Final Report to Arizona Water Institute  
AWI-07-15 AHIS Enhanced Data Availability Using Web Services Technology  
And  
AWI-07-63 Database Connectivity Component for the AHIS

Abstract

This report assesses the progress on Arizona Water Institute (AWI) projects AWI-07-15, Enhanced Data Availability Using Web Services Technology and AWI-07-63, Database Connectivity Component for the AHIS. The primary goal of AWI-07-15 is to develop cyber infrastructure for three aspects of the Arizona Hydrologic Information System (AHIS). The objectives of this project are to complete three proof-of-concept modules that:

- make available on-line the Arizona Department of Water Resources (ADWR) Flood Warning System data (currently archived by the Salt River Project (SRP)) and selected water data hosted by the Arizona Department of Water Quality (ADEQ);
- improve the AHIS search interface through controlled vocabularies and semantic mediation; and
- develop web user interfaces to search, subset, visualize and obtain the data in a variety of formats using the latest web application technologies.

The primary goal of AWI-07-63 is to design and implement an extensible database connectivity component for AHIS’s Metadata Catalog Manager. The component will capture metadata in an XML-based format, it will store the metadata directly into a Native XML Database (NXD), and it will allow for easy translation between different metadata standards (three for the purpose of this project).

The team has met the following milestones. Detailed discussion of progress is found in the body of the report, which follows.

Milestone # 1: Team kick-off meeting was held on Thursday, November 30th with the following participants:

Aguilar, Raul, ASU, Andrzeoczak, Kathryn, AZDEQ, Callaway, Steve, ADEQ, Carpenter, Kyle, SAHRA/UA, Davulcu, Hasan, CSE/ASU, Gries, Corinna, GIOS/ASU, Haffer, Tony, NWS, Hood, Wayne, ADEQ, Jackson, Randy, ADEQ, Jacobs, Kathy, AWI, Johnson, Mike, ADWR, Jordan, Darrell, ADWR, Kimmel-Guy, Alethea, ASU, Liu, Huan, CSE/ASU, Mason, Dale, ADWR, Otte, Wolf-Dieter, NAU, Patterson, Ken, SRP, Spitz, Anna, AWI, Vazquez, Ramon, SAHRA/UA, Woodard, Gary, SAHRA/UA.

Milestone # 2: Release of V1.0 Web Services was accomplished on May 20th. The web services may be viewed here:


and search interface may be viewed here:

Although not funded under this proposal, V0.1 of the AHIS meta data editor may be viewed here:


**Milestone #3:** Common Java API client to query web services

The new search interface can be viewed here:
http://gdws.asu.edu:8080/ecologyse08
Final Report

1. Bringing databases online: Development of web-services to access ADEQ ground water database:

Completed tasks

a) Learn data model and data transfer to local server
   Learned the data model of the groundwater data obtained from ADEQ. Ported the database from MS Access to MySql.

b) Develop web services
   Designed the data access module for the water data based upon the standard web services defined by CUAHSI (Consortium of Universities for the Advancement of Hydrologic Science). Prepared a mapping of the water data to the web service objects in consultation with experts from ADEQ. Developed the access module for the ADEQ data. The module exposes the following four web services as defined by CUAHSI.
   i. getValues
      Given a site number, a variable, a start date, and an end date, this method returns a time series.
   ii. getSite
      Given a site number, this method returns the site metadata.
   iii. getVariableInfo
      Given a variable code, this method returns the variable metadata.
   iv. getSites
      Given an array of site numbers, this method returns the site metadata for each one.

   The web services were developed in Java and are based on JAX-WS 2.0 standard. JAX-WS has inbuilt support for SOAP and XML technologies. Hibernate, an object-relational persistence and query service is used for data access. Though currently the database used is MySql, the use of hibernate allows for easy portability to other databases by modifying XML configuration files. The services are deployed on Apache Tomcat web server.

c) Develop client to test web services
   A JSP client to test the web services was also developed. This client is deployed on Tomcat web server. (see address above)

d) Go through steps a – c for SRP flood warning database
   SRP was contacted in May regarding collaboration to make the historic data stored in the flood warning system available to AHIS. Several meetings and e-mails followed, however, in the end the project could not be completed due to incompatibility issues between the programming languages used, which could not be resolved due to time constraints the SRP programmer had. When the first signs of potential problems appeared we decided to devote our programming efforts to a different dataset, one which would allow us to parse the data into the Observation Data Model (ODM), the standard data model for storage developed by CUAHSI. Webservices were developed to access data in ODM format which now allows us to access all data that are stored in this
standard. The data set chosen was the historic climate data for Arizona which are made available by NOAA for several reasons. This is a dataset highly sought out by scientists and difficult to access on the NOAA website, it is finite and will not need any maintenance.

e) Additional accomplishments: Parsing ADEQ water quality data supplied in XML format into the ODM.

As outlined in the interim report ADEQ is now providing its data in an XML format required by EPA. A java program was written to automatically parse these data into the ODM (see above). Because webservices to access data in ODM format were developed as well, we are expecting the ADEQ data to be easily accessible as well. However, this is still in progress.

Schedule challenges

Obtaining and learning of the data model took longer than anticipated. Communication was slow due to different backgrounds and expectations of participating parties. We learned that this particular phase needs more time scheduled in the future.

According to the information we received SRP was restructuring their data storage system but was intending to make data available to us via a portal that was programmed by SRP. Therefore, our programming efforts were not to access the databases directly but through this portal. However, this portal returns data in proprietary Microsoft format and does not conform to web service standard, creating an incompatibility issue that needed resolving. During the project SRP saw some personnel turnover. Efforts have been made since to address the incompatibility issue. However, time constraints at SRP have delayed the progress considerably.

2. Develop custom client application in collaboration with stakeholders

Completed tasks

a) Determine functionality with stakeholders

A first client demonstration was created among the technical group. This site was presented to a group of stakeholders on March 22.

From this presentation perhaps the main feedback suggested by stakeholders was the disclosure of quality of the data that users will be downloading from the website. Stakeholders agreed that a mechanism to better identify data aspects would have to be provided by this client. For instance, some data are never inspected for QA/QC (Quality Assurance/Quality Control), which is the main reason stakeholders are suggesting that we prominently disclose these aspects of the data to the end-user.

The solution to this problem is to record these notes in the metadata editor, and show this metadata information whenever a user downloads data from the website.

b) Evaluate web technologies

The web service client website will contain open source technologies so that this system is as portable as possible among institutions. For this reason it was decided to use the following technologies to support this system:
Presentation (End-user) Layer
   HTML, XHTML, and images
   JSP, JavaServer Faces, and MyFaces Trinidad pages for forms, error pages, and actions
   Declarative configuration of page-flow sequence using Java Server Faces
   Google Maps for end-user geographic condition creation
   Javascript and AJAX

Session Container Layer
   Apache Tomcat (6.0)
   Managed Beans
   JNDI Lookup (for database access)

Data Access Layer
   Java Persistence API (JPA)
   Java DataBase Connectivity (JDBC)

Database
   MySQL (5.0) and corresponding JDBC Driver (Including Spatial Extension which supports OpenGIS Simple Features Specification)

Web Services
   Apache Axis for Simple Object Access Protocol (SOAP)

c) Decide on framework to be used, write code

Certain data are expected to be relatively static and therefore are cached locally in order to reduce the number of round-trips to a web service. For example, a GetSiteInfo request consists of a given site location identifier, and the response that corresponds to that request, consists of CUAHSI’s Observations Data Model (ODM) variable identifiers, for the variables associated with that site. This particular type of request is expected to be made very frequently to populate dynamic forms on the end-user interface, and as a preliminary query before specific queries for a given site or variable. The same can be done for GetVariable and GetVariableInfo requests.

Detailed queries like GetValues are not be cached in this manner, although these could be considered for data whose Variable Info indicates it is infrequently updated by the source. Variable Info responses tend to have much more variety than Site Info responses, and many such responses represent real-time data.

All queries are made online, that is, when the user requests them. In the future, some features need to be added to be able to handle queries that will take a longer time than what the end-user expects in a single session.

Code has been created to cache GetSiteInfo, GetVariable and GetVariableInfo which data will be stored in a MySql database.
The caching module was created using ADEQ and CUAHSI’s GWSI groundwater web services. Since SRP’s web services will be created using the same CUAHSI’s standards, their addition to the list of data providers can be executed as soon as they are available.

d) Web services client
A Java-based client was created and is available for end-users to query available web services and their data. This module has the ability to integrate any CUAHSI-compliant web services and present them using a single programming interface. This functionality will allow any programmer to create interfaces and other modules that can enhance the data discovery under AHIS. For now, the caching mechanism works for ADEQ groundwater data, and CUAHSI’s GWSI groundwater data.

In the future, this module needs to be modified so that it can receive filtering criteria in the form of Java objects. These objects would then be transformed into a web service query that fetches the data from each of the web services selected by the end-user. The data would then be retrieved from each of the web services using the standard data access methods and presented in Java format.

e) Query Handler Module (started)
A Query Handler Module was initiated to improve the data retrieval capabilities of the Web services client described before (see d.) This module’s objective is to present the end-user with the data (sources and variables) available through AHIS. This module will also filter these data to better depict what the user is looking for.

This module consists of a web-based interface with different components to filter queries. These components include available data sources, and their associated variables; and geographic and temporal constraints. The output from this module will be the actual data that the user is looking for. In the near future data will be presented in text-based format. In the longer term the output can also be presented in other formats. The Query Handler Module will be completed during the first trimester of 2008.

3. Develop the search interface

Completed tasks

a) Gather controlled vocabularies, thesauri, dictionaries relevant to water subject

In data mining, the data preprocessing and data cleaning steps are very important. Preprocessing data affects the quality and the efficiency of the data mining algorithms.

In our problem, we obtained an average of around 5000 terms contained in titles and keyword lists. If we take into account terms in existing ecological metadata files description of attributes, bibliography titles, the number of terms is about 25000. We would like to reduce the dimension of the term-document matrices to an, empirically determined, N dimensions. We construct such sets of terms by first computing strongly correlated term sets. We compute such sets by co-occurrence clustering.
b) Develop ontologies in collaboration with CUAHSI

The data gathering for a geographic ‘Arizona water bodies’ ontology was accomplished, however, formal representation in ontology language (OWL) is not yet finished.

c) Develop interface to collect search terms from users

In June a web-based search engine prototype was implemented. The software consisted of two components: The indexing engine that does the dimensionality reduction, correlation computing and retrieval engine that obtains the query, matches the data files and showcases the result.

Each obtained cluster contains highly correlated terms and the cardinality of each cluster is much smaller compared to that of the whole collection of terms.

For each cluster, we built a hash-table. If the search term is in a hash-table, we build a vector of elements in that table and compute the cosine similarity between this vector and documents in the collection. Note that each document corresponds to a metadata file.

d) Evaluate search terms and integrate into gathered information

Continuously ongoing as new metadata files are added and more sources for relevant keywords are incorporated.

e) Improve interface as information becomes available

The current application allows new metadata files to be added. They are automatically mined for key terms, the user is presented with the terms for editing (choosing the relevant and discarding the useless ones), and then the system recalculates the similarity matrix and re-indexes the terms for better search performance. Therefore, the search engine will become richer and more versatile with every metadata file added to the database. The programming was accomplished in Java and jsp, the metadata files are stored, managed and accessed in a native XML database.

4. Develop a metadata transformation module

The goal of this project (AWI-07-63) is to provide a convenient means to programmatically translate a metadata document from one standard format to another standard format (e.g. from EML to NBII).

Completed tasks

- A framework has been designed, implemented and tested, that allows for metadata translations, under the assumption, that the translation is totally controllable by a style sheet.
- The module has been designed as a number of POJOs (“plain old Java object”), using the TrAX API (transformation API for XML).
• The module’s behavior is solely controlled by an XML configuration file that is provided by
the module’s user.

This file contains information about the metadata types that can be handled (regarded as nodes
in a directed graph) as well as information on the available style sheets (which are the edges
between the nodes, i.e. represent elementary transformation paths). Out of these pieces of
information complex transformation paths can be constructed and applied to inputs, that
potentially span any two nodes, as long as there is a path of elementary transformation paths
leading from one to the other. The configuration file can be read during runtime, so that it can
be located anywhere and provided any time prior to executing transformations. The
construction of the transformation paths is done using Dijkstra’s shortest path algorithm.

• For convenience, the available style sheets can be located anywhere in a network, e.g. in a
style sheet repository. The way style sheets are retrieved can be configured very flexibly by
providing any data retrieval protocol on the fly, i.e. “file”, “ftp”, “xml:db” for accessing eXist
etc. Unknown protocols can be used as well, as long as there is a
URLStreamHandler/URLConnection pair of Java objects being made available.

• Not all transformations are as simple as just applying a style sheet. When transforming a
document from a less to a more elaborate format, information needs to be supplemented to fill
in the gaps. This implies some sort of user interaction. The solution implemented in the
module is to automatically perform a schema validity check against the target format,
programmatically triggered by the program this module will be embedded in. The resulting
information about the schema compliance can then be used to decide whether further actions
are needed.

• The project’s code has been integrated to work in the context of the metadata editor, see above.

• The project’s code can be checked out from this code repository:
svn://flagstaff.cse.nau.edu/MetaDataTransformation/trunk

The link retrieves a NetBeans project, which contains an example that runs right out of the
box. It showcases the dynamic capabilities of the module (specifically the loading of a
communication protocol on the fly, which in turn retrieves a style sheet from an eXist DB,
which in turn processes an example transformation).

5. Presentation of Final Product at Year-end AWI Meeting or Other Forum
The progress of AHIS has been presented throughout the year at various meetings as oral presentation:

March 2007: Arizona Department for Water Resources
May 2007: Arizona Department for Water Resources
October 2007: Regional CUAHSI Conference, Austin TX
October 2007: Navajo Nation Hydroclimatic Data Workshop, Flagstaff

Posters:
Aguilar, R. & C. Gries, AHIS Metadata Editor, LTER Information Manager annual meeting, San Jose,
CA.
Nguyen, H., C. Gries, H. Davulcu, AHIS Dataset Search Engine, LTER Information Manager annual meeting, San Jose, CA.
Aguilar, R., C Gries, I San Gil Another Piece for the Metadata Management System, CAP-LTER annual poster symposium
Nguyen, H., C. Gries, and H. Davulcu, AHIS Dataset Search Engine: An intelligent approach to EML Data Management, CAP-LTER annual poster symposium

And a paper for the proceedings has been submitted.

6. Personnel Changes

The project was initiated with Kyle Carpenter as the Principal Investigator at the University of Arizona (UA). Kyle left UA in March 2007. Ramon Vazquez assumed Kyle’s responsibilities on this project. James McGill joined the project at UA in March 2007.

7. Concluding Remarks

Despite personnel changes and data structure changes by the agencies the AHIS projects (AWI-07-15 and AWI-07-63) have accomplished most activities proposed in the Memorandum of Agreement. One data set could not be accessed, however, it has been substituted with another. In addition to proposed activities, we explored CUAHSI’s Observation Data Model and developed access modules, now being able to access all data in this standard format.