CUAHSI
universities allied for water research

HydroSeek

Functional Description Document
(Version 1.0)

A guide through and summary of the underlying technologies for the global search engine HydroSeek

March 2008

Prepared by:
Michael Piasecki

Department of Civil, Architectural & Environmental Engineering
Drexel University
Philadelphia, PA 19104

Distribution

Table of Contents

1 INTRODUCTION

The HydroSeek search engine is a means for discovering data that is published through use of the standard WaterOneFlow Web Services developed by the CUAHSI HIS group. WaterOneFlow Web Services have been developed for agency data (USGS NWIS, EPA STORET, NCDC) as well as academic data (WATERS Network test beds). HydroSeek provides integrated federated search capability across these multiple data sources through a single interface and thereby 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. In short, this system is designed to provide a one-stop search center in which a user can query for specific data groups across all registered data sources regardless the storage and description conventions used by the respective warehouses.

HydroSeek is supported by (1) a registry for Web Services that are searchable by the system, (2) a catalog database that stores key metadata for the data available from each Web Service, (3) a conceptual hierarchy of keywords (ontology) onto which each entry in the catalog is mapped to support the thematic search capability. The HydroSeek Catalog is stored in a modified version of the CUAHSI HIS Observations Data Model (ODM) (reference?) that has stripped out information not needed for the catalog and added tables needed for mapping onto the ontology and recording additional spatial information (e.g., HUC codes used in spatial search). HydroSeek Web Services are used to access the catalog and ontology. The catalog is populated automatically by a Catalog Harvester component that periodically scans the registered Web Services for any new variables that may be available (reference to the catalog harveseter in the tagger doc?). Mapping of catalog entries onto the ontology is done manually using the HydroTagger interface (reference?).

The HydroSeek system was originally coded in JAVA, but the decision by the HIS group to use the .NET framework for easier integration into existing software applications and better compatibility with future software developments prompted the need to recode all software components using the .NET environment. This was done also with the intention of supplying the Data Access System for Hydrology (DASH), an application based on the ArcGIS environment, with a sophisticated search system that could be invoked within DASH. Throughout the time of development it became clear that the basic search components were in need of their own Graphical User Interface (GUI) environment thus resulting in a standalone application that eventually was named HydroSeek.

This document seeks to outline the functionality requirements that support the HydroSeek search environment as well as to define key technologies used in this approach, namely the knowledge base comprised of a collection of layered ontologies that support the keyword search.

2 HydroSeek Search Engine

The HydroSeek search engine is a web application for discovering and downloading hydrologic observations data. The objectives behind creating this application are described in this section, followed by a discussion of search strategies. Technologies that facilitate implementation of the search strategy, including an underlying ODM-based database and Web Services, are then described. Finally, the workflow for executing a search on the system is provided.

2.1 Objectives

One of the primary objectives of the HydroSeek search engine is to permit "smart" searches. What is meant by this is the desire to avoid so called high-precision-low-recall and low-precision-high-recall returns that deliver too little in the first case and too much in the second, a problem that prevails when using Google-type searches. Key to a successful search experience is:

- To provide search keywords that are easily understood or commonly used without having to learn the search vocabulary a priori
- To prevent too much return by blocking keywords that are too general on the one side, and too specific on the other thus preventing a successful search return
- To aid the user in formulating keywords through help functions like auto-complete features that offer up what keywords the system can accept
- To provide synonyms in case several keyword terms exist that should lead to the exact same return of search results
- To provide a detailed search result classification system that structures the return according to the underlying concepts and ODM attributes like DataType
- To offer an intuitive graphical user interface that puts the spatial search components into a geospatial context be it a bounding box or the Hydrologic Unit Code (HUC) classification system
- To ensure that the search execution is carried out fast, i.e., the need to limit the search activity to just a few seconds thus enhancing the user experience
- To permit a selection process in which users can request and download selected data series
- To provide a system that is extensible, can grow over time, that automatically updates the underlying metadata catalogues, and that is easily accessible from within a web browser thus avoiding downloads of specific software

Many of the above objectives can be addressed through the use of available technologies for example the Asynchronous JavaScript and XML, AJAX, the use of Microoft's Virtual Earth environment, but also the use of ontologies and their implementation language OWL.

2.2 Search Strategies

A fundamentally important aspect entering the development process is the objective of "having a good search experience". This is also equivalent to addressing what is commonly referred to as

- High-Precision-Low-Recall problem
- Low-Precision-High-Recall problem

The former represents a situation in which the search keyword is too detailed so that little or nothing is being returned, while the latter means that the search request was so general that an inordinate number of returns result. Neither situation is desirable during a search prompting the need to select a path in the middle. Obviously, the degree of precision (or generality) represents lower and upper bounds of a region that one could call “reasonable entries”. A good choice on the upper end is motivated by the desire that the user should not be overwhelmed by data returns, i.e., there should be a reasonable degree of focus on a general area of interest, like “Nutrients”, when looking for Water or Soil Quality parameters. This is somewhat of a subjective choice and decision but it is aided by experience and some initial testing.

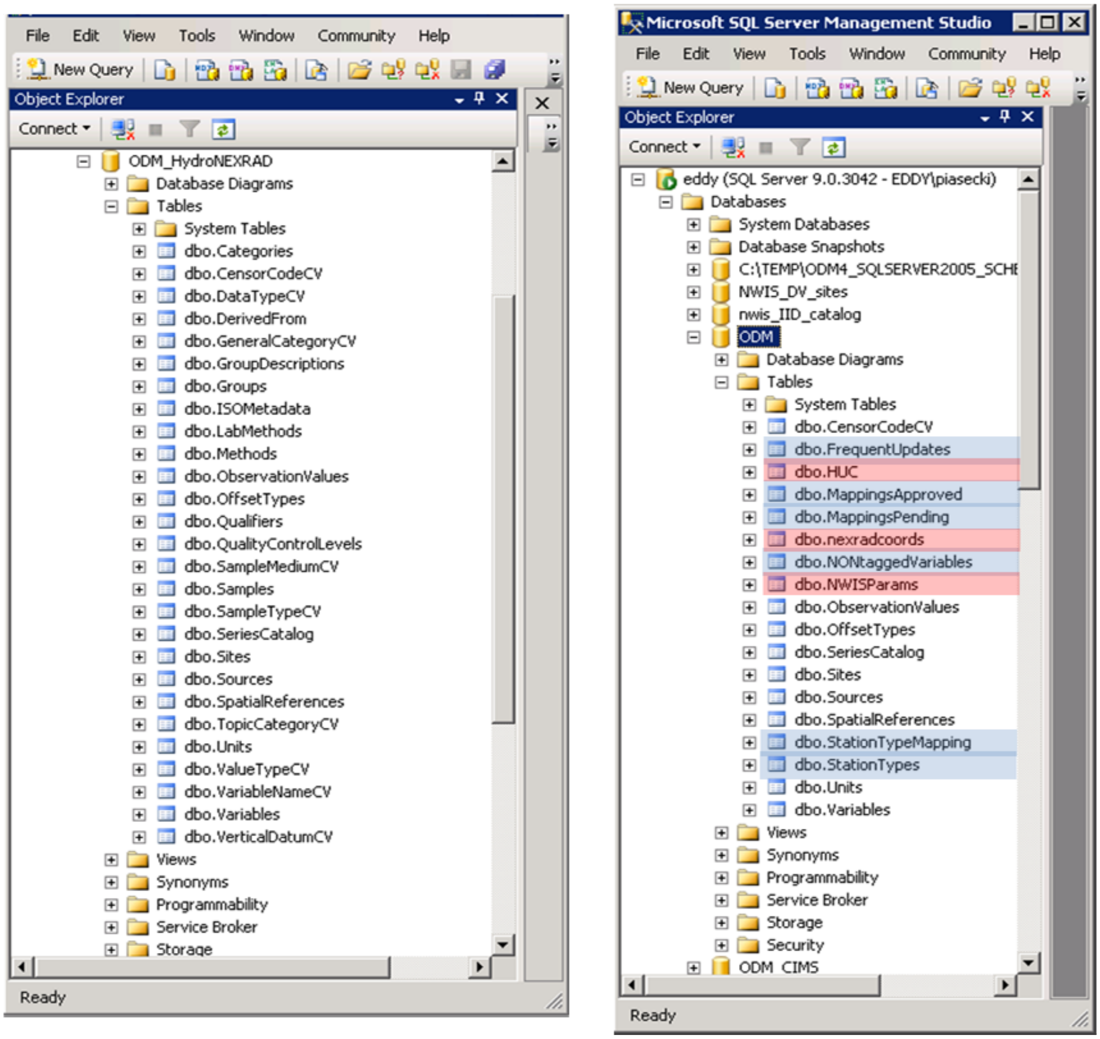
The lower bound on the other hand is easier to define because it is terminated at the leaf level (see later in section 3 for an explanation), i.e., the user should be removed from having to know any of the network (data source) specific codes or names that are used to identify the parameters and variables in any of those networks.. This is precisely where the reduction in work and the semantic mediation takes place: Do not search for individual network variable codes and names, rather search for a concept just one (upward) level more general. For example, instead of needing to know a variable code like “NWIS:76008” or the associated variable name “Nitrate, suspended sediment, total, milligrams per liter as nitrogen” to search for this variable from the National Water Information System, just know a concept like “Nitrate” that will search for the above variable in addition to 28 other Nitrate variables that have been mapped to this concept.

Search performance can also be measured in speedy returns, which is another important aspect when deciding the upper and lower bounds. Too general a search means that the server side application needs more time to execute the search because a larger selection of keywords must be traversed in the ontology and then compiled into a XML based result file, which in turn needs to be sent across the internet. Tests have shown that the patience threshold of waiting for web-page-based activity is about 6 seconds, sometimes less. Hence, search strategy in this case is also influenced by the GUI performance and by how quick the search of the tables can actually be executed and results returned to the browser for display. In addition, there is only so much information that can be displayed on the GUI without making it appear cluttered possibly even prompting scrolling.

2.3 MODIFIED ODM DATABASE

The supporting database for HydroSeek contains all site and variable records from all networks that are registered in HIS Central; i.e., it represents a conglomeration and collection of all data records that are available in the WATERS network as well as several national data sources. The database itself (see right panel of Figure 1) is a modified version of the original ODM 1.0 design (see left panel of Figure 1). Because the search engine makes use of a limited number of the tables in the ODM, the HydroSeek DB is for the most part a stripped down version of the ODM 1.0. However, it does retain those tables that are necessary to support the WaterOneFlow Web Services (minus the GetValues service) and actually adds a few tables in order to support the search functionality in addition to those tables needed by the semantic tagging application HydroTagger. Hence, while the origin of the database is that of the ODM, the HydroSeek DB is a customized derivative of it that cannot be considered an ODM installation any more.

In Figure 1, the tables highlighted in blue are those that have been added to support the updating features whereby the tables addressing StationTypes and StationTypeMapping are an attempt to reduce the “wealth” of available station types in EPA STORET (25 types) and USGS NWIS Daily Values (38 types) to a more manageable number in the CUAHSI HIS system (5). This reduction can then be used in HydroSeek to further classify search results. The tables in red have been added purely as an auxiliary measure, i.e., to have the information available in case it is needed, but could have been placed elsewhere also. The HUC table contains (up to 8-digit codes) the codes and their respective name, the table NWISParams contains all parameters used by NWIS (9610, of which about 400 have been mapped in HydroSeek), and the table nexradcoords contains the station latitude/longitude pairs for the Chesapeake Bay watershed (this is a table that logically belongs to a different DB and will be moved in the next “cleanup”).


Microsoft SQL Server Management Studio
File Edit View Tools Window Community Help
New Query
Object Explorer
Connect
Ready

```
ODM_HydroNEXRAD
    Database Diagrams
    Tables
        System Tables
        dbo.Categories
        dbo.CensorCodeCV
        dbo.DataTypeCV
        dbo.DerivedFrom
        dbo.GeneralCategoryCV
        dbo.GroupDescriptions
        dbo.Groups
        dbo.ISOMetadata
        dbo.LabMethods
        dbo.Methods
        dbo.ObservationValues
        dbo.OffsetTypes
        dbo.Qualifiers
        dbo.QualityControlLevels
        dbo.SampleMediumCV
        dbo.Samples
        dbo.SampleTypeCV
        dbo.SeriesCatalog
        dbo.Sites
        dbo.Sources
        dbo.SpatialReferences
        dbo.TopicCategoryCV
        dbo.Units
        dbo.ValueTypeCV
        dbo.VariableNameCV
        dbo.Variables
        dbo.VerticalDatumCV
    Views
    Synonyms
    Programmability
    Service Broker
    Storage
```

```
eddy (SQL Server 9.0.3042 - EDDY\piasecki)
    Databases
        System Databases
        Database Snapshots
        C:\TEMP\ODM4_SQLSERVER2005_SCHE
        NWIS_DV_sites
        nwis_IID_catalog
        ODM
            Database Diagrams
            Tables
                System Tables
                dbo.CensorCodeCV
                dbo.FrequentUpdates
                dbo.HUC
                dbo.MappingsApproved
                dbo.MappingsPending
                dbo.nexradcoords
                dbo.NONtaggedVariables
                dbo.NWISParams
                dbo.ObservationValues
                dbo.OffsetTypes
                dbo.SeriesCatalog
                dbo.Sites
                dbo.Sources
                dbo.SpatialReferences
                dbo.StationTypeMapping
                dbo.StationTypes
                dbo.Units
                dbo.Variables
            Views
            Synonyms
            Programmability
            Service Broker
            Storage
            Security
        ODM_CIMS
```

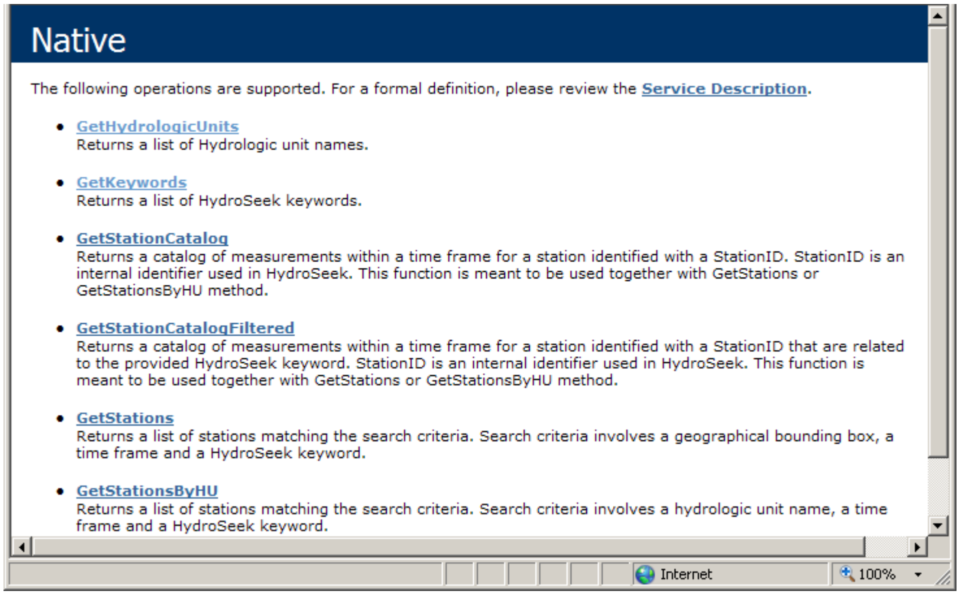
Figure 1 Original ODM (left) and the modified HydroSeek ODM (right).

2.4 Web Services

HydroSeek uses the regular WaterOneFlow Web Services in addition to some “native” services. Native services include:

- GetHydrologicUnits
- GetKeywords
- GetStationCatalog
- GetStationCatalogFiltered
- GetStations
- GetStationsByHU

These services and descriptions are shown in the figure below.



Native

The following operations are supported. For a formal definition, please review the **Service Description**.

- **GetHydrologicUnits**
  Returns a list of Hydrologic unit names.
- **GetKeywords**
  Returns a list of HydroSeek keywords.
- **GetStationCatalog**
  Returns a catalog of measurements within a time frame for a station identified with a StationID. StationID is an internal identifier used in HydroSeek. This function is meant to be used together with GetStations or GetStationsByHU method.
- **GetStationCatalogFiltered**
  Returns a catalog of measurements within a time frame for a station identified with a StationID that are related to the provided HydroSeek keyword. StationID is an internal identifier used in HydroSeek. This function is meant to be used together with GetStations or GetStationsByHU method.
- **GetStations**
  Returns a list of stations matching the search criteria. Search criteria involves a geographical bounding box, a time frame and a HydroSeek keyword.
- **GetStationsByHU**
  Returns a list of stations matching the search criteria. Search criteria involves a hydrologic unit name, a time frame and a HydroSeek keyword.

Figure 2 HydroSeek Native Web Services.

The first two return alphabetically ordered lists of Hydrologic Unit Code (HUC) names (not the codes but the watershed names) and all permissible search keywords, respectively. The former is used in the “search-by-watershed” function of the search engine GUI, and the latter is used by the AJAX auto-complete when typing the keywords.

The latter 4 are used to return lists of stations and/or measurements based on the time period, keyword, and then either Bounding Box or HUC code based query. These are repeatedly invoked depending on how many variable codes have been identified in the search branch. The Web Service WSDL can be found at: http://hydroseek.org/search/webservices/Native.asmx

2.5 Search Execution

The search is executed using the following sequence:

a) Select a search region, currently either bounding box or (up to 8-digit) HUC watershed. Current default is Chesapeake Bay.

b) Select time frame. Default is last 10 years.

c) Select keyword. There is an auto-complete once the first 4 letters are typed. Permissible keywords are those defined in the “Compound” and “Core” ontology layers (see next section). There is no default. No keyword means no return.

d) The keyword along with the spatial and temporal bounds is passed to the search code on the server.

e) Next the code interprets the keyword and locates its position in the ontology.

f) It then traverses DOWN the branch and collects all leaf concepts (actually the ConceptCodes) that are part of this branch. It keeps the branching information and stores it an XML file.

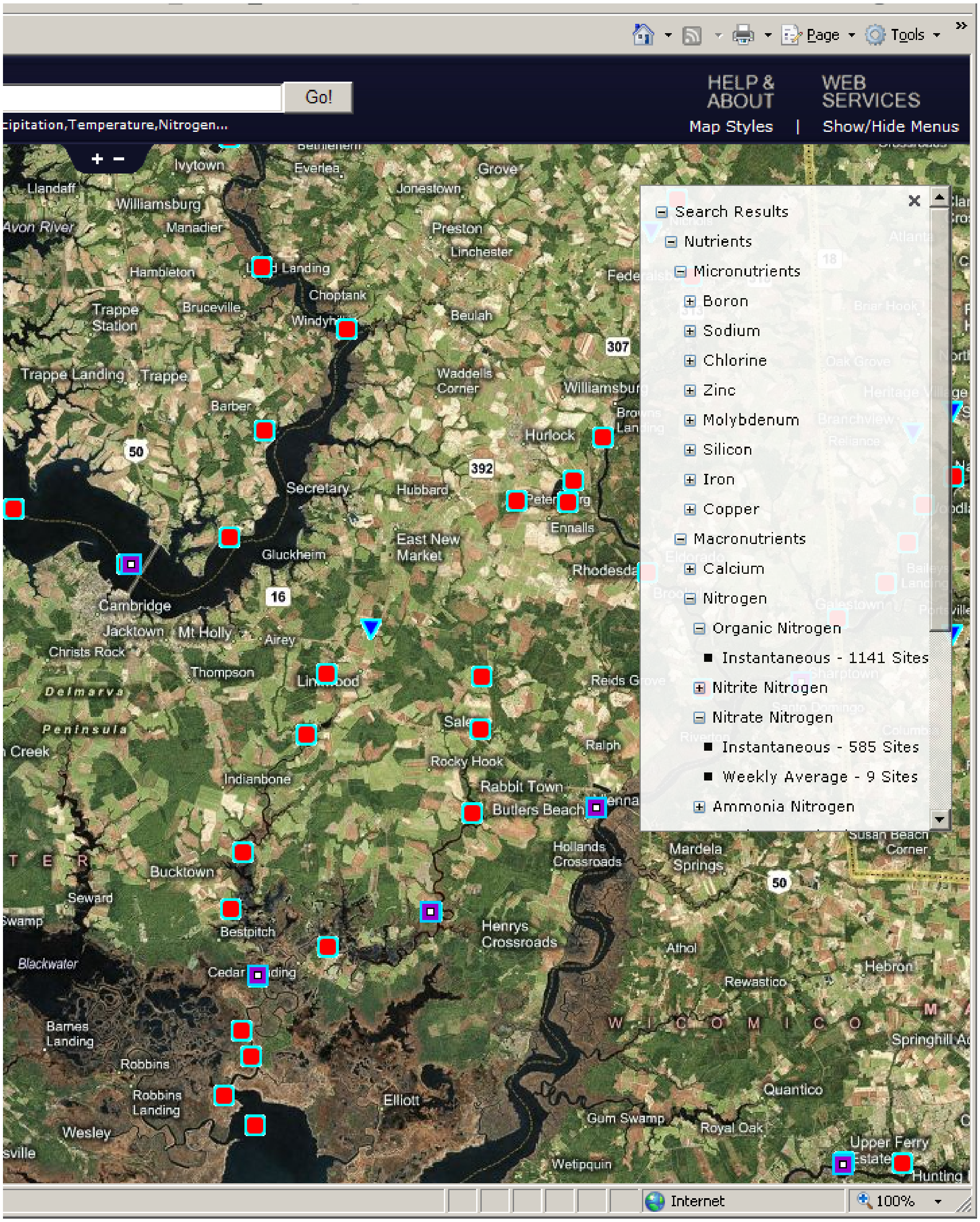
g) It then goes to the MappingsApproved table to identify all VarCodes that are mapped to all previously identified leaf ConceptCodes

h) It then uses the Web Services to find stations and variables that are inside the data cube of time, space, and keyword.

i) It opens the XML file again and fills in for each search keyword previously identified in the search ontology the resulting stations for the selected search region and the time frame.

j) It passes the result XML file back to the GUI which interprets its contents and then prepares to fill the result window in the GUI.

k) It classifies the return results using two approaches. Firstly, it uses the branch structure defined in the search ontology to create an expandable and collapsible tree that can be navigated in the result pane. This structure necessarily ends at the concept leaf level. In order to add another classification level to these results the code uses the entries for DataType and TimeInterval and combines them to another (final) classifier together with a site count, as shown in Figure 3.



- ⊟ Search Results
  - ⊟ Nutrients
    - ⊟ Micronutrients
      - ⊞ Boron
      - ⊞ Sodium
      - ⊞ Chlorine
      - ⊞ Zinc
      - ⊞ Molybdenum
      - ⊞ Silicon
      - ⊞ Iron
      - ⊞ Copper
    - ⊟ Macronutrients
      - ⊞ Calcium
      - ⊟ Nitrogen
        - ⊟ Organic Nitrogen
          - ■ Instantaneous - 1141 Sites
        - ⊞ Nitrite Nitrogen
        - ⊟ Nitrate Nitrogen
          - ■ Instantaneous - 585 Sites
          - ■ Weekly Average - 9 Sites
        - ⊞ Ammonia Nitrogen

Figure 3 Search Result Classification System.

3 HydroSeek Ontology System

A center piece is the development of a concept framework that features concept layers of various specificity for the search. This is rather than to expect someone to know specific parameter codes for successfully querying a number of data sources one by one. These concepts are realized in so-called ontologies that are coded in OWL, a specialized language for ontologies that is based on XML. The OWL Editor Protégé was used to create them. In addition, one of the main reasons to use OWL for encoding, besides the semantic richness its rules allow for, is that that one can deploy a parser of which several are available for free. These are codes that can also be incorporated into JAVA or .NET applications. This represents a significant time savings in terms of coding and functionality.

3.1 A few Ontology Design Rules

When designing an ontology several rules of thumb should be followed (this is just a selection; there are several more concerning the definition of ranges and domains, but those don't necessarily apply here):

a) An ontology needs to serve a specific purpose. Without defining the purpose first there can be no ontology.

b) One needs to realize that in most cases there are several feasible solutions to creating an ontology, i.e., there is no "best" ontology.

c) When creating an ontology and making decisions about how many subclasses or branches to start on one should look at how many subclasses (or instances in class) are generated. One should avoid branches that hold more than about 12 subclasses. If the number goes beyond that then it is probably possible to find another criterion to branch off an additional set of subclasses.

d) On the other hand one should avoid situations in which one class has only one subclass. This suggests a structure that is too granular leading to an extraordinary (and unnecessary) number of branches.

3.2 The Layer System

The ontology system is actually a collection of several ontology layers that fulfill different purposes, as shown in Figure 4.

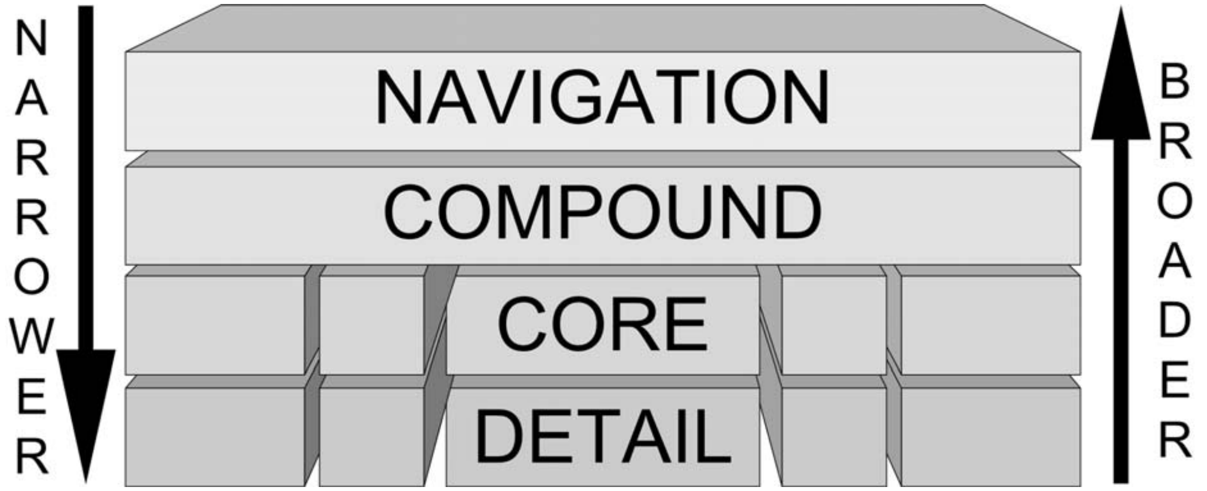


Figure 4 Ontology Layer System.

The top layer "Navigation" provides the backbone of the ontology and contains the collection of broadest terms in addition to being the start point, i.e., hosting the top or start concept, which is "Water Resources". In ontology terms this becomes the *root class*. From this class the ontology branches off into (currently) 5 subclasses, i.e., "Surface Hydrology", "SubSurface Hydrology", "Water/Soil Quality", "Atmospheric Hydrology" and then a "Supplemental" class that collects subclasses that do not necessarily have a conceptual link to any of the other subclasses or that are common to all other branches. In developing ontologies these subclasses should be separated out so as not to repeat them. Members of the "Navigation" layer are not permitted as search keywords because the sheer bulk of the data returns are considered overwhelming and not suitable for reasonable display in the HydroSeek GUI. Table 1 highlights search keyword availability for the ontology layers. Figure 5 shows the complete navigation layer and then selected regions of the "Compound" and "Core/Leaf" layers.

Table 1 Layer properties in terms of permissible search keywords and mappings.

Navigation	No search keywords	No mappings
Compound	Search keywords	No mappings
Core/Leaf	Search keywords	Mappings
Details	No search keywords	No mappings

The basic rationale for structuring these layers is that each layer has different properties in terms of offering or containing permissible search keywords and whether or not it can be mapped to. In Figure 5, notice that the navigation layer also contains the subclasses for “Physical”, “Chemical”, and “Biological Parameters”, which means that each general layer (“Navigation”) does not necessarily contain an evenly distributed or equal number of classes or subclass layers on each branch. For example, while the three subclass branches of “Soil/Water Quality” are considered part of the navigation layer, subclasses “Ground Water Flow” and “Water Level” which are directly attached to “Subsurface Hydrology” are not.

Figure 5 Color coding to highlight Layers in the Ontology System.

In fact, subclass “Ground Water Flow” does not belong to the next layer “Compound” layer either as it is part of the “Core” or “Leaf” layer already, i.e, it is a concept that can be mapped to. The subclass “Water Level” however is part of the “Compound” layer because it cannot be mapped to. In essence, there can be segments in the ontology structure that skip one layer and go directly from “Navigation” to the “Core/Leaf” layer as is the case in the previous example.

3.3 NITROGEN AS EXAMPLE

The layer system is perhaps best demonstrated using “Nitrogen” as shown in Figure 6. In this figure it also becomes clear that the “Detail” layer is not really part of the ontology system, i.e., there is no OWL file. Rather, it is the layer that contains the variable codes from each individual network that are mapped to the leaves in the “Core” layer. For example, there are currently (March 2008) 29 nitrate variable codes coming from all registered networks mapped to the leaf “Nitrate” in the “Core” layer. Notice that “Organic N” belongs to the “Compound” layer because it is not a leaf that can be mapped to.

In the above figure “Nutrients” is the most general (highest) permissible search concept that a user can query for, while any one of the leaf concepts is the most specific. Entries from the “Navigation” layer are not permitted because of the low-precision-high-recall problem, while entries in the “Detail” layer are not permitted because of the high-precision-low-recall issue (this is equivalent to having to know what the respective variable codes are in each of the networks).

Nitrate however is not the only permissible keyword for Nitrate. The keyword vocabulary contains additional Nitrate terms based on entries in the medium table. For example, when using the auto-complete function in the search interface there will be 4 different keywords associated with Nitrate

- Nitrate Nitrogen
- Nitrate Nitrogen (Sediment)
- Nitrate Nitrogen (Water)
- Nitrate Nitrogen (Soil)

This level of granularity however, is not yet fully developed and only exists for selected variables (like Nitrogen) as the result of earlier demo efforts. The utility of these additional classifications has not yet been tested and may turn out to be superfluous. This should be assessed via a review conducted by, for example, test bed managers.

3.4 Auxiliary Functions

The ontology system is actually a collection of several ontology layers that have different purposes. For example, there are a number of other ontologies that fulfill auxiliary functions. One of them are a number of “sync” ontologies that define the synonyms that exist for a certain number of keywords, in this case “flowRateGroundWater” and “dischargeGroundWater”, as shown in the text box below. These equivalence statements permit the user to use either “Flow Rate, Ground Water” or “Discharge, Ground Water” as a permissible search entry both ensuring the return of the exact same data sets.

```
<owl:Class rdf:ID="flowRateGroundWater">
 <rdfs:subClassOf rdf:resource="http://www.cuahsi.org/navigation#subsurfaceHydrology"/>
 <rdfs:label xml:lang="en">Flow Rate, Ground Water</rdfs:label>
 <owl:equivalentClass>
  <rdf:Description rdf:about="http://www.cuahsi.org/flow#groundWaterFlow">
   <owl:equivalentClass rdf:resource="#flowRateGroundWater"/>
   <owl:equivalentClass>
    <owl:Class rdf:ID="dischargeGroundWater"/>
   </owl:equivalentClass>
  </rdf:Description>
 </owl:equivalentClass>
</owl:Class>
```

These smaller ontologies are loaded together with the much larger layer ontologies into a final “search” ontology that comprises the underlying structure. The file **search.owl** is the backbone of the search ontology; however, it does little more than include all sub-ontologies to form a single file, as shown in the text box below. This file is accessed by the search engine when resolving the keyword search and drilling down to the leaf levels to find the ConceptCodes that lead to the variable codes in the ODM. While most entries into the ontologies are straightforward class definitions without any domain or range restrictions, some of the classes actually do have restrictions placed on them. These restrictions, mostly “single parent” restrictions, are necessary because the class itself does not have a unique definition and could be placed elsewhere in the ontology. A restriction to a certain parent class ensures that a given subclass cannot be confused (or moved) to any other class but the identified parent class in effect making it a unique class with a unique location inside the ontology.

```
<?xml version="1.0"?>
<rdf:RDF
    xmlns:wq-ext="http://www.cuahsi.org/waterquality-extended#"
    xmlns:nwis="http://www.cuahsi.org/nwis#"
    xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
    xmlns:owl="http://www.w3.org/2002/07/owl#"
    xmlns:met="http://www.cuahsi.org/meteorology#"
    xmlns:nav="http://www.cuahsi.org/navigation#"
    xmlns:wq-syn="http://www.cuahsi.org/waterquality-syn#"
    xmlns:wq="http://www.cuahsi.org/waterquality#"
    xmlns:flow="http://www.cuahsi.org/flow#"
    xmlns:epastoret="http://www.cuahsi.org/epastoret#"
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:met-ext="http://www.cuahsi.org/meteorology-extended#"
    xmlns:met-syn="http://www.cuahsi.org/meteorology-syn#"
    xmlns:flow-ext="http://www.cuahsi.org/flow-extended#"
    xmlns:flow-syn="http://www.cuahsi.org/flow-syn#"
    xmlns="http://www.cuahsi.org/search#"
  xml:base="http://www.cuahsi.org/search">
  <owl:Ontology rdf:about="">
    <owl:imports rdf:resource="http://hydroseek.org/search/ontology/meteorology.owl"/>
    <owl:imports rdf:resource="http://hydroseek.org/search/ontology/flow-syn.owl"/>
    <owl:imports rdf:resource="http://hydroseek.org/search/ontology/meteorology-syn.owl"/>
    <owl:imports rdf:resource="http://hydroseek.org/search/ontology/waterquality-syn.owl"/>
    <owl:imports rdf:resource="http://hydroseek.org/search/ontology/waterquality-extended.owl"/>
    <owl:imports rdf:resource="http://hydroseek.org/search/ontology/flow.owl"/>
    <owl:imports rdf:resource="http://hydroseek.org/search/ontology/waterquality.owl"/>
    <owl:imports rdf:resource="http://hydroseek.org/search/ontology/navigation.owl"/>
    <owl:imports rdf:resource="http://hydroseek.org/search/ontology/flow-extended.owl"/>
    <owl:imports rdf:resource="http://hydroseek.org/search/ontology/meteorology-extended.owl"/>
  </owl:Ontology>
</rdf:RDF>
```

4 General Comments on HydroSeek

4.1 Future Additions

The current capabilities of HydroSeek are fairly comprehensive, making it a usable system. However, during the beta testing period it became clear that the search engine is in need of a number of upgrades and additions. While the email notification system problem encountered in late 2007 has been fixed together with a number of smaller glitches (like the size of the Where?When? panel and the visibility of the lower parts on screen with less resolution) several issues remain that need short or medium term attention. More specifically these are:

a) The MicroSoft Virtual Earth map control needs to be updated so web browsers other than Internet Explorer can be used with HydroSeek. This applies to Firefox running on Windows but also browsers on other operating system like MACs. This will be addressed by integrating version 6 of the Virtual Earth map control.

b) The system currently does not provide metadata information neither when working on the interface nor on the data deliveries (even though a small minimum set is included in the header of the data file). There is a specific metadata Web Service available at
http://cbe.cae.drexel.edu/wateroneflow/MetadataForm.aspx
that could be harvested after inclusion into the GUI.

c) The system currently does have a means to supply citation and notification services. These services would include automatic email notifications sent to the provider of data each time a third party downloads it. There is also the need to supply citation wording for those that download third party data so it can be properly acknowledged.

d) HydroSeek does not have a plotting feature in which a preliminary assessment could be conducted. There are several options for this feature to be included, one of which is the inclusion of the plotting module used inside the ODM Tools application; another is to use a separate plotting application to be purchased from a third party vendor.

Current search capabilities are focused on Bounding Box or HUC code (spatial) and time brackets (temporal) together with a keyword search that groups hydrologic terms in a hierarchical fashion. While this is an intuitive way of organizing the hydrologic realm, there are other ways of organization, one of which is via processes. For example, if one defines a hydrologic process called "Runoff", then one could group or register datasets that are relevant for this process, like rainfall (intensity and duration), slope (topography), soil characteristics, and land use (for roughness calculations) to have all the data available to compute a hydrograph for a specific basin.

This "process" ontology would bring a higher degree of meaning to the datasets because they are now grouped along processes and thus are being given a scientific context. It would be implemented next to the current keyword search feature and could be accessed as an alternative to the default keyword search. The timeline for this extension is long term and in all likelihood will constitute the PhD work of a graduate student.

4.2 Access to HydroSeek

HydroSeek can currently be accessed at two sites:

a) http://www.hydroseek.net at the San Diego Supercomputer Center
b) http://www.hydroseek.org at Drexel University

Both sites use the same code base (which can be obtained on the Source Control system at SDSC also); however they do not (currently) operate of the same ODM installation. Deciding on the approach to synchronize the underlying database contents is one of the immediate tasks that needs addressing by HIS.

Future plans have addressed the need to provide a better production type service that ensures that HydroSeek is up and running 24/7 with a desirable 99.99 up time. This is typically a task that only larger and dedicated IT departments or institutions can afford. To this end, the primary production site will be established at SDSC with a mirror site established at the Center for Research in Water Resources (CRWR) at the University at Texas in Austin. The CRWR site will be a true mirror site of the one at SDSC, i.e., updates on code and DB will be transferred from SDSC to CRWR, thus effectively giving SDSC the lead on the HydroSeek operations.

Other copies of HydroSeek may be downloaded at any time from the Source Control system at SDSC and established elsewhere for both local installation and research purposes. However, the official HydroSeek installations supported by CUAHSI will be those at SDSC and CRWR.