CUAHSI
universities allied for water research

CUAHSI WaterML 1.1

Draft Specification

Part 2: Changes compared with WaterML 1.0

June 5, 2009

by:

David Valentine
Ilya Zaslavsky
San Diego Supercomputer Center
University of California at San Diego
San Diego, California, USA

Distribution

Funding and acknowledgements

Funding for this document was provided by the Consortium of Universities for the Advancement of Hydrologic Science, Inc. (CUAHSI) under NSF Grant No. EAR-0413265. In addition, much input and feedback has been received from the CUAHSI Hydrologic Information System development team. Their contribution is acknowledged here.

We would also like to thank partner agency personnel from USGS (Water Resource Division), EPA (the STORET team), and NCDC, as well as data managers and personnel of hydrologic observatory testbeds for cooperation, discussions and insightful feedback. We are especially grateful to the USGS and NCDC teams, and other partners who implemented WaterML-compliant web services over their repositories.

Scope

Water Markup Language (WaterML) specification defines an information exchange schema, which has been used in water data services within the Hydrologic Information System (HIS) project supported by the U.S. National Science Foundation, and has been adopted by several federal agencies as a format for serving hydrologic data. The goal of the first version of WaterML was to encode the semantics of hydrologic observation discovery and retrieval and implement water data services in a way that is both generic and unambiguous across different data providers, thus creating the least barriers for adoption by the hydrologic research community. Now in version 1.1, WaterML is evolving to reflect the deployment experience at hydrologic observatory testbeds around the U.S., and U.S. federal and state agency practices of serving observational data on the web. Data sources that can be queried via WaterML-compliant water data services include many national and international repositories of water data, and a growing number of academic observation networks registered by researchers associated with the hydrologic observatories.

WaterML 1.0 specification was published as an OGC discussion paper in 2007, and is available at the OGC web site. WaterML 1.1 is an updated version developed during 2008-2009, based on the feedback from HIS 1.0 deployment.

The WaterML 1.1 specification consists of three parts. The first part is a high-level description of WaterML scope, rationale, context and design drivers, main trade-offs in WaterML development, the evolution of WaterML, and the core WaterML constructs. This first part follows a paper by Valentine, Zaslavsky and Whiteaker "CUAHSI WaterML: Design Drivers and Evolution Towards OGC Standards" (2009), currently in review. The second part (this document) reviews changes in WaterML 1.1 compared to the previous published specification. The third part is a detailed technical description of WaterML 1.1 schema.

Support and questions

Contact Dr. David Valentine, SDSC, valentin@sdsc.edu

Table of Contents

SCOPE

This document summarizes WaterML design changes as it evolves from version 1.0 to 1.1, and 2.0. The document starts with detailed project planning for evolving WaterML towards 1.1 and then to an OGC-compliant version (referred to as WaterML 2.0). The core of the document is a listing of specification change requests as expressed by the CUAHSI HIS team and external partners, For each change request, the target implementation version (either 1.1 or 2.0) is proposed, and risks (of breaking client applications, or other uncertainties) are outlined.

GOALS OF INFORMATION MODEL FOR HYDROLOGIC OBSERVATIONS, AND WATERML DEVELOPMENT:

- Maintain semantic information outlined in the CUAHSI Hydrologic Observations Data Model paper
- Create independent conceptual model of Hydrologic Observations
- Move towards OGC Observations and Measurements

BENEFITS OF MOVING TOWARDS OGC STANDARDS:

- Standardize on an information model that can be used for handling both hydrologic time series and hydrologic themes, and potentially other use cases
- Compatibility with GIS software and other COTS software
- Easier cross-domain adoption (within GEOSS)
- No longer need to write CUAHSI services. Utilize OGC service interfaces.

RISKS:

- Loss of understanding and community acceptance
  - Mitigation: Communication, provide API tools and examples
- Difficulty of use, as namespaces, URNs, and generic and flexible notions make it more complex and less domain-oriented
- Difficulty of moving community to new standard
- Possible divergence from the CUAHSI Hydrologic Observations Data Model
- Expectations of CUAHSI Partners

ISSUES

- 20 questions/Use Case issues: we need to figure out usage scenarios and use cases that the data encoding should support
- What are the expectations of the CUASHI Partners, such as USGS and EPA: often these requirements to a data exchange standard are not well verbalized and are rooted in data handling and analysis practices of each agency

PLANNING FOR WATERML UPGRADES

PROPOSED PLAN:

1) Finalize WaterML 1.1 specification
2) Finalize WOF 1.1 services, including examples for method signatures (use c# interface classes), and a generic ODM service
3) Determine future requirements for future WaterML by gathering use cases, reviewing how they are expressed in other data exchange standards or practices, and using this information to derive requirements
4) In parallel, develop a WaterML 2.0, which is OpenGIS services compliant

PROJECTS/TASKS

WATERML 1.1

GOAL

- Expose additional information from the Observations Data Model 1.1
- Address issues with fixed code lists/enumerations, eg ODM “Controlled Vocabularies” DataType, ValueType ,GeneralCategory
- Make changes that improve consistency

RISKS:

- Breaking client applications
    - To avoid breaking present applications, an additional web service that returns the 1.1 schema will be created.
- Changes for Consistency
    - Remove any dependence on ID's; use codes instead (e.g. siteCode, variableCode)

BASIC CHANGES

- Changes for Use Consistency (CR#a1)
- Add sample and lab sample (CR#a2)
- Make extensibility of Site, Variable, Sites simpler, and clearer. (CR#1)
- Specify how multiple qualifiers should be done (CR#2)
- Make attribute value/@count explicitly optional (CR#3)
- Add additional information on site type to site information (CR#4)
- Add speciation (cr#5)
- Address time support issues (CR#6)
- Make Units consistent (cr#9)

BREAKING CHANGES

- Expendable Enumerations (CR#7)
- Make <values> repeatable. (CR#8)

- Ensure naming consistency (CR#18)
- Make changes to values to for multiple time series: <values>(TsValuesSingleVariableType)
    - Multiple variables from one site (cr#21)
    - Allow for unit transformation values (cr#22)
    - Modifications to <timeSeriesResponse> that need to occur
        - (CR#21) Support Multiple variables response
        - (CR#8. waterml 1.1) Make <values> repeatable.
- Changes to how data values are handled (CR#29)
    - Codes and not identifiers (cr# a1)

- Repeatable NoDataValue
    - NoDataValue is a value to be interpreted by a client. Sometimes multiple NoDataValue codes may exist. These are streamed inside of a values list from a service (Ilya, Use case) , They may have the meaning of a censorCode, or a qualifier, but they are represented as a value.

WaterOneFlow 1.1

Goals

Standardize the naming, and avoid overloading the method.

Risks

Low risk A new endpoint that is separate from 1.0 will be used to send WaterML 1.1 over a WaterOneFlow 1.1 API.

WaterOneFlow 1.1

- Rename Web Service Method for Consistency (CR#10)
- GetSites method name (CR#11)
- GetSites by Box (Cr#29)
- Rename GetVariableInfo GetVariables (CR#12)
- Add Capabilities Endpoint or document (CR#13)
- Multiple Sites with SiteInfo (CR#28)
- ExposeMethods, Sources, and Vocabularies (CR#14)

ODM Services

Goals

ODM providers would like to expose groups, and information on derived data values. This is information that not every data source has, and would be difficult to expose in a markup language.

ODM Services for ODM 1.1 databases

- Additional service endpoints (Cr#17)
- Expose Groups, Derived from DataValues in Web Services (CR#18)

Conceptual Basis for Future Version of WaterML

Goals

- Provide an independent conceptual model that can be used for a variety of information that is useful to the hydrologic sciences
- Deliver information over WFS/WCS and/or Modified Water Web Services.

- Understand the implications of the change to the user community

WATERML 2.0/WOML

- Utilize existing OGC models to develop a UML model that can be converted to XML (Cr#18,19).
- Provide prototype samples that match the requirements and use cases.
- Deliver information over services (CR#16)
- Change how Data Values are handled (#CR23, 24,25,26)
- Make values use elements, and not attributes (cr#24)
- Time Precision (cr#25)
- Additional Data Types (CR#26)

RESOURCES

List of resources

COMMUNITY SPECIFICATION PROCESS

WaterML specification development should be a community process, going through a series of steps: submission of change requests, review of change requests, updates of the schema, documenting schema updates and publishing them for review, collecting feedback from CUAHSI HIS team and partners, and finalizing the schema. In parallel, development web services utilizing the new schema shall be developed, to allow developers and reviewers a better feel for the changes.

The following community resources will be used:

- Mailing lists
- Workspaces/Wiki

PROGRAMMING TOOLS

XML SCHEMA DATA BINDING

Adding multiple XML schema files means that coding becomes more complex.
SDSC has license for Liquid XML, and can distribute compiled XML data bindings for .net, java, and c

CHANGE LIST

Versions:

1_0 – Present, as specified in OGC document 07-041r1. http://www.opengeospatial.org/standards/dp

1_1- Basic changes, including ODM 1.1 compliance, conversion to elements, re-arrangements and consistency improvements.

2_0 - Object model based changes, consistent with next major version update.

CHANGE 0. OBJECT MODEL

Proposed Version: 2 -

Description: Develop a conceptual basis for a hydrologic markup language independent of ODM and WaterML. Use the semantic information from the ODM. Utilize the OGC UML models, and convert to XML. Provide prototype samples that match the requirements and use cases.

ODM central concepts are time-variable-space, implemented as Site, Variable, and observations values.
WaterML is service bases, and uses variables, site, series, and value lists.
OGC O&M has observations, measurements, and locations. (verify)
OpenMI (details)
Community Modeling Environment (details)

Risks:

CHANGE DETAILS:

To be determined.
This change requires independent investigation, and an independent task list.

CHANGE REQUEST A1. CONSISTENCY CHANGES

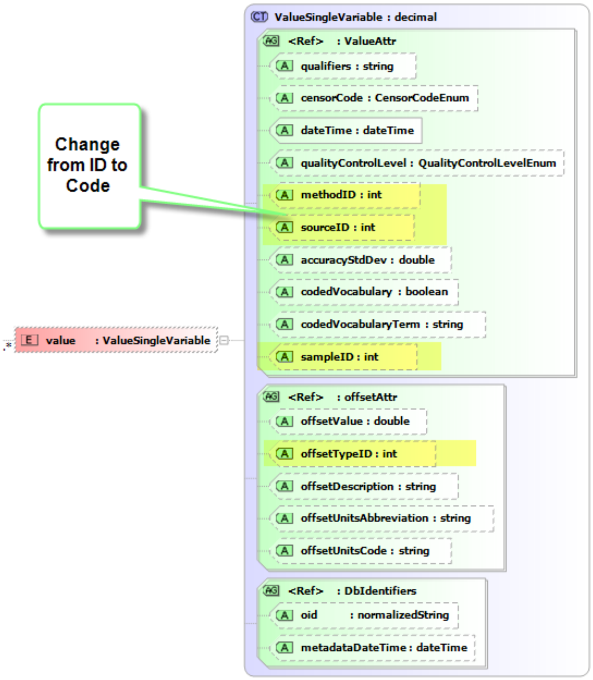
Proposed Version: 1_1

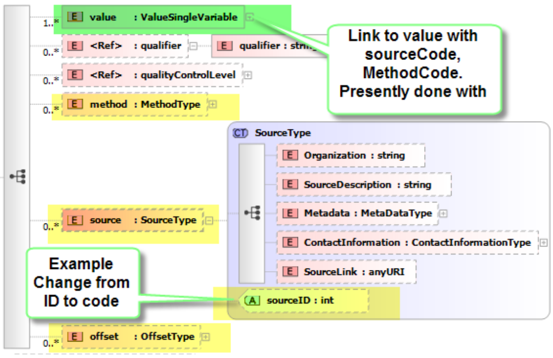
Description: Make changes that improve the consistency. For example, use codes as references between elements. And use consistent types.

Risks: Moderate. Programs will need to be changed to use Code, and not an ID as references

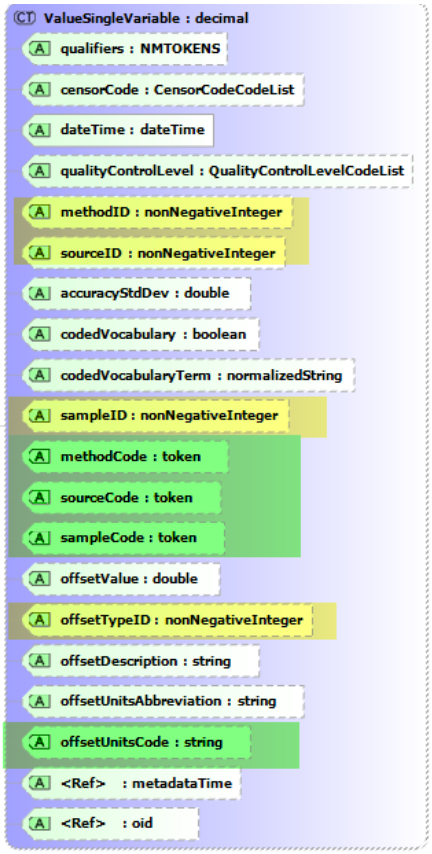
CHANGE DETAILS:

- Remove any dependence on ID's and use codes, instead
  - values/value/@methodID,@sourceID,@sampleID,@offsetTypeID
  - values/offset/@offsetTypeID
  - values/source/@sourceID
  - values/method/@methodID
  - values/samples/@sampleID
- Change attribute types to be consistent
  - to token for *Code (no returns, tabs, no runs of more than one space)
  - to normalizedString for others (no returns, tabs)


Change from ID to Code
ValueSingleVariable : decimal
<Ref> : ValueAttr
qualifiers : string
censorCode : CensorCodeEnum
dateTime : dateTime
qualityControlLevel : QualityControlLevelEnum
methodID : int
sourceID : int
accuracyStdDev : double
codedVocabulary : boolean
codedVocabularyTerm : string
sampleID : int
<Ref> : offsetAttr
offsetValue : double
offsetTypeID : int
offsetDescription : string
offsetUnitsAbbreviation : string
offsetUnitsCode : string
<Ref> : DbIdentifiers
oid : normalizedString
metadataDateTime : dateTime
value : ValueSingleVariable


Link to value with sourceCode, MethodCode. Presently done with

Example Change from ID to code


CT ValueSingleVariable : decimal
A qualifiers : NMTOKENS
A censorCode : CensorCodeCodeList
A dateTime : dateTime
A qualityControlLevel : QualityControlLevelCodeList
A methodID : nonNegativeInteger
A sourceID : nonNegativeInteger
A accuracyStdDev : double
A codedVocabulary : boolean
A codedVocabularyTerm : normalizedString
A sampleID : nonNegativeInteger
A methodCode : token
A sourceCode : token
A sampleCode : token
A offsetValue : double
A offsetTypeID : nonNegativeInteger
A offsetDescription : string
A offsetUnitsAbbreviation : string
A offsetUnitsCode : string
A <Ref> : metadataTime
A <Ref> : oid

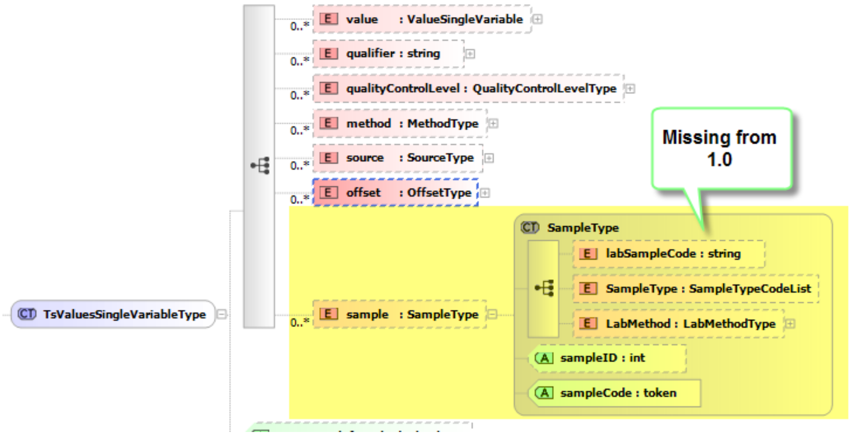
Change Request A2. Add Sample and Lab Sample

Proposed Version: 1_1

Description: Sample is not included in 1.1, although @sampleID can be on a value. @sampleCode should be use as a reference.

Risks: low.

CHANGE DETAILS:



CHANGE REQUEST 1. EXTENSIBILITY FIXES

Proposed Version: 1_1

Description: Make extensibility of Site, Variable, Sites simpler, and clearer.

- Make extensibility of Site, Variable, Sites simpler, and clearer.
    - Use OGC concept of “property” instead of note element.
    - Properties provide clearer communication by saying “siteProperty”, “State” is “California”
    - Additional elements
        - siteInfo/siteProperty
        - variable/variableProperty
        - series/seriesProperty

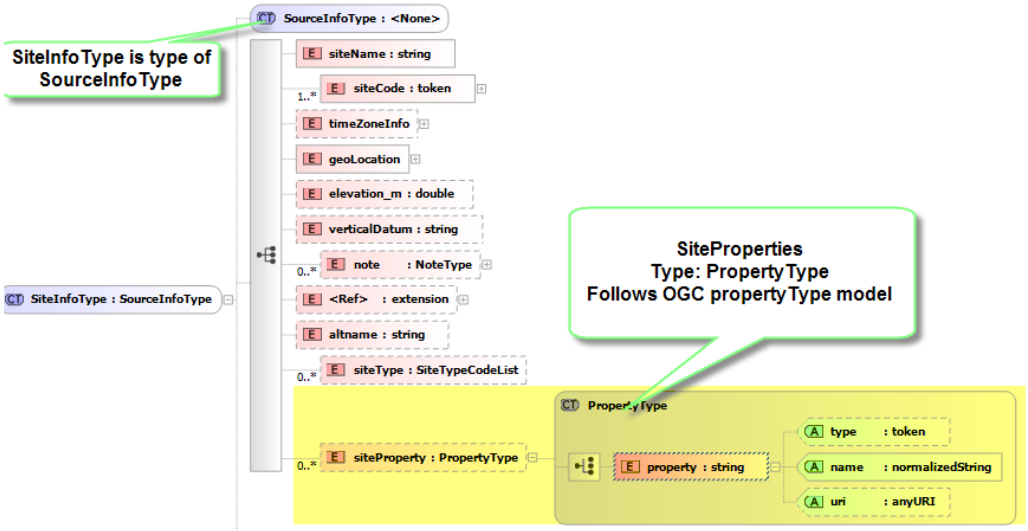
Risks:

CHANGE DETAILS:

- Make extensibility of Site, Variable, Sites simpler, and clearer.
    - Use OGC concept of “property” instead of note element.
    - Properties provide clearer communication by saying “siteProperty”, “State” is “California”

```
<siteProperty @name=’State’>California</siteProperty>
```

- Additional elements
    - siteInfo/siteProperty
    - variable/variableProperty
    - series/seriesProperty



Change Request 2. Specify Multiple qualifiers

Proposed Version: 1_1

Description: Specify how multiple qualifiers should be done. This will be accomplished by space delimiting qualifiers.

Risks: low. A string is a string.

Change Details:

Specify how multiple qualifiers should be done

- value/@qualifiers redefine as space delimited set of tokens.
- Change data type to MNTOKENS

```
<value @qualifies="usgs:A usgs:e annotation:X">1244</value>
```

Change Request 3. Explicity flag values@count as optional

Proposed Version: 1_1

Description: some programs have relied on that a count is included with the list of values. Services coded by third parties often do not include this... since sometimes the count may not be known in advance.
XML attributes are optional. **Explicitly specify this as attribute as optional**

Risks: medium. Need to communicate not to rely on this attribute. The length of the array is easily obtained.

Change Details:

```
<xsi:attribute name="count" type="positiveInt" use="optional">
    </xsi:attribute>
```

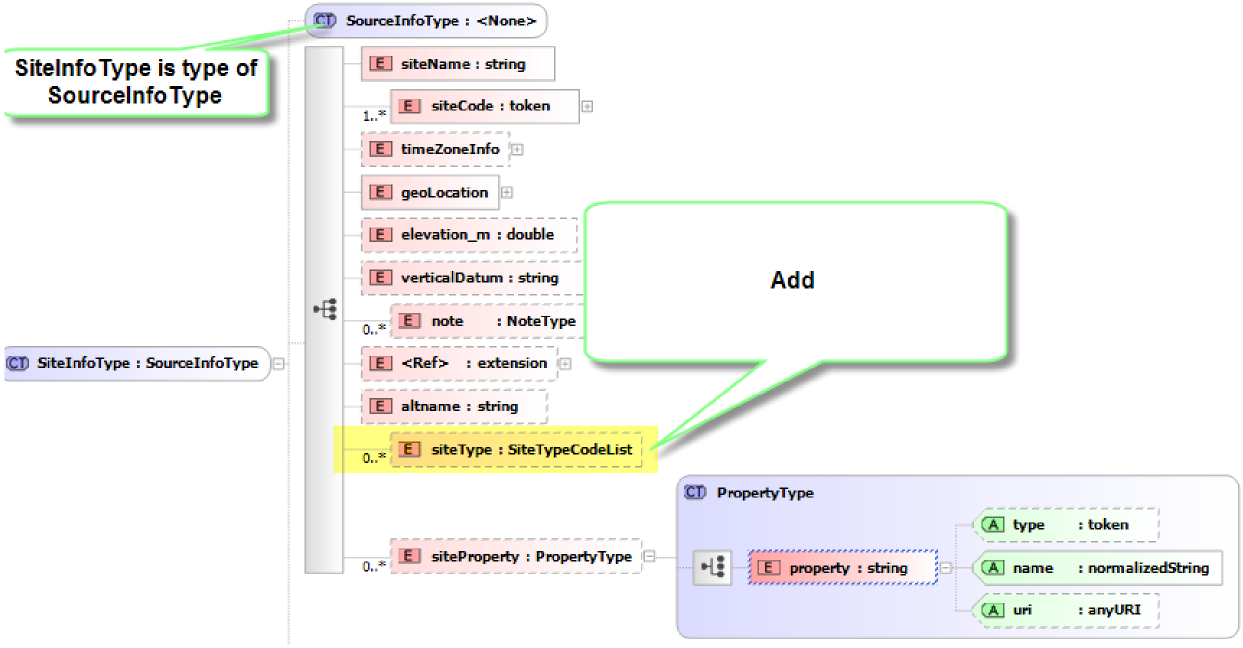
Change Request 4. Add siteType element

Proposed Version: 1_1

Description: SiteTypes are use in the USGS and EPA.
Eg. Suface water, ground water, estuary
They could be communicated with siteProperty, but if we want a suggested set of terms, then an element is best.

Risks: low. It might be more appropriate to communicate as a siteProperty, since it is not in ODM.

Change Details:


SiteInfoType is type of SourceInfoType

Add

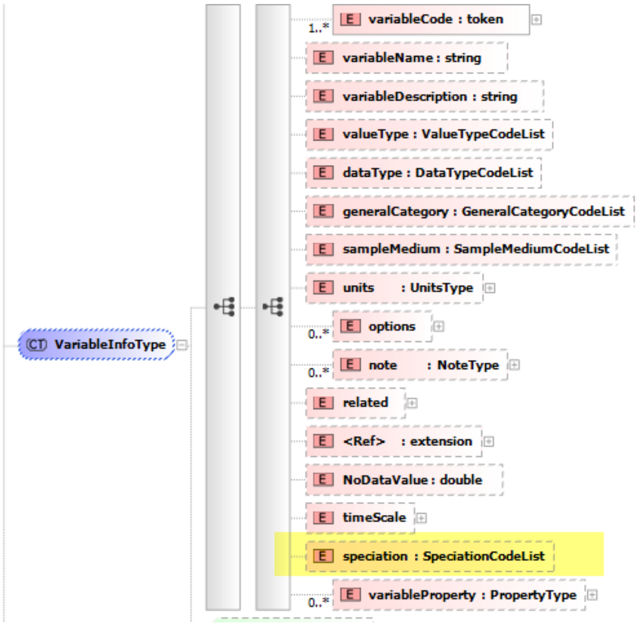
Change Request 5. Add Speciation

Proposed Version: 1_1

Description: Speciation is new column in ODM db schema. Add to variableInfo Type

Risks: low

CHANGE DETAILS:


CT VariableInfoType
1..* E variableCode : token
E variableName : string
E variableDescription : string
E valueType : ValueTypeCodeList
E dataType : DataTypeCodeList
E generalCategory : GeneralCategoryCodeList
E sampleMedium : SampleMediumCodeList
E units : UnitsType
0..* E options
0..* E note : NoteType
E related
E <Ref> : extension
E NoDataValue : double
E timeScale
E speciation : SpeciationCodeList
0..* E variableProperty : PropertyType

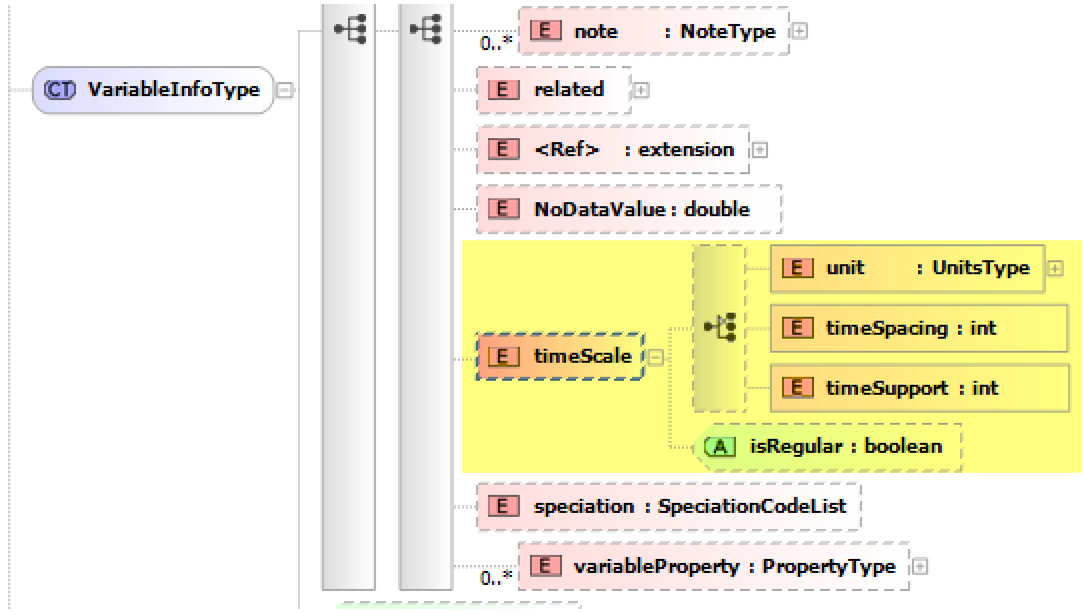
CHANGE REQUEST 6. ADDRESS TIME "SUPPORT" ISSUES

Proposed Version: 1_1

Description: Address issues with existing time support information. All dimensions need to be covered: timeSupport, timeSpacing, regularity.
A timeScale element is to be added to VariableInfoType, and timeSupport is to be dropped.
We will need to externally specify how clients are to use this element to determine time precision, and use, and check that our client code properly output the correct precision (eg YYYY-MM-DD, YYYY-MM-DDT00:00)

Risks: medium. Services need to coded to send out timeScale, and clients need to properly utilize it.

CHANGE DETAILS:



CHANGE REQUEST 7. EXPANDABLE ENUMERATIONS

Proposed Version: 1-1

Description:
Expendable Enumerations. Elements that were restricted to an enumerated list of values, are no longer restricted. Suggested lists of values are still included in the XML schema, but they are not enforced. Basically, all ODM CV elements become list of terms, plus the ability to add any string.

Risks: Medium. If a 1.0 service reads an unknown value, it will through an error. For 1.1 services, this will work, but any consistency between data sources relies on cooperation.

CHANGE DETAILS:

This is mainly an internal schema change, externally, all the CV's will look like strings.
Elements that were enumerations will be a union of the previous enumeration, and string. Basically, it will be treated as a string. Smart Clients may use the enumeration to display a list of known values. The example below uses CensorCode:

```
<xsi:simpleType name="CensorCodeCodeList">
    <xsi:union memberTypes="CensorCodeEnum xsi:string" />
</xsi:simpleType>

<xsi:simpleType name="CensorCodeEnum">
    <xsi:restriction base="xsi:string">
      <xsi:enumeration value="lt" />
      <xsi:enumeration value="gt" />
      <xsi:enumeration value="nc" />
      <xsi:enumeration value="nd" />
      <xsi:enumeration value="pnq" />
    </xsi:restriction>
  </xsi:simpleType>
```

Effects:

- CensorCode
- QaulityControlLevel
- SampleType
- ValueType
- SampleMedium
- Speciation
- TomepCatecory
- VerticalDatum
- Sitetype

Change Request 8. Make Values Repeatable

Proposed Version: 1_1

Description: A USGS site can have multiple streams of the same variable parameter from different instruments.

Station: NWISDV:02289050

Variable: NWISDV:00065 or NWISDV:00065/statistic=00003 or NWISDV:00065/ValueType=Average

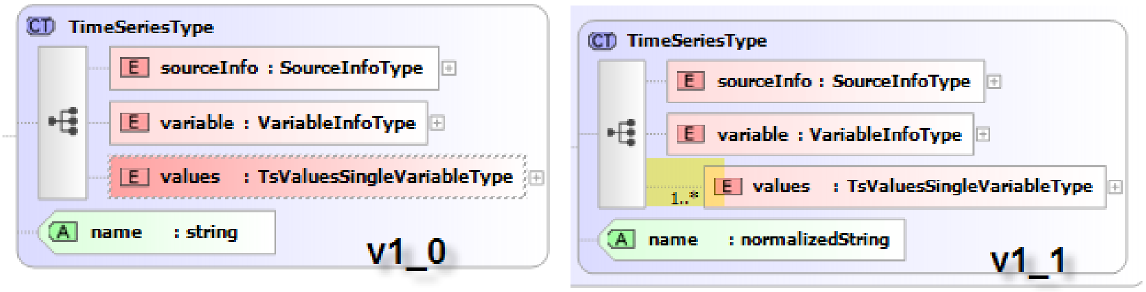
DateRange 2003-01-01 to 2004-01-01

```
     <ws:GetValuesObject>
         <ws:location>NWIS:02289050</ws:location>
         <ws:variable>NWIS:00065</ws:variable>
         <ws:startDate>2003-01-01</ws:startDate>
         <ws:endDate>2004-01-01</ws:endDate>
         <ws:authToken>?</ws:authToken>
      </ws:GetValuesObject>
```

Risks: medium. Clients must understand that multiple value lists can be returned.
Clients hand coded to the XML for the path may only access the first instrument
Clients objects compiled from WSDL should handle this.

CHANGE DETAILS:

Change Cardinatlity to allow for more than one:


CT TimeSeriesType
E sourceInfo : SourceInfoType
E variable : VariableInfoType
E values : TsValuesSingleVariableType
A name : string
v1_0
CT TimeSeriesType
E sourceInfo : SourceInfoType
E variable : VariableInfoType
1..*
E values : TsValuesSingleVariableType
A name : normalizedString
v1_1

WaterML 1.1

```
         <timeSeriesResponse xmlns="http://www.cuahsi.org/waterML/1.0/">
            <queryInfo>
               <creationTime>2008-09-04T18:35:52.191-04:00</creationTime>
               <criteria>
                  <locationParam>USGS:02289050/agency=USGS</locationParam>
                  <variableParam>USGS:00065/statistic=00003</variableParam>
                  <timeParam>
                     <beginDateTime>2003-01-01</beginDateTime>
                     <endDateTime>2003-01-01</endDateTime>
                  </timeParam>
               </criteria>
            </queryInfo>
            <timeSeries>
               <sourceInfo xsi:type="SiteInfoType">
                  <siteName>TAMIAMI CANAL AT S-333 NR MIAMI, FL</siteName>
                  <siteCode network="NWISDV"
siteID="2380231">02289050</siteCode>
                  <timeZoneInfo>
                     <defaultTimeZone ZoneAbbreviation="EST" ZoneOffset="-
05:00"/>
                     <daylightSavingsTimeZone ZoneAbbreviation="EDT"
ZoneOffset="-04:00"/>
                  </timeZoneInfo>
                  <geoLocation>
                     <geogLocation xsi:type="LatLonPointType"
srs="EPSG:4269">
                        <latitude>25.76121208</latitude>
                        <longitude>-80.6739499</longitude>
                     </geogLocation>
                  </geoLocation>
                  <note>Agency:USGS</note>
               </sourceInfo>
               <variable>
                  <variableCode vocabulary="NWISDV">00065</variableCode>
                  <variableName>Gage height</variableName>
                  <variableDescription>Gage height,
feet</variableDescription>
                  <valueType>Derived Value</valueType>
                  <dataType>Average</dataType>
                  <units unitsAbbreviation="ft">feet</units>
                  <options>
                     <option name="Statistic"
optionCode="00003">Mean</option>
                  </options>
                  <NoDataValue>-999999</NoDataValue>
                  <timeSupport isRegular="true">
                     <unit>
                        <UnitName>day</UnitName>
                        <UnitType>Time</UnitType>
                        <UnitAbbreviation>d</UnitAbbreviation>
                     </unit>
                     <timeInterval>1</timeInterval>
                  </timeSupport>
               </variable>
               <values count="1">
                  <value dateTime="2003-01-01T00:00:00"
qualifiers="A">10.03</value>
                  <qualifier qualifierCode="A" network="USGS"
vocabulary="dv_rmk_cd">Approved for publication -- Processing and review
completed.'</qualifier>
                  <method methodID="2">
                     <MethodDescription>sensor:</MethodDescription>
                  </method>
               </values>
               <values count="1">
                  <value dateTime="2003-01-01T00:00:00"
qualifiers="A">7.48</value>
                  <qualifier qualifierCode="A" network="USGS"
vocabulary="dv_rmk_cd">Approved for publication -- Processing and review
completed.'</qualifier>
                  <method methodID="8">
                     <MethodDescription>sensor:DOWNSTREAM
PUBLISHED</MethodDescription>
                  </method>
               </values>
            </timeSeries>
         </timeSeriesResponse>
```

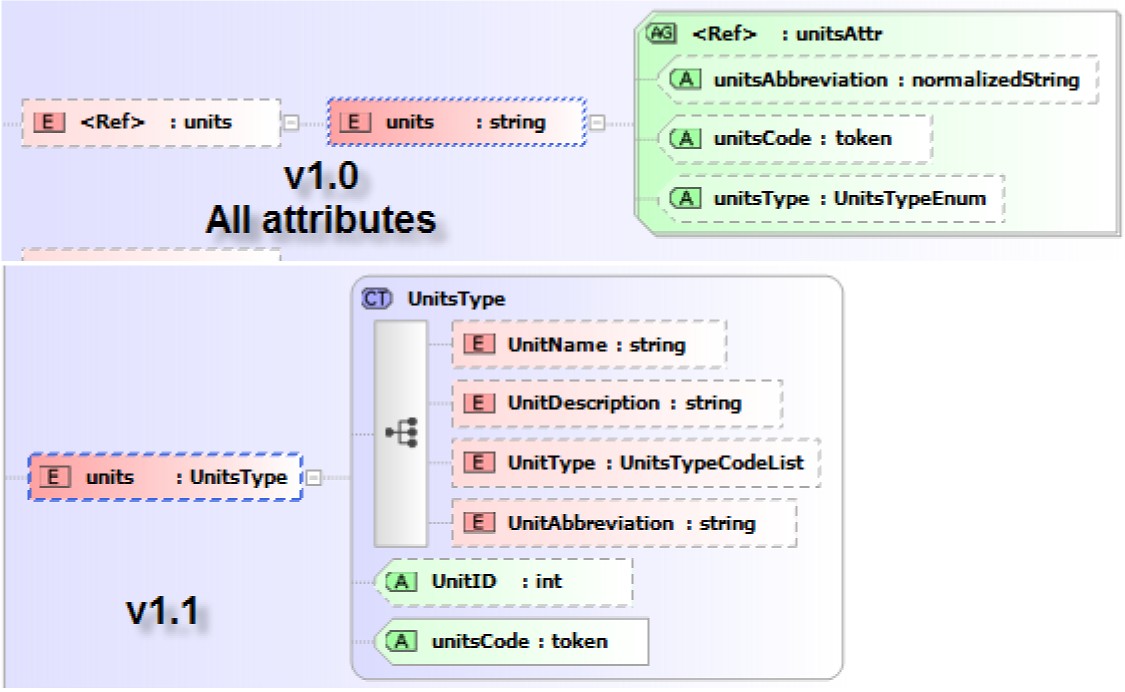
CHANGE REQUEST 9. STANDARDIZE UNIT ELEMENTS

Proposed Version: 1_1

Description: A units type was added as a way to standardize the way units are communicated. The original “units” element in variables need to be changed

Risks: high. While name “units’ would be the same, the units element would contain elements, and not attributes.

Change Details:



Change Request 10. Rename Web Service Method for Consistency

Proposed Version: webservices 1_1

Description: The base names are GetValues,GetVariableInfo,GetSiteInfo.
The“Object” methods are really the more SOAP-like, GetValuesObject, GetVariableInfoObject, GetSiteInfoObject
Whereas, the base names in Web Services 1.0 are really “String” They take the object, and write out a string.

GetSites and GetSitesXml are incorrectly named. See CR#10
Do we need to rename GetVariableInfo GetVariables. See CR#11

Risks: Conversation. We talk GetValues... not GetValuesXXXX

CHANGE DETAILS:

WATERONEFLOW METHOD RENAMING (ASLO SEE CR#10 AND CR#11)

Method v1	Method v 1.1/2	
GetSites	GetSitesObject	rename
GetSitesXML	GetSitesString	rename
GetValues	GetValuesString	rename
GetVaraibleInfo	GetVariablesString	rename
GetVariableInfoObject	GetVariablesObject	rename
GetSiteInfo	GetSiteInfoString	rename
	GetCapabilities	add

CHANGE REQUEST 11. FIX GETSITES METHOD NAME

Proposed Version: webservices 1_1

Description: GetSites and GetSitesXml are incorrectly named.

Risks: low.

CHANGE DETAILS:

WATERONEFLOW METHOD RENAMING

Method v1	Method v 1.1/2	
GetSites	GetSitesObject	rename
GetSitesXML	GetSitesString	rename

Change Request 12. Rename GetVariableInfo GetVariables.

Proposed Version: webservices 1_1

Description: Conversationally, we have been saiying GetVariables()... should we standardize

Risks: low.

Change Details:

Method v1	Method v 1.1/2	
GetVaraibleInfo	GetVariablesString	rename
GetVariableInfoObject	GetVariablesObject	rename

Change Request 13. Add Capabilities Endpoint or document

Proposed Version: webservices 1_1

Description: We may have different endpoints(service versions, or services). If we have a standard format for communicating this information, and put it at a standard location then clients could look for it.
This document would describe the various services, and their capabilities.
Services could have a method that returned the capabilities document, and could alert clients to other services.

Risks: Medium. 1.0 service does not have a capabilities method
Other data providers will need to implement it.

Change Details:

Need document format

WaterOneFlow method

Method v1	Method v 1.1/2	
	GetCapabilities	add

Change Request 14. Expose Methods, Sources, and Vocabularies

Proposed Version: webservices 1_1

Description: Sources, Methods, and Vocabularies should be exposed so that clients can harvest the information, without having to go through GetValues.

Risks: low

Change Details:

TBD.
Need to review document from DT, and provide functionality specification. Need to insure that information for sources, methods is exposed. Need to add vocabulary response to WaterML, if this is needed.

Change Request 15. Expose Groups, Derived from DataValues in Web Services

Proposed Version: webservices 1_1

Description: Data in an ODM should be exposed as outlined at:
http://river.sdsc.edu/wiki/Exposing%20full%20ODM%20content%20in%20Web%20Services.ashx

Risks: TBD.
Adds complexity to services.
Functionality is only for a single type of data source

Change Details:

TBD

Change Request 16. Open GIS Mappings

Proposed Version: Unknown

Description: Possible mappings to OpenGIS methods in web feature services, and Sensor services needs to be investigated

Risks: TBD

CHANGE DETAILS: TBD

CHANGE REQUEST 17. ADDITIONAL SERVICE ENDPOINTS

Proposed Version: 1_1

Description: Data in an ODM should be exposed. Should this be an service. What other possible service may be needed.

Risks: Dependencies on CR 14 and 15

CHANGE DETAILS:

CHANGE REQUEST 18. MAKE WATERML SIMPLE GML COMPLIANT

Proposed Version: 2

Description: Make waterML able to be use in services that understand GML.
This involves more than just using GML geometries for geographic information.
The responses need to derive from the abstract type, and the links to communicate where the GML is hiding need to be added.

Risks: extreme. We may not be able to map all responses into GML. GetSite and GetSiteinfo sites responses would map, but time series may be more difficult.

Could be a complex process, that

Change Details:

TBD. Needs planning.
Need to prototype a sitesResponse simpleGML schema.
Remove LatLonBox, LatLonPoint replace with GML equivalents.

Change Request 19. Use Simple GML for the Geometries

Proposed Version: 2

Description: Use GML for the geographic information elements.
Benefits are than we can describe line, polygons and other objects. But we really don't have those in our databases.

Risks: high. Namespaces are introduced. Documentation will need to be changed to handle getting information from elements with namespaces.

Change Details:

Remove LatLonBox, LatLonPoint replace with GML equivalents.

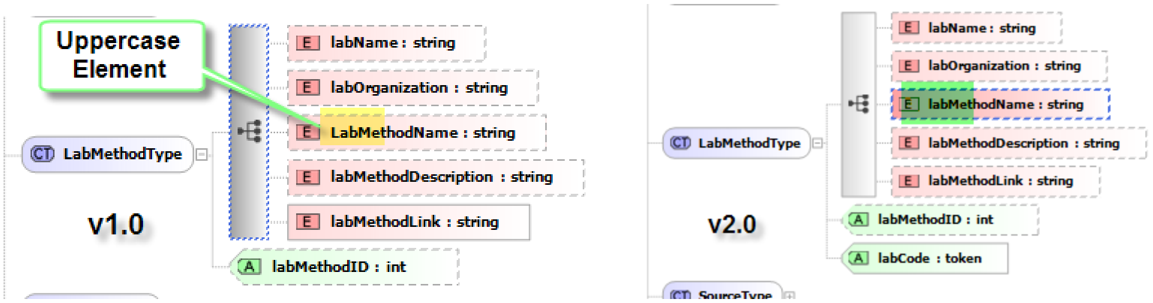
Change Request 20. Ensure naming consistency

Proposed Version: 1_1

Description: Casing rules got overlooked during rush to implement some elements and attributes. Changing case is not done lightly, since XML is caSe sensitive, and so is our preferred language, c#.
This is a change that does require coding changes.

Risks: High. Client code that is not ported will break.

CHANGE DETAILS:



CHANGE REQUEST 21. MULTIPLE VARIABLES

Proposed Version: 1_1

Description: timeSeries will now be repeatable, so multiple series are returned. Each with a set of data values

Add units element inside of values. Remove from attributes of values. (CR#22)

Risks: low.

CHANGE DETAILS:

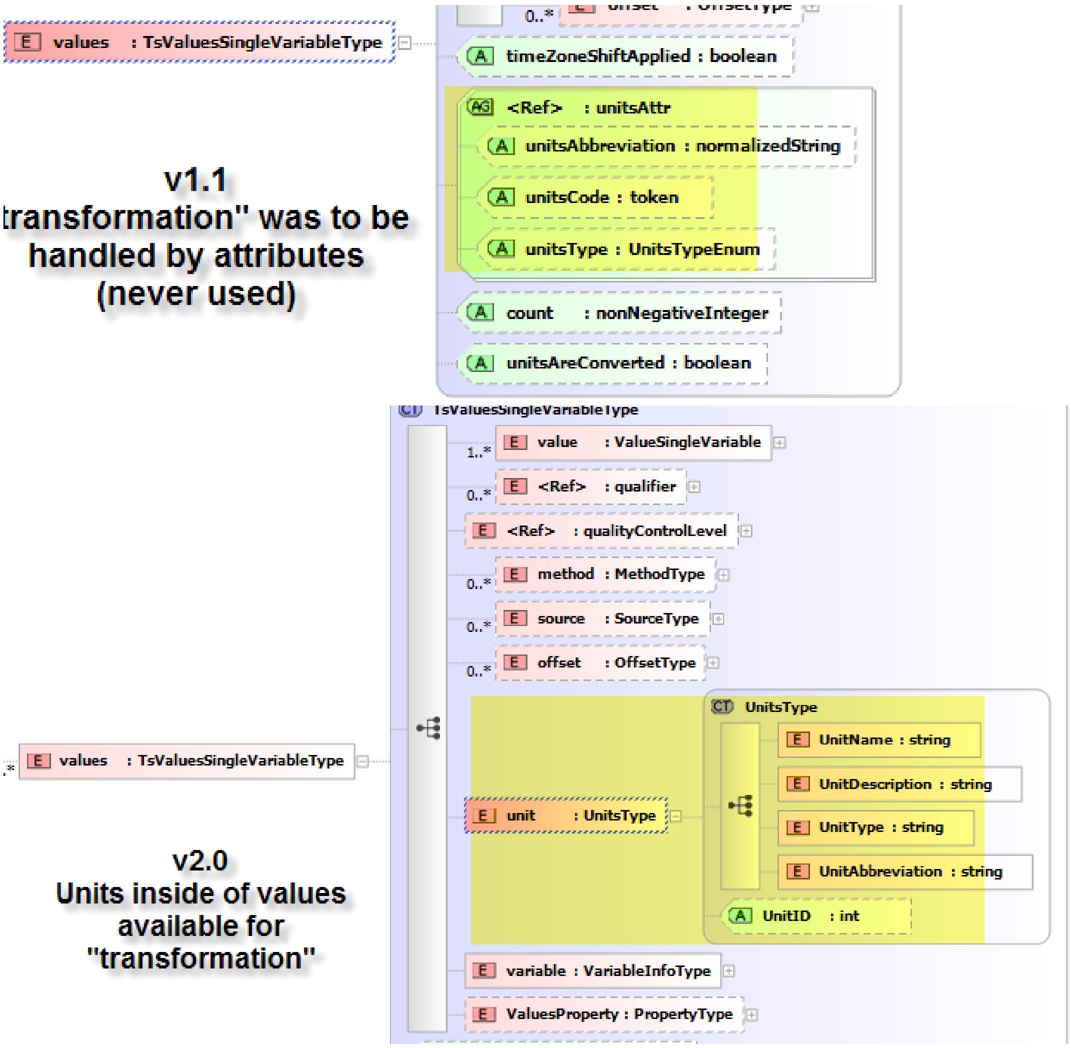
CHANGE REQUEST 22. ALLOW FOR UNIT TRANSFORMATION VALUES

Proposed Version: 1_1

Description: in WaterML 1.0 and 1.1 it was proposed that the attributes attached to values could allow of transformation of units to occur.
Add units element inside of values. Remove from attributes of values.

Risks: Medium. Clients must now look for units. But the unit element is standardized. Transformation never implemented.

Change Details:



values : TsValuesSingleVariableType
0..* offset : OffsetType
timeZoneShiftApplied : boolean
<Ref> : unitsAttr
unitsAbbreviation : normalizedString
unitsCode : token
unitsType : UnitsTypeEnum
count : nonNegativeInteger
unitsAreConverted : boolean
TsValuesSingleVariableType
1..* value : ValueSingleVariable
0..* <Ref> : qualifier
<Ref> : qualityControlLevel
0..* method : MethodType
0..* source : SourceType
0..* offset : OffsetType
UnitsType
UnitName : string
UnitDescription : string
UnitType : string
UnitAbbreviation : string
UnitID : int
unit : UnitsType
values : TsValuesSingleVariableType
variable : VariableInfoType
ValuesProperty : PropertyType

v1.1
transformation" was to be
handled by attributes
(never used)

v2.0
Units inside of values
available for
"transformation"

Change Request 23. Change how Data Values are handled

Proposed Version:

Description: Multiple changes are proposed.
We should move from attributes to elements
We should allow for more precise times that just dateTime
We should allow for more than just a numeric value, such as a null, categorical, or vector data types.

Risks: Low to Moderate
Clients would need to be recoded, and would need to better be able to handle more than a numeric value, and should be able to handle null values, in the same manner as NoDataValues.

CHANGE DETAILS:

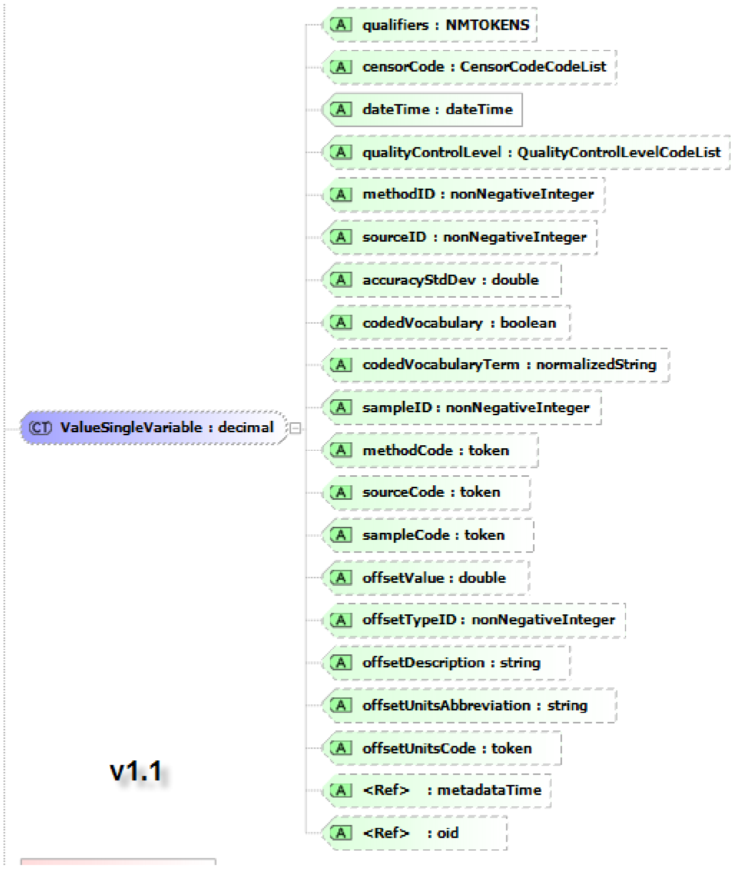
CHANGE REQUEST 24. MOVE ATTRIBUTES TO ELEMENTS ON VALUE

Proposed Version: 2

Description: It can be difficult to use attributes in some programs. Elements also provide more flexibility, such as the ability to have defined time with more precision.

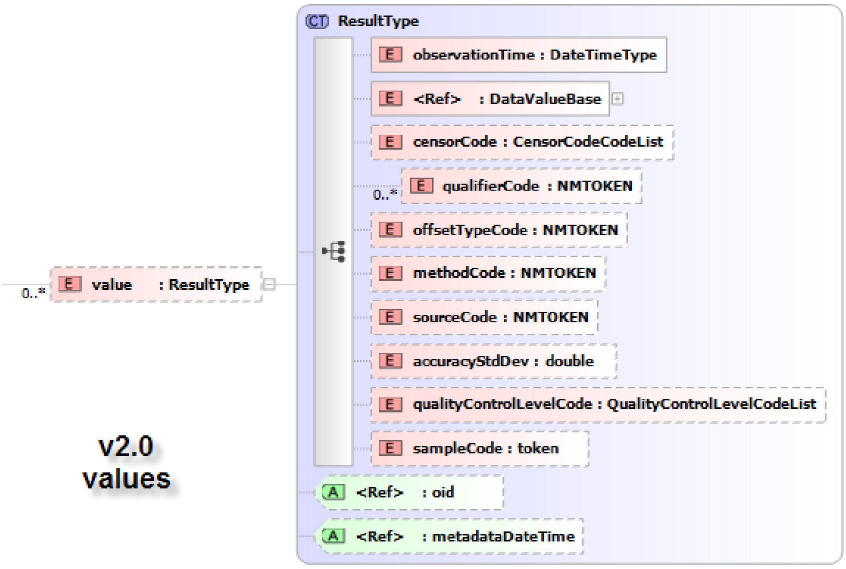
Risks: medium. Large change.
Files become larger

CHANGE DETAILS:



CT ValueSingleVariable : decimal
A qualifiers : NMTOKENS
A censorCode : CensorCodeCodeList
A dateTime : dateTime
A qualityControlLevel : QualityControlLevelCodeList
A methodID : nonNegativeInteger
A sourceID : nonNegativeInteger
A accuracyStdDev : double
A codedVocabulary : boolean
A codedVocabularyTerm : normalizedString
A sampleID : nonNegativeInteger
A methodCode : token
A sourceCode : token
A sampleCode : token
A offsetValue : double
A offsetTypeID : nonNegativeInteger
A offsetDescription : string
A offsetUnitsAbbreviation : string
A offsetUnitsCode : token
A <Ref> : metadataTime
A <Ref> : oid
v1.1

Original 1.1 ‘value’ element



CT ResultType
E observationTime : DateTimeType
E <Ref> : DataValueBase
E censorCode : CensorCodeCodeList
0..*
E qualifierCode : NMTOKEN
E offsetTypeCode : NMTOKEN
E methodCode : NMTOKEN
E sourceCode : NMTOKEN
E accuracyStdDev : double
E qualityControlLevelCode : QualityControlLevelCodeList
E sampleCode : token
A <Ref> : oid
A <Ref> : metadataDateTime
0..*
E value : ResultType
v2.0
values

Change Request 25. Make it possible to use XML data types to specify time precision

Proposed Version:

Description:
Make it possible to use XML data types to specify time precision. eg DateTime, Year-Month-Day, Month-Day, Year
USGS suggestion. Send out times at the appropriate precision so that users do not misuse data..

Risks: moderate. Clients will need to know that some values do not convert to dateTime, and will have to do appropriate processing
Best practices means that client code needs to be able to handle a variety of date formats, properly.

Change Details:

Easily possible in XML, when it the temporal reference is an element. Element observationTime is defined as DateTimeType, which is defined as the union of date, dateTime, gYear, gYearMonth. This will cover DateTime, Year-Month-Day, Month-Day, Year

```
<xs:element name="observationTime" type="DateTimeType" />
  <xs:simpleType name="DateTimeType">
    <xs:union memberTypes="xs:date xs:dateTime xs:gYear xs:gYearMonth" />
  </xs:simpleType>
```

NEED XML Example

Change Request 26. Allow for other data value types

Proposed Version: 2

Description: Numeric, and categorical values are represented in the same form. It is possible to define different elements which could communicate what the value is, eg dataValueNumeric, dataValueCategorical, dataValueVector.

- **Allow for empty data values; <value>**
- **Change Client Code need to handle null values.**
- **Handle categorical differently than numeric data values**

- **Categorical data type: dataValueCategorical**
- **Vector data types**
- **Vector data type: dataValueVector**

Risks: High.
Clients need to know how to handle different variations of observed value.

CHANGE DETAILS:

Utilize substituionGroup’s as a way to have different elements that contain the ‘value’

A dataValueNumeric element would contain a standard data value (eg float), and a

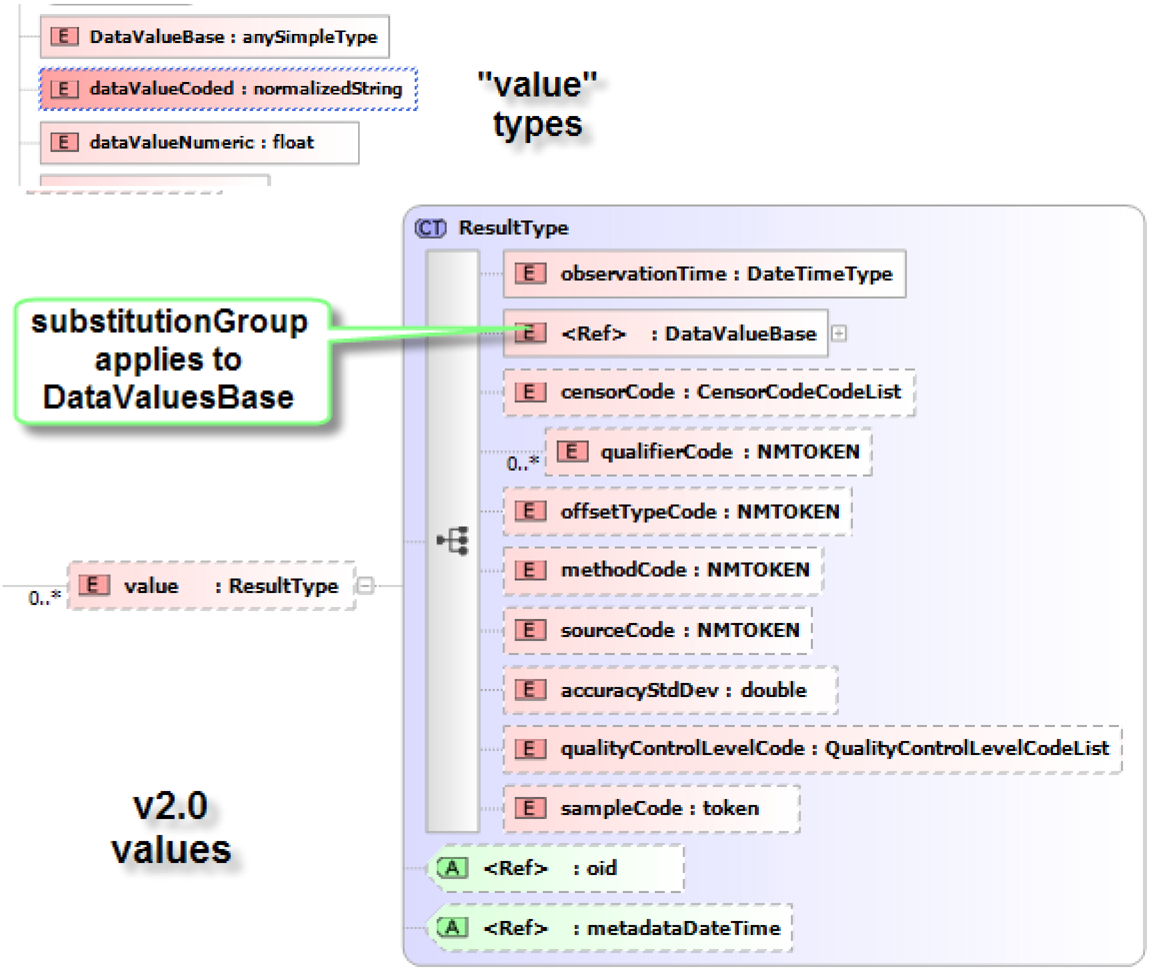
The are defined to substitute for DataValueBase which is used in the type that defines value.

TODO: Needs a graphical XML output

```
<xs:element abstract="true" name="DataValueBase" type="xs:anySimpleType" />

 <xs:element name="dataValueCoded" substitutionGroup="DataValueBase"
type="xs:normalizedString" />

  <xs:element name="dataValueNumeric" substitutionGroup="DataValueBase"
type="xs:float" />
```


E DataValueBase : anySimpleType
E dataValueCoded : normalizedString
E dataValueNumeric : float
"value" types
CT ResultType
E observationTime : DateTimeType
substitutionGroup applies to DataValuesBase
E <Ref> : DataValueBase
E censorCode : CensorCodeCodeList
0..* E qualifierCode : NMTOKEN
E offsetTypeCode : NMTOKEN
0..* E value : ResultType
E methodCode : NMTOKEN
E sourceCode : NMTOKEN
E accuracyStdDev : double
E qualityControlLevelCode : QualityControlLevelCodeList
v2.0 values
E sampleCode : token
A <Ref> : oid
A <Ref> : metadataDateTime

Change Request 27. Time Zone/Offset Issues

Proposed Version: 1_1

Description: Language support for time zones is poor. Time zones should be carried as a separate attribute.

Risks:

Change Details:

Change Request 28. Multiple Sites with SiteInfo

Proposed Version: 1_1

Description:

Risks:

Change Details:

Change Request 29. GetSites by Box

Proposed Version: 1_1

Description:

Risks:

Change Details:

Change Request 30. Return values for a site

Proposed Version:

Description:

Risks:

CHANGE DETAILS:

CHANGE REQUEST 31. TITLE

Proposed Version:

Description:

Risks:

CHANGE DETAILS: