OGC 12-031r2

Open Geospatial Consortium

Approval Date: 2012-06-22

Publication Date: 2012-07-12

External identifier of this OGC® document: http://www.opengis.net/doc/DP/waterml-timeseries-netcdf

Reference number of this OGC® project document: OGC 12-031r2

Category: OGC® Discussion Paper

Editor: Doug Palmer

WaterML 2.0 – Timeseries – NetCDF Discussion Paper

Warning

This document is not an OGC Standard. This document is an OGC Discussion Paper and is therefore not an official position of the OGC membership. It is distributed for review and comment. It is subject to change without notice and may not be referred to as an OGC Standard. Further, an OGC Discussion Paper should not be referenced as required or mandatory technology in procurements.

Document type:	OGC® Discussion Paper
Document subtype:	Encoding
Document stage:	Approved for public release
Document language:	English

License Agreement

Contents

i. Abstract

This discussion paper investigates the possible uses of NetCDF as a representation of WaterML timeseries data. The work is largely based on the WaterML 2.0 standard for timeseries, the NetCDF core and extensions standards and the CF-NetCDF and ADCC conventions.

ii. Keywords

Ogc, ogcdoc, waterml, netcdf, timeseries

iii. Preface

This is an OGC discussion paper for review by OGC members and other interested parties. It is a working draft document and may be updated, replaced by other documents at any time. It is inappropriate to use OGC Discussion Papers as reference material or to cite them as other than "work in progress." This is a work in progress and does not imply endorsement by the OGC membership.

This discussion paper is being developed through the joint WMO-OGC Hydrology Domain Working Group.

iv. Submitting organizations

The following organizations submitted this Discussion Paper to the Open Geospatial Consortium Inc.

a) Commonwealth Science and Industry Research Organisation (CSIRO)

v. Submission contact points

All questions regarding this submission should be directed to the editor or the submitters:

CONTACT	COMPANY
Doug Palmer	CSIRO

vi. Revision history

Date	Release	Author	Paragraph modified	Description
2012-02-13	0.1	Doug Palmer	All	Initial Draft
2012-05-25	0.1.2	Doug Palmer	6, 9	Tables correctly labelled. Clarify relationship between the WaterML 2.0 conceptual model and the XML encoding. Remove dropping indices for single stations.

vii. Changes to the OGC® Abstract Specification

The OGC® Abstract Specification does not require changes to accommodate this OGC® standard.

Foreword

This work has, for the most part, been funded through a water information research and development alliance between CSIRO's Water for a Healthy Country Flagship and the Australian Bureau of Meteorology.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. Open Geospatial Consortium shall not be held responsible for identifying any or all such patent rights. However, to date, no such rights have been claimed or identified.

Recipients of this document are requested to submit, with their comments, notification of any relevant patent claims or other intellectual property rights of which they may be aware that might be infringed by any implementation of the specification set forth in this document, and to provide supporting documentation.

Introduction

The WaterML 2.0 timeseries[1] standard provides a consistent conceptual model for hydrological timeseries data. The WaterML model follows the OGC approach towards modelling. Models are expressed as UML and are built by specialising existing abstract standards. The models are designed to allow rich and complex data and metadata. An XML encoding of the conceptual model, which faithfully represents the model, is part of the standard.

NetCDF is a set of APIs, libraries and self-describing machine-independent data formats, commonly used in climatology, meteorology and oceanography applications for array-oriented data. NetCDF is used as an input/output format for many GIS and scientific applications, as well as for archival storage and scientific data exchange. The NetCDF model is relatively constrained. Data is encoded in multi-dimensional arrays. Metadata attributes can be attached to the arrays, or to the dataset as a whole. A range of conventions for encoding data and attributes has grown up around NetCDF for the purposes of consistent data exchange.

A NetCDF encoding of WaterML would facilitate the archiving, exchange, import and export of hydrological observations. The first major goal of this paper is to develop a NetCDF encoding of WaterML that correctly represents a useful profile of the WaterML model. The second major goal is to ensure that any NetCDF representation of Water ML is a useful NetCDF document in the context of normal NetCDF usage; the contents of the document can be readily comprehended and processed by existing users and software. Since the conceptual models underpinning WaterML and NetCDF are not readily compatible, the paper explores the restrictions and mapping strategies needed to permit the encoding.

[1] Unless otherwise specified, from now on "WaterML" refers to "WaterML 2.0."

WaterML 2.0 Timeseries — NetCDF Encoding

1 Scope

This discussion paper investigates the possible uses of NetCDF as a representation of WaterML timeseries data. The work is largely based on the WaterML 2.0 standard for timeseries, the NetCDF core and extensions standards and the CF-NetCDF and ADCC conventions. The goal is to develop a set of best practices that allows WaterML timeseries to be represented as NetCDF documents for the purposes of archiving and import/export from applications.

2 Normative references

The following normative documents contain provisions which, through reference in this text, constitute provisions of this part of OGC 12-031. For dated references, subsequent amendments to, or revisions of, any of these publications do not apply. However, parties to agreements based on this part of OGC 12-031 are encouraged to investigate the possibility of applying the most recent editions of the normative documents indicated below. For undated references, the latest edition of the normative document referred to applies.

OGC 10-126r2 WaterML 2.0: Part 1 - Timeseries

OGC 10-090r3 NetCDF Core Encoding Standard version 1.0

OGC 10-091r3 CF-netCDF Core and Extensions Primer

OGC 10-092r3 NetCDF Binary Encoding Extension Standard: NetCDF Classic and 64-bit Offset Format

OGC 07-036 OpenGIS® Geography Markup Language (GML) Encoding Standard

ISO 19156:2011 Geographic information - Observations and measurements

W3C XML Linking Language (XLink) Version 1.0

W3C XML Path Language (XPath) Version 1.0

3 Terms and definitions

For the purposes of this document, the following terms and definitions apply /the terms and definitions given in … and the following apply.

3.1

Classic Model
The classic NetCDF data model (NetCDF-3).

3.2

Document
A representation of some information. For the purposes of this paper, a document may be an actual document, such as a file, or a virtual document, such as a response to a web request or a response from a database query.

3.3

Dataset
A related collection of data.

3.4

Extended Model
The extended NetCDF data model (NetCDF-4).

3.5

Hydrology
The study of the movement, distribution, and quality of water on Earth (and other planets)

3.6

Profile

1. A sampling of a property along a vertical line
2. A logical restriction, defining a subset of a schema, model or standard

4 Conventions

4.1 Symbols (and abbreviated terms)

ACDD	Attribute Conventions for Dataset Discovery
AP	Application Profile
API	Application Program Interface
COARDS	Cooperative Ocean/Atmosphere Research Data Service
CF	Climate and Forecast Metadata Convention
CSML	Climate Science Modelling Language
FOI	Feature of Interest

GEO-IDE	Global Earth Observation Integrated Data Environment
ISO	International Organization for Standardization
NetCDF	Network Common Data Format
NOAA	National Oceanic and Atmospheric Administration
OGC	Open Geospatial Consortium
OPeNDAP	Open-source Project for a Network Data Access Protocol
O&M	Observations and Measurements
PMEL	Pacific Marine Environmental Laboratory
SOS	Sensor Observation Service
THREDDS	Thematic Realtime Environmental Distributed Data Services
UCUM	Unified Code for Units of Measure
UML	Unified Modelling Language
WaterML	Water Markup Language (version 2.0, unless otherwise specified)
XLink	XML Linking Language
XPath	XML Path Language
XML	eXtended Markup Language
1D	One Dimensional
2D	Two Dimensional
3D	Three Dimensional

4.2 UML Notation

The diagrams that appear in this standard are presented using the Unified Modelling Language (UML) static structure diagram.

4.2.1 Use of XPath Expressions

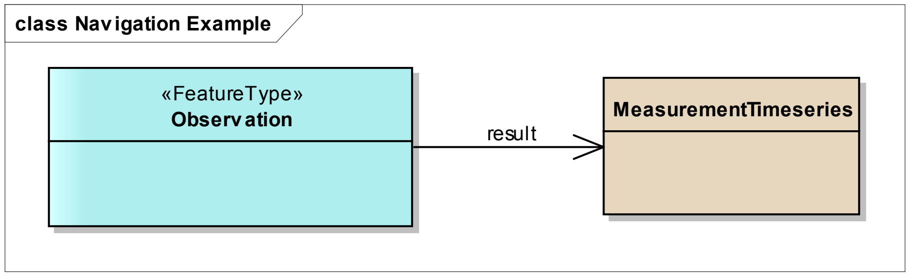

Figure 1 — Navigation Example

This document attempts to map parts of the WaterML model onto specific NetCDF variables and attributes. The WaterML model uses UML, a graphical notation. Paths through UML associations are usually expressed by including UML diagrams – see Figure 1 for an example. The number of paths described in this paper makes a graphical approach impractical. There are many cases in this document where some sort of textual notation for navigating object graphs would be useful. However, there is no generally accepted notation.

WaterML has an existing XML encoding that is auto generated from the models following the GML encoding rules, Annex E. This encoding does not cover the entirety of WaterML, but encapsulates the intended use of the model by the hydrology community and provides comprehensive metadata.

Rather than use a UML-based graph notation language, the WaterML instance is considered to be expressed as its XML encoding and elements are located via the XPath language. For example, the reference shown in Figure 1 would be written as `om:OM_Observation/om:result/wml2:MeasurementTimeseries` If an abstract schema type is used, then the element name is italicised, for example `om:OM_Observation/om:result/`*`wml2:TimeseriesType`*

5 Motivation

The WaterML 2.0 timeseries standard is an Observations and Measurements (O&M) application profile (AP) that captures the semantics of hydrological observational data for data exchange. WaterML has an existing XML encoding that encapsulates the intended use of WaterML in the hydrology community and is well-suited to data exchange applications.

The Network Common Data Format (NetCDF) provides a common data access method for self-describing, machine-independent datasets(Rew, et al., 2011). Array data can be compactly encoded in NetCDF documents, along with attributes that describe the meaning of the array data.

The NetCDF format is domain-independent and can be used by many branches of science and engineering. Over the years, a number of domain- and application-specific conventions have developed to allow consistent data representation and exchange. An influential set of conventions are the NetCDF Climate and Forecast Metadata conventions (the CF conventions) which have developed to allow users of climate and forecast data to decide which quantities are compatible and consistently extract, sample and display data(Eaton, et al., 2011).

NetCDF is also a common format for web-based data catalogues. Suitable attribute conventions, such as the Attribute Convention for Dataset Discovery(Davis, 2005), can be used to allow dataset servers, such as THREDDS(Unidata Program Center) to make datasets searchable. Servers that support the Open-source Project for a Network Data Access Protocol (OPeNDAP) allow applications to request slices of large datasets from network repositories, with NetCDF one of the supported formats(OPeNDAP, 2009).

6 Use Cases

An instance of the WaterML model is a specific representation of timeseries data that conforms to the WaterML conceptual model. It may be an in-memory object that is part of a program or a suitably encoded document. At present, the only instances of WaterML that formally conform to the model are documents in the XML encoding, which conform to a profile of the model.

Ideally, a NetCDF encoding of WaterML should represent a relationship between the WaterML conceptual model and a NetCDF document, rather than XML encoded WaterML and NetCDF encoded WaterML. In the following use cases, a WaterML instance refers to any representation of WaterML.

6.1 Archival Storage

NetCDF provides a convenient archival format, capable of handling large quantities of data in a reasonably compact format and, via tools such as THREDDS, making it available in a searchable manner. In some cases, timeseries collected as WaterML may need to be stored as part of a larger data store, where the format of choice is NetCDF.

In the case of archival storage, it becomes necessary not to lose information contained in the original WaterML instance. In principle, the original instance should be re-constructible from the NetCDF document.

An archival document also needs to conform to the requirements of the archive. The archival document needs to adhere to the metadata requirements of the archive, so that the data can be usefully retrieved. The archival document also needs to adhere to any useful domain-specific conventions, so that the data can be directly used by domain-specific software.

6.2 NetCDF as a Data Payload

Services such as OGC's Sensor Observation Service (SOS) allow a variety of encodings for the data returned as part of a query. Similarly, markup languages based on O&M, such as the CSML, allow referenced datasets encoded as NetCDF. Markup languages for

NetCDF that allow aggregation, such as NcML, also allow composition of multiple referenced NetCDF documents.

In all cases, a data payload document may become separated from the larger context from which it derives. The document, therefore, needs to contain enough metadata and ancillary data to allow the data presented to be interpreted correctly without any surrounding context. It is not necessary that the data be a complete representation of the underlying WatertML model.

6.3 Presenting NetCDF Timeseries as WaterML

There are hydrological timeseries that are already stored as NetCDF. Since these timeseries are sensor data, a SOS may be used to serve the timeseries data. A SOS can return a reference to the NetCDF document directly, or convert data contained within the file into a WaterML model and return the XML-encoding of the model.

In the case of conversion to XML, the NetCDF document needs to contain enough information to allow the construction of a valid WaterML model that matches the document. Or, if complete data is not available, use sensible default values.

6.4 Import Format for Tools

Many geospatial, visualisation and analysis applications can read NetCDF with suitable attribute conventions directly. Exporting data described by a WaterML instance as a NetCDF document allows these tools to ingest the data.

The exported NetCDF document need not be a complete representation of the WaterML instance. Only that subset of the model that is interpretable by the tool, with sufficient metadata encoded in the conventions expected by the tool, is absolutely necessary. However, as with the data payload use case, the exported document may become separated from context and take on a life of its own. Sufficient generic and domain-specific metadata is needed to ensure that the exported document is useful in its own right.

7 Relationship to Existing Standards and Conventions

7.1 WaterML 2.0

The WaterML 2.0 timeseries standard is a profile of the Observations and Measurements (O&M) standard designed to accurately report timeseries data collected by hydrological monitoring points. The key features of WaterML, from the point of view of this paper are:

- Timeseries represent the result of an O&M observation. In addition to the result itself, the observation contains contextual information about the nature of the observation:

  - A potentially large and complex collection of metadata describing the source of the information;
  - The geospatial and temporal features that the result pertains to;
  - Information about the procedure used to collect the result;
  - Information about the property that the result measures;
  - Information about the quality of the result;
  - Information about related observations; and
  - Other parameters
- The timeseries itself contains:
  - Metadata about the timeseries itself, particularly the time extent and frequency of measurements;
  - Default metadata for the individual points in the timeseries;
  - Extensions containing arbitrary named additional data; and
  - A sequence of time-value pairs. The sequence may be represented in a number of different ways. Time- and value-data may be interleaved or separated into matching domain- and range-sequences. Times may be specified or generated.
- A time-value pair can encode a measurement (e.g. 3.45m) or a categorical value (e.g. *Cloudy*). Each pair is associated with metadata, which may either be explicitly present or the timeseries default:
  - The quality of the value, in the form of a standardised term;
  - Qualifiers that describe additional information about the observation process;
  - Any related observations;
  - Any processing performed on the value, in the form of a standardised term;
  - Any comments on the measurement; and
  - Standardised terms that give the reason why a value is nil.

  Measurements may also contain:

  - The controlled vocabulary from which a category term is drawn;
  - The unit of measurement, either as a URI or as a UCUM(Schadow & McDonald, 2009) unit;
  - The *interpolation type,* a description of the way a value has been accumulated over the measurement period; and
  - Standardised terms that give the reason why a value is not available or censored

Groups of related observations may be gathered into collections.

The abstract timeseries model that is used in WaterML is capable of representing complex data as timeseries. In the XML encoding of WaterML, measurements are limited to single values at present. However, it is expected that future versions of the standard may include record-like, 1D, 2D or higher dimensional values.

7.2 NetCDF

There are two forms of NetCDF of interest, the classic and extended forms. These forms consist of a data model, an abstract description of the structure and type of data that can be contained in a NetCDF document, and a number of encodings. In particular, each form has a binary encoding. The major focus of this paper is the representation of data; as such it focuses on the data models.

The classic NetCDF form, NetCDF-3, has a data model that allows a dataset to be described in terms of a collection of variables. Each variable has a set of dimensions that describe the size and shape of the variable array, with scalars being single, dimensionless values. Each variable, and the dataset as a whole, can be associated with a set of attributes that provide metadata describing the variable and the dataset as a whole. Attributes and variables are typed, with a limited selection of data types. The classic NetCDF data model is shown in Figure 2.

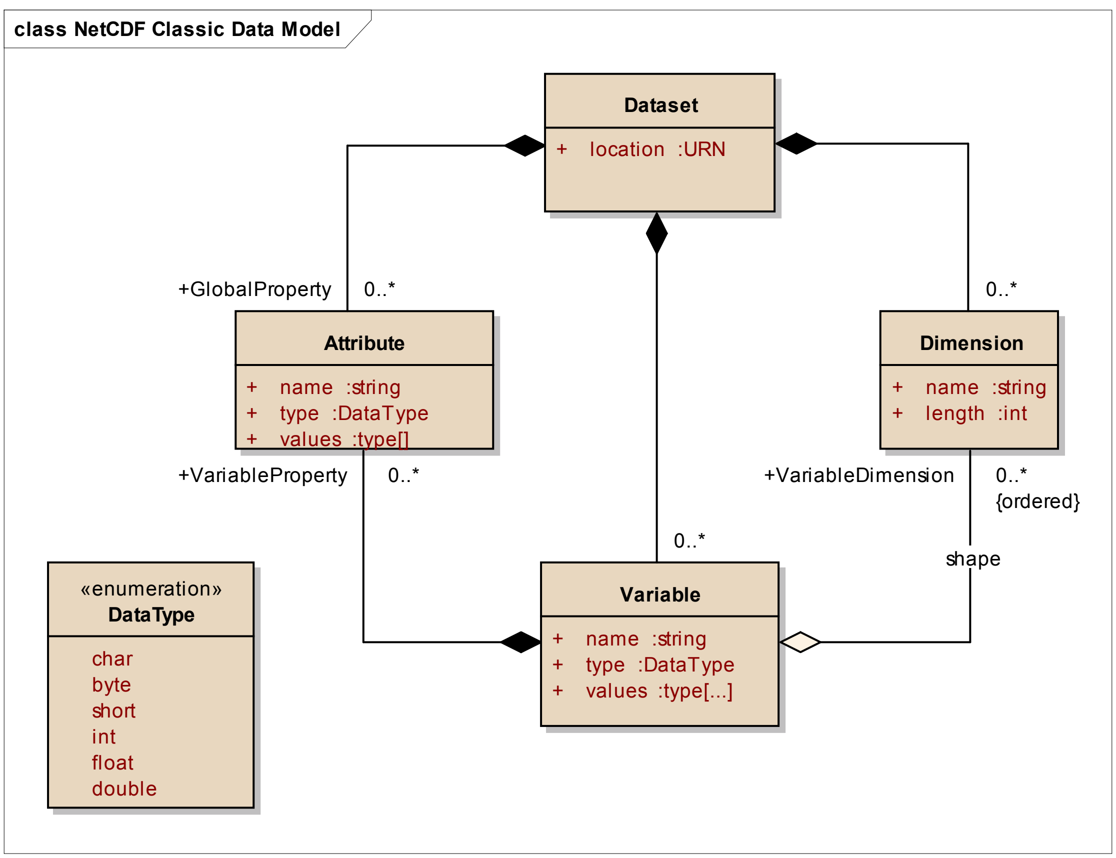

Figure 2 — Classic NetCDF Data Model

The classic data model provides a simple way of presenting array-like data with minimal structure. The classic data model is described in *OGC 01-090r3 OGC Network Common Data Form (NetCDF) Core Encoding Standard version 1.0*. A binary encoding of the data model is described in *OGC 10-092r3 NetCDF Binary Encoding Extension Standard: NetCDF Classic and 64-bit Offset Format.*

A minimal set of attribute conventions for all NetCDF documents has been defined in Appendix B of the NetCDF user's guide(Rew, et al., 2011). These conventions provide a way of documenting the name, units and valid range of a variable, as well as information about the dataset as a whole, in particular, the specific set of conventions that the dataset adheres to. Units in a NetCDF document are conventionally given using the UDUNITS library and database(Unidata Program Center, 2011).

The classic NetCDF data model is widely used but is highly restrictive. An enhanced form has been developed, NetCDF-4, which provides a number of extensions:

- The number of possible primitive data types has been expanded. In particular, variables may now be variable-length strings (NetCDF-3 variables intended to be strings were fixed-length character arrays);
- A recursive, grouping structure has been introduced to allow hierarchies of variables and attributes; and
- User-defined enumerated, compound, variable length and opaque data types may be created.

The extended NetCDF data model is shown in Figure 3.

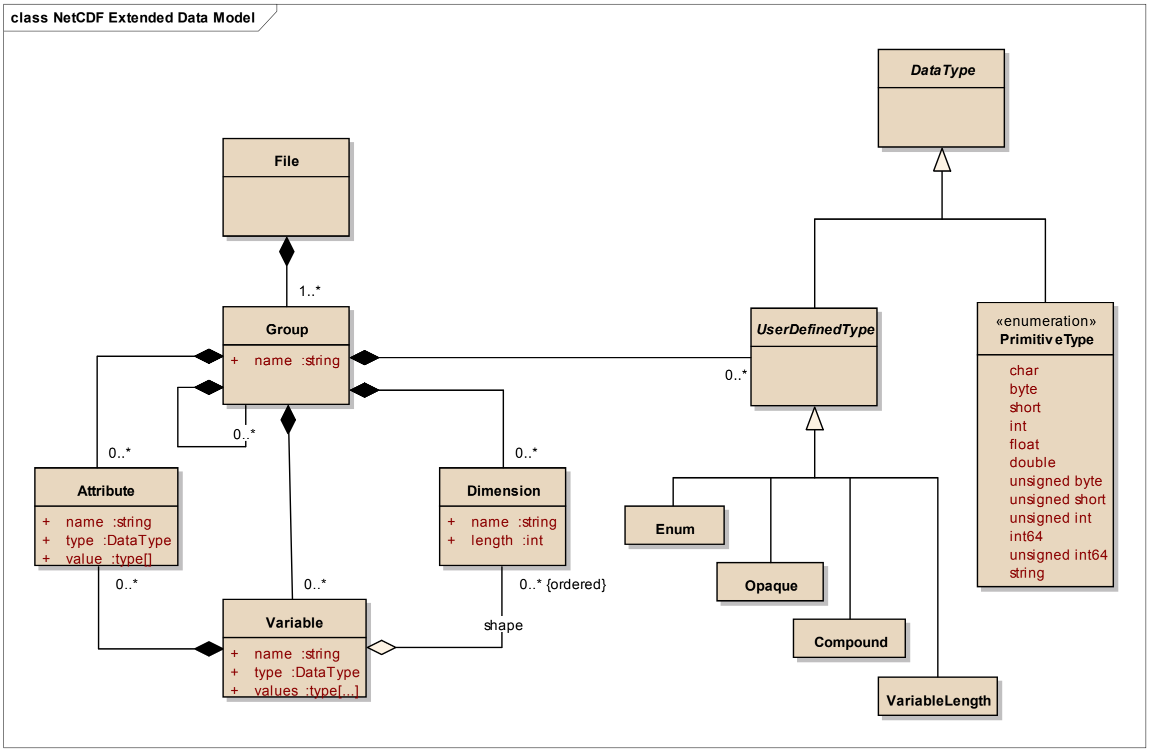

Figure 3 — Extended NetCDF Data Model

A separate binary format, based on the Hierarchical Data Format, Version 5 (HDF5) (The HDF Group, 2011) has been developed for NetCDF-4. It is possible to produce documents encoded in the NetCDF-4 binary format but which adhere to the classic data model, called the NetCDF-4 classic model format.

The extended form is not part of any formal OGC standard, although it is included in the *OGC 10-091r3 CF-netCDF Core and Extensions Primer* roadmap for further development and a draft standard has been developed(Domenico, 2012). NetCDF-4.0 was released in 2008. Since then, there has been what has been described as a "chicken and egg logjam"(Rew, Adapting Software to NetCDF's Enhanced Data Model, 2010) where data providers and data consumers wait for the other to adopt the new standard. Most applications now support the NetCDF-4 classic model format(Rew, Update on Unidata Technologies for data Access, 2011).

Support for the enhanced data model, however, is still spotty; the NetCDF Java library is still read-only, for example. Since most of the use-cases described in Section 6 can be handled using the classic data model, the classic data model is used wherever possible.

There are a few cases where the enhanced data model is desirable, identified by **(enhanced model)** in the text. These usually revolve around the need to record variable-length metadata strings on a per-point basis. Generally, metadata at the timeseries or observation level can be placed in attributes or ancillary variables. However, per-point comments or complex metadata will need to be placed in a character string. Classic data model character arrays are FORTRAN-style fixed-length arrays. If fixed-length arrays are used to record sparse data, such as comments, the result would be wasteful. Using an indexing strategy, similar to ragged arrays in timeseries is also possible but would require an array that can accommodate the largest possible string for each entry. The variable length strings of the enhanced data model provide a direct solution.

There are also a few cases where the use of the enhanced data model would alter the mapping strategies discussed in Section 8. As an example, the grouping mechanism provided by the enhanced data model would allow a more natural way of expressing URIs as internal references. Rather than develop a completely separate mapping for the enhanced data model, the mappings used in the classic data model are used.

7.3 NetCDF Attribute Convention for Dataset Discovery

The Attribute Convention for Dataset Discovery (ACDD) provides a set of metadata conventions that are recommended for describing NetCDF documents for discovery systems, such as digital libraries and cataloguing services(Davis, 2005). The attributes specified by the ACDD conventions provide identification, keywords and vocabulary definitions, geospatial and temporal boundaries, origins and validity information.

The ACDD recommends the use of a vocabulary of standard names for variables. No specific vocabulary is recommended and the vocabulary used can be specified as part of the dataset metadata.

Use of the ACDD is highly recommended in NetCDF files. A draft set of mappings is being developed to map the ACDD conventions onto other metadata standards, such as Dublin Core, ISO 19115-2 and THREDDS.

7.4 Climate and Forecast Conventions

The Climate and Forecast (CF) conventions are a commonly used set of conventions for providing semantic information on the data contained in variables(Eaton, et al., 2011). The conventions have been accepted by the OGC as an extension to the NetCDF core in *10-091r3 CF-netCDF Core and Extensions Primer*. From the point of view of this paper, the conventions provide guidance on:

- Naming conventions for files, dimension and variable names;
- Provision of ancillary data, such as flags, quality control or confidence levels;
- Geospatial and temporal coordinates, coordinate systems and projections;
- Descriptions of aggregation methods for aggregate measures; and
- Sampling and timeseries methods.

The CF conventions extend the COARDS conventions(Hankin & Searight, 1995), a set of conventions that are used by many geospatial and visualisation packages to map gridded data onto a coordinate system. Similarly, the CF conventions extend the GDT conventions(Gregory, Drach, & Tett, 1999), particularly those relating to coordinate axes.

The CF conventions provide a set of conventions for discrete sampling geometries; datasets that represent paths through a larger region of space-time. The discrete sampling geometries conventions provide a standardised way of describing timeseries from multiple sampling points, corresponding to a collection of monitoring stations.

The current WaterML timeseries standard is built around monitoring stations collecting point data. However, the WaterML model can be adapted to allow more complex data forms. In the future, stations that collect data along a horizontal line or a vertical surface may be considered, as may be the output of multi-station processes. Although the CF conventions have feature types that are very close to these cases (e.g., a timeSeriesProfile collects data along a line) these features are tied to specific axes (the timeSeriesProfile uses a vertical line). Future versions of the CF conventions may relax these restrictions.

The CF conventions provide a table of standard names for the physical quantities being recorded. As implied by the name of the convention, these names are largely drawn from the terminology of the climate and forecasting domains. There are a number of terms in common use in the hydrology domain, such as turbidity, that have no standard name. There are also terms that have a common name in hydrology that have a more generic name in the CF standard names (e.g. stage is water_surface_height_above_reference_datum). See Section 7.8 for a further discussion of standard names.

7.5 Climate Science Modelling Language

The Climate Science Modelling Language (CSML) is a data model for encoding climate, atmospheric and oceanographic data in terms of geometry-based observation classes such as points, profiles, trajectories and grids(Woolf, 2011; Natural Environment Research Council, 2011).

The CSML model defines a number of observation types that are aligned with the CF discrete sampling geometries feature types; for example Point and PointSeries in CSML correspond to point and timeSeries in CF. The CSML model also contains a number of additional observation types, such as swaths and scanning radars, that are not defined in the current CF conventions. These observation types are described in a pair of draft documents on the NetCDF web site (Unidata Program Center, 2008; Unidata Program Center, 2008).

The CSML model also contains a mechanism for embedding data documents into a larger metadata context. A CSML document contains a description of a dataset in the form of general metadata, an O&M observation and a storage descriptor. The storage descriptor describes the data storage format and how it can be accessed; e.g. as inline data or as a segment of an externally accessible file. Storage descriptors can be either embedded into a data description or reference by the XLink mechanism. For example, a NetCDF extract can be described by and referenced as:

```
<csml:NetCDFExtract gml:id="imos-uwnd">
  <csml:arraySize>23</csml:arraySize>
  <csml:fileName>
```

```
    http://opendap-
qcif.arcs.org.au/thredds/dodsC/IMOS/ABOS/ASFS/SOFS/2011/IMOS_ABOS-
ASFS_CMST_20110131T002900Z_SOFS_FV01.nc
  </csml:fileName>
  <csml:variableName>UWND</csml:variableName>
</csml:NetCDFExtract>
...
    <csml:rangeSet xlink:href="#imos-uwnd"/>
...
```

7.6 Other Timeseries Conventions

The US National Oceanographic Data Center (NODC) Argo profiling float conventions(Carval, et al., 2011) have been designed to support the recording of a series of profiles from the Argo free-drifting floats. The conventions allow for either a single profile or a trajectory of profiles.

The NCAR-RAF conventions(National Center for Atmospheric Research, Research Aviation Facility, 2009) support synchronous timeseries for both vectors and scalars with varying sampling rates. Data is collected at a one second sample interval. Measurements that occur at a rate higher than 1Hz are collected into additional dimensions; for example, a 25Hz signal will be broken into groups of 25 readings, aligned with other samples from the same time-period.

EPIC is a collection of libraries and software packages developed and maintained by the Pacific Marine Environmental Laboratory (PMEL). EPIC was developed to maintain the collection of datasets collected as part of NOAA climate study programs. The EPIC conventions provide simple profile and timeseries conventions, strongly oriented towards regular longitude/latitude/depth/time grids. A notable feature is that, rather than using standard names, properties are given standard numerical codes – for example, latitude has a code of 500, temperature a code of 20.

The Argo, NCAR-RAF and EPIC conventions are strongly domain-specific and specify a large number of variables and attributes that are tied to the underlying collection model and domain. These conventions could be made to fit hydrological data, but only at the cost of considerable violence to both the conventions and the data.

7.7 Choice of Conventions

A significant motivation for the use of NetCDF for WaterML is access to the large collection of software libraries, tools and systems that use NetCDF. Choice of conventions, therefore, is also driven by the practical considerations of whether the conventions add value in terms of software use. An additional driver is the need to make the resulting NetCDF document usefully comprehensible to humans so that collected data can be reused for scientific research.

To get a feel for the sort of conventions that are used by software, 55 of the packages listed on the NetCDF web site were examined to what conventions were explicitly used. The results are shown below:

Table 1 Use of NetCDF Conventions in Software Packages

Conventions	Count
No specific conventions listed or application-specific conventions	37
COARDS	4
CF	10
EPIC	1
NCAR-RAF	1
GDT	1
ACDD	1

Visualisation and graphics packages tend to either use the COARDS conventions directly or via the CF conventions. The conventions of choice, used for the rest of this document, are:

- The minimal NetCDF attribute conventions defined in the NetCDF user's guide.
- The ACDD conventions. These conventions capture a subset of the metadata available from the ISO 19115 conformant data contained in a WaterML model and provide essential metadata for data reuse.
- The CF conventions. These conventions are widely used and provide ways of describing timeseries, geospatial referencing, measurement annotations and vocabulary control.

In addition, it will be necessary to develop some WaterML-specific conventions to cover elements not easily modelled by the existing conventions.

7.8 Relationship Between Hydrology and NetCDF Conventions

This paper is situated in the hydrology domain. Many of the standards and conventions discussed above have been developed for use in the climatology, meteorology and oceanographic domains. In many cases, these domains face similar issues to hydrology and have developed solutions that can be directly adapted to the hydrology domain. However, in some cases there are mismatches in vocabularies, the sort of features being monitored and assumptions about the way measurements are taken.

An example of vocabulary is the CF standard name table, discussed in Section 7.4. An example of measurement methodology is the use of vertical sampling profiles in the natural domains of the CF, CSML and Argo conventions.

Some standards provide administrative methods for expansion. For example, there is a mechanism for constructing and adding new standard names to the CF conventions.

However, it is not clear that the hydrology vocabulary should be imported into the CF standard name table. A separate standard name vocabulary, possibly constructed according to the CF guidelines and format may be preferable. In general, it is not clear that hydrology- and WaterML-specific constructs should be added to conventions oriented towards other domains. For the purposes of this paper, separate sets of conventions are assumed to have been developed where necessary and mechanisms for specifying the vocabularies used have been developed. Wherever possible, these conventions should use the patterns of existing conventions, so that they can be adsorbed back into the existing conventions at a later date.

8 Mapping Strategies

8.1 Mapping Names, URIs and Tokens

Standard terms in WaterML tend to be either URIs, preferably URIs that resolve to semantic information, or terms drawn from controlled vocabularies, with the vocabularies referenced. An example standard term in WaterML is the URI http://www.opengis.net/WaterML/2.0/def/quality/good which is resolves to a semantic description of the meaning of "good quality".

Standard terms in NetCDF documents tend to be tokens, words or phrases connected by underscores, or enumerated values. As an example, the equivalent to "good quality" in NetCDF would be a token such as `quality_good`. The meaning of the tokens and enumerations in NetCDF is generally defined by reference to a manual or conventions document, referenced by use of the `Conventions` attribute and by further attributes.[2]

The tokens in NetCDF fulfil the function of URIs; unambiguous references to concepts. Since a WaterML model is likely to draw vocabularies from several sources, including the URIs contained in the WaterML standard, mechanisms for mapping URIs and controlled vocabularies onto tokens or enumerations need to be developed, as do mechanisms for mapping the resulting tokens back onto definitions.

The enhanced NetCDF model allows enumerations to be directly specified. In keeping with the policy of avoiding conventions that require the enhanced model, use of enhanced model enumerations is not considered.

[2] For example, the Argo float conventions uses attributes called `DATA_TYPE`, `HANDBOOK_VERSION` and `FORMAT_VERSION` to provide a reference to the conventions used.

8.1.1 Standard Names

The standard names used in the CF conventions are a special case of a NetCDF standard term. They are designed to provide an unambiguous reference to the property that a variable measures. The CF standard name table provides a small amount of semantic information: canonical units, descriptive text, aliases and mappings onto a small number of similar codes. However, standard names are generally used in software simply to ensure that like is compared with like.

Where possible, the variables in the document should have a standard name that reflects the property of the observation. If the standard name is taken from a specific vocabulary, the vocabulary should be recorded in the observation metadata and specified in the NetCDF global attributes (see Section 10.2). Where standard names can be mapped onto URIs, a dictionary should be included (see Sections 8.1.4 and 10.4.1)

Standard names can be derived from either a repository of standard names, such as the CF name table, by tokenizing a property URI (see Section 8.1.2) or by using data available from RDF (see Section 8.1.3).

8.1.2 URI Tokenization

Enumerations for URIs or terms from a standardized vocabulary can be constructed by tokenizing either the xlink:title value from the link to the URI, the fragment of the URI, the last path segment of the URI or the label or id of an AbstractSWEType. Tokenizing replaces whitespace and other non-alphanumeric characters with underscores and converts camelcase to lower case separated with underscores. For example:

Table 2 URI Tokenization Examples

Term	Tokenized
http://www.opengis.net/def/timeseriesType/WaterML/2.0/MinSucc	min_succ
Water Temperature	water_temperature
http://purl.oclc.org/NET/ssnx/cf/cf-property#rainfall_amount	rainfall_amount

Tokenized URIs can be associated with the actual URI by use of a dictionary – see Section 8.1.4.

8.1.3 Use of RDF

In many cases, URIs can be resolved to semantic descriptions of the URI, usually in the form of RDF. In such cases, values derived from the RDF can be used to provide standard name tokens and other property-specific information.

For example, the following RDF description of rainfall is extended from the CF properties ontology[3], developed by the Semantic Sensor Networks working group. The rdfs:label property can be used to derive a suitable long name and the cf:standardName can be used for the CF standard name.

```
<dim:SurfaceDensity rdf:about="http://purl.oclc.org/NET/ssnx/cf/cf-property#rainfall_amount">
  <rdfs:label>Rainfall Amount</rdfs:label>
  <rdfs:comment>"Amount" means mass per unit area.</rdfs:comment>
  <dc:source>Climate and Forecast (CF)</dc:source>
  <cf:standardName>rainfall_amount</cf:standardName>
  <qu:generalQuantityKind>
    <dim:SurfaceDensity
rdf:about="http://purl.oclc.org/NET/ssnx/qu/quantity#surfaceDensity"/>
  </qu:generalQuantityKind>
  <ssn:isPropertyOf>
    <cf-feature:Precipitation rdf:about="http://purl.oclc.org/NET/ssnx/cf/cf-feature#rainfall"/>
  </ssn:isPropertyOf>
  <qu:propertyType>
    <qu:PropertyKind rdf:about="http://purl.oclc.org/NET/ssnx/qu/quantity#scalar"/>
  </qu:propertyType>
</dim:SurfaceDensity>
```

8.1.4 Dictionaries and Vocabularies

Terms drawn from standard vocabularies in a WaterML model need to be reduced to tokens and enumerations. Tokens and enumerations used within a NetCDF document may need to be directly associated with their meanings, rather than by reference to a conventions document. This section outlines a strategy for constructing terms and enumerations from the URIs and vocabularies in a WaterML document and providing an embedded dictionary within the document.

For enumerated variables, the CF conventions specify a set of conventions for describing flag values. These conventions are largely concerned with supplying ancillary data, such as quality control attributes, but can be adapted to more general dictionary roles.

To provide a dictionary for an enumeration, three attributes need to be specified: `flags_mask`, `flags_values` and `flags_meanings`. The first, `flags_mask`, is a vector of masks intended to allow multiple flag values within a single variable. The actual values of the enumeration are listed in the vector `flags_values`. The meanings of the flags, as tokens, are listed in a space-separated string. For example:

```
weather:flags_mask = 7b;
weather:flags_values = 1b, 2b, 3b, 4b, 5b, 6b;
weather:flags_meanings = "fine cloudy showers rain snow hail";
```

The use of the flags attributes allows an enumeration to be mapped onto a set of tokens. The tokens then need to be associated, if possible, with a dictionary.

If the tokens come from a standardised vocabulary, there needs to be a link to the vocabulary, performing the same function as the `codeList` or `codeSpace` attribute

[3] http://www.w3.org/2005/Incubator/ssn/ssnx/cf/cf-property.owl

in the WaterML model. The `flags_vocabulary` attribute is the URI of the vocabulary or the name of an attribute containing a dictionary (see below).

Where possible, the `flags_vocabulary_typeof` attribute should describe the type or class of vocabulary, in keeping with the terminology of (Lefort, 2009). The `flags_vocabulary_typeof` attribute can be a URI, a mime type or descriptive text. For example, a SKOS vocabulary might have a `flags_vocabulary_typeof` of http://www.w3.org/2004/02/skos/core#ConceptScheme or `text/x-skos`.

It may be necessary to directly supply the vocabulary dictionary, particularly if the vocabulary consists of tokenised URIs. Dictionaries that have an XML document-like structure, such as SKOS concept schemes or WaterML local GML dictionaries can be directly embedded into the attribute and identified by the typeof attribute. The `flags_dictionary` attribute contained the embedded dictionary.

In the case of tokens are derived from URIs, a simple dictionary of URIs can be constructed by defining a simple space-separated key=value dictionary, mapping tokens onto URIs[4]. The `flags_vocabulary_typeof` attribute in this case is "simple". For example:

```
weather:flags_vocabulary = "flags_dictionary";
weather:flags_vocabulary_typeof = "simple";
weather:flags_dictionary =
"fine=http://sweet.jpl.nasa.gov/2.3/stateVisibility.owl#Sunny
cloudy=http://sweet.jpl.nasa.gov/2.3/stateVisibility.owl#Cloudy
...";
```

Vocabularies and dictionaries for terms other than variable enumerations can be constructed using the patterns described above.

8.2 General Metadata

8.2.1 External Metadata

Although not strictly part of the ACDD conventions, a `metadata_link` attribute has been introduced to allow references to more complete metadata(NOAA GEO-IDE, 2011).

Since a WaterML model is a profile of an O&M model, the observation can reference a metadata object that contains ISO 19115 metadata. If the NetCDF document has been generated from a WaterML document that can be referenced by a URI, the `metadata_link` attribute can be set to the document fragment that contains the observation metadata.

More generally, an instance of WaterML is likely to contain a number of references to external objects. Features of interest (FOIs), process descriptions, datums and other objects likely to be common to multiple observations are all likely to be included as

[4] In keeping with common programming language syntax for dictionaries, key:value would be preferred. However, URNs also use colon separators.

references. Generally, these objects tend to be ancillary contextual information, not immediately necessary to the comprehension of the data. The major exception to this rule is that it may be necessary to process the FOI so that location data can be extracted.

A linked open data approach to external objects is taken(Lefort, 2009). Links to external objects can be added as attributes. Wherever possible, a pair of attributes is used. The basic attribute contains the name or title of the object. The link attribute contains the URI of the object. This approach modifies the NetCDF linked data pattern proposed in (Bigagli, Lorenzo, & Nativi, 2011) to cover multiple links and to keep with the metadata link pattern in the ACDD conventions. For example, a reference to a sampled feature might be:

```
:sampled_feature = “Lake George”;
:sampled_feature_link = “http://www.example.com/foi/LG”;
```

8.2.2 Embedded Metadata

There are a large number of attributes in the CF and ACDD conventions that can be directly translated from and to elements of a WaterML model. Section 10 discusses the mappings between the WaterML model and specific attributes. Even if this metadata is repeated, either internal to the NetCDF document or via an external link, these attributes should be present, if possible, since they are the attributes most likely to be used by processing software.

Complete metadata on observations, results or individual data points may also need to be included, if the document is intended as an archival form of the model. The metadata attached to even a single data point of a WaterML timeseries can be quite complex. As a consequence, attempting to provide NetCDF-style attributes or flags for all but the most common and useful metadata would represent a great deal of additional complexity for little return.

Instead, complete metadata can either be added to attributes or variables in the form of XML documents conforming to the WaterML XML encoding. Each document would need to be a stand-alone document, including namespace declarations. A reader could then reconstruct the WaterML model by processing the documents and attaching them to the model. **(enhanced model)** Use of XML in variables, when complete per-point metadata is included in the document, necessitates the use of the string data type from the enhanced model.

8.2.3 Default Metadata

WaterML 2.0 allows the specification of default metadata values for certain things, such as units or interpolation types. If a NetCDF dataset is imported into a WaterML model, sensible default metadata needs to be constructed. There are two strategies for constructing the default metadata:

- Require an importer to scan the dataset and construct defaults.
- Introduce additional attributes that describe the defaults.

Additional attributes are attractive, since they enrich the semantic data available without imposing a great deal of processing overhead.

The interpolation type metadata corresponds to a timeseries implementation of CF conventions cell methods. If a timeseries has a single, consistent interpolation type, then the cell methods variables can be used to accurately characterise the time bounds of the interpolation type – see Section 8.7.

8.3 Units

Units of measure in WaterML are specified using the Unified Code for Units of Measure (UCUM)(Schadow & McDonald, 2009). NetCDF conventions use the UDUNITS-2 unit database (Unidata Program Center, 2011). Both of these systems provide a mechanism for the specification of units in an unambiguous manner.

In most cases, the differences between UCUM and UDUNITS-2 systems are syntactic and the units can be readily translated. One exception is that NetCDF attributes referring to longitude and latitude use units of `degrees_east` and `degrees_north`, attaching an additional level of interpretation to the unit. In practice, this extra requirement is unlikely to represent a difficulty, as it should be obvious when a location is being used.

The WaterML standard allows changes in units within the timeseries by per-point metadata. NetCDF variables have a single unit for the entire variable. NetCDF-encoded WaterML, therefore, needs to have a single default unit for all measurements.

8.4 Categories and Measurements

Simple measurements in WaterML are UML double precision numbers, represented by XML schema doubles in the XML encoding.

Categories represent qualitative data that, in WaterML, are described by terms drawn from a controlled vocabulary. Category data can be either encoded as an array of characters, or a string in the enhanced model, or by creating an enumeration and providing a dictionary – see Section 8.1.4.

A simple approach to category data is to simply use arrays of characters and provide the category names for each data point. However, classic model character arrays are of a fixed size and tend to be wasteful. Instead, existing conventions such as the Argo float conventions tend to encode category data as enumerations with the meaning of the enumeration given in a reference document. This approach is considerably more space-efficient and emphasises the controlled nature of the terminology. Since the category vocabulary is known ahead of time, the enumeration/dictionary approach is preferred.

8.5 Locations

NetCDF timeseries that conform to the CF conventions need to be associated with location information. A single static station timeseries has a point location. A collection of static stations has a location associated with each station. A trajectory has a location associated with each measurement. The conventions for describing CF discrete sampling geometries are discussed in Section 9.

Hydrological observations are generally of a feature of interest (FOI) – a river, a lake or some other geographical feature. WaterML observations are generally, but not exclusively, generated by sensors located at monitoring points. WaterML uses the O&M sampling features model to describe one or more monitoring points that sample a larger FOI. If the observation FOI is a sampling feature, timeseries locations can be derived from the sampling points described in the FOI.

In the case of sampling features, the sampled feature is generally an important part of the dataset metadata. The sampled feature should also be referenced in the NetCDF document – see Section 10.4.1.

If the FOI is not a sampling feature, the timeseries location will need to be derived from the FOI itself. If the timeseries represents a value that covers the entire feature, the CF spatial cell boundaries, using a p-sided cell for irregular features, can be used to describe the location of the timeseries.

If the location data can be derived from a timeseries itself, for example a series of GPS readings from a mobile sensor, the trajectory feature type can be used to encode the location timeseries – see Section 9.

8.6 Coordinate Reference Systems

The CF conventions provide a mechanism for specifying the coordinate reference system (CRS) used when specifying locations. A suitably named scalar variable contains attributes that describe the coordinate system, with the main type of the coordinate system described by a `grid_mapping_name` attribute. A variable that uses the specified coordinate system uses the `grid_mapping` attribute to name the variable that describes the

Individual points, such as monitoring points can usually be reliably geolocated and can be expressed as latitude and longitude. GML locations usually have a reference system specified by the srsName attribute, which can refer to an EPSG (International Association of Oil & Gas Producers, 2011) OGC URN(Whiteside, 2007). For example, the URN of *urn:ogc:def:crs:EPSG::4326* refers to the WGS84 CRS, which can be specified using the CF conventions as:

```
variables:
  int WGS84 ;
    WGS84:grid_mapping_name = "latitude_longitude";
    WGS84:longitude_of_prime_meridian = 0.0 ;
    WGS84:semi_major_axis = 6378137.0 ;
    WGS84:inverse_flattening = 298.257223563 ;
```

Horizontal grids usually use some sort of projection. The CF conventions provide a mechanism for specifying non latitude/longitude coordinates. True lat/long coordinates are supplied by the coordinates attribute. The `grid_mapping` attribute refers to the coordinate reference system used by the grid. The following example shows a mapping for a timeseries where the sampling grid is based on the Map Grid of Australia 1994 in zone 55(Intergovermenal Committee on Surveying and Mapping, 2009)

```
dimensions:
  y = 10;
  x = 10;
  station = 1;
  t = unlimited;
variables:
  double x(station, x) ;
    x:standard_name = "projection_x_coordinate" ;
    x:units = "m" ;
  double y(station, y) ;
    y:standard_name = "projection_y_coordinate" ;
    y:units = "m" ;
  double lat(station, y, x);
    lat:units = "degrees_north";
  double lon(station, y, x);
    lat:units = "degrees_east";
  float temperature(station, time, y, x) ;
    temperature:standard_name = "temperature"
    temp:long_name = "water temperature" ;
    temp:units = "degC" ;
    temp:coordinates = "lat lon" ;
    temp:grid_mapping = "MGA94_55" ;
  int MGA94_55 ;
    MGA94_55:grid_mapping_name = "transverse_mercator";
    MGA94_55:scale_factor_at_central meridian = 0.9996 ;
    MGA94_55:longitude_of_central_meridian = 147.0 ;
    MGA94_55:latitude_of_projection_origin = 0.0 ;
    MGA94_55:semi_major_axis = 6378137.0000;
    AMG84_55:semi_minor_axis = 6356752.3141;
    AMG84_55:inverse_flattening = 298.257222101;
    AMG84_55:false_easting = 500000.0 ;
    AMG84_55:false_northing = 10000000.0;
```

In this example, the x and y variables contain the x and y coordinates of the grid, according to the MGA94 grid. The lat and lon variables map the x and y coordinates onto latitudes and longitudes.

8.7 Representation of Aggregated Values

Many WaterML measurements represent some sort of aggregate value collected over a time period – for example, a maximum temperature. The WaterML interpolation type describes the aggregation process. The following section provides a way of explicitly representing the aggregation process in a form that conforms to the cell representation of the CF conventions. The resulting data mixes a number of WaterML elements together in a way that is likely to make it impossible to reconstruct WaterML from the resulting cell representation. The results of this section simply provide a representation of aggregated data that can be interpreted by suitable CF-aware software.

The CF conventions describe a way of specifying the cell boundaries for measurements that represent some characteristic of the entire cell in a grid, rather than point measurements. An example cell would be the processed results from a series of flow meters strung across a river; the result from each flow meter represents the total flow in a cell representing a section of the river.

The WaterML specification does not explicitly define how spatial cells are specified, since monitoring points have a point geometry and the current WaterML profile does not explicitly cover the results of the sort of complex data products that might produce grids(Taylor, 2010, p. 11). For this reason, spatial cells are not considered in this document beyond noting that a non-sampling FOI could be used to define cell boundaries – see Section 8.5.

Cells can also refer to cell in the temporal, as well as spatial, dimension, for timeseries that represent accumulated values collected over some period. An example timeseries cell would be the measurements from a tipping bucket.

The WaterML specification does provide enough information to build cell boundaries for cumulative measurements. The default interpolation type is used to model the type of accumulation. If the interpolation type specifies a preceding-type of interpolation, then the cell boundaries are constructed between the time of the previous data point and the time of the current data point. If the interpolation type is a succeeding-type of interpolation, then the cell boundaries are constructed between the time of the current data point and the time of the next data point. If the start or end anchor points exist, then they can be used as the preceding or succeeding times for the first and last points in the timeseries respectively. If anchor points do not exist, then the temporal extent of the timeseries can be used to mark the start and end of cell bounds. If the time distance between two points exceeds the `maxGapPeriod`, then the bounds start or end at the `maxGapPeriod` before or after the current point.

If a timeseries is cumulative, then natural cell boundaries may be better represented by the accumulation anchor time and accumulation interval. However, the cell boundaries conventions reflect the boundaries of individual data points, rather than cumulative sequences. At a future stage, an extension to group timeseries data into samples, in the style of an NCAR-RAF timeseries (see Section 7.6) could be considered to cover this case.

Table 3 gives a summary of time bounds construction. If the time variable is called *time* then the bounds variable can be called *time*`_bounds` and the bounds constructed from the interpolation type using the following table. Similarly, it indicates the `cell_methods` method that applies to the WaterML interpolation type.

Table 3 Interpolation Type – Cell Methods Mapping

Interpolation Type	cell_methods	Lower bound	Upper bound
Continuous	*NA*		
Discontinuous	*NA*		
InstantTotal	point	p	c

AveragePrec	mean	p	c
MaxPrec	maximum	p	c
MinPrec	minimum	p	c
TotalPrec	sum	p	c
ConstPrec	mode?	p	c
AverageSucc	mean	c	s
MaxSucc	maximum	c	s
MinSucc	minimum	c	s
TotalSucc	sum	c	s
ConstSucc	mode?	c	s

Where *p* means the time of the preceding point, or the start anchor point, beginning time extent or time of the current point less the maximum gap, as appropriate, *c* means the time of the current point and *s* the time of the successor point, modified in the same way as the preceding point.

The constant interpolation types are intended to represent constant values, such as an alarm setting. There is no real equivalent to the constant interpolation type in the CF conventions. The use of mode for the method is based on a constant value being the setting for the majority of the readings gathered in the cell.

As an example, consider the following XML-encoded WaterML 2.0 timeseries:

```
<wml2:MeasurementTimeseries   >
  <wml2:temporalExtent>
    <gml:TimePeriod>
      <gml:beginPosition>2011-11-21T12:27:00+10:00</gml:beginPosition>
      <gml:endPosition>2011-11-21T12:30:00+10:00</gml:endPosition>
    </gml:TimePeriod>
  </wml2:temporalExtent>
  <wml2:metadata>
    <wml2:MeasurementTimeseriesMetadata>
      <wml2:startAnchorPoint>2011-11-21T12:26:00+10:00</wml2:startAnchorPoint>
    </wml2:MeasurementTimeseriesMetadata>
  </wml2:metadata>
  <wml2:defaultPointMetadata>
    <wml2:DefaultTVPMeasurementMetadata>
      <wml2:uom uom="http://www.opengis.net/def/uom/UCUM/0/m"/>
      <wml2:interpolationType
        xlink:href="http://www.opengis.net/def/timeseriesType/WaterML/2.0/MaxPrec"
        xlink:title="Maximum Preceding"/>
    </wml2:DefaultTVPMeasurementMetadata>
  </wml2:defaultPointMetadata>
  <wml2:point>
```

```
    <wml2:MeasurementTVP>
      <wml2:time>2011-11-21T12:27:00+10:00</wml2:time>
      <wml2:value>3.0</wml2:value>
    </wml2:MeasurementTVP>
  </wml2:point>
    <wml2:MeasurementTVP>
      <wml2:time>2011-11-21T12:28:00+10:00</wml2:time>
      <wml2:value>3.5</wml2:value>
    </wml2:MeasurementTVP>
  </wml2:point>
    <wml2:MeasurementTVP>
      <wml2:time>2011-11-21T12:30:00+10:00</wml2:time>
      <wml2:value>3.4</wml2:value>
    </wml2:MeasurementTVP>
  </wml2:point>
  <wml2:point>
    <wml2:MeasurementTVP>
      <wml2:time>2011-11-21T12:31:00+10:00</wml2:time>
      <wml2:value>3.3</wml2:value>
    </wml2:MeasurementTVP>
  </wml2:point>
</wml2:MeasurementTimeseries>
```

which would have the following NetCDF representation:

```
dimensions:
  time = UNLIMITED;
  station = 1;
  bounds = 2;
variables:
  double maximum_stage(station, time);
    maximum_stage:long_name = "maximum stage";
    maximum_stage:units = "m";
    maximum_stage:cell_methods = "time: maximum";
  int time(station, time);
    time:standard_name = "time";
    time:units = "seconds since 2011-11-21 12:27:00 +10:00";
    time:bounds = "time_bounds";
  int time_bounds(station, time, bounds);
data:
  time = 0, 60, 180, 240;
  time_bounds = -60, 0,   0, 60,   60, 180,   180, 240;
  maximum_stage = 3.0, 3.5, 3.4, 3.3;
```

8.8 Ranges

NetCDF variables have a number of attributes designed to provide guidance to a reader on the likely range of a variable. Since NetCDF documents can be large and can be partially read, the range attributes can supply useful hints to tools such as plotting programs. WaterML has time-range metadata but no value-range metadata. Prior to generating a NetCDF encoding, the timeseries needs to be scanned and sensible ranges computed.[5] The attributes that supply range information are shown in Table 4.

Table 4 Standard NetCDF Range Attributes

Attribute	Description	Derivable From
valid_min	The minimum *valid* value for this variable. This may	• Semantic data for the timeseries property – see Section 8.1.3

[5] The case of an unending, stream-like timeseries is not considered in this paper.

	be below the minimum actual value of the variable.	• The actual minimum • The actual minimum rounded down to a human-friendly value (eg. -10.0 for an actual minimum of -7.7)
valid_max	The maximum *valid* value for this variable. This may be below the minimum actual value of the variable.	See valid_min
valid_range	A vector of the valid_min and valid_max if both are defined. This attribute replaces valid_min and valid_max if both are present.	
_FillValue	The value to fill allocated but unused space allocated to the variable.	A value outside the valid range. Ideally, this should be a readily recognisable value an order of magnitude or so beyond the valid range. For example, if the valid minimum was -10.0 then a fill value of -99.9 would be easily recognisable. If the variable is an enumeration, then the missing value (0) should be used.
missing_value	The specific value used to indicate missing data.[6]	Set to the _FillValue

It is also possible to define transformations, using the `scale_factor` and `add_offset` attributes, to pack data. This paper does not consider scale transformations, although they could be applied to timeseries data.

As an example, a timeseries with the values -2.5, 3.4, 5.6, 8.9, -5.6 might have the following range attributes:

```
example:valid_range = -10.0, 10.0;
example:_FillValue = -99.9;
example:missing_value = -99.9;
```

[6] The missing_value attribute is deprecated in some versions of the CF conventions.

8.9 Responsible Parties

The ACDD conventions contain a number of attributes that describe actors and responsible parties. The observation metadata contains ISO 19115 compliant metadata. The GEO-IDE project has mapped a number of ISO 19115 metadata entries onto the ACDD conventions (NOAA GEO-IDE, 2011). These mappings form the core of the mapping between WaterML observation metadata and the ACDD conventions, modified when the WaterML metadata contains more specific information. The mapping is described further in Section 10.2.

8.10 Data Structures

Timeseries in WaterML can be represented as either interleaved time-value pairs or as separate domain-range sequences. Timeseries in NetCDF are always structured in domain-range form, with separate arrays for the timestamp and the value. This is true even if you use the *compound* construct of enhanced mode, since compounds appear to be internally translated into the domain-range form.

To allow NetCDF to be correctly read into a WaterML model, an additional global attribute with the intended structure (interleaved or domain-range) can be used to give a hint to the reader.

8.11 Data Payload Conventions

In the case of NetCDF as a data payload, the NetCDF document represents a timeseries. The parent document will be an O&M observation conforming to the WaterML profile with a `result` that refers to the NetCDF document. If the observation has a suitable URI, then the `metadata_link` attribute in the NetCDF document (see Section 8.2.1) should refer to the observation metadata.

The link to the payload uses the Xlink conventions. In keeping with (Lefort, 2009), the `xlink:role` is either the URI of the NetCDF format or the NetCDF mime type of `application/x-netcdf`

8.12 Reconstruction of WaterML Models from NetCDF documents

This paper largely considers the creation of a NetCDF document from a WaterML model. However, the mappings discussed in Sections 8-10 are intended to allow the construction of a WaterML model from a suitably encoded NetCDF document.

Embedding XML representations of metadata into a NetCDF document (see Section 8.2.2) ensures that a complete model can be constructed. In many cases, embedded XML will not be present and the model needs to be constructed from the data and attributes contained in the document.

In the case of metadata, most NetCDF attributes and ancillary variables are associated with specific elements in the model and can be simply mapped onto those elements. Some parsing may be necessary, where lists of model elements map onto a single attribute. Additional metadata is introduced in Section 10.4 to ensure that terms can be correctly interpreted and mapped onto WaterML terminology.

In the case of measurement timeseries, the timeseries values and metadata can be extracted directly from the variables containing the time, value and ancillary data.

Features of interest are monitoring points, modelled as stations by the timeseries encoding. The names and locations of monitoring points can be derived from the location information contained in the timerseries variables, along with any coordinate reference systems. Additional monitoring point data can be derived from the global attributes.

9 Encoding of Timeseries

The CF conventions for discrete geometries provide a consistent way of describing timeseries.

Timeseries can be recorded over a number of indices. In the following examples, the index names refer to the following dimensions:

Table 5 Timeseries Indices

Index	Description
i	station instance
p	sample index (time)
o	sample index (direction or trajectory)

In many cases, the document will only contain a single station. Where there is a dimension of size one and the dimension refers to some top-level data common to the entire document, the dimension is often dropped and the variables become scalars. This practice improves the readability of NetCDF documents. However, dropping dimensions (and indices) complicates software that needs to read and comprehend the data. Dropping dimensions also decouples timeseries from station information, such as location. Rather than follow this practice, consistency is recommended, with the station index always retained.

Generally, a WaterML document contains a single timeseries, which can be simply mapped onto a single station. If a collection contains timeseries of the same type and time period, then they can be encoded using station indices.

We can use the CF conventions on timeseries for two cases, with an obvious extension possible.

Table 6 Timeseries Feature Types

Feature Type	Use	Data variable	Space-Time Coordinates
timeSeries	Single point sampling station	data(i, p)	x(i), y(i), t(i, p)
timeSeriesProfile	Single point sampling station recording a vertical profile of readings	data(i,p,o)	x(i), y(i), z(i,p,o), t(i,p)
timeSeriesTransect (non-CF feature)	Single point sampling station recording readings along an arbitrary transect with fixed sampling points along the transect.	data(i,p,o)	x(i,o), y(i,o), z(i,o), t(i,p)

9.1 Data Point Metadata

The WaterML standard allows individual data points to have complex metadata, allowing per-measurement variations in units, interpolation types, reasons for missing data, general comments and other information.

The CF conventions allow ancillary variables and bit-masked flags. The following metadata elements can be directly encoded:

- quality: Encoded as a set of standard URIs (see Section 9.11.1 of the WaterML 2.0 standard). The qualify values can be represented as flags.
- nilReason: A reference to either a URI, or an explicit nil reason. The URI references are expected to refer to a standardized vocabulary.
- censoredReason: A reference to either a URI, or an explicit nil reason. The URI references are expected to refer to a standardized vocabulary.
- interpolationType: Encoded as a set of standard URIs (see Section 9.11.2 of the WaterML 2.0 standard). The interpolation types can be represented as flags.
- processing: A reference to a URI from a standardized vocabulary.
- **(enhanced model)** comment: A free text field containing additional information. If comments are sparse or drawn from a restricted vocabulary, comments can be converted into an enumeration.
- relatedObservation: A reference to the URI of another observation
- accuracy: If the accuracy is provided as a simple value, with an associated consistent unit, then it can be encoded as a variable. If the accuracy is a simple id

or label from a controlled vocabulary, then it can be tokenized encoded as an enumeration (see Sections 8.1.2 and 8.1.4). If the accuracy is a reference to a URI from a standardized vocabulary, then it can also be tokenized and encoded as an enumeration.

(enhanced model) The following metadata elements do not have a direct encoding. If these elements are to be included, they need to be encoded as XML – see Section 0.

- qualifier: A series complex field containing further quality information
- uom: Individual changes to units of measure. See Section 8.3
- accuracy: Accuracy values that do not have a simple numeric representation.

If the is no per-point metadata, then flags do not need to be used. Instead, the default metadata attributes can be used (see Section 8.2.3). If per-point metadata is needed, then the following variables can be used:

`name_quality`, where *name* is the name of the variable, encodes the common quality variables. The `flag_masks` and `flag_meanings` encode the enumeration as follows:

```
name_quality:flag_masks = 7b;
name_quality:flag_values = 1b, 2b, 3b, 4b, 5b, 6b;
name_quality:flag_meanings = "good suspect estimate poor unchecked missing";
```

`name_interpolation_type`, where *name* is the name of the variable, encodes the common interpolation types. The `flag_masks` and `flag_meanings` encode the enumeration as follows:

```
name_interpolation_type:flag_masks = 15b;
name_interpolation_type:flag_values = 1b, 2b, 3b, 4b, 5b, 6b,
    7b, 8b, 9b, 10b, 11b, 12b;
name_flags:flag_meanings = "continuous discontinuous instant_total
    average_prec max_prec min_prec total_prec average_succ
    max_succ min_succ total_succ const_prec const_succ";
```

The flag meaning tokens are constructed via the process described in Section 8.1.2. Each token is assigned an enumeration value, with 0 indicating a missing value. The collected enumerations can then be assigned `flag_values`, `flag_masks` and `flag_meanings` as appropriate.

`name_nil_reason` and `name_censored_reason` are enumerations of the nil or censored URIs or reasons, using the construction method described above.

`name_processing`, `name_related_observation` and `name_accuracy` are an enumeration of the processing URIs, using the construction method described above.

(enhanced model) `name_comment` is an enhanced model string containing any additional comments. Similarly, `name_metadata` contains the complete XML of any data point metadata.

`name_accuracy` is a double value, if the accuracy gives a numeric accuracy. The `name_accuracy:unit` attribute is set to the common unit. Accuracies may also be drawn from an enumeration, in a similar manner to nil or censored reasons.

(enhanced model) If there is a need for complete metadata, then `name_metadata` is an enchanced model string that contains the XML encoding of the point metadata.

All ancilliary variables must be listed in the `name:ancillary_variables` attribute, in conformance with the CF conventions.

9.2 Simple point measurement timeseries

A simple point timeseries represents the most common WaterML use case. This case corresponds to a set of monitoring points along some feature of interest. Simple timeseries are encoded using the CF conventions for timeseries using the timeSeries feature type. Information about the sampling point is contained in the `station_name` variable, which is assigned the role of "timeseries_id".

The CF conventions draw a distinction between the nominal, default or preferred positions of a sample and the actual or precise positions of a sample. This distinction allows sensors that are subject to drift, such as tethered buoys or floating platforms, to be correctly handled. The monitoring point feature of WaterML is designed to record the nominal position of a sensor. WaterML does not handle precise location information for an individual timeseries point explicitly, since the value of a measurement can only be a single double value and the point metadata does not contain a location element. However, a time-synchronous collection of properties from a single station (see Section 11.1) can be used to provide position information. If an actual measurement location is used, then the axis attribute refers to the nominal location and the coordinates attribute for a variable refers to the actual location.

The following table shows the mapping from a WaterML observation to a NetCDF encoding. In the table, *name* is the standard name for the property, mapped using the strategies described in Section 8.1.1. The dimensions *station* and *time* are the i and p indices, respectively. The *identifier* dimension is the maximum station name length. A WaterML column that contains 'Set to "*value*"' means that the attribute is set to the constant value *value*.

Table 7 Timeseries Encoding

Variable	Attribute	Description	WaterML
	featureType	A succinct description of what is in the dataset.	Set to "timeSeries"

char station_name(station,identifier)		The station identifier. Note that the timeseries_id identifies the station, rather than the specific timeseries.	om:OM_Observation/om:featureOfInterest/*sam:SF_SamplingFeatureType*/gml:identifier or om:OM_Observation/om:featureOfInterest/ *sam:SF_SamplingFeatureType* /gml:name
	standard_name		Set to “station_name”
	long_name		om:OM_Observation/om:featureOfInterest/ *sam:SF_SamplingFeatureType* /gml:description
	cf_role		Set to “timeseries_id”
double lat(station)		The latitude of the monitoring point	om:OM_Observation/om:featureOfInterest/ *sams:SF_SpatialSamplingFeatureType* /sams:shape/gml:Point/gml:pos[0]
	standard_name		Set to “latitude”
	long_name		Set to “station latitude” or “station nominal latitude”
	units		Set to “degree_north”
	axis		Set to “Y”
double lon(station)		The longitude of the monitoring point	om:OM_Observation/om:featureOfInterest/ *sams:SF_SpatialSamplingFeatureType* /sams:shape/gml:Point/gml:pos[1]
	standard_name		Set to “longitude”
	long_name		Set to “station longitude” or “station nominal longitude”

	units		Set to “degree_east”
	axis		Set to “X”
double alt(station)		The altitude of the monitoring point	om:OM_Observation/om:featureOfInterest/ *sams:SF_SpatialSamplingFeatureType* /sams:shape/gml:Point/gml:pos[2]
	standard_name		Set to “altitude”
	long_name		Set to “station altitude above the geoid”
	units		Set to “m”
	axis		Set to “Z”
	positive		Set to “up”
int time(station,time)		The measurement times. The WaterML time element can have a wide set of representations, including references to indeterminate positions. These representations need to be reduced to a single consistent timestamp.	om:OM_Observation/om:result/wml2:MeasurementTimeseries/wml2:point/wml2:MeasurementTVP/wml2:time converted to the units given by the units attribute or the times generated from om:OM_Observation/om:result/wml2:MeasurementTimeseries/wml2:metadata/wml2:MeasurementTimeseriesMetadata/{wml2:baseTime, wml2:spacing}
	standard_name		Set to “time”
	long_name		Set to “sample time”
	units		Set to “seconds since 1970-01-01 00:00:00 UTC ” or similar resolution and reference point

	axis		Set to “T”
	calendar		Set to “standard” (Gregorian, since crossing the Gregorian/Julian boundary is not expected.)
double/byte *name*(station,time)	The measurement or category value		om:OM_Observation/om:result/wml2:MeasurementTimeseries/wml2:point/wml2:MeasurementTVP/wml2:value or the missing_value derived from Section 8.8
	standard_name		*name*
	long_name		om:Observation/om:observedProperty/@xlink:title See Section 8.1.2
	units		om:OM_Observation/om:result/*wml2:MeasurementTimeseriesType*/wml2:defaultPointMetadata/*wml2:DefaultTVPMetadata*/wml2:uom/@uom Unit names will need to be translated, see Section 8.3.
	coordinates		Set to “time lat lon alt”
	calendar		Set to “standard” (Gregorian, since crossing the Gregorian/Julian boundary is not expected.)
	_FillValue, missing_value etc.		See Section 8.8
	flag_masks, flag_values, flag_meanings, flag_vocabulary, flag_vocabulary_typeof, flag_dictionary		For category data, see Section 8.1.4

	ancillary_variables		Variables that supply additional metadata, see Section 9.1
byte *name*_quality(station,time)		Additional metadata See Section 9.1	om:OM_Observation/om:result/wml2:MeasurementTimeseries/wml2:point/wml2:MeasurementTVP/wml2:metadata/wml2:quality
	standard_name		*Name* quality
	long_name		Set to "quality for *name*"
	_FillValue		Set to 0
	flag_masks		See Section 9.1
	flag_values		See Section 9.1
	flag_meanings		See Section 9.1
byte *name*_interpolation_type(station,time)		Additional metadata See Section 9.1	om:OM_Observation/om:result/wml2:MeasurementTimeseries/wml2:point/wml2:MeasurementTVP/wml2:metadata/wml2:interpolationType
	standard_name		*Name* interpolation_type
	long_name		Set to "interpolation type for *name*"
	_FillValue		Set to 0
	flag_masks		See Section 9.1
	flag_values		See Section 9.1
	flag_meanings		See Section 9.1
byte *name*_nil_reason(station, time) (and others)		Additional metadata See Section 9.1	om:OM_Observation/om:result/wml2:MeasurementTimeseries/wml2:point/wml2:MeasurementTVP/wml2:metadata values

	standard_name		*Name* nil_reason
	long_name		Set to "Missing reason for *name*"
	_FillValue		Set to 0
	flag_masks		See Section 9.1
	flag_values		See Section 9.1
	flag_meanings		See Section 9.1
string *name*_comment(station,time) (and others) **(enhanced model)**		Comment metadata, possibly encoded as an enumeration See Section 9.1	om:OM_Observation/om:result/wml2:MeasurementTimeseries/wml2:point/wml2:MeasurementTVP/wml2:metadata/wml2:comment
	standard_name		*Name* comments
	long_name		Set to "Additional comments for *name*"
string *name*_metadata(station,time) **(enhanced model)**		Additional metadata See Section 9.1	om:OM_Observation/om:result/wml2:MeasurementTimeseries/wml2:point/wml2:MeasurementTVP/wml2:metadata as an XML document
	standard_name		*Name* metadata
	long_name		Set to "Metadata for *name*"

9.3 Complex Timeseries

The WaterML model is not restricted to simple point timeseries. An abstract timeseries may also contain higher-dimensional data or record-like structures. Although the WaterML model allows higher dimensional and record-like timeseries, the standard concentrates on the more simple timeseries gathered from in-situ observations. The following sections describe two common cases that can be expected when complex timeseries are used.

9.3.1 Higher-Dimensional Timeseries

One-dimensional, two-dimensional and higher dimensional measurements represent either the results from a sensor that measures along a transect or over an area, or the synthetic results of a complex process. An example of a higher-dimensional sensor is a rain radar. An example of a complex process is a soil moisture map constructed by processing a set of individual samples, integrating the results with a topographic model and producing a 2D coverage.

In most cases, higher dimensional timeseries can be modelled by extending the simple timeseries conventions used in Section 9.2 to have a value that consists of a 1D or 2D array. The following additional considerations apply:

- In many cases, data is still being gathered from stations and the `station_name` variable can still be used to capture the station identifier. Synthetic data can either discard the station identification completely or include the name of the process as the station name.
- In some cases, such as fixed transects or grids, the feature type can be modelled by using an extension to the feature types defined in the CF discrete sampling geometries. In those cases, a set of distinct feature type names needs to be identified. Table 8 contains an incomplete list of suggested feature types.
- A network of monitoring stations, each with a sampling grid, may use different grid mappings (see Section 8.6) for the station location and for the grid. For example, monitoring points may be located via GPS, each with a sensor grid laid out according to a transverse Mercator map grid. Generally, grid values are eventually mapped down to longitude and latitude variables. However, station locations may need to be encoded as a separate variable.

Table 8 Hydrology Feature Types

Observation Type	CF Feature Type	CSML Observation Type	Name	Variables
Simple monitoring point	timeSeries	PointSeries	timeSeries	data(i,p), x(i), y(i), z(i), t(i, p)
Horizontal transect from a fixed monitoring point			timeSeriesTransect	data(i,p,o), x(i,o), y(i,o), z(i), t(i,p)
Horizontal grid		GridSeries[7]	timeSeriesGrid	data(i,p,o_1, o_2),

[7] CSML defines a GridSeries as a series of volumes

from a fixed point or a process				$x(i,o_1,o_2)$, $y(i,o_1,o_2)$, z(i), t(i,p)
Trajectory from a floating sensor	trajectory	Trajectory	trajectory	data(i,p), x(i,p), y(i,p), z(i,p), t(i,p)

9.3.2 Record Timeseries

Record timeseries represent collections of sensors that produce synchronised readings. An example of a record timeseries would be a weather station that provides a block of temperature, humidity and wind-speed values for each sample.

Record samples can be represented in NetCDF using the same model as the simple timeseries described in Section 9.2. Each element of the record structure can be assigned a separate variable name, based on the standard name of the property being measured.

10 Observation and Timeseries Metadata

The various NetCDF metadata conventions allow a NetCDF document to be self-describing. The following sections describe conventions-specific metadata mappings between WaterML and NetCDF. The overall strategy used in mapping metadata is to map the metadata associated with the O&M observation onto global attributes and metadata associated with a specific timeseries onto variable attributes.

10.1 NetCDF Attribute Conventions

The NetCDF attribute conventions provide a set of recommended attributes that will be recognized by most applications that process NetCDF data. Where possible, the conventions followed by NOAA's Global Earth Observation Integrated Data Environment (GEO-IDE) are used (NOAA GEO-IDE, 2011).

The CF conventions only allow Conventions to be set to "CF-1.6" There is a proposal for relaxing the rules and allowing space or comma-seperated conventions.[8]

10.1.1Global Attributes

Table 9 NetCDF Attribute Conventions – Global Attributes

Attribute	Description	WaterML
title	A succinct description of what is in the dataset.	om:OM_Observation/gml:name or

[8] https://cf-pcmdi.llnl.gov/trac/ticket/76

		om:OM_Observation/gml:result/@xlink:title or om:OM_Observation/gml:description or om:OM_Observation/om:metadata/wml2:ObservationMetadata/gmd:identificationInfo/gmd:MD_DataIdentification/gmd:citation/gmd:CI_Citation/gmd:title/gco:CharacterString
history	An attribute containing a line for each modification of the dataset. Applications should append a line giving date, time, user, program and arguments.	om:OM_Observation/om:metadata/wml2:ObservationMetadata/gmd:dataQualityInfo/gmd:DQ_DataQuality/gmd:lineage/gmd:LI_Lineage/gmd:statement/gco:CharacterString
Conventions	A list of the conventions, separated by spaces, used by the document	Set to the space-separated concatenation of “CF-1.6/ancillaryData”, “CF-1.6/coordinates”, “CF-1.6/cells”, “CF-1.6/timeSeries” and the standard name convention chosen.

10.1.2 Per-Timeseries Attributes

Table 10 NetCDF Attribute Conventions – Timeseries Attributes

Attribute	Description	WaterML 2.0
units	Units expressed in the UDUNITS library format	om:OM_Observation/om:result/*wml2:MeasurementTimeseriesType*/wml2:defaultPointMetadata/*wml2:DefaultTVPMetadata*/wml2:uom/@uom Unit names will need to be translated, see Section 8.3.
long_name	A long, descriptive name.	om:Observation/om:observedProperty/@xlink:title See Section 8.1.2
_FillValue	The value used to pre-fill disk space allocated to the variable, indicating values that have not been written.	See Section 8.8

missing_value	A value outside the valid range of the variable, indicating a missing value.	See Section 8.8
valid_min	The minimum valid value for the variable	See Section 8.8
valid_max	The maximum valid value for the variable	See Section 8.8
valid_range	A vector giving the minimum and maximum values	See Section 8.8
scale_factor	The scale factor to apply to the stored data	Not used – see Section 8.8
add_offset	The offset to apply to the stored data	Not used – see Section 8.8
signedness	Deprecated	Do not use.
C_format	The format used by C applications to print values for this variable.	Possibly derived from an RDF property description. See Section 8.1.2
FORTRAN_format	The format used by FORTRAN applications to print values for this variable.	Possibly derived from an RDF property description. See Section 8.1.2

10.2 Attribute Convention for Dataset Discovery

The NetCDF Attribute Convention for Dataset Discovert (ACDD) describes a dataset to discovery systems, such as digital libraries. The NOAA GEO-IDE project has produced a mapping from ACDD attributes to ISO 19115-2 metadata. Where possible, the conventions followed that mapping are used, adjusted where more explicit data is available. The ACDD attributes do not cover the complete metadata set; if complete metadata exchange is required (e.g. the INSPIRE rules for exchange metadata(INSPIRE Drafting Team "Data Specifications", 2010)) then either embedded XML (see Section 8.2.2) or a data payload model (see Section 6.2) should be used.

A great deal of metadata is buried in the observation citation metadata, with codes identifying various roles and events. WaterML also contains observation-level metadata. Which source of metadata takes precedence is largely based on how precise the mapping is between the ACDD attribute and the metadata term.

To locate responsible parties in the citation metadata, **RP[*code*]** stands for om:OM_Observation/om:metadata/wml2:ObservationMetadata/gmd:contact/gmd:citedResponsibleParty/gmd:CI_ResponsibleParty[gmd:role/gmd:CI_RoleCode=*code*] **or** om:OM_Observation/om:metadata/wml2:ObservationMetadata/gmd:identificationInfo/gmd:MD_DataIdentification/gmd:citation/gmd:CI_Citation/gmd:citedResponsibleParty/gmd:CI_ResponsibleParty[gmd:role/gmd:CI_RoleCode=*code*]

To locate dates in the citation metadata, **Date[*code*]** stands for om:OM_Observation/om:metadata/wml2:ObservationMetadata/gmd:identificationInfo/gmd:MD_DataIdentification/gmd:citation/gmd:CI_Citation/gmd:date/gmd:CI_Date/gmd:date/gco:Date[gmd:dateType/gmd:CI_DateTypeCode=*code*]/gmd:date

Geospatial bounding boxes assume that the coordinates in envelopes or other bounding boxes are in latitude, longitude and altitude in metres, using the WGS 84[9] coordinate reference system.

10.2.1 Global Attributes

Table 11 ACDD Conventions - Global Attributes

Attribute	Description	WaterML 2.0
Metadata_Conventions	Set to indicate that the file conforms to the ACDD conventions	Set to "Unidata Dataset Discovery v1.0"
id	The unique identifier for the dataset	om:OM_Observation /gml:identifier or om:OM_Observation/om:metadata/wml2:ObservationMetadata/gmd:fileIdentifier/gco:CharacterString
naming_authority		If om:OM_Observation/om:metadata/wml2:ObservationMetadata/gmd:fileIdentifier is of type gml:CodeType then use the @codeSpace attribute.
summary	A paragraph describing the dataset	om:OM_Observation/gml:description if not used as the title or

[9] Identified by http://www.opengis.net/def/crs/OGC/1.3/CRS84, urn:ogc:def:crs:OGC:1.3:CRS84 or *urn:ogc:def:crs:EPSG::4326*

		om:OM_Observation/om:metadata/wml2:ObservationMetadata /gmd:identificationInfo/gmd:MD_DataIdentification/gmd:abstract/gco:CharacterString
keywords	Keywords and phrases	om:OM_Observation/om:metadata/wml2:ObservationMetadata /gmd:identificationInfo/gmd:MD_DataIdentification/gmd:descriptiveKeywords/gmd:MD_Keywords/gmd:keyword/gco:CharacterString Keywords are separated by commas
keywords_vocabulary	The guidelines used for the keywords attribute	om:OM_Observation/om:metadata/wml2:ObservationMetadata/gmd:identificationInfo/gmd:MD_DataIdentification/gmd:descriptiveKeywords/gmd:MD_Keywords/gmd:thesaurusName/gmd:CI_Citation/gmd:title/gco:CharacterString
cdm_data_type	The THREDDS data type	Set to "Station"
date_created	The date on which the data was created.	wml2:Collection/wml2:metadata/wml:DocumentMetdata/wml2:generationDate or om:OM_Observation/om:metadata/wml2:ObservationMetadata/gco:dateStamp or **Date[creation]**
creator_name	The name of the data creator	**RP[originator]**/gmd:individualName/gco:CharacterString
creator_url		**RP[originator]**/gmd:contactInfo/gmd:CI_Contact/gmd:onlineResource/gmd:CI_OnlineResource/gmd:linkage/gmd:URL
creator_email		**RP[originator]**/gmd:contactInfo/gmd:CI_Contact/gmd:onlineResource/gmd:CI_OnlineResource/gmd:linkage/gmd:URL
institution	The creator's institution	**RP[originator]**/ gmd:organisationName/gco:CharacterString

project	The scientific project which produced the data	TBD
processing_level	The processing (quality control level) of the data	om:OM_Observation/om:metadata/wml2:ObservationMetadata/wml2:status
acknowledgement	Acknowledgement of support	om:OM_Observation/om:metadata/wml2:ObservationMetadata/gmd:identificationInfo/gmd:MD_DataIdentification/gmd:credit/gco:CharacterString
geospatial_lat_min	The coordinates of a simple geospatial bounding box.	om:OM_Observation/gml:boundedBy/gml:Envelope/gml:lowerCorner[0] (using the WGS 84 CRS) or om:OM_Observation/gml:featureOfInterest/*gml:AbstractFeatureType*/gml:boundedBy/gml:Envelope/gml:lowerCorner[0] or wml2:Collection/gml:boundedBy/gml:Envelope/gml:lowerCorner[0]
geospatial_lat_max		om:OM_Observation/gml:boundedBy/gml:Envelope/gml:upperCorner[0] (or FOI or Collection bounds)
geospatial_lat_units		Set to “degrees_north”
geospatial_lon_min		om:OM_Observation/gml:boundedBy/gml:Envelope/gml:lowerCorner[1] (or FOI or Collection bounds)
geospatial_lon_max		om:OM_Observation/gml:boundedBy/gml:Envelope/gml:upperCorner[1] (or FOI or Collection bounds)
geospatial_lon_units		Set to “degrees_east”
geospatial_vertical_min		om:OM_Observation/gml:boundedBy/gml:Envelope/gml:lowerCorner[3] (or FOI or Collection bounds)

geospatial_vertical_max		om:OM_Observation/gml:boundedBy/gml:Envelope/gml:upperCorner[3] (or FOI or Collection bounds)
geospatial_vertical_units		Set to "m"
geospatial_vertical_positive		Set to "up"
time_coverage_start	The temporal coverage of the data set	om:OM_Observation/om:result/*wml2:TimeseriesType*/wml2:temporalExtent/gml:TimePeriod/gml:beginPosition or wml2:Collection/wml2:temporalExtent/gml:TimePeriod/gml:beginPosition
time_coverage_end	The temporal coverage of the data set	om:OM_Observation/om:result/*wml2:TimeseriesType*/wml2:temporalExtent/gml:TimePeriod/gml:endPosition or wml2:Collection/wml2:temporalExtent/gml:TimePeriod/gml:endPosition
time_coverage_duration	The temporal coverage of the data set	om:OM_Observation/om:result/*wml2:TimeseriesType*/wml2:temporalExtent/gml:TimePeriod/gml:duration or wml2:Collection/wml2:temporalExtent/gml:TimePeriod/gml:duration
time_coverage_resolution	The temporal coverage of the data set	om:OM_Observation/om:metadata/wml2:ObservationMetadata/wml2:intendedSamplingInterval
standard_name_vocabulary	The name of the controlled vocabulary which variable standard names are taken.	om:OM_Observation/om:metadata/wml2:ObservationMetadata/gmd:identificationInfo/gmd:MD_DataIdentification/gmd:descriptiveKeywords/gmd:MD_Keywords/gmd:thesaurusName/gmd:CI_Citation/gmd:title/gco:CharacterString Since this term is used by the ACDD conventions, it should not be set to the attribute containing

		the standard name dictionary.
licence	Restrictions on data access and distribution	om:OM_Observation/om:metadata/wml2:ObservationMetadata/gmd:identificationInfo/gmd:MD_DataIdentification/gmd:resourceConstraints/gmd:MD_LegalConstraints/gmd:useLimitation/gco:CharacterString
contributor_name	The name and role of contributors to the data set.	**RP[*]**/gmd:individualName/gco:CharacterString
contributor_role		**RP[*]**/gmd:role/gmd:CI_RoleCode
publisher_name	The publisher of the data set.	**RP[publisher]**/gmd:organisationName/gco:CharacterString or **RP[publisher]**/gmd:individualName/gco:CharacterString
publisher_url		**RP[publisher]**/gmd:contactInfo/gmd:CI_Contact/gmd:onlineResource/gmd:CI_OnlineResource/gmd:linkage/gmd:URL
publisher_email		**RP[publisher]**/gmd:contactInfo/gmd:CI_Contact/gmd:address/gmd:CI_Address/gmd:electronicMailAddress/gco:CharacterString
date_modified	The date the data set was modified	**Date[modified]** or Wml2:Collection/wml2:metadata/wml:DocumentMetdata/wml2:generationDate or om:OM_Observation/om:resultTime
date_issued	The date the data set was issued	**Date[issued]** or Wml2:Collection/wml2:metadata/wml:DocumentMetdata/wml2:generationDate

		or om:OM_Observation/om:resultTime

10.3 Climate and Forecast Conventions

The CF conventions describe a dataset in scientific terms, providing a consistent description of the data collected in a way that can be interpreted by analysis and visualisation packages. A large portion of the CF metadata has already been described in Section 8. This section covers a few pieces of metadata that have yet to be described.

10.3.1Global Attributes

Table 12 CF Conventions - Global Attributes

Attribute	Description	WaterML
featureType	The type of feature sampled by the discrete sampling geometry	If a simple time series (see Section 9.2) then set to "timeSeries"

10.3.2Per-Timeseries Attributes

The `long_name` and `units` attributes are described in Section 10.1.2. See Section 9.2 for a comprehensive mapping for a timeseries.

Table 13 CF Conventions - Timeseries Attributes

Attribute	Description	WaterML
standard_name	A standard name chosen from a domain-specific vocabulary	See Section 8.1.1
axis	The direction of the coordinate axis	Set to X, Y, Z or T to describe the timeseries axis, if this variable is a coordinate axis.

10.4 WaterML Specific Attributes

The following attributes represent useful additional information for WaterML users. Most of these attributes represent descriptive metadata that can be used to reconstruct a WaterML model from the NetCDF encoding.

10.4.1Global Attributes

Table 14 WaterML 2.0 - Global Attributes

Attribute	Description	WaterML
standard_name_vocabulary_link	A link to the standard name vocabulary	The URI of the standard name vocabulary, or set to “standard_name_dictionary” for an embedded standard name dictionary
standard_name_vocabulary_type of	The vocabulary format	The type of standard name vocabulary. See Section 8.1.4.
standard_name_dictionary	A dictionary mapping terms from the standard name vocabulary onto URIs	A dictionary of standard names, constructed according to Section 8.1.4
quality_dictionary	The quality term dictionary	A dictionary of quality terms, constructed according to Section 8.1.4
nil_reason_dictionary	The nil reason dictionary	A dictionary of nil reasons, constructed according to Section 8.1.4
processing_dictionary	The processing dictionary	A dictionary of process types, constructed according to Section 8.1.4
interpolation_type_dictionary	The interpolation type dictionary	A dictionary of interpolation types, constructed according to Section 8.1.4
censored_reason_dictionary	The censored reason dictionary	A dictionary of censored reasons, constructed according to Section 8.1.4
sampled_medium_dictionary	The sampled reason dictionary	A dictionary of sampled medium terms, constructed according to Section 8.1.4
status_dictionary	The measurement status dictionary	A dictionary of status terms, constructed according to Section 8.1.4
process_type_dictionary	The observation process type dictionary	A dictionary of process types, constructed according to Section 8.1.4
accuracy_dictionary	The observation accuracy dictionary, if accuracies are from a set of terms	A dictionary of accuracies, constructed according to Section 8.1.4
intended_sampli	The intended sampling	om:OM_Observation/om:metadata/wml2:O

ng_interval	interval as a simple value	bservationMetadata/wml2:intendedSamplingInterval (amount)
intended_sampling_interval_unit	The unit of the intended sampling interval	om:OM_Observation/om:metadata/wml2:ObservationMetadata/wml2:intendedSamplingInterval (units)
maximum_gap	The maxium time gap between measurements as a simple value	om:OM_Observation/om:metadata/wml2:ObservationMetadata/wml2:maximumGap (amount)
maximum_gap_unit	The unit of the maxium gap	om:OM_Observation/om:metadata/wml2:ObservationMetadata/wml2:maximumGap (units)
sampled_medium	The sampled medium for this timeseries	om:OM_Observation/om:metadata/wml2:ObservationMetadata/wml2:sampledMedium
sampled_medium_vocabulary	The sampled medium vocabulary	Set to "sampled_medium_dictionary" (see Section 10.4.1) or to the URI of the sampled medium vocabulary
sampled_medium_vocabulary_typeof	The sampled medium vocabulary format	The type of sampled medium vocabulary. See Section 8.1.4.
status	A term describing the status of the measurements	om:OM_Observation/om:metadata/wml2:ObservationMetadata/wml2:status
status_vocabulary	The status vocabulary	Set to "status_dictionary" (see Section 10.4.1) or to the URI of the status vocabulary
status_vocabulary_typeof	The vocabulary format	The type of status vocabulary. See Section 8.1.4.
sampled_feature	The name of the feature being sampled	om:OM_Observation/om:featureOfInterest/wml2:MonitoringPoint/sam:sampledFeature/@xlink:title or om:OM_Observation/om:featureOfInterest/@xlink:title if the FOI is not a sampling feature

sampled_feature_link	A reference to the feature being sampled	om:OM_Observation/om:featureOfInterest//wml2:MonitoringPoint/sam:sampledFeature/@xlink:href or om:OM_Observation/om:featureOfInterest/@xlink:href if the FOI is not a sampling feature
gauge_datum	The name of the vertical datum used for monitoring	om:OM_Observation/om:featureOfInterest//wml2:MonitoringPoint/wml2:gaugeDatum/@xlink:title or om:OM_Observation/om:procedure/wml2:ObservationProcess/wml2:gaugeDatum/@xlink:title
gauge_datum_link	A reference to the gauge vertical datum	om:OM_Observation/om:featureOfInterest//wml2:MonitoringPoint/wml2:gaugeDatum/@xlink:href or om:OM_Observation/om:procedure/wml2:ObservationProcess/wml2:gaugeDatum/@xlink:href
monitoring_point_timezone	The timezone that the monitoring point resides in.	om:OM_Observation/om:featureOfInterest//wml2:MonitoringPoint/wml2:timeZone/wml2:zoneOffset
process_type	The observation process type	om:OM_Observation/om:procedure/wml2:ObservationProcess/wml2:processType
process_type_vocabulary	The observation process type vocabulary	Set to "process_type_dictionary" or to the URI of the status vocabulary
process_type_vocabulary_typeof	The vocabulary format	The type of process type vocabulary. See Section 8.1.4.
procedure	The name of the procedure, if referenced	om:OM_Observation/om:procedure/@xlink:title

procedure_link	A reference to the procedure	om:OM_Observation/om:procedure/@xlink:href
originating_process	The name of the source process	om:OM_Observation/om:procedure/wml2:ObservationProcess/wml2:originatingProcess/@xlink:title
originating_process_link	A reference to the source process	om:OM_Observation/om:procedure/wml2:ObservationProcess/wml2:originatingProcess/@xlink:href
process_reference	The name of the process definition	om:OM_Observation/om:procedure/wml2:ObservationProcess/wml2:processReference/@xlink:title
process_reference_link	A reference to the process definition	om:OM_Observation/om:procedure/wml2:ObservationProcess/wml2:processReference/@xlink:href
result_time	The time when the result has come available	om:OM_Observation/om:resultTime
valid_time_start	The start time at which this observation is valid	om:OM_Observation/om:validTime/gml:TimePeriod/gml:beginPosition
valid_time_end	The end time at which this observation is valid	om:OM_Observation/om:validTime/gml:TimePeriod/gml:endPosition
valid_time_duration	The duration over which this observation is valid	om:OM_Observation/om:validTime/gml:TimePeriod/gml:duration
phenomenon_time_start	The start time at which the phenomenon being measured occurs	om:OM_Observation/om:phenomenonTime/.../gml:TimePeriod/gml:beginPoisition if the phenomenon time can be reduced to a time period
phenomenon_time_end	The end time at which the phenomenon being measured occurs	om:OM_Observation/om:phenomenonTime/.../gml:TimePeriod/gml:endPosition if the phenomenon time can be reduced to a time period
phenomenon_time_duration	The duration over which the	om:OM_Observation/om:penomenonTime/.../gml:TimePeriod/gml:duration if the phenomenon time can be reduced to a time

	phenomenon occurs	period
metadata	The observation metadata, expressed as XML	The contents of the om:OM_Observation except for the om:OM_Observation/om:result element

Certain attributes that might be considered global, such as the default code space for a category timeseries, are placed as timeseries attributes.

10.4.2 Timeseries Attributes

Table 15 WaterML 2.0 - Timeseries Attributes

Attribute	Description	WaterML
timeseries_type	The type of WaterML timeseries	Set to either “interleaved” for interleaved timeseries or “domain-range” for domain-range timeseries.
time_base	The base time for generated timestamps	om:OM_Observation/om:result/*wml2:TimeseriesType*/wml2:metadata/wml2:baseTime
time_spacing	The interval for generated timestamps	om:OM_Observation/om:result/*wml2:TimeseriesType*/wml2:metadata/wml2:spacing
quality	The default quality code	om:OM_Observation/om:result/*wml2:TimeseriesType*/wml2:defaultPointMetadata/*wml2:DefaultTVPMetadataType/*wml2:quality
quality_vocabulary	The default quality vocabulary	If quality terms are used, then set to “quality_dictionary” (see Section 10.4.1) or to the URI of the quality vocabulary
quality_vocabulary_typeof	The vocabulary format	The type of quality vocabulary. See Section 8.1.4.
nil_reason	The default nil reason for a point	om:OM_Observation/om:result/*wml2:TimeseriesType*/wml2:defaultPointMetadata/*wml2:DefaultTVPMetadataType/*wml2:nilReason
nil_reason_vocabulary	The default nil reason vocabulary	Set to “nil_reason_dictionary” (see Section 10.4.1) or to the URI of the nil reason vocabulary
nil_reason_voca	The vocabulary format	The type of nil reason vocabulary. See

bulary_typeof		Section 8.1.4.
comment	The default comment for a point	om:OM_Observation/om:result/*wml2:TimeseriesType*/wml2:defaultPointMetadata/*wml2:DefaultTVPMetadataType/*wml2:comment
related_observation	The URI of a default related observation	om:OM_Observation/om:result/*wml2:TimeseriesType*/wml2:defaultPointMetadata/*wml2:DefaultTVPMetadataType/*wml2:relatedObservation/@href
processing	The default processing type for a point	om:OM_Observation/om:result/*wml2:TimeseriesType*/wml2:defaultPointMetadata/*wml2:DefaultTVPMetadataType/*wml2:processing
processing_vocabulary	The default processing vocabulary	Set to "processing_dictionary" (see Section 10.4.1) or to the URI of the processing vocabulary
processing_vocabulary_typeof	The vocabulary format	The type of processing vocabulary. See Section 8.1.4.
start_anchor_point	The start point for cell measurements	om:OM_Observation/om:result/*wml2:TimeseriesType*/wml2:metadata/wml2:MeasurementTimeseriesMetadata/wml2:startAnchorPoint
end_anchor_point	The end point for cell measurements	om:OM_Observation/om:result/*wml2:TimeseriesType*/wml2:metadata/wml2:MeasurementTimeseriesMetadata/wml2:endAnchorPoint
cumulative	Whether this timeseries represents cumulative values	om:OM_Observation/om:result/*wml2:TimeseriesType*/wml2:metadata/wml2:MeasurementTimeseriesMetadata/wml2:cumulative
accumulation_anchor_time	The start point for cumulative measuremens	om:OM_Observation/om:result/*wml2:TimeseriesType*/wml2:metadata/wml2:MeasurementTimeseriesMetadata/wml2:accumulationAnchorTime
accumulation_interval_length	The period of accumulation for cumulative measurements	om:OM_Observation/om:result/*wml2:TimeseriesType*/wml2:metadata/wml2:MeasurementTimeseriesMetadata/wml2:accumulationIntervalLength

accumulation_interval_length_unit	The unit for the accumulation interval	om:OM_Observation/om:result/*wml2:TimeseriesType*/wml2:metadata/wml2:MeasurementTimeseriesMetadata/wml2:accumulationIntervalLength
aggregation_duration	The period of aggregation for cumulative measurements	om:OM_Observation/om:result/*wml2:TimeseriesType*/wml2:metadata/wml2:MeasurementTimeseriesMetadata/wml2:aggregationDuration
aggregation_duration _unit	The unit for the aggregation duration	om:OM_Observation/om:result/*wml2:TimeseriesType*/wml2:metadata/wml2:MeasurementTimeseriesMetadata/wml2:aggregationDuration
flags_mask flags_values flags_meanings flags_vocabulary	The enumeration definition, for categorical timeseries	For categorical timeseries, the enumeration for the category – see Section 8.1.4. The flags_vocabulary may be defined by om:OM_Observation/om:result/*wml2:TimeseriesType*/wml2:defaultPointMetadata/wml2:DefaultTVPCategoricalMetadata/wml2:codeSpace
flags_vocabulary_typeof	The vocabulary format	The type of category vocabulary. See Section 8.1.4.
flags_dictionary		
interpolation_type	The default interpolation type	om:OM_Observation/om:result/*wml2:TimeseriesType*/wml2:defaultPointMetadata/wml2:DefaultTVPMeasurementMetadata/wml2:interpolationType
interpolation_type_vocabulary	The default interpolation type vocabulary	Set to "interpolation_type_dictionary" (see Section 10.4.1) or to the URI of the interpolation type vocabulary
interpolation_type_vocabulary_typeof	The vocabulary format	The type of interpolation type vocabulary. See Section 8.1.4.
censored_reason	The default censored reason	om:OM_Observation/om:result/*wml2:TimeseriesType*/wml2:defaultPointMetadata/wml2:DefaultTVPMeasurementMetadata/wml2:censoredReason
censored_reason	The default censored	Set to "censored_reason_dictionary" (see Section 10.4.1) or to the URI of the censored

_vocabulary	reason vocabulary	reason vocabulary
censored_reason_vocabulary_typeof	The vocabulary format	The type of standard name vocabulary. See Section 8.1.4.
accuracy	The default accuracy, if accuracy can be expressed as a number	om:OM_Observation/om:result/*wml2:TimeseriesType*/wml2:defaultPointMetadata/wml2:DefaultTVPMeasurementMetadata/wml2:accuracy/swe:Quality/swe:value
accuracy_unit	The unit used to express the accuracy	om:OM_Observation/om:result/*wml2:TimeseriesType*/wml2:defaultPointMetadata/wml2:DefaultTVPMeasurementMetadata/wml2:accuracy/swe:Quality/swe:uom
accuracy_vocabulary	The default censored reason vocabulary, if accuracy is from a set of terms	Set to “accuracy_dictionary” (see Section 10.4.1) or to the URI of the accuracy vocabulary
accuracy_vocabulary_typeof	The vocabulary format	The type of accuracy vocabulary. See Section 8.1.4.
metadata	The timeseries metadata, expressed as XML	The contents of the om:OM_Observation/om:result element except for the wml2:point elements – see Section 0

11 Profiles

The mappings described in Sections 8-10 are intended to provide for the following cases:

- A complete mapping from a WaterML model to a NetCDF document and its reverse. This complete mapping may be required for archival and other purposes, but may not be particularly useful for processing, data provision and analysis.
- An adequate mapping between WaterML and NetCDF, where some data will be lost. Adequate, in this case, means that the resulting document will be fit for its intended purpose. It is expected that this case will be the more common case.

Some application profiles of WaterML are necessary to allow representation as NetCDF. This section discusses the restrictions necessary.

11.1 Collections

The mappings described in Sections 8-10 are largely built on individual WaterML observations, with the observation itself representing the dataset and the result being encoded as a set of variables and ancillary variables. The WaterML timeseries standard also allows collections of related timeseries.

- Multi-station or profile data of the same property. For example, a series of flow meters along a river course. Another example would be a profile of soil moisture at different depths. Multi-station data is represented by a station index in the CF timeseries conventions discussed in Section 9. Profile data can be represented as a timeSeriesProfile using the CF timeseries conventions.
- Multi-property data from a single station. For example, weather data from a weather station. The samples would need to have synchronous timestamps, so that they can use a common time variable in NetCDF. This sort of data could be used to provide actual position data for drifting monitoring points.

In the case of collections of observations, the observation metadata in the collection needs to be *compatible*: all global attribute mappings described in Section 10 must either derive from the document metadata or map onto the same value for each observation.

11.2 Use Case Profiles

The following table provides a summary of the mapping elements required or recommended for each use case.

Table 16 Profile Requirements

Mapping	Section(s)	Use Case			
		Archival	Data Payload	Convert NetCDF to WaterML	Import Export
External Metadata	8.2.1	O	R	U	O
XML Metadata	8.2.2	O[10]	-	U	-
Dictionaries	8.1.4	M	R	U[11]	R

[10] XML metadata may be necessary if the original WaterML model contains more information than can be represented by standard attributes (for example, complex data point accuracy information or timeseries comment blocks).

Default Metadata	8.2.3	M	M	R	O
Common Units	8.3	M	M	M	M
Coordinate Reference Systems	8.6	M	R	O	R
Cell Representation	8.7	O	O	U[12]	O
Data Point Metadata[13]	9.1	M	M[14]	U	R
Timeseries Conventions	9	M	M	M	M
Common NetCDF Metadata	10.1	M	M	M	M
ACDD Metadata	10.2	M	M	R	R
CF Metadata	10.3	M	M	M	M
WaterML Metadata	10.4	M	R	U	R
	M mandatory, R recommended, O optional, U useful if present				

12 Summary

This discussion paper attempts to provide a mapping between the WaterML 2.0 timeseries conceptual model and the NetCDF format and conventions, particularly the CF timeseries conventions. The resulting mapping covers most uses of NetCDF-encoded

[11] If dictionaries are not present, process-specific mapping of terms onto WaterML URIs and terms will need to be used.

[12] The cells contained within a document using the CF conventions do not have an explicit representation in the WaterML model. However, the cell boundaries could be used to infer the WaterML interpolation type, anchor points, maximum gap etc. if this data is not explicitly provided.

[13] As required. If default metadata suffices for a particular type of metadata, then individual data point metadata should not be used.

[14] Individual profiles of data point metadata may be chosen to ensure that the resulting NetCDF document adheres to the classic data model. For example, per-point comments may be discarded.

WaterML, although there are points where the inherent potential complexity of WaterML defeats a clear, direct mapping onto NetCDF.

In constructing the mapping, a number of issues have been identified. In most cases, potential solutions have been provided. However, the issues flag areas that might be usefully explored at a more abstract level than the context of WaterML. The main issues identified are:

- The CF conventions provide a complete package of conventions for handling timeseries and associated metadata. However, the conventions are strongly influenced by their origins in terms of vocabulary and sampling geometries. Splitting the CF conventions into domain-independent and domain-dependent parts and being able to specify domain-specific vocabularies and methods would ease the process of mapping the conventions onto the hydrology domain.
- WaterML uses the patterns of semantic and linked open data. The enumeration, dictionary and linking methods described in this paper attempt to bridge the gap between the convention-oriented NetCDF patterns and the link-oriented WaterML patterns. As the use of linked data becomes more common an established set of conventions for NetCDF would be useful.
- Ultimately, it is not completely possible to encode a complex object model, such as WaterML, into the simpler classic NetCDF data model. The embedding of XML into NetCDF is a stop-gap. The nested structure of the enhanced data model suggests that it may be able to provide a more complete encoding. However, a set of conventions for NetCDF that use the facilities of the enhanced data model (and which are commonly used) need to be developed.
- The NetCDF cell methods conventions provide a clean way of presenting the bounds for data aggregation. The bounds produced by the process described in this paper convolve a number of parameters, making the resulting time-bounds hard to trace back to the original parameters. Providing per-point collection boundaries within WaterML would harmonise the two methods.[15]

[15] This extension is already being considered as part of the WaterML standard.

Annex A
(informative)

Example NetCDF Encodings

An example category timeseries observation:

```
<om:OM_Observation gml:id="xsd-timeseries-observation.example">
 <gml:identifier codeSpace="http://www.example.com/ids">
  http://www.example.com/observations/EX1234
 </gml:identifier>
 <gml:description>Example timeseries for XML encoding
http://www.opengis.net/spec/waterml/2.0/req/xsd-timeseries-
observation</gml:description>

 <om:metadata>
     <wml2:ObservationMetadata gml:id="metadata">
         <gmd:contact xlink:href="http://www.example.com" xlink:title="Example Pty
Ltd"/>
         <gmd:dateStamp>
             <gco:DateTime>2011-11-21T12:30:00+10:00</gco:DateTime>
         </gmd:dateStamp>
         <gmd:identificationInfo
xlink:href="http://www.example.com/observations/example"
             xlink:title="Example Observation"/>
     </wml2:ObservationMetadata>
 </om:metadata>
 <om:phenomenonTime>
     <gml:TimePeriod gml:id="time-period.1">
         <gml:beginPosition>2011-11-21T12:26:00+10:00</gml:beginPosition>
         <gml:endPosition>2011-11-21T12:27:00+10:00</gml:endPosition>
     </gml:TimePeriod>
 </om:phenomenonTime>
 <om:resultTime>
     <gml:TimeInstant gml:id="time-instant.1">
         <gml:timePosition>2011-11-21T12:27:00+10:00</gml:timePosition>
     </gml:TimeInstant>
 </om:resultTime>
 <om:procedure>
     <wml2:ObservationProcess gml:id="observation-process.1">
         <wml2:processType

 xlink:href="http://www.opengis.net/def/processType/WaterML/2.0/Algorithm"
             xlink:title="Algorithmic Process"/>
         <wml2:processReference
xlink:href="http://www.example.com/algorithms/sample_quality/1.5"
             xlink:title="Sample Quality Evaluation 1.5"/>
     </wml2:ObservationProcess>
 </om:procedure>
 <om:observedProperty
xlink:href="http://sweet.jpl.nasa.gov/2.3/propOrdinal.owl#Quality"
     xlink:title="Quality"/>
 <om:featureOfInterest xlink:href="http://www.example.com/foi/LG1"
     xlink:title="Lake George 1"/>
 <om:result>
     <wml2:CategoricalTimeseries gml:id="timeseries.1">
         <wml2:temporalExtent>
             <gml:TimePeriod gml:id="time-period.2">
                 <gml:beginPosition>2011-11-21T12:26:00+10:00</gml:beginPosition>
                 <gml:endPosition>2011-11-21T12:27:00+10:00</gml:endPosition>
             </gml:TimePeriod>
         </wml2:temporalExtent>
         <wml2:defaultPointMetadata>
             <wml2:DefaultTVPCategoricalMetadata>
```

```
                <wml2:quality
                  xlink:href="http://www.opengis.net/WaterML/2.0/def/quality/good"
                  xlink:title="Good"
                />
                  <wml2:codeSpace>http://www.example.com/terms/2.3</wml2:codeSpace>
              </wml2:DefaultTVPCategoricalMetadata>
          </wml2:defaultPointMetadata>
          <wml2:point>
              <wml2:CategoricalTVP>
                  <wml2:time>2011-11-21T12:27:00+10:00</wml2:time>
                  <wml2:value>
                      <swe:Category>
                      <swe:value>Poor</swe:value>
                    </swe:Category>
                  </wml2:value>
              </wml2:CategoricalTVP>
          <wml2:point>
          </wml2:point>
              <wml2:CategoricalTVP>
                  <wml2:time>2011-11-21T12:27:30+10:00</wml2:time>
                  <wml2:value>
                      <swe:Category>
                      <swe:value>Biased</swe:value>
                    </swe:Category>
                  </wml2:value>
                <wml2:metadata>
                  <wml2:TVPMetadata>
                    <wml2:comment>Sensor drift detected</wml2:comment>
                  </wml2:TVPMetadata>
                </wml2:metadata>
              </wml2:CategoricalTVP>
          </wml2:point>
      </wml2:CategoricalTimeseries>
  </om:result>
</om:OM_Observation>
```

As a NetCDF data payload (CDL format):

```
Netcdf xsd_timeseries_observation_example {
dimensions:
  station = 1;
  time = unlimited;
  station_name = 32;
variables:
  // Global Metadata
  :Conventions = "CF-1.6/timeSeries CF-1.6/coordinates";
  :Metadata_Conventions = "Unidata Dataset Discovery v1.0"
  :title = "Example timeseries for XML encoding
     http://www.opengis.net/spec/waterml/2.0/req/xsd-timeseries-observation";
  :id = "http://www.example.com/observations/EX1234";
  :naming_authority = "http://www.example.com/ids";
  :cdm_data_type = "Station";
  :geospatial_lat_min = -35.154004; // Derived from dereferenced featureOfInterest
  :geospatial_lat_max = -35.154004;
  :geospatial_lat_units = "degrees_north";
  :geospatial_lon_min = 149.390391; // Derived from dereferenced featureOfInterest
  :geospatial_lon_max = 149.390391;
  :geospatial_lon_units = "degrees_east";
  :time_coverage_start = "2011-11-21T12:26:00+10:00";
  :time_coverage_end = "2011-11-21T12:27:00+10:00";
  :featureType = "timeSeries";
  :quality_dictionary =
    "good=http://www.opengis.net/WaterML/2.0/def/quality/good ...";
  :process_type_dictionary =
    "algorithmic= http://www.opengis.net/def/processType/WaterML/2.0/Algorithm ...";
  :sampled_feature = "Lake George";
  :sampled_feature_link = "http://www.example.com/foi/LG";
  :gauge_datum = "Australian height datum";
```

```
  :gauge_datum_link = "urn:ogc:def:crs:EPSG::5711";
  :monitoring_point_timezone = "+10:00";
  :process_type = "algorithmic";
  :process_type_vocabulary = "process_type_dictionary";
  :process_type_vocabulary_typeof = "simple";
  :process_reference = "Sample Quality Evaluation 1.5";
  :process_reference_link = "http://www.example.com/algorithms/sample_quality/1.5";

  // Geolocation variables
   double lat;
   lat:standard_name = "latitude";
   lat:units = "degrees_north";
   lat:long_name = "station latitude";
   lat:axis = "X";

   double long = 149.390391;
   long:standard_name = "longitude";
   long:units = "degrees_east";
   long:long_name = "station longitude";
   long:axis = "Y";

   // Timestamps
   int time(station, time);
   time:standard_name = "time";
   time:long_name = "sample time";
   time:units = "seconds since 2011-11-21 12:26:00 +10:00";
   time:calendar = "standard";
   time:axis = "T";

   // Monitoring point names
   char station_id(station, station_name);
   station_id:standard_name = "station_name";
   station_id:long_name = "Made up monitoring point";
   station_id:cf_role = "timeseries_id";

   // Timeseries values
   byte measure_quality(station, time);
   measure_quality:standard_name = "measure_quality";
   measure_quality:long_name = "measure quality";
   measure_quality:coordinates = "time lat long";
   measure_quality:ancillary_variables = "measure_quality_comment";
   measure_quality:_FillValue = 0b;
   measure_quality:missing_value = 0b;
   measure_quality:valid_range = 0b, 4b;
   measure_quality:flag_masks = 7b;
   measure_quality:flag_values = 1b, 2b, 3b, 4b;
   measure_quality:flag_meanings = "good poor biased invalid";
   measure_quality:flag_vocabulary = "http://www.example.com/terms/2.3";
   measure_quality:flag_vocabulary_typeof = "text/x-skos";
   measure_quality:timeseries_type = "interleaved";
   measure_quality:quality = "good";
   measure_quality:quality_vocabulary = "quality_dictionary";

   // Ancillary variable
   string measure_quality_comment(station, time);
   measure_quality_comment:standard_name = "measaure_quality comments";
   measure_quality_comment:long_name = "Additional comments for measure_quality";

data:
  lat = -35.154004; // Derived by de-referencing the featureOfInterest
  long = 149.390391;
  time = 0, 30;
  station_id = "Lake George 1";
  measure_quality = 2b, 3b;
  measure_quality_comment = "", "Sensor drift detected";
}
```

Note that this document, since it has a string-typed variable requires the enhanced data model. Removing the comments metadata makes it compatible with the classic data model.

If the encoding was represented as an observation/payload pair, then the resulting documents would be similar to the examples given above, with the following changes to the observation

```
<om:OM_Observation gml:id="xsd-timeseries-observation.example">
  ...
 <om:result
   xlink:title="Example timeseries"
   xlink:role="application/x-netcdf"
   xlink:href=" http://www.example.com/observations/EX1234.nc"
 />
</om:OM_Observation>
```

and to the NetCDF payload

```
Netcdf xsd_timeseries_observation_example {
...
  // Global Metadata
  :metadata_link = "http://www.example.com/observations/EX1234#metadata";
  :title = "Example timeseries";
...
}
```

Bibliography

Bigagli, Lorenzo, & Nativi, S. (2011). *NetCDF Uncertainty Conventions (NetCDF-U) 1.0.* Open Geospatial Consortium.

Carval, T., Keely, B., Takatsuki, Y., Yoshida, T., Stephen, L., Schmid, C., et al. (2011, June 14). *Argo Data Management User's Manual.* Retrieved January 31, 2012, from http://www.argodatamgt.org/Media/Argo-Data-Management/Argo-Documentation/General-documentation/Data-format/obsolete-documents/Argo-User-s-manual-version-2.3

Davis, E. (2005, September 28). *NetCDF Attribute Convention for Dataset Discovery*, 1.0. Retrieved February 6, 2012, from http://www.unidata.ucar.edu/software/netcdf-java/formats/DataDiscoveryAttConvention.html

Domenico, B. (2012). *OGC Network Common Data Form (NetCDF) Enhanced Data Model Extension Standard.* Open Geospatial Consortium.

Eaton, B., Gregory, J., Drach, B., Taylor, K., Hankin, S., Caron, J., et al. (2011, December 5). *NetCDF Climate and Forecast (CF) Metadata Conventions*, 1.6.5. Retrieved February 6, 2012, from CF Metadata: http://cf-pcmdi.llnl.gov/documents/cf-conventions/1.6/cf-conventions.pdf

Gregory, J., Drach, B., & Tett, S. (1999, March 14). *GDT NetCDF Conventions for Climate Data.* Retrieved February 16, 2012, from GDT netCDF conventions for climate data

Hankin, S., & Searight, K. (1995, April 24). *Conventions for the standardization of NetCDF files.* Retrieved February 17, 2012, from ftp://ftp.unidata.ucar.edu/pub/netcdf/Conventions/COARDS

INSPIRE Drafting Team "Data Specifications". (2010). *INSPIRE Guidelines for the Encoding of Spatial Data.* Infrastructure for Spatial Information in Europe.

Intergovermenal Committee on Surveying and Mapping. (2009). *Geocentric Datum of Australia Technical Manual, Version 2.3 (1).* Commonwealth of Australia.

International Association of Oil & Gas Producers. (2011, June). *Using the EPSG Geodetic Parameter Dataset.* Retrieved January 2012, from http://www.epsg.org/guides/docs/G7-1.pdf

Lefort, L. (2009, December). Review of semantic enablement techniques used in geospatial and semantic standards for legacy and opportunistic mashups. *Proceedings of the 5th Australasian Ontologies Workshop , CRPIT 112*.

National Center for Atmospheric Research, Research Aviation Facility. (2009, June 29). *NCAR-RAF netCDF Conventions.* Retrieved January 31, 2012, from http://www.eol.ucar.edu/raf/Software/netCDF.html

Natural Environment Research Council. (2011). *climate-science-modelling-language-v3-0*. Retrieved January 30, 2012, from http://csml.badc.rl.ac.uk/

NOAA GEO-IDE. (2011, August 18). *NetCDF Attribute Convention for Dataset Discovery.* Retrieved February 20, 2012, from GEO-IDE Guidelines and Best Practices Wiki: https://geo-ide.noaa.gov/wiki/index.php?title=NetCDF_Attribute_Convention_for_Dataset_Discovery

OPeNDAP. (2009, September 29). *OPeNDAP User Guide*. Retrieved February 6, 2012, from http://docs.opendap.org/index.php/UserGuide

Portele, C. (2007). *OpenGIS® Geography Markup Language (GML) Encoding Standard.* Open Geospatial Consortium.

Rew, R. (2010, May). *Adapting Software to NetCDF's Enhanced Data Model.* Retrieved February 13, 2012, from http://www.unidata.ucar.edu/presentations/Rew/rew-egu.pdf

Rew, R. (2011, May 10). *Update on Unidata Technologies for data Access.* Retrieved February 13, 2012, from http://www.unidata.ucar.edu/presentations/Rew/rew-2011-goessp.pdf

Rew, R., Davis, G., Emmerson, S., Davies, H., Harnett, E., Heimburger, et al. (2011, June). The NetCDF Users Guide. (Version 4.1.3). Unidata Program Center. Retrieved from http://www.unidata.ucar.edu/software/netcdf/docs/netcdf.pdf

Schadow, G., & McDonald, C. J. (2009, July 18). *The Unified Code for Units of Measure*, 1.8. Retrieved February 14, 2012, from http://unitsofmeasure.org/

Taylor, P. (2010). *OGC Water ML 2.0: Part 1 - Timeseries.* Open Geospatial Consortium.

The HDF Group. (2011, November). *HDF5 User's Guide*, 1.8.8. Retrieved February 13, 2012, from http://www.hdfgroup.org/HDF5/doc/PSandPDF/HDF5_UsersGuide.PDF

Unidata Program Center. (2008, June 20). *CDM Feature Types.* Retrieved February 14, 2012, from http://www.unidata.ucar.edu/software/netcdf-java/archive/CDMfeatures.doc

Unidata Program Center. (2008, March 17). *CDM Point Feature Types.* Retrieved February 16, 2012, from http://www.unidata.ucar.edu/software/netcdf-java/archive/CDMpoints.doc

Unidata Program Center. (n.d.). *THREDDS (Thematic Realtime Environmental Distributed Data Services)*. Retrieved February 6, 2012, from http://www.unidata.ucar.edu/projects/THREDDS/

Unidata Program Center. (2011). *UDUNITS 2.1.24 Manual.* Retrieved February 14, 2012, from http://www.unidata.ucar.edu/software/udunits/udunits-2/udunits2.html

Whiteside, A. (2007). *Definition identifier URNs in OGC namespace.* Open Geospatial Consortium.

Woolf, A. (2011). *Climate Science Modelling Language (CSML): Sampling Coverage Observations for the Met/Ocean Domain.* Open Geospatial Consortium.

