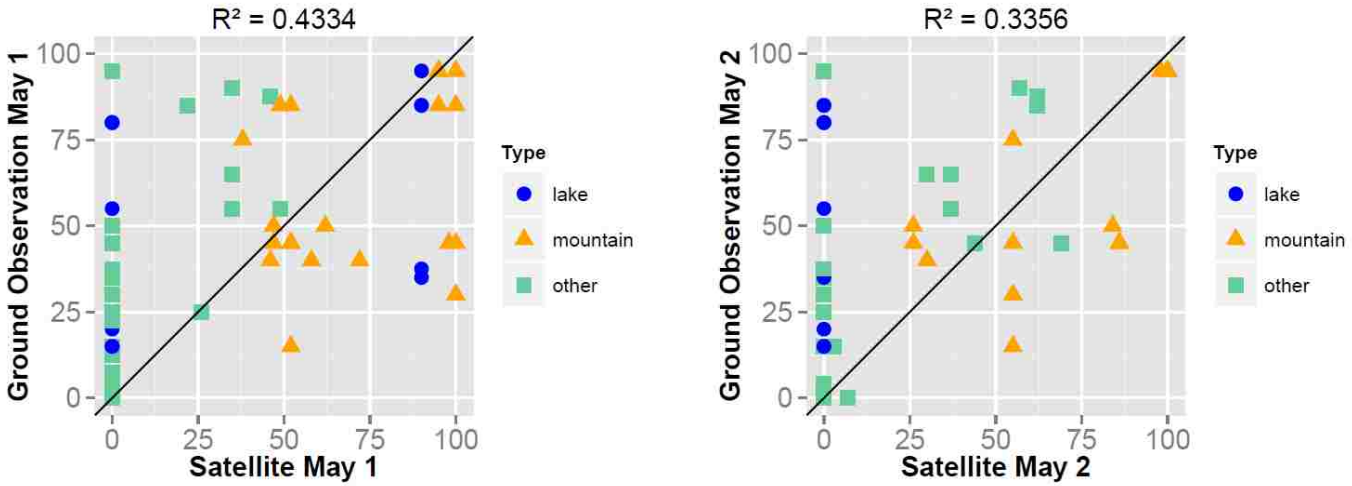

Figure 3-11 Comparison of Ground and Satellite Percent of Snow-Covered Area in the Pixel on May 1 and May 2, 2015 at Selected Locations in Yellowstone National Park

The comparison shows a large number of sites where the satellite underestimated the observed snow cover. According to ground validation, the pixel was partially snow-covered, but no snow was detected by the satellite. It is noticeable that the majority of the pixels where no snow was detected by the satellite had less than 50% observed snow covered area on the ground. While Figure 3-11 shows good agreement for mountain sites above the tree line, it also shows significant underestimation and large day-to-day variation for snow on the ice-covered lakes. The large day-to-day variation on the ice-covered lakes could be explained by the start of an ice melting event that occurred in the first week of May 2015 (Niemi et al. 2012; Xin et al. 2012).

A closer look at the pixels with no detection of snow by the satellite revealed that these pixels had patchy snow and were located in areas of evergreen forest. Tree canopies partially obscure the MODIS view of the ground, so the satellite observed fraction of snow may be less than the true snow fraction (Rittger et al. 2013; Simic et al. 2004). Additionally tree shadows may further obscure the satellite view, especially in cases of large zenith angle (Niemi et al. 2012; Xin et al. 2012). For the "other" pixels (not lake or mountain above tree line), the remaining pixels were divided in two categories to explore the effect of tree canopy thickness (Figure 3-12).

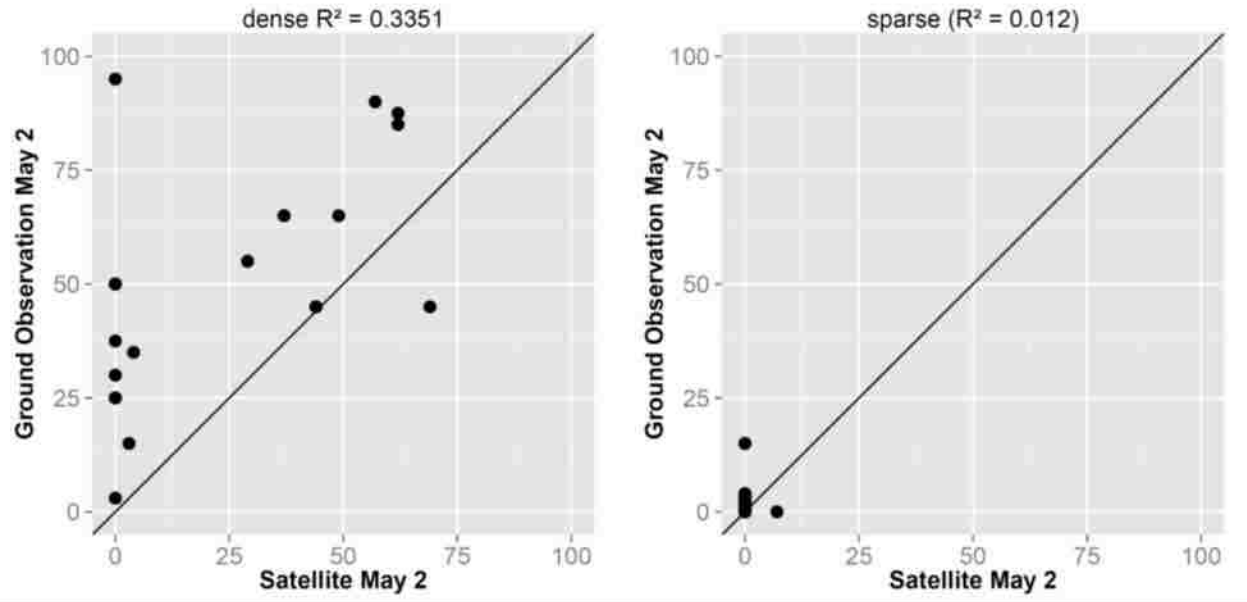

Figure 3-12 Comparison of Ground and Satellite Percent of Snow-Covered Area in the Pixel on May 2 for Pixels with Dense (Left) and Sparse (Right) Tree Cover

The "dense" category labels areas of live evergreen forest with generally thick tree canopy. The "sparse" category labels areas with thin tree cover, such as recently burned areas. As shown in Figure 3-12, the satellite generally underestimated snow cover in pixels with both dense and sparse tree cover. The satellite failed to detect patchy snow (with less than 15% observed snow

on the ground) in sparse tree cover areas. Similar observations where the MODIS sensor failed to detect patchy snow in forested areas in the melting season were made by Parajka et al. (2012).

On the second visit on May 9, 2015, the satellite reported cloud cover for the majority of the sites. Therefore to get the satellite estimate of snow cover, I used two methods: (1) Nearest available previous date, (2) Nearest available next date. A similar approach like the two methods chosen is described by Hall et al. (2010) and Gao et al. (2010). They recommended using nearest non-cloud observations from prior days for filling data gaps in the snow cover product. Figure 3-13 shows the comparison of ground and satellite observations using the "nearest available date" method.

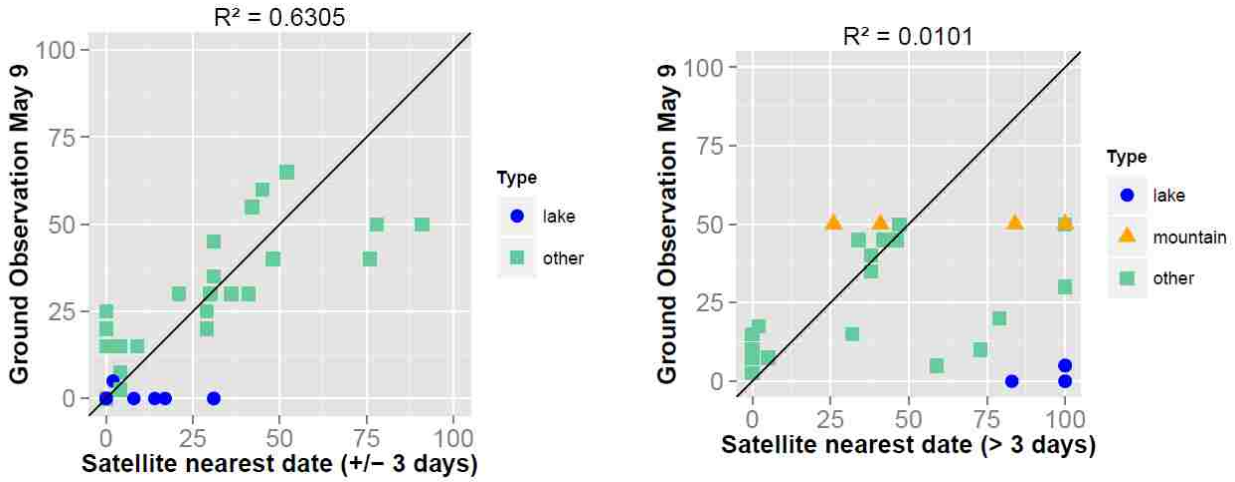

Figure 3-13 Comparison of Ground and Satellite Percent of Snow-Covered Area in the Pixel on May 9, 2015 at Selected Locations in Yellowstone National Park with Cloud-Free Satellite Data Available within 3 Days before or after Ground Observation (Left) or within More than 3 Days before or after Ground Observation (Right)

The nearest available cloud-free dates were between 0 and 7 days before or after the May 9, 2015 visit. Figure 3-13 shows separately the pixels where cloud-free data were available within 3 days and within 4 – 7 days of the ground observation.

Similar to the first visit, there were a number of sites where the satellite underestimated the snow cover percentage. However, the proportion of sites with above-zero snow cover fraction on the ground and zero satellite snow cover fraction is smaller than in the first visit on May 2, 2015. The satellite also overestimated the percent of snow cover on lakes. This could be partially explained by the ice melt event that may have occurred between the last available cloud-free date and the ground observation date. The false identification of open water lakes as snow has also been noticed by Hall and Riggs (2007) especially if the water has high turbidity or if it is shallow with a bright bottom. As shown in Figure 3-13, the agreement between satellite and ground observation was much better for pixels with cloud-free data available within three days of the ground observations. For the time lag of four days or longer, there were cases when the satellite overestimated snow cover possibly due to snow melting in the period between the satellite and the ground observation. The same effect is apparent for tree-covered pixels, where a better match is seen for pixels with the shorter time lag (see Figure 3-14 and Figure 3-15).

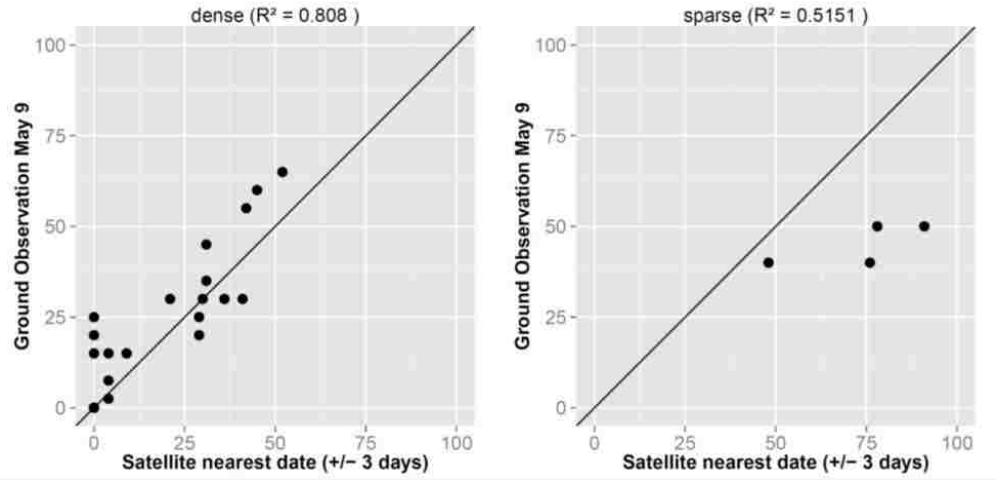

Figure 3-14 Comparison of Ground and Satellite Percent of Snow-Covered Area in the Pixel on May 9 for Pixels with Dense (Left) and Sparse (Right) Tree Cover, Using Pixels with Three Days or Less Time Lag Between Ground and Satellite Observation

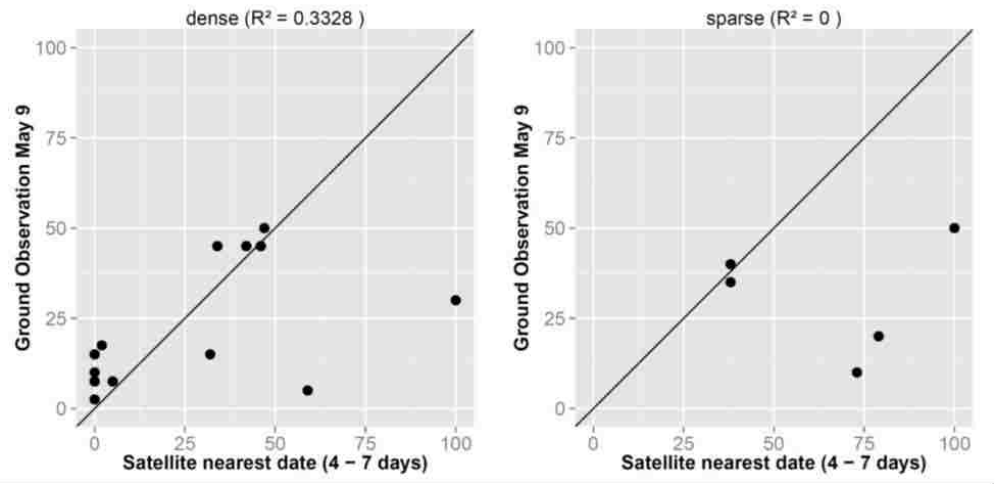

Figure 3-15 Comparison of Ground and Satellite Percent of Snow-Covered Area in the Pixel on May 9 for Pixels with Dense (Left) and Sparse (Right) Tree Cover, using Pixels with 4 - 7 Day Time Lag Between Ground and Satellite Observation

Another method of ground validation is the "presence" and "absence" of snow. Here each ground and satellite observation was labeled as "presence of snow" if more than 50% of the ground was snow-covered. Table 3-3 and Table 3-4 show the confusion matrix of observed versus predicted values on May 2 and May 9, 2015. It is evident from these tables, that the specificity (true negatives) is higher than the sensitivity (true positives), indicating that snow-free ground is detected by the satellite more successfully than snow-covered ground. Table 3-5 shows the Percent Correctly Classified (PCC) indicator for the 50% snow cover threshold.

Table 3-3 Comparison of Ground and Satellite Snow Covered Area on May 2, 2015

	Ground		
Satellite		Snow	No Snow
	Snow	13	9
	No Snow	19	61

Table 3-4 Comparison of Ground and Satellite Snow Covered Area on May 9, 2015

	Ground		
Satellite		Snow	No Snow
	Snow	10	8
	No Snow	12	52

Table 3-5 Percent Correctly Classified (PCC)

Date	**May 2, 2015**	**May 9, 2015**
PCC	0.73	0.76

The results of the ground survey suggest that rapid melting of snow occurred in the Yellowstone National Park in the first week of May 2015. One application of the Snow Inspector API is finding the approximate snowmelt date. For this application the "pixel-borders" function in the API can be used. This function has the lonmin (minimum longitude), latmin (minimum latitude), lonmax (maximum longitude), latmax (maximum latitude), and date (the examined date) parameters and it returns the MODIS pixel boundaries together with the fractional snow cover number for each pixel for the specified date. For each pixel a time series of fractional snow cover in the pixel can be obtained, showing the snow melt date. The exact date of transition of the pixel from partially snow covered to snow-free might not be known due to cloud cover. However, the last available date with presence of snow and the first available date with absence of snow can be found from the time series. The following R code documents getting the snow data for each pixel in the Yellowstone National Park area for a selected date:

```
north = 45.13
south = 44.02
east = -109.25
west = -111.17
api_uri = "https://appsdev.hydroshare.org/apps/snow-
inspector/pixel-borders/"
uri = paste0(api_uri, "?lonmin=",west, "&latmin=", south,
"&lonmax=", east,"&latmax=", north, "&date="2015-01-01")

#download geoJSON
download.file(base_uri, "pixels.GeoJSON")
pixels = readOGR("pixels.GeoJSON", layer="OGRGeoJSON)
centroids <- coordinates(pixels)
pixelvals = as.numeric(pixels$val)
```

The approximate snow melt date is in the interval between these two dates. Figure 3-16 shows a map of the Yellowstone National Park for the snow melt season (March – June) of 2015.

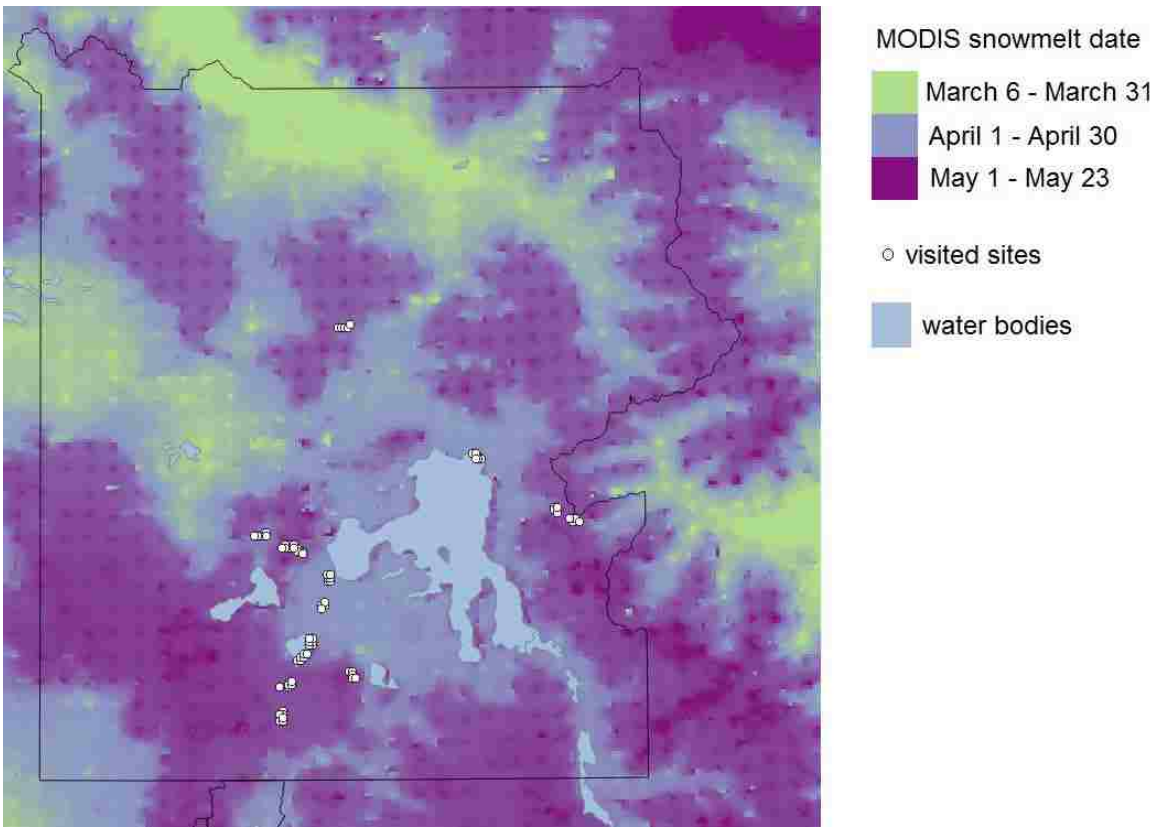

Figure 3-16 Approximate Date of Snowmelt in 2015 in the Yellowstone National Park according to MODIS Satellite Data

The northern section of the national Park (Lamar Valley) already lost snow in March. The central sections of the park around Yellowstone Lake became snow-free during the course of April, while in the high elevation areas in the southern and eastern parts of the park the main snowmelt occurred in May. A part of the visited sites is located in the region with snow melting between 1st and 31st May. This is consistent with the quick changes in the snow-covered area that were observed in the 2nd May and 9th May field survey.

2.3 Discussion

The goal of the Snow Inspector application was to design a fast and simple web-based interface for retrieving time series information regarding the presence or absence of snow that can be accessed from anywhere in the world. It is accessible on the website apps.hydroshare.org. The extracted time series data and the associated metadata are available not only through a user interface, but also in the CSV and WaterML format. Therefore it can be used to develop third-party applications that require time series information about snow cover. The WaterML output for locations of interest can also be shared on public repositories such as HydroShare (Horsburgh et al. 2015). According to preliminary testing, the data retrieval duration is between 5 and 30 seconds for a 90 day period of interest, which is faster than downloading and opening a series of large raster files through FTP from the original MODIS/Terra MOD10_L2 data source. This response time could be further reduced by making multiple parallel requests to the WMTS web service. A limitation is that only point-based data retrieval is supported. For applications in hydrology, it would be useful to add polygon-based retrieval functionality to get the percentage of a watershed covered by snow.

Ground validation of MODIS fractional snow cover data provided through the Snow Inspector during the 2015 snowmelt season in the Yellowstone National Park indicated a good accuracy for open ground (above tree line). However, the values shown by the Snow Inspector for lakes did not match well with ground observations. Therefore I do not recommend using this application for retrieving time series of lake ice cover.

A large portion of the inspected pixels had the presence of trees. In some cases, the presence of snow was not detected in forested areas (omission error) possibly due to patchy snow, tree shadow effect, or obscured satellite field of view. Similar omission errors in evergreen forest in the snowmelt season were documented by Parajka et al. (2012), Rittger et al. (2013) and Xin et al. (2012). When the MODIS data were not available on the same day due to cloud cover, I found that satellite values from a previous or subsequent day could be used. However, the accuracy was limited when the time lag exceeded three days. One use of the Snow Inspector time series view is showing the nearest cloud-free observations for the selected pixel. Implementing this feature also in the map view could further improve the application's usability for snow cover validation activities.

Compared to other MODIS snow cover data distribution methods, the Snow Inspector has limited period of record and limited data accuracy. It only provides time series data since May 2012, while the original MODIS data go back to year 2000. The data are based on the MODIS swath MOD10_L2 product. The MOD10_L2 swath product becomes available online within 3 hours of observation, making it suitable for near real time monitoring. However, it has limited accuracy compared to the standard daily MOD10A1 daily snow cover product. Using the swath MOD10_L2 is complicated by the so-called "bowtie effect" of the MODIS instrument, which causes an overlap of the satellite field of view, producing a data repetition. This effect increases

with the distance from nadir and can be especially dramatic at the edge of the image (Gómez-Landesa et al. 2004; Ren et al. 2010). Several studies therefore recommend using the MOD10A1 daily or the MOD10A2 8-day data products for multi-temporal evaluation of changes in snow covered area (Gascoin et al. 2014; Hall and Riggs 2007; Parajka et al. 2012). In addition, it is possible that extra inaccuracies in the data may have been introduced during the re-gridding from the original MODIS swath to the WMTS spherical Mercator projection grid. For future studies it would be interesting to quantify these inaccuracies by comparing the WMTS output to the original MOD10_L2 swath and the processed MOD10A1 datasets. The reason why I used the less accurate MOD10_L2 swath product in this study is that it was the only MODIS snow product readily available through a standard WMTS web service interface. For future studies, I recommend publishing the more accurate MOD10A1 and MOD10A2 data product through a WMTS, WMS or WCS web service and making them accessible for online interactive snow cover time series extraction tasks.

After the initial deployment of the application, several changes were requested by the users and added to the application. The users requested adding more details to the landing page about the source of the data, credit for its generation, and information on how to cite it. The time series shown by the Snow Inspector are the result of a chain of many derived data sets and data products. Therefore, special attention was paid to include correct and complete the citation and provenance information about the data origin and data processing steps in the metadata in the WaterML data download link. These changes simplified using the Snow Inspector application for MODIS ground validation activities. For any inspected location, the pixel boundaries with satellite-based snow cover values can be shown in the map and overlaid with an aerial

photograph or a topographic map layer. This detailed view assists in showing sub-pixel land cover variability and locating satellite pixel extent in the field.

Currently the Snow Inspector application supports access to only one web service: The MODIS Terra daily snow cover data product published as WMTS on the GIBS web service. However it is possible to customize the data extraction technique and user interface to work with any other multi-temporal hydrologic or climate dataset that provides a WMTS or WMS web service interface. This customization would require setting up a lookup table associating each image color with an observed value or category, and finding out the pattern of how the time parameter is encoded in the WMTS tile URL or in the WMS GetMap request. In the case of WCS, the procedure for downloading multiple images and extracting pixel values could be replaced by a single web request to the WCS to extract a time slice for a selected pixel. Indeed I view the WCS as the recommended protocol for distributing spatio-temporal data such as snow cover fields. However, not all WCS currently support the time slice subsetting requests and only a limited number of publishers provide a WCS interface to access their snow data. With a growing need for data interoperability, I expect that more public snow cover datasets including the IMS and SNODAS will be published in one of the standard web service format (WCS, WMS or WMTS), enabling rapid access and comparison of hydrological and climate time series for any point on Earth.

3 USING CROWDSOURCED AND STATION DATA TO FILL CLOUD GAPS IN MODIS SNOW DATASETS

Maps and time series datasets of snow-covered area, snow depth, and snow water equivalent are critical for water resources management (Mankin et al. 2015), climatology (Derksen and Brown 2012), transportation (Bica et al. 2012), and outdoor recreation (Ghaderi et al. 2014) in snow-dominated regions. The cross-country skiing community, in particular, has high interest in detailed snow coverage maps. For example in Czechia in 2015 there were 2305 km of designated cross-country skiing routes (OpenStreetMap 2015), all of which relied on sufficient snow cover to be navigable. The existence of such extensive snow recreation trail networks creates an interesting opportunity to use crowd-sourced snow observation data to improve large scale spatial estimates of snow cover needed for water management and hydrologic applications.

A number of observation and mathematical modeling methods are commonly used to estimate spatial snowpack extent. These methods include measuring snow depth or snow water equivalent with manual or automated sensors (Pohl et al. 2014), detecting snow physical properties through remote sensing (Metsämäki et al. 2012; Rees 2005), and calculating snow water balance using hydrological models (Anderson 1973; Koivusalo et al. 2001; Lehning et al. 2006). The temporal and spatial continuity of spatially distributed estimates of snow-covered area (SCA) are limited by the availability of cloud-free satellite imagery (Molotch et al. 2004). Passive microwave and radar methods can detect snow at night and on cloudy days, however

their spatial resolution is limited due to large field size required to gather the electromagnetic energy at longer wavelengths (Dietz et al. 2012). Furthermore, reconstructing SCA and snow water equivalent (SWE) using water balance models is limited by availability and errors in the snow accumulation and snowmelt forcing inputs (Slater et al. 2013).

A key part of snow cover estimation – particularly when using remotely sensed data – is filling in spatial gaps in snow cover maps caused by cloudy conditions. Existing gap filling methods include use of neighboring time steps (Hall et al. 2010), spatial-temporal filtering based on neighboring pixels (Parajka and Blöschl 2008; Yang et al. 2014), cubic spline time series interpolation (Dozier et al. 2008), and fusion of multiple remote sensing datasets (Foster et al. 2011; Ramsay 1998; Sirguey et al. 2008; Tait et al. 2000). A sequence of multiple spatial-temporal filtering steps is recommended to completely fill the cloud cover gaps (Gafurov and Bárdossy 2009). Information reconstruction via machine learning methods has also been used to automate the cloud removal process (Chang et al. 2015).

To develop a continuous snow extent dataset, continuous ground station observations or other ancillary information is often used together with the discontinuous remote sensing observations. An example of this approach is the regional snowline method (Parajka et al. 2010), where cloud-free pixels are used to obtain the mean snowy pixel (snowline) and mean snow-free pixel (landline) elevations. The cloud-covered pixels are then reclassified as snow-free, snow-covered, or partially snow-covered by comparing their elevation to the snowline and landline elevations. The regional snowline method provided best results when at least 15 % of the study area was cloud-free. Molotch et al. (2004) used air temperature measured at a high density station network, and the accumulated degree-day index to define snow-free areas under clouds. Spatially distributed, physically-based or conceptual snowpack models have also been used to

model SCA under cloud (Rodell and Houser 2004; Zaitchik and Rodell 2009). A review of spatial interpolation methods used for interpolation of snow depth measured at points is given by López‐Moreno and Nogués‐Bravo (2006) and Li and Heap (2014). Manual interpretation of satellite images (more than 15 geostationary and orbital satellite sensors) and ground station measurements has been used by the U.S National Snow and Ice Data Center (NSIDC) to produce global daily-updated maps in the Interactive Multi-Sensor Snow and Ice Mapping System (IMS) at 4 km resolution (Ramsay 1998). The IMS operational product is continuously enhanced by including new data sources including new satellite sensors, ground sensors and snowpack models (Helfrich et al. 2007).

Volunteer geographic information (VGI) is increasingly being used in real-time mapping such as flood mapping (Schnebele et al. 2014), routing navigation (Bakillah et al. 2014), detecting land use changes (Jacobson et al. 2015) and in mapping of air quality (Mooney et al. 2013; Reis et al. 2015). According to Reis et al. (2015) the mobile phone owners can be considered as smart sensors producing ubiquitous data, which can be integrated with existing environmental models. Several studies have explored the potential of crowdsourcing snow and other weather related data from online social network sources. For example, snow depth related messages from the Twitter network have been used to update a real-time snow depth map of the U.K (Muller 2013). In a review of crowdsourcing approaches for gathering meteorological data, Muller et al. (2015) classify such methods as passive (an automated sensor operated by a volunteer and connected to a social network) or active (the volunteer actively connects and uploads data to a central hub). Wang et al. (2013) successfully tested the detection of snow on public photographs on the Flickr social network website. This experience indicates that VGI

from snow-sports related social networks represents a potentially highly informative dataset for updating continuous snow cover maps.

The goal of this study is to design, develop, and test an algorithm for updating probability maps of snow cover using publicly available crowdsourcing data from multiple data sources, including online VGI repositories. The paper is organized as follows: First I describe the volunteer data sources, and how to retrieve and organize these sources. Next I discuss automated data fusion methods for generating the snow cover maps. Finally, I run a validation of the models in the Czechia region of Europe, test the contribution of VGI (crowdsourcing) data to snow map accuracy, and also present an online interactive snow map extent application based on this approach.

3.1 Material and Methods

The following section introduces the study area and the snow data sources: ground stations, MODIS snow cover datasets, volunteer snow reports, Strava GPS tracks, and Garmin GPS tracks. An interpolation method for generating snow probability maps is described. A validation method for evaluating map accuracy is presented. The software architecture and user interface design for making the generated maps available online are also discussed in this section.

3.1.1 Study Area

Located in central Europe region between 12°5 and 18°50 E, 48°33 and 51°3 N, Czechia (Czech Republic) falls into the humid continental climate zone according to the Köppen climate classification (Peel et al. 2007). The elevation range is between 115 and 1600 m (Figure 4-1).

The average number of days with snow cover is less than 40 days in the lowest elevations and up to 160 days in the highest mountains. According to Bednorz (2004), Czechia falls within the active region with snow-cover probability of between 10 and 90% during the winter season and the snow pack is intermittent in most of the area – typically the snowy period is interrupted several times. There is a large year-to-year variation in the extent and duration of snow-cover, influenced by atmospheric circulation patterns. A large and statistically significant negative correlation has been noted between the number of days with snow cover and the North Atlantic oscillation index (Bednorz 2004). As a region with a long tradition of snow recreation, high population density, and fast changes in the spatial and temporal extent of the snow, Czechia is an ideal study area for exploring the potential of crowd-sourced data in snow mapping.

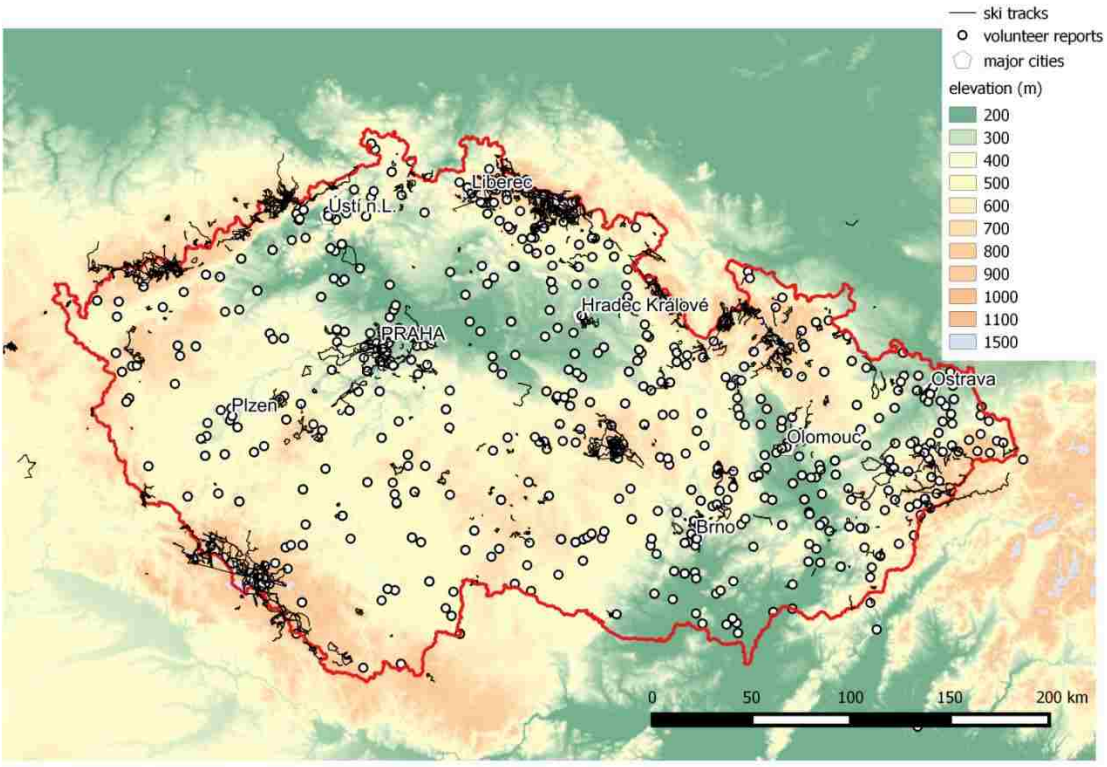

Figure 4-1 All Publicly Available Cross-Country Skiing Tracks (Garmin and Strava) and Volunteer Snow Report Locations from the Period 2013 - 2015

Currently snow depth maps are published by the Czech Hydrometeorological Institute (CHMI) on the website: http://portal.chmi.cz/files/portal/docs/poboc/OS/OMK/mapy/. These maps are updated daily and they are created using a two-step process with geographically weighted regression (elevation and snow depth), followed by inverse distance weighted (IDW) interpolation of the residuals (Stříž 2011). The maps are provided in the form of an image with pre-defined snow depth categories. This work can serve to improve these maps through the integration of multiple data sources as described below.

3.1.2 Data Sources – Reports, Tracks and Stations

For the purposes of this study I collected VGI point observations of snow presence and absence from an online open access web service provided by InMeteo Ltd. Located at http://in-pocasi.cz. The methods presented here can be generically applied to any similar repository of VGI snow coverage data. The in-pocasi.cz web service provides an online form for submitting a snow observation location, snow depth, and text comments. The user can also send the report by text message from a cell phone. The reports for any selected date are accessible using a web address in the form www.in-pocasi.cz/pocasi-u-vas/seznam.php?historie=M-D-Y where M is the month, D is the day, and Y is the year. During the two-year period 1/2013 – 4/2015 a total of 10,156 reports were submitted and approved. These reports are fairly consistently provided throughout the week (Table 4-1) but are significantly more common in January and February than other winter months (Table 4-2). VGI snow reports are also stratified by elevation with most reports occurring in the 200m to 500m elevation range and near population centers (Figure 4-1). The average elevation of the report locations is 410 m (slightly lower than the average elevation of the study area), and 90% of the reports are within the 115 – 617 m elevation range (Figure 4-2).

Table 4-1 Reports and Ski Tracks Distribution by Day of Week

Day of Week	Reports (% of total)	Ski Tracks (% of total)
Sunday	15.9	25.6
Monday	14.3	7.9
Tuesday	15.0	10.0
Wednesday	13.2	9.8
Thursday	13.2	9.1
Friday	13.7	10.6
Saturday	14.7	27.0

Table 4-2 Reports and Ski Tracks Distribution by Month

Month	Reports (% of total)	Ski Tracks (% of total)
November	4.5	0.5
December	14.8	17.8
January	36.0	33.5
February	26.5	33.8
March	17.8	12.5
April	0.5	1.8

Figure 4-2 Elevation of Ski Tracks, Volunteer Reports and Stations

The skiing community uses location-enabled mobile devices and online websites to share information about cross country ski trips and associated snow conditions. Popular social networks used by cross-country skiers include Strava (www.strava.com) and Garmin Connect (www.connect.garmin.com). Using a Global Positioning System (GPS) receiver on the user's mobile device, the route and velocity of each ski trip is recorded and uploaded to an online database. If the user marks the trip as "public", the recording of the trip route can be viewed by anyone connected to the internet. Both Strava and Garmin Connect provide an application programming interface (API) for searching and downloading public routes by geographical area, time range, keyword, and type of activity. For this study, I used the Strava and Garmin Connect APIs to query all routes with activity type marked as "backcountry ski" or "cross-country ski" in Czechia and neighboring border regions in the period 2013 - 2015.

All public cross-country skiing tracks retrieved from the Garmin and Strava websites during the time period of 2012 – 2015 are shown as a data layer in Figure 4-1. Initially some of the recorded GPS tracks contained errors (long straight lines), which were likely the result of switching off the device and activating it again in a different location before saving the track. To remove the straight line artifacts, I applied a quality control filter that deleted straight track segments that were 1 km or longer. A closer look at the cross-country ski tracks also revealed some tracks following a major road for a long distance. These cases were likely a result of the users continuing GPS recording of a ski trip after entering a motor vehicle. To filter out the motor vehicle sections, I removed track segments with speed greater than 50 km/h. Looking at spatial distribution of the tracks, one can note a higher concentration of tracks in the vicinity of the two largest cities (Praha, Brno) located to the east and to the west of the center of the country (Figure 4-1).

The highest density of ski tracks is found in mountainous regions above 1000 m elevation (Figure 4-2), with 90% of the tracks located above 600 m elevation. This is considerably higher than the mean elevation of the study area (433 m). Unlike in the volunteer snow report data, there is a distinct weekly fluctuation in the number of recorded tracks with a maximum on Saturdays and a minimum on Mondays (Table 4-1). The number of recorded cross-country skiing tracks was highest in January and February (Table 4-2). In the time period 2012 – 2015 the total number of recorded tracks depended on the winter season: The number of tracks decreased in 2014, but again increased in 2015 possibly due to changes in snow-covered area (Figure 4-3).

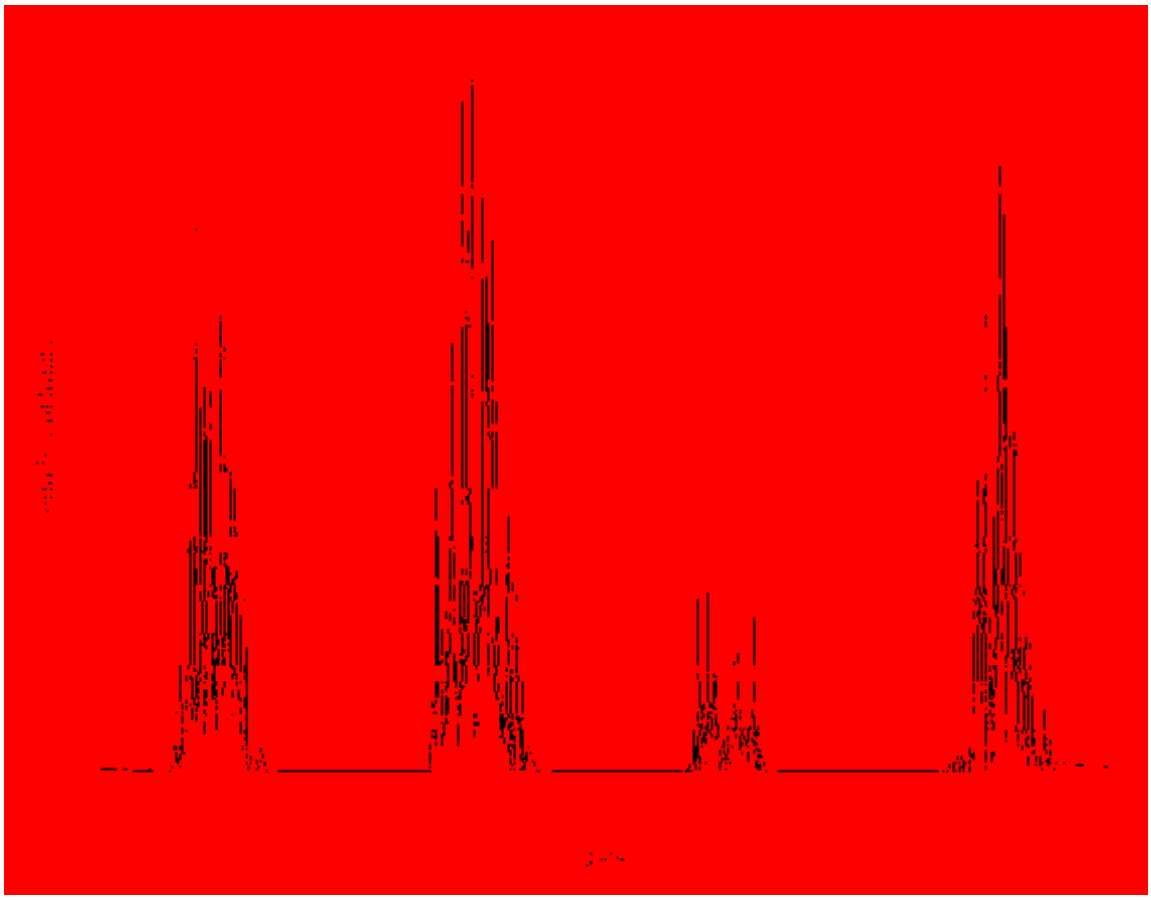

Figure 4-3 Number of Cross Country Skiing Tracks (from Garmin Connect) per Day

The Czech Hydro-meteorological Institute (CHMI) and the local watershed management authorities operate a network of professional synoptic and climatological stations in Czechia. Daily snow depth measurements from up to 56 of the stations are published on the Internet (Figure4). Historical measurements (2006 – 2015) from these selected stations are also available in the WaterML data exchange format using a public web service API at: http://hydrodata.info/chmi-d/ (Kadlec and Ames 2011). The station network is evenly distributed with samples in all elevation zones (Figure 4-4). However, only a part of the stations had data available for the whole period of interest (2012 – 2015), and there were frequent data gaps at some of the stations.

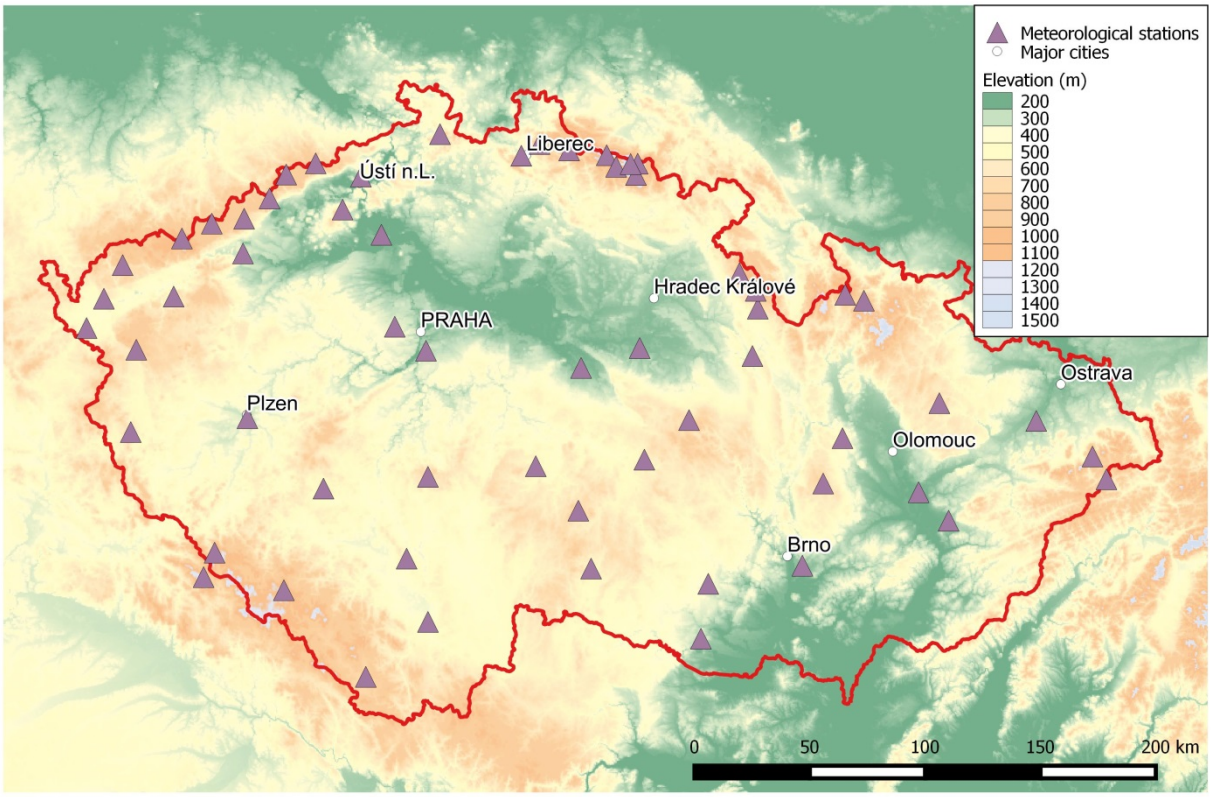

Figure 4-4 Meteorological Stations with Publicly Available Snow Measurements during the Period 2013 – 2015

3.1.3 Data Sources: MODIS Satellite Snow Cover Maps

The Moderate Resolution Imaging Spectroradiometer (MODIS) is a multispectral satellite sensor on board the Terra and Aqua polar orbiting satellites. After applying a cloud mask, the Normalized Difference Snow Index (NDSI) is used to identify pixels as snow and to calculate fractional snow cover in each MODIS pixel. The detailed procedure is described by Hall et al. (2002). In this study I use the web coverage service (WCS) from the CryoLand website (www.cryoland.org) to retrieve daily MODIS Terra fractional snow cover (MOD10A1) raster datasets covering the study area region. These raster data have been transformed to the WGS 1984 UTM zone 33 coordinate system in a horizontal resolution of 500 meters. At a daily time step resolution, MODIS provides the most detailed observation of snow on the ground in the study area on cloud-free days. However, the presence of cloud cover in the winter months makes large parts of the study area invisible to the satellite optical sensor for most of the time. In the winter months (November – April) of 2012 – 2015 the mean cloud cover was 74% of the study area. As shown in Table 4-3, the area was nearly cloud-free (< 20% cloud cover) on 9.7 % of the winter days. On 58.9 % of the days, it was nearly overcast (> 80% cloud cover). Especially at lower elevations of Czechia, an inversion associated with low stratus cloud forms in the enclosed basin, and may last for many days of the winter season (see Figure 4-5).

Table 4-3 Frequency of Cloud Cover in the Winter Season (November - April 2013 - 2015) in the Study Area According to MODIS data

Cloud Cover (%)	Frequency (%)
0 - 20	9.7
20 - 40	6.1
40 - 60	9.8
60 - 80	15.5
80 - 100	58.9

The MODIS datasets occasionally contain isolated pixels (patches of one or two pixels) classified as “snow”. These pixels are typically considered as noise in the classification. Based on the recommendations of (Gafurov and Bárdossy 2009) I used a 3 x 3 moving window noise reduction filter with majority (mode) function to preprocess each MODIS raster for further analysis.

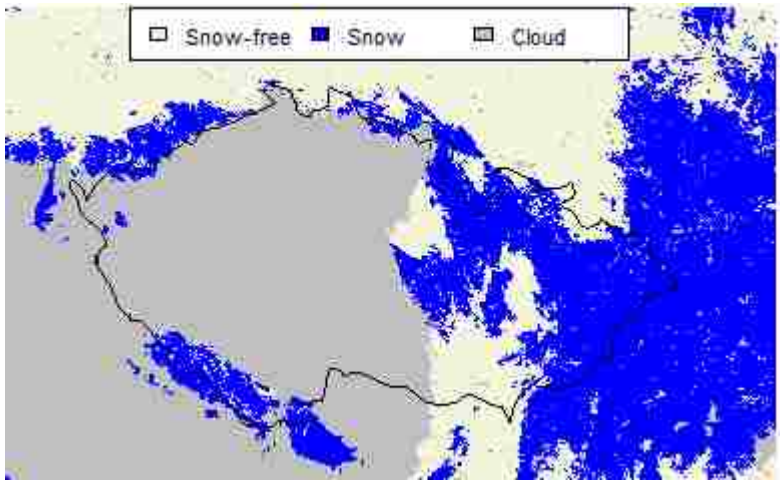

Figure 4-5 Example of Inversion Low Cloud Situation (9th February 2015)

3.1.4 Interpolation Method

Our interpolation method begins by re-projecting to UTM Zone 33, each of the quality-controlled input datasets (satellite snow raster, tracks, stations, volunteer report points). In the next step each dataset is classified in two categories: present and absent. For stations and volunteer snow reports the value is labeled as “present” if the reported snow depth is greater or equal to 1 cm. Reports of snow dusting or discontinuous snow are labeled as “absent”. All of the cross-country ski tracks are marked as “present”. If bicycle tracks are available from Strava or Garmin Connect for the selected date, the bicycle tracks are marked as “absent” tracks. For the

MODIS MOD10A1 snow dataset, a pixel is marked as "present" if the satellite fractional snow cover is greater than 50%. Pixels with cloud or other categories (open water, ice) are marked as "no data". The next steps of the interpolation are shown in Figure 4-6.

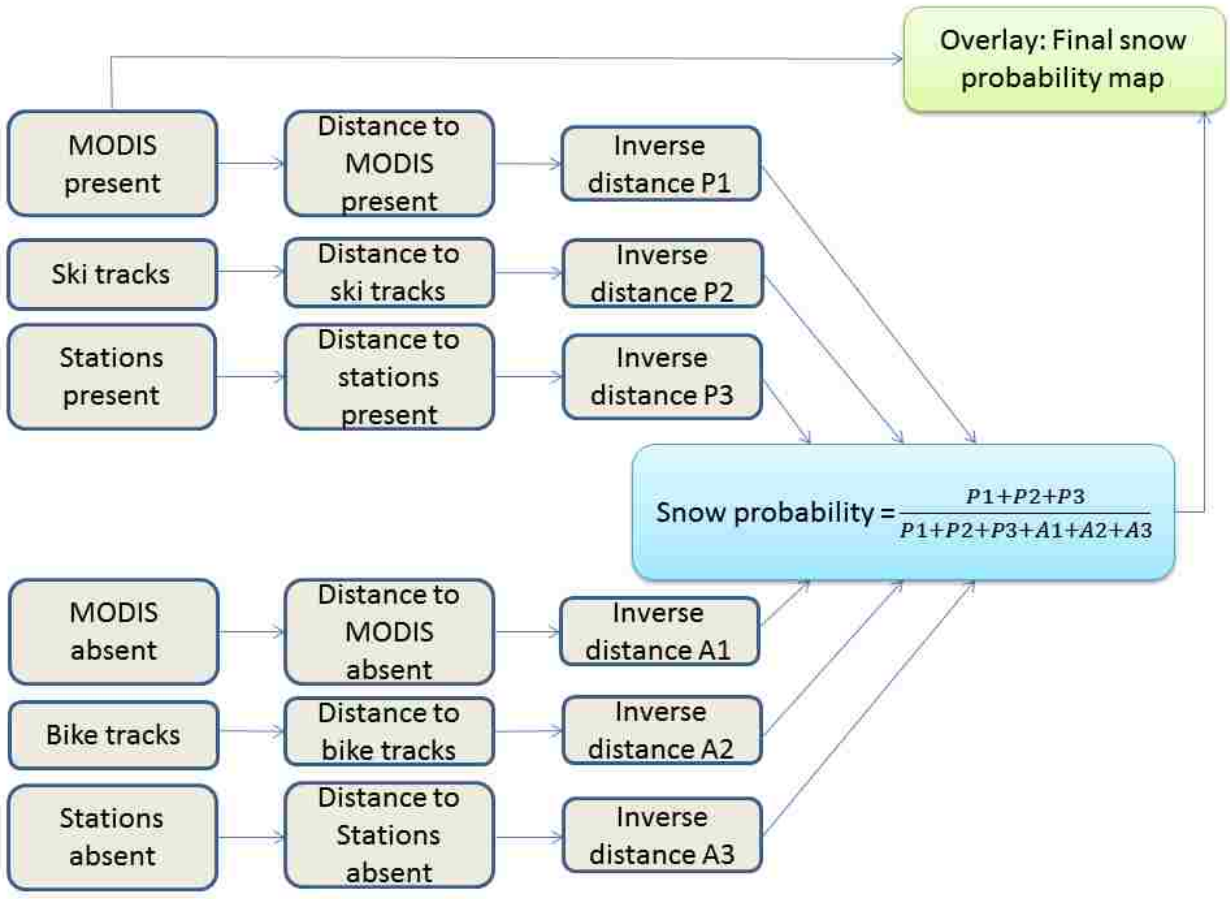

Figure 4-6 Interpolation Steps to Create Snow Probability Map

Figure 6 shows the interpolation workflow in the case where the MODIS satellite raster, stations, and tracks are available. Additional reclassified input data sources (such as reports – present, reports – absent) can be added to the schema if they are available for the selected date. For each pixel in the study area, a distance to the nearest point of each "present" and "absent" dataset is calculated. The distance value is related to the influence of each snow observation on

its surrounding area. In this study I examined two different distance calculation methods. The first method uses the horizontal Euclidean distance and elevation-weighted least-cost path distance. The horizontal distance is calculated as:

$$\boldsymbol{distance_i} = \sqrt{\left(\boldsymbol{x_i} - \boldsymbol{x_p}\right)^2 + \left(\boldsymbol{y_i} - \boldsymbol{y_p}\right)^2} \quad \textbf{(4-1)}$$

where x_i, y_i are the coordinates of the center of each pixel and x_p, y_p the coordinates of the nearest point of the nearest dataset. At the scale of the study area, the horizontal distance value in the UTM projection is similar to the great circle distance, and therefore I did not consider Earth curvature in the distance calculation.

The elevation weighted least cost distance takes into account topography and terrain configuration in the area (Van Etten 2012). For each raster cell I define a conductance value. The conductance is the inverse to the cost of passing through a cell or between neighboring cells. When searching for optimal path (least-cost path), a route through cells with high conductance is preferred. For each cell the conductance can be calculated using a transition function based on the cell and its neighbors:

$$\text{Conductance} = 1 / \text{elevation}^p \quad \textbf{(4-2)}$$

where elevation is the mean elevation of a nine-cell neighborhood and p is an exponent for weighing the importance of elevation. For elevation of cells in the study area I used a digital elevation model (DEM) dataset from the Shuttle Radar Topography Mission (SRTM) with 500 meter cell size. The elevation weighted least cost path distance is calculated from source cells (cells with snow observation) to destination cells (all other cells of the study area) using the

gdistance R software package (Van Etten 2012) to produce a cost surface with an elevation-weighted distance at each destination raster cell.

In the next step the confidence (inverse distance) is calculated as:

$$\mathbf{conf_i} = \frac{\mathbf{1}}{\boldsymbol{distance^p}} \quad \textbf{(4-3)}$$

where $conf_i$ is the inverse distance at each pixel, and p is the inverse distance exponent. I chose the exponent value p=3 (rather than the typical value 2 used in inverse distance weighted interpolation) to assign greater influence to values closest to the present or absent snow dataset.

The resulting inverse distance raster datasets are combined to create a snow probability map as follows:

$$\boldsymbol{snow\ probability} = \frac{\sum_{i=1}^{n} \boldsymbol{conf_{present\ i}}}{\sum_{i=1}^{n} \boldsymbol{conf_{present\ i}} + \sum_{j=1}^{m} \boldsymbol{conf_{absent\ j}}} \quad \textbf{(4-4)}$$

where confpresent $_i$ is the inverse distance to a "present" dataset, n is the number of "present" inverse distance datasets, confabsent j is the inverse distance to an "absent" dataset, m is the number of "absent" inverse distance datasets. For example, if four different presence and absence datasets are available, then the snow probability would be calculated as:

$$\boldsymbol{snow\ probability} = \frac{\boldsymbol{P1+P2+P3+P4}}{\boldsymbol{P1+P2+P3+P4+A1+A2+A3+A4}} \quad \textbf{(4-5)}$$

where P1 is the inverse distance to snow-covered MODIS pixels, P2 is the inverse distance to ski tracks, P3 is the inverse distance to stations with snow, P4 is the inverse distance to reports with snow, A1 is the inverse distance to snow-free MODIS pixels, A2 is the inverse distance to non-

ski tracks, A3 is the inverse distance to snow-free stations, and A4 is the inverse distance to reports without snow.

To create a final map, the snow probability map is overlaid with the MODIS satellite raster. Snow-covered MODIS pixels are assigned the value of 1, and snow-free MODIS pixels are assigned the value of 0. The remaining MODIS pixels with no data (cloud-covered pixels) are assigned the calculated snow probability value. Using a threshold of 0.5, the entire study area can be classified as “Snow” or “No snow”.

3.1.5 Validation

For validation of the created snow maps, I selected two random samples of 10 days with small cloud cover (< 25% cloud) and 10 days with high cloud cover (> 75% cloud). For each of the 10 cloud-free days, I combined the original MODIS raster with the cloud layer from each of the cloudy days. I used the inverse distance method described above to predict snow probability inside the cloud-covered area, and compared the predicted probability with the observed (cloud-free) snow presence or absence values. As a result, I obtain 100 comparisons (10 cloud-free days * 10 cloudy days). The model predicts a probability (between 1 and 0) and the validation cloud-free pixels show an occurrence (either 1 or 0). To measure model performance, I used the following indicators: Commission Error, Omission error, Percent Correctly Classified (PCC), and Area under the curve (AUC). These indicators are based on the confusion matrix (Table 4-4).

Table 4-4 Confusion Matrix

	Observations		
Predictions		Present	Absent
	Present	True Positive	False Positive
	Absent	False Negative	True Negative

The commission error indicates the overestimation of snow cover. It is calculated as:

$$Commission\ Error = False\ Positive / (False\ Positive + True\ Positive) \quad \textbf{(4-6)}$$

The omission error indicates the underestimation of snow cover. It is calculated as:

$$Omission\ Error = False\ Negative / (False\ Negative + True\ Negative) \quad \textbf{(4-7)}$$

The PCC indicator is calculated as:

$$\boldsymbol{PCC = \frac{true\ positive + true\ negative}{N}} \quad \textbf{(4-8)}$$

where N is the total number of validated points (pixels inside study are that were cloud-free on the validation day). To count the true positives and true negatives for the PCC, I need to define a threshold value. The threshold is the occurrence probability above which I consider the value as "present". In the validation method I used the default threshold value of 0.5.

To examine the effect of the threshold in more detail, I also used another indicator: the Area under the Curve (AUC). The AUC is extracted from the Receiver Operating Characteristic (ROC) by plotting the true positive rate (sensitivity), against the false positive rate (1-specificity). The area under the curve is indicator of the model performance (Metz 1978). It has possible values between 0 and 1. The AUC value of 1 would be the best possible prediction with 100% sensitivity (no false negatives) and 100% specificity (no false positives).

3.1.6 Software Design

To make the resulting snow extent maps available to the winter sports community, a web-based software system was designed. The system has four main components: (1) Snow map

generation script, (2) Snow web processing service, (3) Snow web map service, (4) Snow map user interface. I have built these components using the Tethys platform (Jones et al. 2014) and HydroShare (Horsburgh et al. 2015). Tethys is an open-source software platform for developing interactive web-based hydrologic applications. It is built using existing open source web frameworks including Django, OpenLayers and GeoServer based on recommendations by Swain et al. (2015). HydroShare is a social network for publishing and viewing hydrology-related data sets that provides free space for storing large amounts of multidimensional data. A general schema of the software architecture is shown in Figure 4-7.

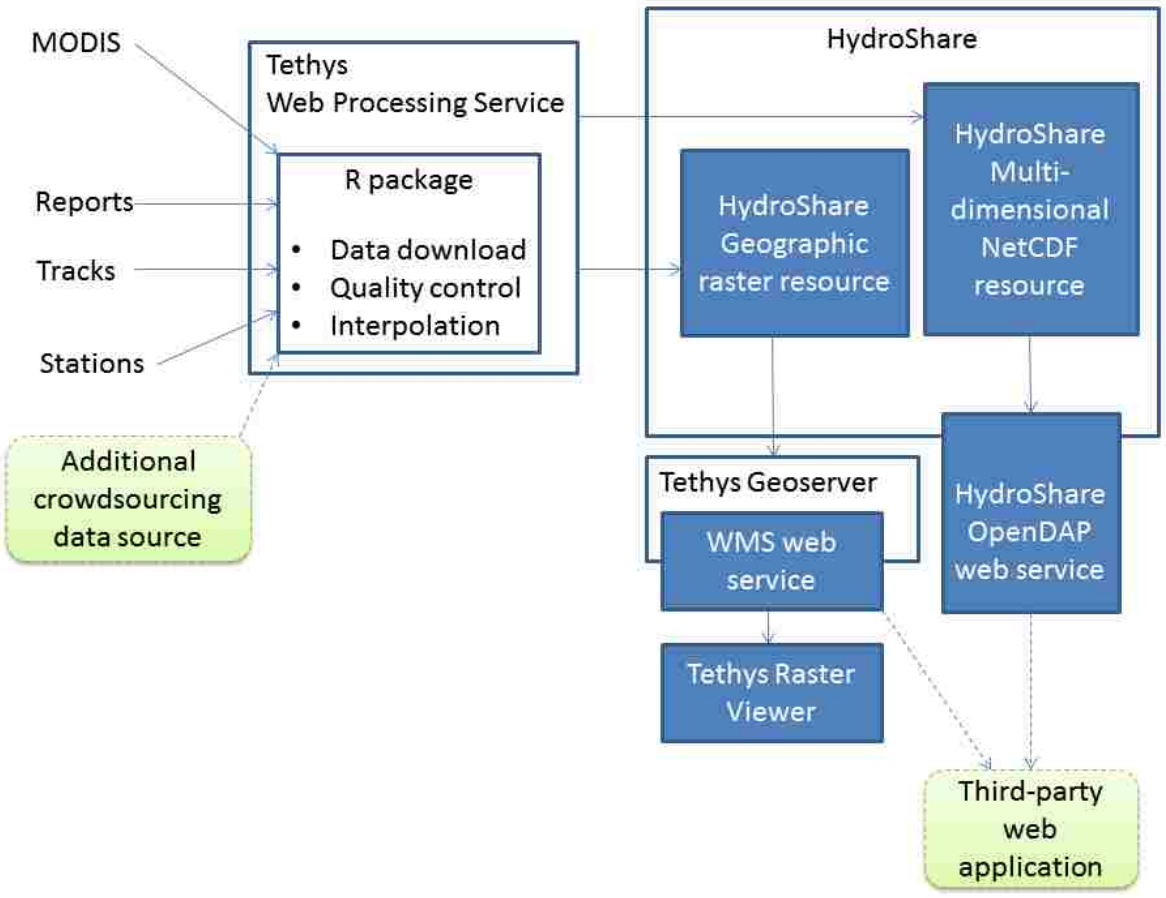

Figure 4-7 Software Architecture for Online Snow Map with WPS, Tethys Platform and HydroShare

3.1.7 Snow Map Generating Script

For automating the snow-covered area map creation, I used the R scripting language. The R computation environment (R Development Core Team 2015) is a multi-platform software for statistical analysis, and it uses a number of packages (rgdal, sp, raster, gdistance) for spatial analysis and geoprocessing tasks. In the first part of the R script the input datasets (MODIS snow cover raster, ski tracks, snow reports, snow station measurements) for the requested date are downloaded from the online sources. For downloading meteorological data from stations the WaterML R package (Kadlec et al. 2015) is used. In the second part of the script a data quality control is executed: The snow reports outside of expected range (negative values) are excluded, and the ski track data are checked for geometry errors. Finally, the interpolation procedure is executed, resulting in a snow cover probability raster with 500 x 500 m cell size. This cell size is same as the original MODIS snow dataset cell size. For better organization and maintenance, I organized the R source code as an R package. In the R package each function is placed in a separate .R script file, and documentation with input parameters and examples how to use each function are provided.

3.1.8 Web Processing Service

A web processing service (WPS) is a mechanism for executing geoprocessing operations on the Internet (OGC 2012). It defines the list of available processes, input data formats, and output data formats. In this design I used the 52North implementation of the WPS that is installed as part of Tethys. The WPS is configured to install and execute user-defined R-scripts. To be recognized as a valid process, the R-script uses a predefined annotation with names of input and output parameters (Hinz et al. 2013). The snow probability map R-script uses the following parameters (Table 5).

Table 4-5 Input Parameters of the Web Processing Service

Parameter	Description
date	The date in YYYY-mm-dd format
stations	One or more point feature sets with snow station data
reports	One or more point feature sets with snow depth report
tracks	One or more line feature sets with ski tracks

3.1.9 Web Map Services for Accessing the Dataset

The output of each web processing service run is a raster file in a GeoTiff file format and UTM zone 33 projection. The permanent disk space for storing the output raster file is provided by HydroShare (Horsburgh et al. 2015). Each raster file stored in HydroShare has the type "geographic raster resource". It has a unique resource identifier, and is associated with resource metadata including geographic coverage, time coverage, author, original data source, and data processing method. HydroShare provides an Application Programming Interface (API) for automatically adding or updating a resource. This API is invoked from the WPS after completion of processing. After the resource is uploaded to HydroShare, it automatically becomes accessible to registered users and applications as a web map service (WMS). The WMS is a standard for publishing re-usable raster maps defined by the Open Geospatial Consortium (OGC). The address of the WMS web request is in the key-value pair format with multiple parameters (Table 4-6). The Coordinate Reference System (CRS) parameter specifies the output map projection name or code. The sld_body optional parameter can be specified in the Styled Layer Descriptor (SLD) format. It is used for defining the map color scheme and specifying map color categories. The transparency value can also be specified, allowing some categories to be displayed as fully or partially transparent.

Table 4-6 Web Map Service Request Parameters

WMS Parameter Name	WMS Parameter Description	WMS Parameter Example
REQUEST	The type of request.	GetMap (for getting map image) or GetFeatureInfo (for getting snow probability value at specified pixel)
FORMAT	The output image format	image/png
LAYERS	The identifier of the dataset and time step	www.hydroshare.org/ 983fd49c63d04ac091388490f8bdd689
CRS	Projection of the map	EPSG:3857
WIDTH	Width of the image in pixels	500
HEIGHT	Height of the image in pixels	500
BBOX	Bounding box of the image in coordinates of the selected projection	1702634, 6344731, 1879510, 6435539
sld_body	Color style of the map (color ramp, transparency)	See Figure 4-8

An example of a SLD specification is shown in Figure 4-8:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<StyledLayerDescriptor version="1.0.0"
xsi:schemaLocation="http://www.opengis.net/sld
StyledLayerDescriptor.xsd" xmlns="http://www.opengis.net/sld"
xmlns:ogc="http://www.opengis.net/ogc"
xmlns:xlink="http://www.w3.org/1999/xlink"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <NamedLayer>
        <Name>www.hydroshare.org:983fd49c63d04ac091388490f8bdd689</Name>
        <UserStyle>
            <Name>mysld</Name>
            <Title>Two color gradient</Title>
            <FeatureTypeStyle>
                <Rule>
                    <RasterSymbolizer>
                        <ColorMap>
                            <ColorMapEntry color="#FFFFFF" quantity="0.0" opacity="0"/>
                            <ColorMapEntry color="#9EC8FF" quantity="0.50" />
                            <ColorMapEntry color="#FF12F7" quantity="1.00" />
                        </ColorMap>
                    </RasterSymbolizer>
                </Rule>
            </FeatureTypeStyle>
        </UserStyle>
    </NamedLayer>
</StyledLayerDescriptor>
```

Figure 4-8 Example of Styled Layer Descriptor (SLD) Specification

In this example the <ColorMapEntry> element specifies the color of the minimum (0.5) and maximum (1) displayed snow probability value. The pixels with snow probability of 0 are displayed as fully transparent (opacity=”0”) to make any background map layer visible.

3.1.10 User Interface Design

Each HydroShare resource type is associated with an interactive visualization application. In the case of the geographic raster resource type, the user can change the map background, projection, and color scheme. User can also change the map scale and view extent. The interactive map functionality in the raster viewer is enabled by using the Tethys map view component. This component is based on the OpenLayers Javascript library (openlayers.org). By default, the map component displays the map in the Mercator projection, because data from numerous Web Map Tile Services (WMTS) that are used as a background map (including Google Maps, OpenStreetMap and OpenSnowMap) are published in the Mercator spatial reference system. The snow probability map layer shown in the viewer is accessed from the HydroShare resource through a Web Map Service (WMS). The translation between the HydroShare resource file and the WMS is enabled by the Tethys platform GeoServer component. The HydroShare raster viewer was implemented using the Tethys platform and is available at https://apps.hydroshare.org.

3.2 Results

The following section describes the validation results of the generated snow probability maps and discusses the effect of using crowdsourcing data on resulting snow extent map accuracy. The web services and web applications for accessing the maps are also presented.

3.2.1 Snow Map Validation Results

For validation, I first randomly selected 10 dates from winter season (December 2014 – March 2015) with MODIS cloud cover more than 75% of the study area (Table 4-7).

Table 4-7 Imposed Cloudy Dates (Cloud Cover > 75%) Used for Snow Map Validation

cloudy date	% cloud cover
12/14/2014	92.0
12/26/2014	83.6
12/31/2014	91.9
1/20/2015	99.8
1/24/2015	88.0
1/25/2015	91.2
2/19/2015	82.1
2/22/2015	92.4
3/22/2015	92.7
3/28/2015	98.4

In the next step I randomly selected 10 dates from the winter seasons (December – March) of 2013, 2014 and 2015 with MODIS cloud cover less than 25% of the study area (Table 4-8). I considered the snow extent on the cloud-free portion of the study area on these dates as "ground truth". Table 4-8 also shows the number of stations, reports and tracks that were available for the selected date.

Finally for each combination of the validation date and cloudy date, I added the cloud mask from the cloudy date to the original MODIS satellite image, and then used the interpolation method to reconstruct the snow map in the area under cloud based on the available stations, reports, tracks and remaining cloud-free MODIS pixels.

Table 4-8 Selected Dates for Validation (Cloud Cover < 25%)

trial #	validation date	# stations	# tracks	# reports	cloud (%)
1	3/17/2013	50	44	26	13.31
2	12/16/2013	49	7	11	13.64
3	3/2/2014	40	7	11	24.28
4	3/8/2014	53	3	10	6.78
5	1/13/2015	45	14	22	1.68
6	2/7/2015	47	122	55	9.08
7	2/16/2015	47	16	28	20.85
8	2/20/2015	48	16	26	21.64
9	3/21/2015	38	5	7	4.83
10	3/24/2015	36	0	7	10.03

The overall validation results (PCC, AUC, commission error and omission error) for each trial are summarized in Table 4-9. The PCC indicator varied from 0.77 to 0.99, and the AUC varied from 0.71 to 0.92. The commission error was higher than the omission error in 8 out of the 10 runs, indicating that the interpolation method has a tendency to overestimate the snow-covered area.

Table 4-9 Results of Validation for 10 Selected Dates using Station, MODIS, Tracks and Reports

trial	date	PCC			AUC			commission error			omission error		
		min	mean	max	min	mean	max	min	mean	max	min	mean	max
1	3/17/2013	0.87	0.88	0.91	0.90	0.91	0.92	0.25	0.30	0.33	0.06	0.08	0.10
2	12/16/2013	0.84	0.85	0.87	0.77	0.78	0.80	0.31	0.39	0.44	0.11	0.12	0.14
3	3/2/2014	0.96	0.97	0.99	0.77	0.84	0.87	0.87	0.91	0.95	0.00	0.01	0.01
4	3/8/2014	0.95	0.96	0.97	0.71	0.73	0.75	0.72	0.77	0.85	0.02	0.03	0.03
5	1/13/2015	0.86	0.87	0.88	0.82	0.84	0.85	0.21	0.26	0.32	0.10	0.11	0.12
6	2/7/2015	0.80	0.80	0.81	0.91	0.91	0.92	0.03	0.04	0.05	0.39	0.41	0.44
7	2/16/2015	0.77	0.78	0.79	0.82	0.83	0.85	0.11	0.16	0.19	0.23	0.24	0.26
8	2/20/2015	0.77	0.79	0.81	0.77	0.78	0.79	0.25	0.29	0.32	0.16	0.19	0.22
9	3/21/2015	0.95	0.96	0.98	0.93	0.94	0.95	0.62	0.68	0.72	0.00	0.01	0.01
10	3/24/2015	0.95	0.96	0.98	0.92	0.93	0.94	0.66	0.73	0.77	0.01	0.01	0.01

The following example shows the steps of the validation procedure for a selected cloud-free / cloudy pair, using the date of 7th February 2015 as the cloud-free date (cloud cover 9 %, see Figure 4-9), and 24th January as the imposed cloudy date (cloud cover 88%, see Figure 4-10).

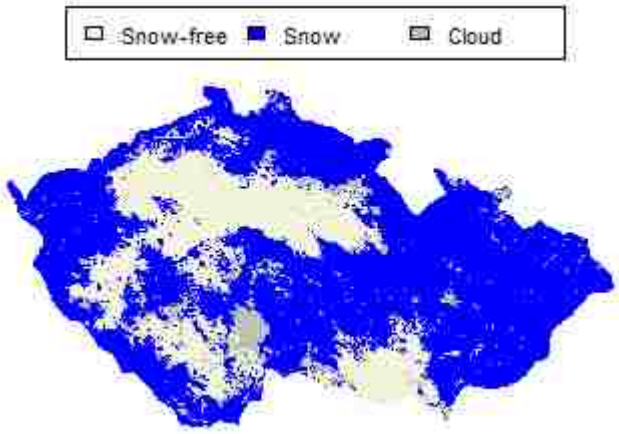

Figure 4-9 Original Dataset with 9% Cloud (7th Feb 2015)

Figure 4-10 Cloud Mask from Cloudy Date (24th Jan 2015)

The overlay of the ground truth and the cloud cover mask is shown in Figure 4-11. The stations, reports, tracks and MODIS snow-covered and snow-free areas are shown in Figure 4-12.

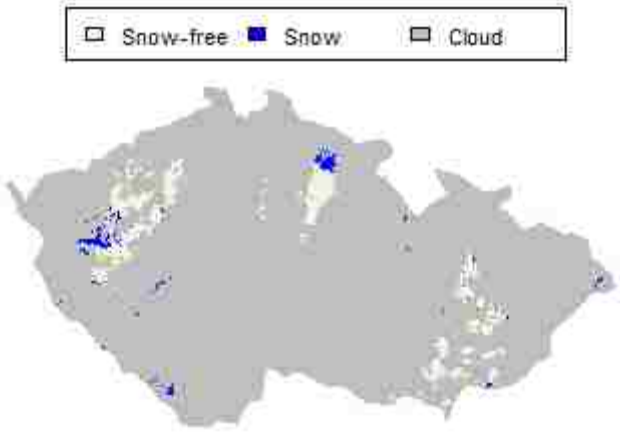

Figure 4-11 Combination of Ground Truth (7th Feb 2015) with Imposed Cloud Mask from 24th Jan 2015

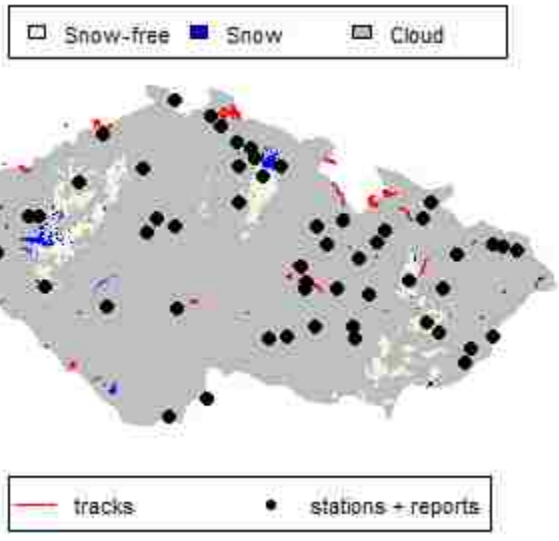

Figure 4-12 Using the Ski Tracks, Stations, and Reports from 7th Feb 2015 to Obtain Snow Probability Map

The resulting snow probability map (Figure 4-13) is reclassified using a threshold value of 0.5 to define snow-covered and snow-free areas (Figure 4-14). The result is compared with values of the original ground-truth map (Figure 4-9) to obtain the confusion matrix, PCC and AUC error indicators.

Figure 4-13 Calculated Snow Probability Map

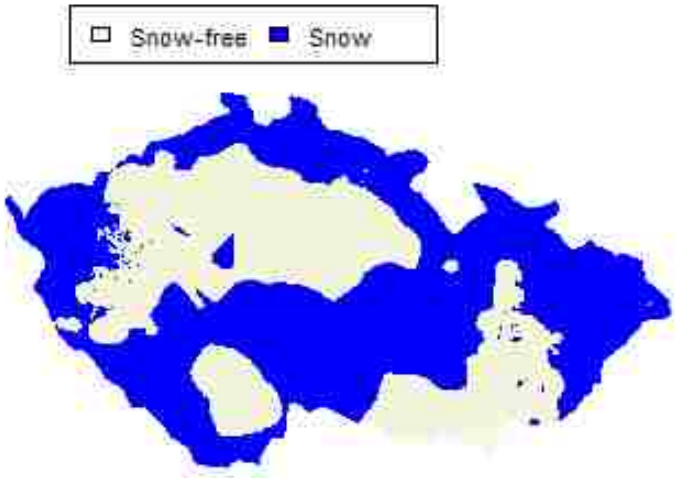

Figure 4-14 Calculated Snow Extent Map (Threshold = 0.5)

3.2.2 Effect of Crowdsourcing Data on Snow Map Accuracy

To determine if the use of crowdsourcing data increased or decreased the overall snow cover probability map accuracy, I repeated the validation procedure described above using only MODIS and station data as inputs. The results of the comparison for mean values of PCC, AUC, commission error, and omission error for the 10 trials are shown in Table 4-10.

Table 4-10 Results of Validation for 10 Selected Dates using Station and MODIS Only

trial	date	PCC		AUC		Commission Error		Omission Error	
		S+M	S+M+R+T	S+M	S+M+R+T	S+M	S+M+R+T	S+M	S+M+R+T
1	3/17/2013	0.88	0.88	0.90	0.91	0.30	0.30	0.09	0.08
2	12/16/2013	0.84	0.85	0.77	0.78	0.42	0.39	0.13	0.12
3	3/2/2014	0.97	0.97	0.81	0.84	0.93	0.91	0.01	0.01
4	3/8/2014	0.96	0.96	0.72	0.73	0.72	0.77	0.03	0.03
5	1/13/2015	0.86	0.87	0.83	0.84	0.29	0.26	0.13	0.11
6	2/7/2015	0.79	0.80	0.88	0.91	0.06	0.04	0.42	0.41
7	2/16/2015	0.74	0.78	0.76	0.83	0.20	0.16	0.27	0.24
8	2/20/2015	0.77	0.79	0.76	0.78	0.33	0.29	0.21	0.19
9	3/21/2015	0.96	0.96	0.95	0.94	0.67	0.68	0.01	0.01
10	3/24/2015	0.96	0.96	0.94	0.93	0.74	0.73	0.01	0.01

As seen from Table 4-10 the overall map accuracy measured by the PCC, AUC, commission error and omission error has improved for runs 5, 6, 7 and 8. For the other runs, the accuracy appears to be equal and in some cases (for example commission error for run 4) the use of reports and tracks leads to a decrease in the map accuracy. Figure 4-15 compares the distribution of the PCC for each validation date, using the cloudy dates from Table 4-7 and the validation dates from Table 4-8. The PCC distribution in Figure 4-15 suggests that the greatest improvement in PCC value was for 16th February 2015 and 20th February 2015 (runs 7 and 8). At the same time, there was an overall decrease in PCC for the trial runs on 2nd March 2014, 8th March 2014, and 21st March 2015 (runs 3, 4 and 9).

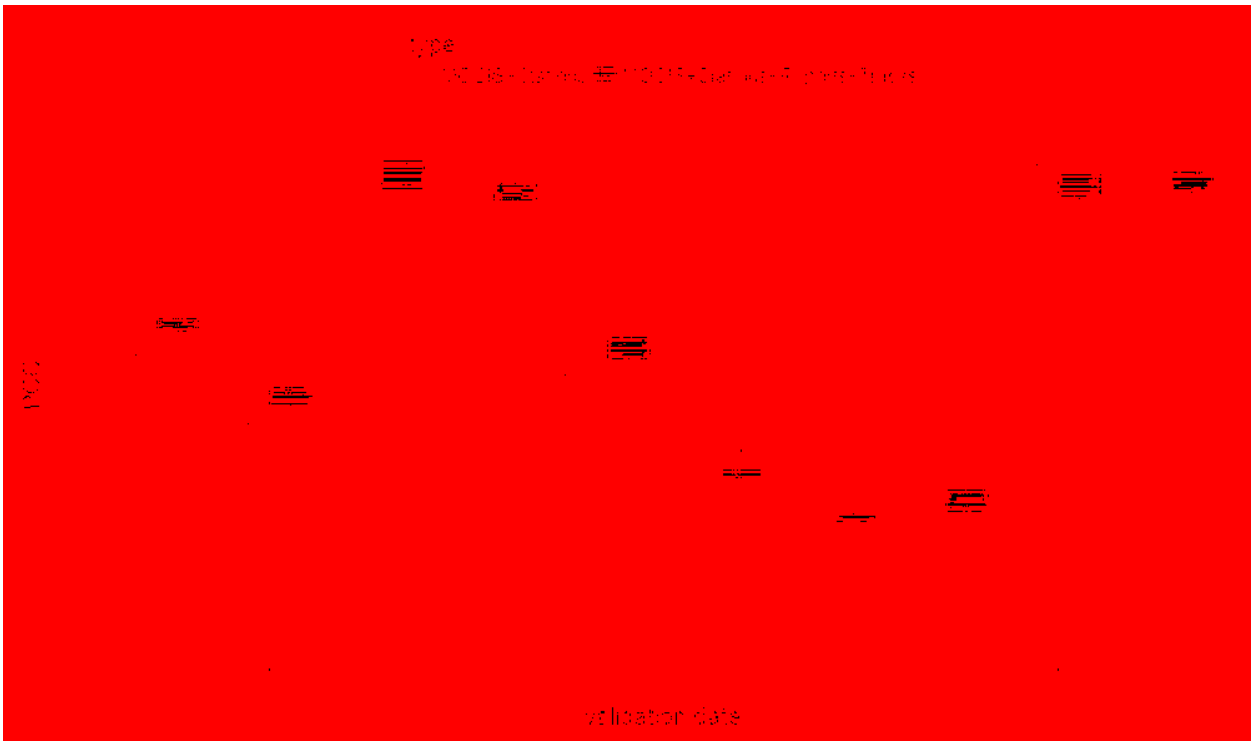

Figure 4-15 Change of PCC when Using Reports and Tracks for Snow Cover Map Creation

To evaluate if using crowdsourcing data (reports and tracks) results in a statistically significant improvement of map accuracy, I run a paired T-test for all 100 combinations of cloud-free and cloudy date that were used in the validation procedure from Table 4-7. The one-sided t-test tests the null hypothesis: "the PCC value of the snow extent map created using MODIS and stations is greater or equal to the PCC value of the snow extent map created using MODIS, stations, reports and tracks". I also run a similar paired T-test for the AUC, commission error, and omission error indicators. Each t-test has 99 degrees of freedom and also provides a 95% confidence interval for the difference in mean accuracy indicator value for the two groups. The outcome of the tests is shown in Table 4-11. According to the t-test, there is a statistically significant difference in the mean values of PCC, AUC, commission error, and omission error when using the reports and tracks. The PCC and AUC indicators have increased by 0.9% and

1.8%, respectively. The commission error has been reduced by 1.3%, and the omission error has been reduced by 0.9%. At the same time, the accuracy improvement is very small. The AUC indicator, which shows the greatest improvement, is only increased by 1.8% (95% confidence interval between 1.3 and 2.3%). This may be due to the trial runs 3, 4 and 9 that showed a deterioration of many of the accuracy indicators.

Table 4-11 Results of the T-Test to Test the Change in Map Accuracy when Using Crowdsourcing Data

Accuracy indicator	t statistic	p-value	Mean of the differences	95% Confidence interval
PCC	7.026	1.367e-10	0.0090	0.0064 – 0.0116
AUC	7.989	1.263e-12	0.0183	0.0137 – 0.0228
Commission error	-3.6877	0.0002	-0.0129	-0.0198 – -0.0059
Omission error	-3.6877	1.213e-12	-0.0095	-0.0118 – -0.0071

3.2.3 Software Results

The software for snow probability map generation can be used in several ways: (1) directly using the SnowDataFusion R package, (2) getting the result data files from HydroShare, (3) using the WMS web map service, (4) using the interactive web map viewer. Figure 4-16 shows an example of the snow map for 7th February 2015 in the web map viewer using a pre-defined style and color scheme. The snow-free areas are marked as transparent, allowing the user to view the snow map together with an OpenStreetMap background map. The user can also modify the color scheme in the map. Each resource in HydroShare is addressed using a unique resource identifier. Figure 4-17 shows the resource metadata page on HydroShare. The resource page contains information about the data source, projection, geographic coverage and time coverage, and allows the user to download the raster file in GeoTiff format for use in custom GIS software.

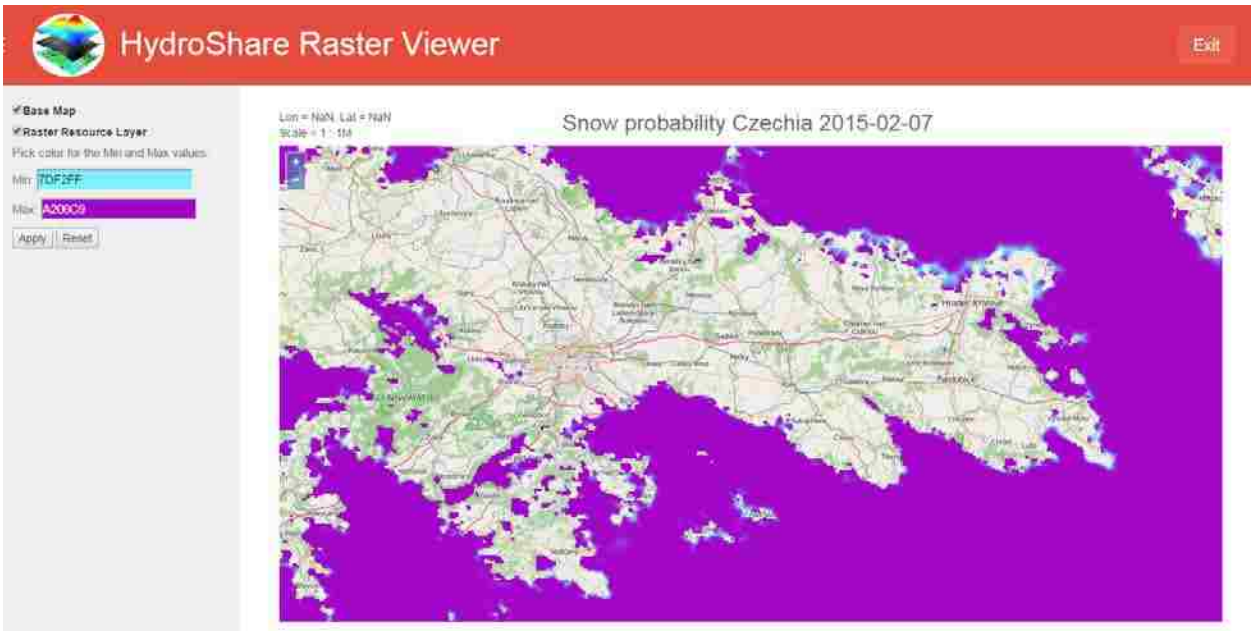

Figure 4-16 Showing Snow Probability Map for 7th Feb 2015 in HydroShare Raster Viewer

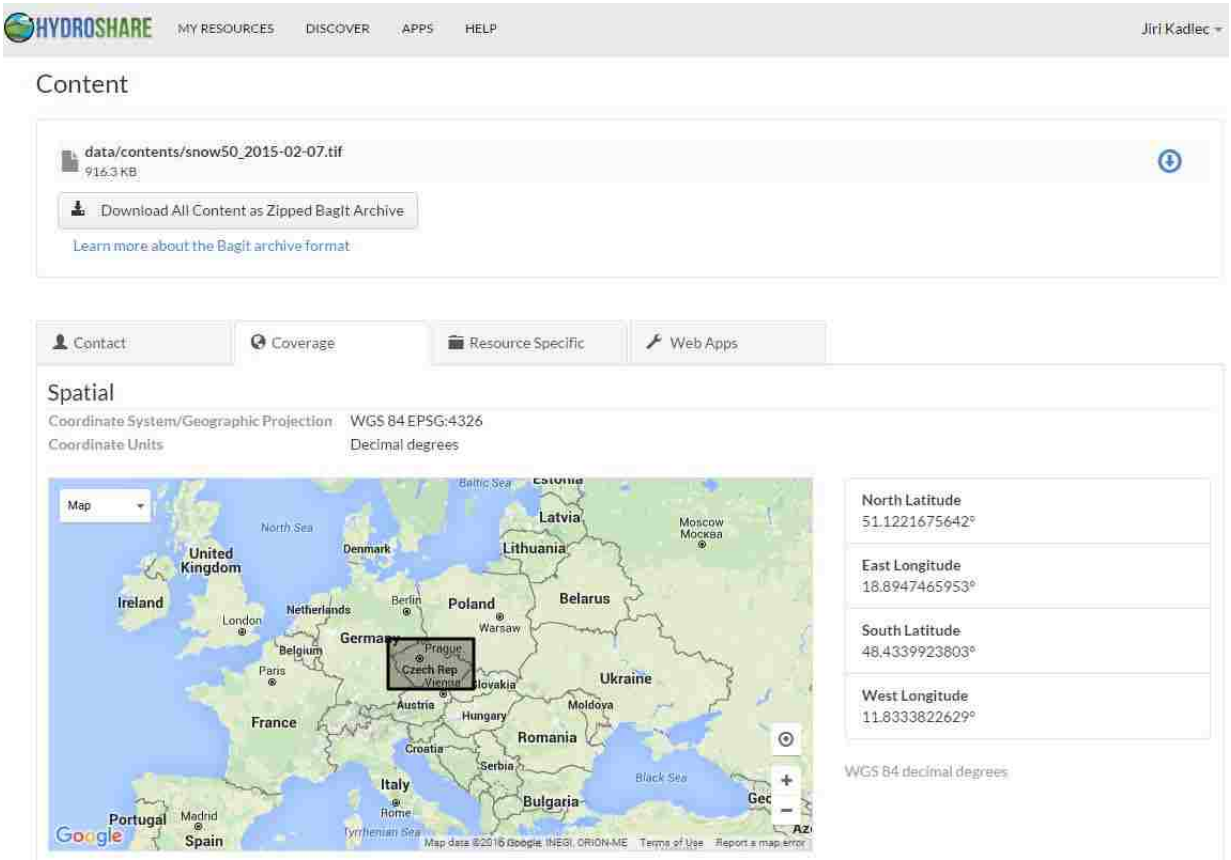

Figure 4-17 HydroShare Resource Metadata Page Showing Spatial Coverage

3.3 Discussion and Conclusions

Like other weather data crowdsourcing applications, this application explores how to use the big data provided by citizen-as-sensor networks for improving the spatial coverage of environmental datasets. In the case described here the raw input is the MODIS snow dataset with gaps in the data, and the value-added output is a continuous, gap-free snow probability map.

Our product is similar to the IMS snow map (Helfrich et al. 2007) in the use of multiple input data sources. As with the IMS, I use the MODIS MOD10A1 satellite dataset as a primary data source. However, the other data sources are significantly different and potentially more informative. IMS primarily uses microwave satellite data to fill gaps in the optical satellite data. According to my knowledge the IMS does not make use of crowdsourcing data. My approach relies on ground observations from meteorological stations, volunteer reports, and cross-country ski track reporting. The IMS products are currently only available in the form of raster data files or overview images. The resulting snow map provides more access options to the data including a WMS web service interface. The WMS interface encourages developing third-party interactive web and mobile map applications. One example application could be overlaying the snow probability map with ski paths from the OpenSnowMap project to show the current navigability of cross-country ski routes. Another potential application is using the snow probability map to constraint a snow depth map interpolated from meteorological stations and volunteer snow depth measurements.

Currently the snow probability map presented here has several limitations. The study area is limited to Czechia. However, the map could be relatively easily expanded to other regions (Germany, Scandinavia) that have good availability of cross-country ski track and volunteer snow reports. The interpolation algorithm currently used relies on inverse distance to assign

weights to each snow data source. In complex mountainous terrain, other factors such as elevation difference or slope aspect change may be more important than distance in influencing the spatial variability of snow cover. Typically, cross-country skiers are interested in presence of "skiable snow". Depending on terrain and land cover, the depth of skiable snow is around 20 cm (8 in). For new snow on arable land, higher depth is needed, while a shallower snow depth (< 10 cm) may already be skiable on grass or on maintained roads. Provided that land cover data about ski path surface are available, the cross-country skiing track reports could be used to estimate not only the presence or absence of snow, but also snow depth.

Finally, some limitations in the crowdsourcing data need to be acknowledged. Most of the cross-country skiers are motivated to report when snow is present, but not when snow is absent. This may lead to an overestimation of snow-covered area (commission error) in the map. Typically, the maintained cross-country ski trails are used as bicycle trails in the snow-free season. Therefore, I could also use reports from bicycle trips as "absence of snow" data. Experience with using GPS-recorded cross-country skiing tracks also showed some errors where the users assigned an incorrect category to the track, for example a part of the cross-country trip was in fact a car trip. I reduced the number of these errors by filtering the tracks by speed. More advanced data quality control methods could be developed.

Our study demonstrates that station ground observations together with crowdsourcing data from volunteer snow depth reports and cross-country skiing tracks can be successfully used to fill cloud gaps in MODIS snow cover maps. Validation has shown that the method can reconstruct the presence or absence of snow under cloud with accuracy between 78% and 97%. The method presented here is extensible. Additional types of crowdsourced data can be easily added to the snow map production script, further increasing the map accuracy, map detail, and

spatial coverage. Furthermore, all of the output datasets are available using a standard WMS web service interface, encouraging a re-use of the data in third-party environmental software applications.

4 CONCLUSIONS

The aim of this dissertation was to contribute to the scientific understanding of snow cover spatial distribution by combining ground station, remote sensing and crowdsourcing observations from multiple sources. The proposed hypothesis was that the integration of volunteer geographic information and/or social-network derived snow data together with other open access data sources results in more accurate and higher resolution – and hence more useful snow cover maps than government agency produced data by itself.

The first step of this research was designing and developing algorithms and software tools to automate the search, retrieval and analysis of three main types of direct and indirect snow observation datasets. The WaterML R package described in Chapter 2 is aimed at automated retrieval of time series data in the WaterML format from the Consortium of Universities for Advancement of Hydrologic Sciences (CUAHSI) Water Data Center (WDC). The SnowInspector tool described in Chapter 3 collects time series from multi-temporal web map tile services including the daily MODIS Terra snow cover dataset. Finally the SnowDataFusion R package and web processing service described in Chapter 4 allows accessing large volumes of volunteer geographic information (VGI) from Garmin Connect and Strava ski route recordings and from the in-pocasi.cz volunteer meteorology social network.

The second step of this research was the design of a validation methodology to assess the effect of using data sources from different origins for updating snow cover maps. The validation

procedure checked the ability of this method to reconstruct MODIS snow cover under cloud by simulating cloud cover datasets and comparing estimated snow cover to actual MODIS snow cover. Unlike other methods that rely on leave-one-out cross validation or on point validation sites, this method uses the remote sensing dataset as a "ground truth", allowing the user to evaluate the map's accuracy in different sub-regions of the study area.

Based on the validation results, the increase of overall snow cover extent map accuracy by using crowdsourcing data was statistically significant. Including volunteer snow reports and cross-country ski tracks improved the overall accuracy of snow cover extent maps in the study area. The percent correctly classified (PCC) indicator was increased by 0.9 % (0.6 % - 1.2 %), the omission error was reduced by 1.0 % (0.7 % - 1.2 %), and the commission error was reduced by 1.3 % (0.5 % - 2.0 %). While these results are statistically significant, the map accuracy improvement is smaller than expected. The best effect of using crowdsourcing data was found on days with a large number snow reports and recorded ski tracks (> 100 volunteer observations). On some days with a small number of available crowdsourcing observations (< 20 data points), the crowdsourcing data was found to reduce the resulting snow extent map accuracy. This may suggest an existence of a minimum number ("critical mass") of required crowdsourcing reports that are required to be able to improve the map accuracy. Examining this threshold in more detail could be an interesting topic for future research.

The snow probability map presented in this dissertation has several limitations. The study area is limited to the Czech Republic. However, the map could be relatively easily expanded to other regions (Germany, Scandinavia) that have high density of cross-country ski track and volunteer snow reports. As an example, expanding the map to Finland would require changing

the volunteer snow report retrieval function to retrieve data from the latutilanne.fi (ski track status) crowdsourcing website.

A major limitation of the snow probability map is the interpolation algorithm, which currently used relies on inverse distance to assign weights to each snow observation. In complex mountainous terrain, other factors such as elevation difference or slope aspect change may be more important than distance in influencing the spatial variability of snow cover. Using an elevation weighted least-cost distance partially takes the topography into account, but other geostatistical methods (indicator kriging, co-kriging) could be tested.

The volume of crowdsourcing data that exists about snow is much greater than the three data sources used in this study. For example, the website kamzasnehem.cz gathers numerous skier reports about snow conditions. These reports are in form of text message and do not contain explicit geographic coordinates. A geocoding function for searching the text and linking the message with a spatial polygon, line or point would need to be developed to use the text reports for snow map generation. Other types of potentially highly informative snow crowdsourcing data are photographs taken by social network users or by automatic web cameras. Typically these photographs contain geocoding information embedded in the image file, and there is a potential of using automated feature recognition techniques to detect snow in the images.

The remote sensing data used in this study was limited to the MODIS Terra satellite sensor because of its long period of record and high temporal coverage. Other remote sensing platforms and especially the recently launched Globsnow data service should be compared with MODIS and included as inputs for the snow cover map.

Finally, this study has not examined the use of physically based meteorological and hydrological models. Physically based models of the snowpack rely on high quality

meteorological data inputs (precipitation, temperature, wind, solar radiation). With increased availability spatial resolution of numerical weather forecasting models, assimilating the physically-based models with ground observations (including the types of crowdsourcing data analyzed in this study) presents the next step towards developing an accurate, high resolution, and open access global datasets of snow cover extent, snow depth, and snow water equivalent.

All of the methods, algorithms and software procedures presented in this work have been published as free and open source software. Using the R environment as a primary development platform enables users of all major operating systems (Windows, Linux, Mac OS) to explore, test and modify the presented methods. Appendix A: software availability lists the addresses of the software repositories on the Internet. All the geographical and time series datasets used in this work are also available to the public as resources on the HydroShare (hydroshare.org) repository or through the CUAHSI HydroServer WaterOneFlow web services. Ensuring the online availability of used software and datasets is an important step to facilitate reproducible research, to expand the snow probability map to other regions, and to integrate new snow observation data sources that will become available in future.

REFERENCES

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

Amintoosi, H., S. S. Kanhere & M. Allahbakhsh, 2015. Trust-based privacy-aware participant selection in social participatory sensing. Journal of Information Security and Applications 20:11-25.

Anderson, E. A., 1973. National Weather Service river forecast system: Snow accumulation and ablation model, vol 17. US Department of Commerce, National Oceanic and Atmospheric Administration, National Weather Service.

Bai, Y., L. Di, D. D. Nebert, A. Chen, Y. Wei, X. Cheng, Y. Shao, D. Shen, R. Shrestha & H. Wang, 2012. GEOSS component and service registry: Design, implementation and lessons learned. Selected Topics in Applied Earth Observations and Remote Sensing, IEEE Journal of 5(6):1678-1686.

Bakillah, M., J. Lauer, S. Liang, A. Zipf, J. Jokar Arsanjani, L. Loos & A. Mobasheri, 2014. Exploiting big VGI to improve routing and navigation services. Big data techniques and technologies in geoinformatics:177-192.

Baltazar, J.-C. & D. E. Claridge, 2006. Study of cubic splines and Fourier series as interpolation techniques for filling in short periods of missing building energy use and weather data. Journal of solar energy engineering 128(2):226-230.

Bambacus, M., C. Yang, J. Evans, Z. Li, W. Li & Q. Huang, Sharing Earth Science Information to Support the Global Earth Observing System of Systems (GEOSS). In: Geoscience and Remote Sensing Symposium, 2008 IGARSS 2008 IEEE International, 2008. vol 1. IEEE, p I-141-I-144.

Barnett, T. P., J. C. Adam & D. P. Lettenmaier, 2005. Potential impacts of a warming climate on water availability in snow-dominated regions. Nature 438(7066):303-309.

Basset, A. & W. Los, 2012. Biodiversity e‐Science: LifeWatch, the European infrastructure on biodiversity and ecosystem research. Plant Biosystems-An International Journal Dealing with all Aspects of Plant Biology 146(4):780-782.

Batty, M., A. Hudson-Smith, R. Milton & A. Crooks, 2010. Map mashups, Web 2.0 and the GIS revolution. Annals of GIS 16(1):1-13.

Baumann, P., P. Mazzetti, J. Ungar, R. Barbera, D. Barboni, A. Beccati, L. Bigagli, E. Boldrini, R. Bruno & A. Calanducci, 2015. Big Data Analytics for Earth Sciences: The EarthServer Approach. International Journal of Digital Earth(ahead-of-print):1-27.

Bednorz, E., 2004. Snow cover in eastern Europe in relation to temperature, precipitation and circulation. International Journal of Climatology 24(5):591-601.

Bica, B., A. Kann & I. Meirold-Mautner, Enhanced road weather warnings and improved communication strategies within central Europe as part of the INCA-CE project. In: 16th International Road Weather Conference (SIRWEC 2012), Helsinki, Finland, 2012.

Blower, J., A. Gemmell, G. Griffiths, K. Haines, A. Santokhee & X. Yang, 2013. A Web Map Service implementation for the visualization of multidimensional gridded environmental data. Environmental Modelling & Software 47:218-224.

Braunisch, V., P. Patthey & R. Arlettaz, 2011. Spatially explicit modeling of conflict zones between wildlife and snow sports: prioritizing areas for winter refuges. Ecological Applications 21(3):955-967.

Brown, R. & D. Robinson, 2011. Northern Hemisphere spring snow cover variability and change over 1922–2010 including an assessment of uncertainty. The Cryosphere 5(1):219-229.

Callaghan, T. V., M. Johansson, R. D. Brown, P. Y. Groisman, N. Labba, V. Radionov, R. G. Barry, O. N. Bulygina, R. L. Essery & D. Frolov, 2011. The changing face of Arctic snow cover: A synthesis of observed and projected changes. Ambio 40(1):17-31.

Cechini, M., K. Murphy, R. Boller, J. Schmaltz, C. Thompson, T. Huang, J. McGann, S. Ilavajhala, C. Alarcon & J. Roberts, Expanding Access and Usage of NASA Near Real-Time Imagery and Data. In: AGU Fall Meeting Abstracts, 2013. vol 1. p 04.

Chang, N.-B., K. Bai & C.-F. Chen, 2015. Smart Information Reconstruction via Time-Space-Spectrum Continuum for Cloud Removal in Satellite Images.

Conner, L. G., D. P. Ames & R. A. Gill, 2013. HydroServer Lite as an open source solution for archiving and sharing environmental data for independent university labs. Ecological Informatics 18(0):171-177 doi:http://dx.doi.org/10.1016/j.ecoinf.2013.08.006.

De Cicco, L. & R. Hirsch, Discussion of US Geological Survey development of R packages in the open source community: Exploration and Graphics for RivEr Trends (EGRET) and dataRetrieval. In: AGU Fall Meeting Abstracts, 2013. vol 1. p 08.

Delucchi, L., pyModis: from satellite to GIS maps. In: FOSS4G 2014, 2014.

Derksen, C. & R. Brown, 2012. Spring snow cover extent reductions in the 2008–2012 period exceeding climate model projections. Geophysical Research Letters 39(19).

Dietz, A. J., C. Kuenzer, U. Gessner & S. Dech, 2012. Remote sensing of snow–a review of available methods. International Journal of Remote Sensing 33(13):4094-4134.

Dozier, J., T. H. Painter, K. Rittger & J. E. Frew, 2008. Time–space continuity of daily maps of fractional snow cover and albedo from MODIS. Advances in Water Resources 31(11):1515-1526 doi:http://dx.doi.org/10.1016/j.advwatres.2008.08.011.

Dwyer, M. J. & G. Schmidt, 2006. The MODIS reprojection tool Earth science satellite remote sensing. Springer, 162-177.

Foster, J. L., D. K. Hall, J. B. Eylander, G. A. Riggs, S. V. Nghiem, M. Tedesco, E. Kim, P. M. Montesano, R. E. Kelly & K. A. Casey, 2011. A blended global snow product using visible, passive microwave and scatterometer satellite data. International journal of remote sensing 32(5):1371-1395.

Fuka, D., M. Walter, J. Archibald, T. Steenhuis & Z. Easton, 2013. EcoHydRology: a community modeling foundation for eco-hydrology. R package version 49.

Gafurov, A. & A. Bárdossy, 2009. Cloud removal methodology from MODIS snow cover product. Hydrology and Earth System Sciences 13(7):1361-1373.

Gao, Y., H. Xie, T. Yao & C. Xue, 2010. Integrated assessment on multi-temporal and multi-sensor combinations for reducing cloud obscuration of MODIS snow cover products of the Pacific Northwest USA. Remote Sensing of Environment 114(8):1662-1675.

Gascoin, S., O. Hagolle, M. Huc, L. Jarlan, J.-F. Dejoux, C. Szczypta, R. Marti & R. Sánchez, 2014. A snow cover climatology for the Pyrenees from MODIS snow products. Hydrology and Earth System Sciences Discussions 11(11):12531-12571.

Gentleman, R. & D. T. Lang, 2007. Statistical analyses and reproducible research. Journal of Computational and Graphical Statistics 16(1).

Ghaderi, Z., M. Khoshkam & J. C. Henderson, 2014. From snow skiing to grass skiing: implications of climate change for the ski industry in Dizin, Iran. Anatolia 25(1):96-107.

Gómez-Landesa, E., A. Rango & M. Bleiweiss, 2004. An algorithm to address the MODIS bowtie effect. Canadian Journal of Remote Sensing 30(4):644-650.

Goodchild, M. F., 2007. Citizens as sensors: the world of volunteered geography. GeoJournal 69(4):211-221.

Groisman, P. Y., R. W. Knight, V. N. Razuvaev, O. N. Bulygina & T. R. Karl, 2006. State of the ground: Climatology and changes during the past 69 years over northern Eurasia for a rarely used measure of snow cover and frozen land. Journal of climate 19(19):4933-4955.

Haklay, M., 2010. How good is volunteered geographical information? A comparative study of OpenStreetMap and Ordnance Survey datasets. Environment and planning B, Planning & design 37(4):682.

Hall, D., G. Riggs & V. Salomonson, 2006a. MODIS/Terra Snow Cover 5-min L2 Swath 500m V005. National Snow and Ice Data Center, Boulder, CO, digital media.

Hall, D., G. Riggs & V. Salomonson, 2006b. MODIS/Terra Snow Cover Daily L3 Global 500m Grid V005, updated daily.

Hall, D., V. Salomonson & G. Riggs, 2006c. MODIS/Terra snow cover daily L3 global 500m grid. Version 5 Boulder, Colorado USA: National Snow and Ice Data Center.

Hall, D. K. & G. A. Riggs, 2007. Accuracy assessment of the MODIS snow products. Hydrological Processes 21(12):1534-1547.

Hall, D. K., G. A. Riggs, J. L. Foster & S. V. Kumar, 2010. Development and evaluation of a cloud-gap-filled MODIS daily snow-cover product. Remote sensing of environment 114(3):496-503.

Hall, D. K., G. A. Riggs, V. V. Salomonson, N. E. DiGirolamo & K. J. Bayr, 2002. MODIS snow-cover products. Remote sensing of Environment 83(1):181-194.

Havlik, D., M. Egly, H. Huber, P. Kutschera, M. Falgenhauer & M. Cizek, 2013. Robust and Trusted Crowd-Sourcing and Crowd-Tasking in the Future Internet. In Hřebíček, J., G. Schimak, M. Kubásek & A. E. Rizzoli (eds) Environmental Software Systems Fostering Information Sharing: 10th IFIP WG 511 International Symposium, ISESS 2013, Neusiedl am See, Austria, October 9-11, 2013 Proceedings. Springer Berlin Heidelberg, Berlin, Heidelberg, 164-176.

Heineman, G. T. & W. T. Councill, 2001. Component-based Software Engineering: Putting the Pieces Together. Addison-Wesley.

Helfrich, S. R., D. McNamara, B. H. Ramsay, T. Baldwin & T. Kasheta, 2007. Enhancements to, and forthcoming developments in the Interactive Multisensor Snow and Ice Mapping System (IMS). Hydrological processes 21(12):1576-1586.

Henderson, G. R. & D. J. Leathers, 2010. European snow cover extent variability and associations with atmospheric forcings. International Journal of Climatology 30(10):1440-1451.

Hinz, M., D. Nüst, B. Proß & E. Pebesma, 2013. Spatial Statistics on the Geospatial Web. Paper presented at the AGILE - Association of Geographic Information Laboratories in Europe, Leuven.

Horsburgh, J. S., M. M. Morsy, A. M. Castronova, J. L. Goodall, T. Gan, H. Yi, M. J. Stealey & D. G. Tarboton, 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 doi:10.1111/1752-1688.12363.

Horsburgh, J. S. & S. L. Reeder, 2014. Data visualization and analysis within a Hydrologic Information System: Integrating with the R statistical computing environment. Environmental Modelling & Software 52:51-61.

Horsburgh, J. S., D. G. Tarboton, D. R. Maidment & I. Zaslavsky, 2008. A relational model for environmental and water resources data. Water Resources Research 44(5).

Horsburgh, J. S., D. G. Tarboton, M. Piasecki, D. R. Maidment, I. Zaslavsky, D. Valentine & T. Whitenack, 2009. An integrated system for publishing environmental observations data. Environmental Modelling & Software 24(8):879-888.

Houser, P. R., G. J. De Lannoy & J. P. Walker, 2012. Hydrologic data assimilation. INTECH Open Access Publisher.

Howe, J., 2008. Crowdsourcing: How the power of the crowd is driving the future of business. Random House.

Izurieta, C., G. Poole, B. McGlynn, W. Cross, L. Marshall, G. Jacobs, S. Cleveland, I. Judson, F. Hauer & B. Kucera, A cyber-infrastructure for a Virtual Observatory and Ecological Informatics System-VOEIS. In: AGU Fall Meeting Abstracts, 2010. vol 1. p 02.

Jacobson, A., J. Dhanota, J. Godfrey, H. Jacobson, Z. Rossman, A. Stanish, H. Walker & J. Riggio, 2015. A novel approach to mapping land conversion using Google Earth with an application to East Africa. Environmental Modelling & Software 72:1-9 doi:http://dx.doi.org/10.1016/j.envsoft.2015.06.011.

James, D. A. & S. DebRoy, 2012. RMySQL: R interface to the MySQL database. R package version 09-3.

James, D. A. & S. Falcon, 2011. RSQLite: SQLite interface for R. R package version 011 1.

Jones, N., J. Nelson, N. Swain, S. Christensen, D. Tarboton & P. Dash, Tethys: A Software Framework for Web-Based Modeling and Decision Support Applications. In: Ames, D. P., N. W. T. Quinn & A. E. Rizzoli (eds) 7th International Congress on Environmental Modelling and Software, San Diego, California, USA, 2014. iEMSs, p 170-177.

Kadlec, J. & D. P. Ames, 2011. Design and development of web services for accessing free hydrological data from the Czech Republic Environmental Software Systems Frameworks of eEnvironment. Springer, 581-588.

Kadlec, J. & D. P. Ames, Development of a Lightweight Hydroserver and Hydrologic Data Hosting Website. In: Proceedings of the AWRA Spring Specialty Conference on GIS and Water Resources, New Orleans, 2012.

Kadlec, J., B. StClair, D. P. Ames & R. A. Gill, 2015. WaterML R Package for Managing Ecological Experiment Data on a CUAHSI HydroServer. Ecological Informatics.

Karl, T. R., P. Y. Groisman, R. W. Knight & R. R. Heim Jr, 1993. Recent variations of snow cover and snowfall in North America and their relation to precipitation and temperature variations. Journal of Climate 6(7):1327-1344.

Klein, A. G., D. K. Hall & G. A. Riggs, 1998. Improving snow cover mapping in forests through the use of a canopy reflectance model. Hydrological Processes 12(10):1723-1744.

Koivusalo, H., M. Heikinheimo & T. Karvonen, 2001. Test of a simple two-layer parameterisation to simulate the energy balance and temperature of a snow pack. Theoretical and Applied Climatology 70(1-4):65-79.

Lapena, D. R. & L. W. Martz, 1996. An investigation of the spatial association between snow depth and topography in a Prairie agricultural landscape using digital terrain analysis. Journal of Hydrology 184(3):277-298.

Lehning, M., I. Völksch, D. Gustafsson, T. A. Nguyen, M. Stähli & M. Zappa, 2006. ALPINE3D: a detailed model of mountain surface processes and its application to snow hydrology. Hydrological Processes 20(10):2111-2128.

Lenzerini, M., Data integration: A theoretical perspective. In: Proceedings of the twenty-first ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems, 2002. ACM, p 233-246.

Li, J. & A. D. Heap, 2014. Spatial interpolation methods applied in the environmental sciences: A review. Environmental Modelling & Software 53:173-189 doi:http://dx.doi.org/10.1016/j.envsoft.2013.12.008.

Loecher, M. & K. Ropkins, 2015. RgoogleMaps and loa: unleashing R graphics power on map tiles. Journal of Statistical Software 63(4).

López‐Moreno, J. I. & D. Nogués‐Bravo, 2006. Interpolating local snow depth data: an evaluation of methods. Hydrological processes 20(10):2217-2232.

Mankin, J. S., D. Viviroli, D. Singh, A. Y. Hoekstra & N. S. Diffenbaugh, 2015. The potential for snow to supply human water demand in the present and future. Environmental Research Letters 10(11):114016.

Masó, J., K. Pomakis & N. Julià, 2010. OGC Web Map Tile Service (WMTS). Implementation Standard Ver 1.

Mason, S. J., S. B. Cleveland, P. Llovet, C. Izurieta & G. C. Poole, 2014. A centralized tool for managing, archiving, and serving point-in-time data in ecological research laboratories. Environmental Modelling & Software 51:59-69.

Metsämäki, S., O.-P. Mattila, J. Pulliainen, K. Niemi, K. Luojus & K. Böttcher, 2012. An optical reflectance model-based method for fractional snow cover mapping applicable to continental scale. Remote Sensing of Environment 123:508-521.

Metz, C. E., Basic principles of ROC analysis. In: Seminars in nuclear medicine, 1978. vol 8. Elsevier, p 283-298.

Michener, W. K. & M. B. Jones, 2012. Ecoinformatics: supporting ecology as a data-intensive science. Trends in ecology & evolution 27(2):85-93.

Molotch, N. P., S. R. Fassnacht, R. C. Bales & S. R. Helfrich, 2004. Estimating the distribution of snow water equivalent and snow extent beneath cloud cover in the Salt–Verde River basin, Arizona. Hydrological Processes 18(9):1595-1611 doi:10.1002/hyp.1408.

Mooney, P., P. Corcoran & B. Ciepluch, 2013. The potential for using volunteered geographic information in pervasive health computing applications. Journal of Ambient Intelligence and Humanized Computing 4(6):731-745.

Muller, C. L., 2013. Mapping snow depth across the West Midlands using social media-generated data. Weather 68(3):82-82 doi:10.1002/wea.2103.

Muller, C. L., L. Chapman, S. Johnston, C. Kidd, S. Illingworth, G. Foody, A. Overeem & R. R. Leigh, 2015. Crowdsourcing for climate and atmospheric sciences: current status and future potential. International Journal of Climatology 35(11):3185-3203 doi:10.1002/joc.4210.

Nativi, S., P. Mazzetti, M. Craglia & N. Pirrone, 2014. The GEOSS solution for enabling data interoperability and integrative research. Environmental Science and Pollution Research 21(6):4177-4192.

Niemi, K., S. Metsämäki, J. Pulliainen, H. Suokanerva, K. Böttcher, M. Leppäranta & P. Pellikka, 2012. The behaviour of mast-borne spectra in a snow-covered boreal forest. Remote Sensing of Environment 124:551-563.

Nüst, D., C. Stasch & E. Pebesma, 2011. Connecting R to the sensor web. Springer.

OGC, 2012a. OGC WaterML 2.0: Part 1-Timeseries. Open Geospatial Consortium OGC 10-126r4(2.0.1).

OGC, O. G. C., 2006. OpenGIS Web Map Service version 1.3.0 No OGC 06-042 in OpenGIS Standard. Open Geospatial Consortium (OGC).

OGC, O. G. C., 2012b. OGC WCS 2.0 Interface Standard - Core No OGC 09-110r4 in the OpenGIS Standard. OGC.

OpenStreetMap, 2015. In. www.openstreetmap.org.

Parajka, J. & G. Blöschl, 2006. Validation of MODIS snow cover images over Austria. Hydrology and Earth System Sciences Discussions Discussions 3(4):1569-1601.

Parajka, J. & G. Blöschl, 2008. Spatio-temporal combination of MODIS images – potential for snow cover mapping. Water Resources Research 44(3):n/a-n/a doi:10.1029/2007WR006204.

Parajka, J., L. Holko, Z. Kostka & G. Blöschl, 2012. MODIS snow cover mapping accuracy in a small mountain catchment–comparison between open and forest sites. Hydrology and Earth System Sciences 16(7):2365-2377.

Parajka, J., M. Pepe, A. Rampini, S. Rossi & G. Blöschl, 2010. A regional snow-line method for estimating snow cover from MODIS during cloud cover. Journal of Hydrology 381(3–4):203-212 doi:http://dx.doi.org/10.1016/j.jhydrol.2009.11.042.

Peel, M. C., B. L. Finlayson & T. A. McMahon, 2007. Updated world map of the Köppen-Geiger climate classification. Hydrol Earth Syst Sci 11(5):1633-1644 doi:10.5194/hess-11-1633-2007.

Pohl, S., J. Garvelmann, J. Wawerla & M. Weiler, 2014. Potential of a low‐cost sensor network to understand the spatial and temporal dynamics of a mountain snow cover. Water Resources Research 50(3):2533-2550.

R Development Core Team, 2015. R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria.

Ramos, F., A. Monfort & J. Huerta, 2014. A Location Aware Mobile Tool for Direct and Indirect Climate Data Sensors. Transactions in GIS 18(3):385-400.

Ramsay, B. H., 1998. The interactive multisensor snow and ice mapping system. Hydrological Processes 12(10):1537-1546.

Rees, W. G., 2005. Remote sensing of snow and ice. CRC Press.

Reichman, O., M. B. Jones & M. P. Schildhauer, 2011. Challenges and opportunities of open data in ecology. Science 331(6018).

Reis, S., E. Seto, A. Northcross, N. W. T. Quinn, M. Convertino, R. L. Jones, H. R. Maier, U. Schlink, S. Steinle, M. Vieno & M. C. Wimberly, 2015. Integrating modelling and smart sensors for environmental and human health. Environmental Modelling & Software 74:238-246 doi:http://dx.doi.org/10.1016/j.envsoft.2015.06.003.

Ren, R.-z., S.-x. Guo & L.-j. Gu, 2010. Fast bowtie effect elimination for MODIS L1B data. The Journal of China Universities of Posts and Telecommunications 17(1):120-126.

Reusser, D. E., Managing hydrological measurements for small and intermediate projects: RObsDat. In: EGU General Assembly Conference Abstracts, 2014. vol 16. p 10052.

Riggs, G., D. Hall & V. Salomonson, 2006. MODIS snow products user guide to collection 5. Digital Media:80.

Rittger, K., T. H. Painter & J. Dozier, 2013. Assessment of methods for mapping snow cover from MODIS. Advances in Water Resources 51(0):367-380 doi:http://dx.doi.org/10.1016/j.advwatres.2012.03.002.

Rodell, M. & P. Houser, 2004. Updating a land surface model with MODIS-derived snow cover. Journal of Hydrometeorology 5(6):1064-1075.

Ryan, J. A. & J. M. Ulrich, 2011. xts: Extensible time series. R package version 08-2.

Ryberg, K. & A. Vecchia, 2012. waterData—An R package for retrieval, analysis, and anomaly calculation of daily hydrologic time series data, version 1.0. US Geological Survey Open-File Report 1168(8).

Salas, F., E. Boldrini, D. Maidment, S. Nativi & B. Domenico, 2012. Crossing the digital divide: an interoperable solution for sharing time series and coverages in Earth sciences. Natural Hazards and Earth System Science 12(10):3013-3029.

Sample, J. T. & E. Ioup, 2010. Tile-based geospatial information systems: principles and practices. Springer Science & Business Media.

Schenk, E. & C. Guittard, 2011. Towards a characterization of crowdsourcing practices.

Schiller, C., S. Meissl, P. Baumann, S. Krause, G. Triebnig, F. Schindler-Strauss, A. Aiordachioiae, J. Yu & D. Misev, 2011. Introducing WCS 2.0, EO-WCS, and Open Source Implementations (MapServer, rasdaman, and EOxServer) Enabling the Online Data Access to Heterogeneous Multi-dimensional Satellite Data. Wichmann Fachmedien-Angewandte Geoinformatik 2011.

Schnebele, E., G. Cervone & N. Waters, 2014. Road assessment after flood events using non-authoritative data. Natural Hazards and Earth System Science 14(4):1007-1015.

Simic, A., R. Fernandes, R. Brown, P. Romanov & W. Park, 2004. Validation of VEGETATION, MODIS, and GOES+ SSM/I snow‐cover products over Canada based on surface snow depth observations. Hydrological Processes 18(6):1089-1104.

Sirguey, P., R. Mathieu, Y. Arnaud, M. M. Khan & J. Chanussot, 2008. Improving MODIS spatial resolution for snow mapping using wavelet fusion and ARSIS concept. Geoscience and Remote Sensing Letters, IEEE 5(1):78-82.

Slater, A. G., A. P. Barrett, M. P. Clark, J. D. Lundquist & M. S. Raleigh, 2013. Uncertainty in seasonal snow reconstruction: Relative impacts of model forcing and image availability. Advances in Water Resources 55:165-177 doi:http://dx.doi.org/10.1016/j.advwatres.2012.07.006.

Şorman, A., Z. Akyürek, A. Şensoy, A. Şorman & A. Tekeli, 2007. Commentary on comparison of MODIS snow cover and albedo products with ground observations over the mountainous terrain of Turkey. Hydrology and Earth System Sciences 11(4):1353-1360.

Stříž, M., Němec, L., 2011. Spatial analysis of snow data (Prostorová analýza sněhových dat). Paper presented at the 16th annual Slovak Snow Meeting, Žiarska dolina, 23-25.3.2011.

Swain, N. R., K. Latu, S. D. Christensen, N. L. Jones, E. J. Nelson, D. P. Ames & G. P. Williams, 2015. A review of open source software solutions for developing water resources web applications. Environmental Modelling & Software 67:108-117 doi:http://dx.doi.org/10.1016/j.envsoft.2015.01.014.

Tait, A. B., D. K. Hall, J. L. Foster & R. L. Armstrong, 2000. Utilizing Multiple Datasets for Snow-Cover Mapping. Remote Sensing of Environment 72(1):111-126 doi:http://dx.doi.org/10.1016/S0034-4257(99)00099-1.

Tarboton, D., J. Horsburgh, D. Maidment, T. Whiteaker, I. Zaslavsky, M. Piasecki, J. Goodall, D. Valentine & T. Whitenack, Development of a community hydrologic information system. In: 18th World IMACS Congress and MODSIM09 International Congress on Modelling and Simulation, ed RS Anderssen, RD Braddock and LTH Newham, Modelling and Simulation Society of Australia and New Zealand and International Association for Mathematics and Computers in Simulation, 2009. p 988-994.

Tedesco, M., C. Derksen, J. S. Deems & J. L. Foster, 2015. Remote sensing of snow depth and snow water equivalent Remote Sensing of the Cryosphere. John Wiley & Sons, Ltd, 73-98.

Triebnig, G., A. Diamandi, R. Hall, E. Malnes, L. Marklund, S. Metsämäki, T. Nagler, J. Pulliainen, H. Rott & C. Schiller, 2011. CryoLand-GMES service snow and land ice-interoperability, service integration and user access Environmental Software Systems Frameworks of eEnvironment. Springer, 341-348.

Valentine, D., P. Taylor & I. Zaslavsky, WaterML, an information standard for the exchange of in-situ hydrological observations. In: EGU General Assembly Conference Abstracts, 2012. vol 14. p 13275.

Valentine, D., I. Zaslavsky, T. Whitenack & D. R. Maidment, Design and implementation of CUAHSI WATERML and WaterOneFlow Web services. In: Proceedings of the Geoinformatics 2007 Conference, San Diego, California, 2007. p 5-3.

Van Etten, J., 2012. gdistance: Distances and routes on geographical grids. URL http://CRAN R-project org/package= gdistance R package version:1.1-4.

Veatch, W., P. Brooks, J. Gustafson & N. Molotch, 2009. Quantifying the effects of forest canopy cover on net snow accumulation at a continental, mid‐latitude site. Ecohydrology 2(2):115-128.

Walker, J. P. & P. R. Houser, 2005. Hydrologic data assimilation. Advances in water science methodologies.

Walter, M. T., E. S. Brooks, D. K. McCool, L. G. King, M. Molnau & J. Boll, 2005. Process-based snowmelt modeling: does it require more input data than temperature-index modeling? Journal of Hydrology 300(1):65-75.

Wang, J., M. Korayem & D. J. Crandall, Observing the natural world with Flickr. In: Computer Vision Workshops (ICCVW), 2013 IEEE International Conference on, 2013. IEEE, p 452-459.

Whiteaker, T., A. Boddupalli & C. Siler, 2009. HYDROEXCEL 1.1. 1 SOFTWARE MANUAL.

Whitenack, T., 2010. CUAHSI HIS CENTRAL 1.2. Technical report, Consortium of Universities for the Advancement of Hydrologic Science, Inc.

Xin, Q., C. E. Woodcock, J. Liu, B. Tan, R. A. Melloh & R. E. Davis, 2012. View angle effects on MODIS snow mapping in forests. Remote Sensing of Environment 118:50-59.

Yang, J., L. Jiang, J. Shi, S. Wu, R. Sun & H. Yang, 2014. Monitoring snow cover using Chinese meteorological satellite data over China. Remote Sensing of Environment 143:192-203 doi:http://dx.doi.org/10.1016/j.rse.2013.12.022.

Yang, K., X. Cheng, L. Hu & J. Zhang, 2012. Mobile social networks: state‐of‐the‐art and a new vision. International Journal of Communication Systems 25(10):1245-1259.

Zaitchik, B. F. & M. Rodell, 2009. Forward-looking assimilation of MODIS-derived snow-covered area into a land surface model. Journal of Hydrometeorology 10(1):130-148.

APPENDIX A: SOFTWARE AVAILABILITY

The following table shows a listing of the software products that were developed or extended as part of this research. All of the developed software tools are free and open source. Their source code is published under the Massachusetts Institute of Technology (MIT) license.

The WaterML R Package is available on the Comprehensive R Archive Network (CRAN) official R package repository. The website of the package with the installation file and documentation is: https://cran.r-project.org/web/packages/WaterML/. The recommended method of installing the package is directly from inside the R statistical software (http://www.r-project.org). R is a free and open source computational environment, and it runs on all major operating systems including Windows, Mac and Linux. The RStudio (https://www.rstudio.com/) is a free user interface and integrated development environment for efficient work with R. The complete source code of the WaterML is available on the Github repository at: https://github.com/jirikadlec2/waterml.

The HydroServer Lite is a web based software application. Users can set up new instances of HydroServer Lite on the free worldwater.byu.edu data hosting website. Alternatively, HydroServer Lite can be installed on any web server that supports PHP (5.4 or higher), MySQL, and write access to a user-specified folder. Examples of low cost webhosting services with PHP and MySQL support are one.com, webfaction.com and openshift.com. The source code and

installation file of HydroServer Lite is available on the Github repository at: http://hydroserverlite.codeplex.com.

The Snow Inspector is a web based application that can be accessed using all major web browsers on PC and mobile devices. The public website of the application is: https://apps.hydroshare.org/apps/snow-inspector. The application can also be installed on a third-party server by downloading the source code from the GitHub repository (https://github.com/jirikadlec2/snow-inspector). This deployment first requires installing the Tethys platform (http://tethys-platform.readthedocs.org/) on the server. The Snow Inspector also requires installing the pypng https://pypi.python.org/pypi/pypng Python software module. The Tethys platform together with the snow inspector web application have been successfully deployed on an Ubuntu Linux (14.04) and CentOS virtual server. For hosting a custom version of the Tethys platform online, I would recommend a cloud based hosting service such as openshift.com with support for the Docker (docker.com) application container system, large disk space (> 10 GB), and sufficient memory (> 1 GB).

The SnowDataFusion R software package is available in the source code form on the GitHub repository (https://github.com/jirikadlec2/snow-data-fusion). It requires R version 3.2.2 or higher, and RStudio with the devtools R package. The SnowDataFusion package can be installed using the command install_github('jirikadlec2/snow-data-fusion'). To run the development version of the package, several R package dependencies must be installed using the R Studio's Tools – Install Packages… command. These dependencies are: `devtools, sp, rgdal, raster, httr, XML, PresenceAbsence, RColorBrewer,` and `gdistance`.

The output snow probability maps are archived on the HydroShare website (https://hydroshare.org) as a collection of geographic raster resources. The user can search for the list of available maps by typing the keywords "snow probability Czechia". A summary of the software products developed and extended as part of this dissertation is shown in Table A-1.

Table A-1 Open Source Software Products Developed or Modified in this Dissertation

Software name	**System Requirements**	**Software Website**	**Source Code Repository**
WaterML R Package	R (3.0 or higher), RStudio (recommended)	cran.r-project.org/web/packages/WaterML/	https://github.com/jirikadlec2/waterml
HydroServer Lite	PHP (5.3 or higher), MySQL, Linux or Windows server	http://worldwater.byu.edu/app/index.php/rushvalley	http://hydroserverlite.codeplex.com
Snow Inspector	Tethys platform, Linux server with Docker support	http://apps.hydroshare.org/apps/snow-inspector	https://github.com/jirikadlec2/snow-inspector
SnowDataFusion R Package	R (3.2.2 or higher), RStudio (recommended), R package dependencies: devtools, sp, rgdal, raster, httr, XML, PresenceAbsence, RColorBrewer, gdistance	https://github.com/jirikadlec2/snow-data-fusion	https://github.com/jirikadlec2/snow-data-fusion
Snow Probability Map	Web browser	https://appsdev.hydroshare.org/apps/snow-probability/	https://github.com/jirikadlec2/snow-data-fusion