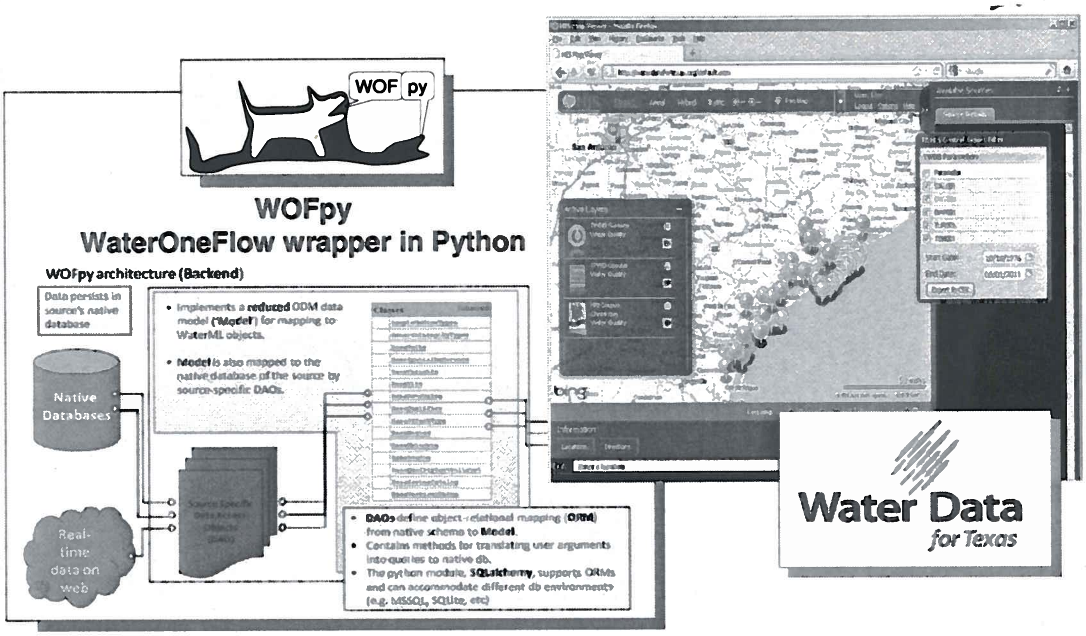
CONTINUAL DEVELOPMENT OF A COASTAL GEODATABASE FOR TEXAS (TWDB Contract#: 1104831137)

Final Report



Texas Water Development Board

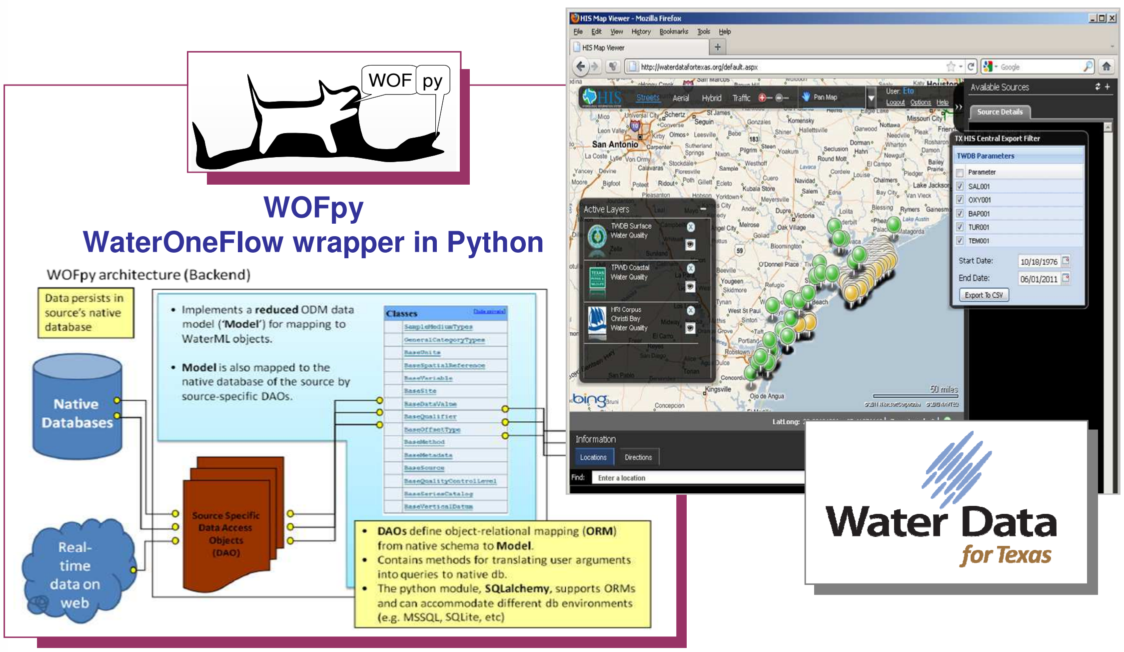
Dec 22, 2011.

Project No. 11016.00

RPS Espey

CRWR

CONTINUAL DEVELOPMENT OF A COASTAL GEODATABASE FOR TEXAS (TWDB Contract#: 1104831137)



Texas Water Development Board

Dec 22, 2011.

Project No. 11016.00

TABLE OF CONTENTS

1.0 EXECUTIVE SUMMARY

RPS-Espey, along with its subconsultant the Center for Research in Water Resources (CRWR) at University of Texas at Austin, was contracted to develop a web service framework to publish data for the Texas Water Development Board's ***Water Data for Texas*** information system. The participants in this project are Mr. James Seppi (CRWR), Dr. Ernest To (RPS-Espey), and Dr. Tim Whiteaker (CRWR).

The resulting product, ***WOFpy*** (WaterOneFlow in Python), is an open-source, cross-platform toolkit written in the Python programming language. ***WOFpy*** enables State agencies and other collectors of environmental data to publish their data to the web using CUAHSI's (Consortium of University for the Advancement of Hydrologic Science) WaterML data language. As such, ***WOFpy***-published data can be retrieved directly into popular desktop applications such as Microsoft Excel and ArcGIS using CUAHSI's open source data clients. In addition, ***WOFpy*** web services can be registered at CUAHSI's HIS central to support intelligent searching and discovery of data across multiple data sources. Such capabilities greatly ease the compilation of environmental data to support scientific and engineering work in Texas.

WOFpy's key strength is that it can be customized to a large variety of databases, thereby making it adaptable to the existing systems of Texas State agencies. It has the ability to read and access many data formats ranging from simple ASCII files to enterprise-level databases (e.g. Microsoft SQL Server). Moreover, ***WOFpy*** allows the user the freedom to retain his data in their original structure instead of organizing them into a preset database schema. To translate data from the native format into WaterML, ***WOFpy*** utilizes data-access objects (DAOs) – which are user-customized Python packages – to link native data objects to WaterML objects at run-time. Several examples of DAOs are installed with ***WOFpy*** to provide users with references on how to build and customize these DAOs. These examples cover a range of database systems – from the sophisticated CUAHSI Observation Data Model (ODM), to basic three-table SQLite databases, and finally to collections of comma-delimited text files. Software documentation and tutorials were created by the developer team to provide additional guidance to the user.

This report provides a summary of the products created through this effort; and, the activities undertaken to promote the use of ***WOFpy***. ***WOFpy*** and other ***Water Data for Texas*** Python modules are being released as open-source software under the BSD license so that Python-users and the broader scientific community may benefit from their continual development and use. ***WOFpy*** is registered on Python Package Index (PyPI) at http://pypi.Python.org/pypi/***WOFpy***/0.1.1-alpha. Its latest source code is available on Github at https://github.com/swtools/***WOFpy***.

2.0 BACKGROUND

The Texas Water Development Board (TWDB) maintains numerous data sets covering all of the major and minor bay systems along the coast and many of the State's river basins, extending back several decades. To support a host of research and analysis activities stemming from recent state legislation on environmental flows (Senate Bill 3), the TWDB is developing ***Water Data for Texas***, a single system to deliver water-related data from state, federal, university and local sources across Texas in a timely fashion with appropriate metadata and quality control.

Water Data for Texas is built upon a service-oriented architecture which inherits the Hydrologic Information System (HIS) technology developed by the Consortium of Universities for the Advancement of Hydrologic Science, Inc (CUAHSI). It is mutually compatible with the tools and data web services developed by CUAHSI for federal sources such as USGS and EPA. To provide access to Texas-centric water data, the TWDB is forming partnerships with Federal and Texas agencies to adopt HIS technology. In addition, the TWDB is building additional components for ***Water Data for Texas*** to provide ease of use and quality control to its partners and clients seeking to publish and use water-related data

A key technical barrier to adoption of HIS technology is that building a custom implementation of CUAHSI WaterOneFlow web services is a non-trivial endeavor (Pothina and Wilson, 2011). It requires an understanding of the web services, XML, and the particulars of the WaterML and WaterOneFlow. Hence, the old paradigm (see Figure 2.1) followed by most participating data providers is to manipulate their data into an ODM database hosted on an MSSQL server. CUAHSI-HIS has a prebuilt generic WaterOneFlow implementation that can then be used to serve data. This approach requires that the data provider either adopt the sophisticated Observations Data Model (ODM) (Horsburgh, 2008) as their internal structure for storing data or they must build a translator and periodically dump data from their in-house database to the ODM database on a regular basis. The ODM schema is designed to hold data from multiple sources and hence is often much more complicated than most data provider's in-house database schemas. It also excludes data providers that use non-Microsoft operating systems.

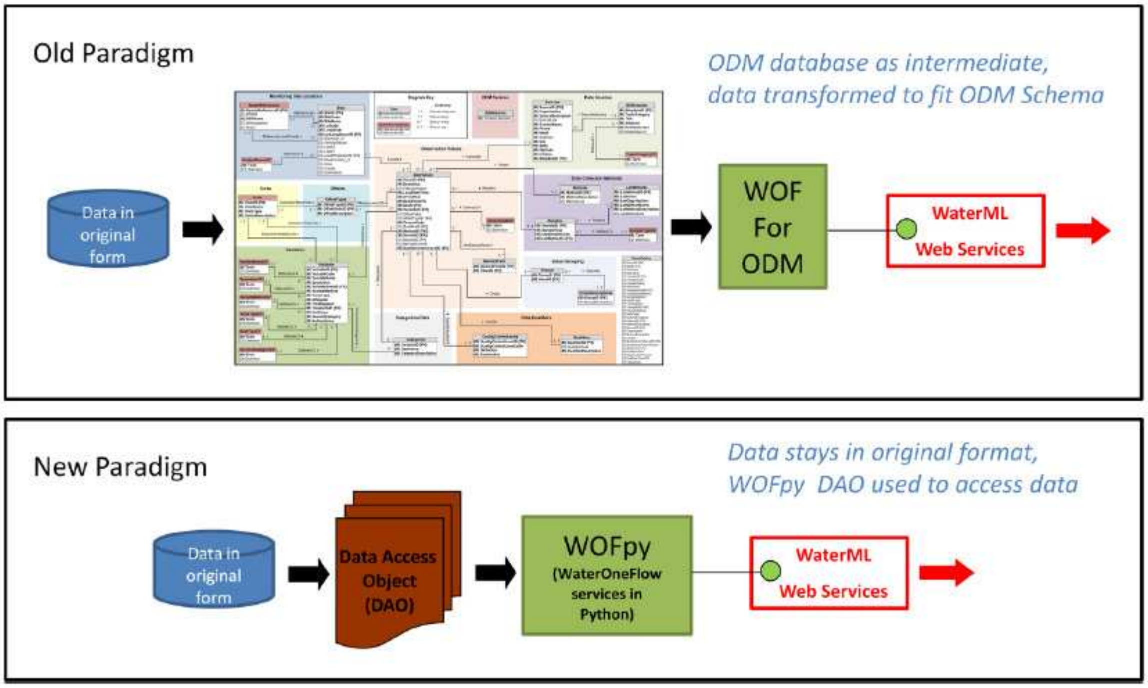


Figure 2.1 Old and New Paradigms to publish environmental data as WaterML Web Services (Pothina and Wilson, 2011).

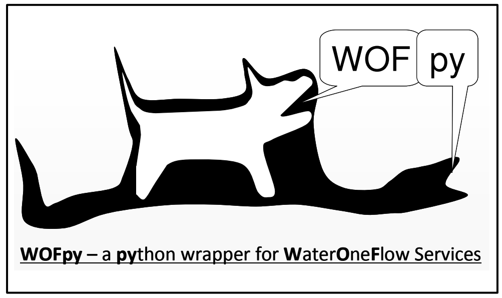
To lower the barriers to implementation of HIS, a new paradigm is proposed where data sharing will no longer require large changes to an organization's internal data systems (Pothina and Wilson, 2011). To bring forth this new paradigm, the following requirements need to be met for data-publishing software:

1. The software should be cross-platform and not be limited to Microsoft applications and operating systems. A good approach is to develop HIS software in Python – a free, cross-platform, stable and highly versatile programming language that is becoming increasingly popular among environmental researchers and professionals. Python also possesses a wealth of features resulting from an ever-expanding library of user-contributed modules and packages.
2. The software should be flexible enough to read from a large variety of databases – ranging from simple ASCII files to enterprise-level databases (e.g. Microsoft SQL Server).
3. The software should allow the user the freedom to maintain his data in its original schema instead of organizing them into a preset database schema. Translation of the data into WaterML should be carried out only during run-time using special interpreters that query data and metadata from the native database schema and use them to populate WaterML objects.

WOFpy is a designed with the above requirements in mind. Its goal is to provide users with a generic customizable web services framework to publish data in WaterML. Detailed description of ***WOFpy*** is given in the next chapter.

3.0 *WOFPY* (WATERONEFLOW IN PYTHON)

3.1 *WOFPY* FRONTEND



WOFpy stands for WaterOneFlow in Python and is a fully-customizable WaterML web service framework developed in Python 2.7. The front end of ***WOFpy*** is the same as any full-fledge WaterOneflow web service and supports the five major data query methods i.e. GetSites, GetVariables, GetSiteInfo, GetVariableInfo and GetValues. However, in addition to being support these queries through the SOAP (Simple Object Access Protocol) web service protocol, ***WOFpy*** can handle queries in REST (Representative State Transfer) web service protocol as well. This means that ***WOFpy*** services can be accessed directly through a web browser URL without the need of an extra SOAP client.

The publishing of data througis is done through the use of the Flask and SOAPlib python packages. Futhermore ***WOFpy*** can publish the results from data queries in both WaterML 1.0 format and the prototype WaterML 2.0 format.

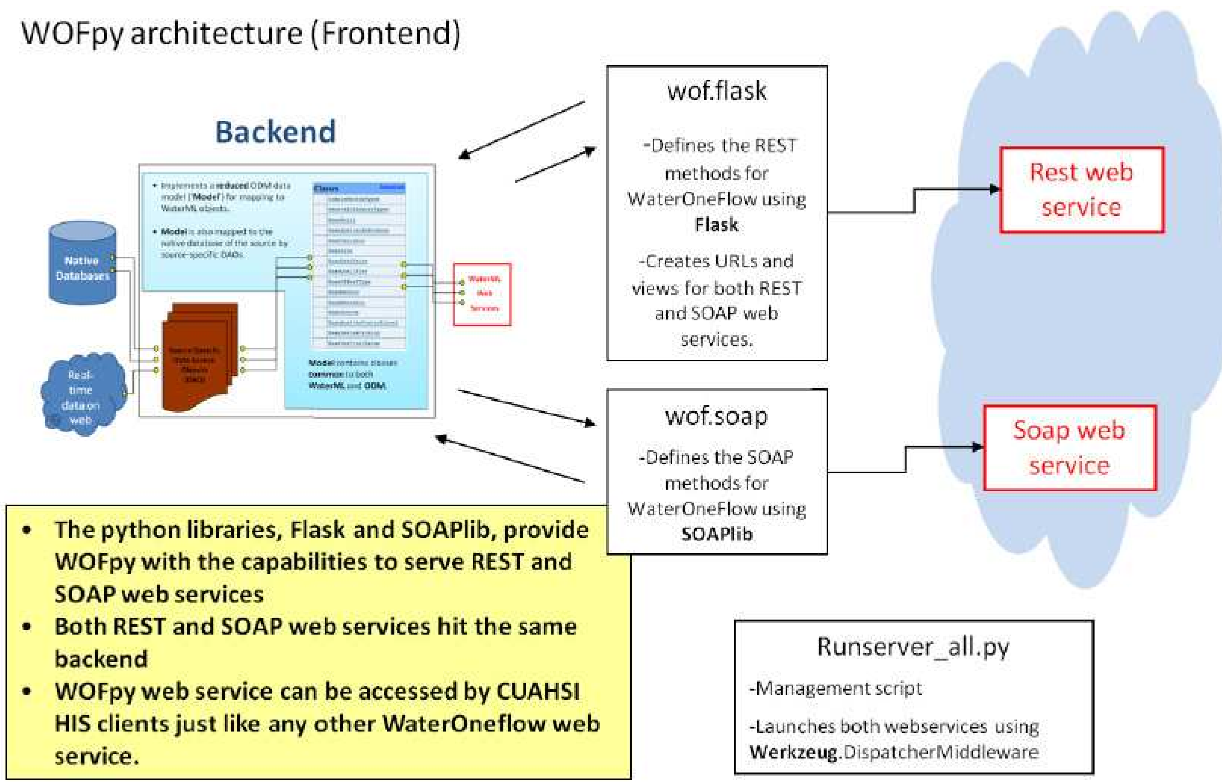


Figure 3.1 Frontend architecture of *WOFpy*

3.2 *WOFPY* BACKEND

On the backend, ***WOFpy*** implements a reduced ODM data model (simply called the "**model**" in ***WOFpy***) that maps to WaterML objects. The **"model"** acts as an intermediary between the user's native database and WaterML (see Figure 3.2). Object relational mappings (ORMs) define how the user's native database schema maps to the "**model**" and in turn how the "**model**" maps to WaterML (see Figure 3.2).

The ORMs between the "**model**" and the user database are defined in python packages called Data Access Objects (DAO's). Through the use of the SQLAlchemy python package ***WOFpy*** can attach to any database backend that SQLAlchemy supports. This allows a large degree of flexibility in attaching the web services to disparate data sources.

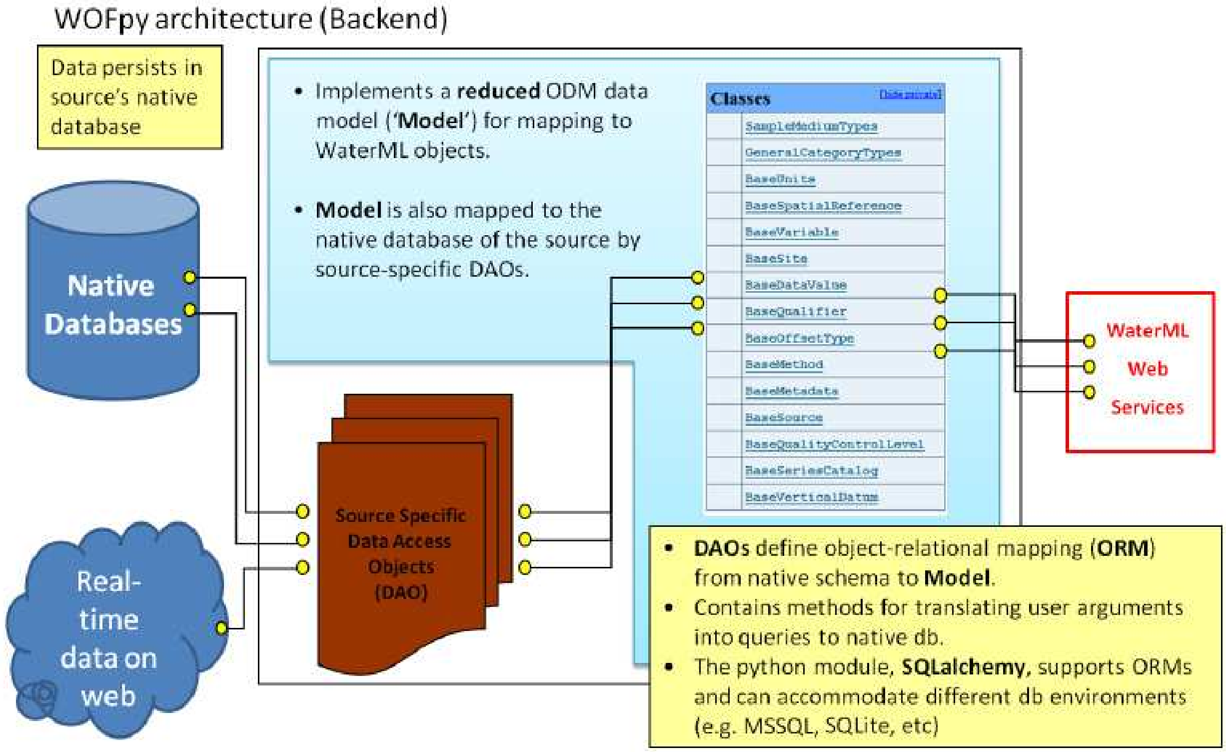


Figure 3.2 Backend architecture of *WOFpy*,

3.2.1 Data Access objects (DAOs)

DAOs and ***WOFpy*** and work in partnership to handle data requests and deliveries from and to the client (see illustration in Figure 3.3). However because every user's database is different, some users may need to created their own DAOs. Fortunately, ***WOFpy*** comes installed with five DAO examples to provide users with references on how to build and customize DAOs. These examples cover a range of database systems – from the sophisticated CUAHSI Observation Data Model (ODM), to basic three-table SQLite databases, and to collections of comma-delimited text files. The examples are described below.

1. **ODM SQL Server Example**
   This DAO connects to a standard CUAHSI Observations Data Model (ODM) (Horsburgh, 2008) which is hosted on Microsoft SQL Server. Among the five examples, the schema of this database is the most sophisticated and comprehensive (see http://his.cuahsi.org/images/ODM1_1SchemaDiagram_md.jpg). It comprises of a collection of tables that describing not only the locations and time series of the data, but also important metadata such as variable information, collection methodologies, source descriptions; and finally a host of controlled vocabularies to define key terms in the database.

2. **SWIS SQLite Example**
   This DAO connects to the TWDB Surface Water Information System (SWIS) database. This database schema is similar to that of ODM in concept and sophistication but is customized to suit the TWDB's data needs and sampling practices. The database format is in SQLite.

3. **CBI External Service Example**
This interesting DAO connects not to a database but to a website. It reads and scrapes real-time observations of wind and tide from the Conrad Blucher Institute's (CBI) Texas Coastal Ocean Observation Network (TCOON) website at http://lighthouse.tamucc.edu/TCOON/HomePage. The DAO then maps this information to the ***WOFpy*** "**model**". To improve performance, this DAO uses local cache to store less-frequently updated information (such as metadata) to reduce internet downloading time. The DAO contains a script which enables the user to refresh this internal cache with the latest metadata from the CBI website.

4. **Three-Table "Barebones" SQLite Example**
This DAO connects to a simple SQLite database that follows the "***Datacube***" model which is the basic core schema behind many environmental databases. The ***"Datacube"*** model only contains three tables: a "Sites", a "Variables" and a "TimeSeries" table which describe, respectively, the location, parameter and time series of field observations. This DAO demonstrates that it is not necessary to maintain complex database schemas in order to publish to WaterML. A three-table database schema is sufficient to support the full range of WaterOneFlow methods.

5. **Comma-Delimited Text File Example**
This interesting DAO does not connect to a normal database but to a collection of ASCII text files. This DAO reads and parses the files and populates the ***WOFpy*** **"model"** with their contents. This simple DAO serves as the basis for a informative **"Hello World"** tutorial for beginner users to learn how to build their own DAO.

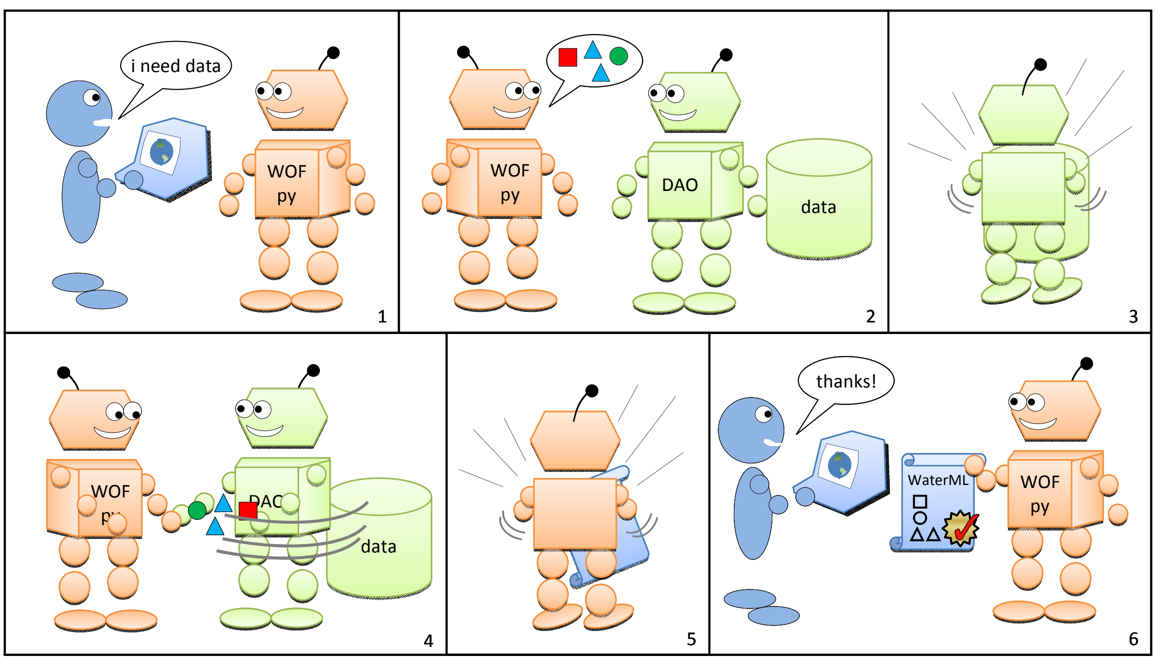


Figure 3.3 *WOFpy* and DAO work together to deliver data to the client (Whiteaker, 2011)

In addition to the above examples, software documentation and tutorials were created by the developer team to provide additional guidance to the user. These are available online at the PyPI website at: http://packages.python.org/***WOFpy***/#how-do-i-use-***WOFpy***-to-publish-my-data.

4.0 RESULTS

4.1 RESEARCH PRODUCTS

In accordance with the scope of work for this project, the developer team has produced the following deliverables for the TWDB:

1. A production quality WaterML1.0 Python framework
2. A prototype WaterML2.0 Python framework
3. A SWIS wrapper Python module
4. A CBI WaterML-TCOON translator Python module

4.2 SOURCE CODE

Source code, development documentation and installation instructions for ***WOFpy*** are maintained through the *github* public source code repository at: [https://github.com/swtools/***WOFpy***](https://github.com/swtools/WOFpy). The software packages ***WOFpy*** is available under a BSD open source license. The alpha version of ***WOFpy*** is registered under the Python Package Index at: [http://pypi.Python.org/pypi/***WOFpy***/](http://pypi.Python.org/pypi/WOFpy/). Unit tests for ***WOFpy*** are provided along with the source code to the developer community as tools to ensure quality and performance of the code.

4.3 DOCUMENTATION

A software manual was developed in Sphinx is to help users get started with ***WOFpy***. It is available online at http://packages.python.org/WOFpy/. In addition, API documentation for WOFpy was created using the python package Epydoc (see Figure 4.1) for the potential developers.

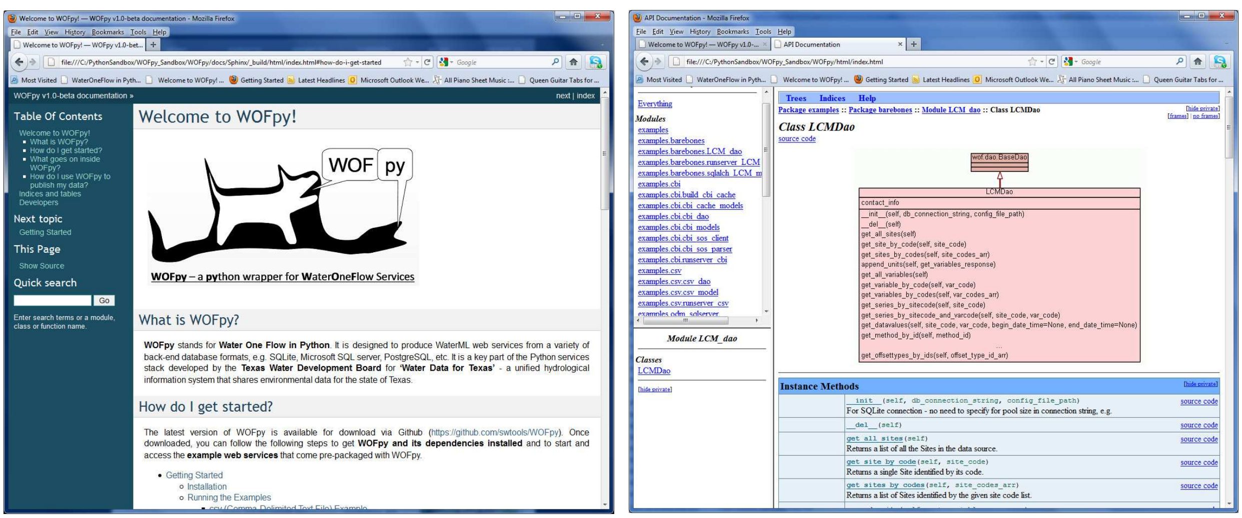


Figure 4.1 *WOFpy* manual (left) and API documentation (right).

4.4 EDUCATION AND OUTREACH

The following presentations were conducted by RPS-Espey and the University of Texas at Austin to educate and promote the use of ***WOFpy*** among the scientific and engineering community.

- May 2011 – Presentation at Espey Consultants Inc, Austin TX
  Presenters: Ernest To, Dharhas Pothina and James Seppi
- June 2011 – Presentation at 2011 CUAHSI Conference on Hydrologic Data and Information Systems, Logan UT.
  Presenters: Ernest To and Tim Whiteaker
- Sept 2011 – CUAHSI-HIS Workshop, The University of Cincinnati, Cincinnati, OH.
  Presenters: Tim Whiteaker
- Nov 2011 – CUAHSI webinar (recorded presentation and materials available at: http://his.cuahsi.org/webinars.html)
  Presenters: Tim Whiteaker and Ernest To

4.5 APPLICATIONS OF WOFPY

At the writing of this document, four major coastal datasets have been integrated into the ***Water Data for Texas*** system using ***WOFpy*** (Pothina and Wilson, 2011). They are listed below:

- Texas Commission on Environmental Quality (TCEQ) Surface Water Quality Monitoring Information System (SWQM) Dataset
- Texas Parks and Wildlife Department (TPWD) Coastal Fisheries Dataset
- Conrad Blucher Institute (CBI) Texas Coastal Ocean Observation Network (TCOON) Dataset
- Texas Water Development Board (TWDB) Bays and Estuaries (B&E) Dataset

5.0 REFERENCES

1 Pothina, D. and A. Wilson, 2011. *Building a Coastal Geodatabase for the State of Texas*: A Report submitted to the Texas General land Office Coastal Impact Assistance Program, Fiscal Year 2009.

2 Horsburgh 2008 Horsburgh, J. S., D. G. Tarboton, D. R. Maidment and I. Zaslavsky, 2008, *A Relational Model for Environmental and Water Resources Data*, Water Resour. Res., 44: W05406, doi:10.1029/2007WR006392.

3 Whiteaker, T.L., 2011. *CUAHSI-HIS workshop at The University of Cincinnati*, http://his.cuahsi.org/workshop-cincinnati2011a.html