COL/NW/08

Utilizing XML/SOAP to provide a machine-to-machine interface to TAO Data

FY 2003 Proposal to the NOAA HPCC Program

August 19, 2002

| Title Page | Proposed Project | Budget Page |

Principal Investigator: **Nancy N Soreide**

Line Organization: OAR
Routing Code: R/PMEL

Address:
NOAA/PMEL
7600 Sandpoint Way
Seattle, WA 98115

Phone: (206) 526-6728
Fax: (206) 526-6700
E-mail Address: Nancy.N.Soreide@noaa.gov

Michael J. McPhaden
Director, TAO Project Office
Michael.J.Mcphaden@noaa.gov

Eugene F Burger
PMEL-JISAO
Eugene.Burger@noaa.gov

Proposal Theme: **Technologies for collaboration, visualization or analysis**

Funding Summary: FY 2003 $ 75,800

Nancy N Soreide	Cynthia L. Loitsch	Eddie N. Bernard
Assoc. Director for IT	Program Support Officer	Director
PMEL	PMEL	PMEL

Utilizing XML/SOAP to provide a machine-to-machine interface to TAO Data

Proposal for FY 2003 HPCC Funding

Prepared by: Nancy N. Soreide

Executive Summary:

TAO forms the observational cornerstone of NOAA's El Nino Observing System, supporting NOAA's mission of improving seasonal to interannual climate forecasts. Data from these moored arrays are available in real-time and incorporated into routine weather and climate forecasts made at NCEP and elsewhere. The data are also used widely by the research community involved in climate studies focusing on El Nino, La Nina, and tropical Atlantic climate fluctuations. Finally, the data are used extensively for educational purposes at primary to postgraduate school levels. TAO data is available from PMEL through web data display and delivery pages and by ftp. However, the TAO project has recently been required to develop creation of custom ftp data files, and users are developing *ad hoc* methods for automated download of customized data subsets to support model assimilation and other research needs.

We propose to develop a W3C standards-based Web Service to support automated machine-to-machine, customized requests for designated subsets of the TAO distribution data sets. Once "published", the TAO Web Service can be automatically located and invoked by other Web Services and applications (such as DODS/OPeNDAP). We will provide users with an Application Programming Interface (API) to this Web Service and a web services registry which will allow remote users to write their own clients or applications to extract customized subsets, and will make TAO data available for realtime ingest into data assimilation models over the NGI. The explosion of Web Services in the commercial sector for machine-to-machine transactions mandates that this technology be applied and tested in the research environment. To our knowledge, no observing systems are providing Web Services such as those proposed here. The TAO project, which has provided over 102,000 TAO data files to numerous user communities world wide, is a superb testbed. The TAO Web Services will serve as a model for all NOAA Observing Systems.

Problem Statement:

Background: TAO (http://www.pmel.noaa.gov/tao) forms the observational cornerstone of NOAA's El Nino Observing System, in support of NOAA's mission of improving seasonal to interannual climate forecasts. PIRATA (http://www.pmel.noaa.gov/pirata) in the Atlantic also forms a critical element of an emerging Atlantic Ocean observing system for improved climate forecasts and analyses. Data from these moored arrays are available in real-time and incorporated into routine weather and climate forecasts made at NCEP and elsewhere. The data are also used widely by the research community involved in climate studies focusing on El Nino,

La Nina, and tropical Atlantic climate fluctuations. Finally, the data are used extensively for educational purposes at primary to postgraduate school levels.

TAO data includes air temperature, ocean temperature, relative humidity, wind, rainfall, longwave radiation, air pressure, salinity, pressure, and ocean currents, as part of the standard dataset. Geographically, the array spans the tropical Pacific from 95W to 137E and from 8N to 8S. Additional sensors and buoy locations are added to support specific research needs. TAO data is presently available from PMEL through web-based data display and data delivery pages and by ftp. In the past year, the TAO project has provided 102,000 TAO data files by these methods. However, the TAO project has recently been required to develop creation of custom ftp data files, and users are developing *ad hoc* methods for automated download of customized data subsets to support model assimilation and other research needs.

Problem details: There is presently no mechanism by which automated machine-to-machine connections can be made for user-customized downloads of TAO data, nor can TAO data be subsetted by a remote machine-to-machine connection. In fact, a few TAO user communities have developed custom *ad hoc* programs to extract the data. For example, one group wrote a program which logs into the TAO ftp site, downloads text files and runs a text parser to extract the data from these files. Another group is accessing CGI scripts provided on the TAO web server to periodically extract standard files.

There are obvious inefficiencies, inherent limitations and possibly even security issues associated with these ad hoc methods, all of which can be addressed by the proposed Web Services implementation. Some limitations of the *ad hoc* methods include: a) There is no error handling mechanism if the data being requested is unavailable, b) The data retrieval mechanisms are hardwired to certain data sets, and cannot be altered without TAO and client personnel intervention, c) Data are being duplicated, as the same data are being made available in several different files formats, all of which are static files on the server, d) Since all the data files are static files on the server, data management becomes a problem, e) Non of these non-human data retrieval methods are well published, which is a hindrance to any new data user wanting to automate their TAO data retrieval process, f) The TAO data files are in non-standard text file formats, which complicates machine parsing, g) It is a cumbersome process to have parameters removed or added to the datasets users download, as they have to call the TAO office to have the ftp files modified.

Relationship to HPCC program objectives: The proposed effort addresses the primary HPCC goal to "*support information technology research and development that enables NOAA to provide greater access to its vast holdings of realtime and historical to users in a more complete, more usable form and much more timely manner through the increased use of advanced technologies associated with the Internet*." TAO is the cornerstone of NOAA's El Nino observing systems. The proposed effort addresses the goal of the HPCC Collaboration, Visualization or Analysis theme because it is a "*modern network based application that demonstrates new techniques for working with NOAA data and information*." The proposed software is standards-based, scalable and extensible, and will be made freely available for use throughout NOAA.

Proposed Solution:

Synopsis: A Web Service consists of a number of technologies specified by the W3C standard, including XML/Simple Object Access Protocol (SOAP), Uniform Description, Discovery and Integration (UDDI), Web Service Descriptive Language (WSDL) and HTTP. We propose to implement this standards-based Web Service to support automated machine-to-machine, customized requests for designated subsets of the TAO distribution data sets. All information transferred by the Web Service will use the XML/SOAP format. Once published using UDDI, the TAO Web Service can be automatically located and invoked by other Web Services and applications (such as DODS/OPeNDAP). We will provide an Application Programming Interface (API) for this Web Service which will allow remote users to write their own clients or applications to extract customized subsets, and will make TAO data available for realtime ingest into data assimilation models over the NGI. *The proposed TAO Web Services will address and overcome the obvious inefficiencies, inherent limitations and potential security issues associated with current ad hoc methodologies for automated machine-to-machine transfers.*

The explosion of Web Services in the commercial sector for machine-to-machine transactions mandates that this technology be applied and tested in the research environment. To our knowledge, no observing systems are providing Web Services such as those proposed here. The TAO project, which has provided over 102,000 TAO data files to numerous user communities world wide, is a superb testbed. This success of the TAO Web Services will be presented at professional society meetings, made freely available, and will serve as a model for all NOAA Observing Systems.

Implementation: We will implement a Web Service portal to TAO El Nino Observing System data to allow the TAO data user community to automate customized, realtime or historical TAO data extracts. The data that will be made available via this Web Service is the complete TAO distribution dataset. Since TAO data is presently being migrated from flat ASCII text files into a SQL database (including comprehensive meta data), the TAO Web will utilize this new TAO database.

Web Services are implemented and published by service providers (the TAO project serving the TAO data) and are discovered and invoked by service requestors (TAO data user community). We propose to develop the TAO Web Service implementation and registry as well as a simple TAO Web Service client that will be used to test the application and provided to the TAO user community as an example. The software to be written can be separated into these components.

- The Web Services application API
- Middleware to provide database connectivity.
- Establish a UDDI service registry
- Create an example client for data access
- Management tools

Technologies to be applied: Some of the more prominent solutions available in the market place are the Microsoft .Net Framework, the IBM WebSphere Commerce Suite provider, the BAE WebLogic platform or Sun Microsystems's Sun ONE effort. Since we have a solid Java

and Apache/Tomcat knowledgebase within PMEL, and these are generic, non-proprietary solutions, the implementation will revolve around these technologies with most of the implementation being done with Java. API's that we will use include Sun's J2SE and core elements for the Sun ONE initiative.

The Web Services application API: This component will be written in Java and will provide the functionality to the data user community. This API will enable data users to request:

- A list of sensors for which data are available for each buoy
- A list of buoys for which data are available
- Sensor data over a time period
- Sensor data over a time period and within a certain area.

The API will be designed to allow it to accommodate new sensors that might be added to buoys in the TAO array. This API can always be expanded in future should this method of data retrieval prove to be popular. The API and underlying software will be implemented fully in Java. There are parts of the component that will be invisible to the user. One of these will be logging sessions that will log the IP Address and amount of data each user downloads (permitted by OMB control # 0648-0342). The information logged is a subset of the information normally logged by any web server.

Middleware: Through Web Services the user will be isolated from the database architecture. Middleware, which in some cases might be legacy software, will be used to retrieve the information from the database and encode it into an XML/SOAP object, and then handed down to the Web Service/Server for transmission to the client application. This software, as the name indicates will do transaction marshalling and processing between the Web Service and Database. Mostly Java will be used although it is possible that legacy components written in other languages will be used.

Service registry: A Uniform Description, Discovery and Integration (UDDI) Service registry is where a potential user will find information about the TAO Data Portal Web Service API. An XML format known as the Web Service Descriptive Language (WSDL) is used to describe the Web Service to the registry. The Web service registry also uses WSDL to describe the Web Service to a potential client. The service registry is included with many of the developer toolkits, as well as tools to generate the WSDL files. This component will then just have to be installed.

Example client: To allow us to test this application we will write a client which can exercise all the web service methods. We will write the client in Java. The source code and client will be made available to prospective users of this Web Service.

Management tools: Since new user data usage logging standards will be set with this project, and small web application will be written to allow data managers manage this application. Data mangers will be able to view the amount of TAO data being retrieved, and also stop and restart the web service application. Access to this application will obviously be available only to selected TAO data managers.

Analysis:

Web Services will provide a standards-based, self-documenting machine-to-machine portal to TAO data, which will interface smoothly with other web services and applications, and will replace myriad *ad hoc* or one-off customized data retrieval paths. Examples of technologies providing similar functionalities as Web Services are Java RMI and CORBA. What distinguishes Web Services from these is that the major industry players agree to the Web Services Architecture and components and Web Services architecture is to become a W3C standard. This enables this technology to be compatible with other platforms. Since clients on any platform can now access this service more clients can use this service. The write-once-and-deploy strategy enables us to build more functionality into this one service to serve a broader user community. Since the technology uses largely self-documenting XML files for messaging, it will be easy for user communities to write their own clients to extract the data.

Another approach would be to allow direct access to the TAO database via an ODBC/JDBC interface. To do this the database server has to be opened to outside public access, which poses serious security risks. Any user wanting to make use of this solution would need detailed knowledge of the database schema and SQL knowledge to allow them to do data extracts. It will also be cumbersome to control overuse of database resources by any one user. Through Web Services the user is insulated from the intricacies on the database schema. Unless data fields are removed from the database schema, the user would not be aware of any changes to the db when they access the data via Web Services. Since Web Services use the HTTP protocol, there is not need to expose any more of the web server than one would by, for instance, serving web pages, or allowing anonymous ftp'ing. This implementation will also automate the data extract logging, which is presently done by hand.

Performance Measures:

A pivotal measure of the success of this application will be the number of users using the application and the amount of data retrieved. The ease of implementation will be compared to other implementations the TAO group is presently using to provide data. However, with today's emphasis on realtime data assimilation into models, and based on the customized data request the TAO project is beginning to receive, we believe this is a forward-looking solution that will meet growing needs within the research community using Observing System datasets.

Milestones

Month 3 – Complete Web Service API design
Month 8– Complete and install Web Service API and registry
Month 12 – Test, make improvements based on feedback, and install Service for operational use

Deliverables

- A TAO Web Service Data Portal and registry for machine-to-machine TAO data retrieval
- A simple TAO Web Services web based management tool.