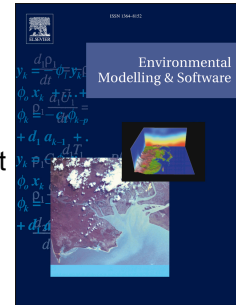


Accepted Manuscript

Map based discovery of hydrologic data in the HydroShare collaboration environment

Zhaokun Xue, Alva Couch, David Tarboton



PII: S1364-8152(17)30597-2

DOI: [10.1016/j.envsoft.2018.09.014](https://doi.org/10.1016/j.envsoft.2018.09.014)

Reference: ENSO 4310

To appear in: *Environmental Modelling and Software*

Received Date: 31 May 2017

Revised Date: 10 March 2018

Accepted Date: 19 September 2018

Please cite this article as: Xue, Z., Couch, A., Tarboton, D., Map based discovery of hydrologic data in the HydroShare collaboration environment, *Environmental Modelling and Software* (2018), doi: <https://doi.org/10.1016/j.envsoft.2018.09.014>.

This is a PDF file of an unedited manuscript that has been accepted for publication. As a service to our customers we are providing this early version of the manuscript. The manuscript will undergo copyediting, typesetting, and review of the resulting proof before it is published in its final form. Please note that during the production process errors may be discovered which could affect the content, and all legal disclaimers that apply to the journal pertain.

Map Based Discovery of Hydrologic Data in the HydroShare Collaboration Environment

Zhaokun Xue^a, Alva Couch^{a,*}, David Tarboton^b

^aComputer Science Department, Tufts University, Medford, MA 02155, USA

^bCivil Engineering, Utah State University, Logan, UT, USA

Abstract

Data discovery refers to the process of locating pre-existing data for use in new research. In the HydroShare collaboration environment for water science, there are more than twenty kinds of data that can be discovered, including data from specific sites on the globe, data corresponding to regions on the globe, and data with no geospatial meaning, such as laboratory experiment results. This paper discusses lessons learned in building a data discovery system for HydroShare. This was a surprisingly difficult problem; default behaviors of software components were unacceptable, use cases suggested conflicting approaches, and crafting a geographic view of a large number of candidate resources was subject to the limits imposed by web browsers, existing software capabilities, human perception, and software performance. The resulting software was a complex melding of user needs, software capabilities, and performance requirements.

Keywords: data discovery, data publication, open data, data sharing

Software Availability

Product Name: HydroShare

Product Type: Web-based

Year first availability: 2015

Developers: The HydroShare Team

Contact: couch@cs.tufts.edu or zhaokun@cs.tufts.edu

Availability: publicly available at <https://www.hydroshare.org/search/>

1. Introduction

Hydrologic data discovery is the process whereby water science researchers gain access to relevant data collected and published by other researchers, perhaps collected for different reasons. With the high cost of data collecting, it becomes important to be able to reuse water data for more than one purpose. Discovering relevant data collected by others has historically been a time consuming process, in which one must review each dataset individually for relevance and usability. Any approach that reduces the amount of manual reviewing would be welcomed by researchers.

1.1. Prior data discovery systems for hydrology

Beran et al [1] proposed one of the first discovery systems for hydrologic data, based upon the Consortium of Universities for the Advancement of Hydrologic Science, Inc. (CUAHSI) Hydrologic Information System (HIS) [2]. This system allowed users to search for and discover time series data of interest to

water researchers. In HIS, all data are public, and all data are time series, i.e., pairs of (time, value) measurements for one of approximately 4000 hydrologic variables that form the CUAHSI controlled vocabulary of variable names [3]. A later data discovery interface for HIS was HydroDesktop [4]: a desktop application for Microsoft Windows that interacts with the HIS data catalog “HISCentral” [5]. HydroDesktop also downloads and manipulates data from HydroServer data servers [6].

This initial work was eventually supplemented via a web-based “CUAHSI Data Access Client” [7] that provides graphical, web-based discovery of HIS water data in the same manner as HydroDesktop but without need for specialized client software installed on a client’s computer. This interface is a mash-up of Google maps and catalog data, that allows one to graphically choose and download data of interest. Features in Google maps and Google Map clustering facility [8] – including clustering of points of interest – are used to ease the task of locating time series of interest, while data faceting provided by an Apache SOLR catalog allows one to filter results and more easily locate data of interest.

As well as the graphical data discovery available in the CUAHSI Data Access Client, much work has been done on metadata completion and matching, so that textual matches are more precise. The CINERGI catalog [9] utilizes aggressive and comprehensive metadata inference to complete incomplete metadata often associated with relatively undocumented data types, resulting in an Elasticsearch [10] catalog.

The contribution of the HydroShare discovery system to this body of work is built upon the ideas in the above discovery systems, including faceted search, Google maps mash-ups, the SOLR search engine, and so on. Our contribution to this work is a mechanism for scalable, interactive, web-based geographic search of a variety of data types, via an innovative user

*Corresponding author. Tel: +1(617)627-3674; Fax: +1(617)627-2227

Email addresses: Zhaokun.Xue@tufts.edu (Zhaokun Xue),

Alva.Couch@tufts.edu (Alva Couch)

interface as described in Section 3. A design problem we addressed was how to depict multiple datasets available for a geographic region without overwhelming the user with visual clutter. This solution evolved from intensive discussions of requirements within the HydroShare community of users/developers.

1.2. The HydroShare Environment

This paper discusses the data discovery mechanisms in HydroShare [11], an innovative environment for sharing hydrologic data. HydroShare is a web-based hydrologic information system for collaborative data collection, management, analysis, and publication. Unlike dedicated data publication environments like DataOne [12], FigShare [13], Harvard Dataverse [14], etc., HydroShare supports not only data publication, but also collaboration on data management and preparation before publication, with the goal of aiding the whole scientific data analysis lifecycle rather than just publishing data. In addition, unlike data discovery systems including FigShare, Harvard Dataverse, CERN Zenodo [15], and Elsevier Mendeley Research Dashboard [16], the HydroShare discovery system described herein supports both list-based and map-based searching. DataOne's map view search divides a map into square regions, and users can zoom in each region to narrow their search results. Instead, HydroShare discovery map view gives users the flexibility to zoom and search in an arbitrary rectangle on the map, and is not limited to predefined sets of rectangles. Unlike the map discovery systems in GEOSS Portal [17] and Open Geoportal [18] that require users to input queries first to get resources shown on the map, the HydroShare discovery map view displays all box and point resources available for the visible map extent, based upon filters including choices for faceted variables and a simple query language based upon SOLR capabilities.

HydroShare manages data as *resources*; each resource is a directory of data files that can contain from one to thousands of files, depending upon the kind of resource. Each resource has a type that comes with structural requirements with which users are required to comply before the data is public. These structural requirements assure that the data is discoverable and reusable for purposes other than originally intended. These requirements result in resources that are compliant with DataOne[12] standards for publication.

The discovery and data sharing mechanisms in HydroShare are built upon the premise that there is value in enabling collaboration between researchers *prior* to data publication. In HydroShare, there are pre-publication sharing mechanisms by which many people can become involved in analyzing and contributing to unpublished and/or partially collected data, even while it remains technically private. Thus, the lifecycle of data has several more steps and possibilities for collaboration than in the classical data publication model.

HydroShare of course supports *publication*, which makes the data publicly available and possible to cite. Data may also be made *public*, which makes it available but does not guarantee that it will not change or be removed in the future. A third option is to make data *discoverable*, which allows everyone to determine existence of resources, but not to access the data itself.

Researchers who know of a resource's existence may contact the resource's owners for access; HydroShare allows owners to share data with individuals and groups without publishing it. Data that is not published, public, or discoverable is *private*; it cannot be discovered through the discovery interface.

In this paper, we discuss the solution to locating discoverable, public, and published data uploaded to HydroShare by other users. This paper is organized as follows. Section 2 discusses design concepts important to discovery, as well as the design of the underlying systems. Section 3 describes and critiques details of user interface design through several design iterations. Section 4 describes lessons learned from this experience, and Section 5 describes conclusions and future work to be done.

2. Materials and methods

This section describes the basic principles upon which the discovery system was based. These include use cases, principles of discovery system design, choice of software components, and how components interact.

2.1. Use Cases

The HydroShare discovery system aims to address several distinct use cases for discovering hydrologic data, including discovery and filtering based upon:

- Dublin core metadata such as abstract, title, author, keywords etc.
- Extent of spatial and temporal coverage.
- Data type and quantity measured.
- Data availability and publication status.

Searching by spatial coverage usually requires a map-based interface, while textual search filters results to match other kinds of metadata.

2.2. Precision and recall

In any discovery system, the goal is to increase both precision and recall. Informally, *precision* is the "fraction of returned records that are relevant". *Recall*, by contrast, is the "fraction of relevant records that are returned"[19]. These goals are in conflict in most discovery systems; raising precision tends to lower recall and vice-versa.

Precision can be improved – hopefully without decreasing recall – by adding filters that make it more likely that matches are desirable. Filters can be based upon data type, geography, and faceting of search terms, as described in the sections below.

2.2.1. Faceting

Of these filters, one of the most effective at increasing precision is “faceted search” [20]. One of the reasons for low precision is that users might not know the keywords being used to describe data of interest. A “facet” can be a list of keywords from which a user can select. For example, “Author” is a facet containing the names of all encountered authors. Selecting one or more authors limits search to those. Other facets include “subject” and “variable”.

Unlike its predecessor CUAHSI HIS, Metadata in HydroShare utilizes both controlled and uncontrolled vocabularies. This allows users greater flexibility in categorizing data, at the expense of not accounting for domain-specific synonyms when searching. In the CUAHSI HIS controlled vocabulary [3], searching for a term searches for all of its synonyms. In HydroShare, for example, hydrologic synonyms “streamflow” and “discharge” appear as separate facet choices for property measured. There are plans to implement synonyms in a future version.

2.2.2. Resource Type

One of the facets in HydroShare discovery is resource type. HydroShare provides the capability to create and share data and hydrologic models of many types [21]. HydroShare supports and allows one to publish several kinds of resources, including Geographic Feature, Geographic Raster, Model Instance, Model Program, Multidimensional, Time Series, and Referenced Time Series. These resource types fall into three distinct categories:

1. Those describing data for a point on the globe (Time Series, Referenced Time Series);
2. Those describing data for an area on the globe (Geographic Feature, Geographic Raster, Multidimensional, Model Instance); and
3. Those with no spatial interpretation (Model Program).

HydroShare also supports several unconstrained resource types, including Generic, Composite, and Collection that can each contain multiple datasets of any of the above types.

2.2.3. Geography

In many earth sciences, the geographic location and context in which data is recorded is important. Hydrology and water resources specifically require a combination of concepts from physics, geography, geology, and atmospheric science, so that the resource types are often geographically linked to specific locations or areas on the Earth’s surface, and refer to one of many related disciplines rather than hydrology in particular. Soil data and weather data are two common examples. One of HydroShare’s most complex resource types – the “SWATShare resource” – contains almost all kinds of data in one resource, for the purpose of running the Soil/Water Assessment Tool (SWAT) model.

The need for geographic filtering suggests use of a map view of discovered data, so that users can search and retrieve spatially defined resources directly from the map. For simplicity, discovered resources are limited to those that appear in the

current map view (at whatever magnification the user selects); this behavior was chosen to be harmonious with the interface for the “CUAHSI Data Access Client” [7], with which HydroShare users are perhaps already familiar.

2.3. Response Time and Usability

Another important consideration in any discovery interface is response time to user queries. If this time is too long, users will stop using the system in favor of faster options. However, some queries intrinsically take a long time. Thus there must be a balance between run time for a query and potential value of the query to the user. In HydroShare discovery, many options proved to be impractical due to excessive response time to user queries on map views, such as initially loading the default map view or querying for all box coverage resources, and the design was adjusted accordingly.

2.4. Tuning Discovery in HydroShare

In this paper, we describe an approach to data discovery that allows users to discover several kinds of data in a large polymorphic data store. Filters and facets are used to reduce the number of results; geographically located data can be filtered by geography. Unlike CUAHSI HIS, which returns results for one kind of data, the HydroShare discovery page that this paper describes attempts to show all resources – regardless of their types – on a single map view. This poses visualization challenges when the number of candidate datasets is large. Thus, a central question in this paper is how one can best visualize and explore data of multiple types in a specific geographic region, while maintaining a reasonable response time to user requests, as well as high precision and high recall.

2.5. User-driven design and development

One of the unique characteristics of this effort is that we had access to constant feedback from a community of developers and designers who were also potential users. Design iterations were discussed as a community, using the “GitHub issue tracker” (that we use for overall discussion of HydroShare features) as a discussion forum to weigh alternatives and critique current implementations. Thus, the design evolved through several iterations based upon user feedback, as discussed below.

2.6. Design

HydroShare is written using the Django framework [22] for the python programming language. We implemented the discovery page as a Django view (web page) in the existing HydroShare project. We chose Apache SOLR [23] as a search engine due partly to its ability to support faceted search, and because there is an existing Django module Haystack [24] that links Django and SOLR. as illustrated in Figure 1. The architecture of the discovery page includes HTML views loaded from Django; these interact with a set of Representative State Transfer (REST) [25] responders that are called via asynchronous JavaScript (AJAX) [26] calls. These responders relay information from both the Django database and the SOLR search engine, as needed.

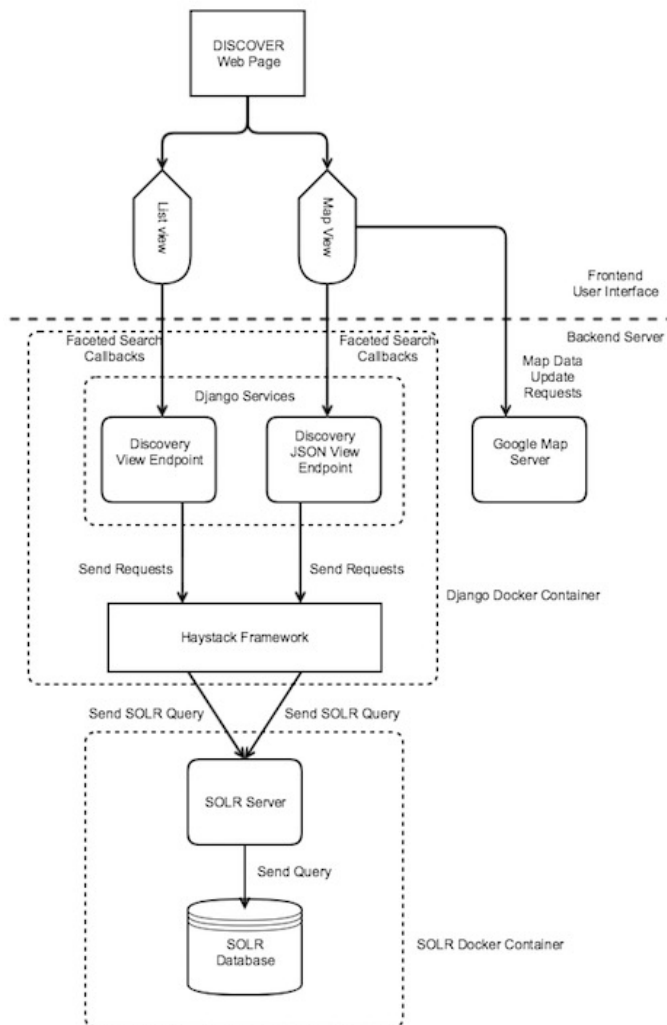


Figure 1: Software design includes Django, SOLR, and Haystack components.

As HydroShare is implemented as a set of Docker [27] containers, we created a Docker container containing a SOLR instance in the HydroShare Docker system. This SOLR container handles data indexing work for the Apache SOLR search engine [23]. As users interact with the Django database, the Django application forwards these changes to the SOLR database, which modifies search results.

This basic scheme – Django, SOLR, Haystack, and Google Maps – provides many capabilities but also many limits. The final design was a matter of intelligently choosing how these components interact. Generic approaches provided for chosen components did not function adequately, and significant thought was given to how to make this interaction work well without rewriting any one of the components.

In Figure 1, Haystack is the glue between Django and SOLR; interfacing Django and SOLR requires telling Haystack what to index of the massive amount of data in Django, as well as which fields to index for faceted search. The system indexes all Dublin Core metadata fields as well as user access information – such as owners’ names – for resources. SOLR initially faceted a small number of metadata fields from the Dublin Core standard for metadata [28], including “Author” and “Subject”, as well as system metadata including “Resource Type”, “Owner” and “Availability” (public, published, or discoverable). Currently, SOLR indexes and facets a subset of metadata fields from specific resource types, including the variable name, which indicates the quantity measured.

2.6.1. List and Map Views

When users first load the discovery page, it shows users a text search bar on the top and a left-side menu displaying the facets for discoverable resources as depicted in Figure 2. In this figure, the user has chosen the value “David Tarboton” for the facet “Author” with a mouse, and all public, published, or discoverable resources for which “David Tarboton” is listed as author are shown on the right. The user can also specify search terms by clicking upon and typing into the box at the top, or time ranges by clicking upon the date range fields below that. Updating search terms or time ranges updates all facets; choosing a facet term just updates the list of results.

The page provides two tabular views to the right of the facets: list view and map view. Tabs are used to switch between these views. The list view is used for displaying all results satisfying the user’s search queries, while the map view shows all available spatially located resources in some geographic region. The map view is provided via the Google Maps API [29]. The user can proceed by typing search terms or selecting facets, or both. Results are updated as the user makes selections. This basic scheme – facets on the left and results on the right – did not change during discovery tuning.

2.6.2. Faceting, Performance, and Usability

In Django Haystack, the default behavior when a user chooses an item from a set of faceted values is to update the contents of all faceted categories based on this choice, so that the whole discovery page is refreshed after each faceting operation. Haystack JSON response can take considerable time,

Discover *Public resources shared with the community.*

Q Search All Public and Discoverable Resources

Show All

Filter by Author

- Christina Bandaragoda 85
- David Tarboton 56
- Joanne Greenberg 34
- John Schaake 33
- GeoTrust CDM 29
- Laurence Lin 24
- Lorne Leonard 24

Filter by Subject

- iUTAH 123
- GAMUT 99
- time series 86
- raw data 55
- Stream 38
- Temperature 36
- NFIE 32

Filter by Resource Type

- Generic 490
- Geographic Feature (ESRI Shapefiles) 88
- Collection Resource 40
- Geographic Raster 39
- HIS Referenced Time Series 36
- Model Program Resource 33

Filter by Owner

- iUTAH Data Manager 125

Temporal Coverage

From Date: To Date:

Sort Order

Sort By: Title Sort Direction: Ascending

List Map


























Type	Title	First Author	Date Created	Last Modified
  	A Conceptual Framework for the National Flood Interoperability Experiment (NFIE)	David Tarboton	Jul 29, 2015 at 1:52 a.m.	Jul 29, 2015 at 1:54 a.m.
  	Clearing your Desk! Software and Data Services for Collaborative Web Based GIS Analysis	David Tarboton	Dec 13, 2015 at 12:17 a.m.	Dec 18, 2015 at 3:50 p.m.
  	Clearing your Desk! Software and Data Services for Collaborative Web Based GIS Analysis	David Tarboton	Nov 18, 2015 at 2:21 p.m.	Nov 18, 2015 at 5:27 p.m.
  	Collection of Great Salt Lake Data	David Tarboton	Apr 23, 2017 at 1:34 p.m.	Jun 04, 2017 at 9:28 a.m.
  	Collection of workshops using HydroShare at the CUAHSI biennial symposium, July 2016	David Tarboton	Jul 21, 2016 at 6:02 p.m.	Jun 06, 2017 at 11:56 p.m.
   	Data and Models as Social Objects in the HydroShare System for Collaboration in the Hydrology Community and Beyond	David Tarboton	Dec 12, 2016 at 9:58 p.m.	Dec 14, 2016 at 7:07 a.m.
  	Digital Elevation Model based Hydrologic And Water Resources Analysis	David Tarboton	May 15, 2017 at 3:03 p.m.	May 15, 2017 at 3:39 p.m.
  	Example of Height above Stream Flood Inundation Mapping Approach	David Tarboton	Nov 28, 2015 at 1:12 a.m.	Aug 18, 2016 at 1:26 p.m.

Figure 2: The list view of the discovery interface depicts all matches as a list, with facets to left and matches to right.

which unacceptably impairs website performance. Based on performance experiments on the map view, the discovery page loading time increases logarithmically with number of matches.¹

Aside from performance impacts, many users found the constant reloading of facets confusing; they easily forgot the choices they had already made, and even what they were doing. To address these issues, we modified default Haystack behavior and arranged for facets to be updated only when the search text in the search bar is changed and not when facets are selected or de-selected. Thus, facet choices always reflect the outcome of searching for the text in the search bar.

2.6.3. Design and Performance

In order to optimize our map's performance, we attempt to minimize the number of AJAX calls. An AJAX back-end update is only requested when a user inputs new search text or selects a new keyword facet, time range, or sort order. Front-end map interactions such as dragging the map, zooming in or out the map, etc., utilize front-end JavaScript filtering to show results within the current map extent without invoking a new back-end search, whenever possible. Although the back-end SOLR query and network communication for JSON objects takes a very short time, time for forming JSON objects and rendering the client display takes a relatively longer time, which discouraged us from making all filtering requests AJAX (back-end) updates. We analyzed the performance experiments by using "Performance" and "Network" panels from Chrome DevTools [30]. We used the "Content Download" time in "Network" as our network communication time, and we take the sum of "Loading", "Scripting", "Rendering" and "Painting" in Performance as our client rendering time. From detailed performance measurements depicted in Figure 3, we conclude that time for server computation and client rendering dominate the time for AJAX calls; SOLR query time is negligible.

2.6.4. Updating the SOLR Index

SOLR functions by "indexing" the metadata from Django on a regular basis; the recommended approach is to index data nightly. Only data that appears in the index will be returned as the result of a query. Initially we thought that this nightly update to the SOLR index would be sufficient, but user feedback indicated that real-time response was important, especially when making a resource public for the first time and when making a (formerly) public resource private. Without such dynamic record indexing, users' new public resources seem not to be public until the daily indexing update happens. At first, we thought we just needed to turn on Haystack's provided

¹With many thanks to one reviewer of an initial draft of this paper, we discovered a subtle performance bug in the map view that dramatically affected performance results. The code was not just using SOLR, but also, loading thousands of records via Django. The impact of this bug was substantive; some response times changed from five minutes to five seconds! While this performance improvement does not suggest changes in design, it provides a dramatic improvement in usability.

"Real Time Signal Processor" to enable dynamic data harvesting. However, the default recipe for this did not work properly for our particular Django database, because the class being changed is often a subclass of the indexed class. To adapt this real-time data harvesting feature for our specific needs, we provided our own save and delete functions for the Haystack "Real Time Signal Processor". These functions add a resource to the index only when it becomes discoverable or public, and remove a resource from the index when it goes from discoverable or public to private and when it is removed from the Django database. Thus, discovery immediately responds to each change in resource status.

2.6.5. Software sustainability concerns

This project was undertaken specifically to create sustainable software that can be managed in perpetuity by the CUAHSI Water Data Center alongside CUAHSI HIS. That specific goal constrained our design in several ways. The choice of SOLR was partly motivated by CUAHSI expertise in this area. Sustainable software must also avoid re-engineering stable and proven solutions. Thus – as much as we might have been tempted – we chose to use Google Maps and its add-ons as-is and not to re-engineer its complex internals to our liking. For example, current map systems could not provide a way to cluster areas. This limited our choices but the final result is more sustainable. Thus, our designs were limited to what the unmodified versions of complex components can accomplish.

2.7. Design summary

The design of the discovery interface was motivated by multiple factors, including conformance to the behavior of prior interfaces such as the "CUAHSI Data Access Client", performance and user interface responsiveness, user expectations, pre-existing models of discovery, and sustainability concerns. Thus, the final discovery interface was a balance between all of these concerns.

3. Results

Although the overall system design described above remained the same for the lifetime of the project, there were many iterations on the structure of the user interface. After implementing the first couple of versions, we collected users' feedback from alpha testers and made modifications based on that feedback. Users found some approaches confusing, but also disagreed about what was confusing and what was acceptable, based upon their backgrounds. For example, Geographical Information System (GIS) users found GIS-style interfaces desirable while users with little experience with GIS found them confusing. As well, there was significant conflict between user desires and performance limits; some of the things users desired, such as querying for resources on the map, made response to user input unacceptably slow. As the group of users providing feedback were a broad sample of the kind of users HydroShare attracts, resolving the conflicts in user feedback was both critical to project success and quite challenging.

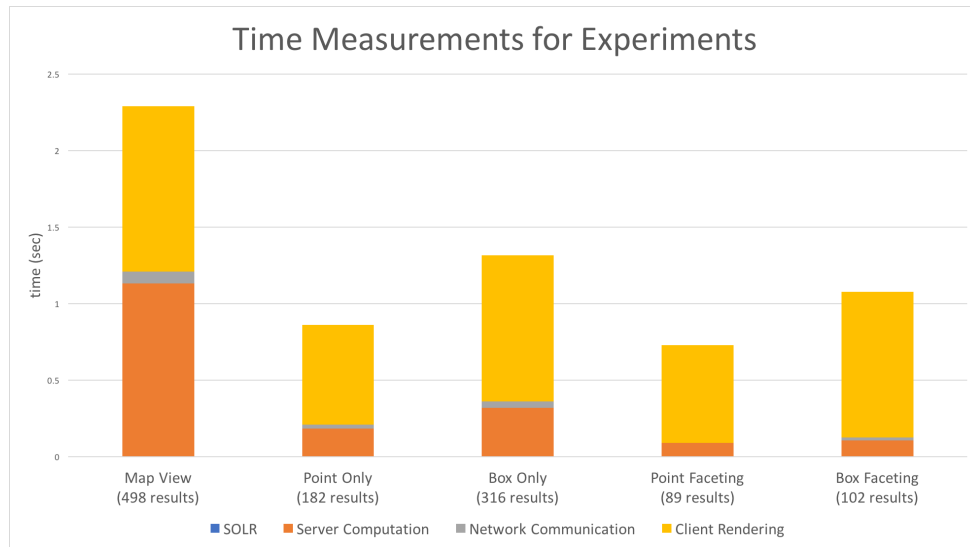


Figure 3: Average time for map loading of both points and boxes, only points, only boxes, faceting on Chris Cox’s point results and faceting on Geographic Feature(ESRI Shapefiles) box results, for a total of 10 trials. SOLR query time is too small to appear on the diagram.

3.1. Visualizing resources on a map

HydroShare contains resources that describe a point on the globe, as well as resources that describe areas. Spatial coverage is one of the metadata elements that HydroShare uses. HydroShare is currently limited to recording rectangular coverages and does not at present support polygonal coverages for performance reasons. We sought to depict both “point coverages” and “box coverages” on a map for geographic filtering. A key design challenge was how to depict these coverages on the map.

Since each time series is data about a point on the map, Google maps was well-suited to depict available time series and other “point coverages” as clickable markers on the map, using the clustering feature for point markers available as an extension to Google Maps to allow one to zoom from large to small geographic areas. When users click on a site marker, an information window appears with a link to the corresponding resource description page(s).

By contrast, the design of the map interface for “box coverages” on the map was controversial, and much prototyping and debate with potential users ensued about how to depict these resources on the map. Metadata for these resources includes a bounding rectangle of the area for which the resource has data, but shape files such as polygons are not depicted to keep depiction acceptably responsive. We refer to these resources as possessing “box coverages”, while time series data are an example of a “point coverage” documenting the measurements at a single point on the map. Several solutions for this problem were tried, as discussed below.

A simple iconography for different kinds of map objects did not change during subsequent experiments. Point coverages are depicted by red markers while the locations of box coverages are depicted by blue markers. Clusters of markers are depicted in yellow. These clusters can potentially contain markers for both point coverages and box coverages.

3.1.1. Describing boxes

We tried several approaches to describing boxes, and informally polled users on their opinions of each.

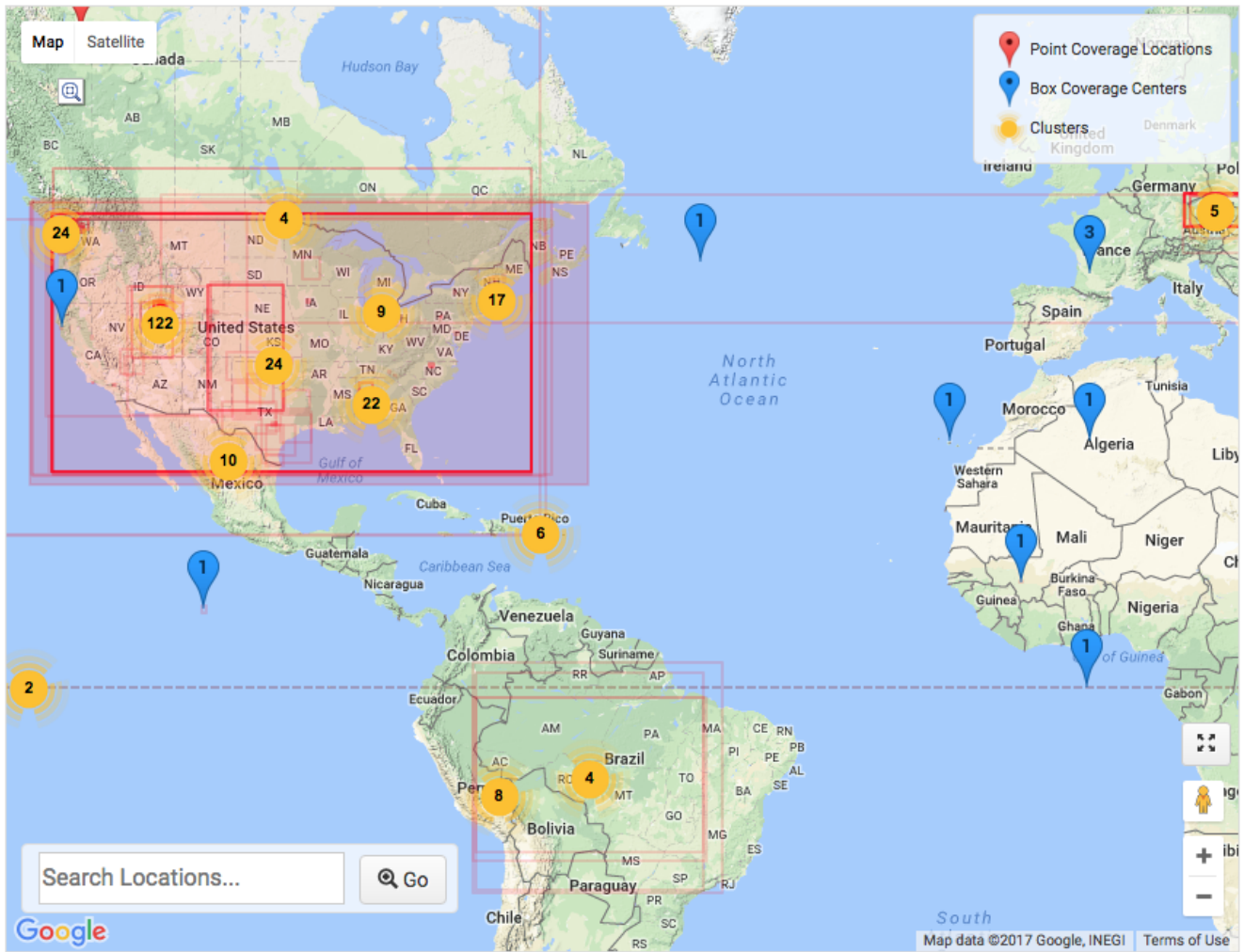
The first idea we tried was to represent all box coverage resources by their centers. Thus, a box coverage is depicted just like a point coverage, with an icon in a distinct color and a list of the coverages to the right. At first glance, the map was easy to understand. However, box coverage resources can have the same center with very different coverage areas, that are local, national, or even global (for satellite data). For instance, a resource that covers the whole area of the United States could have the same center as a resource that only covers a small area around the center of the United States.

To avoid the problem of small boxes disappearing at low zoom levels, we depicted filled boxes along with center markers. Depicting filled boxes looked too cluttered to users, and we settled upon depicting the border of each box instead of shading it. Using additive overlays, multiple overlapping boxes darken common borders.

3.2. Current Solution

When users first load the map view, we display all point coverages, box coverage centers, and box borders as shown on Figure 4. Box coverage centers and time series sites are both clustered using the Google Maps clustering facility [8]. These give users a quick way to check the details for a specific resource, while the markers for box coverages indicate the existence of coverages that are too small to show up on the map until the view is zoomed. We also show a table under the map which lists details for resources selected on the map, as shown on Figure 4.

When users click a marker on the map, the corresponding resource’s region is highlighted and a pop-up window above the marker displays links to all resources associated with the marker, as depicted on Figure 5. Clicking upon any point in an



Search:

Show on Map ▲	Resource Type ▲	Title	First Author ▲
	Generic	gSSURGO-based Floodplain Map for the Continental United States	Venkatesh Merwade
	Collection Resource	LNWB Ch06 Soil Processes and Inputs	Christina Bandaragoda
	Geographic Feature (ESRI Shapefiles)	LNWB Ch06 Soil Processes and Inputs - STATSGO soil spatial data	Christina Bandaragoda
	Collection Resource	Lower Nooksack Water Budget (LNWB)	Christina

Showing 1 to 6 of 6 entries

Figure 4: Clicking on a point on the map highlights all overlapping boxes and lists details in a table.

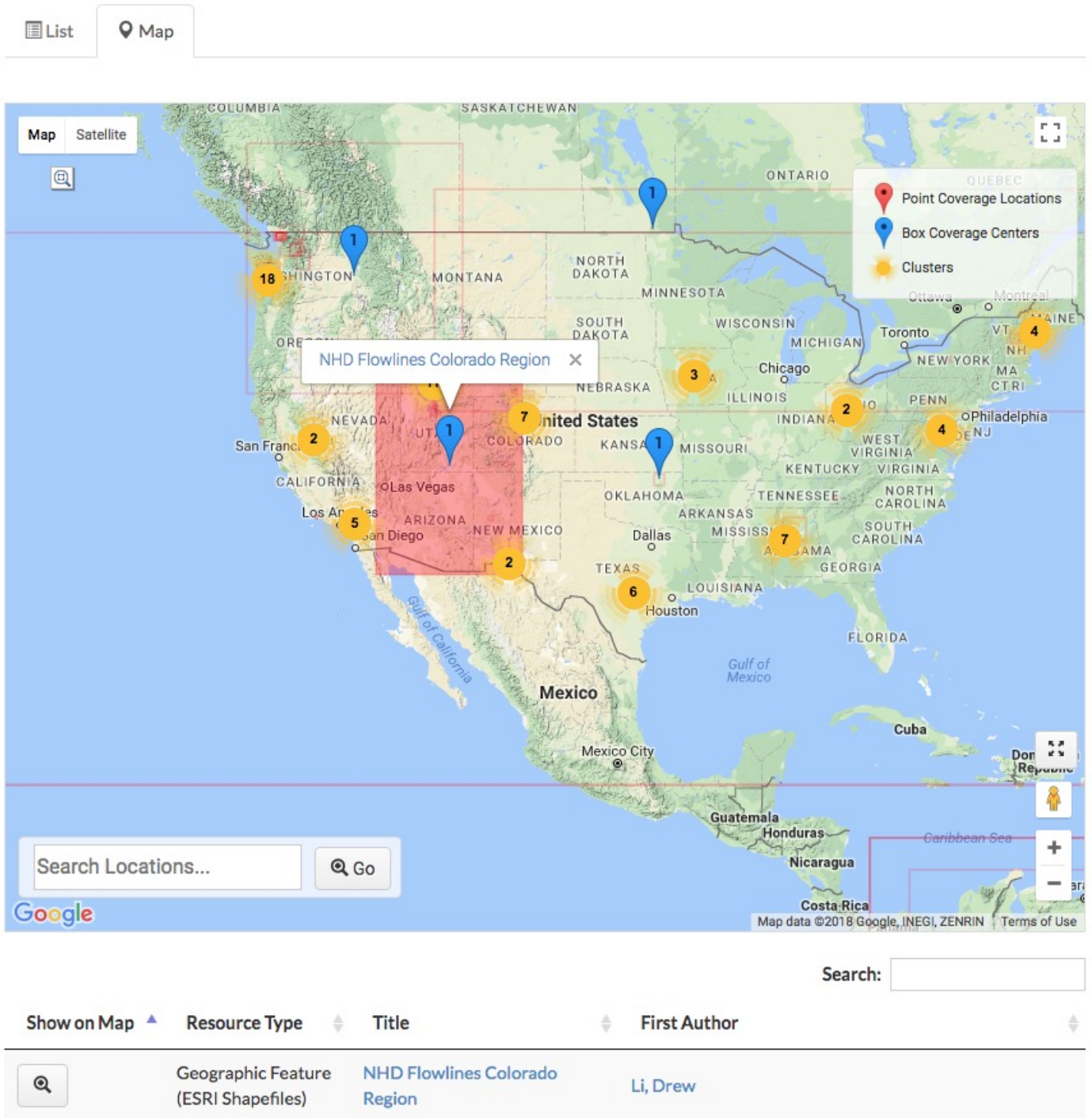


Figure 5: Clicking upon a box coverage center marker highlights the coverage area and displays a pop-up window describing the selected resources.

area highlights and lists all box coverages containing that point, as shown on Figure 4.

This interface remains controversial. Some users find it acceptable; others find it quite confusing to have to click on an unmarked point on the map to see anything in the list.

3.3. *Finer Points of Implementation*

Many details of implementation were dictated by performance requirements. The need for quick response to various kinds of queries – as perceived by a user – led to several development practices to decrease response time.

We used Google Maps and its utility libraries to build our map view. In order to improve performance for the map, we used the Polygon Overlay for drawing the rectangles instead of using Rectangle Overlay. These appear exactly the same on the map but according to our experience, the time for depicting a Polygon Overlay was (counter-intuitively) less than the time for depicting a Rectangle Overlay.

The delays from querying SOLR were larger than expected and required several design decisions. Initially, it took a relatively long time to initially load a map, and we needed to minimize the number of load requests. Thus, we did much search filtering client-side in JavaScript rather than re-querying the server. As well, we made all server queries asynchronous via AJAX, using REST endpoints on the server and the JQuery [31] library on the client (in the browser). More recently, we were able to dramatically reduce the SOLR response time, but the design decisions due to slow SOLR response remain.

4. Discussion

Implementing discovery for HydroShare – and balancing all of the conflicting goals mentioned above – was a surprisingly difficult problem. Meanwhile, users wanted a more expressive search interface that exhibits more precision and recall. These are – to some extent – conflicting goals.

Post-implementation discussion of the current solution indicates that user opinion remains quite divided on the effectiveness of the interface, perhaps due to the specific expertise and personal experience of each user. GIS users mostly like the new system, as it provides a familiar environment where most functions are intuitive, while the environment remains relatively difficult to learn to use for non-GIS users, for whom its conventions are not intuitive. Non-GIS users find the clutter on the maps confusing. They argue that in typical use, a user is looking for either point or area coverages, but not both, and the clutter on the map does not aid – and potentially impedes – the most common use cases. Some users also find the clicking upon a map to view all overlapping box coverages to be counter-intuitive and difficult to remember.

There was a large amount of “churn” in the development process, in the sense that open discussion of the solutions identified distinct groups of potential users who held differing opinions that caused the discussion to oscillate between several different solutions, sometimes returning to a previous implementation. In this process, the potential users were not a homogeneous population, use cases conflicted, and one solution could

not completely please all stakeholders. The uses in conflict included:

- a) Locating a very specific kind of resource to plug into a known model in a specific geographic region.
- b) Characterizing all resources available for a geographic region and/or characterizing the regions for which specific kinds of resources are available.

Initial testing shows that our approach is usable for use case(a) but not use case(b). The display is quite usable when the number of matches is limited, by specifying other search data including choosing facets or refining queries, but remains crowded and difficult to traverse without this refinement. While a small number of researchers are actively engaged in the second kind of activity, the vast majority are engaged in the first. Thus, the interface is likely too difficult to learn for the majority of users.

Thus, we learned during this process that it is exceedingly important to keep the potential use cases in mind, as well as the relative sizes of the user populations that will utilize the discovery service for particular use cases. Much controversy in discussions about the discovery interface was based not on whether it was good or bad, but instead, whether it enables particular use cases that were not – at the time – explicitly mentioned in the conversation. Thus, the important thing to which to pay attention is not the experience of users – but rather – their intent in using the system.

4.1. *Successes*

Compared with previous trials, we believe that the final solution makes remarkable improvements on users’ experience of interacting with the map, especially for searching box coverage resources. The rectangle outlines on the map give users a reasonable and readable indication on where the boxes are located. Users can easily perceive the overlapping relationships and distribution density of box coverage resources from the map, because darker boxes indicate overlaps and that more coverages are present. This solution addresses somewhat some concerns of scalability of box coverage depictions, such as dealing with large numbers of overlapping boxes, in our previous designs.

4.2. *Critique*

After the final solution was released and tested by users, we collected informal feedback and suggestions for future improvements on scalability and usability. One of the main complaints of users is that the map view loads too slowly. If the user requests all resources in the view area, it still takes a relatively long time to load all contents of the map, which discourages users from using the map view.

As for usability, some members of the HydroShare development team believe that depicting overlapping regions – and clicking on a point to determine overlaps – is difficult for new users to learn to interpret. They suggest that instead of showing overlapping regions by clicking on a point, we should show no regions by default, and list box coverages that overlap current extent in a table below the map. Users can highlight and zoom to a specific box resource by clicking on an item from the table.

5. Conclusions and future work

One of the premises of our work is that special-purpose discovery systems based upon the needs of a specific discipline can better serve the discipline than general purpose discovery. In acting upon this premise, we observe that the main use cases often require different discovery solutions and serving them via one discovery interface can be impractical. The map view in our discovery system contributes a potential approach for searching different types of geographic data (point and box coverage resources) just on one map interface. On the map view, users can discover both point and box type resources in a specific geographic region at the same time. We plan to provide an auto-complete search feature for queries, which is a query language that – among other capabilities – allows users to select only box or point resources on the map. For the specific purposes of our website, we believe our work provides users a usable solution for searching for polymorphic geographic data in one discovery system.

One conclusion of this work is that the map interface alone is insufficient to provide a scalable discovery solution. We are very much dependent upon being able to reduce the number of search matches via faceted search and complex queries. Without these, the map interface does not perform well.

There is much room for improvement as described in the critique above. The discovery system needs to become more responsive, and needs to understand some of the nuances of hydrologic terms in order to list resources in order of probable relevance. Sorting by hydrologic term relevance is a feature of SOLR that has not yet been enabled. Likewise, many well-known synonyms for hydrologic terms have not been defined for SOLR.

Also, the lack of a controlled vocabulary for variable names in HydroShare – in contrast with CUAHSI HIS – makes it impossible to utilize scientific synonyms with the same facility as in CUAHSI HIS. The key to CUAHSI HIS indexing is use of a controlled hydrologic vocabulary [3] that is used to describe data content. Other controlled vocabularies, including Climate and Forecast (CF) vocabularies [32] and the Global Change Master Directory (GCMD) of science terms [33], should also be usable to specify search terms. Future enhancements to HydroShare will take advantage of controlled vocabularies, not only from CUAHSI HIS, but also from other aligned geoscience fields, such as CSDMS standard names [34, 35], which are broadly used to describe atmospheric data that might be uploaded to HydroShare.

This work is not the end product, but rather, the beginning of a broader optimization. As this interface is used by the broader water sciences community, we expect user feedback to again mold the project to user needs, and we are confident that the design decisions made here are sustainable as CUAHSI takes management responsibility for the software in the near future.

Work on HydroShare discovery continues. This paper describes the interface of the soon-to-be-released beta version as of March 10, 2018. The version number of this version is 1.15.

6. Acknowledgements

This work was supported by the National Science Foundation under collaborative grants ACI 1148453 and 1148090 for the development of HydroShare (<http://www.hydroshare.org>). Any opinions, findings, conclusions, or recommendations expressed are those of the authors and do not necessarily reflect the views of the NSF. We thank our colleagues and developers from the HydroShare project and from CUAHSI, who provided efforts and expertise that greatly assisted the work. Specifically, Richard Hooper provided much valuable feedback. We also thank the developers of the CUAHSI Water Data Center for their help and support during this implementation process, including Martin Seul and Yaping Xiao. Michael Stealey of the Renaissance Computing Institute at the University of North Carolina was instrumental in installing the docker container for SOLR inside HydroShare.

References

- [1] B. Beran, M. Piasecki, Engineering new paths to water data, *Computers & Geosciences* 35 (4) (2009) 753 – 760, geoscience Knowledge Representation in Cyberinfrastructure. doi:10.1016/j.cageo.2008.02.017. URL <http://www.sciencedirect.com/science/article/pii/S0098300408000988>
- [2] D. G. Tarboton, J. S. Horsburgh, D. R. Maidment, T. Whiteaker, I. Zaslavsky, M. Piasecki, J. Goodall, D. Valentine, T. Whitenack, Development of a community hydrologic information system, in: R. S. Anderssen, R. D. Braddock, L. Newham (Eds.), 18th World IMACS Congress and MODSIM09 International Congress on Modelling and Simulation, Modelling and Simulation Society of Australia and New Zealand and International Association for Mathematics and Computers in Simulation, 2009, pp. 988–994.
- [3] J. S. Horsburgh, D. G. Tarboton, R. P. Hooper, I. Zaslavsky, Managing a community shared vocabulary for hydrologic observations, *Environmental Modelling & Software* 52 (0) (2014) 62 – 73. doi:10.1016/j.envsoft.2013.10.012. URL <http://www.sciencedirect.com/science/article/pii/S1364815213002557>
- [4] D. P. Ames, J. S. Horsburgh, Y. Cao, J. Kadlec, T. Whiteaker, D. Valentine, Hydrodesktop: Web services-based software for hydrologic data discovery, download, visualization, and analysis, *Environmental Modelling & Software* 37 (0) (2012) 146 – 156. doi:10.1016/j.envsoft.2012.03.013. URL <http://www.sciencedirect.com/science/article/pii/S1364815212001053>
- [5] T. Whitenack, CUAHSI HIS central 1.2, Tech. rep., Consortium of Universities for the Advancement of Hydrologic Science, Inc (2010). URL http://his.cuahsi.org/documents/HIS_Central-1.2.pdf
- [6] J. S. Horsburgh, D. G. Tarboton, K. A. T. Schreuders, D. R. Maidment, I. Zaslavsky, D. Valentine, Hydrosriver: a platform for publishing space-time hydrologic datasets, in: AWRA 2010 Spring Specialty Conference, 2010.
- [7] CUAHSI, The cuahsi data access portal, accessed: 2017-05-30 (2017). URL <http://data.cuahsi.org>
- [8] Google map clustering library, accessed: 2017-11-10 (2017). URL <https://developers.google.com/maps/documentation/javascript/marker-clustering>
- [9] The cinergi community inventory of earthcube resources for geosciences interoperability, accessed: 2017-05-30 (2017). URL <https://earthcube.org/group/cinergi>
- [10] Open source search analytics elasticsearch, accessed: 2018-02-17 (2018). URL <https://www.elastic.co/products/elasticsearch>
- [11] T. H. Team, Hydroshare, accessed: 2017-05-30 (2017). URL <https://www.hydroshare.org>

- [12] Dataone, accessed: 2017-11-10 (2017).
URL <https://www.dataone.org/>
- [13] Figshare, accessed: 2017-11-10 (2017).
URL <https://figshare.com/>
- [14] Harvard dataverse, accessed: 2017-11-10 (2017).
URL <https://dataverse.harvard.edu/>
- [15] Cern zenodo, accessed: 2017-11-10 (2017).
URL <https://zenodo.org/>
- [16] Elsevier mendeley research dashboard, accessed: 2017-11-10 (2017).
URL <https://www.elsevier.com/authors/journal-authors/measuring-an-articles-impact/stats>
- [17] Geoss portal, accessed: 2018-02-17 (2018).
URL <http://www.geoportal.org/>
- [18] Open geoportal, accessed: 2018-02-17 (2018).
URL <http://data.opengeoportal.org/>
- [19] S. M. Shafi, R. A. Rather, Precision and recall of five search engines for retrieval of scholarly information in the field of biotechnology, accessed: 2018-02-17 (2018).
URL <http://www.webology.org/2005/v2n2/a12.html>
- [20] A. Bedig, A. Couch, M. Piasecki, Faceted search for hydrologic data discovery, in: Proc. 10th International Conference on Hydroinformatics, 2012.
- [21] J. S. Horsburgh, M. M. Morsy, A. M. Castronova, J. L. Goodall, T. Gan, H. Yi, M. J. Stealey, D. G. Tarboton, 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 52 (4) (2016) 873–889. doi:10.1111/1752-1688.12363.
- [22] Django – the web framework for perfectionists with deadlines, accessed: 2017-05-30 (2017).
URL <https://www.djangoproject.com/>
- [23] Solr is the popular, blazing-fast, open source enterprise search platform built on apache lucene, accessed: 2017-05-30 (2017).
URL <http://lucene.apache.org/solr>
- [24] Welcome to haystack! haystack 2.4.1 documentation, accessed: 2017-05-30 (2017).
URL <https://django-haystack.readthedocs.io/en/v2.4.1/>
- [25] Wikipedia, representational state transfer, accessed: 2017-05-30 (2017).
URL https://en.wikipedia.org/wiki/Representational_state_transfer#cite_note-1
- [26] M. D. Network, Ajax, accessed: 2017-05-30 (2017).
URL https://developer.mozilla.org/en-US/docs/AJAX/Getting_Started
- [27] Docker, accessed: 2017-05-30 (2017).
URL <https://www.docker.com/>
- [28] Dublin core metadata element set, version 1.1, accessed: 2017-05-30 (2017).
URL <http://dublincore.org/documents/dces/>
- [29] G. Developers, Google maps javascript api, accessed: 2017-05-30 (2017).
URL <https://developers.google.com/maps/documentation/javascript/>
- [30] Chrome devtools overview, accessed: 2018-02-17 (2018).
URL <https://developer.chrome.com/devtools>
- [31] JQuery, accessed: 2017-05-30 (2017).
URL <https://jquery.com/>
- [32] Climate and forecast (cf) standard names parameter vocabulary, accessed: 2018-03-08 (2018).
URL <https://www.w3.org/2005/Incubator/ssn/ssnx/cf/cf-property>
- [33] Global change master directory (gcmd), accessed: 2018-03-08 (2018).
URL <https://earthdata.nasa.gov/about/gcmd/global-change-master-directory-gcmd-keywords>
- [34] S. D. Peckham, The csdms standard names: Cross-domain naming conventions for describing process models, data sets and their associated variables, in: 7th International Conference on Environmental Modeling and Software, International Environmental Modelling and Software Society, 2014.
- [35] CSDMS, Csdms standard names, accessed: 2017-05-30 (2017).
URL <http://lucene.apache.org/solr>

- Data discovery of many kinds of typed data requires map depictions that are quickly rendered, understandable, and precise.
- Providing these map depictions requires a balance between user expectations, performance requirements, and capabilities of existing software.
- The HydroShare data discovery environment achieves such a balance via user-driven design, including regular feedback to the developers from potential users.
- The map solution includes a map depiction of both point and area coverages, in which areas are depicted via a novel box boundary depiction.
- This map – together with a powerful pre-map filtering mechanism – provides one solution to browsing large data collections using maps.
- The solution remains controversial and there was much “churn” during development due to conflicting user needs.
- Resolving such conflicts requires careful attention to intended uses that lead users to desire different solutions.