I.M. Systems Group, Inc.
Perot Systems Government Services
Contract #EA133C04CN0044
National Oceanic and Atmospheric Administration

Consideration of the Use of the GML Simple Feature Profile via a Web Feature Service for the Transport of In-situ Observation Data

John Ulmer
Perot Systems Government Services at the National Oceanic and Atmospheric Administration, Coastal Services Center

Contributors
Rob Cermak – Alaska Ocean Observing System
Grant Cameron - Scripps/ Coastal Data Information Program
Jerome King – NOAA Southwest Fisheries Science Center
Vembu Subramanian – University of South Florida (COMPS)

August 22, 2008

I. M. Systems Group, Inc.

perotsystems®
Government Services

Table of Contents

Introduction to the Data Transport Laboratory (DTL)

The Integrated Ocean Observing System (IOOS) data providers operate in a highly heterogeneous computing environment and follow the Ocean.US Data Management and Communications (DMAC) strategy to enhance interoperability and deliver effective data transport. The "on-the-ground" possibilities for implementing the DMAC transport strategies are broad, and each provider may implement them differently. Consequently, in some cases there are redundant research and development efforts and in others an inability to test and deploy a candidate solution because it is new or different from the existing environment. Further, some candidate solutions are backed by political or cultural camps that advocate for their respective candidates from a narrow perspective. There is a risk that the political and cultural dynamics and resource limitations could cause the adoption of candidates that would not necessarily be the best solution from an information technology (IT) perspective.

The Data Transport Laboratory (DTL) seeks to provide a persistent and objective development and testing facility to reduce the effects of political or cultural dynamics and increase an objective community-wide approach to selecting data transport solutions. To that end, the DTL surveys the community for candidate technologies, assesses their relevance to the Ocean.US DMAC and local and regional data providers, and picks relevant candidates to implement. The implementation of those candidates in the DTL will provide an objective basis for discussion and decision-making in the Ocean.US DMAC community.

DTL Project Sequence

The DTL uses a "project" construct to examine technologies relevant to IOOS data transport. Each technology project follows a process that is intended to educate the DTL staff on the technology, produce a reference implementation of the technology, engage external partners to test and critique the technology, and produce a final report. The following are the general steps that are followed in DTL projects.

1. Select next appropriate project for the DTL according to the Technology Selection Plan.
2. Build the project on the DTL servers and document the project.
3. While implementing the technology in the DTL, recruit external partners to participate in the testing and critiquing of the project.
4. Also, while implementing the project in the DTL, design and implement a testing approach for the project. The testing approach should be as objective as possible. However, many important dynamics associated with a given project may be quite subjective. For the subjective material, the DTL staff will be responsible for compiling and organizing contributions from external partners in an appropriate and productive manner.

5. Provide external partners with testing plan for review and modify as appropriate.
6. Initiate internal and external testing of the DTL project.
7. Coordinate testing and solicit results and commentary from the external partners.
8. Compile results and commentary.
9. Draft report.
10. Fully document the project on the DTL website (*www.csc.noaa.gov/DTL/*) with possible content for the Community Information Repository (*www.csc.noaa.gov/cir/*) and pass on to the IOOS DMAC as appropriate.

DTL Project 4 – Transport of In-situ Observation Data with the GML Simple Feature Profile via WFS

Early planning for the IOOS Data Management and Communications (DMAC) called for a service-oriented architecture (SOA). In the *Data Management and Communications Plan for Research and Operational Integrated Ocean Observing Systems (March 2005),* produced by Ocean.US, the DMAC Steering Committee recommended that the "WFS [Web Feature Service] be examined for incorporation into the DMAC data transport suite. . . ." Also, in response to slow adoption to the WFS specification because of its complexity, the Open Geospatial Consortium (OGC) developed the Simple Feature Profile (SFP), which simplified the development of application schema using the WFS specifications.

The WFS specifications, a representational state transfer (REST)-based eXtensible Markup Language (XML) data transport architecture, grew out of the geographic information system community and were known to handle three-dimensional data well. However, there was some question whether the Geography Markup Language (GML) and WFS combination could adequately handle three-dimensional data with the addition of the time dimension.

The two general purposes of this DTL project were to determine whether

- The GML offers a library of XML structures capable of meeting basic or advanced data modeling needs for time series data, and whether
- The WFS specification, as implemented in the test service (microWFS), provides the services necessary to query and retrieve time series data.

Project Selection

Suggestions for candidate technologies to be implemented in the DTL are accepted from the Integrated Ocean Observing System (IOOS) community. The primary sources for the identification of candidates are the local observatories, regional associations, the Ocean.US DMAC team and its expert teams, and deliberate outreach efforts of the DTL staff, such as meetings and workshops.

Candidate technologies identified through these mechanisms are filtered through a set of criteria to assess their relevance and importance to the stakeholders. For more details, read about the DTL technology selection process (*www.csc.noaa.gov/DTL/dtltechselec.html*).

The selection for consideration of the GML Simple Feature Profile via WFS project was based on the following table (Table 1), which lists the scoring and rationale of that scoring for the selection of this project. The scores range from zero to five.

Table 1 – DTL Project Selection Scoring

Criteria	Score	Rationale
Significance to Local Observatories	3	The GML offers a library of structures pertinent for data storage at both the local observatory and data network node level. The WFS offers a standard interface for machine level communications appropriate at the data network node.
Significance to Regional Associations	3	Frequently decision support tools at the regional level utilize map based products, and some of the server applications behind these products inter-operate using the WFS. The WFS also has the potential to be a primary transport service between the data network nodes in the region and the data marts on the federal backbone.
Relevance to DMAC Guidance	5	DMAC currently recommends WFS as pre-operational. Note, there are no recommendations for 'operational'.
Relative Maturity of Candidate	4	The OGC WFS and GML are some of the more mature and stable specifications under consideration for IOOS data sharing. Neither are expected to undergo significant change in the near future, and are open enough to accept a broad range of community designed data models. The WCS is a related service to WFS that may be in a growth phase an has some similar functionality.
Appropriate Level of Effort for DTL	4	The implementation of OGC services is appropriate.
Coincidence of other Center Efforts	3	CSC is currently pursuing the use of the Simple Feature Profile for use in the IOOS Catalog of Local Observations.
Total Score	22	

Related Links

GML Specifications	*http://www.opengeospatial.org/standards/gml*
WFS Specifications	*http://www.opengeospatial.org/standards/wfs*
Simple Feature Profile Spec's.	*http://portal.opengeospatial.org/files/?artifact_id=11266*
ESRI Marine Data Model	*http://support.esri.com/index.cfm?fa=downloads.dataModels.filteredGateway&dmid=21*
DTL Time Series Schema	*http://www.csc.noaa.gov/ioos/schema/insituTimeSeries.xsd*

Use Case Description

The DMAC calls for a Service Oriented Architecture (SOA) that is facilitated by open standards. This SOA should be manifested as a limited number of services that provide access to a contributor's data holdings. As the IOOS data exchange standards evolve, it is unlikely that a single unified data model and service description could be produced to handle all data types at the national, regional, and subregional areas of the IOOS. Instead, it is more likely that the IOOS data exchange standards and infrastructure will be an incremental evolution of both competing and complementary standards and technologies. Since the intent of the DTL is to foster convergence on a complementary set of standards and infrastructure, it is critical that those standards and technologies selected for use in the evolving system present implementers with the greatest possible flexibility and the least risk associated with early adoption.

The DTL focuses on the regional and subregional arenas. The DMAC asserts that the standards and technologies adopted should do no harm to the data provider, or at least should do as little harm as possible. This strongly implies that selected solutions should be minimally invasive to the data provider's data management systems. Reviews of candidate standards and technologies should assess the extent to which a local observatory's or data provider's IT systems would have to be modified to implement the candidate and how easily the data provider would be able to respond to the evolving data model work.

In this project, the DTL considered how these technologies work for the transport of in-situ measured physical and chemical parameters in the local and regional arenas. A general architecture is shown in Figure 1 below. This architecture does not preclude or prevent a local observatory or regional (or subregional) aggregation facility from making its data available in any other way it deems necessary to serve its own mission. But, for the purposes of this project, those non-IOOS data users and services are not relevant. Further, the data transport is between machines where the client is a regional aggregation facility in the case of a local observatory, or a federal assembly center in the case of a regional aggregation facility.

The regional aggregation facilities may be creatures of the regional associations, which may tend to foster a regional approach to the aggregation behavior. But beyond that general tendency, considerations of the relationship between a regional association and a regional aggregation facility are beyond the scope of this project.

The primary consumer of the local observatory data services will likely be a regional or subregional aggregation facility. The facility's primary function will be to aggregate the local data feeds into consistent temporal and spatial repositories from a regional perspective so that they can be efficiently provided to information product developers and national data centers. The primary consumer of data provided by the regional aggregation centers will be a data mart or some component of the federal backbone.

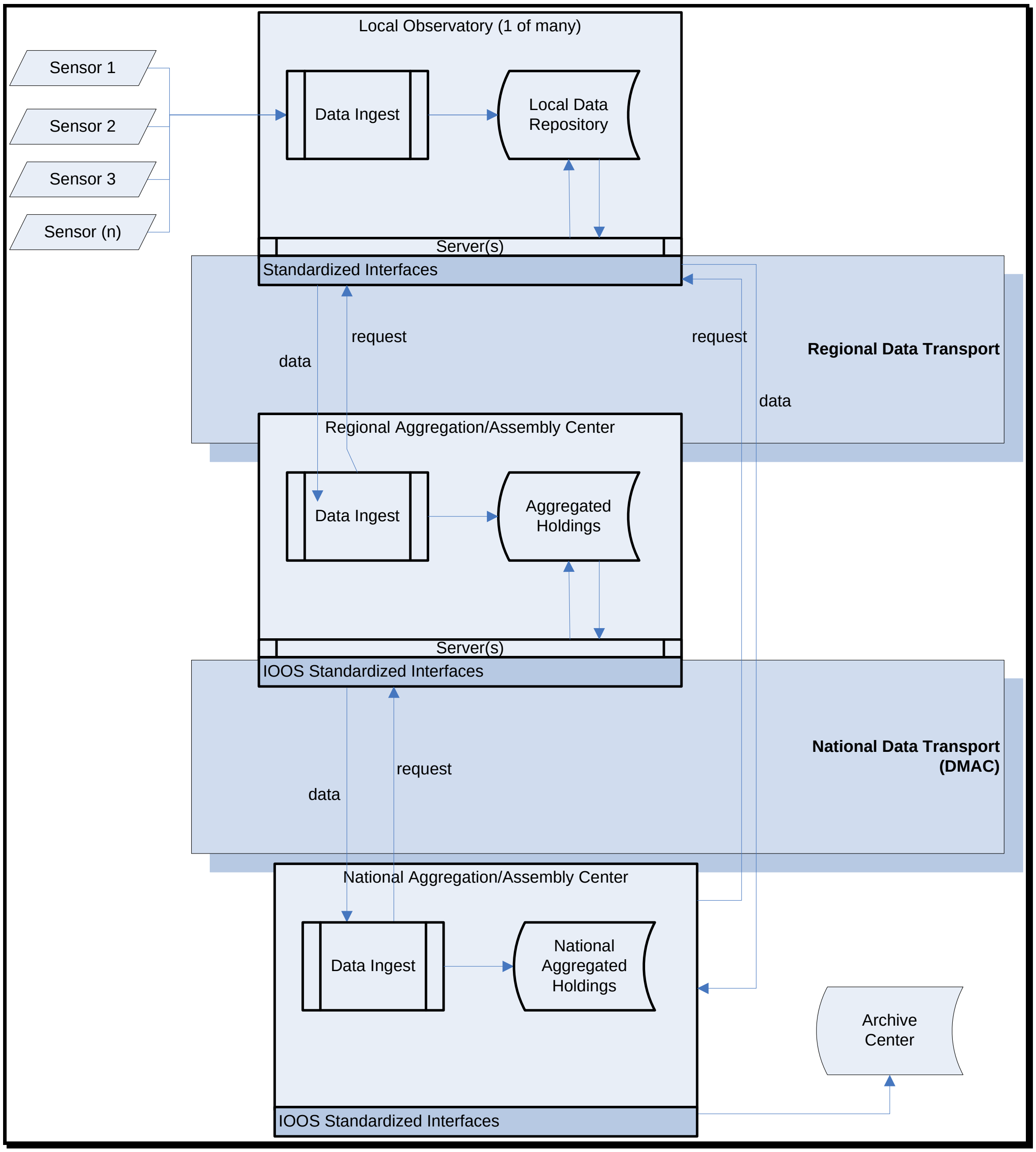

Figure 1 - Suggested General SOA for Regional or Sub-regional Data Transport

Service Architecture and Implementation

To test the utility of the GML Simple Feature Profile via a Web Feature Service (WFS), the DTL developed a simple data transport XML schema and WFS service. The structure of the schema was developed via traditional data model development methods and by comparing the drafts to other data models such as the ESRI Marine Data Model. The data model and schema development assumed several boundaries. The data model was intended for use with physical and chemical in-situ oceanographic observations. The sensors were assumed to be generally stationary, such as buoys or fixed piers. Further, a general data structure was adopted that asserted the following:

- An organization operates one or more platforms;
- A platform holds one or more sensors; and
- A sensor produces a measured value for some observed property repeatedly over time.

Once the data model and associated schema were developed (see Appendix C), the DTL built two implementations of a simple WFS or 'microWFS': one as a Java Servlet that queries data from a Microsoft SQL Server database, and the second as a Perl Common Gateway Interface application that queries data from a PostgreSQL database. Both query data from a simple database view that is specified by the DTL. Both versions assume that the data provider manages its data with a Relational Database Management System. Both versions progressed through the traditional software development processes, yielding an alpha version, review, beta version, review, and final release.

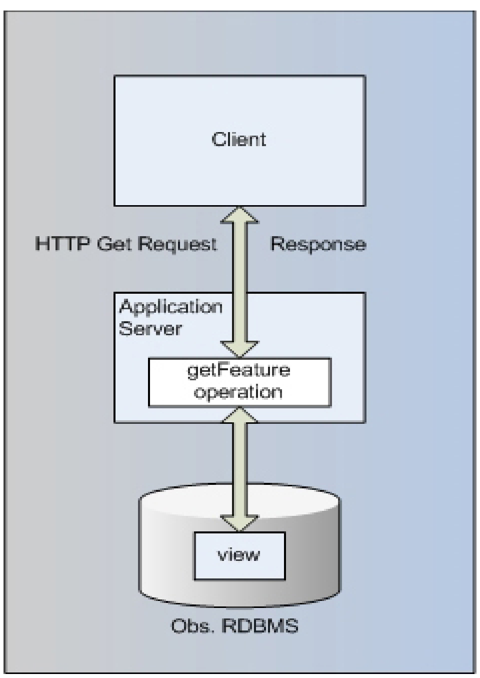

Figure 2 – General Architecture of microWFS

The function and output of the two implementations is identical. Further, they are not fully compliant with the WFS specifications since they only provide one method and basic error responses. There are several mandatory elements of the WFS specification that have been neglected, such as the *getCapabilities* and *describeSensor* operations. These WFS implementations are intended solely as tools to facilitate the exploration of using GML Simple Feature Profile and the WFS to deliver time series data.

To support the installation of the microWFS software, the DTL produced full documentation for both Java and Perl versions and posted that documentation on a project website. Also, an on-line forum was created to facilitate dialogue between the project partners.

In the DTL, the Java Servlet and Perl Common Gateway Interface (CGI) microWFS software were installed on two Dell PowerEdge 1850 machines, each with two 3.0 MHz Xeon CPUs, four gigabytes of RAM, 300 gigabytes of 10,000 RPM disks, and Gigabit network cards. The operating system is Red Hat Linux Enterprise 4, kernel 2.6.9-42.

The XML schema follows a simple hierarchy where the top level entity is a feature collection (ioos:featureCollection). A feature collection contains one or more feature members such as buoys or piers (ioos:featureMember). Each feature member contains one or more time series events (ioos:tsEvent) containing time information and a result value for the observed property (e.g., temperature, salinity, etc.). The schema follows:

.

From **http://www.csc.noaa.gov/ioos/schema/insituTimeSeries.xsd**:


```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!-- W3C Schema generated by NOAA Coastal Services Center 2006-->
<!-- time series insitu schema for DTL testing                    -->
<!-- Primary editor: Daniel Martin. Last updated 2007-06-20        -->
<!--  -->
<xs:schema targetNamespace="http://www.csc.noaa.gov/ioos"
    xmlns:gmlsf="http://www.opengis.net/gmlsf" xmlns:gml="http://www.opengis.net/gml"
    xmlns:ioos="http://www.csc.noaa.gov/ioos" xmlns:xs="http://www.w3.org/2001/XMLSchema"
    elementFormDefault="qualified" version="1.0">
 <xs:annotation>
  <xs: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/gmlsfProfile/1.0.0/gmlsf.xsd</gmlsf:GMLProfileSchema>
  </xs:appinfo>
 </xs:annotation>
<!-- includes and imports -->
 <xs:import namespace="http://www.opengis.net/gml"
     schemaLocation="http://schemas.opengis.net/gml/3.1.1/base/gml.xsd"/>
 <xs:import namespace="http://www.opengis.net/gmlsf"
    schemaLocation="http://schemas.opengis.net/gml/3.1.1/profiles/gmlsfProfile/1.0.0/
    gmlsfLevels.xsd"/>
<!-- define complex feature for the time series values -->
 <xs:complexType name="TSMeasurementPropertyType">
  <xs:sequence>
   <xs:element ref="ioos:TSMeasurement"/>
  </xs:sequence>
 </xs:complexType>
 <xs:element name="TSMeasurement">
  <xs:complexType>
   <xs:sequence>
    <xs:element name="obsDateTime" type="xs:dateTime"/>
    <xs:element name="observation" type="gml:MeasureType"/>
   </xs:sequence>
  </xs:complexType>
 </xs:element>
```

```
<!-- define feature types -->
<!-- the gml:id shall reflect a unique obsID value from originating data source -->
<!-- suggested coordinate axis order is Y,X as defined by the default srsName value -->
<xs:element name="insituTimeSeries" type="ioos:insituTimeSeriesType" substitutionGroup="gml:_Feature"/>
<xs:complexType name="insituTimeSeriesType">
 <xs:complexContent>
  <xs:extension base="gml:AbstractFeatureType">
   <xs:sequence>
    <xs:element name="sensor">
     <xs:complexType>
      <xs:simpleContent>
       <xs:restriction base="gml:CodeType">
        <xs:attribute name="codeSpace" type="xs:anyURI" use="optional" default="http://csc.noaa.gov/ioos/dictionaries/SensorDictionary.xml"/>
       </xs:restriction>
      </xs:simpleContent>
     </xs:complexType>
    </xs:element>
    <xs:element name="observationName">
     <xs:complexType>
      <xs:simpleContent>
       <xs:restriction base="gml:CodeType">
        <xs:attribute name="codeSpace" type="xs:anyURI" use="optional" default="http://csc.noaa.gov/ioos/dictionaries/PhyOceanDictionary.xml"/>
       </xs:restriction>
      </xs:simpleContent>
     </xs:complexType>
    </xs:element>
    <xs:element name="verticalDatum">
     <xs:complexType>
      <xs:simpleContent>
       <xs:restriction base="gml:CodeType">
        <xs:attribute name="codeSpace" type="xs:anyURI" use="optional"
              default="http://csc.noaa.gov/ioos/dicitonary/VerticalDatumDicitonary.xml"/>
       </xs:restriction>
      </xs:simpleContent>
     </xs:complexType>
    </xs:element>
    <xs:element name="verticalPosition" type="gml:MeasureType"/>
```

```
     <xs:element name="horizontalPosition" type="gml:PointPropertyType"/>
     <xs:element name="tsEvent" type="ioos:TSMeasurementPropertyType" maxOccurs="unbounded"/>
    </xs:sequence>
   </xs:extension>
  </xs:complexContent>
 </xs:complexType>
 <!-- define feature collection -->
 <xs:element name="insituTimeSeriesCollection" type="ioos:insituTimeSeriesCollectionType" substitutionGroup="gml:_GML"/>
 <xs:complexType name="insituTimeSeriesCollectionType">
  <xs:complexContent>
   <xs:extension base="gml:AbstractFeatureType">
    <xs:sequence maxOccurs="unbounded">
     <xs:element name="featureMember">
      <xs:complexType>
       <xs:sequence>
        <xs:element ref="gml:_Feature"/>
       </xs:sequence>
      </xs:complexType>
     </xs:element>
    </xs:sequence>
   </xs:extension>
  </xs:complexContent>
 </xs:complexType>
</xs:schema>
```

External Partners

Twelve project partners were recruited via contacts with the IOOS Regional Associations (see Appendix B). All partners participated in early conference calls and considerations of the testing regimen. However, because of resource or timing constraints, several partners where unable to fully participate in the service installation and testing. The following were able to create the necessary database view and install a version of the microWFS application:

- Alaska Ocean Observing System (AOOS)
- Carolinas Coastal Ocean Observing and Prediction System (CaroCOOPS)
- Coastal Data Information Program (CDIP)
- Gulf of Maine Ocean Observing System (GoMOOS)
- NOAA Southwest Fisheries Science Center (SWFSC)
- University of South Florida (USF)

Server Descriptions	AOOS	CaroCOOPS	CDIP	GoMOOS	NOAA SWFSC	USF
No. of CPUs	1	unavailable	unavailable	unavailable	2	2
CPU Speed (GHz)	3	unavailable	unavailable	unavailable	3.6	3.2
RAM (GB)	2	unavailable	unavailable	unavailable	8	4
Network Interface	Gigabit Ethernet	unavailable	unavailable	unavailable	1000BaseT Ethernet	Gigabit Ethernet
Operating System	Linux(Centos)	unavailable	unavailable	unavailable	Linux	Linux(RHEL)

The partners who volunteered to assist with the execution of the testing regimen were: AOOS, CDIP, SWFSC, and USF.

Test and Critique Regimen

The DTL staff drafted a testing regimen that the team reviewed and adopted. The DTL coordinated periodic conference calls using WebEx to facilitate planning, scheduling, project execution, and team discussion. The data sets available via each implementation differed in size and complexity. As each external partner performed the testing regimen against the other server installations, they sent their performance results to the DTL, which calculated simple statistics on the result sets. Subsequently, the external partners submitted written critiques in response to the remaining items of the testing regimen (see Appendix A).

The Apache Project JMeter (*http://jakarta.apache.org/jmeter/*) tool was selected to execute items 1 and 2 of the test plan. JMeter is a graphical desktop Java application that supports a number of query types including HTTP GET and POST, SOAP/XML, JDBC, and others. JMeter facilitates scheduled loading of servers and measures server response time, throughput, and other basic statistics.

Three JMeter tests were designed. The first test issued 10 requests for a single data record, waiting for each response before issuing each subsequent request. The second test issued 10 sequential requests for 500 data records. The third test issued 50 requests for 500 records, ramping from 0 to 50 requests in 10 seconds.

Following the WFS specifications, the input fields with typical input values for the microWFS were:

- SERVICENAME=dtlservice
- REQUEST=getFeature
- SERVICE=microWFS
- VERSION=1.1.0
- OUTPUTFORMAT=text/xml;subType=gml/3.1.1/profiles/gmlsf/1.0.0/1
- BBOX=-93.00,29.00,-95.00,30.00
- TIME=2007-06-01T12:00Z,2007-06-01T14:00Z
- TYPENAME=waterTemperature

Using those inputs, the following is a typical getObservation query.

> http://csc-s-ial-p.csc.noaa.gov/cgi-bin/microwfs/microWFS.cgi?SERVICENAME=dtlservice&REQUEST=getFeature&SERVICE=microWFS&VERSION=1.1.0&OUTPUTFORMAT=text%2Fxml%3BsubType%3Dgml%2F3.1.1%2Fprofiles%2Fgmlsf%2F1.0.0%2F1&BBOX=-93.00%2C29.00%2C-95.00%2C30.00&TIME=2007-06-01T12%3A00Z%2C2007-06-01T14%3A00Z&TYPENAME=waterTemperature.

Using that request, a typical getObservation response looks like:


```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!-- sample insitu time series record-->
<!-- Primary editor: Daniel Martin. Last updated 2007-05-09 -->
<gml:FeatureCollection
   xmlns:ioos="http://www.csc.noaa.gov/ioos"
   xmlns:ows="http://www.opengis.net/ows"
   xmlns:ogc="http://www.opengis.net/ogc"
   xmlns:wfs="http://www.opengis.net/wfs"
   xmlns:gml="http://www.opengis.net/gml"
   xmlns:xlink="http://www.w3.org/1999/xlink"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:schemaLocation="http://www.csc.noaa.gov/ioos http://www.csc.noaa.gov/ioos/schema/insituTimeSeries.xsd">
   <gml:boundedBy>
      <gml:Envelope srsName="urn:ogc:def:crs:EPSG:6.5:4326" srsDimension="2">
         <gml:lowerCorner>29.22 -94.4</gml:lowerCorner>
         <gml:upperCorner>29.73 -93.87</gml:upperCorner>
      </gml:Envelope>
   </gml:boundedBy>

   <gml:featureMember>
     <ioos:insituTimeSeries gml:id="ID5.17.1">
        <ioos:sensor codeSpace="urn:x-noaa:source:insitu:NFRA:2007a">SBPT2</ioos:sensor>
        <ioos:observationName codeSpace="urn:x-noaa:source:def:phyOcean:2007a">waterTemperature</ioos:observationName>
        <ioos:verticalDatum codeSpace="urn:x-noaa:def:verticalDatum"></ioos:verticalDatum>
        <ioos:verticalPosition uom="urn:x-noaa:def:uom:2007a:meter">0</ioos:verticalPosition>
        <ioos:horizontalPosition>
```

```
            &lt;gml:Point&gt;
                &lt;gml:pos&gt;29.73 -93.87&lt;/gml:pos&gt;
            &lt;/gml:Point&gt;
        &lt;/ioos:horizontalPosition&gt;
    &lt;ioos:tsEvent&gt;
        &lt;ioos:TSMeasurement&gt;
            &lt;ioos:obsDateTime&gt;2007-06-01T12:00:00Z&lt;/ioos:obsDateTime&gt;
            &lt;ioos:observation uom="urn:x-noaa:def:noaa.units:2007a:Celsius"&gt;26.8&lt;/ioos:observation&gt;

        &lt;/ioos:TSMeasurement&gt;
    &lt;/ioos:tsEvent&gt;
    &lt;ioos:tsEvent&gt;
        &lt;ioos:TSMeasurement&gt;
            &lt;ioos:obsDateTime&gt;2007-06-01T12:06:00Z&lt;/ioos:obsDateTime&gt;
            &lt;ioos:observation uom="urn:x-noaa:def:noaa.units:2007a:Celsius"&gt;26.8&lt;/ioos:observation&gt;
        &lt;/ioos:TSMeasurement&gt;
    &lt;/ioos:tsEvent&gt;

    &lt;ioos:tsEvent&gt;
        &lt;ioos:TSMeasurement&gt;
            &lt;ioos:obsDateTime&gt;2007-06-01T12:12:00Z&lt;/ioos:obsDateTime&gt;
            &lt;ioos:observation uom="urn:x-noaa:def:noaa.units:2007a:Celsius"&gt;26.8&lt;/ioos:observation&gt;
        &lt;/ioos:TSMeasurement&gt;
    &lt;/ioos:tsEvent&gt;
    &lt;ioos:tsEvent&gt;
        &lt;ioos:TSMeasurement&gt;

            &lt;ioos:obsDateTime&gt;2007-06-01T12:18:00Z&lt;/ioos:obsDateTime&gt;
            &lt;ioos:observation uom="urn:x-noaa:def:noaa.units:2007a:Celsius"&gt;26.8&lt;/ioos:observation&gt;
        &lt;/ioos:TSMeasurement&gt;
    &lt;/ioos:tsEvent&gt;
    &lt;ioos:tsEvent&gt;
        &lt;ioos:TSMeasurement&gt;
            &lt;ioos:obsDateTime&gt;2007-06-01T12:24:00Z&lt;/ioos:obsDateTime&gt;

            &lt;ioos:observation uom="urn:x-noaa:def:noaa.units:2007a:Celsius"&gt;26.8&lt;/ioos:observation&gt;
```

```
        &lt;/ioos:TSMeasurement&gt;
      &lt;/ioos:tsEvent&gt;
      &lt;ioos:tsEvent&gt;
        &lt;ioos:TSMeasurement&gt;
          &lt;ioos:obsDateTime&gt;2007-06-01T12:30:00Z&lt;/ioos:obsDateTime&gt;
          &lt;ioos:observation uom="urn:x-noaa:def:noaa.units:2007a:Celsius"&gt;26.8&lt;/ioos:observation&gt;

        &lt;/ioos:TSMeasurement&gt;
      &lt;/ioos:tsEvent&gt;
```

… several ioos:tsEvents deleted for brevity …

```
    &lt;/ioos:insituTimeSeries&gt;
  &lt;/gml:featureMember&gt;
&lt;/gml:FeatureCollection&gt;
```

General Server Performance

Because of the variability in the server hardware and load, the objective server performance measures are intended as general indications of competent performance. They are not intended as benchmarks. For the 1-record, 500-record light load, and 500-record heavy load tests, all physical servers, both those that were dedicated to the microWFS service and those that hosted other applications, showed competent performance.

Observation data for the light load tests were transferred from the servers to the clients in times that would satisfy typical HTTP clients. The average, shortest, and longest response times were 4792, 31, and 41806 milliseconds, respectively.

Table 2 – Light-Load Testing

10 Separate Requests for 1 Data Record

	AOOS	CaroCOOPS	CDIP	CSC	GoMOOS	SWFSC	USF
Average	1845	1465	18525	561.7	1267.3	7297.975	280.44
Min	120	172	6609	156	31	437	147
Max	24772	14862	41806	10253	31259	27012	429

The second test sequence made 10 sequential requests designed to retrieve 500 data records. Again, all servers performed adequately with average, shortest, and longest response times of 7325, 193, and 65916 milliseconds, respectively.

Table 3 – 500 Record Sequential Light-Load Testing

10 Separate Requests for 500 Data Records

	AOOS	CaroCOOPS	CDIP	CSC	GoMOOS	SWFSC	USF
Average	6393	1396	20346	2814	2752	16574	1000
Min	557	479	7674	1343	272	3496	193
Max	65916	13597	26123	11334	20167	27033	1384

In the third test sequence, 50 parallel requests were made for the same 500 data records. The requests were ramped up in 5 seconds. Again, all servers performed adequately with average, shortest, and longest response times of 70250, 289, and 361489 milliseconds, respectively.

Table 4 – 500 Record Parallel Full-Load Testing

50 Parallel Requests for 500 Data Records

	AOOS	CaroCOOPS	CDIP	CSC	GoMOOS	SWFSC	USF
Average	138113	21530	245459	32176	20556	16479	17437
Min	3497	375	185221	1562	289	366	640
Max	361489	101840	271400	101877	93677	40802	117422

Findings

The following findings are a compilation of the DTL staff and external partner responses to items three through fourteen of the testing plan (see Appendix A).

Assess the relative ease or difficulty of database modification to support the service installation

The approach used for service installation was for the data provider to create a database view with a specific name and with specific columns and data types. The Web application used a preformatted SQL statement to pull the required data from the database. Typical database administration skills and knowledge were required.

The relative ease or difficulty of service installation was generally tied to the ability to create the required view. Some partners did not manage their data in a Relational Database Management System (RDBMS). Others who use RDBMS's had built their table structures in a manner that prohibited the creation of the required view. Generally, this incompatibility resulted from a provider having designed a database table structure from a sensor-centric perspective, where each sensor was given a table. To support queries constrained by a bounding box and time range, serial queries were required across the numerous sensor tables. The partners who encountered this issue generally were not successful in installing the service on their main data stores.

One partner employed a database brand and version that did not support database views. For those, installation was difficult or impossible.

Assess the relative ease or difficulty of installation of server software

For those partners whose databases supported views and whose table structure facilitated the creation of the required view, installation was found to be quick and straightforward for both the Perl and Java versions of the service.

Assess the relative ease or difficulty of installing and configuring any supporting software or libraries

For those partners who were successful in installing the service, the software and library requirements were typical for the version of the service selected. The Perl

Common Gateway Interface (CGI) utilized the traditional CGI, XML::LibXML, and DBI::DBD modules that are widely available on any system supporting the Perl language, or that can be installed using the standard module installation process employed by most Perl modules. Similarly the Java version utilized typical database connection mechanisms that were widely supported by Java Servlet engines.

How would you rate the maintenance burden for the server application including

a) The relative ease of difficulty of routine log file management

Project partners did not find the log file maintenance to be an issue. The service configuration file provided for the specification of a log file location. The file format was concise and file growth was slow.

b) The relative ease or difficulty of modification in the event of evolving data models and standards. Consider the degree of intrusiveness of the transport layer into the data storage facility.

Project partners expressed little concern over the possibility of a changing data model, which is to be expected in the IOOS arena. Also, it was pointed out by one partner that the schema employed by the microWFS was, at the same time, based on widely recognized standards (GML and SFP) and fairly simplistic. If further development using GML and the SFP via WFS is to be pursued, a more robust schema would likely be required.

Assess the cost of server implementation (software, infrastructure, and human resources)

All partners who were able to install the service used existing infrastructure. One partner reported having to install the PostgreSQL database software bundle, which is a free open source package.

Assess the cost of client implementations (software, infrastructure, and human resources)

Any application capable of making an HTTP request can access data served by the microWFS. However, because this WFS implementation and its associated XML schema were developed solely for the purposes of exercising the use of GML/SFP via a WFS, it has no client applications. Even so, the development of an XML parser or XSLTs to reformat the microWFS schema would be very straightforward.

Assess the use of GML in the service and how it supports data aggregation/integration. Does the service use the current or a recent version?

The schema works well with fixed and moving platforms. There is an unfortunate trade-off to very good data aggregation and integration in that the GML is very verbose. The same can be said for Keyhole Markup Language (KML) for Google, so they simply use Zip compression and created KMZ. The use of a messaging wrapper could facilitate creating GMZ (or compressed GML). GML, like KML, is highly compressible.

Assess the general strategy to access data (e.g., via time and location ranges or via individual platform/sensor queries)

a) Is the content of the output record adequate for the variable tested?
b) Is the structure of the output record adequate for the variable tested?
c) Does the service support filtering by a geographic extent?
d) Does the service support filtering by a measured variable (waterTemperature, Salinity)
e) Does the service support filtering by a time instant or range?
f) Does the service respond appropriately to malformed requests?

The content of the microWFS XML schema was found to be generally adequate.

The structure of the schema was also generally found to be adequate. However, external partners expressed some concern over the location of XML elements such as the geographic datums, which were at the top of the hierarchical structure. This forced the data providers, in cases where their data might not always be stored in the same datum, to convert to the schema global datum for the service response. The DTL understands that the schema was simple and concise and that if it were to be the basis for broader work that it would require further deliberate work to extend it appropriately. The service provided appropriate filtering by geographic extent (bounding box), time range, and observed property. External partners expressed the need for better error messages when malformed requests are sent.

Are the dictionaries (vocabularies and ontologies) appropriate?

The dictionaries employed by the test were found to be adequate. Many efforts are underway in the IOOS community to develop appropriate data dictionaries.

Special note should be made of the Marine Metadata Initiative, which has developed a system (Tethys) to map compatible ontologies to each other.

Assess the support for server technology inside and outside of the IOOS community (e.g., continued development, documentation, and user support)

The microWFS service implementations were created for the sole purpose of exploring the use of GML and the Simple Feature Profile via a WFS to handle time series in-situ physical and chemical data. While it is based on recognized standards (OGC GML and WFS), it is not intended to be used beyond this project. Even so, this project has demonstrated that a simple WFS implementation supporting geographic extent, time, and observed property filters could be a very capable and useful data transport mechanism with the IOOS context.

Assess the availability of or support for development of clients inside and outside of the IOOS community

Since the microWFS service was written specifically for this project and since it has no intended or expected use beyond this project, the need to develop client software is not expected.

Assess the relative ease of use with conventional or popular client package(s)

If sufficient value were found in the microWFS design and code base to warrant further development, it could easily be expanded to be a fully compliant WFS. As a fully compliant WFS, many GIS applications could access and utilize the service directly.

Assess the strategy used to implement the OGC specifications. Is the representation of the response for each method (getCapabilities, describeSensor, and getObservation) logical, scalable, and easily used?

As previously mentioned, the microWFS does not implement the describeSensor or getCapabilities operations. The general approach used in the development of the XML schema and the structure of the HTTP request are logical, scalable, and straightforward.

Supports of other IOOS data exchange needs – Discovery

The microWFS servers are not intended as registries of services or data streams. The narrow scope of this project neglects the issue of data discovery.

Supports of other IOOS data exchange needs – Scalability

As previously stated, the microWFS was intended for use in this project only. However, the general service-oriented architecture approach lends itself to supporting multiple data providers. The WFS specification appears to be robust and capable of supporting IOOS data transport needs. The design and code base of the microWFS could be leveraged in future IOOS data transport design and implementations.

The GML and Simple Feature Profile XML dialects provide a rich and powerful language with which to describe and transmit data.

Supports of other IOOS data exchange needs – Extendibility

As mentioned in the previous section, the GML dialect of XML is rich and flexible. Extension of the current schemata to handle new content and new observed properties would not be trivial but could be straightforward.

Supports of other IOOS data exchange needs – Security

As implemented for this DTL project, the microWFS services were typical CGI or Java Servlet Web server applications. As such they present the same threat to the Web server that is presented by any Web application. Further, there are no security or data access controls built into the microWFS software implementations.

Since the expected users for the services installed for the DTL project were known and limited, there was little risk of denial of service (DOS) attacks. However, if this approach were to be employed as a permanent IOOS data-sharing resource, some strategies would have to be employed to prevent DOS attacks. Those strategies could include restricted IP or domain access to each server, limited time or geographic scale in the requests, and limited observed properties reported.

Supports of other IOOS data exchange needs – Metadata Access

The need of data users to assess the character and quality of available data requires that informative metadata exist and be accessible for the holdings. However, while there is general concurrence within the oceanographic and coastal sciences communities to use Federal Geographic Data Committee metadata or International Standards Organization (ISO)-formatted metadata, there is no national leadership on how to implement the use of metadata with a data service. The GML schemata associated with SOS services is capable of containing metadata or pointers to separate metadata services or documents.

Supports of other IOOS data exchange needs – Mission Critical Reliability

The objective load testing and more subjective critiques of the use of GML SFP and WFS as implemented in the microWFS were found to be stable and capable of supporting significant data flows. Both the CGI and Java Servlet Web application environments are well understood and are used widely in industry.

Supports of other IOOS data exchange needs – Data integrity (transactions, error checking missing/redundant/out of range data)

No error checking was performed on any of the servers beyond the immediate proof of function upon installation of the services at each location. The DTL staff performed a validation of the data provided by the NOAA Coastal Services Center (CSC) service installation and determined that the data content was complete and accurate when the results of representative requests were compared directly to the database contents.

Recommendations

The objectives of this project were to explore the use of GML and the Simple Feature Profile to contain time series in-situ physical and chemical oceanographic data and to determine the validity of using the OGC WFS as a service layer. The findings of this project show GML with the Simple Feature Profile to be a competent XML sub-dialect in dealing with the specified data in the described use context. Further, the limited development of the microWFS code bases demonstrated that WFS services can be designed and built with reasonable financial and human resource costs. Each of these two separately, and the two in combination, should be considered carefully for use as foundational elements of an IOOS data-sharing architecture.

The DTL recommends that the IOOS community continue to seriously consider GML SFP and WFS as leading candidate technologies for data transport.

Appendixes

Appendix A: Testing and Critique Plan

The Apache JMeter tool will be used to load servers with HTTP REST requests. JMeter is available from *http://jakarta.apache.org/jmeter/index.html*. The DTL staff will draft light-load and heavy-load JMeter test plans. The DTL External Partners (EP) will review and comment on the draft testing plan. Once there is consensus that the plan is reasonable and appropriate, the team will develop a testing schedule and pursue testing.

Testing Regimen:

Objective Measures

1. DONE – Measure the Light Load Request Response time for a single request using the Apache JMeter application. Conduct 10 Iterations and record the results. A basic JMeter test plan is available from the DTL. Question: Should each partner test against all other partners or should we do round where each tests several others? (I suggest skipping this step). Instead, I suggest that DTL partners provide their modified test plans back to the DTL so that the DTL can conduct remote tests of partner services.

2. DONE – Measure the Heavy Load Request Response time via Apache JMeter. The definition of heavy load is case specific. In general, the test should load the server to the point that requests are queued at the server. Repeat 100 iterations and record results. Calculate average, maximum, and minimum response times. A basic JMeter test plan for the heavy load requests is available from the DTL.

Note the differences between the planned items 1 and 2 and the actual load-testing scenarios described in the previous paragraph.

.

Subjective Measures

1. Assess the relative ease or difficulty of database modification to support the service installation

2. Assess the relative ease or difficulty of installation of server software

3. Assess the relative ease or difficulty of installing and configuring any supporting software or libraries

4. How would you rate the maintenance burden for the server application including
    a) The relative ease of difficulty of routine log file management

   b) The relative ease or difficulty of modification in the event of evolving data models and standards. Consider the degree of intrusiveness of the transport layer into the data storage facility.

5. Assess the cost of server implementation (software, infrastructure, and human resources)

6. Assess the cost of client implementations (software, infrastructure, and human resources)

7. Assess the use of GML in the service and how it supports data aggregation/integration. Does the service use the current or a recent version?

8. Assess the relative reliability of the service

9. Assess the general strategy to access data (e.g., via time and location ranges or via individual platform/sensor queries)
   a) Is the content of the output record adequate for the variable tested?
   b) Is the structure of the output record adequate for the variable tested?
   c) Does the service support filtering by a geographic extent?
   d) Does the service support filtering by a measured variable (waterTemperature, Salinity)
   e) Does the service support filtering by a time instant or range?
   f) Does the service respond appropriately to malformed requests?

10. Are the dictionaries (vocabularies and ontologies) appropriate?

11. Assess the support for server technology inside and outside of the IOOS community (e.g., continued development, documentation, and user support)

12. Assess the availability of or support for development of clients inside and outside of the IOOS community

13. Assess the relative ease of use with conventional or popular client package(s)

14. Assess the strategy used to implement the OGC specifications. Is the representation of the response for each method (getCapabilities, describeSensor, and getObservation) logical, scalable, and easily used?

15. Assess the degree to which the technology supports other IOOS data exchange needs for
   a) Discovery

b) Scalability
c) Extendibility
d) Security
e) Metadata Access to determine appropriate use contexts for a given data source.
f) Mission Critical Reliability
g) Data integrity (transactions, error checking missing/redundant/out of range data)

16. What were the general specifications and configuration of the test host(s)
    a) Number of CPUs
    b) Amount of RAM
    c) Network interface type/speed
    d) Relative load on host machine (was the machine more or less dedicated to this application?)

Appendix B: Project Partners

Partner	URL	Installed microWFS? (Y/N)	Tested microWFS? (Y/N)
Alaska Ocean Observing System	http://www.aoos.org/	Y	Y
Asia-Pacific Data-Research Center	http://apdrc.soest.hawaii.edu/	N	N
Carolina Coastal Ocean Observing System	http://www.carocoops.org	Y	N
Center for Coastal Margin Observation and Prediction	http://www.stccmop.org/	N	N
Coastal Data Information Program	http://cdip.ucsd.edu/	Y	Y
Great Lakes Observing System	http://glos.us/	N	N
Gulf of Maine Ocean Observing System	http://www.gomoos.org/	Y	N
Louisiana Universities Marine Consortium	http://www.lumcon.edu/	N	N
NOAA Coastal Services Center (DTL)	http://www.csc.noaa.gov/DTL/	Y	Y
NOAA IDEA Center	http://www.ideademo.org/	N	N
NOAA Southwest Fisheries Science Center	http://swfsc.noaa.gov/	Y	Y
Pacific Disaster Center	http://www.pdc.org/iweb/pdchome.html	N	N
University of South Florida	http://www.usf.edu/index.asp	Y	Y

Appendix C: microWFS Schema

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!-- W3C Schema generated by NOAA Coastal Services Center 2006-->
<!-- time series insitu schema for DTL testing                                         -->
<!-- Primary editor: Daniel Martin. Last updated 2007-06-20               -->
<!--  -->
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:ioos="http://www.csc.noaa.gov/ioos"
xmlns:gml="http://www.opengis.net/gml" xmlns:gmlsf="http://www.opengis.net/gmlsf"
targetNamespace="http://www.csc.noaa.gov/ioos" elementFormDefault="qualified" version="1.0">
	<xs:annotation>
		<xs: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/gmlsfProfile/1.0.0/gmlsf.xsd</
gmlsf:GMLProfileSchema>
		</xs:appinfo>
	</xs:annotation>
	<!--  -->
	<!-- includes and imports -->
	<xs:import namespace="http://www.opengis.net/gml"
schemaLocation="http://schemas.opengis.net/gml/3.1.1/base/gml.xsd"/>
	<xs:import namespace="http://www.opengis.net/gmlsf"
schemaLocation="http://schemas.opengis.net/gml/3.1.1/profiles/gmlsfProfile/1.0.0/gmlsfLevels.xsd"/>
	<!--  -->
	<!-- define complex feature for the time series values -->
	<xs:complexType name="TSMeasurementPropertyType">
		<xs:sequence>
			<xs:element ref="ioos:TSMeasurement"/>
		</xs:sequence>
	</xs:complexType>
	<xs:element name="TSMeasurement">
		<xs:complexType>
			<xs:sequence>
				<xs:element name="obsDateTime" type="xs:dateTime"/>
```

```
                              <xs:element name="observation" type="gml:MeasureType"/>
                        </xs:sequence>
                  </xs:complexType>
            </xs:element>
            <!-- define feature types -->
            <!-- the gml:id shall reflect a unique obsID value from originating data source -->
            <!-- suggested coordinate axis order is Y,X as defined by the default srsName value -->
            <xs:element name="insituTimeSeries" type="ioos:insituTimeSeriesType"
substitutionGroup="gml:_Feature"/>
            <xs:complexType name="insituTimeSeriesType">
                  <xs:complexContent>
                        <xs:extension base="gml:AbstractFeatureType">
                              <xs:sequence>
                                    <xs:element name="sensor">
                                          <xs:complexType>
                                                <xs:simpleContent>
                                                      <xs:restriction base="gml:CodeType">
                                                            <xs:attribute name="codeSpace"
type="xs:anyURI" use="optional" default="http://csc.noaa.gov/ioos/dictionaries/SensorDictionary.xml"/>
                                                      </xs:restriction>
                                                </xs:simpleContent>
                                          </xs:complexType>
                                    </xs:element>
                                    <xs:element name="observationName">
                                          <xs:complexType>
                                                <xs:simpleContent>
                                                      <xs:restriction base="gml:CodeType">
                                                            <xs:attribute name="codeSpace"
type="xs:anyURI" use="optional" default="http://csc.noaa.gov/ioos/dictionaries/PhyOceanDictionary.xml"/>
                                                      </xs:restriction>
                                                </xs:simpleContent>
                                          </xs:complexType>
                                    </xs:element>
                                    <xs:element name="verticalDatum">
                                          <xs:complexType>
                                                <xs:simpleContent>
                                                      <xs:restriction base="gml:CodeType">
```

```
                                                                        <xs:attribute name="codeSpace"
type="xs:anyURI" use="optional" default="http://csc.noaa.gov/ioos/dicitonary/VerticalDatumDicitonary.xml"/>
                                                                </xs:restriction>
                                                        </xs:simpleContent>
                                                </xs:complexType>
                                        </xs:element>
                                        <xs:element name="verticalPosition" type="gml:MeasureType"/>
                                        <xs:element name="horizontalPosition" type="gml:PointPropertyType"/>
                                        <xs:element name="tsEvent" type="ioos:TSMeasurementPropertyType"
maxOccurs="unbounded"/>
                                </xs:sequence>
                        </xs:extension>
                </xs:complexContent>
        </xs:complexType>
        <!-- define feature collection -->
        <xs:element name="insituTimeSeriesCollection" type="ioos:insituTimeSeriesCollectionType"
substitutionGroup="gml:_GML"/>
        <xs:complexType name="insituTimeSeriesCollectionType">
                <xs:complexContent>
                        <xs:extension base="gml:AbstractFeatureType">
                                <xs:sequence maxOccurs="unbounded">
                                        <xs:element name="featureMember">
                                                <xs:complexType>
                                                        <xs:sequence>
                                                                <xs:element ref="gml:_Feature"/>
                                                        </xs:sequence>
                                                </xs:complexType>
                                        </xs:element>
                                </xs:sequence>
                        </xs:extension>
                </xs:complexContent>
        </xs:complexType>
</xs:schema>
```

Appendix D: microWFS Database View Definition

dbo.vwMicroWFS

fieldName	data type	example	description
platformID	int	100	unique identifier for a collection device issued by observatory: for database implementation
orgID	int	1	unique value issued by CSC
platformName	varchar255	Buoy2002.SensorABC	informative name provided by observatory
obsName	varchar255	waterTemperature	use the dictionary to obtain valid values
vDatum	varchar100	MHHW	use the dictionary to obtain valid values
vPosition	float	-10.2	meters reletive to datum
latitude	float	42.345	
longitude	float	-72.456	
obsDateTime	dateTime	yyyy-mm-dd hh:mm:ss	in UTC reference
obsValue	float	21.23	
stdUnitName	varchar255	celsius	use the dictionary to validate, but get value from from database if available