

Open Geospatial Consortium Inc.

Date: 2006-03-25

Reference number of this document: 06-022r1

Version: 0.0.9

Category: OpenGIS[®] Discussion Paper

Editor: **James Resler**

Temporal recommendations

Copyright © 2006 Open Geospatial Consortium, Inc. All Rights Reserved.
To obtain additional rights of use, visit <http://www.opengeospatial.org/legal/>.

Warning

This proposed document is not an OGC Standard. It is distributed for review and comment. It is subject to change without notice and may not be referred to as an OGC Standard.

Document type:	OpenGIS [®] Discussion Paper
Document subtype:	Not an official OGC position
Document stage:	Proposed
Document language:	English

Contents	Page
i. Preface.....	iii
ii. Document terms and definitions	iii
iii. Document contributor contact points.....	iii
iv. Revision history	iii
v. Changes to OGC Specifications.....	iv
Foreword.....	v
Introduction.....	vi
1 References.....	1
2 Recommended Changes to the Web Feature Service Implementation and Filter Encoding Specifications.....	1
2.1 WFS recommendation 1 – Filter Encoding.....	2
2.2 WFS Recommendation 2 – Feature Instance and History	2
2.3 WFS Recommendation 3 – History Transaction.....	3
2.4 WFS Recommendation 4 – Temporal Extent.....	3
2.5 WFS Recommendation 5 – Temporal Key Value Pair (KVP).....	4
3 Web Maps Service Recommendation – Temporal Map Layer Metadata.....	4
4 Catalogue Service – Web (CSW)	5
4.1 CSW-Recommendation 1 - Harvest Temporal WFS Information	5
4.2 CSW-Recommendation 2 - Data Elements	6
5 OWS Common Recommendation 1 - Time Zone Offset Request.....	6

i. Preface

This document summarizes recommendations for extending geospatial standards with regard to time-varying information. These proposals are the result of the National Technology Alliance program called Temporal Evaluation and Assessment (TEA). The recommendations made are based upon the findings from the first two demonstrations of TEA. OGC has encouraged TEA to submit the results of the standard's evaluation to the revision working groups for the various standards. These recommendations encompass several OGC specifications, WFS, WMS, CSW and OWS Common, but are contained within a single discussion paper for continuity.

ii. Document terms and definitions

This document uses the specification terms defined in Subclause 5.3 of [OGC 05-008]. In particular, the word “shall” (not “must”) is the verb form used to indicate a requirement to be strictly followed to conform to this specification.

iii. Document contributor contact points

All questions regarding this document should be directed to the editor or the contributors:

Name	Organization
Jim Ressler	Northrup Grumman

iv. Revision history

Date	Release	Editor	Primary clauses modified	Description
06-03-25	0.0.9	Carl Reed	Many	Put in proper template format.

v. Changes to OGC Specifications

This document recommends a number of potential changes to WFS, WCS, and so forth in order that these interface definitions can better handle temporal content.

Foreword

This document provides the details for recommended changes to existing OpenGIS Implementation Specification to better support temporal requirements.

- a) Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. The OGC shall not be held responsible for identifying any or

Introduction

In May 2005, Northrop Grumman began work on the TEA program for the National Technology Alliance. Working in conjunction with pertinent standards bodies (ISO, OGC, INCITS, and W3C) TEA has tested and demonstrated ways to manage and use time-varying geospatial information. TEA has enabled experimentation and collaboration with various government departments and industry to achieve the following.

- Evaluate and test standards for capabilities and shortcomings.
- Define data schemas and databases with temporal capabilities.
- Develop a compliance engine that extends to temporal services.
- Identify technology gaps and potential solutions.
- Validate requirements for temporal capabilities and increase stakeholder participation from the intelligence community.
- Deliver a persistent unclassified environment for evaluation, integration, and testing of standards, interfaces, software, and data.

This document is one of the recommendations to standards changes resulting from research on TEA.

After reviewing the changes to the OGC Filter Specification (05-093), Northrop Grumman TASC is submitting the following revisions. This is based upon work we are performing for the National Technology Alliance in studying temporal geospatial standards and interfaces (TEA project). TASC is supportive of modifying the filter specification for temporal and appreciates the change submittal. Please be aware that OGC is a stakeholder in the TEA project and we are also looking at recommendations to the OGC common specification to address temporal interfaces and data.

These modifications specified other references to the temporal operators and clarified the use of time zones in temporal data.

1 References

- [1] **OGC 04-095 05-093 Filter Encoding Implementation Specification change proposal 14 Oct 2005**
- [2] **TR-048-101705-202 TEA Demo 1 Final Report 11 Nov 2005**
- [3] **TR-048-101605-219 TEA Demo 2 Final Report 20 Jan 2006**
- [4] **OGC-04-094 Web Feature Service Implementation Specification 1.1.0 3 May 2005**
- [5] **OGC 04-095 Filter Encoding Implementation Specification 1.1.0 3 May 2005**
- [6] **OGC 04-024 Web Map Service Specification 1.3.8 Aug 2004**
- [7] **OGC 04-021r2 Catalogue Services Specification 2.0 5 May 2004**
- [8] **OGC 03-065r6 Web Coverage Service 1.0.0 27 Aug 2003**
- [9] **OGC 05-008r1 OWS Common Specification 23 May 2005**

2 Recommended Changes to the Web Feature Service Implementation and Filter Encoding Specifications

These recommendations are based upon the addition of metadata for features with temporal information. Three types of temporal features were defined by the TEA research and are summarized below.

1. Feature instance – a single instance of features (usually the most current)
 - i. Currency timestamp – the date and time at which this feature became effective
2. Feature history – a succession of revisions to a feature over time, to include past and future revisions.
 - i. Revision – a unique sequential identifier (integer value) that defines the successive revision of features.
 - ii. Valid time - real world time the change occurred (valid start time) and the change is no longer in effect (valid end time)
 - iii. Transaction time - when the database was updated to reflect the change (transaction start time) and when the change is no longer in effect (transaction end time)
3. Feature observation – a time series of observed values from an in-situ feature sensor.
 - i. Observation type – an enumerated name of the type of report (example, temperature).
 - ii. Time of observation
 - iii. Value of observation

4. Moving Feature – a feature that changes location over time. The trajectory of a single moving feature contains.
 - i. Time
 - ii. Location
 - iii. Other attributes as defined in ISO 19141

The applicability of the WFS recommendations to the types of temporal features is shown in the table below.

Type	R1	R2	R3	R4	R5
Instance	√	√		√	√
History	√	√	√	√	√
Observation	√			√	
Moving Feature	√			√	

The research on TEA has produced recommendations for feature instance and history. The observations and moving features have been prototyped and apply to some of the recommendations, but the implementation of these feature types are the subject of future recommendations.

2.1 WFS recommendation 1 – Filter Encoding

Change request 05-093 was submitted to the OGC WFS RWG by ORNL in October 2005. NG TASC supports the change request with modification. The modifications specify other references to the temporal operators and clarify the use of time zones in temporal data. The modifications recommended to 05-093 have been submitted to the OGC as document number 05-093r1.

2.2 WFS Recommendation 2 – Feature Instance and History

Standard fields for representing feature instance (currency timestamp) and feature history (revision, valid time and transaction time) should be added as an optional implementation of the feature metadata. When applications implement temporal features, these standard fields can be referenced by applications using WFS in filtering and status reporting of features. This recommendation will be made in concert with the ISO Geographic information metadata specification, ISO 19115.

The intent of this change is to add metadata to the capabilities indicating that a featureType has standard temporal properties, as defined by ISO 19115. For instance, it should indicate that featureType A contains “feature instances” and has a property which represents the currency timestamp. The metadata referenced are elements of a feature type, such as contained in Table 6 of the WFS 1.1 specification. Just as

LatLongBoundingBox and SRS are standard elements of a featureType, we proposed that temporal extent, temporal reference system and temporal type become standard elements. Some metadata elements contained in ISO 19115 (see appendix B.1), such as metadata items dataExt and RefSys, can be used to define temporal metadata in the WFS. The discovery of the temporal information would be accomplished through recommendation WFS-R4.

2.3 WFS Recommendation 3 – History Transaction

A WFS transaction for storing feature history is needed. The current WFS transaction is capable of only updating a single instance of a feature. The feature history transaction prototyped by TEA stores the revision number, valid time and transaction time for each revision to a feature. A web feature transaction to update features history for insert, modify and withdraw (remove) operations is recommended. The proposal 05-039 for fine-grained object updates may be applied to this type of transaction. A comment from Galdos on this recommendation:

“We agree that there is a need for Update to be able to change a single TimeSlice and NOT the entire feature. We are working on a proposal for this under the fine grained object update change proposal. Dynamic features (with histories) are a primary use case for this work. In the case of the SnapShot model there are no TimeSlices organized by a history property. Each SnapShot of the feature has a time associated with it. This likely requires changes to WFS – so that one can update the different SnapShots. I think your proposal seems more related to this case than the history case.”

2.4 WFS Recommendation 4 – Temporal Extent

In order for clients to determine if a feature type has temporal information (feature instance and feature history), WFS should inform the client of the temporal data within the layer, similar to the WMS. The feature capabilities document needs to contain the temporal extent of the feature data within the layer. If the feature type has temporal attributes, the WFS GetCapabilities operation should return the temporal extent (start, end, frequency) of the feature type similar to the way time is incorporated into the bounding box of WMS GetCapabilities.

One method of implementing this recommendation is to have a new interface called GetTemporalExtent to complement GetCapabilities. There would be two components to the interface: GetCapabilities has an element that indicates the temporal type (instance, history, observation, moving feature) of each feature type; GetTemporalExtent returns the temporal extent for feature types as selected by filters on feature type and bounding box. For example, this interface can be used to know a navigation aids feature type has features surrounding the Chicago area with history from 2001-01-01 to 2005-12-31. An alternative to GetTemporalExtent implementation would use GetFeature with a value for the resultType parameter called temporal.

2.5 WFS Recommendation 5 – Temporal Key Value Pair (KVP)

For feature instance and feature history layers, a recommendation is made to implement a time key name/value pair (KVP) to account for the various times associated with features in a HTTP GetFeature request, similar to the TIME parameter in the WMS specification for GetMap. This would enable specific temporal feature queries using a key name/value pair without using the XML filter. The TIME parameter would query feature instance layers on the currency timestamp. The TIME parameter would query feature history layers on the valid start timestamp. For example, to query a point feature within a range of time would be a much simpler with this interface than building a XML filter.

For example, if the most recent feature within a time interval was needed, the following request could be implemented with a TIME parameter.

```
GetFeature typename="XYZ" TIME="start time/end time"
```

This would return a feature if current within the time range; for a feature history, it would return all features within the period.

If the feature had a specific instance date, this could be done with TIME (a temporary feature would be a good example).

```
GetFeature typename="XYZ" TIME="start date"
```

Note: It may be preferable to have a different TIME parameter for instances and history, such as TIMEINSTANCE, TIMERANGE and TIMESLICE.

These examples illustrate the simplicity of temporal feature queries vs. the longer filter query, which would still be used for more complex queries.

If the temporal KVP recommendation has use outside of WFS, then this recommendation should be implemented in OWS Common specification.

3 Web Maps Service Recommendation – Temporal Map Layer Metadata

The web map service has an optional TIME attribute associated with each image served and a temporal extent for each layer in the capability document. This is useful information for temporal analysis, but only provides a range of time and frequency for which maps (images) exist. Clients that need to browse for specific images have to perform a GetMap request for each time of interest, which is a cumbersome and inefficient process. The GetCapabilities or another operation should be capable of returning a list of images for a specified layer with metadata for each map, and optionally a thumbnail of the map, for all maps within a given time range.

The time information can be implemented with a new interface such as GetTemporalMaps or modifying an existing interface. The WMS interface has a mechanism to provide this information if it were extended. Using properties of the temporal extent (called dimension in WMS 1.3), the Capabilities document can provide a

list of times with maps. The “time” extent (dimension) in WMS could return multiple values instead of a period.

A portion of GetCapabilities would currently return this metadata:

```
<Dimension name="time" nearestValue="0" multipleValues="0"
current="0">2002-12-01/2005-09-01</Extent>
```

A temporal extension of WMS could return this metadata:

```
<Dimension name="time" nearestValue="0" multipleValues="1"
current="0">2002-12-01,2002-12-02, 2002-12-03, 2002-12-06, ....
....
2005,08,31,2005-09-01</Extent>
```

Note the gap in coverage between 2002-12-03 and 2002-12-06. This data is not available in the current WMS GetCapabilities.

It is permissible to specify the specific times supported by listing multiple values in the extent string (but a specific request for this format is needed). However, a better structured format for returning the values would be helpful, since the string can get unmanageable if you support a large number of images collected at irregular times, so structuring the Dimension element or adding a GetTemporalExtent operation would be in order.

This enables the WMS interface (using GetTemporalMaps or GetCapabilities) to provide a list of times with maps. With this information, the client could perform a GetMap to retrieve the map on a specific date. A thumbnail view of maps could be implemented with GetTemporalMaps and GetMap with a new FORMAT. For example, FORMAT=text/html might create an HTML page showing thumbnails, or FORMAT=mpeg might create a mpeg video with each frame representing a different time.

The capability desired in WMS is analogous to WCS which has a TemporalDomain to describe the valid time constraints for GetCoverage requests (WCS 1.0, section 8.3.2.3). In WCS GetCoverage, the TemporalDomain can contain a sequence of time instants and/or time periods. A harmonization between WMS and WCS could define the same interface for temporal extent.

4 Catalogue Service – Web (CSW)

4.1 CSW-Recommendation 1 - Harvest Temporal WFS Information

The catalogue data model needs an extension to support time in harvesting the WFS capabilities document to store the temporal extent of each layer in the catalogue registry.

4.2 CSW-Recommendation 2 - Data Elements

The Describe Record operation does not provide all the hierarchical data elements, including named slots, which makes it difficult to determine which records have temporal data, when to extend the ebRIM model to add data elements, and how to build queries on these records. Therefore, the Describe Record operation should have the option to return all data elements for a given record type. The result of this recommendation will ensure this limitation is not unique to a particular Web Registry Service (WRS) implementation.

5 OWS Common Recommendation 1 - Time Zone Offset Request

The Time Zone for temporal data is not standard in data sources but knowledge of the time zone is necessary for consistent queries and unambiguous results, especially queries that encompass data from multiple time zones or filter by the hour, minute or second. Web services should have a default time zone of UTC as defined by ISO 8601 (or Greenwich Mean Time, also referred to as “Z” for Zulu Time) for all temporal data passed or returned. ISO 8601 accounts for local time by specifying an offset to UTC. Web services should allow the client to request a time zone offset to be applied to all temporal data through a parameter passed on operations. For example, if the client requested all data referenced in U.S. eastern time zone, a time zone offset of -0500 (five hours) from UTC would be applied to all temporal fields. The web service must be able to convert and return temporal data values from the time zone of the data to the specified time offset given in the request.