HYDROSEEK: An Ontology-Aided Data Discovery System for Hydrologic Sciences

A Thesis

Submitted to the Faculty

of

Drexel University

by

Bora Beran

in partial fulfillment of the

requirements for the degree

of

Doctor of Philosophy

September 2007

DEDICATIONS

I dedicate this work to my wonderful parents,

my mother Inci Beran

and

my father Timucin Beran.

ACKNOWLEDGEMENTS

I would like to express my appreciation to my advisor Dr. Michael Piasecki for believing in my ideas and supporting my work. I’d like to acknowledge the guidance and support from all the members of my committee; Dr. Charles Haas, Dr. David Lanter, Dr. Michael Piasecki, Dr. Richard Weggel and Dr. Ilya Zaslavsky. I would like to thank the members of Consortium of Universities of Advancement of Hydrologic Science Inc. (CUAHSI) for their support for this work, Dr. David Lanter for his endless supply of books on Artificial Intelligence, Dr. Saiful Islam for many productive discussions during the early years of my studies.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

HYDROSEEK: An Ontology-Aided Data Discovery System for Hydrologic Sciences
Bora Beran
Michael Piasecki, Supervisor, Ph.D.

Search engines have made considerable contributions to the overall web experience. However locating scientific data remains a problem since databases are not readily accessible by search engine bots. Considering different temporal, spatial and thematic coverage of different scientific data repositories, especially for interdisciplinary research it is typically necessary to work with multiple data sources. Today integration of hydrologic data sources are mostly at the level of content aggregation by providing links to several data providers on a web page. However being able to query multiple databases simultaneously is a feature that has been sought after since the first data repositories; USGS' National Water Information System (NWIS) and EPA's Storage and Retrieval System (EPA STORET) came online. This study examines the current state of hydrologic data availability and dissemination in the US. It identifies the data accessibility problem and suggests a data discovery mechanism named *Hydroseek* as a solution. *Hydroseek* enables querying multiple hydrologic data repositories through a single interface and effectively combines spatial, temporal and thematic aspects of search in order to make it possible to discover more of the desired data in less time. It provides a unified view despite heterogeneity issues within and among data repositories, allows data discovery using keywords which eliminates the need to know source specific parameter codes, improves data browsing capabilities by incorporating data classification based on conceptual hierarchy and has an interface design capable of providing access to a large data inventory without overwhelming the user. System's performance was evaluated based on statistical analysis of a user study in which

users were asked to perform a certain data retrieval task using currently available systems and *Hydroseek*.

CHAPTER 1: INTRODUCTION

1.1. Hydrologic Data in the US: State of the Art

The advent of the Internet opened the floodgates of information to hydrologists. Decades of data migrated from punch cards to digital environment. Large amounts of data have become available sometimes free of charge. In a USGS report Dr. Robert Ward from Colorado State University says “I’ve been waiting for something like this for a long time” with regard to the NWIS website which was thrilling for hydrologists when it was launched in September 2000 [1]. The USGS invests $7.1[1] million a year in maintaining the National Water Information System (NWIS) [2]. Despite the advancing technology, essentials don’t seem to have changed for hydrologists. A survey conducted by Consortium of Universities for Advancement of Hydrologic Science Inc (CUAHSI) indicates that 60.8% of hydrologists consider NWIS[2] stream flow data necessary for their research [3]. NWIS is followed by NCDC[3] precipitation data (35.1%). NCDC pan evaporation, NWIS groundwater levels, EPA STORET[4] [4] water quality, National Land Cover Dataset, National Elevation Dataset, State Soil Geographic (STATSGO) & Soil Survey Geographic (SSURGO) datasets, National Hydrography Dataset and remote sensing data (e.g. LANDSAT) are other datasets that hydrologist also utilize frequently.

Three data sources, NWIS, EPA STORET and NCDC will be examined below because;

- These data sources offer a vast variety of data.

[1] Estimate for fiscal years 2004, 2005 and 2006
[2] National Water Information System
[3] National Climatic Data Center
[4] Environmental Protection Agency Storage and Retrieval System

- All three offer fundamental meteorological data for hydrologists such as precipitation and evaporation.
- Both NWIS and EPA STORET offer water quality, surface/groundwater level and stream flow data.
- All three offer point measurements with nation-wide coverage.
- NWIS and EPA STORET data is available free of charge

1.2. Major Data Sources

1.2.1. National Water Information System

The USGS' National Water Information System (NWIS) is a distributed water database in which data can be processed over a network of workstations and fileservers at US Geological Survey (USGS) offices throughout the US. The system is composed of four subsystems:

- Ground-Water Site-Inventory (GWSI) System
- Water-Quality (WQ) System
- Automated Data-Processing System (ADAPS)
- Water-Use Data System (WUDS)

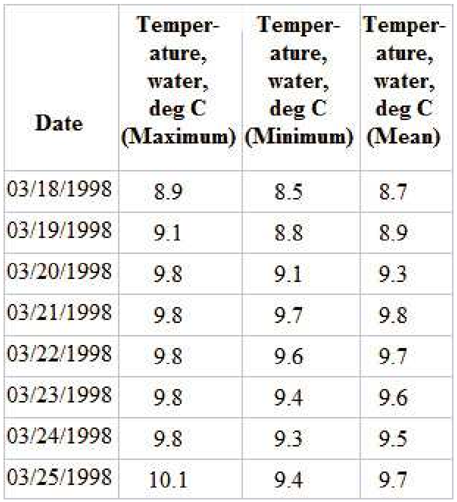
The GWSI System contains and provides access to inventory information about sites at streams, wells, springs, tunnels, drains, lakes, reservoirs, ponds and water-use facilities. The WQ System contains results of more than 3.5 million analyses of water samples that describe the chemical, physical and biological characteristics of surface and ground water. Data availability will be further examined in the following chapters. Types of chemical data include concentrations of major ions, trace elements, nutrients, pesticides and various organic compounds. Physical characteristics data include pH, specific conductance, water and air temperature, dissolved oxygen and barometric pressure. The results of laboratory analyses are verified by laboratory personnel and transmitted to the originator of the data to be stored in their water quality database. Sediment data in the WQ System include suspended-sediment concentrations in water, sediment-size distributions, and chemical concentrations of suspended sediments and bottom sediments. Biological data in the system include population densities and diversity indexes of periphyton, phytoplankton, and benthic invertebrates. ADAPS contains more than 850,000 station years of time-series data that describe stream-water levels, stream flow (discharge), reservoir water levels, surface-water and ground-water quality, ground water levels, and rainfall. WUDS stores summary data on water use throughout the US and includes two database systems: the Site-Specific Water-Use Data System (SWUDS), and the Aggregate Water-Use Data System (AWUDS) [6]. NWIS database contains measurements for a total of 9448 different parameters. Data are collected by field personnel or relayed through telephones or satellites to offices where it is stored and processed.

Data could be daily (e.g. daily mean, maximum or minimum stage height), less frequent or irregular (e.g. water quality samples) or real-time (e.g. instantaneous discharge, water temperature, pH). Real-time data are time-series recorded at fixed intervals from automated equipment and represent the most recent hydrologic conditions. Measurements are commonly recorded at 5-60 minute intervals and transmitted to the NWIS every 1-4 hours. Daily values are summarized from time-series data for each day for the period of record and may represent the daily mean, median, maximum, minimum value. In addition to automated measurements, manual field measurements are also taken periodically to supplement and/or verify the time series measurements. Table 1.1 shows the availability of aforementioned data types within the NWIS.

Table 1.1. Stations and data availability in National Water Information System

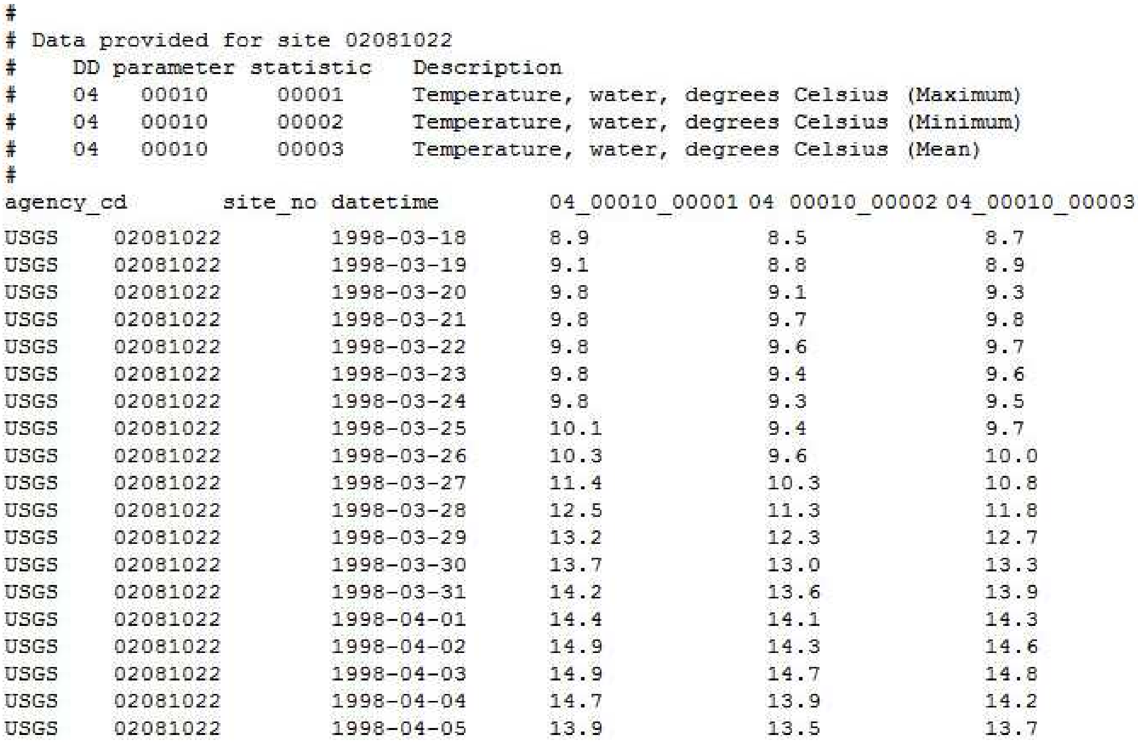
Surface Water	
Real-time data available at	8,091 sites
Daily data available at	24,461 sites
Manual field measurements available at	14,290 sites
Groundwater	
Real-time data available at	920 sites
Daily data available at	4,488 sites
Manual field measurements available at	722,529 sites
Water Quality	
Real-time data available at	1,111 sites
Daily data available at	2,685 sites
Laboratory analysis results available at	357,689 sites

Since September 2000, NWIS data holdings can be accessed through the NWIS’ website. NWISweb [7] allows querying of the database using web forms while data can be retrieved in the form of plots, HTML tables or text files. Figures 1.1 and 1.2 show example HTML and text outputs from NWIS web, respectively. NWIS data can be obtained at no cost.



Date	Temper-ature, water, deg C (Maximum)	Temper-ature, water, deg C (Minimum)	Temper-ature, water, deg C (Mean)
03/18/1998	8.9	8.5	8.7
03/19/1998	9.1	8.8	8.9
03/20/1998	9.8	9.1	9.3
03/21/1998	9.8	9.7	9.8
03/22/1998	9.8	9.6	9.7
03/23/1998	9.8	9.4	9.6
03/24/1998	9.8	9.3	9.5
03/25/1998	10.1	9.4	9.7

Figure 1.1. Example HTML output from NWISweb



```
#
# Data provided for site 02081022
#    DD parameter statistic   Description
#    04   00010     00001     Temperature, water, degrees Celsius (Maximum)
#    04   00010     00002     Temperature, water, degrees Celsius (Minimum)
#    04   00010     00003     Temperature, water, degrees Celsius (Mean)
#
agency_cd       site_no datetime        04_00010_00001 04_00010_00002 04_00010_00003
USGS    02081022        1998-03-18      8.9             8.5             8.7
USGS    02081022        1998-03-19      9.1             8.8             8.9
USGS    02081022        1998-03-20      9.8             9.1             9.3
USGS    02081022        1998-03-21      9.8             9.7             9.8
USGS    02081022        1998-03-22      9.8             9.6             9.7
USGS    02081022        1998-03-23      9.8             9.4             9.6
USGS    02081022        1998-03-24      9.8             9.3             9.5
USGS    02081022        1998-03-25      10.1            9.4             9.7
USGS    02081022        1998-03-26      10.3            9.6             10.0
USGS    02081022        1998-03-27      11.4            10.3            10.8
USGS    02081022        1998-03-28      12.5            11.3            11.8
USGS    02081022        1998-03-29      13.2            12.3            12.7
USGS    02081022        1998-03-30      13.7            13.0            13.3
USGS    02081022        1998-03-31      14.2            13.6            13.9
USGS    02081022        1998-04-01      14.4            14.1            14.3
USGS    02081022        1998-04-02      14.9            14.3            14.6
USGS    02081022        1998-04-03      14.9            14.7            14.8
USGS    02081022        1998-04-04      14.7            13.9            14.2
USGS    02081022        1998-04-05      13.9            13.5            13.7
```

Figure 1.2. Example tab-delimited text output from NWISweb

1.2.2. EPA STORET

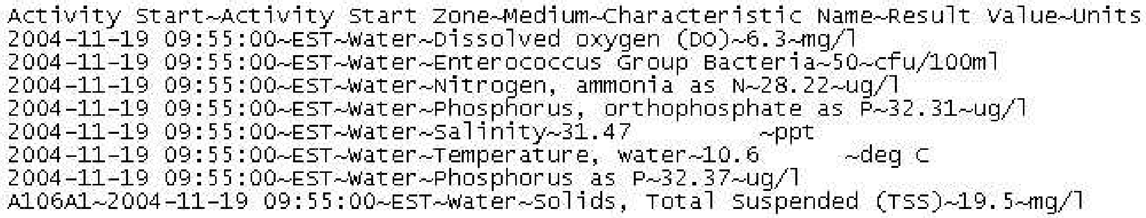
The U.S. Environmental Protection Agency (EPA) maintains two data management systems containing water quality information.

- Legacy Data Center (LDC) is a static, archived database containing historical water quality data dating back to the early part of the 20th century and collected up to the end of 1998.
- Modernized STORET is an operational system actively being populated. It contains data collected beginning in 1999, along with older data that has been properly documented and migrated from the LDC.

Both systems contain raw biological, chemical, and physical data on surface and ground water in addition to meteorological data collected by federal, state and local agencies, Indian tribes, volunteer groups, academics, and others. All 50 States, territories, and jurisdictions of the U.S. are represented in these systems. Both the LDC and STORET are web-enabled and available to the public [8]. STORET database contains measurements for a total of 9384 different parameters.

In STORET organizations are the primary owners of data and control access to it. STORET gives organizations several options for identifying their stations. Each station has a unique identifier, assigned at the discretion of the organization that owns the data. As of today EPA STORET offers data from 279 organizations with a total of 274,918 stations.

Monitoring data can be submitted to STORET from individual monitoring groups at varied rates (i.e. monthly, quarterly, etc.). However STORET Data Warehouse is refreshed with new data on the 2nd Monday of each month thus does not offer any real-time data. Monitoring organizations who wish to submit data to STORET must operate the STORET System locally. The local STORET System is a data management system with data entry and reporting software modules that operate on personal computers. STORET web page allows querying of the records using web forms while data can be retrieved in the tilde (~) delimited text files. Figure 1.3 shows example data output from STORET.



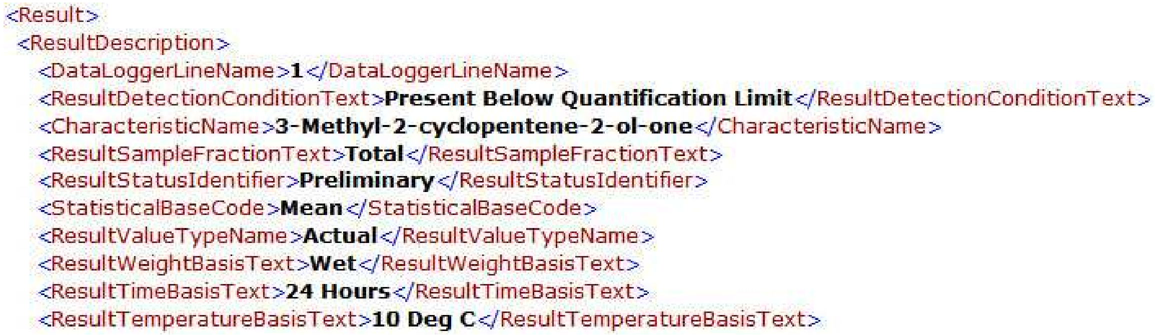
```
Activity Start~Activity Start Zone~Medium~Characteristic Name~Result Value~Units
2004-11-19 09:55:00~EST~Water~Dissolved oxygen (DO)~6.3~mg/l
2004-11-19 09:55:00~EST~Water~Enterococcus Group Bacteria~50~cfu/100ml
2004-11-19 09:55:00~EST~Water~Nitrogen, ammonia as N~28.22~ug/l
2004-11-19 09:55:00~EST~Water~Phosphorus, orthophosphate as P~32.31~ug/l
2004-11-19 09:55:00~EST~Water~Salinity~31.47         ~ppt
2004-11-19 09:55:00~EST~Water~Temperature, water~10.6      ~deg C
2004-11-19 09:55:00~EST~Water~Phosphorus as P~32.37~ug/l
A106A1~2004-11-19 09:55:00~EST~Water~Solids, Total Suspended (TSS)~19.5~mg/l
```

Figure 1.3. Example output from EPA STORET

Starting January 2007 STORET will start accepting water quality data submissions through new Water Quality Exchange (WQX) system. The new system will allow data submission to EPA using a national data exchange network [9] and EPA's Central Data Exchange (CDX) [10] thus eliminating the need to maintain a local STORET database system to be able to contribute data. It will also transition the system to the use of EPA's Environmental Sampling, Analysis, and Results (ESAR) standard [11], which gives consistent names and definitions to common data elements that are used across the Agency. Data will be submitted to the WQX system using XML in the format of the WQX schema which follows the ESAR standard closely with influences from Water Quality Data Elements[5] (WQDE) [12].

[5] Sets of data elements developed by National Water Quality Monitoring Council (NWQMC) and approved by Advisory Committee on Water Information (ACWI) as the minimum elements necessary to facilitate the exchange of chemical, microbiological, population/community (ecological and

Moreover in WQX, XML replaces the tilde delimited text file format and the system becomes accessible through web services. Modernized STORET will be supported until September 2009. Figure 1.4 shows an excerpt from an XML document that conforms to WQX schema. EPA STORET data can be accessed free of charge.



```
<Result>
 <ResultDescription>
   <DataLoggerLineName>1</DataLoggerLineName>
   <ResultDetectionConditionText>Present Below Quantification Limit</ResultDetectionConditionText>
   <CharacteristicName>3-Methyl-2-cyclopentene-2-ol-one</CharacteristicName>
   <ResultSampleFractionText>Total</ResultSampleFractionText>
   <ResultStatusIdentifier>Preliminary</ResultStatusIdentifier>
   <StatisticalBaseCode>Mean</StatisticalBaseCode>
   <ResultValueTypeName>Actual</ResultValueTypeName>
   <ResultWeightBasisText>Wet</ResultWeightBasisText>
   <ResultTimeBasisText>24 Hours</ResultTimeBasisText>
   <ResultTemperatureBasisText>10 Deg C</ResultTemperatureBasisText>
```

Figure 1.4. Excerpt from a WQX XML document

1.2.3. National Climatic Data Center

NCDC is the world's largest active archive of weather data. The US Weather Bureau, Air Force and Navy Tabulation Units in New Orleans, LA were combined and formed into the National Weather Records Center in Asheville, NC in November 1951. The Center was eventually renamed the National Climatic Data Center. The National Archives and Records Administration has designated NCDC as the Commerce Department's only Agency Records Center. NCDC archives weather data obtained by the National Weather Service, Military Services, Federal Aviation Administration, and the Coast Guard, as well as data from voluntary cooperative observers in addition to NEXRAD (Next Generation Weather Radar) and ASOS (Automated Surface Observing System). The Center has more than 150 years of

bioassessment) and (eco)toxicological assessment data. It consists of several modules that can be used independently as needed for different types of monitoring data.

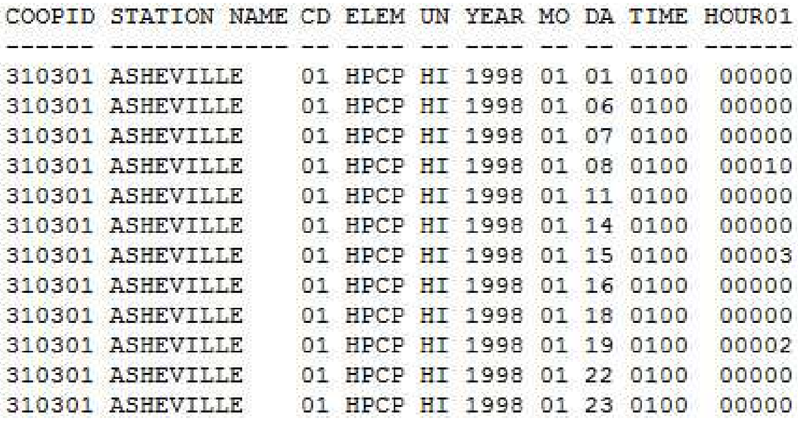
data on hand with 224 gigabytes of new information added each day. NCDC archives 99 percent of all NOAA[6] data, including over 320 million paper records; 2.5 million microfiche records; over 1.2 petabytes of digital data residing in a mass storage environment with satellite weather images back to 1960. The NCDC website receives over 100 million hits per year [13].

Routine NCDC publications can be given under following categories.

- The Local Climatological Data: LCD is produced monthly and annually for some 270 cities. The LCD contains 3-hourly, daily, and monthly values. The annual issue contains a review of the year (normals, means and extremes).
- The Climatological Data: CD is produced monthly and annually, contains daily temperature and precipitation data for over 8,000 locations. It is grouped by state or region (e.g. New England).
- The Hourly Precipitation Data: HPD is produced monthly. It contains data on nearly 3,000 hourly precipitation stations, and is published by state or region.
- The Storm Data: SD documents significant U.S. storms and contains statistics on property damage and human injuries and deaths.
- The Monthly Climatic Data for the World: MCDW provides monthly statistics for some 1,500 surface stations and approximately 800 upper air stations.

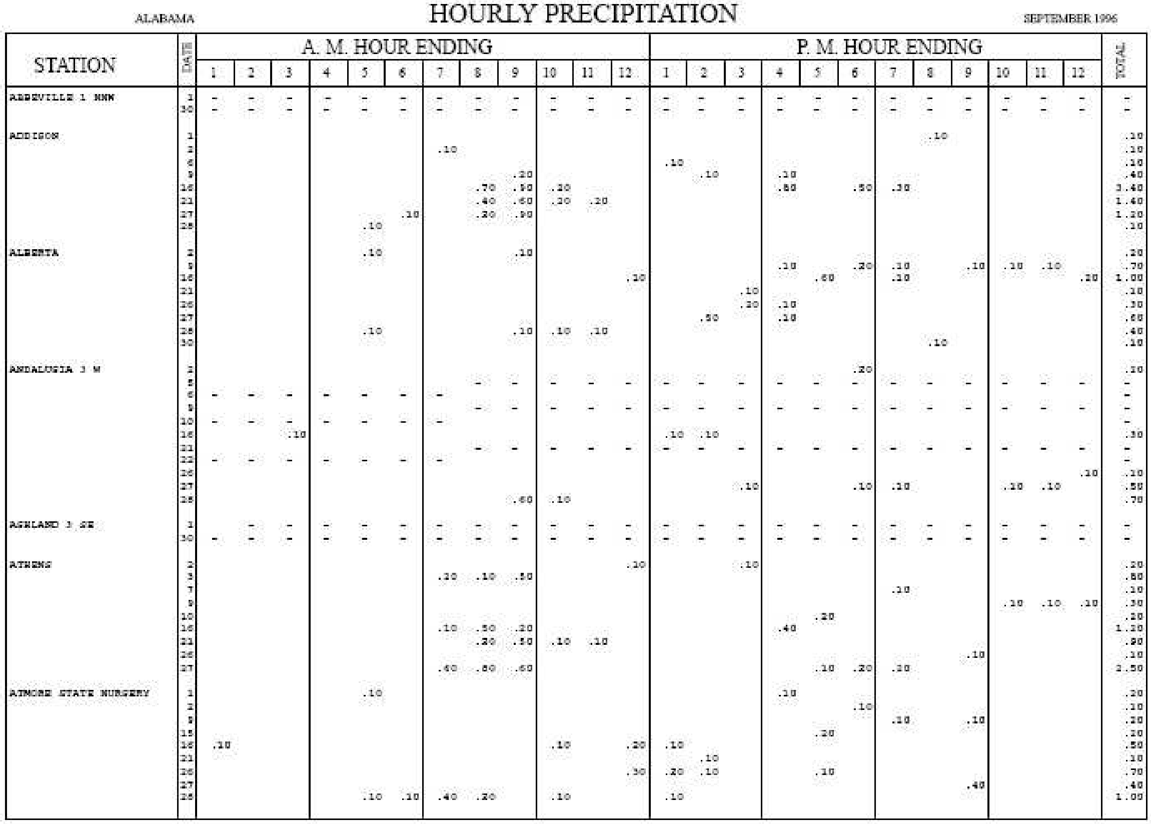
NCDC maintains over 500 digital data sets. Ozone data, Lightning Archive, Cyclone Intensity and Sea Ice Database are some examples. Time series data are offered in tab delimited text or PDF format. Figures 1.5 and 1.6 show example outputs of hourly precipitation data from NCDC system.

[6] National Oceanic and Atmospheric Administration



```
COOPID STATION NAME CD ELEM UN YEAR MO DA TIME HOUR01
------ ------------ -- ---- -- ---- -- -- ---- ------
310301 ASHEVILLE    01 HPCP HI 1998 01 01 0100  00000
310301 ASHEVILLE    01 HPCP HI 1998 01 06 0100  00000
310301 ASHEVILLE    01 HPCP HI 1998 01 07 0100  00000
310301 ASHEVILLE    01 HPCP HI 1998 01 08 0100  00010
310301 ASHEVILLE    01 HPCP HI 1998 01 11 0100  00000
310301 ASHEVILLE    01 HPCP HI 1998 01 14 0100  00000
310301 ASHEVILLE    01 HPCP HI 1998 01 15 0100  00003
310301 ASHEVILLE    01 HPCP HI 1998 01 16 0100  00000
310301 ASHEVILLE    01 HPCP HI 1998 01 18 0100  00000
310301 ASHEVILLE    01 HPCP HI 1998 01 19 0100  00002
310301 ASHEVILLE    01 HPCP HI 1998 01 22 0100  00000
310301 ASHEVILLE    01 HPCP HI 1998 01 23 0100  00000
```

Figure 1.5. Example tab-delimited text output from NCDC



ALABAMA

HOURLY PRECIPITATION

SEPTEMBER 1996

STATION	DATE	A. M. HOUR ENDING												P. M. HOUR ENDING												TOTAL
		1	2	3	4	5	6	7	8	9	10	11	12	1	2	3	4	5	6	7	8	9	10	11	12	
ABBEVILLE 1 NNW	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	30	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
ADDISON	1																				.10					.10
	2							.10																		.10
	6													.10												.10
	9									.20					.10		.10									.40
	16								.70	.90	.20						.80		.50	.30						3.40
	21								.40	.60	.20	.20														1.40
	27						.10		.20	.90																1.20
	28					.10																				.10
ALBERTA	2					.10				.10																.20
	9																.10		.20	.10		.10	.10	.10		.70
	16												.10					.60		.10					.20	1.00
	21															.10										.10
	26															.20	.10									.30
	27														.50		.10									.60
	28					.10				.10	.10	.10														.40
	30																				.10					.10
ANDALUSIA 3 W	2																		.20							.20
	5								-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	6	-	-	-	-	-	-	-																		-
	9								-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	10	-	-	-	-	-	-	-																		-
	16			.10										.10	.10											.30
	21								-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	22	-	-	-	-	-	-	-																		-
	26																								.10	.10
	27															.10			.10	.10			.10	.10		.50
	28									.60	.10															.70
ASHLAND 3 SE	1		-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	30	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
ATHENS	2												.10			.10										.20
	3							.20	.10	.50																.80
	7																			.10						.10
	9																						.10	.10	.10	.30
	10																	.20								.20
	16							.10	.50	.20							.40									1.20
	21								.20	.50	.10	.10														.90
	26																					.10				.10
	27							.60	.80	.60								.10	.20	.20						2.50
ATMORE STATE NURSERY	1					.10											.10									.20
	2																		.10							.10
	9																			.10		.10				.20
	15																	.20								.20
	16	.10									.10		.20	.10												.50
	21														.10											.10
	26												.30	.20	.10			.10								.70
	27																					.40				.40
	28					.10	.10	.40	.20		.10			.10												1.00

Figure 1.6. Example PDF output from NCDC

Different types of data such as Satellite imagery, maps, and radar data can be available in different formats such as NetCDF, ESRI Shapefile or Arc/Info ASCII Grid.

The NOAA National Operational Model Archive and Distribution System (NOMADS) allows access to NCDC data through Open Data Access Protocol (OpenDAP) [14]. NOAAServer provides another interface for querying and obtaining NCDC data of various formats [15]. Certain NCDC datasets can also be accessed using Web Services [16]. Access to NCDC data requires subscription or individual payments per data request. Academic institutions can retrieve data free of charge.

1.3. USGS Geospatial Datasets

1.3.1. National Elevation Dataset

The USGS National Elevation Dataset (NED) has been developed by merging the highest-resolution elevation data available across the United States into a seamless raster format. NED is the result of the maturation of the USGS effort to provide 1:24,000-scale Digital Elevation Model (DEM)[7] data for the conterminous US and 1:63,360-scale DEM data for Alaska. The dataset provides seamless coverage of the United States, HI, AK, and the island territories. NED has a consistent projection (Geographic), resolution (1 arc second), and elevation units (meters). The horizontal datum is NAD83, except for AK, which is NAD27. The vertical datum is NAVD88, except for AK, which is NGVD29. NED is a living dataset that is updated bimonthly to incorporate the "best available" DEM data [17]. NED can be accessed through The Seamless Data Distribution System (SDDS) map interface [18] but

[7] DEM is a digital, gridded version of a topographic map

downloading data requires payment. Available formats are ArcGrid, .BIL, GridFloat, and GeoTiff. Digital Elevation Models used in the production of NED can be obtained from Center for Earth Resources Observation and Science (EROS) [19]. 1/9 Arc Second resolution generated using Light Detection and Ranging (LIDAR) source data NED is available for limited areas.

1.3.2. National Hydrography Dataset

The National Hydrography Dataset (NHD) is a vector geospatial theme for surface water hydrography obtained from topographic maps and additional sources. It is available nationwide as medium resolution at 1:100,000-scale, and in much of the Country as high resolution of 1:24,000-scale or better. A few “local resolution” areas are also available at varying scales. The NHD is available in ESRI geodatabase format known as NHDinGEO, in ESRI coverage format known as NHDGEOinARC, and in ESRI Shapefile format known as NHDGEOinShape. The NHD is organized by hydrologic units, but can be downloaded in various extents. The NHD is based upon the content of USGS Digital Line Graph (DLG) hydrography data integrated with reach-related information from the EPA Reach File Version 3 (RF3) [20]. NHD data can be viewed using a map interface; NHD viewer [21] and can be downloaded via FTP free of charge [22]. A newer flavor of NHD called NHDPlus is also available at 1:100,000-scale complete over the conterminous U.S. and Hawaii but not Alaska, Puerto Rico, and the Virgin Islands. NHDPlus is a suite of geospatial products that build on and extend the capabilities of the NHD by integrating it with the National Elevation Dataset and the Watershed Boundary Dataset. NHDPlus includes up/downstream analysis, stream order, catchment attributes (e.g. land cover), streamflow volume and velocity estimates [23]. A new cycle of NHDPlus using the soon to be completed 1:24,000-scale NHD and new Watershed Boundary Dataset is under consideration [24].

1.3.3. Elevation Derivatives for National Applications

EDNA is a database derived from the National Elevation Dataset (NED), which has been hydrologically conditioned for improved hydrologic flow representation. The seamless EDNA database provides 30 meters resolution raster and vector data layers including aspect, contours, flow accumulation, flow direction, reach catchment seedpoints, reach catchments, shaded relief, sinks, slope and synthetic streamlines [25]. Data can be in ESRI Shapefile or ArcGrid formats. Creation of EDNA data is a 3 step process and can be summarized as modification of NED such that predictions based on digital elevation models (e.g. streamlines) are accurate. [25, 26] EDNA data can be viewed using an interactive map interface [27] but download requires password access.

1.3.4. National Land Cover Database

NLCD 92 is a 21-category land cover classification scheme that has been applied consistently over the conterminous U.S. It is based primarily on the unsupervised classification of Landsat 5 TM (Thematic Mapper) 1992 imagery. Ancillary data sources included topography, census, agricultural statistics, soil characteristics, other land cover maps, and wetlands data. The NLCD 92 classification is provided as raster data with a spatial resolution of 30 meters. Even though classified under USGS datasets in this dissertation, effort is sponsored by Multi-Resolution Land Characteristics (MRLC) Consortium; a group of 11 federal agencies including NASA, USGS, EPA, Natural Resources Conservation Service (NRCS), NOAA and USDA Forest Service. In 1999 MRLC started working on a second-generation NLCD dataset, this time using Landsat 7 imagery for the entire United States to produce of a comprehensive land cover database for the US called the National Land Cover Database (NLCD 2001) [28]. However as of November 21, 2006 NLCD 2001 is not yet available for the entire US; certain states are still under construction [29]. Data can be available in ArcGrid, .BIL and GeoTiff formats. Imagery data requires payment. Tree canopy, urban imperviousness and land cover data can be downloaded via FTP free of charge.

1.4. NRCS Geospatial Datasets

1.4.1. SSURGO, STATSGO and NATSGO

The U.S. Department of Agriculture's (USDA) Natural Resources Conservation Service (NRCS), formerly the Soil Conservation Service (SCS) offers three geographic soil data bases.

- Soil Survey Geographic (SSURGO) database
- State Soil Geographic (STATSGO) database
- National Soil Geographic (NATSGO) database

The SSURGO data base provides the most detailed level of information and was designed primarily for farm and ranch, landowner/user, township, or county natural resource planning and management. Using the soil attributes, this data base serves as an excellent source for determining erodible areas, making land use assessments, identifying potential wetlands and sand and gravel aquifer areas.

Soil maps in the SSURGO data base are made using field methods while aerial photographs are used as the field map base. Maps are made at scales ranging from 1:12,000 to 1:63,360. Typical scales are 1:15,840, 1:20,000, or 1:24,000 [30].

The STATSGO data base was designed primarily for regional, multi-state, river basin, State, and multi-county resource planning, management, and monitoring. STATSGO data are not detailed enough to make interpretations at a county level. Soil maps for STATSGO are compiled by generalizing more detailed soil survey maps (SSURGO). Where more detailed soil survey maps are not available, data on geology, topography, vegetation, and climate are assembled, together with Land Remote Sensing Satellite (LANDSAT) images. Soils of like areas are studied, and the probable classification and extent of the soils are determined. Map unit composition for a STATSGO map is determined by transecting or sampling areas on the more detailed maps and expanding the data statistically to characterize the whole map unit. STATSGO has been renamed to the U.S. General Soil Map while the abbreviation remains the same [31].

The NATSGO data base is used primarily for national and regional resource appraisal, planning, and monitoring. The boundaries of the major land resource areas (MLRA)[8] and regions were used to form the NATSGO data base. Map unit composition for NATSGO was determined by sampling done as part of the 1982 National Resources Inventory. The NATSGO map was compiled on an NRCS-

[8] Geographically associated areas of land that are characterized by particular patterns of soil, climate, landforms and land cover. There are 278 major land resource areas in the US. For example, MLRA 1 (Northern Pacific Coast Range, Foothills, and Valleys), MLRA 157 (Arid and Semiarid Low Mountain Slopes), MLRA 270 (Humid Mountains and Valleys), MLRA 190 (Stratovolcanoes of the Mariana Islands).

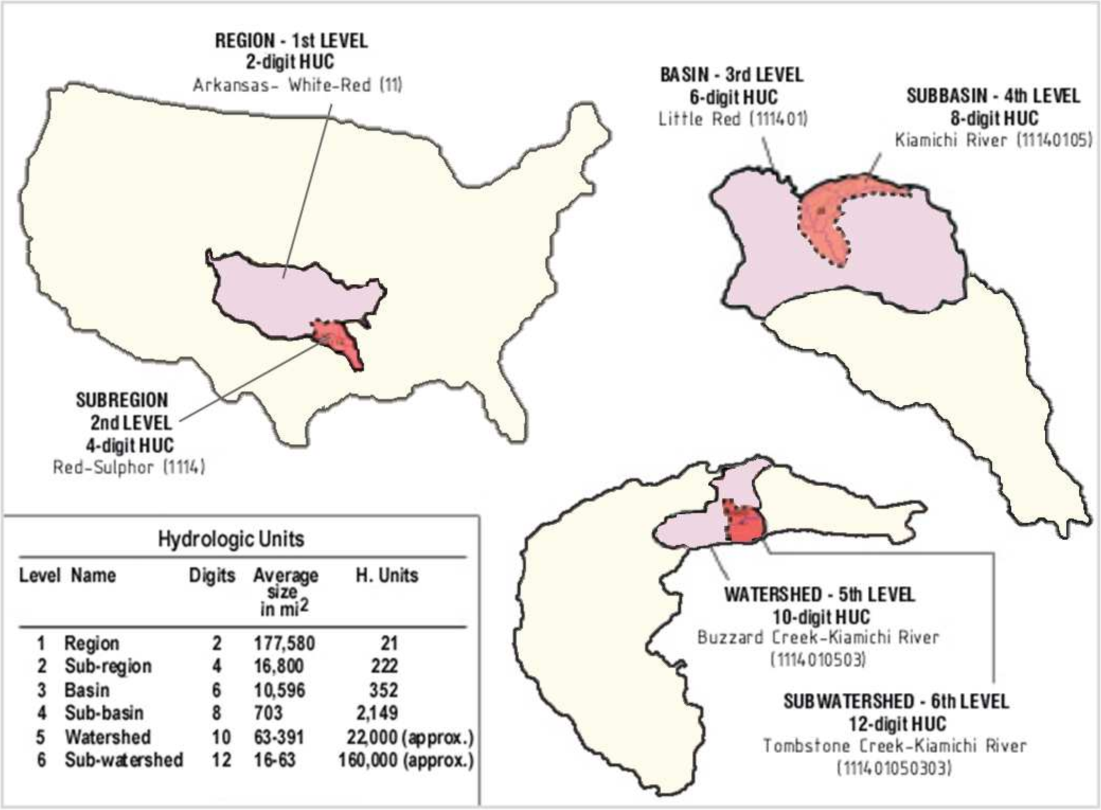
adapted version of the 1970 Bureau of Census Automated State and County Map Database and it was digitized from the USGS 1:5,000,000 scale U.S. base map [31]. Soil data can be downloaded free of charge and available as ESRI Shapefile and pipe (|) delimited text files.

1.4.2. Watershed Boundary Dataset

Watershed boundaries define the aerial extent of surface water drainage to a point. Watershed Boundary Dataset defines hydrologic units to establish a base-line drainage boundary framework, accounting for all land and surface areas. A hydrologic unit has a single flow outlet except in coastal or lakefront areas. "A hydrologic unit is a drainage area delineated to nest in a multi-level, hierarchical drainage system. Its boundaries are defined by hydrographic and topographic criteria that delineate an area of land upstream from a specific point on a river, stream or similar surface waters. A hydrologic unit can accept surface water directly from upstream drainage areas, and indirectly from associated surface areas such as remnant, non-contributing, and diversions to form a drainage area with single or multiple outlet points. Hydrologic units are only synonymous with classic watersheds when their boundaries include all the source area contributing surface water to a single defined outlet point." [32].

Hydrologic units through four levels were created in the 1970's and have been used extensively throughout the United States. During that time the U.S. Geological Survey (USGS) developed a hierarchical hydrologic unit code (HUC) for the United States. This system divides the country into 21 Regions, 222 Subregions, 352 Accounting Units, and 2,149 Cataloging Units based on surface hydrologic features.

The smallest USGS unit (8-digit hydrologic unit) is approximately 448,000 acres. Over the last ten years, many federal and state agencies have realized current 8-digit hydrologic unit maps are unsatisfactory for many purposes because of inadequate bases or scales. In 2002 Advisory Committee on Water Information (ACWI), the Federal Geographic Data Committee (FGDC) and NRCS wrote new guidelines for hydrologic units. According to new guidelines 3rd level is officially called "basin" (formerly "cataloging unit") and the 4th level is called "sub-basin" (formerly "accounting unit"), a 5th level (10-digit code) designates a watershed and 6th level (12-digit code) a sub-watershed. [32] Figure 1.7 shows hydrologic unit code system which is currently in effect [34].



Hydrologic Units

Level	Name	Digits	Average size in mi^2	H. Units
1	Region	2	177,580	21
2	Sub-region	4	16,800	222
3	Basin	6	10,596	352
4	Sub-basin	8	703	2,149
5	Watershed	10	63-391	22,000 (approx.)
6	Sub-watershed	12	16-63	160,000 (approx.)

Figure 1.7. Hydrologic Unit Code System

The Watershed Boundary Dataset is being developed under the leadership of the Subcommittee on Spatial Water Data, which is part of the Advisory Committee on Water Information (ACWI) and the Federal Geographic Data Committee (FGDC). The USDA Natural Resources Conservation Service (NRCS), along with many other federal agencies and national associations. Data can be downloaded at 1:24,000 scale as ESRI ShapeFile from SDDS map interface [18] or via FTP [33].

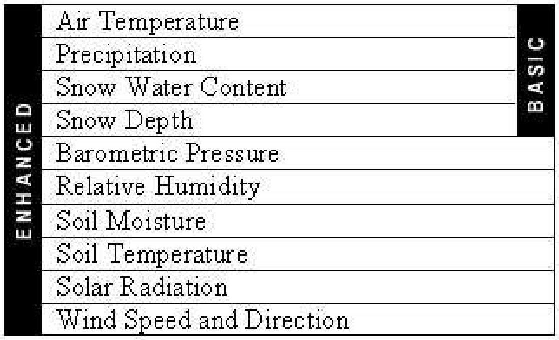
1.4.3. Parameter-elevation Regressions on Independent Slopes Model

PRISM (Parameter-elevation Regressions on Independent Slopes Model) is a hybrid statistical-geographic approach to climate mapping developed at Oregon State University. PRISM uses point measurements of climate data and a DEM) to generate estimates of annual, monthly and event-based climatic elements. These estimates are derived for a horizontal grid. PRISM is not a static system of equations; rather, it is a coordinated set of rules, decisions and calculations designed to mimic the decision-making process an expert climatologist would invoke when creating a climate map. PRISM was originally developed in 1991 for precipitation estimation but more recently has been generalized and successfully applied to other climate elements and derived variables, including temperature, snowfall, degree-days (heat units) and frost dates [35]. Data can be downloaded free of charge in Arc/Info ASCII Grid or Arc/Info Coverage Export (.e00) formats [36].

1.4.4. Snowpack Telemetry

History of SNOTEL (SNOwpack TELemetry) system dates back to mid 1930's when Congress mandated that NRCS (Soil Conservation Service at the time) shall measure snowpack in the mountains in the West and forecast the water supply. Today SNOTEL is an automated system that collects snowpack and related climatic data at over 660 remote sites in the Western United States and Alaska. A basic SNOTEL site provides snowpack water content, snow depth, precipitation accumulation and air temperature with daily statistics (max, min, average). Enhanced SNOTEL sites can provide weather station functions in addition to soil moisture/temperature measurements at various depths.

Table 1.2. Available parameters at SNOTEL sites



ENHANCED	Air Temperature	BASIC
	Precipitation	
	Snow Water Content	
	Snow Depth	
	Barometric Pressure	
	Relative Humidity	
	Soil Moisture	
	Soil Temperature	
	Solar Radiation	
	Wind Speed and Direction	

Table 1.2 shows a list of parameters measured at basic and enhanced SNOTEL sites. Data is available free of charge in tab-delimited text format from SNOTEL website at NRCS [37].

1.5. NOAA Geospatial Datasets

1.5.1. Coastal Change Analysis Program

The NOAA Coastal Change Analysis Program (C-CAP) is a nationally standardized database of land cover and change information, developed using remotely sensed imagery, for the coastal regions of the US. It covers coastal intertidal areas, wetlands, and adjacent uplands with the goal of monitoring changes in these habitats, on a one-to-five year cycle. An immediate objective of C-CAP is to complete a national baseline of coastal land cover and change data. Once this baseline is complete, additional imagery will be used to track coastal changes through time. This trend information is expected to provide important feedback to managers on the success or failure of management policies and programs and aid in developing a scientific understanding of the Earth system and its response to natural and human-induced changes. This understanding will eventually allow for the prediction of impacts due to these changes and the assessment of their cumulative effects, helping coastal resource managers make more informed regional decisions.

The C-CAP mapping boundary was determined using the inland extent of estuarine drainage basins, coastal counties, and Coastal Zone Management Act boundaries. The boundary was then adjusted to reflect natural breaks based on features in the landscape [38]. Figure 1.8 shows the current status of C-CAP map. Data can be downloaded free of charge using C-CAP interactive map interface [39] (user receives e-mail when requested data is ready) or directly for predefined regions in ERDAS Imagine format (.img).

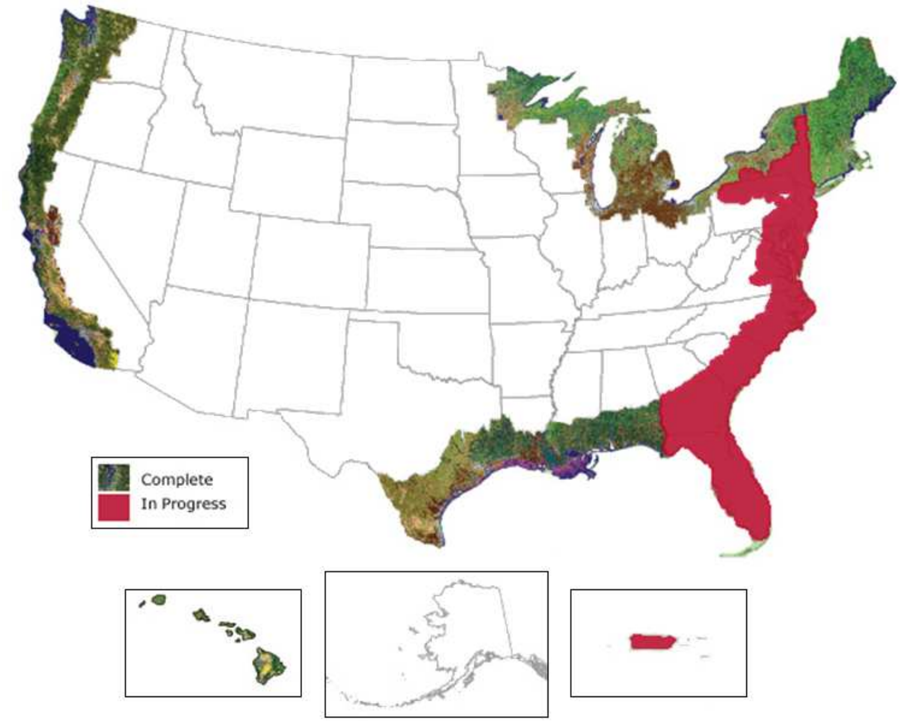


Figure 1.8. Current Status of C-CAP

1.5.2. Remote Sensing Data (AVHRR, GOES)

Advanced Very High Resolution Radiometer (AVHRR) and Geostationary Operational Environmental Satellite (GOES) data can be accessed through The Comprehensive Large Array-data Stewardship System (CLASS) electronic library of NOAA [40]. AVHRR is mounted on satellites in Polar Orbiting Satellite (POES) system. AVHRR data is commonly used in flood monitoring, estimation of snow cover, surface temperature and vegetation indices[9] and has fairly continuous global coverage since June 1979. It provides global coverage 4 times daily.

GOES data is commonly used for estimating snow depth, snow cover, cloud cover and precipitation. Unlike POES, a geostationary satellite maintains a constant

[9] VI represent vegetative characteristics such as plant leaf area, total biomass and vigor

position relative to the surface of the earth. GOES-10 (West) and GOES-12 (East) cover most of the western hemisphere and routinely transmit data every 15 minutes of the continental United States. During severe weather events, such as hurricanes, data can be transmitted every 5 minutes for constant monitoring. GOES generate denser data streams of thermal and visible imagery than AVHRR due to its higher temporal resolution, but has a coarser spatial resolution [41]. Data can be downloaded via FTP freely however system receives the requests first and notifies the user via e-mail when processing is complete. Relevant products of NOAA satellite data can be found at different agencies (e.g. National Snow and Ice Data Center, USGS) as well.

1.6. Other Relevant Datasets

1.6.1. NASA Remote Sensing Data (MODIS, LANDSAT, ASTER)

Landsat 7; launched on April 15, 1999 is the latest NASA satellite in a series that has produced an uninterrupted multi-spectral record of the Earth's land surface since 1972. Landsat 7 ETM+ (Enhanced Thematic Mapper Plus) provides high-resolution (15 to 60 meter) data and has a 16-day repeat cycle [42]. Landsat data is commonly used for determining water boundaries/surface areas, snow/ice coverage, mapping floods and estimating snowmelt runoff. National Land Cover Database examined in this chapter is also a product of Landsat data. Landsat 7 data products are available from the USGS Center for Earth Resources Observation and Science (EROS). These products are not free of charge but academic institutions receive considerable discounts.

The MODIS (Moderate Resolution Imaging Spectroradiometer) instrument operates on the Terra and Aqua spacecrafts and provides 250 to 1000 meter multi-spectral data. It views the entire surface of the Earth every one to two days. Some MODIS data products are cloud cover, total precipitable water, land surface temperature, land cover, vegetation indices, leaf area index, evapotranspiration, net photosynthesis, sea surface temperature, photosynthetically active radiation and chlorophyll a concentration [43]. Depending on the product needed, data can be ordered through Earth Observing System (EOS) Data Gateway, NASA Warehouse Inventory Search Tool (WIST) or USGS Land Processes Distributed Active Archive Center (LP DAAC) or Goddard Earth Sciences (GES) Data and Information Services Center's Mirador interfaces and downloaded via FTP free of charge. However users need to submit the order and wait for downloading instructions.

ASTER (Advanced Spaceborne Thermal Emission and Reflection Radiometer) instrument operates on Terra satellite and offers high-resolution (15 to 90 meter) multi-spectral data. ASTER does not acquire data continuously, and its sensors are activated only to collect specific scenes upon request. If the desired ASTER observations have not yet been acquired, a new data acquisition request can be submitted [44]. A calendar of scheduled ASTER data acquisitions can be viewed online [45]. Any data that ASTER has already acquired are available for searching and ordering from the EOS Data Gateway, USGS EROS, Japan Ground Data System (GDS). Calibrated and derived products however are created on-demand for each user [44]. Some ASTER products can be downloaded freely from LP DAAC while most requires payment [46]. ASTER data is commonly used for estimating land surface temperature, elevation, evapotranspiration, mapping vegetation, bedrock and soil distribution and monitoring deforestation, flooding and other hazards. Boken and

Easson (2005) used ASTER data to model groundwater depth in Mississippi Delta [47].

All data mentioned in section 1.6.1 are in the HDF-EOS format.

1.6.2. North American Regional Reanalysis

National Centers for Environmental Prediction (NCEP) North American Regional Reanalysis (NARR) dataset is a long-term, high-resolution, high-frequency, atmospheric and land surface hydrology dataset for the North American domain. Following the 25 year retrospective production period (1979-2003), today NARR is being continued near real-time as the Regional Climate Data Assimilation System (R-CDAS). Data is produced at 32 km (spatial), 3-hourly (temporal) resolution [48]. Each time step creates approximately 60 MB data. But coarser resolution data is also available e.g. 64 km or 96 km grid and daily or monthly analysis. Some variables that NARR offers are temperature, snow depth, soil temperature, soil moisture, precipitation rate, total precipitation, evaporation, surface runoff, vegetation cover, canopy conductance, upward/downward solar radiation flux, relative humidity, dew point temperature, wind speed and surface drag coefficient.

Data is available in GRIB[10] format and can be downloaded from NOMADS and National Center for Atmospheric Research (NCAR). Users are provided with an FTP script to download the data for selected parameters and spatio-temporal frame [49].

[10] GRIB (GRid In Binary) is the World Meteorological Organization (WMO) standard for gridded meteorological data

1.6.3. DAYMET

DAYMET is a meteorological model that generates daily surface temperature, precipitation, humidity, and radiation using a digital elevation model and daily observations of minimum and maximum temperatures and precipitation from ground-based meteorological stations. Developed at the University of Montana, Numerical Terradynamic Simulation Group (NTSG) to fulfill the need for fine resolution, daily meteorological and climate data necessary for plant growth models, the system offers data from 1980 until 1997 at 1 km resolution [50].

Data can be obtained free of charge from EOS Training Center Natural Resource Project in 32 bit, floating point files (binary raster) for the conterminous USA or as time series in tabular format (HTML or text) using web forms for a given point identified with a latitude-longitude [51].

1.6.4. Ameriflux

The AmeriFlux network was established in 1996 in response to a workshop entitled 'Strategies for Long Term Studies of CO2 and Water Vapor Fluxes over Terrestrial Ecoystems'. It is currently composed of 103 sites from North America, Central America, and South America [52]. AmeriFlux is a federated system with sites operated by different agencies who wished to contribute data and thus became a part of the network. Individual sites may monitor different sets of parameters. As an overview, available data include CO2 flux, sensible/latent heat flux, air temperature, relative humidity, net, shortwave, longwave, and photosnythetically active radiation (incoming/outgoing), canopy wetness, throughfall, stem flow, leaf area density, soil/leaf nutrient content, canopy height and rooting depth. Data can be downloaded free of charge from Oak Ridge National Laboratory Carbon Dioxide Information Analysis Center (CDIAC) via FTP as tab-delimited text files [53].

1.7. Summary

This section examined the Nation's major hydrologic data sources from different aspects including data formats, coverage, retrieval methods and data variety. If they are classified into two broad categories such as GIS/remote sensing datasets and point data, it can be seen that the latter is not as centralized as the former. Datasets in the former group can be obtained from agencies such as NASA, NRCS and USGS or through geospatial one-stops. Individual datasets in this group have rather limited scopes when compared to most point data sources such as the USGS NWIS which measures about 10 thousand different parameters. In the former group datasets have nationwide coverage and not many alternatives exist for any given data type. For example when radar precipitation is considered, NEXRAD is the source that comes to mind; when it is MODIS, Landsat, ASTER satellite imagery, NASA comes to mind. This is of course not surprising considering the cost of operating a stream gage versus a satellite. These sources offer a vast variety of data and many of them have their own ways of dealing with data which causes a heterogeneity problem. Last but not least, hydrologists list USGS NWIS, EPA STORET and NCDC (precipitation/evaporation) point data as the three data sources of highest interest [3]. Unfortunately data from NCDC weather stations are not freely available. As a result it is understandable that for information systems in hydrology domain point data is of highest priority, especially data from NWIS and EPA.

CHAPTER 2: BACKGROUND ON RELEVANT CONCEPTS

This chapter presents an overview of information technologies employed in this work such as Asynchronous JavaScript and XML (AJAX), ontologies and web services.

2.1. Extensible Markup Language (XML)

In 1969 IBM researchers; Charles Goldfarb, Edward Mosher, and Raymond Loriewas developed the Generalized Markup Language (GML), as a means of facilitating text management in large information systems, also coining the term; "markup language". Markup languages can be defined as "a formal way of annotating a document or collection of digital data in order to indicate the structure of the document or data file and the contents of its data elements." [54]. These annotations also serve to provide a computer with information about how to process and display the document. Later Goldfarb with Charles Card and Norman Scharpf created Standard Generalized Markup Language (SGML) under American National Standards Institute's (ANSI) committee on Information Processing which was published as a working draft in 1981. It was eventually adopted by the International Standards Organization (ISO) under the name ISO 8879 in 1986 [55]. SGML was designed to enable the sharing of machine-readable documents in large projects in government, legal and the aerospace industry and was adopted by thousands of organizations whose complex systems and products required massive amounts of documentation ranging from nuclear plants, oil rigs and military systems to government laws and regulations. Major adopters of the standard included the US Internal Revenue Service and Department of Defense. It has also been used extensively in the printing and publishing industries, since it allowed multiple renditions of a single document in different styles to be generated automatically. However, its complexity has prevented its widespread application for small scale, general purpose use. SGML also introduced document grammars or schemas called "Document Type Definitions" (DTD).

A few years after SGML became an international standard Tim Berners-Lee of the European Particle Physics Laboratory (CERN) invented Hyper Text Markup Language (HTML). The main structure of modern HTML was agreed at the first WWW Conference held in May, 1994. Today HTML is the primary language for publishing Web pages. HTML is defined by a DTD and is a simplified version of SGML with a fixed set of tags used primarily to define how documents will be rendered in Web browsers [56].

While HTML allowed the content to be formatted in different ways e.g. <i>Schuykill</i> for italic font, it didn't offer any means to describe the content such as indicating that the text between the tags is a river e.g. <river>Schuykill</river>. This led to another implementation of SGML, namely eXtensible Markup Language (XML). World Wide Web Consortium (W3C) defines XML as "an extremely simple dialect of SGML" [57]. As a W3C recommendation since 1998; XML has become the de-facto standard for data exchange over the internet. Unlike HTML, XML does not specify a set of tags. XML remains a meta-language like SGML, allowing users to create any tags needed (hence "extensible") and the structural relationships between them. The simplicity of XML has encouraged active development work around XML, including software development and related languages. It has been utilized to publish markup languages such as the Chemical Markup Language (CML), Mathematical Markup Language (MathML), Earth Science Markup Language (ESML), Geography

Markup Language (GML), Ecological Metadata Language (EML), Bioinformatic Sequence Markup Language (BSML) and BIOpolymer Markup Language (BioML) [58]. XML also introduced a simplified version of SGML DTDs.

In May 2001, W3C published XML Schema, an alternative to DTDs, as a recommendation for documenting the structure of XML documents [59]. XML Schemas (XSD) can be used to validate XML documents based on their structure (e.g. ordering, number of occurrences) and content (e.g. data types; String, integer, dateTime). Another XML schema language that's gaining acceptance is RELAX NG (REgular LAnguage for XML Next Generation); the product of the synthesis of earlier RELAX and TREX (Tree Regular Expressions for XML) schema languages [60].

2.2. Resource Description Framework (RDF)

In March 2002 W3C published Resource Description Framework (RDF); "a language for representing information about resources in the World Wide Web" [61]. Two years later it became a recommendation. A "resource" is an object that can be uniquely identified by a Uniform Resource Identifier (URI) [62, 63]. For example http://www.drexel.edu/index.html#library , ftp://ftp.drexel.edu/software/SSH.zip and telnet://dunx1.drexel.edu/ are all valid URIs. RDF provides a data model for describing resources and contains properties and statements. A property can be a characteristic, attribute or relation that describes a web resource. A statement consists of three parts: a subject, a predicate and an object. The subject represents a resource which has properties, while a predicate represents the property and, an object the value of that property. In RDF, values may be atomic in nature (text strings, numbers, etc.) or other resources, which in turn may have their own properties. RDF specification [61] represents the relationships among resources, properties and values using a directed labeled graph. In RDF graphs, properties are defined as directed arcs, nodes that are resources are shown as ellipses, while nodes that are literals are shown as boxes.

Using this representation, a statement such as "Author of this document is Bora Beran" can be graphically expressed as in Figure 2.1.

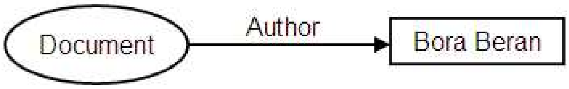


Figure 2.1. Simple RDF graph with one subject, predicate, object triple

However if more information about the author is desired this graph could look like Figure 2.2. It should be noted that in RDF resources are provided with their URIs e.g. http://www.myrdfexample.com/Author or http://www.someuri.org/Document. In Figures 2.1 and 2.2 URIs are not shown for the sake of brevity.

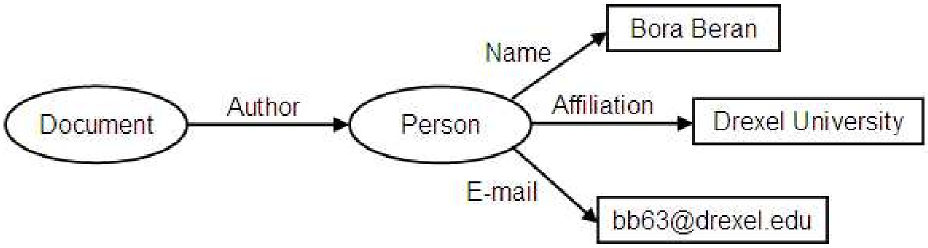


Figure 2.2. A more complicated RDF graph depicting multiple statements

RDF graphs can be serialized into XML or Notation 3 (N3). RDF/XML representation of Figure 2.2 can be seen in Figure 2.3.

```
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
        xmlns:ex="http://www.myrdfexample.com/">
  <rdf:Description rdf:about="http://www.someuri.org/Document">
    <ex:Author>
      <rdf:Description rdf:about="http://www.someuri.org/Person">
        <ex:Name>Bora Beran</ex:Name>
        <ex:Affiliation>Drexel University</ex:Affiliation>
        <ex:Email>bb63@drexel.edu</ex:Email>
      </rdf:Description>
    </ex:Author>
  </rdf:Description>
</rdf:RDF>
```

Figure 2.3. RDF graph presented in RDF/XML

Unlike XML serialization N3 is not verbose. Statements are listed as subject, predicate, object triplets. Figure 2.4 shows how the graph in Figure 2.2 would be written using this notation.

```
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix ex: <http://www.myrdfexample.com/> .

<http://www.someuri.org/Document> ex:Author <http://www.someuri.org/Person> .
<http://www.someuri.org/Person> ex:Name "Bora Beran" .
<http://www.someuri.org/Person> ex:Affiliation "Drexel University" .
<http://www.someuri.org/Person> ex:Email "bb63@drexel.edu" .
```

Figure 2.4. RDF graph presented in Notation 3

Derivatives and subsets of Notation 3 such as Turtle (Terse RDF Triple Language), TriG and NTriples can also be used for the same purpose. Similarly, formats such as TriX (Triples in XML) and RXR (Regular XML RDF) can be named as some other XML serialization approaches.

Going back to Figure 2.2, one could further elaborate on object of the “Affiliation” property and instead of using it as a literal “Drexel University” could be treated as a resource having properties such as address, president, departments etc. which could object as literals or other resources. For example an address could be a literal but also a resource having street address, city, state and zip code as its properties. But which is the right way of doing it? There is no single right answer to this question. The answer depends on the domain requirements; i.e. how much information does one want to capture.

As shown in the example above, RDF can provide a model to represent information; however, it provides no mechanisms for describing these properties, nor does it provide any mechanisms for describing the relationships between these properties and other resources [64]. Questions such as “How does the computer know that a Document has a set of properties such as Author?” or “How does the computer know that Author must be a Person?” can not be answered by RDF alone.

These issues can be overcome to some extent by RDF's vocabulary description language, RDF Schema (RDFS) which became a W3C recommendation in February 2004. RDFS provides mechanisms for describing groups of related resources (i.e. classes) and the relationships between these resources (i.e. properties) [64]. This class-property system at first may resemble object-oriented programming languages. However, RDFS describes properties in terms of the classes of resource to which they apply using the domain and range mechanisms unlike object oriented programming languages which define a class in terms of the properties its instances may have [64]. The two questions posed above can be answered using RDFS by defining the “Author” property to have a domain of “Document” and a range of “Person”. However if we think of the “Affiliation” relationship and create “Student” as a subclass of “Person” and then try to specify that a student must be affiliated with a school, we find that this is not possible using RDFS since it does not permit a restriction of this type on the property. Further expressiveness requires a language with richer semantics than RDFS. In the following section we’ll take a look at knowledge representation techniques to understand what is behind today’s web-based languages such as RDFS and OWL (Web Ontology Language).

2.3. Knowledge Representation and Ontologies

The term 'ontology' originates from the Greek words 'onto' meaning 'existence' or 'being' and logos meaning 'science'. In philosophy ontology represents a systematic account of existence. Two main streams of ontologists are: adequatists who seek taxonomy of things in reality through a descriptive means and, reductionists who describe reality in terms of simple entities and see more complex concepts as a combination of them. While earlier philosophers thought that the only source of ontology should be the study of natural sciences, some 20th Century philosophers argued that ontologies should not necessarily deal with the truth in reality or subjects of physical existence, but rather theories about the world that may or may not be true. Carnap (1950) defends the use of abstract entities as names in ‘linguistic frameworks’ pointing out that the use of abstract linguistic forms can be justified by their efficiency (the ratio of the results achieved to the amount and complexity of the efforts required) as instruments [65].

After migration of ontologies from philosophy to the realm of artificial intelligence, Gruber (1995) defined them as “explicit specifications of conceptualizations“ [66]. Here conceptualization can be described as a simplified, abstract view of the world while specification means a systematic account of the world to be represented. Ontologies have become quite popular in science, engineering and medicine and used for data integration projects in various domains to be examined in the following chapters.

As seen in Gruber’s definition, present day ontologies are logical representations of relationships between concepts; or in a sense, theories about a domain that can be used for answering questions of relevance which is very similar to Carnap’s point of view. It should also be noted that to a machine since there’s no understanding of physical reality/existence, all entities are abstract which strengthens the similarity between the two arguments.

Even though ontologies confined to particular domains were discussed so far, ontologies are also used to describe common-sense knowledge or general concepts that are the same across all domains. Cyc [67], SUMO[11] [68], Wordnet [69], DOLCE[12] [70] and BFO[13] [71] can be given as examples of these ontologies.

There are many formal representations of knowledge. Traditional knowledge representation languages can be roughly classified into logical languages, frame-based languages and graph-based languages [72]. Logical languages express knowledge as logical statements. For example using first-order logic (FOL) a.k.a. first

[11] Suggested Upper Merged Ontology

[12] Descriptive Ontology for Linguistic and Cognitive Engineering

[13] Basic Formal Ontology

order predicate calculus one can state “for every number there exists a greater number” by writing

$$\forall\ x\ \exists\ y : y > x$$

A slightly more complex example could be;

$$\forall\ x\ (\ (\ \exists\ y : 12 * y{=}x\ ) \Rightarrow (\ \exists\ y : 4 * y{=}x\ )\ )$$

representing the statement “if a number is divisible by 12, then it is also divisible by 4”. Of course application of the logic is not limited to numbers. For example,

$$\forall x\ (\mathrm{human}(x) \Rightarrow ((\mathrm{male}(x) \vee \mathrm{female}(x)) \wedge \neg\ (\mathrm{male}(x) \wedge \mathrm{female}(x))))$$

is the representation of the statement “All humans are either female or male but not both.” One of the best-known examples of logical languages is the Knowledge Interchange Format (KIF); an extended version of first-order predicate calculus and an intermediary language for translating different knowledge representation languages developed by DARPA[14] [73]. The statement "All departments have some students that fail some class" can be written in KIF as in Figure 2.5.

```
(forall ?d
     (-> (department ?d)
        (exists (?s ?c)
             (and (student ?s) (course c?)
             (offer ?d ?c) (enroll ?s ?c)
             (fail ?s ?c)
             )
         )
     )
)
```

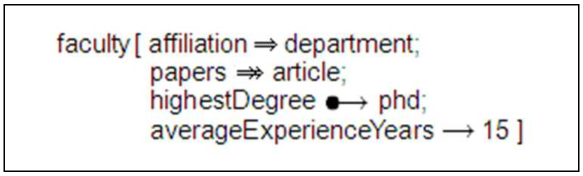
Figure 2.5. An example KIF statement

[14] The Defense Advanced Research Projects Agency

Upper ontologies; DOLCE and SUMO are available in Knowledge Interchange Format while SUMO uses a simplified version of KIF 3.0 named SUO-KIF. KIF is on its way to become an ISO standard under the name Common Logic and as of July 2007 it is a Final Draft International Standard (ISO 24707). Common Logic also has a less expressive sublanguage called Simple Common Logic [74].

First-order logic is probably the most common foundation for a logic-based knowledge representation, but many other logics have also been applied [75]. Modal logics [76], fuzzy logic [77] and probabilistic logics [78] can be named as some examples. A detailed look at logical languages is beyond the scope of this dissertation.

Frame-based systems are in many ways similar to object-oriented programming languages [75]. A frame is a data structure introduced by Marvin Minsky in the early 1970s that can be used for knowledge representation [79, 80]. It represents either an individual or a category, and associates with its subject a collection of pairs of a slot with a corresponding filler. A slot is a feature of an individual and filler is the value of that feature. For example, the ‘color’ slot for ‘sky’ would have the filler ‘blue.’ The filler of a slot for a category contains information of some kind about the fillers of the same slot for individual instances of the category. A filler is not necessarily a value (e.g. blue as in the example above). It can be a range of values, a constraint on values, or a function that computes the value. The best known example of frame-languages is F-logic. F-logic is essentially an integration of frame-based languages and first-order predicate calculus. It includes objects, inheritance, polymorphic types, query methods and encapsulation. It provides classes, attributes with domain and range definitions, is–a hierarchies with set inclusion of subclasses, and logical axioms between elements of an ontology and its instances [81]. RDF Schema is very similar to frame-based languages.



```
faculty [ affiliation ⇒ department;
          papers ↠ article;
          highestDegree •→ phd;
          averageExperienceYears → 15 ]
```

Figure 2.6. An example F-logic class declaration

Figure 2.6 shows description of the class “faculty” using F-logic. Here it is stated that faculty is affiliated with a department (an object from department class), can have multiple articles (hence the double arrow), highest degree is PhD (arrow shape indicates that this value will be inherited) and average experience for faculty is 15 years. Faculty could have subclasses such as assistant professors, associate professors etc. While they’re all PhD’s, each group would have a different average experience in years. That’s why in the class description average experience is set for the class itself but highest degree is inherited by subclasses of faculty. Subclasses and instances (in this example George) can be represented as shown in Figure 2.7.

```
assistantProfessor::faculty
associateProfessor::faculty
fullProfessor::faculty
george:assistantProfessor
```

Figure 2.7. F-logic representation of is-a hierarchy

Frames are closely related to an earlier structure-based knowledge representation technique, called semantic networks [82] which, in turn, are based on the idea of human associative memory [83]. Some claim, "most of 'frames' is just a new syntax for first-order logic" [84]. However this statement doesn't diminish the value of frame systems as easy-to-understand tools for knowledge representation. Some systems even use these similarities to their advantage. For example Ontolingua provides a frame-based syntax but then translates all information into KIF, which is a first order logic encoding of the information [85].

A semantic network is a graphic notation for representing knowledge in patterns of interconnected nodes and arcs [86]. The term semantic network covers a variety of graph based languages from its first example in 3rd century AD (Tree of Porphyry) [86] to Charles Sanders Peirce's Existential Graphs [87], Kamp’s discourse representation structures [88], Sowa's Conceptual graphs [89] and neural networks. Semantic networks have been widely used in the field of artificial intelligence in different forms as ideas were expanded, transformed and applied in different ways over the years. For example, reflections of Tesnière's work on dependency theory in the area of linguistics in late 1950s [90] can be seen in Schank’s work on case-based reasoning as he and his colleagues adopted the idea and developed notations for representing larger structures called scripts [91], thematic organization packets (TOPs) and memory organization packets (MOPs) [92]. After this short introduction to semantic networks, it should be noted that this dissertation is related to only a certain type of semantic networks, namely definitional networks.

Definitional networks emphasize is-a relations between concepts. The resulting network, also called a generalization or subsumption hierarchy, supports inheritance of properties defined for a supertype to its subtypes. The information in these networks is assumed to be necessarily true.

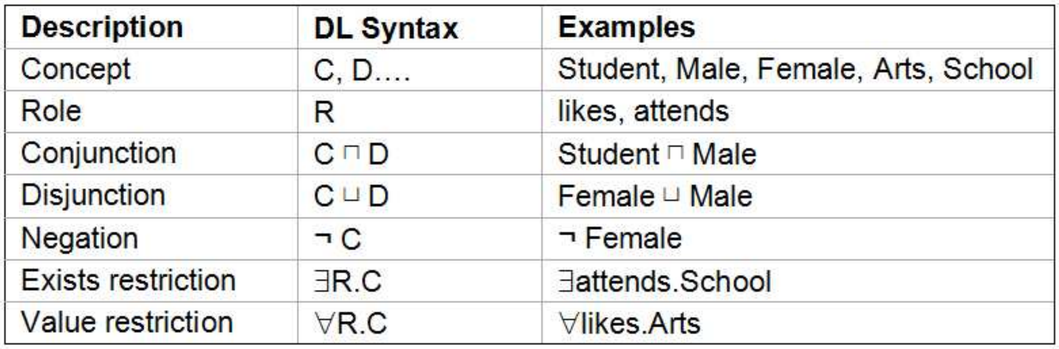
Work on definitional networks led to KL-ONE [93], the predecessor of today's description logics (DL). This work began with an emphasis on making term definitions in semantic networks more precise by combining the power of semantic networks with formal logic [94]. KL-ONE and many versions of description logics use a subset of classical first-order logic which was discussed earlier. This is in order to avoid problems of undecidability and intractability associated with using full first-order logic [95]. A problem is undecidable if it is a decision problem (a "yes or no" question) that cannot be solved. A problem is tractable if the number of steps required to solve the problem is bounded by a polynomial function of the size of the problem such as $T <= aN^k$ where T is the number of steps required to solve the problem, N is the size of the problem while a and k are constants [96].

Most description logics belong to the class of monotonic logics, in which adding new information monotonically increases the number of provable theorems and never falsifies a previous conclusion [97]. For example, assume we only know that Bora is a person. From this information we can neither conclude that Bora is a student, nor that he is not. Thus we admit the fact that our knowledge of the world is incomplete. This is called the open world assumption. If we subsequently learn that Bora is indeed a student, this does not change or disprove any previous statements but extends the knowledge. However, in certain cases this approach may not work perfectly. Consider a list of flight departure times. If it is not explicitly stated that a flight leaves at 10:21 PM, then we usually assume that there is no such flight. In other words, for this scenario the closed world assumption is used assuming that our knowledge about that part of the world is complete and we conclude that there is no flight at 10:21 PM unless we can prove the contrary. Such inference is non-monotonic, meaning that additional knowledge can invalidate previous conclusions.

For example finding out about a flight at 10:21 PM invalidates our earlier conjecture. Such systems can be useful for many applications, but they can also create problems of conflicting defaults. A frequently used example is the so-called Nixon Diamond which consists of the statements “Nixon is both a Quaker and a Republican.” “Quakers, by and large are pacifists.” “Republicans, by and large are not pacifists.” The question is whether we should infer that Nixon is a pacifist or that he is not pacifist based on this information i.e. conflicting default assumptions about members of certain groups [86].

Description Logics are a family of formal languages differing from each other with respect to the constructors they provide. Thus some description logics have a higher degree of expressiveness than others. $\mathcal{AL}$ (Attributive Language) is considered the simplest logic of interest for practical use [97] while $\mathcal{ALC}$ (Attributive Language with Complements) is what most languages are based upon. $\mathcal{C}$ denotes general negation and permits expressing disjunction and existential role restrictions [98]. $\mathcal{ALC}$ can be extended by additional constructs, such as $\mathcal{O}$, $\mathcal{I}$, $\mathcal{H}$, $\mathcal{Q}$ or $\mathcal{N}$ to obtain more expressive languages which will be discussed later. Table 2.1 gives a list of constructs provided by $\mathcal{ALC}$.

Table 2.1. $\mathcal{ALC}$ constructs



Description	DL Syntax	Examples
Concept	C, D….	Student, Male, Female, Arts, School
Role	R	likes, attends
Conjunction	C ⊓ D	Student ⊓ Male
Disjunction	C ⊔ D	Female ⊔ Male
Negation	¬ C	¬ Female
Exists restriction	∃R.C	∃attends.School
Value restriction	∀R.C	∀likes.Arts

Internet led to development of new knowledge representation languages that are based on web standards such as XML and RDF. XOL[15], OIL[16], DAML[17] and OWL[18] are some examples of these standards.

XML-Based Ontology Exchange Language (XOL) was originally a proposal for XML syntax for OKBC[19] Lite. OKBC is an application program interface (API) for accessing frame-based knowledge representation systems and is sometimes referred to as Generic Frame Protocol [99]. XOL was designed to provide a format for exchanging ontology definitions rather than development of ontologies [100]; however it influenced later ontology language specifications. OIL [101] was the first ontology language to combine elements from Description Logics, frame languages and web standards such as XML and RDF [102]. The modeling primitives of OIL are based on those of XOL. OIL extends XOL to make it more suitable for capturing ontologies defined using a logic-based approach (such as DLs) in addition to the frame-based ontologies for which OKBC; thus XOL were designed [100]. It was developed such that its semantics could be specified via mappings to the $\mathcal{SHIQ}$ Description Logic [103]. $\mathcal{SHIQ}$ is a more expressive description logic than $\mathcal{ALC}$ and supports transitive properties ($\mathcal{S}$ is short for $\mathcal{ALC}$ extended with transitive properties), role hierarchy e.g. sub-properties ($\mathcal{H}$), inverse properties ($\mathcal{I}$) and qualified cardinality restrictions ($\mathcal{Q}$) [97]. OIL has syntaxes for both XML and RDF. Figure 2.8 shows an excerpt from an OIL document presented in RDF/XML stating “Herbivores are animals that eat plants. An herbivore is not a carnivore”.

[15] XML-Based Ontology Exchange Language

[16] Ontology Inference Layer

[17] DARPA Agent Markup Language

[18] Web Ontology Language

[19] Open Knowledge Base Connectivity

```
<rdfs:Class rdf:ID="herbivore">
  <rdf:type rdf:resource="http://www.ontoknowledge.org/oil/rdf-schema/#DefinedClass"/>
      <rdfs:subClassOf rdf:resource="#animal"/>
      <rdfs:subClassOf>
         <oil:NOT>
            <oil:hasOperand rdf:resource="#carnivore"/>
         </oil:NOT>
      </rdfs:subClassOf>
   <oil:hasSlotConstraint>
      <oil:ValueType>
         <oil:hasProperty rdf:resource="#eats"/>
         <oil:hasClass rdf:resource="#plant"/>
      </oil:ValueType>
   </oil:hasSlotConstraint>
</rdfs:Class>
```

Figure 2.8. An OIL example in RDF/XML syntax

Around the same time DARPA was working on another language with similar goals. DARPA-ONT was designed as an extension to RDF with language constructors from frame-based knowledge representation languages [104]. Figure 2.9 shows excerpt from a DAML document presented in RDF/XML stating “Every animal has at least 1 parent which is also an animal”.

```
<daml:Class rdf:ID="Animal">
 <rdfs:subClassOf>
   <daml:Restriction daml:minCardinalityQ="1">
     <daml:onProperty rdf:resource="#hasParent"/>
     <daml:hasClassQ rdf:resource="#Animal" />
   </daml:Restriction>
  </rdfs:subClassOf>
</daml:Class>
```

Figure 2.9. A DAML example in RDF/XML syntax

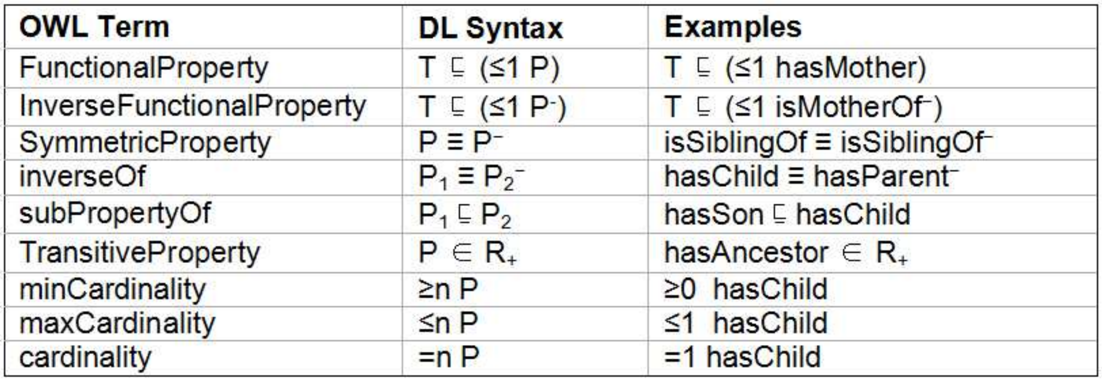
These two projects were eventually merged to form DAML + OIL which served as the basis for web ontology language (OWL); a W3C recommendation since February 2004 [105].

While traditional AI systems operated using a closed-world assumption, the Web operates based on open-world assumption, restricting itself to monotonic reasoning [106]. This is because on the Web, reasoning “needs to always take place in a potentially open ended situation: there is always the possibility that new information might arise from some other source, so one is never justified in assuming that one has ’all’ the facts about some topic” [106]. Thus OWL avoids non-monotonicity and closed world assumptions.

OWL has three different sub-languages with different levels of expressivity; OWL Lite, OWL DL and OWL Full. The OWL Lite and OWL DL species are syntactical variants of Description Logic languages; OWL Lite can be seen as a variant of the $\mathcal{SHIF}^{(\mathcal{D})}$ description logic language, whereas OWL DL is a variant of the $\mathcal{SHOIN}^{(\mathcal{D})}$ language [140]. Superscript ($\mathcal{D}$) indicates use of data type properties. Data type properties have objects such as strings, integers or dates rather than instances of classes e.g. Female, Human, Herbivore etc. OWL DL is a direct extension of OWL Lite. OWL Full extends both OWL DL and RDF(S) and cannot be translated into a Description Logic language. It is the most expressive of all three but is undecidable and offers no computational guarantees i.e. OWL Full doesn’t guarantee that computations will finish in finite time. OWL can be presented using RDF/XML or Notation 3.

OWL Lite extends $\mathcal{ALC}$ description logics with transitive properties ($\mathcal{S}$), role hierarchy e.g. sub-properties ($\mathcal{H}$), inverse properties ($\mathcal{I}$) and functional properties ($\mathcal{F}$). Table 2.2 shows these extensions with examples.

Table 2.2. $\mathcal{SHIF}$ extensions to $\mathcal{ALC}$ with OWL counterparts



OWL Term	DL Syntax	Examples
FunctionalProperty	$T \sqsubseteq (\leq 1\ P)$	$T \sqsubseteq (\leq 1\ \text{hasMother})$
InverseFunctionalProperty	$T \sqsubseteq (\leq 1\ P^-)$	$T \sqsubseteq (\leq 1\ \text{isMotherOf}^-)$
SymmetricProperty	$P \equiv P^-$	$\text{isSiblingOf} \equiv \text{isSiblingOf}^-$
inverseOf	$P_1 \equiv P_2^-$	$\text{hasChild} \equiv \text{hasParent}^-$
subPropertyOf	$P_1 \sqsubseteq P_2$	$\text{hasSon} \sqsubseteq \text{hasChild}$
TransitiveProperty	$P \in R_+$	$\text{hasAncestor} \in R_+$
minCardinality	$\geq n\ P$	$\geq 0\ \text{hasChild}$
maxCardinality	$\leq n\ P$	$\leq 1\ \text{hasChild}$
cardinality	$=n\ P$	$=1\ \text{hasChild}$

It should be noted that in OWL Lite n≤1 i.e. cardinalities are limited to 0 and 1. Minimum and maximum cardinality is used to set a range. However, if these two values are the same one can use “cardinality” instead of having two declarations. In OWL DL there’s no such limitation to the value of n. OWL Lite also doesn’t allow class declarations using unionOf and complementOf constructs. It allows class declarations with intersectionOf construct but only as intersection of named classes and property constraints. Figures 2.10 and 2.11 show legal and illegal uses of intersectionOf in OWL Lite, respectively. Figure 2.10 declares a class as the intersection of two named classes. Figure 2.11 uses a collection of individuals instead of named classes. OWL DL does not have these restrictions.

```
<owl:Class rdf:ID="Woman">
  <owl:intersectionOf rdf:parseType="Collection">
    <owl:Class rdf:about="#Female"/>
    <owl:Class rdf:about="#Human"/>
  </owl:intersectionOf>
</owl:Class/>
```

Figure 2.10. Legal use of intersectionOf construct in OWL Lite

```
<owl:Class>
  <owl:intersectionOf rdf:parseType="Collection">
    <owl:Class>
      <owl:oneOf rdf:parseType="Collection">
        <owl:Thing rdf:about="#CosiFanTutte" />
        <owl:Thing rdf:about="#DonGiovanni" />
      </owl:oneOf>
    </owl:Class>
    <owl:Class>
      <owl:oneOf rdf:parseType="Collection">
        <owl:Thing rdf:about="#DieZauberflöte" />
        <owl:Thing rdf:about="#DieEntführungAusDemSerail" />
      </owl:oneOf>
    </owl:Class>
  </owl:intersectionOf>
</owl:Class>
```

Figure 2.11. Illegal use of intersectionOf construct in OWL Lite

OWL DL extends $\mathcal{ALC}$ description logics with transitive properties ($\mathcal{S}$), role hierarchy e.g. sub-properties ($\mathcal{H}$), nominals ($\mathcal{O}$), inverse properties ($\mathcal{I}$) and unqualified cardinality restrictions ($\mathcal{N}$). Table 2.2 shows these extensions with examples. Earlier we mentioned qualified cardinality restrictions ($\mathcal{Q}$) about OIL. The two examples below can help understand the difference between qualified and unqualified cardinality restrictions. “Color blind people can see a certain number of colors” Here m is an integer representing that number.

$$\text{ColorBlindPerson} \equiv \text{Person} \sqcap \exists^{=m}\, \text{sees.Color}$$

This is a qualified number restriction since the objects connected through the role “sees” have to be of the specific type “Color”. A qualified restriction allows imposing restrictions on the number of objects connected through a certain role, counting only those objects that satisfy a certain condition whereas in the statement “Blind people don’t see anything”;

$$\text{BlindPerson} \equiv \text{Person} \sqcap \exists^{=0}\text{sees}$$

there's no restriction on the concept the role fillers belong to. This form is called unqualified restriction.

Table 2.3. OWL DL extensions to OWL Lite

OWL Term	**DL Syntax**	**Examples**
disjointWith	$C_1 \sqsubseteq \neg C_2$	Female $\sqsubseteq$ ¬Male
oneOf	$\{x_1 \ldots x_n\}$	{John, Jane, Bob}
hasValue	$\exists R.\{x\}$	∃citizenOf.{Turkey}

OWL Lite restrictions about the use of unionOf, intersectionOf, complementOf, minCardinality, maxCardinality and cardinality constructs have been discussed earlier and thus are not included in Table 2.3.

OWL 1.1 is currently at draft stage and will use the more expressive $\mathcal{SROIQ}$ description logics [107].

OWL has become a widely accepted format for publishing ontologies on the internet. Many tools have been developed to allow creation and reasoning of OWL ontologies. Protégé with OWL plugin [108], Altova Semantic Works [109] and SWOOP [110] are tools commonly used for ontology development in OWL.

2.4. Web Services

World Wide Web Consortium (W3C) defines web services as “programmatic interfaces for application to application communication on the World Wide Web” [111]. Web services rely on eXtensible Markup Language (XML) as the message exchange medium. The predominant industry standards for web services are Simple Object Access Protocol (SOAP) and Web Services Description Language (WSDL). SOAP is a protocol for exchange of information in a decentralized, distributed environment [112]. It allows invoking of functions on remote machines as in remote procedure calls (RPC). These functions i.e. web services are exposed to the outside world by a web server. A WSDL is used to describe a service, the operations it supports and structure of exchanged messages (similar to XML-XML Schema relationship). Common Object Request Broker Architecture (CORBA) and XML-RPC are some predecessors of SOAP web services. A relatively well known alternative to SOAP is Representational State Transfer (REST) software architecture which relies solely on HTTP PUT, GET, POST and DELETE methods associating them with CREATE, READ, UPDATE and DELETE operations in database technologies [113].

Web services provide platform-independent data exchange, e.g. a web service developed using JAVA, running on a UNIX server can be consumed by software developed in Visual Basic on a Windows operating system. To ensure web service interoperability, Web Services Interoperability Organization (WS-I) was founded. WS-I is governed by industry heavy weights such as Microsoft, IBM, Sun Microsystems, Oracle, Intel, Fujitsu and Hewlett-Packard. It provides interoperable profiles and test tools to help determine profile conformance [114].

Software developers provide tools to allow the integration of the data streams from web services directly into commercial products like Matlab, Excel and ArcGIS. Moreover other web servers can take advantage of web services to produce new, aggregate services. For example, a web service can get dissolved oxygen data from one service, do the unit conversion using another, pass it on to another service for plotting and return the URL of the plotted image. In this regard web services can also be considered as building blocks for more complicated processes which can also be exposed as web services.

2.5. AJAX

AJAX (Asynchronous JavaScript And XML) [115] is a web development technique used for creating rich, interactive web applications. In this model, changes are made to individual user interface components on a web page, as opposed to the conventional approach which is refreshing the entire page. This allows web applications to exchange data with the server behind the scenes without interrupting user experience [116]. A key advantage of AJAX applications is that they look and act more like desktop applications. It is also argued that AJAX applications outperform traditional Web applications [117]. AJAX is not a technology per se, but a term that refers to the use of a group of technologies some of which are mentioned in the acronym.

According to Garret [115], AJAX incorporates: markup and styling using XHTML[20] and Cascading Style Sheets (CSS)[21], interacting with components of the

[20] XHTML is a version of HTML that is conforms to XML syntax

[21] Cascading Style Sheets are documents containing information on how a page will be displayed (color, size, transparency, visibility, location etc.) in a browser window

web page through the Document Object Model[22] (DOM), asynchronous data exchange using XMLHttpRequest[23] and JavaScript to bind everything together. It focuses on the client side of the web application setting [118]. XML and JSON[24] (JavaScript Object Notation) are commonly used as the format for data transfer between the server and client. These files may be created dynamically by some form of server-side scripting and also by remote web services.

However AJAX-style programming also introduces some problems and potential problems to the browsing experience. First, dynamical partial page updates are not registered with the browser history, so triggering the "Back" function of the browser doesn’t make the page return to its previous state. A similar problem arises if a user wants to bookmark a particular state of the application. Second, AJAX relies on JavaScript, which is implemented differently by different browsers (e.g. Firefox, Safari, IE) or versions of a particular browser. Because of this, JavaScript code may need to be written more than once, each instance specifically tailored to different browsers and/or versions. It should also be noted that browsers may have varying levels of JavaScript support and capabilities thus it is not always possible to implement the same function in multiple browsers. Third, due to security reasons JavaScript may be disabled on certain browsers. Last, but not least, network latency (the interval between user request and server response) could pose a problem for user experience. Since page updates do not require a full page unload and load, the fact that a user’s request is being processed is not indicated by the browser itself. Thus in AJAX applications users are often provided with visual feedback on the

[22] Document Object Model is an application programming interface (API) for HTML and XML documents that defines their logical structure and the way they are accessed and manipulated.

[23] XmlHttpRequest is an interface that allows scripts to perform functions such as submitting form data or loading data from a server.

[24] A lightweight data interchange format

status of the background activity (such as progress bars) so that they don not assume the application has become unresponsive.

2.6. XPath

W3C defines XPath as “a language for addressing parts of an XML document” [119]. It can also be used for testing whether or not a node matches a pattern. XPath models an XML document as a tree of nodes and uses a compact, non-XML, syntax to address them. Figure 2.12 shows an XML document example.

```
<observationData>
<measurement dataType=”Instantaneous”>
<value>10.3</value>
<units>cfs</units>
<parameter>Discharge</parameter>
</measurement>
<measurement dataType=”Daily Average”>
<value>20.8</value>
<units>degreesC</units>
<parameter>Water Temperature</parameter>
</measurement>
<measurement dataType=” Daily Average”>
<value>8.1</value>
<units>cfs</units>
<parameter>Discharge</parameter>
</measurement>
</observationData>
```

Figure 2.12. A simple XML document for XPath demonstration

In this XML document one can select all measurement elements with dataType instantaneous using XPath expression;

```
/observationData/measurement[@dataType='Instantaneous']
```

Or value element of all the measurements with parameter water temperature using;

```
/observationData/measurement[parameter='Water Temperature']/value
```

XPath also allows mathematical operators and selection of multiple paths at once. The following XPath expression selects discharge measurements that have values greater than 9 cfs and water temperature measurements with values greater than 15 degrees Celcius at the same time.

```
/observationData/measurement[parameter='Discharge' and value>9 and units=’cfs’] |
/observationData/measurement[parameter='Water Temperature' and value>15 and
units=’degreesC’]
```

An XPath expression returns either a node-set, a string, a Boolean, or a number. Once nodes are retrieved, values can be read or set/updated as necessary using XML processing tools. XPath can be used with other XML technologies such as XSL (Extensible Stylesheet Language) to render different parts of an XML document in different styles in addition to extracting data from, and updating, XML documents. An example of use of XPath and XSL to render an ISO 19139 metadata document into a human-readable HTML page can be found at;

http://cbe.cae.drexel.edu/isoschemas/ODMMetadata.xml

All the relevant files (XML, XSL and output) are also provided in appendices.

CHAPTER 3: REVIEW OF PREVIOUS STUDIES

Ontology[25]-based search is an emerging field with applications in numerous areas, computer/information science, medicine and geosciences, to name a few. Systems can be classified roughly into two categories; full-text and keyword search. While the common goal is increasing the search accuracy, methods vary. Widyantoro and Yen developed a “fuzzy” search engine to refine free-text search results and applied it on a database with 584 scientific abstracts. [120,121]. Fuzziness comes from the weighting process involved based on the frequency of occurrence of a given word in an abstract identified using a particular keyword by the author. During text mining authors sought certain grammar patterns in the abstracts such as “adjective adjective noun” while adjectives and nouns were identified using the WordNet[26] dictionary. Ontology used in search was automatically created based on the phrases from the abstract body, title and manually entered keywords and were used to suggest to the user more specific keywords to narrow down the search in addition to listing relevant abstracts. Dey et al. used a different weighting system where relevant keywords were identified and rated based on their distance from the provided keyword within the ontology [122]. This system used simple math to calculate the weights/distances of concepts e.g. equivalence gets the highest possible rating while, for a concept that is union of several other concepts, points are divided evenly among the concepts that form the union. Dey’s system expanded a user’s query by adding several relevant keywords from the ontologies. Results that exceed a threshold value) were displayed and ordered by relevance based on the same scores. In this scenario, unlike the previous example, ontologies were created

[25] A way of expressing knowledge in machine readable format. Ontologies are examined in detail in the previous chapter.

[26] WordNet is a large lexical database for the English language

manually [122]. Zhang et al. used a similar ranking scheme to query a database of 477 scientific papers [123]. Kim used an ontology-based search engine to improve search efficiency for querying documents of specific business interests and uses World Bank and OECD as examples. In his study ontologies were created manually while IT specialists interacted with domain experts to understand the business logic and terminology so they can be accurately represented in the ontology. His study focused on querying using controlled vocabularies, rather than free-text [124]. Suomela and Kekäläinen evaluated usefulness of ontology as a search tool. In this study 16 users queried a database of newspaper articles about “Food” with and without conceptual support from the ontology. In one case they were asked to formulate a query (as in Google) themselves while in the second scenario, an ontology visualized in a tree structure was presented to the users allowing them to run queries by clicking on the nodes. Clicking on a node also returned results of the child nodes. Users’ reactions were positive, since the system made the search easier by eliminating guess work and was visually appealing. However, results showed that the search results without ontology support were slightly better [125]. On the other hand Sure and Iosif found that ontology-based search tools are at least as good as the keyword based-tools [126]. The reason for this contradiction will be discussed later in this section. In medical informatics ontology-based search mechanisms are often used with vocabularies such as Gene Ontology (GO) and Unified Medical Language System (UMLS) rather than free text and within relevant repositories such as Gene Ontology Annotation (GOA) Database, GenBank[27], or SwissProt[28][127,128]. The studies that aim to use ontological knowledge in querying medical publications,

[27] A gene database maintained by National Center for Biotechnology Information (NCBI)
[28] A protein database maintained by European Bioinformatics Institute

query authoritative sources e.g. MEDLINE, MERCK rather than the general Internet [129].

GEON (Geosciences Network) uses Semantic Web for Earth and Environmental Terminology (SWEET) [130] ontology and other more specialized ontologies. A common use is in temporal queries that involve geological time scales. A query for Mesozoic era also returns periods within the era such as the Jurassic period. Knowledge that the Jurassic period is part of Mesozoic era is gathered from the USGS Geology Time Scale Ontology [131].

Knowledge Sifter searches images on the Internet given the name of a place. It finds concept synonyms from WordNet, and locations of the geographic feature or places using National Geospatial Intelligence Agency's GEOnet Names Server (GNS) and USGS Geographic Names Information System (GNIS). The search string is sent to Google Images, Yahoo Images and Terra Server and the resulting images are displayed on the screen. Using the coordinates obtained from GNIS and GNS, points can also be viewed in Google Maps [132].

Noesis uses Linked Environment for Atmospheric Discovery (LEAD) ontology as the knowledge base and searches Yahoo, Google and Digital Library for Earth System Education (DLESE). When a user types in a keyword and hits the search button, Noesis retrieves synonyms and related concepts from the ontology with definitions of those concepts from the American Meteorological Society website. After the user selects the keywords, the list of keywords is directed to search interfaces Yahoo, Google and DLESE connected with logical operators (AND, OR) and links to individual web pages grouped by data source are displayed on the screen [133]. Noesis targets the atmospheric science community.

Hydroogle uses an ontology for the hydrologic science domain as the knowledge base and uses Google as the search component. When a user types in a keyword and hits the search button, Hydroogle retrieves synonyms and related concepts from the ontology. After the user selects the keywords, the list of keywords is directed to Google and the results are displayed on the screen [134].

As Suomela and Kekäläinen showed, searching free-text using ontologies may not always work for several reasons. For example for a keyword entry “River”, Hydroogle suggests additional keywords “Water Body” and “Stream”. Even though these concepts are related to “River”, less than 1/20 of websites indexed by Google acknowledge the fact that “River is a water body” within their text. Since search capability is bounded by the search engine (e.g. Google) and decisions are absolute (no fuzzy/weighted search capability as in [120,121,122,123,135]) a miscalculated search keyword can eliminate many useful results. Also when the knowledge in the ontologies are not created based on documents being queried but rely solely on ontology engineer’s decisions, the extent to which ontology can help is limited by the domain knowledge of the authors and what they consider the scope and detail of the ontology should be. This is not necessarily in alignment with the real world.

The Global Change Master Directory (GCMD) offers an extensive set of keywords and allows users to browse through the datasets indexed by these keywords in a hierarchical fashion [136]. The search can also be refined using additional free-text entries. GCMD is an effort by NASA. The keywords are used by several national and foreign agencies especially for remote sensing and related data. GCMD covers a lot of ground but lacks domain specific elements. A total of 30 elements are offered under the “surface water” and “ground water” categories, in addition to 29 elements under the “water quality” category. The system provides links to websites where data can be downloaded or ordered.

Hydroseek uses ontologies to reconcile differences between controlled vocabularies of data repositories and classify the search results. It searches data stored in hydrologic data repositories rather than contents of web pages and in these regards resembles the ontology applications in medical informatics and the GEON system [127,128,130]. *Hydroseek* contains domain specific keywords necessary for detailed classification of search results. However, it also allows querying using GCMD keywords via web services. Components of *Hydroseek* are examined in detail in the following chapter.

CHAPTER 4: DEVELOPMENT STRAGEGY

4.1. Knowledge Base

The biggest challenge in seamlessly integrating multiple data sources is to resolve heterogeneity issues. Semantic heterogeneity occurs when there is a disagreement about the meaning, interpretation or intended use of the same or related data [137]. If we consider EPA STORET and NWIS systems as examples, these systems often refer to the same observation using different parameter codes/names. On the other hand, two terms can sometimes have similar meanings, but not quite the same. Cui and O'Brien (2001) identify "generalization & specification" and "overlapping concepts" as two flavors of semantic heterogeneity of this kind [138]. In this dissertation the former will be referred to as "difference in granularity". A good example is NWIS' Ammonia Nitrogen (Bed Sediment) and Ammonia Nitrogen (Suspended Sediment) parameters versus EPA STORET's Ammonia Nitrogen (Sediment) parameter. Sometimes data sources do not contain sufficient information to resolve such differences.

Structural, syntactic and information system heterogeneities emerge as other types of incompatibilities. Ouskel and Sheth define structural heterogeneity as different information systems storing their data in different document layouts and formats [139]. In the current state of hydrologic data providers it is possible to speak of HTML tables, XML documents or text files where the file format alone does not guarantee homogeneity since data output can be organized in many different ways. Syntactic heterogeneity is the presence of different representations or encodings of data. Date/time formats can be given as an example where common differences are;

local time vs. UTC, 12 hour clock vs. 24 hour clock and standard date format vs. Julian day which is common in Ameriflux data. Whereas information system heterogeneity requires methods of communication specifically tailored to interact with each data providers' servers due to the difference in interfaces, e.g. REST services vs. SOAP services and different arguments that each service takes. Sometimes even responses and requests have different formats. In EPA STORET, data requests require dates to be provided in Dublin Julian days[29] while the server returns Gregorian dates with data.

Even in the absence of heterogeneity data can be hard to manage due to the overwhelming variety of observations. In most data access portals this directly affects the user experience, appearing as tens of web forms and subsequent form submissions. For example, NWIS has more than 175 form elements on a single query page [140]. This leads to the emergence of the term 'expert user' even for a 'simple' data discovery and retrieval process. For content aggregators/data portals this means an even higher number of parameters considering the fact that these systems offer data from multiple sources.

In this dissertation *Hydroseek*, an ontology-aided search, coupled with a clustered navigation system, that relies on web services for consistent data output is proposed as the solution to these problems.

According to Noy and McGuinness the first fundamental rule for ontology engineering is "There is no correct way to model a domain – there are always viable alternatives. The best solution almost always depends on the application you have in

[29] Days since the noon between December 31st , 1899 and January 1st , 1990

mind and the extensions that you anticipate.” [141]. Likewise the architecture of ontologies used in this system were shaped by the problems hydrologic community is trying to solve and the scope of the data repositories of interest. These ontologies are used both for searching (local/remote sources) and mapping (linking variable names in the database with concepts in the ontology). There are several problems that need to be addressed.

Problem 1

With free-text search, sometimes finding useful data is like looking for a needle in a haystack (high recall, low precision) while using additional search keywords can eliminate useful data from the search results (high precision, low recall).

Solution

Hydroseek has the advantage of working with a controlled list of keywords, rather than free text. So instead of trying to interpret pages of text, the algorithm deals with a parameter code or a phrase that identifies a particular type of measurement. Thus the system does not need a fuzzy search scheme. A thesaurus relates terms using broader/narrower term (hierarchy) and preferred/non-preferred term (synonymy) relationships. A list of measured parameters can be organized in a thesaurus and the user can be asked to make selections from this list for both indexing and searching the data. Most hydrologic data providers have their own controlled vocabularies which can be mapped to a central thesaurus to overcome semantic interoperability problem. If a data provider does not have a controlled vocabulary it can be reverse-engineered from the keywords used in the past.

Hydroseek follows three basic rules to enhance search accuracy.

1. Datasets should be indexed at the highest possible detail.

For example use of keyword ‘Cadmium’ should be preferred over ‘Heavy Metal’ since Cadmium, being a heavy metal, can be inferred from the concept hierarchy while keyword ‘Heavy Metal’ does not give any clues as to what kind of heavy metal the dataset is about. Indexing application and extensibility of a keyword list are examined in Chapter 5.

2. Queries should be limited to a reasonable level of detail.

To avoid a ‘high precision, low or no recall’ problem the user should not be exposed to keywords that are too specific. For example the ontology used here shows that ‘Stream Stage’ can be provided with reference to 3 different datums (MSL, NGVD29, NAVD88). In this case, for any given station that offers stream gage measurements, hiding datum options the from user triples the chances of getting a result. This also does not sacrifice from the needed data since these values can be easily converted to one another. A more drastic example can be given from NWIS which has about 10,000 searchable terms without any hierarchical linkage while each station measures about 20 parameters. So for any given station, the chance of getting a result for a randomly picked parameter is 20/10,000 or 0.002.

3. Results should be clustered at the highest level of detail possible.

Without the third rule, the second rule is likely to create the ‘low precision, high recall’ problem mentioned earlier. At this point ‘detailed classification’ (e.g. datum options in the previous example) previously hidden from the user should be used to group the search results.

The varying levels of detail required for each procedure shows that a layered model is the best option. Thus the ontology or thesaurus should consist of several pieces, each representing different level of detail that can be used in different combinations depending on the purpose of use.

Problem 2

Breaking parameter names into pieces jeopardizes the discoverability of data. Sometimes errors arise in unexpected places. For example, in EPA STORET 55.3% of air temperature measurements are indexed as “Medium=Water”. While this is an apparent error, for some measurements medium is rather unclear. For parameter ‘Precipitation’ medium can be air or water depending on whether user considers precipitation an atmospheric phenomenon or water falling to earth’s surface. For EPA STORET only 37.7% of precipitation data belongs in the first category.

Solution

Parameter names can be formulated to embody the measurement medium information as well. This will increase the number of terms but the right user interface design can eliminate any effect, this might have on the user. However not all the parameters need to be associated with a medium in a search because sometimes there is either no alternative medium, and the medium is obvious (as in Wind Speed) or the medium is ambiguous (as in Precipitation).

Different language alternatives were considered for knowledge base development. The SKOS[30] Core Vocabulary is a set of RDF properties and RDFS classes and is suitable for development of taxonomies and thesauri. While SKOS [142] was sufficient for broader concept, narrower concept and synonymy relationships, it was necessary to introduce new relationships and impose certain restrictions on concepts at times. As a result, the knowledge base evolved from a thesaurus into a light-weight ontology; a hierarchical structure with few properties presented in OWL (Web Ontology Language); specifically using OWL-DL sublanguage.

This collection of ontologies serves several purposes. It provides:

- The knowledge base for the search algorithm
- The vocabulary for form based (AJAX auto-complete) search
- The vocabulary and structure for visualization (clustering) of search results
- The vocabulary and structure for mapping application

The ontology model consists of four layers (Fig. 4.1). The 'Core' and 'Compound' layers provide the vocabulary of search keywords used in the auto-complete function. The 'Core' layer consists of several ontologies each focusing on a certain domain e.g. meteorology or flow. The 'Detail' layer contains further classification of concepts in the 'Core' layer while the 'Navigation' layer is a single ontology that consists only of higher level concepts to make the ontology easier to navigate when visualized. For example, the term 'Evapotranspiration' would be in the

[30] Simple Knowledge Organization System

'Core' layer while classification of methods for calculating potential evapotranspiration, for example the Penman method or the Priestley-Taylor method would be in the 'Detail' layer. Conversely, a broad concept like water quality belongs in the 'Navigation' layer, while a term such as flood would be placed in the 'Compound' layer which contains concepts that are related to elements from multiple ontologies in the 'Core' layer. The 'Compound' layer can also be used for adding terms from other keyword lists such as GCMD allowing the system to be queried using other controlled vocabularies. This approach makes the system semantically interoperable with other systems without losing any functionality.

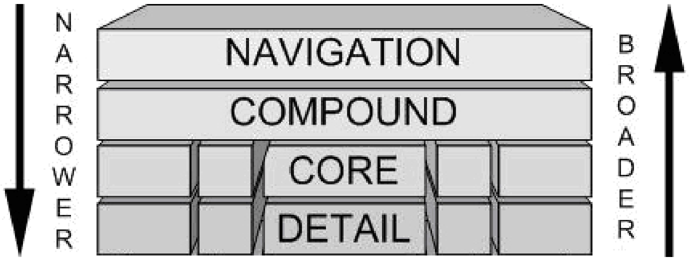


Figure 4.1. Layered ontology model used in the system

Different purposes require the use of different layers. For example the search algorithm uses the three lower layers shown in Fig. 4.1. The search form is populated using the 'Core' and 'Compound' layers. Visual concept-database mapping uses all four layers. For the mapping application, while all the layers are displayed, only the lowest layer can be selected. The navigation layer makes keyword selection more manageable by grouping them under broad concepts. Alternative paths may exist to the same keyword i.e. a given concept can have multiple parents. For example Zinc is both a heavy metal and a micronutrient.

Ontology development is an iterative process. *Hydroseek* ontologies will also evolve in time. As new data repositories are included in the system concept hierarchies can be reorganized and new keywords can be added without disrupting operation. Currently the ontologies cover groundwater, atmosphere and surface water concepts with more than 300 categories.

4.2. Metadata Library

In addition to the ontologies another central component in the system is the metadata library. The Metadata library holds information about data providers. Collected metadata is mostly static in nature, such as station lists and relevant metadata. This metadata can be used to enrich the returned search results even though the metadata library is used mainly to identify sites matching the search criteria submitted by users. The library is implemented as a relational database and populated with metadata harvested from data providers' websites programmatically using screen-scrapers. Screen scraping is a technique by which a computer program extracts data from the output of another program which is meant to interact with human users.

When dealing with data at remote sources, local metadata catalogs can be quite useful for several reasons:

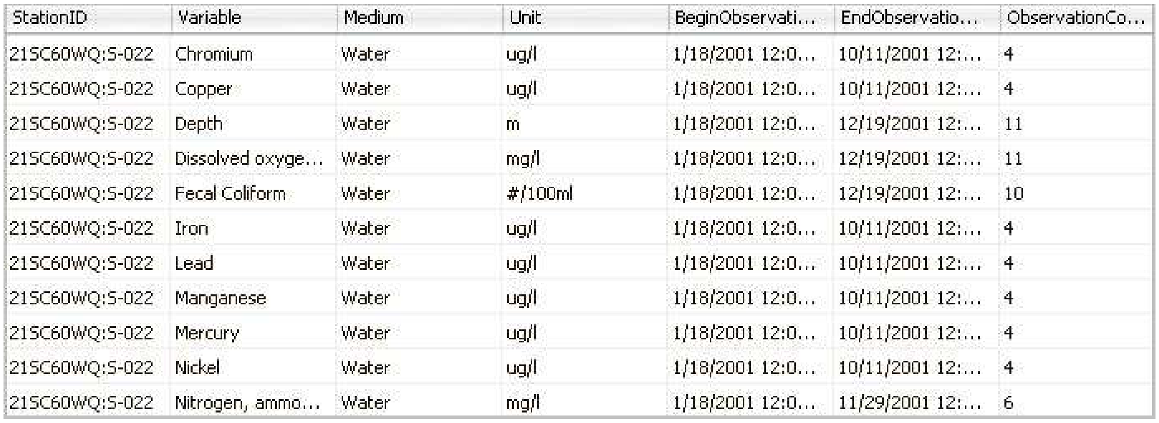
1. Better query performance – The performance of web scrapers relies on the functionality of the web pages they interact with. Sometimes a simple request may require that a chain of functions be invoked by the web scrapers thus increases the response time. With a local database this operation can be performed much faster.

2. Freedom - Catalogs created by scraping data from different pages give different query options and simplifies the process. For example using the Chesapeake Information Management System's (CIMS) interface, it is not possible to locate stations by their coordinates although coordinates for each station are known. However, once all the station data is in the local database, the missing search functionality can be easily implemented.
3. Fewer errors - Original metadata may be incomplete or incorrect and it may be possible to fix these errors in the local catalog. Table 5.1 shows the availability of geographic identifiers for EPA STORET stations.

Table 5.1. Availability of geographic identifiers for stations in EPA STORET

Total Number of Sites	274,918
Sites with geographic coordinates	274,435
Sites with State/County information	273,113
Sites with Hydrologic Unit Codes	128,646

The metadata library contains information about measurement site locations, names, a list of measured parameters for each site, measurement periods and the number of available data points for each variable measured. It does not contain actual measurement values. In this system sites matching a particular search criteria are identified using the local metadata library, while the actual data is retrieved from the data provider behind the scenes and passed on to the user. Figure 4.2 shows a screenshot from the metadata library.



StationID	Variable	Medium	Unit	BeginObservati...	EndObservatio...	ObservationCo...
21SC60WQ:S-022	Chromium	Water	ug/l	1/18/2001 12:0...	10/11/2001 12:...	4
21SC60WQ:S-022	Copper	Water	ug/l	1/18/2001 12:0...	10/11/2001 12:...	4
21SC60WQ:S-022	Depth	Water	m	1/18/2001 12:0...	12/19/2001 12:...	11
21SC60WQ:S-022	Dissolved oxyge...	Water	mg/l	1/18/2001 12:0...	12/19/2001 12:...	11
21SC60WQ:S-022	Fecal Coliform	Water	#/100ml	1/18/2001 12:0...	12/19/2001 12:...	10
21SC60WQ:S-022	Iron	Water	ug/l	1/18/2001 12:0...	10/11/2001 12:...	4
21SC60WQ:S-022	Lead	Water	ug/l	1/18/2001 12:0...	10/11/2001 12:...	4
21SC60WQ:S-022	Manganese	Water	ug/l	1/18/2001 12:0...	10/11/2001 12:...	4
21SC60WQ:S-022	Mercury	Water	ug/l	1/18/2001 12:0...	10/11/2001 12:...	4
21SC60WQ:S-022	Nickel	Water	ug/l	1/18/2001 12:0...	10/11/2001 12:...	4
21SC60WQ:S-022	Nitrogen, ammo...	Water	mg/l	1/18/2001 12:0...	11/29/2001 12:...	6

Figure 4.2. Metadata Library on SQL Server 2005

Hydroseek uses an extended version of the Observations Data Model (ODM) database schema [143] developed at Utah State University for the CUAHSI Hydrologic Information Systems project.

Hydrologic data repositories covered by the system are USGS NWIS, EPA STORET, TCEQ (Texas Commission on Environmental Quality) TRACS (Texas Regulatory and Compliance System), Chesapeake Information Management Systems (CIMS), Penn State University's RTHNet observatory database and database of Burd Run Interdisciplinary Watershed Research Laboratory at Shippensburg University of Pennsylvania. These six sources range from largest networks in the US to the smallest (single station) as an indicator of system's ability to perform at all scales.

Data catalogs need to be updated regularly since the amount of data increases over time. However harvesting of a catalog is a time consuming process and keeping it up-to-date costs several gigabytes of bandwidth to major data repositories monthly. This issue led to an agreement between CUAHSI, NWIS and

STORET and *HydroSeek* recently started receiving periodic database dumps from these agencies. TCEQ TRACS catalog is contributed by University of Texas at Austin.

4.2.1. Metadata Enhancements

Some of the missing metadata can be restored using the available metadata. For example, sites that are not associated with a state or hydrologic unit (Table 4.1) can be improved by adding this information if coordinates are available. For the *Hydroseek* system hydrologic unit codes (HUC) were of special importance since the system also allows searching by watershed name. To achieve this, station identifiers and coordinates of sites with missing metadata were exported using the SQL Server Integration Services (SSIS) and imported into ArcMap. When superimposed on a political boundaries map, the intersection of two layers associated each point with an administrative area, while using the NHD+ dataset these points were linked with their hydrologic units. However, this method did not work for adding state information to sites located at the sea or at estuaries. Figure 4.3 shows the process used for these sites on using the Chesapeake Bay example. Shown in Figure 4.3 are the actual screenshots from different stages of the process.

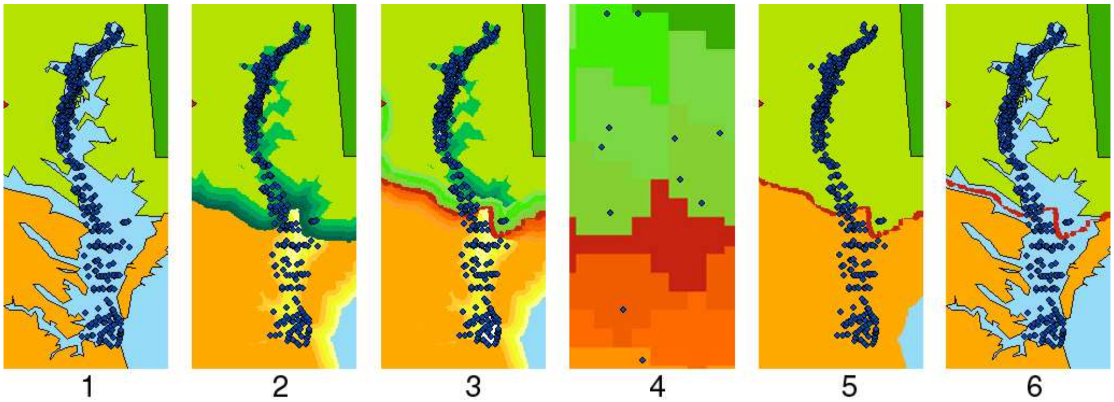


Figure 4.3. Process used to associate states with sites at the sea and at estuaries

In Figure 4.3 Maryland and Virginia are shown in light green and orange respectively. Delaware and the District of Columbia appear in the first map as dark green and red respectively. In this case the Euclidian distance from Maryland and Virginia were calculated to create a buffer around the states. Distances were assigned different signs e.g. farthest from Maryland (-1), farthest from Virginia (+1) with increasing absolute values as the points get closer to each state which made them a measure of proximity. When these two layers were added negative areas indicated the zone for Maryland, positive areas for Virginia while a value of 0 was equidistant from both states represented as the red line in images 3 through 6. In this example no stations were on the red line. Had that been the case, the same process could be repeated with smaller raster cells. Image 4 shows a close-up that shows stations on either side of the red line. Stations appearing in the green zone of image 5 were associated with Maryland while the rest were associated with Virginia. Improving and reconciling differences in metadata among repositories also allows one to take a look at the state of data availability over all 6 networks from various aspects which will be examined in section 4.2.2.

4.2.2. Data Availability

Consistency of metadata makes it possible to make comparisons between different data repositories on the basis of data availability, site distribution, data priorities and to aggregate the results to see the big picture. To facilitate this analysis Online Analytical Processing (OLAP) technology was employed. OLAP uses a multidimensional data model, allowing for complex queries with significantly reduced execution times. This model supports pre-computed aggregations of records such that summaries based on certain data attributes can easily be created [144]. Maps were generated using ArcMap based on data from OLAP data cubes. Figure 4.4 shows the distribution of sites on a map of Continental US for the entire system i.e. the total of 6 networks covered.

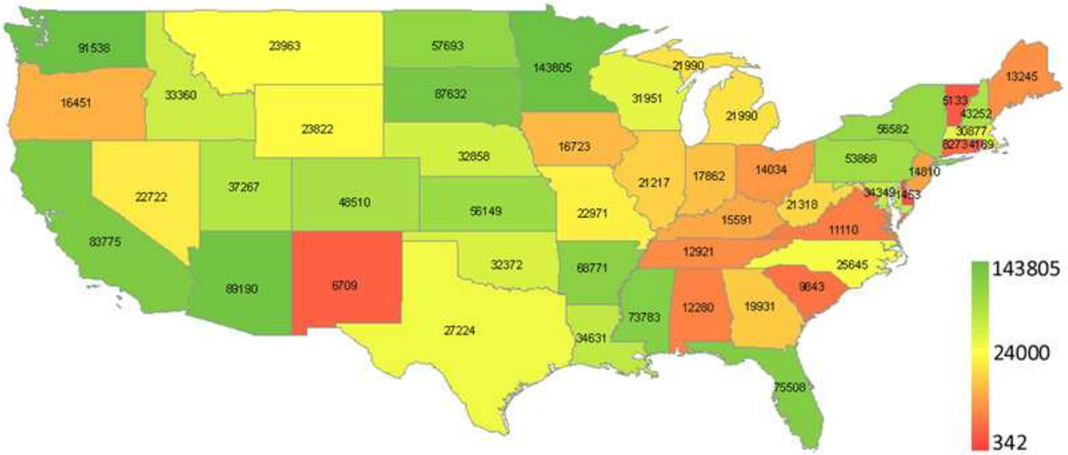


Figure 4.4. Geographical distribution of sites

The sites not visible on this map are shown in Figure 4.5. Coverage is not limited to the US but is also available from countries such as Canada, Mexico, Ukraine, Japan, Afghanistan, Iraq and several Pacific and Caribbean islands.

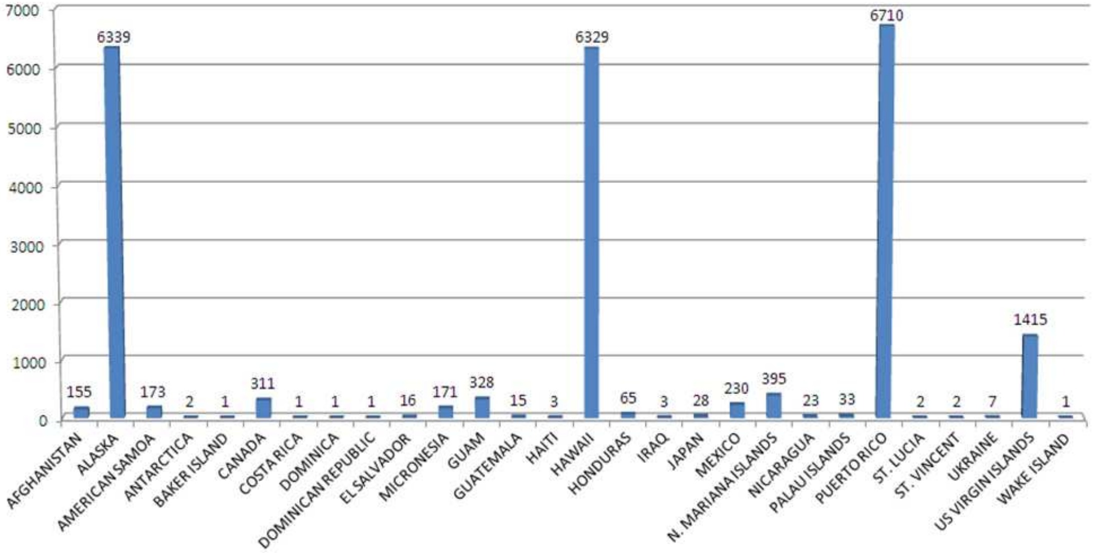


Figure 4.5. Sites shown in Figure 4.4

If sites are classified into seven categories, namely; stream/river, groundwater, lake/reservoir, estuary, coastal, meteorological and other, their weights in total site count for each data repository can be provided as shown in Figure 4.6.

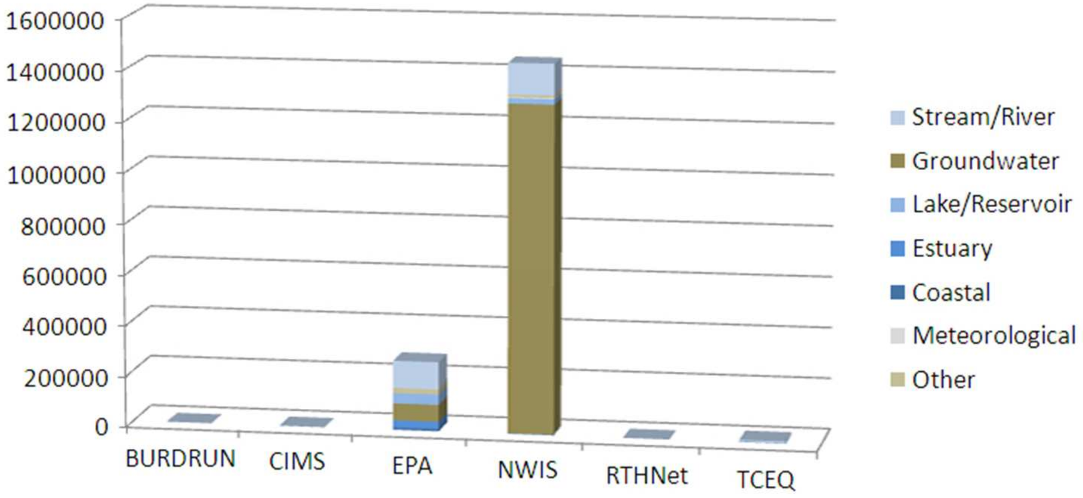


Figure 4.6. Site types for all networks

Since all repositories have different names for station types and different levels of detail for classifying them, aggregating sites for this plot required reconciliation of these differences. For detailed classification schemes like those used by STORET and NWIS, sites were grouped according to their primary types. EPA's "Well" site type, as well as NWIS' "Groundwater" and "Spring" were listed under the category "Groundwater". For systems that didn't differentiate between "Coastal" and "Estuary" types, stations were plotted on the map and sites in the areas that received fresh water from rivers (e.g. Corpus Christi Bay, Matagorda Bay, Aransas Bay and Chesapeake Bay) were classified under "Estuary" type. "Other" represents sites in wetlands, facilities (e.g. treatment plant effluent) and man-made drainage channels. In Figure 4.6 it can be easily seen that NWIS' groundwater sites comprise the majority of the sites. Based on this figure one might be led to assume that available groundwater data would comprise a considerable portion of total data. However, Figure 4.7 shows data availability for each network which contradicts this assumption.

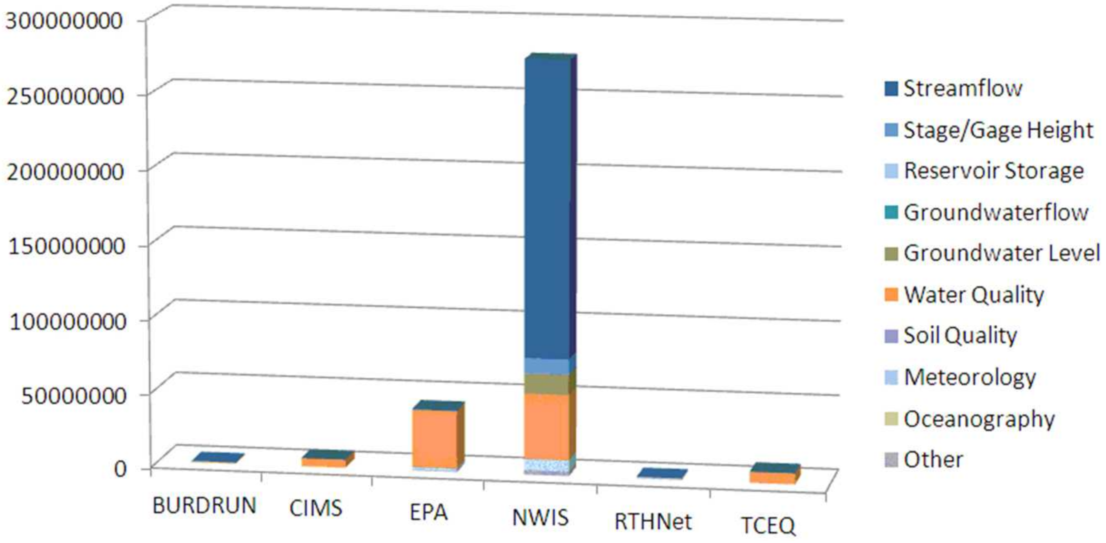


Figure 4.7. Data availability for all networks

Figure 4.7 shows a total of about 350 million data points most of which are stream flow data from NWIS. It also shows that groundwater data is only a small portion of available NWIS data and that EPA and NWIS have similar amounts of water quality data. However, since there's a significant difference between the repositories it is not possible to see the data distribution for smaller networks such as CIMS, TCEQ, BurdRun and RTHNet. Figure 4.8 provides a much better visualization of weights of different data types per network. Except RTHNet, which contains significant amount of precipitation data, it can be seen from Figure 4.9 that water quality data comprises a considerable portion of each network's archives.

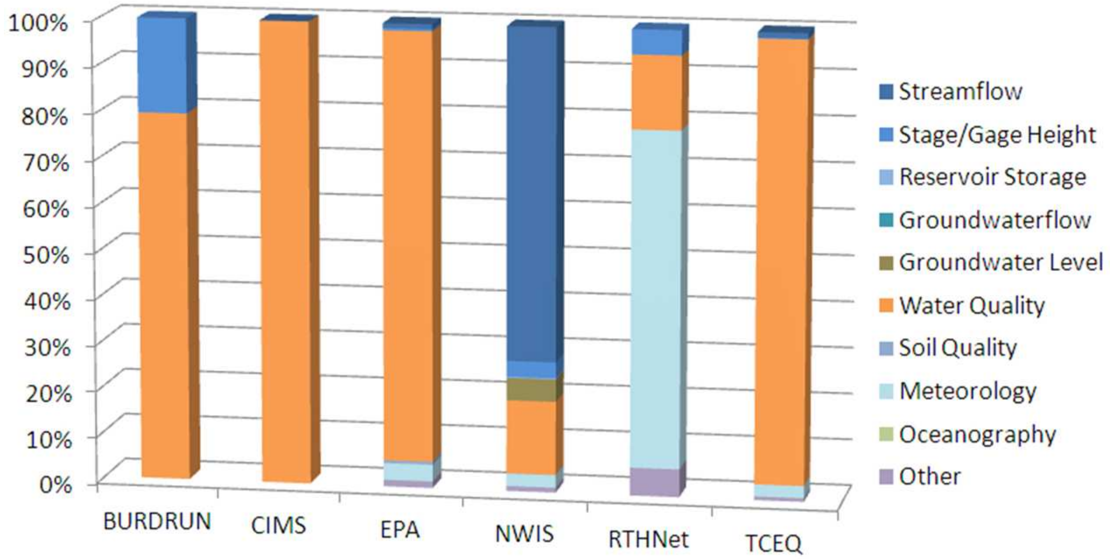


Figure 4.8. Percentage of different data types for each network

Following the problem shown in Figure 4.7 and 4.8 regarding disproportional amount of groundwater data in comparison with site availability, examination of the database dump provided by USGS reveals that most groundwater sites do not have data available to the public.

The following link points to one of those sites on NWIS web.

http://waterdata.usgs.gov/nwis/inventory/?site_no=374028101001001

Updating Figure 4.6 to show sites with publicly available data gives Figure 4.9. While groundwater sites still comprise a considerable portion of the total number, there are about 1 million fewer than shown in Figure 4.6. Figure 4.10 shows that stream, coastal and estuary station types comprise the most stations in the majority of networks.

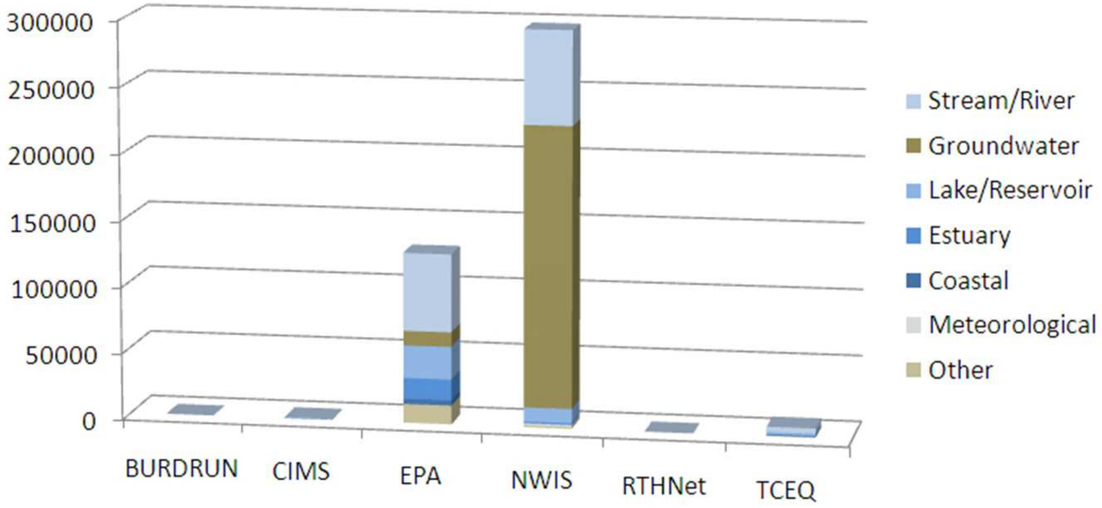


Figure 4.9. Sites with data for all networks

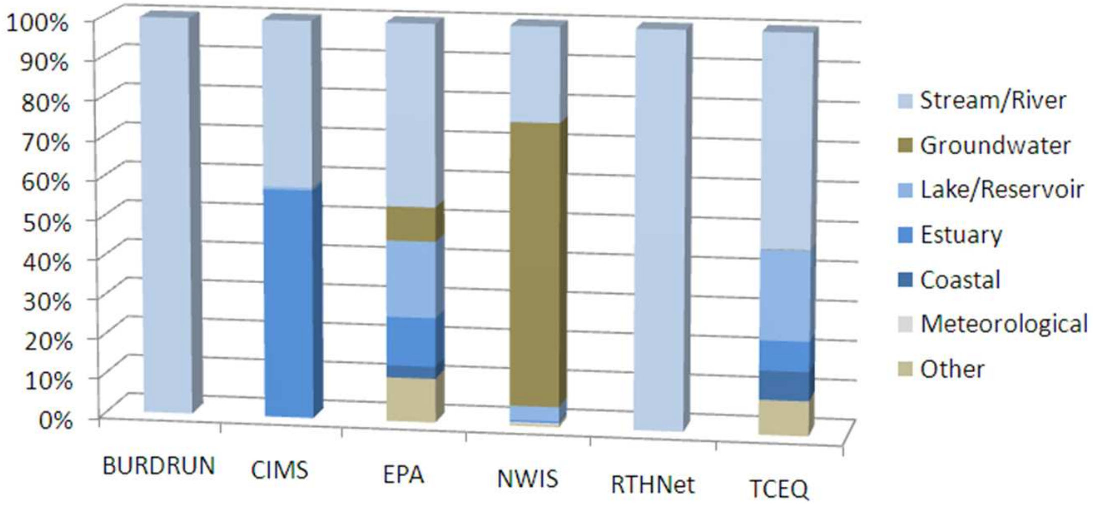


Figure 4.10. Percentage of different station types for each network

Figure 4.11 shows coverage change from 1800 to 2007 for the aggregation of 6 networks. To generate this figure, it is necessary to know the dates the sites were founded. However this information is not readily available at the original sources. The catalogs, on the other hand, contain measurement start date and end date for each parameter at a given site. Thus to generate Figure 4.11 the earliest of the start dates for each station was considered the date it was founded. In this figure states with no sites are shown in gray.

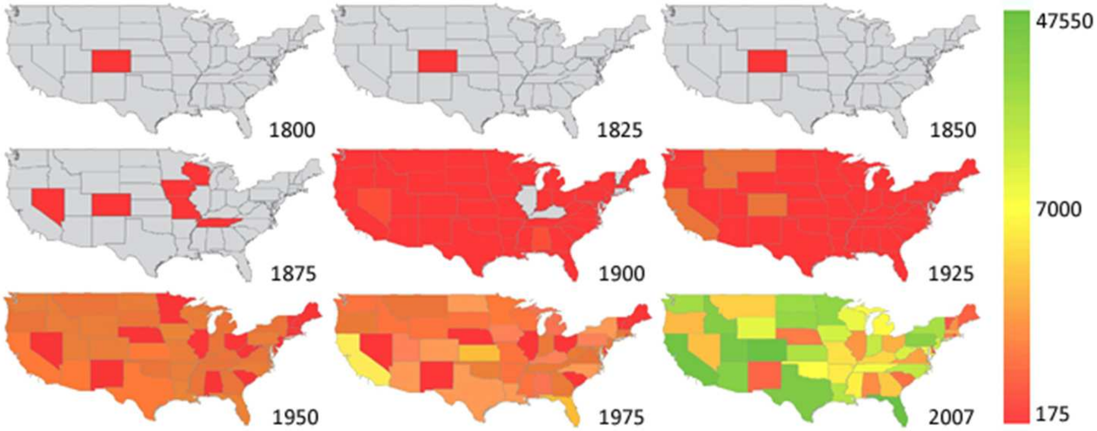


Figure 4.11. Measurement sites over time

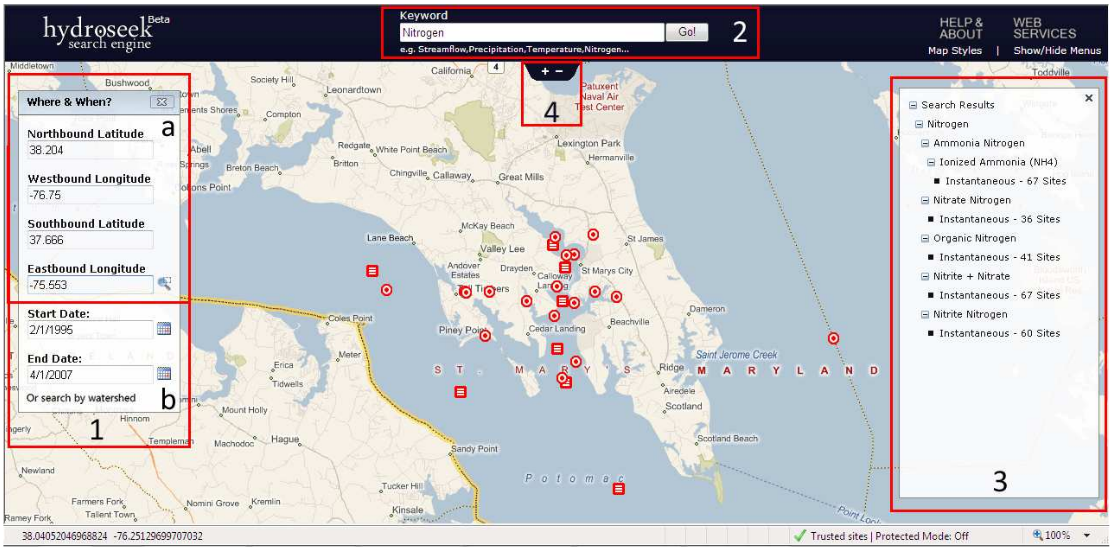
4.3. User Interface for the Search Mechanism

Hydroseek's graphical user interface was developed in ASP.NET using Asynchronous JavaScript and XML (AJAX) programming style. The system utilizes Microsoft's Virtual Earth [145] map control for the display of geographic information.

The interface is designed to make maximum use of the available screen area. Most graphical user interface (GUI) elements are floating frames which can be turned on and off as needed. The GUI provides necessary tools to identify a time frame, a geographic region and a keyword (Figure 4.12). The desired geographic region can be identified using a geographical bounding box or hydrologic unit name (e.g. watershed). Bounding box coordinates, if known, can be entered manually in the area marked 1(a) in Figure 4.12 or can be selected graphically over the map. Hydrologic unit name selection is guided with an auto-complete textbox which pulls names from a database that contains approximately 2900 watersheds in the US. Hydrologic unit codes (HUC) are generally preferred over hydrologic unit names. The advantage of using HUC codes is that they are unique. The disadvantage is that they are hard to remember. Hydrologic unit names are easy to remember (e.g. Potomac River) but not necessarily unique. This might cause confusion. In *Hydroseek* this problem is avoided by concatenating the watershed name with a location identifier such as a state name. “Delaware River, Kansas”, "Delaware River, New Mexico/Texas", "Delaware River, Pennsylvania/New Jersey" are some examples. These identifiers are to give an idea about whereabouts of the river and do not necessarily name all the states that a river goes through. Area 1(b) contains form elements related to the time frame selection.

Keyword selection is guided with an auto-complete textbox which relies on the underlying knowledge base (ontologies). For each keyword the knowledge base contains several synonyms to take the guess work out of the search experience. A keyword is entered in the area marked with 2 in Figure 4.12. Users are encouraged to use broad concepts as search terms when necessary since the results are classified by the system based on the knowledge base. For example, a keyword search such as ‘Precipitation’ would return all data on precipitation amount, precipitation duration and dry days classified under these categories and further classified based on their temporal characteristics e.g. instantaneous, daily average, cumulative, 15-minute incremental etc. Classification makes use of the cached output of “Ontology Query” and “Get Stations” functions which will be examined in the Web Services section. Based on the output of “Ontology Query” and “Get Stations” services, a new database query is written which identifies the number of stations for each branch in the tree shown in the area marked with 3. During this process the output of “Ontology Query” function also gets modified as the branches that contain no data get deleted recursively.

Stations containing the desired data are displayed on the map with pushpins of different shape indicating the repository to which they belong. Figure 4.12 shows the results of a data inquiry on Potomac River concerning ‘Nutrients’. Here circle and rectangle icons represent Chesapeake Bay Information Management System (CIMS) and EPA STORET data repositories, respectively. Once the query is complete, classification of the results appears in the form of a tree; in the area marked 3 in Figure 4.12. By clicking on the branches of this tree users can narrow down their search. Their actions reflect upon the number of stations displayed on the map in real-time.



Keyword
Nitrogen
e.g. Streamflow,Precipitation,Temperature,Nitrogen...

Where & When?

Northbound Latitude
38.204

Westbound Longitude
-76.75

Southbound Latitude
37.666

Eastbound Longitude
-75.553

Start Date:
2/1/1995

End Date:
4/1/2007

Or search by watershed

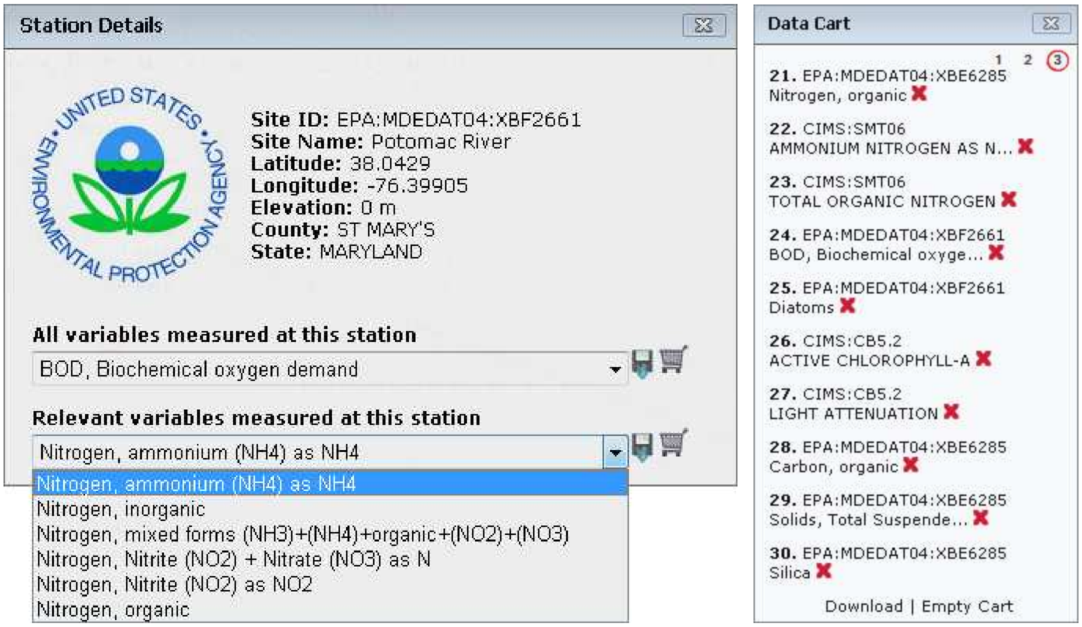
- Search Results
  - Nitrogen
    - Ammonia Nitrogen
      - Ionized Ammonia (NH4)
        - Instantaneous - 67 Sites
    - Nitrate Nitrogen
      - Instantaneous - 36 Sites
    - Organic Nitrogen
      - Instantaneous - 41 Sites
    - Nitrite + Nitrate
      - Instantaneous - 67 Sites
    - Nitrite Nitrogen
      - Instantaneous - 60 Sites

38.04052046968824 -76.25129699707032

Figure 4.12. Elements of the search interface

This kind of interface eliminates the need to know parameter codes for different data sources in order to get data from them. It does not allow queries that are too specific, thus increases the chances of relevant data discovery while by clustering the results, it tackles possible high-recall problem and facilitates accurate discovery.

Once sites are located, the user can take a closer look at individual sites and download relevant data. Station details contain metadata such as station name, state, county, elevation, latitude and longitude and two lists containing the different variables available at the selected station. The first list gives the available variables while the second list contains variables that are relevant to the keyword the user provided. Figure 4.13 shows station details popup to the left.



Station Details

Site ID: EPA:MDEDAT04:XBF2661
Site Name: Potomac River
Latitude: 38.0429
Longitude: -76.39905
Elevation: 0 m
County: ST MARY'S
State: MARYLAND

All variables measured at this station

BOD, Biochemical oxygen demand

Relevant variables measured at this station

Nitrogen, ammonium (NH4) as NH4

- Nitrogen, ammonium (NH4) as NH4
- Nitrogen, inorganic
- Nitrogen, mixed forms (NH3)+(NH4)+organic+(NO2)+(NO3)
- Nitrogen, Nitrite (NO2) + Nitrate (NO3) as N
- Nitrogen, Nitrite (NO2) as NO2
- Nitrogen, organic

Data Cart

1 2 3

21. EPA:MDEDAT04:XBE6285 Nitrogen, organic
22. CIMS:SMT06 AMMONIUM NITROGEN AS N...
23. CIMS:SMT06 TOTAL ORGANIC NITROGEN
24. EPA:MDEDAT04:XBF2661 BOD, Biochemical oxyge...
25. EPA:MDEDAT04:XBF2661 Diatoms
26. CIMS:CB5.2 ACTIVE CHLOROPHYLL-A
27. CIMS:CB5.2 LIGHT ATTENUATION
28. EPA:MDEDAT04:XBE6285 Carbon, organic
29. EPA:MDEDAT04:XBE6285 Solids, Total Suspende...
30. EPA:MDEDAT04:XBE6285 Silica

Download | Empty Cart

Figure 4.13. Station details and data cart windows

In Figure 4.13, search keyword is Nitrogen thus the second box contains a list of different nitrogen forms. Data can be downloaded directly by clicking on the diskette icon to the right of the variable list but since communication with data repositories can be lengthy at times, a data cart option is also provided. By clicking the cart icon, the selected parameter can be added to the cart. The data cart allows users to submit persistent (independent of session continuity) data requests from up to 30 stations at once and does not interfere with browsing. The data cart window is shown in Figure 4.13 to the right of station details. As a shortcut data can be directly added to the data cart by moving the mouse pointer over a site and selecting “Add to Cart”. This action adds all the relevant datasets available at that site to the cart without having to open the site details window.

Once the data is ready, the data cart sends an e-mail to the user with the URL where the data can be downloaded. Data is provided as a zipped Microsoft Excel

Workbook in a standardized format regardless of the file formats and structures the underlying repositories use. Surveys show that Excel is the most widely used software among hydrologists [3].

One limitation of virtual globes such as Google Maps or MS Virtual Earth is the number of pins they can display on the map. Since it is not uncommon to define larger bounding boxes that may entail a large watershed or perhaps an entire state the number of returned stations may be very high posing a problem for the graphic user interface to properly display them all. To solve this issue, a code was written to group nearby pins into a single pin representing a cluster depending on the zoom level. It is possible to zoom in or out using the scroll button of the mouse, plus/minus keys on the keyboard or the small panel marked with number 4 in Figure 4.12. As the user zooms in, clusters break into smaller clusters and eventually individual stations become visible. Since zooming decreases the area displayed on the screen, at high zoom levels the number of pins displayed on the map is low enough for viewing purposes. This is achieved using a static grid over the map. Grid cells have a constant size defined in pixels with static locations on the screen. Thus the number of grids on a given map is a function of screen size and resolution, not geographic extent. Every time user zooms or pans, grid boundaries are calculated in latitude and longitude and compared to those of the stations that the user’s query returned. A cell is represented by a cluster icon on the map if it contains stations. Figure 4.14 shows how the area contained in a grid cell varies with increasing zoom level from image 1 through 4. The (123) icon in image 2 represents a cluster.

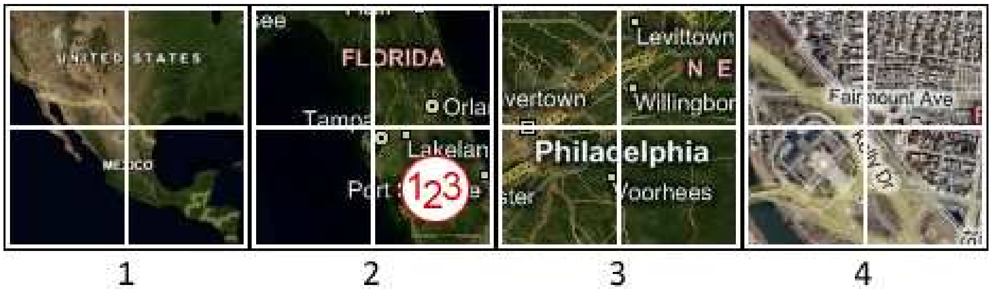


Figure 4.14. Pushpin clustering example

The search system; *Hydroseek* is operational and can be accessed from http://cbe.cae.drexel.edu/search/ , www.hydroseek.org or www.hydroseek.net.

4.4. Web services

To ensure reusability of its components, *Hydroseek* has been designed following a services-oriented architecture (SOA). Most web services were developed to interact with the ontology and the database to feed the user interface with the necessary information. However screen scrapers also constitute a crucially important group in the web services stack. Since none of the data repositories discussed here have yet implemented web-service-based access to their data, these programs can be considered as web service wrappers which mimic the actions of a human user, read the results from the screen, and return them as an XML document. While customized screen scrapers were written for each repository, their inputs and outputs are standardized. Regardless of the repository a screen scraper service always receives 4 parameters (station code, variable code, begin date, end date) and returns the data according to CUAHSI WaterML [146] specification thus providing a solution to the structural, syntactic and information system heterogeneity problem mentioned earlier. EPA STORET, CIMS, TCEQ, Burd Run and RTHNet services have been developed at Drexel University while the NWIS web service wrappers were developed and are being hosted at the San Diego Supercomputer Center (SDSC). Web service wrappers not only standardize the output format but also standardize syntax whenever possible. A total of 900 different units from NWIS and EPA STORET have been mapped to a list of 302 common units to ensure web services return values with consistent units. Surprisingly inconsistencies are not only between different agencies but within the same repository, the same quantities can be given with different units. This confounds software that uses simple string matching to interpret them. Inconsistencies may be due to the use of different symbols as in the example ‘acre feet’ vs. ‘acre-feet’, or due to typographical errors as in the example ‘micrograms per kilogram’ vs. ‘micrograms per kilgram’ or just because of choices such as ‘FTU’ vs. ‘NTU’ or ‘mho’ vs. ‘Siemens’ or ‘mg/kg’ vs. ‘ppm’.

The web services that dealt with ontologies made use of JENA [147]. JENA is a powerful JAVA application programming interface (API) for ontology processing. Since the development environment is not JAVA, it was necessary to leverage IKVM.NET [148] to use JENA in *Hydroseek*. IKVM.NET is a third-party Java Virtual Machine (JVM) allowing the use of JAVA libraries within the .NET environment.

Auto-complete form elements i.e. keyword and watershed rely on two web services for content. The keyword list is created on the fly from the ontologies (all the concepts in compound and core layers) and ordered alphabetically by the web service. During the compilation of the list, the program reads not only the labels but also the properties of classes. As shown in Figure 4.15 not all classes have the same number of properties. Here one can consider Class A as a measurement of “Cadmium” and Class B as “Wind Speed”. The medium class provides a list of permissible values of the “MeasuredIn” property e.g. Water, Soil or Sediment rather than a single value.

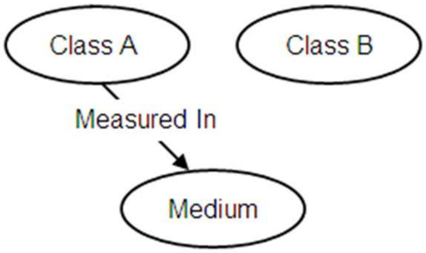


Figure 4.15. Measurement – Medium relationship

When the class “Cadmium” is read and its relationship with “Medium” discovered, keywords such as “Cadmium (Water)” and “Cadmium (Sediment)” are added to the list with “Cadmium”. The keyword “Cadmium” can be used for searching regardless of the medium. The measurement – medium relationship is provided as a shortcut to keep the ontology simple without losing the functionality to include the sampling medium in the search. It eliminates the need to create multiple entries for the same parameter. This makes ontology development easier and keeps the ontology compact. However the fact that a medium option exists does not prevent administrators (who maintain *Hydroseek*) from creating parameters such as “XXX (Air)” which embodies medium information as well.

When using auto-complete form elements, every keystroke causes a new request to be sent to the web service and the web service response is used in updating the list of keywords to display only the keywords starting with the letters typed in. The procedure is similar for watershed names however instead of an ontology the watershed list is read from a database.

The web services used in querying the knowledge base and metadata catalog offer methods such as OntologyQuery, GetStations, GetMeasurementCatalog, GetMeasurementCatalogFiltered, GetStationDetail and GetData.

4.4.1. Ontology Query Service

“Ontology Query” breaks down the keyword into relevant concepts and variables. This is done by querying the ontology to identify the relevant concepts and then querying the database to find variables that are related to these concepts. The former is a downward link traversal in concept hierarchy as depicted in Figure 4.16 to extract a subset of the ontology. Here the item at the top represents the match for user’s keyword entry.

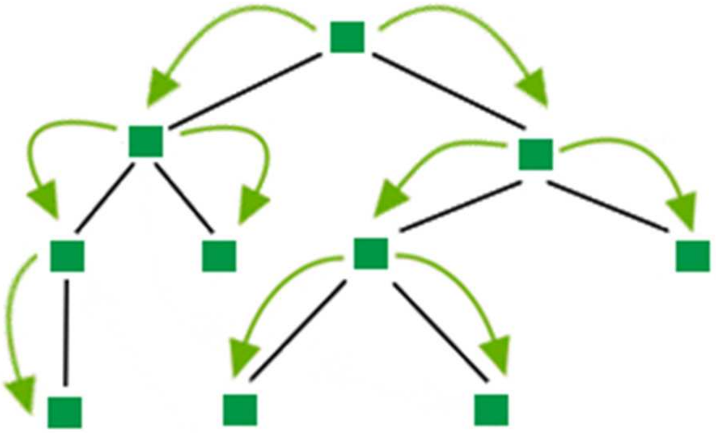


Figure 4.16. Schematic of downward link traversal operation used in ontology query function

This approach recursively finds direct subclasses of each class until there are no subclasses left in the hierarchy. An alternative to this approach relies on the transitivity of sub-superclass relationship e.g. if A is a subclass of B and B is a subclass of C, then A is a subclass of C. By exploiting this option it is possible to get a list of all the subclasses (both direct and indirect) of the class at the top of the hierarchy in Figure 4.16 at once. However, this approach requires a one-level upward link traversal to find the direct superclass of each class in order to maintain the same hierarchical structure in the output. Figure 4.17 shows this process graphically.

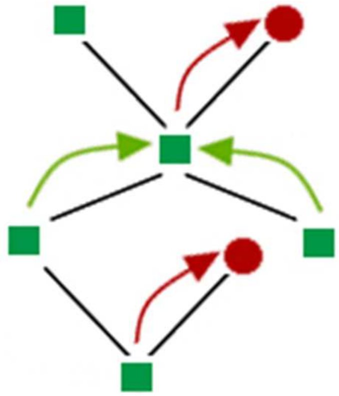


Figure 4.17. Possible problems with upward link traversal

In this figure the green box at the upper left corner is the match for provided keyword whereas the other green boxes are its subclasses. Since *Hydroseek* allows one concept to have multiple parent concepts (e.g. Zinc is a heavy metal and also a micronutrient) it is possible for the software to pick the wrong parents indicated by red circles in the figure. The output can be fixed by finding broken links and attaching them to different superclasses iteratively until the hierarchy is complete however, this makes the process unnecessarily complex and defeats the purpose of using the “transitivity shortcut”. Thus the robust first approach is preferred in *Hydroseek*.

Following this step, the program queries the database to find any chemical/physical/biological variables associated with the concepts extracted from the ontology. Figure 4.18 shows the link between the database and ontology using the graph data model [149].

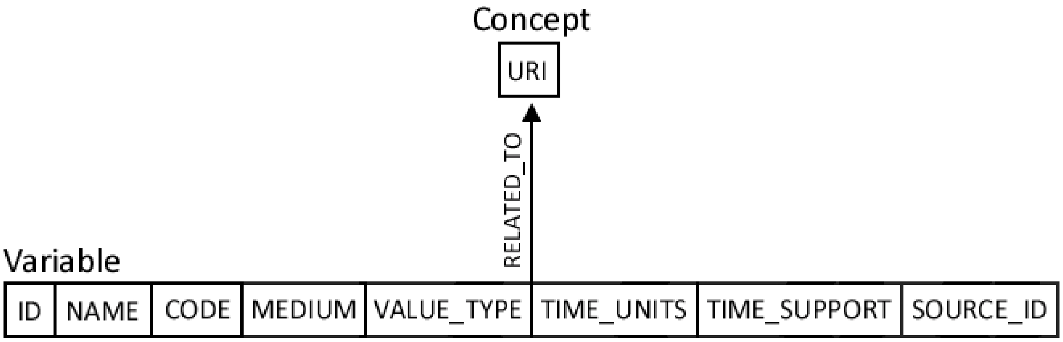


Figure 4.18. Relationship between the ontology and the database

Variables are related to concepts and linked in a database table with their unique identifiers. Matching criteria varies. The most basic involves the use of only a concept-variable relation. However, a keyword entry such as “Zinc (Sediment)” also requires the Medium criteria to be satisfied.

Matching variables are grouped according to their data types. Here data type is used to represent Value_Type, Time_Support and Time_Units attributes in Figure 4.18 all together. The collection of these groups can be denoted using the following algebraic expression.

$$G_{R_i}(X) = \{ g_{R_i}(X, x) : x \in R_i[X] \}$$

where;

$$g_{R_i}(X, x) = \{ r : r \in R_i \wedge r[X] = x \}$$

representing individual groups [150]. Here r is a record of record type R for which the value of an occurrence of X is the same as that of every occurrence within its group.

The Value_Type attribute can take values such as Instantaneous, Categorical, Average, Incremental, Minimum and Maximum. Time_Support and Time_Units are used to provide the time period over which Value_Type is applied such that Time Support + Time Units + Value Type gives for example “1 Day Average” or “15 Minute Incremental”. For Value_Types such as Instantaneous and Categorical, time support is 0. These attributes and controlled vocabularies are based on the ODM database schema. Since many time units are supported, it is possible for one variable (Figure 4.18) to have a data type “1 Week Incremental” while another “7 Day Incremental”. If grouped using plain string matching, these two would fall into different categories. To avoid this problem *Hydroseek* is given the ability to convert time units to one another. Moreover system converts “1 Day…” to “Daily…”, “1 Week…” to “Weekly…”, “1 Hour…” to “Hourly…” etc. As a result “1 Hour Average” and “60 Minute Average” both appear in the output as “Hourly Average” while “10 Minute Average” and “600 Second Average” as “10-minute Average”.

These groups are added to the hierarchy of the ontology extract from the previous step followed by the addition of matching variables. Code or VariableCode is the identifier for the measured parameter used by the original data source, e.g. USGS, which is necessary to communicate with their server to retrieve the actual data. SourceID is the unique identifier in the database for data sources which provides the domain of usability for the variable code. Some data sources such as USGS NWIS may have more than one SourceID since they operate multiple sub-networks/databases. The output of the “Ontology Query” service is an XML document like that shown in Figure 4.19. In this XML document ‘RootClass label’ shows the keyword provided by the user. Classes such as streamflowDailyAverage, streamflowInstantaneous and streamflowCategorical are products of grouping of database records as explained above.

```
<?xml version="1.0" encoding="UTF-8"?>
<Results>
<Classes>
<RootClass label="Streamflow">streamflow</RootClass>
<Class label="Daily Average" hasParent="streamflow">streamflowDailyAverage</Class>
<Class label="Instantaneous" hasParent="streamflow">streamflowInstantaneous</Class>
<Class label="Categorical" hasParent="streamflow">streamflowCategorical</Class>
</Classes>
<Variables>
<Variable hasParent="streamflowInstantaneous" sourceID="2"
variableCode="NWIS:00060" />
<Variable hasParent="streamflowCategorical" sourceID="4" variableCode="EPA:241-1"
/>
<Variable hasParent="streamflowInstantaneous" sourceID="5"
variableCode="CIMS:FLOW_INS" />
<Variable hasParent="streamflowDailyAverage” sourceID="1"
variableCode="NWIS:00060" />
<Variable hasParent="streamflowInstantaneous” sourceID="3"
variableCode="NWIS:00061" />
<Variable hasParent="streamflowInstantaneous” sourceID="3"
variableCode="NWIS:50051" />
<Variable hasParent="streamflowInstantaneous” sourceID="2"
variableCode="NWIS:30209" />
</Variables>
</Results>
```

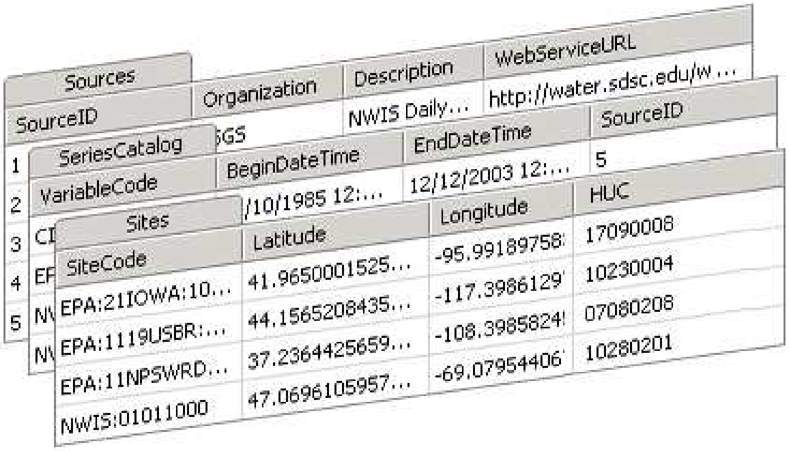
Figure 4.19. Output of ontology query function

Each class has a reader-friendly label and a unique identifier to be used with hasParent attribute for providing linkages in the hierarchy. Variables are linked to parent concepts in a similar fashion and are provided with sourceIDs and variableCodes for the purposes explained above.

This output serves as the knowledge base in the rest of the process, i.e. there is no further interaction with the keyword ontology until a new search is initiated. This minimizes the time spent processing the ontology. “Ontology Query” is directly accessible as a web service; however its output is mostly consumed by other web services rather than being an end product.

4.4.2. Get Stations Service

Using the subset of the knowledge base from the “Ontology Query” service with the geographic bounding box and time frame submitted by the user via the user interface, the Sites and SeriesCatalog tables in the database are interrogated (Figure 4.20). This figure provides only the columns essential for the rest of the process and does not include all the columns in these tables.



Sources

SourceID	Organization	Description	WebServiceURL
1	GS	NWIS Daily…	http://water.sdsc.edu/w …
2			
3			
4			
5			

SeriesCatalog

VariableCode	BeginDateTime	EndDateTime	SourceID
	/10/1985 12:…	12/12/2003 12:…	5
CI			
EF			
NV			
NV			

Sites

SiteCode	Latitude	Longitude	HUC
EPA:21IOWA:10…	41.9650001525…	-95.99189758	17090008
EPA:1119USBR:…	44.1565208435…	-117.3986129	10230004
EPA:11NPSWRD…	37.2364425659…	-108.3985824	07080208
NWIS:01011000	47.0696105957…	-69.07954406	10280201

Figure 4.20. Sources, Sites and Series Catalog tables

Record matching is done by checking if the site coordinates fall into the user supplied bounding box, whether or not there is an overlap in time frames and if the site has any relevant measurements identified by comparing VariableCode and SourceID pairs with the output of the ontology query service. A list of distinct stations that satisfy these criteria with their coordinates, unique IDs and name of the repositories to which they belong, comprise the output of the Get Stations service. Latitude and longitude are used for plotting the stations on the map, repository name (e.g. NWIS) is used in pushpin assignment while station ID provides a link to the database. The station list is stored in memory until a new query is initiated or a session expiration occurs to avoid delays on map zoom and pan events.

As mentioned in section 4.3 it is also possible to search using a watershed name instead of a geographical bounding box. Both USGS and STORET restrict the use of hydrologic unit codes to certain levels in the hierarchy. STORET allows only 8 digit codes [151], while NWIS allows only 2 digit codes [152] (hydrologic region) for querying their system. In *Hydroseek*, sites are associated with their 8 digit HUCs while the user is given the option to type in a watershed name which could have a HUC of 2 digits to 8 digits. Since hydrologic units with higher digit codes are contained within one with lower digit codes, it is possible to relate different levels in the hierarchy to each other using mathematical operators.

If X is a hydrologic region represented by a 2 digit HUC, 07 then X contains all hydrologic units that start with 07 such as 07080208. If 6 digits are added to 07, the result is 07000000. 07080208 is greater than 07000000 and is thus a part of 07. All the hydrologic units in region X have HUCs greater than 07000000 and less than 08000000, the boundary for region 08. This simple solution can be applied at any level in the HUC hierarchy. For example if a user enters a watershed name that corresponds to HUC 070801, *Hydroseek* returns stations that have HUCs greater than 07080100 and less than 07080200.

4.4.3. Get Measurement Catalog and Station Details Services

The “Station Details” service is used to provide information such as Site Name, Elevation, County and State for the pop up window shown in Figure 4.14. Information about a given site is located using the site’s unique identifier which the “Get Stations” service passes on to map interface. “Get Measurement Catalog” has two methods. The first provides the first drop down menu in Figure 4.14 with variable names, variable codes and source IDs for all the entries at the selected station. “Get Measurement Catalog Filtered”, however, uses the “Ontology Query” service’s output to identify the parameters related to the search keyword to be used to populate the second drop down menu. The same function is used for the “Add Site to Cart” function, as well.

4.4.4. Get Data Service

The “Get Data” service receives data retrieval requests with parameters, SiteCode, VariableCode, BeginDateTime, EndDateTime and SourceID. It identifies the web service URLs from Sources table using the sourceID (Figure 4.20) and passes on the request to the services which were referred to earlier as screen scrapers or wrappers. However, these services do not have to be screen scrapers for the “Get Data Service” to function properly. For example, Drexel maintains copies of Burd Run and RTHNet databases, thus the services read directly from the database while NWIS services are not only screen scrapers but also reside on a different server at the San Diego Supercomputer Center. Since inputs and outputs are consistent, how the web service operates and where it resides have no effect on compatibility. Once the “Get Data” service receives a response, it reads the XML, writes it into an Microsoft Excel file, zips it and returns the URL where the data can be downloaded.

4.4.5. Data Cart Service

The “Data Cart” service uses Microsoft Message Queuing (MSMQ) technology to allow for persistent data retrieval requests. Requests are added to a message queue which is periodically checked by a windows service developed specifically for this task on the server. A Windows service is an application that runs in the background and starts automatically when the operating system is booted. Once the message is read it is removed from the queue and processed. Processing is similar to the “Get Data” service but since the “Data Cart” handles multiple requests, each response is written in a single Excel workbook but in separate worksheets. Considering the system’s distributed structure, data retrieval which relies on third parties may not always be successful. In such cases worksheets contain an error message rather than failing the entire process, thus allowing users to retrieve at least those portions of the data that is available.

4.4.6. GCMD Services

Even though it is not necessary for the system’s operation, querying with Global Change Master Directory (GCMD) keywords is also supported via web services to promote semantic interoperability with other communities. GCMD web services also support some additional options for formulating requests and output that is returned. GCMD services contain a variant of “Get Stations” service that can be used with GCMD keywords which returns results in Geography Markup Language (GML). For this function, an application profile of GML Simple Feature Profile was developed. The XML Schema for the profile with example web service output is provided in Appendix II.

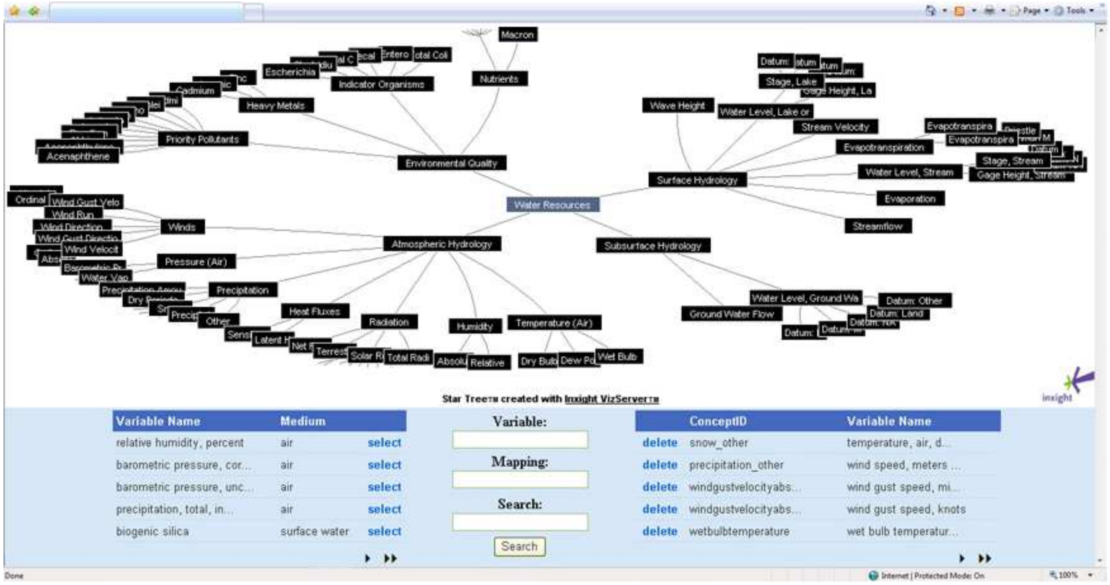
The GCMD keyword list is exhaustive. It covers a large domain from seismology to solar winds. *Hydroseek* only covers keywords that are relevant to hydrology and environmental science. Thus a service was developed specifically to provide a list of supported GCMD keywords.

The “Get Data” services require variableCodes and sourceIDs to be entered to operate. One can use screen scraper services directly which eliminates the need for sourceIDs. However, this increases the number of services one needs to use. Currently “Get Data” service channels the requests to a total of 8 different services. While this isn’t a problem when everything happens behind the scenes, it wouldn’t be correct to expect from a GCMD service developed specifically for providing a common ground to have these requirements. The assumption must be that the user only knows a GCMD keyword in addition to when and where to look for data. For this reason a new service was developed. This service expects only a GCMD keyword, a site ID and a time frame. It finds all the relevant measurements for a given site and then identifies which services to call using the catalog. After requesting data from all necessary services, it combines the responses and returns them to the user as an XML document.

CHAPTER 5: MAPPING TOOL

Hydroseek provides a unified view over heterogeneous data sources. While this unified view involves a common syntax and file format, probably the most important portion is semantics. The search system implements the ODM database schema developed at Utah State University for CUAHSI Hydrologic Information Systems project. *Hydroseek* extends this database with one additional table which contains the links between the database and the ontology (Figure 4.19). The ODM [143] is not limited to a schema and also provides tools for data browsing, loading and content management (such as ODM Tools, ODM Data Loader, and ODM Streaming Data Loader). However since the additional table is not in the original database, ODM tools or data loaders don't perform any operations on it. As a result mappings need to be done by directly accessing the database and using an ontology editor such as Protégé [108] to locate the relevant concepts which requires expertise. To simplify the process to a level that anybody without need to have any knowledge of database management systems or ontologies can use, a mapping tool was developed which brings the database view and ontology visualization together in a web based interface. Using this mapping tool one can associate a variable name with a concept in the ontology with a few mouse clicks. Figure 5.1 shows a screen shot of this mapping tool which uses the Inxight 2D Hyperbolic Tree software for the visualization of the ontology. It should be noted that this interface is designed to simplify the work of administrators who want to extend the coverage of *Hydroseek* by adding more variables, not for the users who interact with *Hydroseek* to discover and retrieve hydrologic data. Mapping system also allows extension of the knowledge base. For example if a user wants to associate the heavy metal "Copper" with a measurement while Copper is not provided as an option he/she could simply choose

"Other" under Heavy Metals category and suggest a new category "Copper". These suggestions are stored in the system and are approved by a higher-level administrator. Until addition is approved results appear using the hierarchy "Heavy Metal > Other". Admin could associate this measurement with an already existing entry in the ontology (if there's already a match that the person who suggested the addition didn't see), or add it to the ontology after modifying the suggested name (e.g. replace Copper with Cu) or as is. Mapping application adds this new category to the system which appears in the search results as well as a new option in the mapping interface. The functionality is not meant to provide an alternative to a full-fledged ontology editor however it simplifies handling of this commonly observed ontology update task.



Variable Name	Medium	
relative humidity, percent	air	select
barometric pressure, cor...	air	select
barometric pressure, unc...	air	select
precipitation, total, in...	air	select
biogenic silica	surface water	select

Variable:

Mapping:

Search:

Search

	ConceptID	Variable Name
delete	snow_other	temperature, air, d...
delete	precipitation_other	wind speed, meters ...
delete	windgustvelocityabs...	wind gust speed, mi...
delete	windgustvelocityabs...	wind gust speed, knots
delete	wetbulbtemperature	wet bulb temperatur...

Figure 5.1. Mapping tool

CHAPTER 6: EVALUATION

In order to assess the performance of *Hydroseek* on a quantifiable basis some statistical tests were performed. 12 users were provided with a geographic area, a timeframe and a science question that required them to query NWIS, STORET and CIMS repositories. They were asked to investigate a hypothetical fish kill case on the basis that unionized ammonia, nitrite and nitrate are toxic to fish [153,154]. Users were asked to record the time they spent to get the data. Among the first time users (6 users) average time using *Hydroseek* was 57.7 seconds. The same exercise took an average of 552.67 seconds using the original data sources. For experienced users results were 36.167 and 255.5 seconds respectively for *Hydroseek* and the data repositories. We evaluated the difference in averages using a paired two-tailed student's t-test. Results provided in Table 6.1 indicate that difference is confirmed also by statistical analysis.

Table 6.1 Results of statistical analysis

	t	**$t_{critical}$**	**P(T<=t)**	**alpha**	**Degree of freedom**
First time user	25.2823	2.5706	1.807E-06	0.05	5
Expert user	96.3902	2.5706	2.278E-09	0.05	5

During this analysis some users stated that they were able to find more data with *Hydroseek* than with the original repositories, however this issue was not further analyzed and quantified.

CHAPTER 7: CONCLUDING REMARKS

This dissertation addressed several issues regarding hydrologic data discovery and retrieval by:

1. Providing a unified view over multiple heterogeneous data repositories
2. Simplifying data discovery using keywords and commonly used terms eliminating the need to know source specific parameter codes,
3. Applying a layered approach that reduces problems such as 'low or no search results' commonly observed with the available search tools even when relevant data is available,
4. Making it easier to deal with a high number of results using a hierarchical classification system based on a knowledge base,
5. Providing a simple, functional interface design able to provide access to a large data inventory without overwhelming the user
6. Providing web-service-based access to allow embedding these functionalities into third party applications
7. Supporting established controlled vocabularies such as GCMD and output formats such as GML to promote interoperability with other communities and thus bringing hydrologic data sources closer to other related science communities such as atmospheric and marine science.

After examination of data sources in the US and under the light of surveys among hydrologists, data repositories of primary interest were identified as USGS NWIS and EPA STORET; the two largest repositories in the area of interest. Metadata catalogs were harvested over the web using specifically designed programs for each repository. These catalogs and programs can be considered analogous to search engine indexes and bots. Examination of these systems in addition to the knowledge of several failed attempts to create unified views over these two systems over the past few years by NWIS and EPA gives a good idea about the extent of the problem. Later metadata catalogs were extended by including TCEQ TRACS, CIMS, Burd Run and RthNET databases. These six datasets span very large to very small networks showing extensibility and the ability of the system to work at all scales. To solve the heterogeneity problem between repositories and also to provide a mean to better manage data, a knowledge base was developed using Web Ontology Language (OWL). OWL is recommended by World Wide Web Consortium. It is a powerful way of storing knowledge using web standards. The nature of the data also required a map interface. Microsoft Virtual Earth was used for this purpose. The interface was developed using AJAX-style programming with rich interactive menus. Components were developed as web services to allow for use of the functionalities by different software products as well as different web interfaces. System is designed to be easily extensible with the additional mapping tool. Hard-coding was avoided, thus updates require additions to either the database or the ontology. *Hydroseek* website is visited 350-900 times every day by 40-100 unique users, each visit taking 20 minutes on average.

Although this dissertation provides a solid framework for dealing with heterogeneous data sources, complete integration of the hydrologic data realm requires further efforts. Some of these efforts can focus on extension of coverage by adding more data sources and the addition of support for gridded data types. Additional languages can be supported with small additions to the ontologies. However, a weakness of the system is screen scraper services that need to be addressed in the shorter term. Since screen scrapers consume the output of a webpage, when the web design changes to improve functionality or user-friendliness of the webpage, these services tend to fail. CUAHSI has been communicating with some of the data repositories and it is expected to have web services implemented at the original data sources to replace the screen scrapers. This will significantly improve the reliability and performance of these web services. It is important that the output of web services be compliant with OGC standards to have more acceptability and usability across communities. The GML Application Schema developed for GCMD services in this study could serve as an example in that direction. Finally, the metadata supplied by web services is fairly limited. This can be improved by embedding more metadata into the data envelope or having a separate service for downloading metadata. It is desirable that metadata also follow an international standard such as ISO 19139 (XML Schema implementation of ISO 19115). An example with extended ISO 19139 metadata is developed and provided with this study for this purpose.

LIST OF REFERENCES

[1] The United States Department of Interior, Budget Justifications and Annual Performance Plan Fiscal Year 2002; Water Data Collection and Management Subactivity.
http://www.usgs.gov/budget/2002_Justification/07waterinfo.html

[2] The United States Department of Interior, Budget Justifications and Performance Information Fiscal Year 2006; USGS. pp J-76.
http://www.usgs.gov/budget/2006/usgs_fy06_bdgt_just_02-25-05.pdf

[3] Maidment D. CUAHSI HIS Status Report. September 15, 2005; pp 48-87.
http://www.cuahsi.org/docs/HISStatusSept15.pdf

[4] Stations registered in EPA STORET. http://www.epa.gov/storet/coverage.html

[5] Window to My Environment. http://www.epa.gov/enviro/wme/background.html

[6] NWISweb. http://waterdata.usgs.gov/nwis

[7] Mathey S. B. (editor). US Geological Survey Manual (432-1-S2) 2006; USGS, Reston, VA. http://pubs.usgs.gov/fs/FS-027-98/fs-027-98.pdf

[8] EPA STORET. http://www.epa.gov/storet/

[9] Environmental Information Exchange Network. http://exchangenetwork.net/

[10] Central Data Exchange. http://www.epa.gov/cdx/

[11] Environmental Sampling, Analysis and Results Data Standard. EDSC, January 2006;
http://www.envdatastandards.net/files/693_file_ESAR_Overview_01_06_2006__Final_.pdf

[12] Water Quality Data Elements: A User Guide, NWQMC Technical Report No 3. April, 2006; http://acwi.gov/methods/pubs/wdqe_pubs/wqde_trno3.pdf

[13] National Climatic Data Center. http://www.ncdc.noaa.gov/oa/ncdc.html

[14] National Operational Model Archive and Distribution System. http://nomads.ncdc.noaa.gov/

[15] NOAAServer. http://www.esdim.noaa.gov/cgi-bin/NOAAServer

[16] Ansari S., Del Greco S. A., Frederick H., Nelson B. R. The Severe Weather Inventory (NSWI): spatial query tools, web services and data portals at NOAA's National Climatic Data Center. 22nd International Conference on Interactive Information Processing Systems for Meteorology, Oceanography, and Hydrology, January 28-February 3, 2006; American Meteorological Society, Atlanta, GA, United States.

[17] USGS National Elevation Dataset. http://ned.usgs.gov/

[18] USGS Seamless Data Distribution System. http://seamless.usgs.gov/

[19] EROS USGS Geographic Data Download. http://eros.usgs.gov/geodata/

[20] USGS National Hydrography Dataset. http://nhd.usgs.gov/

[21] NHD Viewer. http://nhdgeo.usgs.gov/

[22] National Hydrography Dataset FTP Data Access. ftp://nhdftp.usgs.gov/SubRegions

[23] Dewald T. Putting NHDPlus to Work. ESRI International User Conference, August 7-11, 2006; San Diego, California, United States.

[24] Simley, J. NHD Stakeholders Meet to Discuss Future of the NHD. USGS National Hydrography Dataset Newsletter, October 2006; 5(12).

[25] Elevation Derivatives for National Applications http://edna.usgs.gov/

[26] Kost J. R., Kelly G. G. Watershed Delineation Using the National Elevation Dataset and Semiautomated Techniques. ESRI International User Conference, July 9 – 13, 2001; San Diego, California, United States.

[27] EDNA Viewer. http://gisdata.usgs.net/website/EDNA/viewer.php

[28] Multi-Resolution Land Characteristics Consortium. http://www.mrlc.gov/

[29] NLCD Production Status Maps. http://www.mrlc.gov/mrlc2k_nlcd_map.asp

[30] Soil Survey Geographic (SSURGO) Database. http://www.ncgc.nrcs.usda.gov/products/datasets/ssurgo/description.html

[31] STATSGO Data User Information, Publication Number 1492, USDA NRCS 1995.

[32] Federal Standards for Delineation of Hydrologic Unit Boundaries. FGDC, 2004; ftp://ftp-fc.sc.egov.usda.gov/NCGC/products/watershed/hu-standards.pdf

[33] Watershed Boundary Dataset. http://www.ncgc.nrcs.usda.gov/products/datasets/watershed/

[34] Laitta M. T., Legleiter K. J., Hanson J. M. The National Watershed Boundary Dataset, Hydro Line Newsletter, Environmental Systems Research Institute (ESRI) 2004; Summer.

[35] PRISM at NRCS. http://www.ncgc.nrcs.usda.gov/products/datasets/climate/docs/fact-sheet.html

[36] PRISM Group. http://www.ocs.oregonstate.edu/prism/index.phtml

[37] SNOTEL. http://www.wcc.nrcs.usda.gov/snotel/

[38] NOAA Coastal Change Analysis Program. http://www.csc.noaa.gov/crs/lca/ccap.html

[39] C-CAP Data Map Server. http://www.csc.noaa.gov/crs/lca/locate.html

[40] Comprehensive Large Array-data Stewardship System. http://www.class.noaa.gov/nsaa/products/welcome

[41] NOAA Office of Satellite Operations. http://www.oso.noaa.gov/

[42] LANDSAT. http://landsat.gsfc.nasa.gov/

[43] MODIS. http://daac.gsfc.nasa.gov/MODIS/

[44] ASTER. http://asterweb.jpl.nasa.gov/

[45] ASTER Observation Schedule. http://asterweb.jpl.nasa.gov/gettingdata/calendar.asp

[46] LP DAAC. http://edcdaac.usgs.gov/main.asp

[47] Boken, V. K. ,Easson, G. L. Modeling Groundwater Depth in the Mississippi Delta Using Weather and ASTER Satellite Data. Eos. Trans. AGU, 2005; 86(18).

[48] Mesinger F., DiMego G., Kalnay E., Mitchell K., Sharfran P. C., Ebisuzaki W., Jovic D., Woollen J., Rogers E., Berbery E. H., Ek M. B., Fan Y., Grumbine R., Higgins W., Li H., Lin Y., Manikin G., Parrish D., Shi W. North American Regional Reanalysis. American Meteorology Society Annual Meeting, January 9-13 2005; San Diego, California, United States.

[49] Ebisuzaki W., Alpert J., Wang J., Jovic D., Shafran P. (North American Regional Reanalysis: End User Access to Large Datasets. American Meteorology Society Annual Meeting, January 11-15 2004; Seattle, Washington, United States.

[50] DAYMET. http://www.daymet.org/

[51] The University of Montana EOS Training Center Natural Resource Project. http://eostc.umt.edu/Forestry/

[52] Ameriflux Network. http://public.ornl.gov/ameriflux/

[53] Ameriflux data at CDIAC. ftp://cdiac.ornl.gov/pub/ameriflux/data/

[54] Library of Congress. http://www.loc.gov/ead/ag/agappf.html

[55] Goldfarb, C. F. The SGML Handbook. Oxford University Press 1990; Oxford, UK.

[56] Berners-Lee, T., Cailliau, R., Luotonen, A., Nielsen, H., F., Secret, A. The World-Wide Web. Communications of the ACM 1994; 37(8):76-82.

[57] http://www.w3.org/TR/WD-xml-970807.html

[58] Achard, F., Vaysseix, G., Barillot, E. XML, bioinformatics and data integration. Bioinformatics Review 2001; 17(2):115-125.

[59] XML Schema. http://www.w3.org/XML/Schema

[60] RELAX NG Specification. http://www.oasis-open.org/committees/relax-ng/spec-20011203.html

[61] RDF Specification. http://www.w3.org/TR/rdf-primer/

[62] IETF RFC1738 IETF (Internet Engineering Task Force). RFC 1738: Uniform Resource Locators (URL), ed. T. Berners-Lee, L. Masinter, and M. McCahill. 1994.

[63] IETF RFC1808 IETF (Internet Engineering Task Force). RFC 1808: Relative Uniform Resource Locators, ed. R. Fielding, 1995.

[64] RDF Schema Specification http://www.w3.org/TR/rdf-schema/

[65] Carnap, R. Empiricism, Semantics, and Ontology. Revue Internationale de Philosophie 1950 ; 4:20-40. Reprinted in In R. Rorty, ed., The Linguistic Turn. Chicago: University of Chicago Press.

[66] Gruber. T.R. Toward the principles for the Design of Ontologies Used for Knowledge Sharing. In International Journal of Human and Computer Studies 1995; 43(5/6), 907-928.

[67] Lenat D. and Guha R. Building Large Knowledge Based Systems. Reading, Massachusets, 1990.

[68] Pease, A., and Niles, I. IEEE Standard Upper Ontology: A Progress Report. Knowledge Engineering Review, Special Issue on Ontologies and Agents, 2002; 17, 65-70.

[69] Wordnet in RDFS and OWL.
http://www.w3.org/2001/sw/BestPractices/WNET/wordnet-sw-20040713.html

[70] DOLCE. http://www.loa-cnr.it/DOLCE.html

[71] BFO. http://www.ifomis.uni-saarland.de/bfo

[72] Baclawski, K., Kokar, M. G., Kogut, P. A., Hart, L., Smith, J., Letkowski, J., Emery, P. Extending the Unified Modeling Language for ontology development. Software System Model, 2002; 1:142-156.

[73] J.R.G. Pulido, M.A.G. Ruiz, R. Herrera, E. Cabello, S. Legrand and D. Elliman, Ontology languages for the semantic web: A never completely updated review, Knowledge-Based Systems, Volume 19, Issue 7, Creative Systems, November 2006; pp 489-497.

[74] Delugach, H. (editor) Information technology — Common Logic (CL): a framework for a family of logic based languages ISO/IEC FDIS 24707:2007.

[75] Davis, E., Knowledge Representation, International Encyclopedia of the Social & Behavioral Sciences. 2004; pp 8132-8139.

[76] Turner R. Logics for Artificial Intelligence. Wiley,1984; New York.

[77] Zadeh L. Commonsense and fuzzy logic. In: Cercone N, McCalla G (eds.) The Knowledge Frontier: Essays in the Representation of Knowledge. Springer-Verlag, 1987; New York, pp. 103–36.

[78] Bacchus F. Representing and Reasoning with Probabilistic Knowledge: A Logical Approach to Probabilities. MIT Press, 1990; Cambridge, MA.

[79] Minsky, M. A Framework for Representing Knowledge. MIT AI Lab Memo AIM-306 June 1974.

[80] Minsky, M. A Framework for Representing Knowledge. The Psychology of Computer Vision, 1975; NY:McGraw-Hill.

[81] Kifer, M., Lausen, G., et al., Wu, J. Logical foundations of object-oriented and frame-based languages, Journal of the Association for Computing Machinery, 1995; 42 (4):741–843.

[82] Woods, W. A. What's in a Link: Foundations for Semantic Networks. Representation and Understanding: Studies in Cognitive Science, 1975; pp 35-82, Academic Press, New York.

[83] Quillian, M. R. Word Concepts: A Theory and Simulation of some Basic Semantic Capabilities. Behavioral Science, 1967; 12:410-430.

[84] Hayes, P. J. The Logic of Frames, Frame Conceptions and Text Understanding, 1979; 46-61, Walter de Gruyter and Co., Berlin.

[85] Farquhar, A., Fikes, R., Rice, J. The Ontolingua Server: a Tool for Collaborative Ontology Construction Intl. Journal of Human-Computer Studies 1997.

[86] Shapiro, S. C. (ed). Encyclopedia of Artificial Intelligence. 2nd Edition, John Wiley & Sons, New York, 1992.

[87] Kauffman, L., H. The Mathematics of Charles Sanders Peirce. Cybernetics & Human Knowing, 2001; 8(1–2) 79–110.

[88] Kamp, H. Events, discourse representations, and temporal references. Languages,1981; (64):39-64.

[89] Sowa, J. F. A conceptual schema for knowledge based systems. Proceedings of the Workshop on Data Abstraction, Databases, and Conceptual Modeling, SIGMOD Record, ACM, 1981; 11(2):193-195.

[90] Tesnière, L. Éléments de Syntaxe Structurale. 2nd edition, Librairie C. Klincksieck, Paris, 1965.

[91] Schank, R., C., Abelson R. P. Scripts, Plans, Goals and Understanding. Lawrence Erlbaum Associates, 1977; Hillsdale, NJ.

[92] Schank, R., C. Dynamic Memory, Cambridge University Press,1982; New York.

[93] Brachman, R. J., Schmolze, J., G. An Overview of the KL-ONE Knowledge Representation System. Cognitive Science, 1985; 9(2):171-216.

[94] Lassila, O., McGuinness, D., L. The Role of Frame-Based Representation on the Semantic Web. KSL Tech Report Number KSL-01-02, 2001.

[95] Borgida, A. On the Relative Expressiveness of Description Logics and Predicate Logics", Artificial Intelligence, 1996; 82(1-2):353-367.

[96] Garey, M., Johnson, D. Computers and Intractability; A Guide to the Theory of NP-completeness, W. H. Freeman, 1979; New York, NY.

[97] Baader, F., McGuinness, D., Nardi, D., and Patel-Schneider, P., Eds. The Description Logic Handbook: Theory, Implementation, and Applications, 1st ed. Cambridge University Press, Cambridge, Great Britain, January 2003.

[98] Calvanese, D., de Giacomo, G., Nardi, D., and Lenzerini, M. Reasoning in expressive Description Logics. Handbook of Automated Reasoning, A. Robinson and A. Voronkov, Eds., vol. II. Elsevier Science Publishers, 2001; ch. 23, pp. 1582–1634.

[99] Chaudhri, V.K., Farquhar, A., Fikes , R., Karp, P.D., Rice, J. OKBC: A programmatic foundation for knowledge base interoperability. Proceedings of the 15th National Conference on Artificial Intelligence (AAAI'98), Madison, Wisconsin, USA, July 26–30, 1998, pp. 600–607.

[100] Karp, P.D., Chaudhri, V.K., Thomere, J. XOL: An XML-Based Ontology Exchange Language, Technical Report SRI AI Technical Note 559. SRI International, Menlo Park, CA, USA, 1999.

[101] Fensel D., Van Harmelen F., Horrocks I., McGuinness D.L., Patel-Schneider P.F. OIL: an ontology infrastructure for the Semantic Web, IEEE Intelligent Systems, 2001; 16(2):38–45.

[102] Horrocks I., Patel-Schneider P.F., Van Harmelen F. From SHIQ and RDF to OWL: the making of a Web Ontology Language Journal of Web Semantics, Web Semantics: Science, Services and Agents on the World Wide Web, 2003; (1):7–26.

[103] Horrocks I., Sattler U., Tobies S., Practical reasoning for expressive Description Logics, in: H. Ganzinger, D. McAllester, A. Voronkov (Eds.), Proceedings of the Sixth International Conference on Logic for Programming and Automated Reasoning (LPAR'99), Lecture Notes in Artificial Intelligence, vol. 1705, Springer, Berlin, 1999, pp. 161–180.

[104] Hendler J., McGuinness D.L. The DARPA Agent Markup Language, IEEE Intelligent Systems, 2000; 15(6):67–73.

[105] Web Ontology Language (OWL) Reference. http://www.w3.org/TR/owl-ref/

[106] Hayes, P. Why must the web be monotonic? Technical report, 2001; IHMC.

[107] OWL 1.1 Draft Specification. http://webont.org/owl/1.1/overview.html

[108] Protégé Ontology Editor. http://protege.stanford.edu/

[109] Altova Semantic Works.
http://www.altova.com/products/semanticworks/semantic_web_rdf_owl_editor.html

[110] SWOOP. http://www.mindswap.org/2004/SWOOP/

[111] W3C Web Services Activity Group. http://www.w3.org/2002/ws/

[112] Simple Object Access Protocol (SOAP). http://www.w3.org/TR/2000/NOTE-SOAP-20000508/

[113] Fielding, R. T., Architectural Styles and the Design of Network-based Software Architectures, 2000; PhD thesis, UC Irvine.

[114] Web Services Interoperability Organization. http://www.ws-i.org/

[115] Garrett, J. AJAX: A new approach to web applications. Adaptive Path, 2005; http://www.adaptivepath.com/publications/essays/archives/000385.php

[116] Crane, D., Pascarello E., James, D. Ajax in Action. Manning Publications Co. 2005.

[117] Paulson, L.D Building rich web applications with Ajax, Computer, 2005; 38(10):14–17.

[118] Mesbah, A.,Van Deursen A. An Architectural Style for Ajax, Technical Report TUD-SERG-2006-016, Faculty of Electrical Engineering, Mathematics and Computer Science, 2006; Delft University of Technology.

[119] XPath Specification. http://www.w3.org/TR/xpath

[120] Widyantoro D. H., Yen J. Using Fuzzy Ontology for Query Refinement in a Personalized Abstract Search Engine. Joint 9th IFSA World Congress and 20th NAFIPS International Conference, July 25-28, 2001, Vancouver, Canada.

[121] Widyantoro D. H., Yen J. A Fuzzy Ontology-based Abstract Search Engine and Its User Studies. IEEE International Fuzzy Systems Conference, December 2-5, 2001, Melbourne, Australia.

[122] Dey L., Singh S., Rai R., Gupta S. Ontology Aided Query Expansion for Retrieving Relevant Texts. Advances in Web Intelligence, 2005; Springer Berlin / Heidelberg, pp. 16-132.

[123] Zhang J., Peng Z., Wang S.,Nie H., Si-SEEKER: Ontology-Based Semantic Search over Databases. International Conference on Knowledge Science, Engineering and Management, August 5-8, 2006, Guilin, China.

[124] Kim H. H. ONTOWEB: Implementing an Ontology-Based Web Retrieval System", Journal of the American Society for Information Science and Technology, 2005; 56(11):1167–1176.

[125] Suomela S., Kekäläinen J. Ontology as a Search-Tool: A Study of Real Users' Query Formulation With and Without Conceptual Support. 27th European Conference on IR Research, ECIR 2005, Santiago de Compostela, Spain, March 21-23, 2005, pp. 315 – 329.

[126] Sure, Y., & Iosif, V. First results of a Semantic Web Technologies Evaluation. Proceedings of the Common Industry Program Held in Conjunction With Confederated International Conferences: On the Move to Meaningful Internet Systems (CoopIS,DOA, and ODBASE 2002) (pp. 69–78). Irvine, CA, United States.

[127] Bussey K. J., Kane D, Sunshine M, Narasimhan S., Nishizuka S., Reinhold W., Zeeberg B, Weinstein A., Weinstein J. N. MatchMiner: a tool for batch navigation among gene and gene product identifiers. Genome Biology, 2003; 4(4) R27.

[128] Marquet G., Golbreich C., Burgun A. From an ontology-based search engine towards a more flexible integration for medical and biological information. Proceedings of the Semantic Integration Workshop, October 20, 2003; Sanibel Island, Florida, United States.

[129] Zhu B., Leroy G., Chen H., Chen Y. MedTextus: An Intelligent Web-Based Medical Meta-Search System. Joint Conference on Digital Libraries July 14-18, 2002; Portland, Oregon, United States.

[130] Raskin, R.G., Pan, M. J. Knowledge representation in the semantic web for Earth and environmental terminology (SWEET), Computers & Geosciences, 2005; 31(9):1119-11125.

[131] Lin K., Bertram L. GEON: Ontology-Enabled Map Integration. 24th Annual ESRI International User Conference, August 9-13, 2004; San Diego, California, United States.

[132] Kerschberg L., Chowdhury M., Damiano A., Jeong H., Mitchell, S., Jingwei S., Smith, S. Knowledge Sifter: agent-based ontology-driven search over heterogeneous databases using semantic Web services. First International Conference on Semantics of a Networked World, June 17-19, 2004; Paris, France.

[133] Ramachandran R., Movva, S., Graves S., Tanner S. Ontology-based Semantic Search Tool for Atmospheric Science. 22nd International Conference on Interactive Information Processing Systems for Meteorology, Oceanography, and Hydrology, January 28-February 3, 2006; American Meteorological Society, Atlanta, GA, United States.

[134] Bermudez L. ONTOMET: Ontology Metadata Framework, PhD Thesis, 2004; Drexel University.

[135] Zhang K., Tang J., Hong M. C., Li J. Z., Wei W. Weighted Ontology-based Search Exploiting Semantic Similarity. Frontiers of WWW Research and Development-AP Web 2006; 8th Asia-Pacific Web Conference, Harbin, China.

[136] Olsen L. Controlled Vocabularies Boost International Participation and Normalization of Searches. Proceedings of 7th International Conference on HydroScience and Engineering (ICHE 2006), September 10-13, 2006, Philadelphia, Pennsylvania, United States.

[137] Sheth, A.P. and Larsen, J. Federated Database Systems for Managing Distributed, Heterogeneous and Autonomous Databases, *ACM Computing Surveys: Special Issue on Heterogeneous Databases* 1990; 22(3):183-236.

[138] Cui, Z., Jones D., O'Brien, P. Issues in Ontology-based Information Integration. Proceedings of IJCAI01, August 5, 2001; Seattle, USA.

[139] Ouskel, A. M. and Sheth A. Semantic Interoperability in Global Information Systems. A Brief Introduction to the Research Area and the Special Section. SIGMOD Record, 1999; 28(1):5-12.

[140]http://waterdata.usgs.gov/nwis/dv?referred_module=sw&search_criteria=state_cd&search_criteria=station_type_cd&submitted_form=introduction

[141] N. F. Noy and D. L. McGuinness. Ontology Development 101: A Guide to Creating Your First Ontology. Stanford Knowledge Systems Laboratory Technical Report KSL-01-05 and Stanford Medical Informatics Technical Report SMI-2001-0880, March 2001.

[142] Simple Knowledge Organization System (SKOS). http://www.w3.org/2004/02/skos/

[143] CUAHSI Observations Data Model (ODM). http://water.usu.edu/cuahsi/odm/files/ODM1.pdf

[144] Chaudhuri, S. and Dayal, U. An overview of data warehousing and OLAP technology. SIGMOD Record, 1997; 26(1):65-74.

[145] Microsoft Virtual Earth. http://www.microsoft.com/virtualearth/

[146] CUAHSI WaterML. http://water.sdsc.edu/WaterOneFlow/documentation/schema/cuahsiTimeSeries.xsd

[147] HP Labs. JENA API, 2000; http://www.hpl.hp.com/personal/bwm/rdf/jena/

[148] Frijters J. IKVM, 2004; http://www.ikvm.net/

[149] Kunii, H. S. DBMS with graph data model for knowledge handling. Proceedings of the 1987 Fall Joint Computer Conference on Exploring technology: today and tomorrow, pp: 138 – 142 IEEE Computer Society Press Los Alamitos, CA, USA.

[150] Kunii, H. S. Graph data model and its data language, Springer-Verlag, 1990; New York, Inc., New York, NY.

[151] EPA Surf Your Watershed. http://iaspub.epa.gov/WatershedSummaryUI/watershed.do

[152] NWIS Site Inventory. http://waterdata.usgs.gov/nwis/inventory

[153] Tilak K.S., Lakshmi S. J., Susan T., A. The toxicity of ammonia, nitrite and nitrate to the fish, Catla catla (Hamilton). Journal of Environmental Biology 2002; 23(2):147-149.

[154] Fuller, S. A., Henne, J. P., Carmichael, G. J., Tomasso, J. R. Toxicity of Ammonia and Nitrite to the Gila Trout. North American Journal of Aquaculture, 2003; 65:162–164.

APPENDIX I : Example GML Compliant Output From GCMD Service

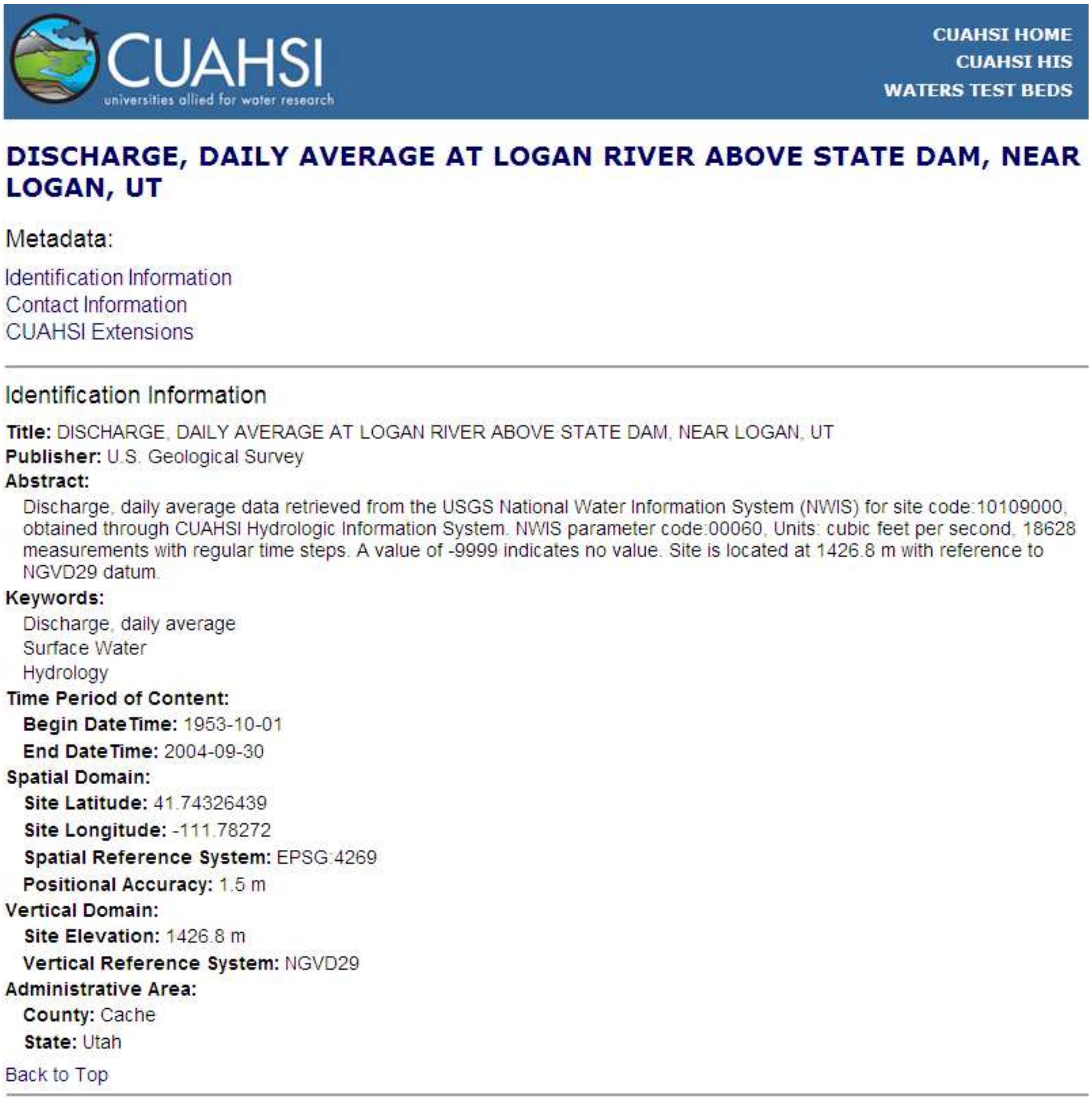
```
<?xml version="1.0" encoding="utf-8"?>
<StationCollection xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns="http://www.cuahsi.org/wmlf">
  <description>A collection of stations returned for GCMD keyword "AIR
TEMPERATURE"</description>
  <boundedBy xmlns="http://www.opengis.net/gml">
    <EnvelopeWithTimePeriod srsName="urn:ogc:def:crs:EPSG:6.6:4326">
      <lowerCorner>37.766 -75.762</lowerCorner>
      <upperCorner>38.372 -76.481</upperCorner>
      <timePosition indeterminatePosition="after">1996-01-
01T00:00:00</timePosition>
      <timePosition indeterminatePosition="before">2009-01-
01T00:00:00</timePosition>
    </EnvelopeWithTimePeriod>
  </boundedBy>
  <featureMembers xmlns="http://www.opengis.net/gml">
    <Station xmlns="http://www.cuahsi.org/wmlf">
      <location xmlns="http://www.opengis.net/gml">
        <Point>
          <pos>37.917911529541 -76.4716186523438</pos>
        </Point>
      </location>
      <stationID>1164197</stationID>
      <stationCode>NWIS:375502076281801</stationCode>
      <organization>USGS</organization>
    </Station>
    <Station xmlns="http://www.cuahsi.org/wmlf">
      <location xmlns="http://www.opengis.net/gml">
        <Point>
          <pos>37.9545745849609 -76.4313430786133</pos>
        </Point>
      </location>
      <stationID>1164261</stationID>
      <stationCode>NWIS:375716076255401</stationCode>
      <organization>USGS</organization>
    </Station>
    <Station xmlns="http://www.cuahsi.org/wmlf">
      <location xmlns="http://www.opengis.net/gml">
        <Point>
          <pos>38.1231803894043 -76.4218978881836</pos>
        </Point>
      </location>
      <stationID>1736636</stationID>
      <stationCode>NWIS:380724076251901</stationCode>
      <organization>USGS</organization>
    </Station>
  </featureMembers>
</StationCollection>
```

APPENDIX II : GML Application Schema for Measurement Sites

```
<?xml version="1.0" encoding="UTF-8"?>
  <schema
    targetNamespace="http://www.cuahsi.org/wmlf"
    xmlns:wmlf="http://www.cuahsi.org/wmlf"
    xmlns:gml="http://www.opengis.net/gml"
    xmlns:gmlsf="http://www.opengis.net/gmlsf"
    xmlns="http://www.w3.org/2001/XMLSchema"
    elementFormDefault="qualified"
    version="0.0.4"><annotation>
<appinfo
source="http://schemas.opengis.net/gml/3.1.1/profiles/gmlsfProfile/1.0.0/gmlsfLevels.
xsd"><gmlsf:ComplianceLevel>1</gmlsf:ComplianceLevel>
<gmlsf:GMLProfileSchema>http://schemas.opengis.net/gml/3.1.1/profiles/gmlsfProfil
e/1.0.0/gmlsf.xsd</gmlsf:GMLProfileSchema>
</appinfo>
</annotation>
<import namespace="http://www.opengis.net/gml"
schemaLocation="http://schemas.opengis.net/gml/3.1.1/base/gml.xsd"/>
<import namespace="http://www.opengis.net/gmlsf"
schemaLocation="http://schemas.opengis.net/gml/3.1.1/profiles/gmlsfProfile/1.0.0/gm
lsfLevels.xsd"/>
<element  name="Station" substitutionGroup="gml:_Feature">
 <complexType>
```

```
<annotation>
<documentation>Describes a data collection site</documentation>
</annotation>
<complexContent>
<extension base="gml:AbstractFeatureType">
<sequence>
<element name="stationID"  type="string"  minOccurs="1" maxOccurs="1"/>
<element name="stationCode" type="string"  minOccurs="1" maxOccurs="1"/>
<element name="organization"  type="string"  minOccurs="1" maxOccurs="1"/>
</sequence>
</extension>
</complexContent>
</complexType>
</element>
<element name="StationCollection" substitutionGroup="gml:_FeatureCollection">
<complexType>
<annotation>
<documentation>A collection of sites</documentation>
</annotation>
<complexContent>
<extension base="gml:AbstractFeatureCollectionType"/>
</complexContent>
</complexType>
</element>
</schema>
```

APPENDIX III : Rendered Metadata Output



DISCHARGE, DAILY AVERAGE AT LOGAN RIVER ABOVE STATE DAM, NEAR LOGAN, UT

Metadata:

Identification Information
Contact Information
CUAHSI Extensions

Identification Information

Title: DISCHARGE, DAILY AVERAGE AT LOGAN RIVER ABOVE STATE DAM, NEAR LOGAN, UT
Publisher: U.S. Geological Survey
Abstract:

Discharge, daily average data retrieved from the USGS National Water Information System (NWIS) for site code:10109000, obtained through CUAHSI Hydrologic Information System. NWIS parameter code:00060, Units: cubic feet per second, 18628 measurements with regular time steps. A value of -9999 indicates no value. Site is located at 1426.8 m with reference to NGVD29 datum.

Keywords:

Discharge, daily average
Surface Water
Hydrology

Time Period of Content:

Begin DateTime: 1953-10-01
End DateTime: 2004-09-30

Spatial Domain:

Site Latitude: 41.74326439
Site Longitude: -111.78272
Spatial Reference System: EPSG:4269
Positional Accuracy: 1.5 m

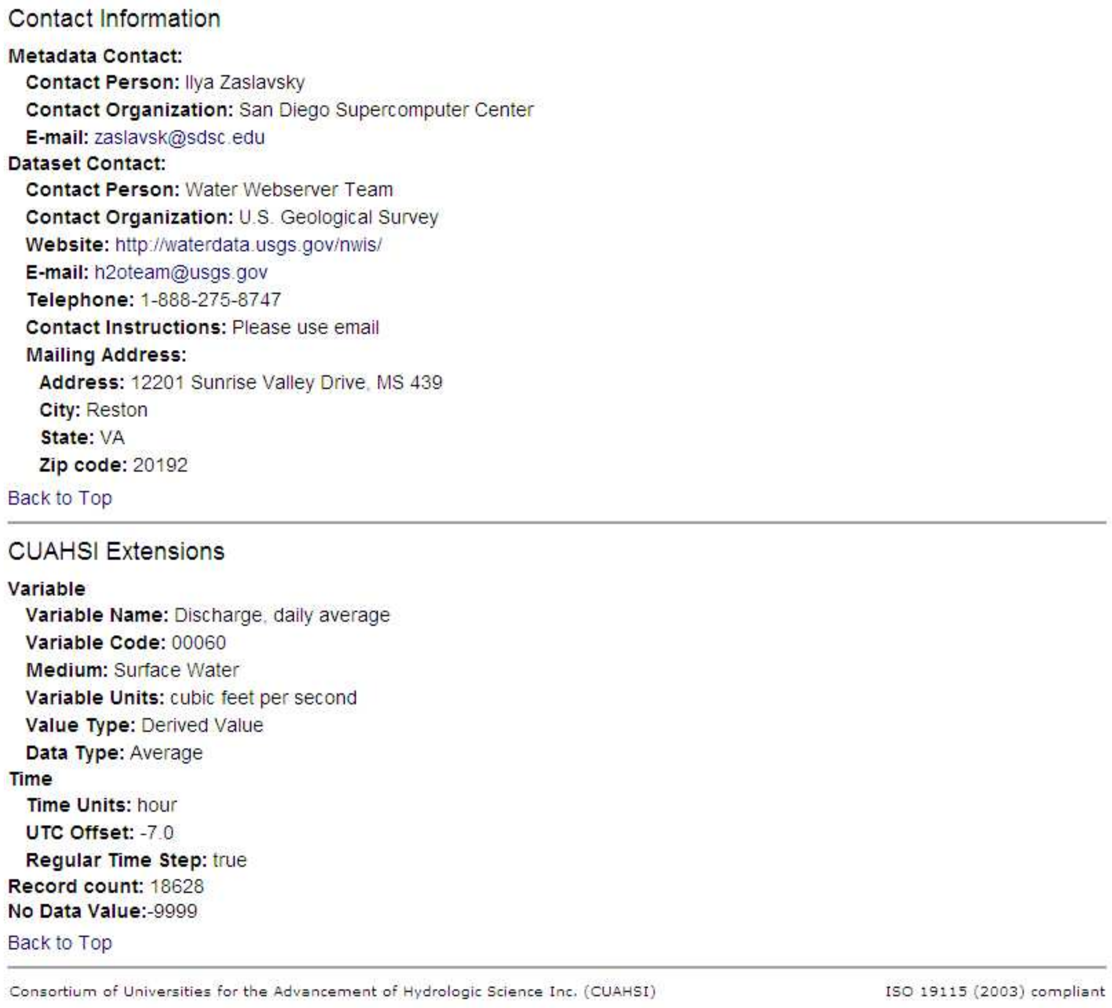
Vertical Domain:

Site Elevation: 1426.8 m
Vertical Reference System: NGVD29

Administrative Area:

County: Cache
State: Utah

Back to Top



Contact Information

Metadata Contact:
Contact Person: Ilya Zaslavsky
Contact Organization: San Diego Supercomputer Center
E-mail: zaslavsk@sdsc.edu

Dataset Contact:
Contact Person: Water Webserver Team
Contact Organization: U.S. Geological Survey
Website: http://waterdata.usgs.gov/nwis/
E-mail: h2oteam@usgs.gov
Telephone: 1-888-275-8747
Contact Instructions: Please use email
Mailing Address:
Address: 12201 Sunrise Valley Drive, MS 439
City: Reston
State: VA
Zip code: 20192

Back to Top

CUAHSI Extensions

Variable
Variable Name: Discharge, daily average
Variable Code: 00060
Medium: Surface Water
Variable Units: cubic feet per second
Value Type: Derived Value
Data Type: Average

Time
Time Units: hour
UTC Offset: -7.0
Regular Time Step: true

Record count: 18628
No Data Value:-9999

Back to Top

Consortium of Universities for the Advancement of Hydrologic Science Inc. (CUAHSI) ISO 19115 (2003) compliant

APPENDIX IV : ISO 19139 metadata instance with CUAHSI extensions

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="cuahsi-iso19115.xsl"?>
<gmd:MD_Metadata xmlns:gmd="http://www.isotc211.org/2005/gmd"
        xmlns:gco="http://www.isotc211.org/2005/gco"
        xmlns:gml="http://www.opengis.net/gml"
        xmlns:cuahsi="http://www.cuahsi.org/his"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="http://www.isotc211.org/2005/gmd
http://cbe.cae.drexel.edu/isoschemas/gmd/metadataEntity.xsd
http://www.isotc211.org/2005/gco
http://cbe.cae.drexel.edu/isoschemas/gco/gco.xsd ">
<gmd:fileIdentifier><gco:CharacterString>604</gco:CharacterString></gmd:fileIdentif
ier>
<gmd:language><gco:CharacterString>eng</gco:CharacterString></gmd:language>
<gmd:contact>
<gmd:CI_ResponsibleParty>
<gmd:individualName>
<gco:CharacterString>Ilya Zaslavsky</gco:CharacterString>
</gmd:individualName>
<gmd:organisationName>
<gco:CharacterString>San Diego Supercomputer Center</gco:CharacterString>
</gmd:organisationName>
<gmd:contactInfo>
```

```
<gmd:CI_Contact>
<gmd:address>
<gmd:CI_Address>
<gmd:electronicMailAddress><gco:CharacterString>zaslavsk@sdsc.edu</gco:Chara
cterString></gmd:electronicMailAddress>
</gmd:CI_Address>
</gmd:address>
</gmd:CI_Contact>
</gmd:contactInfo>
<gmd:role>
<gmd:CI_RoleCode
codeList="http://thor.cae.drexel.edu/cuahsi/ISO19139/resources/codeList.xml#CI_Ro
leCode"  codeSpace="Domain Code"
codeListValue="002">custodian</gmd:CI_RoleCode>
</gmd:role>
</gmd:CI_ResponsibleParty>
</gmd:contact>
<gmd:dateStamp>
<gco:Date>2004-09-30</gco:Date>
</gmd:dateStamp>
<gmd:metadataStandardName>
<gco:CharacterString>CUAHSI 19115</gco:CharacterString>
</gmd:metadataStandardName>
<gmd:metadataStandardVersion>
<gco:CharacterString>1.0</gco:CharacterString>
</gmd:metadataStandardVersion>
```

```
<gmd:identificationInfo>
<gmd:MD_DataIdentification>
<gmd:citation>
<gmd:CI_Citation>
<gmd:title><gco:CharacterString>DISCHARGE, DAILY AVERAGE AT LOGAN
RIVER ABOVE STATE DAM, NEAR LOGAN, UT</gco:CharacterString></gmd:title>
<gmd:date>
<gmd:CI_Date>
<gmd:date>
<gco:Date>2004-09-30</gco:Date>
</gmd:date>
<gmd:dateType>
<gmd:CI_DateTypeCode
codeList="http://thor.cae.drexel.edu/cuahsi/ISO19139/resources/codeList.xml#CI_DateTypeCode"  codeSpace="Domain Code"
codeListValue="003">revision</gmd:CI_DateTypeCode></gmd:dateType>
</gmd:CI_Date></gmd:date>
</gmd:CI_Citation>
</gmd:citation>
<gmd:abstract><gco:CharacterString>Discharge, daily average data retrieved from
the USGS National Water Information System (NWIS) for site code:10109000,
obtained through CUAHSI Hydrologic Information System. NWIS parameter
code:00060, Units: cubic feet per second, 18628 measurements with regular time
steps. A value of -9999 indicates no value. Site is located at 1426.8 m with reference
to NGVD29 datum. </gco:CharacterString></gmd:abstract>
<gmd:pointOfContact>
```

```xml
<gmd:CI_ResponsibleParty>
<gmd:individualName>
<gco:CharacterString>Water Webserver Team</gco:CharacterString>
</gmd:individualName>
<gmd:organisationName>
<gco:CharacterString>U.S. Geological Survey</gco:CharacterString>
</gmd:organisationName>
<gmd:contactInfo>
<gmd:CI_Contact>
<gmd:phone>
<gmd:CI_Telephone>
<gmd:voice>
<gco:CharacterString>1-888-275-8747</gco:CharacterString>
</gmd:voice>
</gmd:CI_Telephone>
</gmd:phone>
<gmd:address>
<gmd:CI_Address>
<gmd:deliveryPoint>
<gco:CharacterString>12201 Sunrise Valley Drive, MS 439</gco:CharacterString>
</gmd:deliveryPoint>
<gmd:city><gco:CharacterString>Reston</gco:CharacterString></gmd:city>
<gmd:administrativeArea>
<gco:CharacterString>VA</gco:CharacterString></gmd:administrativeArea>
<gmd:postalCode>
<gco:CharacterString>20192</gco:CharacterString>
```

```
</gmd:postalCode>
<gmd:electronicMailAddress>
<gco:CharacterString>h2oteam@usgs.gov</gco:CharacterString>
</gmd:electronicMailAddress>
</gmd:CI_Address>
</gmd:address>
<gmd:onlineResource>
<gmd:CI_OnlineResource><gmd:linkage><gmd:URL>http://waterdata.usgs.gov/nwis
/</gmd:URL>
</gmd:linkage>
</gmd:CI_OnlineResource></gmd:onlineResource>
<gmd:contactInstructions><gco:CharacterString>Please use
email</gco:CharacterString></gmd:contactInstructions>
</gmd:CI_Contact>
</gmd:contactInfo>
<gmd:role>
<gmd:CI_RoleCode
codeList="http://thor.cae.drexel.edu/cuahsi/ISO19139/resources/codeList.xml#CI_Ro
leCode"  codeSpace="Domain Code"
codeListValue="002">custodian</gmd:CI_RoleCode>
</gmd:role>
</gmd:CI_ResponsibleParty>
</gmd:pointOfContact>
<gmd:descriptiveKeywords>
<gmd:MD_Keywords>
<gmd:keyword>
```

```
<gco:CharacterString>Discharge, daily average</gco:CharacterString>
</gmd:keyword>
<gmd:keyword>
<gco:CharacterString>Surface Water</gco:CharacterString>
</gmd:keyword>
<gmd:keyword>
<gco:CharacterString>Hydrology</gco:CharacterString>
</gmd:keyword>
</gmd:MD_Keywords>
</gmd:descriptiveKeywords>
<gmd:language><gco:CharacterString>eng</gco:CharacterString></gmd:language>
<gmd:topicCategory>
<gmd:MD_TopicCategoryCode>inlandWaters</gmd:MD_TopicCategoryCode>
</gmd:topicCategory>
<gmd:extent>
<gmd:EX_Extent>
<gmd:description>
<gco:CharacterString>Site is located in Cache, Utah</gco:CharacterString>
</gmd:description>
<gmd:geographicElement>
<gmd:EX_BoundingPolygon>
<gmd:polygon>
<gml:Point gml:id="NWIS10109000" srsName="urn:ogc:def:crs:EPSG:6.7:4269">
<gml:pos>41.74326439 -111.78272</gml:pos>
</gml:Point>
</gmd:polygon>
```

```
</gmd:EX_BoundingPolygon>
</gmd:geographicElement>
<gmd:temporalElement>
<gmd:EX_TemporalExtent>
<gmd:extent>
<gml:TimePeriod gml:id="TS608">
<gml:beginPosition>1953-10-01</gml:beginPosition>
<gml:endPosition>2004-09-30</gml:endPosition>
</gml:TimePeriod>
</gmd:extent>
</gmd:EX_TemporalExtent>
</gmd:temporalElement>
<cuahsi:positionalAccuracy cuahsi:uom="m">1.5</cuahsi:positionalAccuracy>
<cuahsi:elevation cuahsi:verticalDatum="NGVD29"
  cuahsi:uom="m">1426.8</cuahsi:elevation>
<cuahsi:administrativeArea>
<cuahsi:county>Cache</cuahsi:county>
<cuahsi:state>Utah</cuahsi:state>
</cuahsi:administrativeArea>
</gmd:EX_Extent>
</gmd:extent>
<cuahsi:timeSeries>
<cuahsi:generalCategory>Hydrology</cuahsi:generalCategory>
<cuahsi:valueType>Derived Value</cuahsi:valueType>
<cuahsi:dataType>Average</cuahsi:dataType>
<cuahsi:valueCount>18628</cuahsi:valueCount>
```

```xml
<cuahsi:timeAxis cuahsi:uom="hr" cuahsi:unitLongName="hour" >
<cuahsi:isRegular>true</cuahsi:isRegular>
<cuahsi:utcOffset>-7.0</cuahsi:utcOffset>
</cuahsi:timeAxis>
<cuahsi:valueAxis cuahsi:uom="cfs" cuahsi:unitLongName="cubic feet per second" >
<cuahsi:noDataValue>-9999</cuahsi:noDataValue>
<cuahsi:variableCode>00060</cuahsi:variableCode>
<cuahsi:variableName>Discharge, daily average</cuahsi:variableName>
<cuahsi:measurementMedium>Surface Water</cuahsi:measurementMedium>
</cuahsi:valueAxis>
</cuahsi:timeSeries>
</gmd:MD_DataIdentification>
</gmd:identificationInfo>
</gmd:MD_Metadata>
```

APPENDIX V : XML Stylesheet Used in Rendering the ISO 19139 document

```
<?xml version="1.0"?>
<xsl:stylesheet version="2.0"
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
xmlns:gco="http://www.isotc211.org/2005/gco"
xmlns:gml="http://www.opengis.net/gml" xmlns:cuahsi="http://www.cuahsi.org/his"
xmlns:gmd="http://www.isotc211.org/2005/gmd">
<!--
=================================================================
============================== -->
<!-- = Created by Bora Beran for CUAHSI Hydrologic Information Systems Project.
December 19, 2006 = -->
<!--
=================================================================
============================== -->
<xsl:template match="/">
<html>
<head>
<title>CUAHSI HIS Metadata</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1"/>
<link rel="stylesheet" href="cuahsiXMLstylesheet.css" type="text/css"/>
</head>
<body bgcolor="#FFFFFF" text="#000000">
<table width="800" border="0" cellspacing="0" align="center">
```

```
<tr><td bgcolor="#34689A" rowspan="3"><img src="cuahsi.jpg" width="248"
height="75"/></td>
<td bgcolor="#34689A" height="31" valign="bottom"><div align="right"><b><font
class="TITL"><a href="http://www.cuahsi.org">CUAHSI HOME</a></font></b></div>
</td>
<td bgcolor="#34689A" height="30" valign="bottom">&#160;</td>
</tr><tr><td bgcolor="#34689A" height="23" valign="middle"><div
align="right"><b><font class="TITL"><a
href="http://www.cuahsi.org/his/hdas.html">CUAHSI HIS</a></font></b></div></td>
<td bgcolor="#34689A" height="23" valign="middle">&#160;</td>
</tr><tr><td bgcolor="#34689A" height="31" valign="top"><div align="right"><b><font
class="TITL"><a href="http://www.hydrologicscience.org/wtbs/index.html">WATERS
TEST BEDS</a></font></b></div></td>
<td bgcolor="#34689A" height="30" valign="top">&#160;</td>
</tr><tr><td colspan="3">&#160;</td></tr><tr><td colspan="3"><b><font
color="#000066" size="4" face="Verdana, Arial, Helvetica, sans-serif"
class="BTIL"><xsl:value-of
select="gmd:MD_Metadata/gmd:identificationInfo/gmd:MD_DataIdentification/gmd:cit
ation/gmd:CI_Citation/gmd:title/gco:CharacterString"/></font></b></td>
</tr><tr><td colspan="3">&#160;</td></tr><tr>
<td colspan="3"><font class="METATITL"><a name="top"></a>Metadata:</font>
</td></tr><tr> <td colspan="3" height="7"></td></tr><tr><td colspan="3"><font
size="3"><b><font face="Arial, Helvetica, sans-serif"><a href="#idinfo">Identification
Information </a></font></b></font></td>
</tr><tr align="left"><td colspan="3"><font size="3"><b><font face="Arial, Helvetica,
sans-serif"><a href="#continfo">ContactInformation</a></font></b></font></td>
```

```
</tr><tr><td colspan="3"><font size="3"><b><font face="Arial, Helvetica, sans-
serif"><a href="#tsinfo">CUAHSI Extensions</a></font></b></font></td>
</tr><tr> <td colspan="3" height="7"></td></tr><tr> <td colspan="3">
<hr noshade="noshade"/></td></tr><tr valign="top"><td colspan="3"
class="METATITL" height="30"><b><a name="idinfo"></a></b> Identification
Information</td></tr><tr><td colspan="3" class="BODYT"><b>Title: </b>
<xsl:value-of
select="gmd:MD_Metadata/gmd:identificationInfo/gmd:MD_DataIdentification/gmd:cit
ation/gmd:CI_Citation/gmd:title/gco:CharacterString"/></td></tr><tr><td colspan="3"
class="BODYT"><b>Publisher: </b><xsl:value-of
select="gmd:MD_Metadata/gmd:identificationInfo/gmd:MD_DataIdentification/gmd:po
intOfContact/gmd:CI_ResponsibleParty/gmd:organisationName/gco:CharacterString"
/></td></tr><tr> <td colspan="3" class="BODYT"><b>Abstract:</b></td></tr><tr><td
colspan="3" class="BODYT"><table width="100%" border="0" cellspacing="0">
<tr><td width="10">&#160;</td>
<td class="BODYT" width="790"><xsl:value-of
select="gmd:MD_Metadata/gmd:identificationInfo/gmd:MD_DataIdentification/gmd:ab
stract/gco:CharacterString"/></td></tr></table></td></tr><tr><td colspan="3"
class="BODYT"><b>Keywords:</b></td></tr><tr><td colspan="3" class="BODYT">
<table width="100%" border="0" cellspacing="0"><xsl:for-each
select="gmd:MD_Metadata/gmd:identificationInfo/gmd:MD_DataIdentification/gmd:de
scriptiveKeywords/gmd:MD_Keywords/gmd:keyword"><tr><td
width="10">&#160;</td><td class="BODYT" width="790"><xsl:value-of
select="gco:CharacterString"/></td></tr></xsl:for-each></table>
</td></tr><tr><td colspan="3" class="BODYT"><b>Time Period of Content:</b></td>
```

```
</tr><tr><td colspan="3" class="BODYT"><table width="100%" border="0"
cellspacing="0"><tr><td width="10">&#160;</td><td class="BODYT"
width="790"><b>Begin DateTime: </b><xsl:value-of
select="gmd:MD_Metadata/gmd:identificationInfo/gmd:MD_DataIdentification/gmd:ex
tent/gmd:EX_Extent/gmd:temporalElement/gmd:EX_TemporalExtent/gmd:extent/gml
:TimePeriod/gml:beginPosition"/></td></tr></table></td></tr><tr><td colspan="3"
class="BODYT">
<table width="100%" border="0" cellspacing="0"><tr><td width="10">&#160;</td><td
class="BODYT" width="790"><b>End DateTime: </b><xsl:value-of
select="gmd:MD_Metadata/gmd:identificationInfo/gmd:MD_DataIdentification/gmd:ex
tent/gmd:EX_Extent/gmd:temporalElement/gmd:EX_TemporalExtent/gmd:extent/gml
:TimePeriod/gml:endPosition"/></td></tr></table></td></tr>
<xsl:variable name="coordinates"
select="gmd:MD_Metadata/gmd:identificationInfo/gmd:MD_DataIdentification/gmd:ex
tent/gmd:EX_Extent/gmd:geographicElement/gmd:EX_BoundingPolygon/gmd:polygo
n/gml:Point/gml:pos"/><tr><td colspan="3" class="BODYT"><b>Spatial
Domain:</b></td></tr><tr><td colspan="3" class="BODYT"><table width="100%"
border="0" cellspacing="0"><tr><td width="10">&#160;</td><td class="BODYT"
width="790"><b>Site Latitude: </b><xsl:value-of select="substring-
before($coordinates,' ')" /></td></tr></table></td></tr>
<tr><td colspan="3" class="BODYT"><table width="100%" border="0"
cellspacing="0"><tr><td width="10">&#160;</td><td class="BODYT"
width="790"><b>Site Longitude: </b><xsl:value-of select="substring-
after($coordinates,' ')" /></td></tr>
</table>
</td>
```

```
</tr><tr><xsl:variable name="srs"
select="gmd:MD_Metadata/gmd:identificationInfo/gmd:MD_DataIdentification/gmd:ex
tent/gmd:EX_Extent/gmd:geographicElement/gmd:EX_BoundingPolygon/gmd:polygo
n/gml:Point/attribute::srsName"/><td colspan="3" class="BODYT"><table
width="100%" border="0" cellspacing="0"><tr><td width="10">&#160;</td><td
class="BODYT" width="790"><b>Spatial Reference System: </b>EPSG:<xsl:value-of
select="substring-after($srs,'urn:ogc:def:crs:EPSG:6.7:')" /></td></tr>
</table></td></tr>
<tr><td colspan="3" class="BODYT"><table width="100%" border="0"
cellspacing="0"><tr><td width="10">&#160;</td><td class="BODYT"
width="790"><b>Positional Accuracy: </b><xsl:value-of
select="gmd:MD_Metadata/gmd:identificationInfo/gmd:MD_DataIdentification/gmd:ex
tent/gmd:EX_Extent/cuahsi:positionalAccuracy"/> m</td></tr>
</table></td></tr><tr><td colspan="3" class="BODYT" height="5"><b>Vertical
Domain:</b></td></tr><tr><td colspan="3" class="BODYT"><table width="100%"
border="0" cellspacing="0"><tr><td width="10">&#160;</td><td class="BODYT"
width="790"><b>Site Elevation: </b><xsl:value-of
select="gmd:MD_Metadata/gmd:identificationInfo/gmd:MD_DataIdentification/gmd:ex
tent/gmd:EX_Extent/cuahsi:elevation"/> m</td></tr></table></td></tr>
<tr><td colspan="3" class="BODYT"><table width="100%" border="0"
cellspacing="0"><tr><td width="10">&#160;</td><td class="BODYT"
width="790"><b>Vertical Reference System: </b><xsl:value-of
select="gmd:MD_Metadata/gmd:identificationInfo/gmd:MD_DataIdentification/gmd:ex
tent/gmd:EX_Extent/cuahsi:elevation/attribute::cuahsi:verticalDatum"/></td></tr>
</table></td></tr><tr><td colspan="3" class="BODYT" height="5"><b>Administrative
Area:</b></td></tr><tr><td colspan="3" class="BODYT"><table width="100%"
```

```
border="0" cellspacing="0"><tr><td width="10">&#160;</td><td class="BODYT"
width="790"><b>County: </b><xsl:value-of
select="gmd:MD_Metadata/gmd:identificationInfo/gmd:MD_DataIdentification/gmd:ex
tent/gmd:EX_Extent/cuahsi:administrativeArea/cuahsi:county"/></td></tr>
</table></td></tr>
<tr><td colspan="3" class="BODYT"><table width="100%" border="0"
cellspacing="0"><tr><td width="10">&#160;</td><td class="BODYT"
width="790"><b>State: </b><xsl:value-of
select="gmd:MD_Metadata/gmd:identificationInfo/gmd:MD_DataIdentification/gmd:ex
tent/gmd:EX_Extent/cuahsi:administrativeArea/cuahsi:state"/></td></tr>
</table></td></tr><tr><td colspan="3" class="BODYT" height="5"></td></tr>
<tr><td colspan="3" class="BODYT"><a href="#top">Back to Top</a></td>
</tr><tr><td colspan="3" class="BODYT" height="10"><hr noshade="noshade"/>
</td></tr><tr valign="top"><td colspan="3" class="METATITL" height="30"><a
name="continfo"></a>ContactInformation</td></tr>
<tr><td colspan="3" class="BODYT"><p><b>Metadata Contact:</b></p></td></tr>
<tr><td colspan="3" class="BODYT"><table width="100%" border="0"
cellspacing="0"><tr><td width="10">&#160;</td><td class="BODYT"
width="790"><b>Contact Person: </b><xsl:value-of
select="gmd:MD_Metadata/gmd:contact/gmd:CI_ResponsibleParty/gmd:individualNa
me/gco:CharacterString"/></td></tr></table></td></tr>
<tr><td colspan="3" class="BODYT"><table width="100%" border="0"
cellspacing="0"><tr><td width="10">&#160;</td><td class="BODYT"
width="790"><b>Contact Organization: </b><xsl:value-of
select="gmd:MD_Metadata/gmd:contact/gmd:CI_ResponsibleParty/gmd:organisation
Name/gco:CharacterString"/></td></tr>
```

```
</table></td></tr><tr><td colspan="3" class="BODYT"><table width="100%"
border="0" cellspacing="0"><tr><td width="10">&#160;</td><td class="BODYT"
width="790"><b>E-mail: </b><a><xsl:attribute name="href">mailto:<xsl:value-of
select="gmd:MD_Metadata/gmd:contact/gmd:CI_ResponsibleParty/gmd:contactInfo/
gmd:CI_Contact/gmd:address/gmd:CI_Address/gmd:electronicMailAddress/gco:Char
acterString"/></xsl:attribute><xsl:value-of
select="gmd:MD_Metadata/gmd:contact/gmd:CI_ResponsibleParty/gmd:contactInfo/
gmd:CI_Contact/gmd:address/gmd:CI_Address/gmd:electronicMailAddress/gco:Char
acterString"/></a></td></tr></table></td></tr><tr><td colspan="3"
class="BODYT"><p><b>Dataset Contact:</b></p></td></tr>
<tr><td colspan="3" class="BODYT"><table width="100%" border="0"
cellspacing="0"><tr><td width="10">&#160;</td><td class="BODYT"
width="790"><b>Contact Person: </b><xsl:value-of
select="gmd:MD_Metadata/gmd:identificationInfo/gmd:MD_DataIdentification/gmd:po
intOfContact/gmd:CI_ResponsibleParty/gmd:individualName/gco:CharacterString"/>
</td></tr></table></td></tr>
<tr><td colspan="3" class="BODYT"><table width="100%" border="0"
cellspacing="0"><tr><td width="10">&#160;</td><td class="BODYT"
width="790"><b>Contact Organization: </b><xsl:value-of
select="gmd:MD_Metadata/gmd:identificationInfo/gmd:MD_DataIdentification/gmd:po
intOfContact/gmd:CI_ResponsibleParty/gmd:organisationName/gco:CharacterString"
/></td></tr></table></td></tr><tr><td colspan="3" class="BODYT"><table
width="100%" border="0" cellspacing="0"><tr><td width="10">&#160;</td><td
class="BODYT" width="790"><b>Website: </b><a><xsl:attribute
name="href"><xsl:value-of
select="gmd:MD_Metadata/gmd:identificationInfo/gmd:MD_DataIdentification/gmd:po
```

```
intOfContact/gmd:CI_ResponsibleParty/gmd:contactInfo/gmd:CI_Contact/gmd:online
Resource/gmd:CI_OnlineResource/gmd:linkage/gmd:URL"/></xsl:attribute>
<xsl:value-of
select="gmd:MD_Metadata/gmd:identificationInfo/gmd:MD_DataIdentification/gmd:po
intOfContact/gmd:CI_ResponsibleParty/gmd:contactInfo/gmd:CI_Contact/gmd:online
Resource/gmd:CI_OnlineResource/gmd:linkage/gmd:URL"/></a></td></tr>
</table></td></tr><tr><td colspan="3" class="BODYT"><table width="100%"
border="0" cellspacing="0"><tr><td width="10">&#160;</td><td class="BODYT"
width="790"><b>E-mail: </b><a><xsl:attribute name="href">mailto:<xsl:value-of
select="gmd:MD_Metadata/gmd:identificationInfo/gmd:MD_DataIdentification/gmd:po
intOfContact/gmd:CI_ResponsibleParty/gmd:contactInfo/gmd:CI_Contact/gmd:addre
ss/gmd:CI_Address/gmd:electronicMailAddress/gco:CharacterString"/></xsl:attribute
><xsl:value-of
select="gmd:MD_Metadata/gmd:identificationInfo/gmd:MD_DataIdentification/gmd:po
intOfContact/gmd:CI_ResponsibleParty/gmd:contactInfo/gmd:CI_Contact/gmd:addre
ss/gmd:CI_Address/gmd:electronicMailAddress/gco:CharacterString"/></a><b></b><
/td></tr></table></td></tr><tr><td colspan="3" class="BODYT"><table width="100%"
border="0" cellspacing="0"><tr><td width="10">&#160;</td><td class="BODYT"
width="790"><b>Telephone: </b><xsl:value-of
select="gmd:MD_Metadata/gmd:identificationInfo/gmd:MD_DataIdentification/gmd:po
intOfContact/gmd:CI_ResponsibleParty/gmd:contactInfo/gmd:CI_Contact/gmd:phone
/gmd:CI_Telephone/gmd:voice/gco:CharacterString"/></td></tr>
</table></td></tr>
<tr><td colspan="3" class="BODYT"><table width="100%" border="0"
cellspacing="0"><tr><td width="10">&#160;</td><td class="BODYT"
width="790"><b>Contact Instructions: </b><xsl:value-of
```

```
select="gmd:MD_Metadata/gmd:identificationInfo/gmd:MD_DataIdentification/gmd:po
intOfContact/gmd:CI_ResponsibleParty/gmd:contactInfo/gmd:CI_Contact/gmd:conta
ctInstructions/gco:CharacterString"/></td></tr>
</table></td></tr><tr><td colspan="3" class="BODYT"><table width="100%"
border="0" cellspacing="0"><tr><td width="10">&#160;</td><td class="BODYT"
width="790"><b>Mailing Address:</b></td></tr></table></td></tr><tr><td
colspan="3" class="BODYT"><table width="100%" border="0"
cellspacing="0"><tr><td width="20">&#160;</td><td class="BODYT"
width="790"><b>Address: </b><xsl:value-of
select="gmd:MD_Metadata/gmd:identificationInfo/gmd:MD_DataIdentification/gmd:po
intOfContact/gmd:CI_ResponsibleParty/gmd:contactInfo/gmd:CI_Contact/gmd:addre
ss/gmd:CI_Address/gmd:deliveryPoint/gco:CharacterString"/></td></tr>
</table></td></tr><tr><td colspan="3" class="BODYT"><table width="100%"
border="0" cellspacing="0"><tr><td width="20">&#160;</td><td class="BODYT"
width="790"><b>City: </b><xsl:value-of
select="gmd:MD_Metadata/gmd:identificationInfo/gmd:MD_DataIdentification/gmd:po
intOfContact/gmd:CI_ResponsibleParty/gmd:contactInfo/gmd:CI_Contact/gmd:addre
ss/gmd:CI_Address/gmd:city/gco:CharacterString"/></td></tr></table></td></tr>
<tr><td colspan="3" class="BODYT"><table width="100%" border="0"
cellspacing="0"><tr><td width="20">&#160;</td><td class="BODYT"
width="790"><b>State: </b><xsl:value-of
select="gmd:MD_Metadata/gmd:identificationInfo/gmd:MD_DataIdentification/gmd:po
intOfContact/gmd:CI_ResponsibleParty/gmd:contactInfo/gmd:CI_Contact/gmd:addre
ss/gmd:CI_Address/gmd:administrativeArea/gco:CharacterString"/></td></tr>
</table></td></tr><tr><td colspan="3" class="BODYT"><table width="100%"
border="0" cellspacing="0"><tr><td width="20">&#160;</td><td class="BODYT"
```

```
width="790"><b>Zip code: </b><xsl:value-of
select="gmd:MD_Metadata/gmd:identificationInfo/gmd:MD_DataIdentification/gmd:po
intOfContact/gmd:CI_ResponsibleParty/gmd:contactInfo/gmd:CI_Contact/gmd:addre
ss/gmd:CI_Address/gmd:postalCode/gco:CharacterString"/></td></tr>
</table></td></tr><tr><td colspan="3" class="BODYT" height="5"></td></tr>
<tr><td colspan="3" class="BODYT"><a href="#top">Back to Top</a></td>
</tr><tr><td colspan="3" class="BODYT" height="10"><hr noshade="noshade"/>
</td></tr><tr valign="top"><td colspan="3" class="BODYT" height="30"><p
class="METATITL"><a name="tsinfo"></a>CUAHSI Extensions</p>
</td></tr><tr><td colspan="3" class="BODYT"><b>Variable</b></td>
</tr><tr><td colspan="3" class="BODYT"><table width="100%" border="0"
cellspacing="0"><tr><td width="10">&#160;</td><td class="BODYT"
width="790"><b>Variable Name: </b><xsl:value-of
select="gmd:MD_Metadata/gmd:identificationInfo/gmd:MD_DataIdentification/cuahsi:
timeSeries/cuahsi:valueAxis/cuahsi:variableName"/></td></tr></table></td></tr>
<tr><td colspan="3" class="BODYT"><table width="100%" border="0"
cellspacing="0"><tr><td width="10">&#160;</td><td class="BODYT"
width="790"><b>Variable Code: </b><xsl:value-of
select="gmd:MD_Metadata/gmd:identificationInfo/gmd:MD_DataIdentification/cuahsi:
timeSeries/cuahsi:valueAxis/cuahsi:variableCode"/></td></tr></table></td></tr>
<tr><td colspan="3" class="BODYT"><table width="100%" border="0"
cellspacing="0"><tr><td width="10">&#160;</td><td class="BODYT"
width="790"><b>Medium: </b><xsl:value-of
select="gmd:MD_Metadata/gmd:identificationInfo/gmd:MD_DataIdentification/cuahsi:
timeSeries/cuahsi:valueAxis/cuahsi:measurementMedium"/></td></tr>
```

```
</table></td></tr><tr><td colspan="3" class="BODYT"><table width="100%"
border="0" cellspacing="0"><tr><td width="10">&#160;</td><td class="BODYT"
width="790"><b>Variable Units: </b><xsl:value-of
select="gmd:MD_Metadata/gmd:identificationInfo/gmd:MD_DataIdentification/cuahsi:
timeSeries/cuahsi:valueAxis/attribute::cuahsi:unitLongName"/></td></tr></table>
</td></tr><tr><td colspan="3" class="BODYT"><table width="100%" border="0"
cellspacing="0"><tr><td width="10">&#160;</td><td class="BODYT"
width="790"><b>Value Type: </b><xsl:value-of
select="gmd:MD_Metadata/gmd:identificationInfo/gmd:MD_DataIdentification/cuahsi:
timeSeries/cuahsi:valueType"/></td></tr></table></td></tr><tr><td colspan="3"
class="BODYT"><table width="100%" border="0" cellspacing="0"><tr><td
width="10">&#160;</td><td class="BODYT" width="790"><b>Data Type:
</b><xsl:value-of
select="gmd:MD_Metadata/gmd:identificationInfo/gmd:MD_DataIdentification/cuahsi:
timeSeries/cuahsi:dataType"/></td></tr></table></td></tr><tr><td colspan="3"
class="BODYT"> <b>Time</b></td></tr>
<tr><td colspan="3" class="BODYT"><table width="100%" border="0"
cellspacing="0"><tr><td width="10">&#160;</td><td class="BODYT"
width="790"><b>Time Units: </b><xsl:value-of
select="gmd:MD_Metadata/gmd:identificationInfo/gmd:MD_DataIdentification/cuahsi:
timeSeries/cuahsi:timeAxis/attribute::cuahsi:unitLongName"/></td></tr>
</table></td></tr><tr><td colspan="3" class="BODYT"><table width="100%"
border="0" cellspacing="0"><tr><td width="10">&#160;</td><td class="BODYT"
width="790"><b>UTC Offset: </b><xsl:value-of
select="gmd:MD_Metadata/gmd:identificationInfo/gmd:MD_DataIdentification/cuahsi:
timeSeries/cuahsi:timeAxis/cuahsi:utcOffset"/></td></tr>
```

```
</table></td></tr><tr><td colspan="3" class="BODYT"><table width="100%"
border="0" cellspacing="0"><tr><td width="10">&#160;</td><td class="BODYT"
width="790"><b>Regular Time Step: </b><xsl:value-of
select="gmd:MD_Metadata/gmd:identificationInfo/gmd:MD_DataIdentification/cuahsi:
timeSeries/cuahsi:timeAxis/cuahsi:isRegular"/></td></tr>
</table></td></tr><tr><td colspan="3" class="BODYT"><b>Record count:
</b><xsl:value-of
select="gmd:MD_Metadata/gmd:identificationInfo/gmd:MD_DataIdentification/cuahsi:
timeSeries/cuahsi:valueCount"/></td></tr><tr><td colspan="3"
class="BODYT"><b>No Data Value:</b><xsl:value-of
select="gmd:MD_Metadata/gmd:identificationInfo/gmd:MD_DataIdentification/cuahsi:
timeSeries/cuahsi:valueAxis/cuahsi:noDataValue"/></td></tr><tr><td colspan="3"
class="BODYT" height="5"></td></tr><tr><td colspan="3" class="BODYT"><a
href="#top">Back to Top</a></td></tr><tr><td colspan="3" class="BODYT"
height="10"><hr noshade="noshade"/></td></tr><tr><td colspan="3" class="BODYT"
height="20"><div align="left"> <table width="100%" border="0"
cellspacing="0"><tr><td width="76%"><div align="left"><font size="1">Consortium of
Universities forthe Advancement of Hydrologic Science Inc.
(CUAHSI)</font></div></td><td width="24%"><div align="right"><font size="1">ISO
19115 (2003) compliant</font></div></td></tr></table></div></td></tr></table>
</body>
</html>
</xsl:template>
</xsl:stylesheet>
```

APPENDIX VI : ACRONYMS

ACWI	Advisory Committee on Water Information
ADAPS	Automated Data-Processing System
AJAX	Asynchronous JavaScript And XML
ANSI	American National Standards Institute
API	Application Programming Interface
ASCII	American Standard Code for Information Interchange
ASOS	Automated Surface Observing System
ASP	Active Server Pages
ASTER	Advanced Spaceborne Thermal Emission and Reflection Radiometer
AVHRR	Advanced Very High Resolution Radiometer
AWUDS	Aggregate Water-Use Data System
BFO	Basic Formal Ontology
BioML	Biopolymer Markup Language
BSML	Bioinformatic Sequence Markup Language
C-CAP	Coastal Change Analysis Program
CD	Climatological Data
CDIAC	Carbon Dioxide Information Analysis Center
CDX	Central Data Exchange
CERN	Conseil Européen pour la Recherche Nucléaire (European Council for Nuclear Research)

CIMS	Chesapeake Information Management System
CLASS	Comprehensive Large Array-data Stewardship System
CML	Chemical Markup Language
CORBA	Common Object Request Broker Architecture
CSS	Cascading Style Sheets
CUAHSI	Consortium of Universities for Advancement of Hydrologic Science Inc.
DAML	DARPA Agent Markup Language
DARPA	Defense Advanced Research Projects Agency
DAYMET	Daily Meteorological Summaries
DEM	Digital Elevation Model
DL	Description Logics
DLESE	Digital Library for Earth System Education
DLG	Digital Line Graph
DOLCE	Descriptive Ontology for Linguistic and Cognitive Engineering
DOM	Document Object Model
DTD	Document Type Definition
EDNA	Elevation Derivatives for National Applications
EML	Ecological Metadata Language
EPA	Environmental Protection Agency
EROS	Earth Resources Observation and Science
ESAR	Environmental Sampling, Analysis, and Results
ESML	Earth Science Markup Language

ESRI	Environmental Systems Research Institute
ETM	Enhanced Thematic Mapper
FGDC	Federal Geographic Data Committee
FOL	First-order Logic
FTP	File Transfer Protocol
FTU	Formazin Turbidity Unit
GCMD	Global Change Master Directory
GDS	Ground Data System
GEON	Geosciences Network
GES	Goddard Earth Sciences
GIS	Geographical Information System
GML	Generalized Markup Language or Geography Markup Language
GNIS	Geographic Names Information System
GNS	Geonet Names Server
GO	Gene Ontology
GOA	Gene Ontology Annotation database
GOES	Geostationary Operational Environmental Satellite
GWSI	Groundwater Site Inventory
HP	Hourly Precipitation
HTML	Hypertext Markup Language
HUC	Hydrologic Unit Code
ISO	International Organization for Standardization
JSON	JavaScript Object Notation

JVM	Java Virtual Machine
KIF	Knowledge Interchange Format
LANDSAT	Land Remote Sensing Satellite
LCD	Local Climatological Data
LDC	Legacy Data Center
LEAD	Linked Environment for Atmospheric Discovery
LIDAR	Light Detection and Ranging
LP DAAC	Land Processes Distributed Active Archive Center
MathML	Mathematical Markup Language
MODIS	Moderate Resolution Imaging Spectroradiometer
MOP	Memory Organization Packets
MRLC	Multi-Resolution Land Characteristics
MSL	Mean Sea Level
NAD27	North American Datum of 1927
NAD83	North American Datum of 1983
NARR	North American Regional Reanalysis
NASA	National Aeronautics and Space Administration
NATSGO	National Soil Geographic database
NAVD88	North American Vertical Datum of 1988
NCAR	National Center for Atmospheric Research
NCDC	National Climatic Data Center
NCEP	National Centers for Environmental Prediction
NED	National Elevation Dataset

NEXRAD	Next Generation Weather Radar
NGVD29	National Geodetic Vertical datum of 1929
NHD	National Hydrography Dataset
NLCD	National Land Cover Database
NOAA	National Oceanic and Atmospheric Administration
NOMADS	National Operational Model Archive and Distribution System
NRCS	Natural Resources Conservation Service
NTU	Nephelometric Turbidity Unit
NWIS	National Water Information System
NWQMC	National Water Quality Monitoring Council
ODM	Observations Data Model
OIL	Ontology Inference Layer aka Ontology Interchange Language
OKBC	Open Knowledge Base Connectivity
OLAP	Online Analytical Processing
OpenDAP	Open Data Access Protocol
OWL	Web Ontology Language
PDF	Portable Document Format
POES	Polar Orbiting Satellite
PRISM	Parameter-elevation Regressions on Independent Slopes Model
R-CDAS	Regional Climate Data Assimilation System
RDF	Resource Description Framework
RDFS	RDF Schema
RELAX NG	Regular Language for XML Next Generation

RELAX	Regular Language for XML
REST	Representational State Transfer
RPC	Remote Procedure Call
RXR	Regular XML RDF
SCS	Soil Conservation Service
SD	Storm Data
SDDS	Seamless Data Distribution System
SDSC	San Diego Supercomputer Center
SGML	Standard Generalized Markup Language
SKOS	Simple Knowledge Organization System
SNOTEL	Snowpack Telemetry
SOA	Services-Oriented Architecture
SOAP	Simple Object Access Protocol
SSURGO	Soil Survey Geographic database
STATSGO	State Soil Geographic database
STORET	Storage and Retrieval system
SUMO	Suggested Upper Merged Ontology
SWEET	Semantic Web for Earth and Environmental Terminology
SWUDS	Site-Specific Water-Use Data System
TCEQ	Texas Commission on Environmental Quality
TOP	Thematic Organization Packets
TRACS	Texas Regulatory and Compliance System
TREX	Tree Regular Expressions for XML

TriX	Triples in XML
Turtle	Terse RDF Triple Language
UMLS	Unified Medical Language System
URI	Uniform Resource Identifier
USDA	U.S. Department of Agriculture
USGS	United States Geological Survey
UTC	Coordinated Universal Time
W3C	World Wide Web Consortium
WaterML	Water Markup Language
WIST	Warehouse Inventory Search Tool
WQ	Water Quality
WQDE	Water Quality Data Elements
WQX	Water Quality Exchange
WSDL	Web Services Description Language
WS-I	Web Services Interoperability Organization
WUDS	Water-Use Data System
WWW	World Wide Web
XHTML	Extensible Hypertext Markup Language
XML	Extensible Markup Language
XOL	XML-Based Ontology Exchange Language
XSD	XML Schema Definition
XSL	Extensible Stylesheet Language

VITA

Education

- Ph.D. Civil Engineering, Drexel University, Philadelphia, PA. 2007
- M.S. Environmental Science, Dokuz Eylul University, Turkey. 2003
- B.S. Environmental Engineering, Yildiz Technical University, Turkey. 2001

Awards

- George Hill, Jr. Endowed Fellowship Recipient, Drexel University, Philadelphia, PA (2006/07)
- College of Engineering, Outstanding Teaching Assistant Award Recipient, Drexel University, Philadelphia, PA (2005/06)

Peer-Reviewed Publications

- Beran B., Piasecki M., (2007). Engineering new paths to water data, Submitted to Computers and Geosciences.
- Beran B., Piasecki M., (2007). Availability and Coverage of Hydrologic Data in the US: A Close Look at the USGS National Water Information System (NWIS) and EPA Storage and Retrieval System (STORET), Submitted to Water Resources Research
- Ruddell B. L., Zaslavsky I., Beran B., Kumar P., Fu Q. and Piasecki M. (2007). Lessons Learned from the Implementation of a Prototype Virtual Observatory Digital Library for the Illinois River Basin, Submitted to Geoinformatica
- Beran B., Kargi F., (2005). A dynamic mathematical model for wastewater stabilization ponds, Ecological Modelling, (181) 39-57