Open Geospatial Consortium, Inc.

Date: 2010-09-08

Reference number of this document: OGC 10-035r2

Category: Public Engineering Report

Editors: David Rosinger, Stan Tillman

# OGC® OWS-7 Information Sharing Engineering Report



**Warning**

This document is not an OGC Standard. This document is an OGC Public Engineering Report created as a deliverable in an OGC Interoperability Initiative and is 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, any OGC Engineering Report should not be referenced as required or mandatory technology in procurements.

| | |
|---|---|
| Document type: | OpenGIS® Engineering Report |
| Document subtype: | NA |
| Document stage: | Approved for public release |
| Document language: | English |

## Preface

Suggested additions, changes, and comments on this draft report are welcome and encouraged. Such suggestions may be submitted by email message or by making suggested changes in an edited copy of this document.

The changes made in this document version, relative to the previous version, are tracked by Microsoft Word, and can be viewed if desired. If you choose to submit suggested changes by editing this document, please first accept all the current changes, and then make your suggested changes with change tracking on.

# Contents

Page

# Figures

Page



# Tables

Page

# OGC® OWS-7 Information Sharing Engineering Report

## 1 Introduction

### 1.1 Scope

This Engineering Report describes an investigation and evaluation of various methods of sharing information within a collaborative environment accomplished during the OGC Web Services Testbed, Phase 7 (OWS-7). The intent of the OWS-7 Information Sharing activity was to move toward a standardized method of sharing geospatial data between Integrated Clients and potentially catalogs. This report reviews past OGC work within this area, makes recommendations based on the best parts of previous collaboration techniques, and provides recommendations for encoding documents for use in information sharing.

### 1.2 Document contributor contact points

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


| Name | Organization |
|---|---|
| David Rosinger | Intergraph |
| Stan Tillman | Intergraph |
| Jim Groffen | Lisasoft |
| Raj Singh | OGC |


### 1.3 Revision history

| Date | Release | Editor | Primary clauses modified | Description |
|---|---|---|---|---|
| 03/05/2010 | Draft | Stan Tillman | | Draft Release. |
| 06/08/2010 | Initial release | David Rosinger | All | Version released prior to Silver Spring TC. Extensive reorganization and new content added. |
| 06/24/2010 | Revision 1 | David Rosinger | Clauses 3. 6 and 7, Annex B | Numerous additions and improvements. |
| 08/12/2010 | Revision 2 | David Rosinger | Clauses 2, 3.2, 3.4, 4.1, 5.3, 5.6.1, 6.3.3 | Incorporate final reviewer feedback; only minor changes to the listed clauses. |
| 8/17/10 | 2 | Carl Reed | Various | Prepare for publication |

 

### 1.4 Future work

Improvements in this document are desirable to address open issues; to correct errors or enhance existing document content.

Ultimately, the goal of this study is to move the best of collaboration formats toward a formal standard. Therefore, any future work to this document should advance the ability to share data of all kinds among Integrated Clients.

### 1.5 Forward

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. The Open Geospatial Consortium Inc. shall not be held responsible for identifying any or all such patent rights.

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 standard set forth in this document, and to provide supporting documentation.

## 2 References

The following documents are referenced in this document. For dated references, subsequent amendments to, or revisions of, any of these publications do not apply. For undated references, the latest edition of the normative document referred to applies.

OGC 06-121r3, *OpenGIS® Web Services Common Standard*

NOTE This OWS Common Specification contains a list of normative references that are also applicable to this Implementation Specification.

OGC 05-005, *OpenGIS Web Map Context Implementation Specification*

OGC 08-050, *OpenGIS Web Map Context Documents Corrigendum 1 (1.1.0)*

OGC 05-062, *OGC Web Services Context Interoperability Experiment Final Report*

Request for Comments (RFC) 4287, *The Atom Syndication Format ("Atom 1.0"), December 2005, http://www.ietf.org/rfc/rfc4287.txt*

XML Base, *XML Base (Second Addition), W3C Recommendation 28 January 2009, http://www.w3.org/TR/xmlbase/*

RFC 3986, *Uniform Resource Identifier (URI): Generic Syntax, January 2005, http://www.ietf.org/rfc/rfc3986.txt*

GeoRSS website, *a geographic extension for RSS, http://georss.org/*

OGC 07-124r2, *OpenGIS® OWS-5 KML Engineering Report*

 

OGC 07-147r2, *OGC KML, version 2.2.0*

In addition to this document, this report includes several XML Schema Document files as specified in Annex A.

# 3 Terms and definitions

For the purposes of this report, the definitions specified in Clause 4 of the OWS Common Implementation Specification [OGC 06-121r3] shall apply. In addition, the following terms and definitions apply.

**3.1**
**application state**
the state of an Integrated Client including its connections to data and services, the bounding box of the map view and its legend configuration

**3.2**
**common operational picture**
a single identical display of relevant information shared by more than one command; a common operational picture facilitates collaborative planning and assists all echelons to achieve situational awareness (source: DoD)

**3.3**
**context document**
a file used to record application state

**3.4**
**integrated client**
a geospatial software application that presents a legend and map view that is capable of bringing together data from a variety of sources

**3.5**
**layer**
a dataset offered by a WMS, but more generally may refer to any entry depicted in the legend of an Integrated Client

**3.6**
**resource**
a dataset such as a WMS Layer or WFS Feature Type or a media type such as a document, image or video

# 4 Conventions

## 4.1 Abbreviated terms

Some more frequently used abbreviated terms:

| | |
|---|---|
| ADSD | Authoritative Data Source Directory |
| COP | Common Operational Picture |
| ER | Engineering Report |
| FDF | Feature Decision Fusion |
| GIS | Geographic Information System |
| HTTP | Hypertext Transfer Protocol |
| LOC | Location Organizer Folder |
| KML | Keyhole Markup Language |
| KVP | Key-Value Pair |
| OGC | Open Geospatial Consortium |
| OWC | OWS Context |
| OWS | OGC Web Service |
| SDI | Spatial Data Infrastructure |
| SOA | Service-Oriented Architecture |
| SOS | Sensor Observation Service |
| SWG | Standards Working Group |
| WCS | Web Coverage Service |
| WFS | Web Feature Service |
| WMS | Web Map Service |
| WPS | Web Processing Service |
| XML | Extensible Markup Language |

### 4.2 UML notation

Some diagrams that appear in this standard are presented using the Unified Modeling Language (UML) static structure diagram, as described in Subclause 5.2 of [OGC 06-121r3].

### 4.3 Used parts of other documents

This document uses significant parts of document OGC 07-124r2 *OpenGIS® OWS-5 KML Engineering Report* and the Atom specification. To reduce the need to refer to those documents, this document copies some of those parts with small modifications. To indicate those parts to readers of this document, the largely copied parts are shown with a light grey background (15%).

# 5 OWS-7 Information Sharing overview

This clause presents an overview of the Information Sharing subthread activity in OWS-7. The Information Sharing subthread was an activity within the OWS-7 Feature and Decision Fusion (FDF) thread.

## 5.1 Background

This OWS-7 engineering report summarizes the results of the investigation and evaluation of approaches for sharing data defining a common operational picture (COP) between multiple users. Specifically, this activity within OWS-7 involved the evaluation and comparison of the capabilities developed within:

- The Web Map Context Document (OGC 05-005, OGC 08-050),
- The OGC Web Services Context Interoperability Experiment Final Report (OGC 05-062),
- The use of KML, and
- The OGC Geospatial Fusion Services Testbed (2000-2001) concept for a Location Organizer Folder, an XML document created to share the results of a work session with anyone else.

These approaches were studied and prototyped to determine the viability of merging these concepts for the efficient sharing of resources that collectively define a common operational picture.

## 5.2 Resource types

Conceptually, a *common operational picture* is defined by bringing together a collection of *resources*. As a modeling exercise, the OWS-7 Information Sharing subthread first classified resources into two broad categories:

- OGC Datasets, and
- Media Types.

Media Types were further classified as:

- OGC Media Types, and
- Non-OGC Media Types.

### 5.2.1 OGC datasets

Past initiatives on information sharing have focused on the sharing of OGC datasets. Representative types of OGC datasets *(and the services that operate on them)* are:

- Layer *(Web Map Service (WMS))*
- Feature Type *(Web Feature Service (WFS))*
- Coverage *(Web Coverage Service (WCS))*
- Sensor Observation *(Sensor Observation Service (SOS))*

NOTE In order for a client to retrieve these datasets, *service metadata* must also be included as part of the resource. See subclause 6.5.1.

### 5.2.2 OGC media types

Past information sharing initiatives have also considering the sharing of OGC media types. Representative OGC media types are:

- Geographic Markup Language (GML) documents
- KML documents
- Geo-Referenced Images (e.g., Geo-TIFF)

NOTE An important use case for GML and KML is to represent imagery annotations (e.g., Points, Lines, Rectangles, Polygons and Text – geo-referenced and in paper-space).

### 5.2.3 Non-OGC media types

The OWS-7 Information Sharing subthread also considered the potential for including *non*-OGC media types ("unstructured data") in a common operational picture. Such media types are not traditionally considered when thinking of geospatial resources. But it is very common for an information source to contain geospatial information – either directly via geospatial coordinates, but more frequently via the inclusion of one or more place names (from which coordinates can be derived by geo-coding). And by including common media types, the potential content that can be included in a common operational picture is unbounded.

Representative non-OGC media types *(and common formats)* are:

- Document/Text *(Plain Text, HTML, XML, Word Processing Document, Presentation Document, Spreadsheet)*
- Photo/Still Image *(PNG, JPEG, TIFF)*
- Sound/Audio *(WAV, MP3, AAC)*
- Video/Moving Image *(WMV, AVI, MPEG)*

### 5.2.4 Resource UML class diagram

Figure 1 contains a class diagram that presents a visual representation of the OWS Context resource model developed in OWS-7.

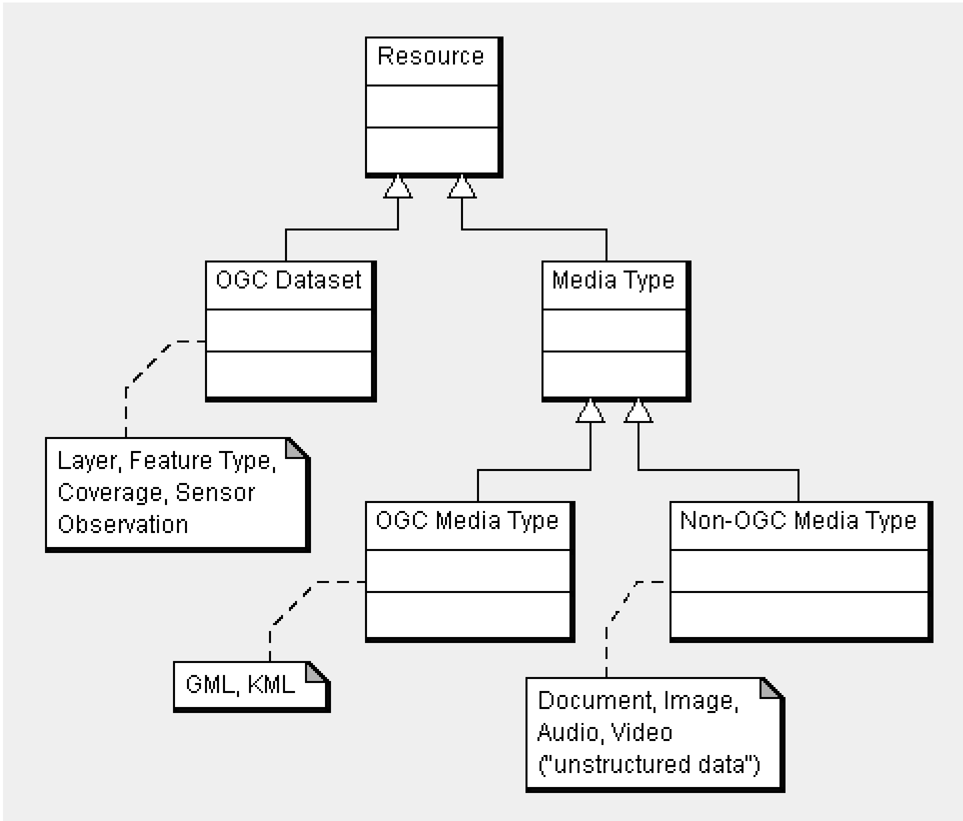


**Figure 1 — Resource UML class diagram**

### 5.3 Purpose — Capture common GIS application state

The principle use case for an OWS Context document is for defining the *application state* of an Integrated Client. When this application state is reproduced by two or more Integrated Clients, such clients are said to share a *common operational picture*.

The application state possesses these characteristics:

– A geospatial extent that is associated with the map window.

– A legend which contains a collection of legend entries.

– Each legend entry represents a resource.

– The sequence of legend entries in the legend is generally considered significant in that this commonly determines the z-order (visual stacking order) of the legend entry graphics (layers) in the map window. (And also note that the order of resources in a context document typically influences the order in which those resources are visually depicted in the legend of an Integrated Client.)

– Each legend entry may have associated symbology that is used to portray its associated layer in the map view.

 

– One legend entry may be designated as representing the currently selected (active) layer. †

– A legend entry may be designated as hidden in the legend and/or its associated layer may be designated as hidden in the map view. †

NOTE † indicates application state characteristics that were not captured in the OWS Context model in OWS-7.

### 5.4 Overview of historical efforts for context sharing

This clause presents an overview of approaches explored in the past for information sharing and collaboration.

#### 5.4.1 WMS Context Document

The OGC baseline currently contains a standard that defines a document called a *Web Map Context Document*. This standard defines an XML structure for describing *"…information about the server(s) providing layer(s) in the overall map, the bounding box and map projection shared by all the maps, sufficient operational metadata for Client software to reproduce the map, and ancillary metadata used to annotate or describe the maps and their provenance for the benefit of human viewers."* (OGC 05-005, *Web Map Context Document*, p viii.)

In OGC 05-005, a layer is defined to represent *only* a layer served by an OGC *Web Map Service (WMS)*. However, the capability to record a list of layers for persistence across sessions or for collaboration with other clients was quickly realized as a major benefit.

#### 5.4.2 OWS Context Document

Following the success of the Web Map Context document, some began discussing the possibility of extending the context document to include resources served by additional OGC services. Thus was born the concept of the *OWS Context Document*.

The work in OWS-7 on exploring the OWS Context Document follows previous efforts. OGC 05-062, *OGC Web Services Context Interoperability Experiment Final Report* summarizes an early OGC-sponsored activity. On April 4th, 2007, the Context RWG was officially formed to extend the WMS Context standard to include resources served by additional types of services such as the *OGC Web Feature Service (WFS)* and *Web Coverage Service (WCS)*.

These previous activities have produced a schema for an OWS Context document whose version prior to OWS-7 stood at 0.3.1. The OWS-7 Information Sharing subthread continued work on those schemas and the new schema produced in that testbed is versioned 0.7.1.

#### 5.4.3 Keyhole Markup Language (KML)

KML is an XML based schema that was invented primarily to serve as an annotation language for Google Earth, and support for KML was later incorporated into Google

 

Maps. In 2007, Google agreed to submit KML to OGC for international standardization; the current version of KML is 2.2.

KML combines geometry and styling which has made it ideally suited to its principle role as an annotation language. Some though have suggested that KML might also be applied more broadly to the problem of capturing the application state of an Integrated Client.

#### 5.4.4 A Debate – OWS Context vs. KML

At the first meeting of the OWS Context RWG the question was asked, “Why do we need OWS Context when we have KML?” Thus began the OWS Context verses KML debate.

During the OWS-5 testbed, an Engineering Report was written to document a study about KML. A portion of this document was devoted to a comparison and evaluation of KML and OWS Context. The following is an excerpt from this document (07-124r2 – page 18) that provides an argument for bringing best of both formats together:

*Finally, similar to the recent OWS Context experiments that include GML data inline, KML has always been able to include geographic content and style it. So due to all these similarities between the goals, if not the syntax, of these two XML file formats, it made sense to evaluate merging the two to create a single OGC standard for describing a map view that can consist of a mixture of inline geographic content, as well as content brought in from external services.*

But what might be a sensible way of bringing these formats together? On page 9, the authors summarize the outcome of the debate:

*One topic of discussion was defining a services' module, which would add grammar to KML to describe bindings to OGC web services like WMS, WFS, WCS as well as things like GML and GeoRSS. After much debate and discussion, the group decided that this would be duplicative work, as the OWS Context document serves the same purpose. Therefore, to add this feature to KML might increase the complexity of KML for people who don't work with OGC web services, and not add much value to those who do. Our work in this area instead concentrated on collaborating with the OWS Context working group to help define the integration of KML into OWS Context.*

In summary, the participants in OWS-5 concluded that it didn't make sense to extend (complicate) a mass-market format like KML so that it could be used to represent application state. (In other words, this would take KML in a direction beyond its intended purpose.) Instead, the OWS-5 participants concluded that somehow “integrating” KML into an OWS Context document was a more sensible approach.

#### 5.4.5 KML integration with OWS Context

On page 19 of 07-124r2, the authors detail an approach for integrating KML into an OWS Context document:

*The Agile Geography group concluded that the best way to accomplish the integration of OWS Context and KML is not to add open web service bindings to KML, but to allow the* *inclusion of KML in OWS Context documents. This is a nice improvement for Context, since its use is primarily visualization, and KML is much more visually oriented than GML. Imposing SLD as the only way to style in-line GML is a high barrier, while KML has a lighter weight way of including a style alongside a feature. In practice, this simply means allowing a <kml:Document> tag in the <Layer>.*

...

*One should also be able to reference a remote KML file sitting on server, just like one can with GML. This would be a layer, looking something like:*

```
<Layer hidden="0">
  <Server service="urn:ogc:serviceType:KML" version="2.2"
      title="KML Tasmania 2.2">
    <OnlineResource xlink:href="kml/topp_tas_ows5.kml"/>
  </Server>
</Layer>
```

#### 5.4.6 Location Organizer Folders (LOFs)

In 2000, OGC conducted the Geospatial Fusion Services Testbed which explored a variety of topics including the Geoparser, Geocoder, and Geolinker. However, a concept called a Location Organizer Folder was introduced that might be used in conjunction with Location Organizer Client applications as shown in Figure 2.

Within this concept the Location Organizer Client applications (LOCs) are used by analysts to compile related sets of spatio-temporal information from multi-source information for any intelligence problems. This identified information is captured in Location Organizer Folders (LOFs) which are geo-organized, geo-connected collections of information that analysts use to conduct spatio-temporal analyses. Using this framework, cooperating analysts can use the LOCs to discover, access, register, correlate and analyze information, and then store and share the resulting LOFs.

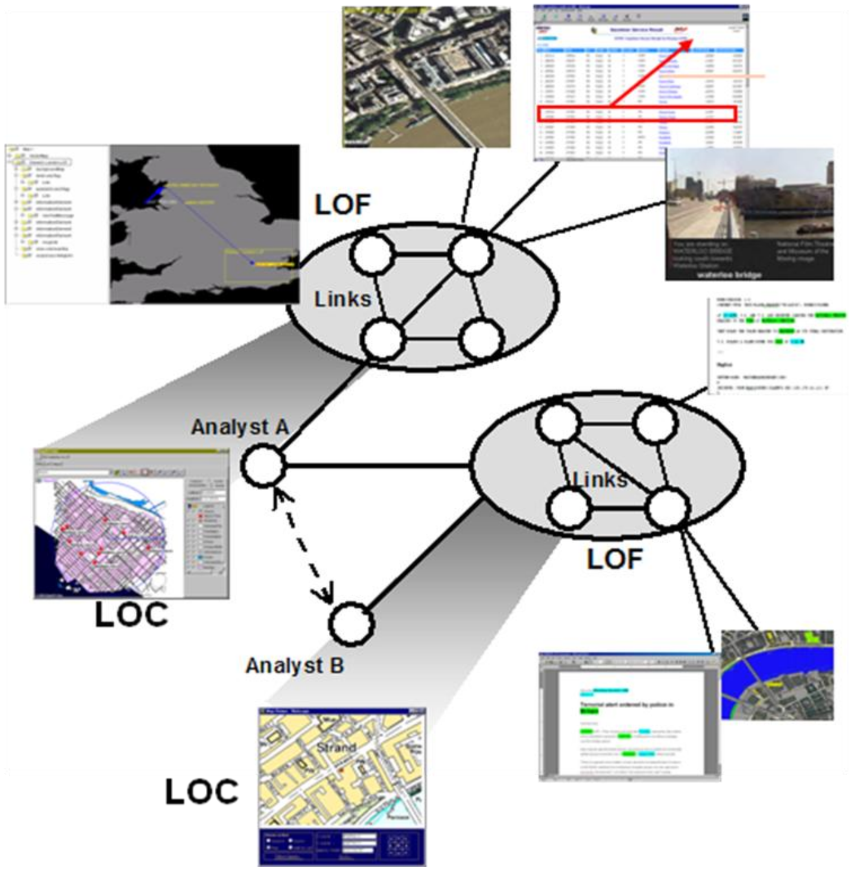


**Figure 2 — Location organizer clients operating on location organizer folders**

### 5.5 Goals of the test bed

The principle goal in the OWS-7 Information Sharing subthread was to push hard for an information sharing solution that would address the shortcomings of previous approaches. The goals also included meeting the requirements of the OWS-7 sponsors.

The OWS-7 information sharing activities focused on OWS Context. Here is a summary of the goals the Information Sharing subthread set for the OWS Context Document proposal that would emerge from the testbed:

- Incorporate the best aspects of WMS Context, OWS Context, KML and LOF into a new OWS Context proposal that would expedite the adoption of a normative 1.0 version of an OWS Context specification.

- Emphasize ease of implementation to encourage adoption of the standard.

- “Atomicity” – clients should not need more information than what is in the OWS Context document to make use of the document. Minimize the need to communicate with referenced servers. This is especially important for the initial application context presented.

- Incorporate existing formats and protocols where possible including Atom, RSS, OpenSearch, HTTP, etc.

- Meet the OWS-7 sponsor use cases for incorporating multimedia (media types) and search results into OWS Context documents.

### 5.6 OWS-7 proposal – explore Atom as an encoding for OWS Context documents

Early in the testbed, the Information Sharing subthread started its work by revising the OWS Context version 0.3.1 schema [1]. But before significant software development activities were underway, a proposal emerged for using the Atom Syndication Format as the encoding for the OWS Context document.

NOTE The Atom Syndication Format is also known as “Atom 1.0” or just “Atom”. For conciseness, this document uses the simple term “Atom” in the material that follows.

When the idea for using Atom first emerged, the Information Sharing subthread created a table that maps the elements and attributes from the OWS Context 0.3.1 schema to Atom. That table is reproduced in Annex B.

#### 5.6.1 Atom pros

Leading up to the decision to choose Atom as an alternative encoding for an OWS Context document, the pros and cons of using Atom for this purpose were weighed. Here are the pros:

- Atom is a mass market format with extensive support. (For instance, in popular web browsers and in feed readers.) Extending Atom increases the likelihood that OWS Context documents stored on the Web would be crawled and indexed by mass market search engines, exposing a valuable geo-spatial metadata resource on the Web.

- The OWS-7 Information Sharing subthread determined that most of the information content that was expected to be handled in the OWS Context document that emerged from OWS-7 had a placeholder in the core elements defined by the Atom specification.

- Atom defines a flexible *extension mechanism* so that elements from other namespaces can be added to an Atom document.

  NOTE The Atom namespace is http://www.w3.org/2005/Atom.

- Atom defines a mechanism for associating media with an Atom document. Media can be embedded directly into an Atom document, or associated with an Atom document via a URI (or, more generally, an IRI).

 

- Work is underway in the Catalog 3.0 SWG to add an OpenSearch interface to OGC catalogs. Atom is a popular format for the results returned by an OpenSearch query. (OpenSearch was also explored in the OWS-7 ADSD subthread.)

### 5.6.2 Atom cons

The Information Sharing subthread also recognized these aspects of Atom that might limit the ability of Atom to effectively represent an OWS Context document:

- Nesting support in Atom is poor. The elegance of nested resources could be lost.
- The Atom schema is informative, not normative; the Atom schema by itself is not sufficient to validate an Atom document. (However, web-based validators are available. [5])
- The lack of a normative schema complicates OGC's ability to develop compliance tests for OWS Context. OGC's current method for validating an XML encoding format relies exclusively on XML Schema validation. OGC would likely need to adopt Atom's validation techniques to establish a compliance test for OWS Context.

NOTE Atom includes an informative RELAX NG compact schema in appendix B of its specification. Also, web-based Atom validators are available.

### 5.6.3 Further discussion on the proposed move to Atom

Atom is a mass-market format that was developed for the purpose of syndication. The OWS-7 Information Sharing subthread explored the possibility of using the Atom format for representing an OWS Context document. The results of that effort are recorded in this engineering report.

Ultimately, the OGC membership will determine whether Atom is a good fit for this purpose. If the eventual consensus is that it is not, then that would mean in the opinion of the OGC membership that this was a proposal to use Atom in a way in which it was not intended.

In the end, Atom is just an encoding. What is really most important is that the "information content" in an OWS Context document is sufficient to capture everything needed so that a common operational picture can be shared among Integrated Clients with no information loss. If the OGC membership finds that Atom *is* a good fit, then that will have the benefit of opening up OWS Context documents to mass-market clients and APIs.

## 6 OWS Context information model

This clause presents the information content of an OWS Context document and a proposed Atom encoding as explored in OWS-7.

### 6.1 OWS Context information model overview

At a high level, an OWS Context document can be thought of as a container that holds a collection of *resources*. A resource can be characterized as follows:

- is representative of a particular data entity that contains geographic information or is associated with a geographic location,
- belongs to one or more categories,
- optionally, includes information on how it should be portrayed, and
- optionally, may have a relationship to other resources. Examples include:
    - a child's reference to a parent
    - a reference to an alternative file format or representation
    - a reference to a more authoritative version, and
    - a reference to an annotation.

In this clause these characteristics of a resource are further developed and mapped into specific elements provided by the Atom format.

In the clauses that follow, the prefix **atom:** refers to the Atom namespace http://www.w3.org/2005/Atom. The reader will also notice that most of the Atom elements serve a dual use in that they appear as children of **atom:feed** as well as **atom:entry**.

Finally, the presence of a particular property is indicated as follows:

- *[conditionally] required by atom*
- *[conditionally] required by OWC*
- *optional*

### 6.2 OWS Context document example

Here is an example of a complete OWS Context document that is representative of the work of the OWS-7 Information Sharing subthread. The elements and attributes included in this example are explained in this subclause and the subclauses that follow.

EXAMPLE

```
<?xml version="1.0" encoding="utf-8"?>
<feed xmlns="http://www.w3.org/2005/Atom"
      xmlns:georss="http://www.georss.org/georss"
      xmlns:gml="http://www.opengis.net/gml"
```

 

```
      xmlns:owc="http://www.opengis.net/owc">

  <id>urn:uuid:0fe0d764-b495-496c-ba30-c40b7c68b171</id>
  <title>OWS-7 OWS Context Document Example: Haiti</title>
  <updated>2010-06-04T20:25:20.38Z</updated>

  <General xmlns="http://www.opengis.net/owc">
    <Version>0.7.1</Version>
  </General>

  <where xmlns="http://www.georss.org/georss">
    <gml:Envelope srsName="EPSG:4326">
      <gml:lowerCorner>15.623 -76.171</gml:lowerCorner>
      <gml:upperCorner>22.522 -66.758</gml:upperCorner>
    </gml:Envelope>
  </where>

  <!--**************-->
  <!-- A WMS Layer. -->
  <!--**************-->
  <entry>
    <id>urn:uuid:5d33bd66-b73a-415f-8cc7-cbfcdec7692d</id>
    <title>Lakes</title>
    <updated>2010-06-04T20:25:20.39Z</updated>

    <where xmlns="http://www.georss.org/georss">
      <gml:Envelope srsName="EPSG:4326">
        <gml:lowerCorner>15.623 -76.171</gml:lowerCorner>
        <gml:upperCorner>22.522 -66.758</gml:upperCorner>
      </gml:Envelope>
    </where>

    <category
        label="A dataset which belongs to a WMS (Web Map
Service)."
        scheme="urn:ogc:def:ebRIM-
ClassificationScheme:DatasetObjectTypes"
        term="urn:ogc:def:ebRIM-
ObjectType:OGC:Dataset:Layer"/>

    <content type="text/xml">
      <Service type="urn:ogc:serviceType:WebMapService"
               xmlns="http://www.opengis.net/owc">
        <Name>MINUSTAH.HTI_LAKERESA_LAKE</Name>
        <ServiceVersion>1.3.2</ServiceVersion>
        <OnlineResource

xlink:href="http://demo.cubewerx.com/demo/cubeserv/cubeserv.cg
i?CONFIG=haiti"
            xmlns:xlink="http://www.w3.org/1999/xlink"/>
        <OutputFormat>image/png</OutputFormat>
      </Service>
    </content>
```

```
  </entry>

  <!--********************************-->
  <!-- A Simple In-Line Text Resource. -->
  <!--********************************-->
  <entry>
    <id>urn:uuid:4c368be3-4257-4db0-973d-4c0960bff92b</id>
    <title>Geographic information about Haiti (Text)</title>
    <updated>2010-06-04T20:25:20.39Z</updated>

    <category
        label="A resource consisting primarily of words for
reading. Examples include books, letters, dissertations,
poems, newspapers, articles, archives of mailing lists. Note
that facsimiles or images of texts are still of the genre
Text."
        scheme="urn:ogc:def:ObjectType:OtherResources"
        term="urn:ogc:def:ebRIM-ObjectType:OGC:Document"/>

    <content type="text">Haiti, officially the Republic of
Haiti is a Caribbean country. Along with the Dominican
Republic, it occupies the island of Hispaniola, in the Greater
Antillean archipelago.</content>
  </entry>

  <!--****************************************-->
  <!-- A PDF Document Resource (By Reference). -->
  <!--****************************************-->
  <entry>
    <id>urn:uuid:829440e2-f9fb-4236-b9c3-5252e00e77a4</id>
    <title>Background on Haiti &amp; Haitian Health
Culture</title>
    <updated>2010-06-04T20:25:20.39Z</updated>

    <where xmlns="http://www.georss.org/georss">
      <gml:Envelope srsName="EPSG:4326">
        <gml:lowerCorner>15.623 -76.171</gml:lowerCorner>
        <gml:upperCorner>22.522 -66.758</gml:upperCorner>
      </gml:Envelope>
    </where>

    <category
        label="A resource consisting primarily of words for
reading. Examples include books, letters, dissertations,
poems, newspapers, articles, archives of mailing lists. Note
that facsimiles or images of texts are still of the genre
Text."
        scheme="urn:ogc:def:ObjectType:OtherResources"
        term="urn:ogc:def:ebRIM-ObjectType:OGC:Document"/>

    <content
        src="http://www.cookross.com/docs/haiti.pdf"
        type="application/pdf"/>
```

```
  </entry>
</feed>
```

## 6.3 The OWS Context document container

This subclause discusses the OWS Context document container which represents the OWS Context document itself.

### 6.3.1 The container node *(required by Atom)*

The OWS-7 Information Sharing subthread selected the Atom format for representing an OWS context document. Like all XML documents, an Atom document has a single root node. And the name of the root node in an Atom document is **atom:feed** as shown here:

EXAMPLE

```
<feed xmlns="http://www.w3.org/2005/Atom"
      xmlns:gml="http://www.opengis.net/gml"
      xmlns:georss="http://www.georss.org/georss"
      xmlns:owc="http://www.opengis.net/owc">
  <id> … </id>
  <title> … </title>
  …
  <entry/>
  <entry/>
  <entry/>
  …
  <entry/>
</feed>
```

All elements defined by the Atom specification appear in the http://www.w3.org/2005/Atom namespace. The http://www.opengis.net/owc namespace holds extension elements introduced by OWS Context. The http://www.georss.org/georss and http://www.opengis.net/gml namespaces are together used to assign a geographic location to a context document or resource. Elements from other namespaces may also be present as required by a specific instance of a context document (most likely within the <content> element).

### 6.3.2 Document version *(required by OWC)*

An OWS Context document contains a version number that indicates the version of the specification to which an instance document conforms.

The OWS-7 Information Sharing subthread could not identify a suitable element in the Atom namespace in which to place a version number. For this reason, extension elements in the http://www.opengis.net/owc namespace were introduced to hold the version number.

EXAMPLE

```
<feed>
```

```
  …
  <owc:General>
    <owc:Version>0.7.1</owc:Version>
  </owc:General>
  …
</feed>
```

The final version of the schemas and associated examples created in OWS-7 were given the version number **0.7.1**.

### 6.3.3 Geospatial location *(required by OWC)*

From the beginning, one of the key goals of OWS Context modelling has been the desire to associate a geographic location with every resource. The OWS-7 Information Sharing subthread found that GeoRSS was a logical choice to serve as the container for the location geometry. Indeed, GeoRSS was invented primarily to provide the ability to add location information to a conventional RSS feed or item.

EXAMPLE

```
<feed>
  …
  <georss:where>
    <gml:Envelope srsName="EPSG:4326">
      <gml:lowerCorner>15.623 -76.171</gml:lowerCorner>
      <gml:upperCorner>22.522 -66.758</gml:upperCorner>
    </gml:Envelope>
  </georss:where>
  …
</feed>
```

NOTE This proposal for the use of **georss:where** comes directly from the GML profile for GeoRSS: http://georss.org/gml .

A geographic location assigned to the context document represents the overall extent of the context document. A very common use case can be seen in the above example where the geographic location is represented by **gml:Envelope**. To a typical Integrated Client, this would represent the initial extents of its map window after loading a context document.

NOTE The content of **georss:where** is limited to GML, such as **gml:Envelope** or **gml:Point**. Other information often used in geocoding such as the name of a town or a building number and additional descriptive information could be carried in other elements within an Atom entry such as **atom:title**, **atom:subtitle** or **atom:summary**. It would also be possible to add one or more extension elements in the OWC namespace to target a specific purpose.

### 6.3.4 Identifier *(required by Atom)*

As required by Atom, an OWS Context document shall be assigned a unique identifier. The **atom:id** element serves this purpose:

EXAMPLE

```
<feed>
  …
  <id>urn:uuid:ACC10485-5A70-4D90-B033-7AE229AA54B3</id>
  …
</feed>
```

According to the Atom specification, **atom:id** *"conveys a permanent, universally unique identifier for an entry or feed."* And *"its content MUST be an IRI, as defined by RFC3987."*

### 6.3.5 Title *(optional)*

The title conveys a human-readable purpose for the context document; Atom provides **atom:title** for this purpose.

EXAMPLE

```
<feed>
  …
  <title>Damaged Roads in Haiti</title>
  …
</feed>
```

### 6.3.6 Author *(conditionally required by Atom)*

One or more authors may be directly assigned to an OWS Context document via **atom:author**.

Atom conditionally requires the presence of **atom:author** as a child of **atom:feed**: *"**atom:feed** elements MUST contain one or more **atom:author** elements, unless all of the **atom:feed** element's child **atom:entry** elements contain at least one **atom:author** element."*

EXAMPLE

```
<feed>
  …
  <author>
    <name>ACME Corporation</name>
    <email>information@acme.com</email>
    <uri>http://www.acme.com/</uri>
  </author>
  …
</feed>
```

### 6.3.7 Updated *(required by Atom)*

Atom provides **atom:updated** to assign a creation (or update) date to an OWS Context document.

Atom declares *"**atom:feed** elements MUST contain exactly one **atom:updated** element."*

EXAMPLE

 

```
<feed>
  …
  <updated>2010-05-18T16:43:00Z</updated>
  …
</feed>
```

Atom also states *“The "**atom:updated**" element is a Date construct indicating the most recent instant in time when an entry or feed was modified in a way the publisher considers significant. Therefore, not all modifications necessarily result in a changed* **atom:updated** *value.”*

And about the Date construct: *“A Date construct is an element whose content MUST conform to the "date-time" production in [RFC3339]. In addition, an uppercase "T" character MUST be used to separate date and time, and an uppercase "Z" character MUST be present in the absence of a numeric time zone offset.”*

### 6.3.8 Link *(optional)*

According to the Atom specification, *“The "atom:link" element defines a reference from an entry or feed to a Web resource. This specification assigns no meaning to the content (if any) of this element.”*

The **rel** attribute indicates the relation type of the link. Here is an example that demonstrates how **atom:link** might be used in an OWS Context document to specify relations of type “self” and “related”.

EXAMPLE

```
<feed>
  …
  <!-- A link that references the hosted location of the
       enclosing document. -->
  <link rel="self" href="http://example.org/context/haiti" />

  <!-- A link that references the OWS Context spec. -->
  <link rel="related" href="http://www.opengis.net/owc" />
  …
</feed>
```

Note that according to the Atom specification, *“If the "rel" attribute is not present, the link element MUST be interpreted as if the link relation type is "alternate".”*

### 6.3.9 Other elements

These elements are defined by the Atom specification but the OWS-7 Information Sharing subthread did not evaluate their potential use with respect to the OWS Context document container: **atom:contributor**, **atom:generator**, **atom:icon**, **atom:logo**, **atom:published**, **atom:rights**, **atom:source** and **atom:subtitle**.

 

One should not infer that because OWS-7 did not make use of these items, that their use is not recommended. It only means that they did not provide much value in the rapid development and deployment environment that characterizes a testbed. In fact, these optional elements should be employed whenever possible, following the guidance in the Atom specification itself.

### 6.4 The OWS Context document resource

As stated in clause 6, an OWS Context document is a container for one or more *resources*. A resource represents a data entity that has a geographic location. A discussion of resource types is in subclause 5.2. This subclause discusses the mapping of those resource types into Atom.

#### 6.4.1 The resource container *(required by Atom)*

Atom defines **atom:entry** as a container for each individual item in a feed. The Information Sharing subthread found **atom:entry** a good fit for representing a resource in an OWS Context document.

EXAMPLE

```
<entry>
  …
  <id> … </id>
  <title> … </title>
  …
</entry>
```

In most situations, one would expect a context document to contain one or more resources. An *empty* OWS Context document would be defined as a document that contains no resources. This situation can be represented in Atom since it is legitimate to create an **atom:feed** document that does not contain any **atom:entry** elements.

#### 6.4.2 Geospatial location *(required by OWC)*

Subclause 6.3.3 shows the use of **georss:where** for assigning a geographic location to the OWS Context document itself. That element also serves the purpose for assigning a geographic location to a resource.

EXAMPLE

```
<entry>
  …
  <georss:where>
    <gml:Envelope srsName="EPSG:4326">
      <gml:lowerCorner>15.623 -76.171</gml:lowerCorner>
      <gml:upperCorner>22.522 -66.758</gml:upperCorner>
    </gml:Envelope>
  </georss:where>
  …
</entry>
```

 

### 6.4.3 Identifier *(required by Atom)*

As required by **atom:entry**, a resource shall be assigned a unique identifier. The **atom:id** element serves this purpose.

EXAMPLE

```
<entry>
  …
  <id>urn:uuid:EE2B5689-2181-41DF-9E6E-4B6486CF0CD6</id>
  …
</entry>
```

See subclause 6.3.4 for more information regarding **atom:id**.

### 6.4.4 Title *(required by Atom)*

The title conveys a human-readable purpose for the resource. Atom provides **atom:title** for this purpose.

EXAMPLE

```
<entry>
  …
  <title>Damaged Roads in Haiti</title>
  …
</entry>
```

When the resource represents an OGC dataset (e.g., Layer, Feature Type, Coverage), **atom:title** should contain the title assigned to the dataset as it appears in a service capabilities document.

### 6.4.5 Author *(conditionally required by Atom)*

One or more authors may be directly assigned to a resource via **atom:author**.

Atom conditionally requires the presence of **atom:author** as a child of **atom:entry**: ***atom:entry*** *elements MUST contain one or more* ***atom:author*** *elements, unless the* ***atom:entry*** *contains an* ***atom:source*** *element that contains an* ***atom:author*** *element or, in an Atom Feed Document, the* ***atom:feed*** *element contains an* ***atom:author*** *element itself.*

EXAMPLE

```
<entry>
  …
  <author>
    <name>ACME Corporation</name>
    <email>information@acme.com</email>
    <uri>http://www.acme.com/</uri>
  </author>
  …
```

 

```
</entry>
```

### 6.4.6 Summary *(optional)*

**atom:summary** may be used to associate a short summary with a resource. ("Abstract" and "excerpt" are synonyms for summary.)

Subclause 4.1.1.1 in the Atom specification says this about **atom:summary**: *It is advisable that each **atom:entry** element contain a non-empty **atom:title** element, a non-empty **atom:content** element when that element is present and a non-empty **atom:summary** element when the entry contains no **atom:content** element. However, the absence of **atom:summary** is not an error, and Atom Processors MUST NOT fail to function correctly as a consequence of such an absence.*

### 6.4.7 Category *(optional)*

Atom provides **atom:category** for classifying an entry. **atom:entry** may contain one or more **atom:category** elements. **atom:entry** contains three attributes:

- **term** *(must be present on **atom:category**)*
- **scheme** *(optional, but if present must be an IRI)*
- **label** *(optional, but if present is language-sensitive)*

Atom does not assign any values to those attributes; that is left to the discretion of the feed creator.

The OWS-7 Information Sharing subthread saw the availability of **atom:category** as an opportunity to optionally classify each context document resource. The group concluded that the presence of one or more instances of **atom:category** whose properties are populated from an established vocabulary would provide valuable processing guidance to consumers of OWS Context documents.

The Information Sharing subthread agreed that the values for **term** and **scheme** should eventually come from a list of authoritative identifiers published by OGC. For the testbed experiments, the Information Sharing subthread chose identifiers that were previously defined for use in the OpenSearch catalogs deployed in the ADSD subthread.

Here are three examples of the application of **atom:category** to a resource. In the first, **atom:category** is used to classify as a resource as a WMS Layer:

EXAMPLE 1 — Classify Resource as a LAYER

```
<entry>
  …
  <category term="urn:ogc:def:ebRIM-
ObjectType:OGC:Dataset:Layer"
            scheme="urn:ogc:def:ebRIM-
ClassificationScheme:DatasetObjectTypes"
```

```
            label="A dataset which belongs to a WMS (Web Map
Service)"/>
  …
</entry>
```

In this second example, **atom:category** is applied twice: once to indicate that a resource is a WFS Feature Type, the second to indicate that that data type is Vector:

EXAMPLE 2 — Classify Resource as a FEATURE TYPE and that it is VECTOR

```
<entry>
  …
  <category term="urn:ogc:def:ebRIM-
ObjectType:OGC:Dataset:FeatureType"
            scheme="urn:ogc:def:ebRIM-
ClassificationScheme:DatasetObjectTypes"
            label="A dataset which belongs to a WFS (Web
Feature Service)"/>

  <category term="urn:ogc:def:ebRIM-
ObjectType:OGC:Dataset:Vector"
            scheme="urn:cslt:def:DataResources"
            label="Thematic or topographic data represented by
single points, strings of points (lines or arcs), or closed
lines (polygons)."/>
  …
</entry>
```

In this third example, **atom:category** is present to indicate that the resource is a Document (Text):

EXAMPLE 3 — Classify Resource as Media Type DOCUMENT

```
<entry>
  …
  <category term="urn:ogc:def:ebRIM-ObjectType:OGC:Document"
            scheme="urn:ogc:def:ObjectType:OtherResources"
            label="A resource consisting primarily of words
for reading. Examples include books, letters, dissertations,
poems, newspapers, articles, archives of mailing lists. Note
that facsimiles or images of texts are still of the genre
Text."/>
  …
</entry>
```

#### 6.4.8 Content *(required by OWC)*

Atom provides **atom:content** to contain (embed) or link to (reference) the content of an entry. Subclause 6.5 provides a detailed discussion on representing the content of a resource.

#### 6.4.9 Link *(optional)*

Subclause 6.3.8 presents **atom:link** and its potential use with respect to the OWS Context document container. **atom:link** may also be used to reference one or more information sources to an OWS Context resource.

Possible uses include:

- a child's reference to a parent
- a reference to an alternative file format or representation (e.g., a Layer may refer to a Feature Type or vice-versa)
- a reference to a more authoritative version, and
- a reference to an annotation or authoritative metadata.

EXAMPLE

```
<entry>
  …
  <link rel="related"
    href="http://example.org/metadata/ISO19115"/>
  …
</entry>
```

#### 6.4.10 Other elements

These elements are defined by the Atom specification but the OWS-7 Information Sharing subthread did not evaluate their potential use with respect to an OWS Context document resource: **atom:contributor**, **atom:generator**, **atom:icon**, **atom:logo**, **atom:published**, **atom:rights**, **atom:source** and **atom:subtitle**.

Note The commentary in subclause 6.3.9 applies here as well.

### 6.5 Resource Content Details

Atom provides **atom:content** to contain (embed) or link to (reference) the content of an entry. The content of **atom:content** is language-sensitive.

In OWS-7, the resource model described in subclause 5.2 was used as a basis for modeling the **atom:content** component of a resource:

- When the resource represents an OGC dataset, the content of **atom:content** is service metadata.
- When the resource represents a media type, **atom:content** links to or contains the media type as described by Atom.

 

**atom:content** has two attributes: **type** and **src**. For use within an OWS Context document, the Information Sharing subthread applied these attributes as follows:

- **type** may be assigned one of the Atom-predefined values “text”, “html” or “xhtml”. Otherwise, it shall conform to the syntax of the MIME media type of the resource type.
- The value of the **src** attribute is an IRI that references a remote resource. The **src** attribute is omitted if the content of the resource is embedded in **atom:content**. If the **src** attribute is present, then **atom:content** shall be empty.

### 6.5.1 Content for OGC datasets (service metadata)

For the situation when a resource represents an OGC dataset, the Information Sharing subthread defined elements in the OWS Context namespace to hold service metadata. The root element of this service metadata is **owc:Service**. The **type** attribute is set to the value “text/xml”. The child elements of **owc:Service** are used to hold the values of the parameters that should be supplied to the service in order to retrieve the dataset.

NOTE 1 The Information Sharing subthread had a robust discussion regarding an alternative approach to the representation of service metadata in an OWS Context document. See subclause 6.5.6for a summary of that discussion.

NOTE 2 Incorporating datasets created by a WPS were briefly discussed but not modeled in OWS-7.

This example shows how **atom:content** with a child of **owc:Service** is used to represent a WMS Layer:

EXAMPLE

```
<entry>
  …
  <content type="text/xml">
    <owc:Service type="urn:ogc:serviceType:WebMapService">
      <owc:Name>MINUSTAH.HTI_LAKERESA_LAKE</owc:Name>
      <owc:ServiceVersion>1.3.2</owc:ServiceVersion>
      <owc:OnlineResource
xlink:href="http://demo.cubewerx.com/demo/cubeserv/cubeserv.cg
i?CONFIG=haiti"/>
      <owc:OutputFormat>image/png</owc:OutputFormat>
    </owc:Service>
  <content type="text/xml">
  …
</entry>
```

**owc:content** may contain style information; see subclause 6.5.4. Provisions have also been made for modeling related datasets or other resource types; see subclause 6.4.9. Finally, the Information Sharing subthread discussed the possibility of including search results in an OWS Context document; see the discussion in subclause 6.6. (Note that in the general case search results can contain references to OGC datasets as well as media types. That is, search results can be heterogeneous or “mixed”.)

 

### 6.5.2 Content for media types

Setting the content of **atom:content** to contain OGC and non-OGC media types is straightforward. In this first example the content is plain text.

EXAMPLE 1 — PLAIN TEXT

```
<entry>
  …
  <content type="text">Haiti, officially the Republic of Haiti
is a Caribbean country. Along with the Dominican Republic, it
occupies the island of Hispaniola, in the Greater Antillean
archipelago.</content>

  <!–- Text courtesy of http://en.wikipedia.org/wiki/Haiti -->
  …
</entry>
```

NOTE See the Atom specification for the rules governing embedding content of type “html” and “xhtml”.

In this second example, a PDF document is linked to **atom:content** via the **src** attribute:

EXAMPLE 2 — NON-OGC MEDIA TYPE (PDF), BY REFERENCE

```
<entry>
  …
  <content type="application/pdf"
    src="http://www.cookross.com/docs/haiti.pdf" />
  …
</entry>
```

NOTE See the discussion in on the handling of relative IRI references in Atom.

In the third example, GML, an OGC media type, is included by reference.

EXAMPLE 3 — OGC MEDIA TYPE (GML), BY REFERENCE

```
<entry>
  …
  <content type="application/gml+xml; version=3.2.1"

src="http://www.ogcnetwork.net/schemas/ows7/data/Land_Use.gml"
/>
  …
</entry>
```

NOTE Would it be of value to refer to the GML application schema? If so, how?

It is possible to set the content of **atom:content** to any well-formed XML document. Via that provision, GML and KML may be embedded directly into an OWS Context document. **atom:content** can also also hold *binary* data by first encoding it as base64. (Note that binary or text data could first be compressed before encoding it as base64.) See the Atom specification for details.

An important use case for a resource with GML or KML embedded content would be for that resource to represent an annotation for an image. Good practice would dictate limiting the size of embedded content and instead using the **src** attribute to refer to a "large" instance of a media type by reference.

### 6.5.3 Resolving a relative reference to a media type

The OWS-7 Information Sharing subthread investigated the means of handling relative references to media types (via the **atom:content src** attribute) in Atom. On this topic the Atom specification says, *"Any element defined by this specification MAY have an xml:base attribute [W3C.REC-xmlbase-20010627]. When xml:base is used in an Atom Document, it serves the function described in section 5.1.1 of [RFC3986], establishing the base URI (or IRI) for resolving any relative references found within the effective scope of the xml:base attribute."*

In the absence of **xml:base**, the Atom specification does not provide guidance on how a relative URI is to be resolved. However, the OWS-7 Information Sharing thread came to the conclusion that in the absence of **xml:base**, a relative URI should be resolved relative to the base URI of the encapsulating OWS Context document. This position is supported by XML Base and RFC 3986.

Support for relative references is crucial for supporting the packaging mechanism described in subclause 6.7.

### 6.5.4 Styling the resource content

A common practice in OGC service implementations is to allow client-directed styling of resources such as Layers and Feature Types. A WMS may advertise named default and supported styles for each of its Layers in its capabilities document. Additionally, an SLD-enabled WMS may style a Layer or Feature Type via an SLD document submitted as part of a *GetMap* request.

The schema package bundled with this document (see Annex A) includes examples of how styling information may be incorporated into the resources in an OWS Context document.

### 6.5.5 Remote content and HTTP headers

Prior to accessing a remote resource, it is potentially useful to have more information than the just the content type. In addition to *Content-Type*, HTTP response headers provide information such as *Content-Length* and *Last-Modified*. Last-Modified conveniently maps to **atom:updated**. But what about the many other HTTP response headers?

The information typically provided in HTTP response headers can also be useful in the event that access to a remote resource is not possible. Additional information could also be useful in dealing with embedded content, local files or other protocols such as FTP.

Regarding the potential value of incorporating the information from other HTTP headers, the OWS-7 Information Sharing subthread asked the following:

- In addition to Last-Modified, are there other cache control headers that would be useful? Or should the typical client developer instead rely on the network infrastructure rather than worry about implementing a full-blown caching capability?
- Are there other HTTP headers that could be of value in an OWS Context document? (e.g., Charset)
- Would it be better to add these properties to the "URLType" definition in OWS Common?

#### 6.5.6 Content for OGC datasets, individual parameters or pre-formed request?

Past modeling exercises of the OWS Context documents has been method agnostic; all request information is represented in the context document as individual elements (parameters) with a service endpoint expressed as a URL. It is up to the client to put the request back together based on this information. If the method is GET then the client is expected to rebuild a KVP string to send to the service. If the method is a POST then the client is expected to rebuild the XML request document.

The Information Sharing subthread considered an alternative approach: How about storing the request itself – not a set of information about the request? If the method is GET then the service URL would contain the entire query string. If the method is POST then content would include a POST request. If the method is SOAP then the content would include the WS Online Resource Element.

Here are pros and cons with respect to embedding the entire pre-formed request:

<u>Pros</u>:

- Captures the request without need for transformation and without the risk of loss of information.
- Adaptive to future changes to the request – the context document doesn't need to have knowledge of the implementation specifics involved in the request.

<u>Cons</u>:

- Clients might have to parse requests – they normally only parse responses. (For example, to extract a filter from a WFS *GetFeature* request.) This is extra work for the client implementers.
- If the entire pre-formed request is embedded, then a client would potentially need to deal with any number of encodings – and would be unlikely to support them all.
- If the entire pre-formed request is embedded, that could make it more difficult to share context documents or use them in different environments. All clients would

need to support the same encodings. (Alternatively, if the method-agnostic approach is used a client can use any method it prefers.)

The OWS-7 Information Sharing thread ultimately decided to stay with the method-agnostic approach for modeling the OWS Context document. However, the group expected this topic to eventually be revisited by a future SWG.

### 6.6 OWS Context and Search

In the FDF thread in OWS-7, the Information Sharing subthread discussed possibilities for harmonizing the use of Atom for representing OWS-Context documents with the Atom search results returned by an OpenSearch search engine. (The ADSD subthread in OWS-7 explored OpenSearch.)

Two general ideas emerged from the discussions:

- Treat an OpenSearch results document as another resource type and embed or reference those results via **atom:content**. If the encoding of the OpenSearch results were harmonized with the proposed Atom encoding of the OWS Context document, this would resemble the inclusion of another context document.

- Consider the OWS Context document *itself* to be a response from an OpenSearch query. This idea goes to the very heart of the harmonization ideas put forth by some participants in the OWS-7 Information Sharing thread.

In particular, during the course of prototyping in OWS-7, some participants noted the potential value of applying the data/service binding approach developed for OWS Context to the ADSD OpenSearch results. The ADSD OpenSearch services deployed in OWS-7 did not produce search results that allowed clients to directly bind to a discovered dataset (such as a Layer or Feature Type). Binding could only take place by visiting a web page reference in the search results and then traversing one or more additional web pages to find a service URL.

### 6.7 Packaging OWS Context and referenced local media

The OWS-7 Information Sharing subthread discussed possibilities for packaging an OWS Context document and any referenced local media types into an archive for transport. Interest in the packaging problem was in part inspired by an earlier proposal for the Location Organizer Folder (LOF). (See subclause 5.4.6.)

An approach worth considering is an archiving model associated with KML. [2] KML files are often bundled together with referenced media types into a KMZ file, which is a “zip” file with a “.kmz” extension. By convention, a file named “doc.kml” in the root folder is considered the root document, and referenced files are placed in subfolders.

Ultimately, it is the relationship between a main context document and its associated files that should be established, regardless of whether they are packaged in an archive file, or reside as distinct entities on a file system. Here some basic principles that should apply:

 

– The OWS Context document should be placed in a top-level folder and named, say, “context.xml”.

– All links in the OWS Context document that are relative should be considered relative to the location of the OWS Context document. This principle should apply whether the OWS Context document is located on a local disk, hosted on a web server or placed in an archive. (See subclause 6.5.3 for a discussion on handling relative references in Atom.)

### 6.8 Integrated Client snapshots from OWS-7

This subclause presents screenshots of an Integrated Client loading an OWS Context document. All screenshots are provided courtesy of Intergraph Corporation who extended their Integrated Client during OWS-7 to support the Atom-based OWS Context document format developed in the Information Sharing subthread.

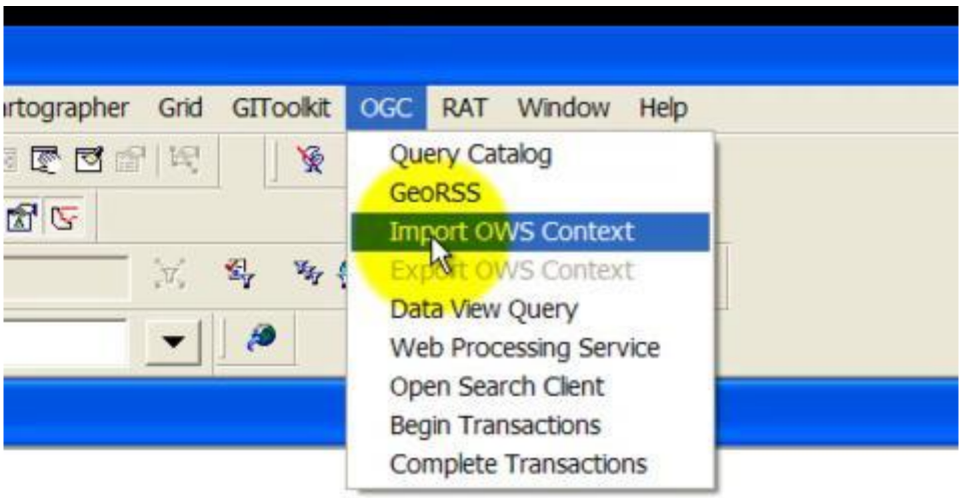


**Figure 3 — Accessing “Import OWS Context” from the Integrated Client menu bar**

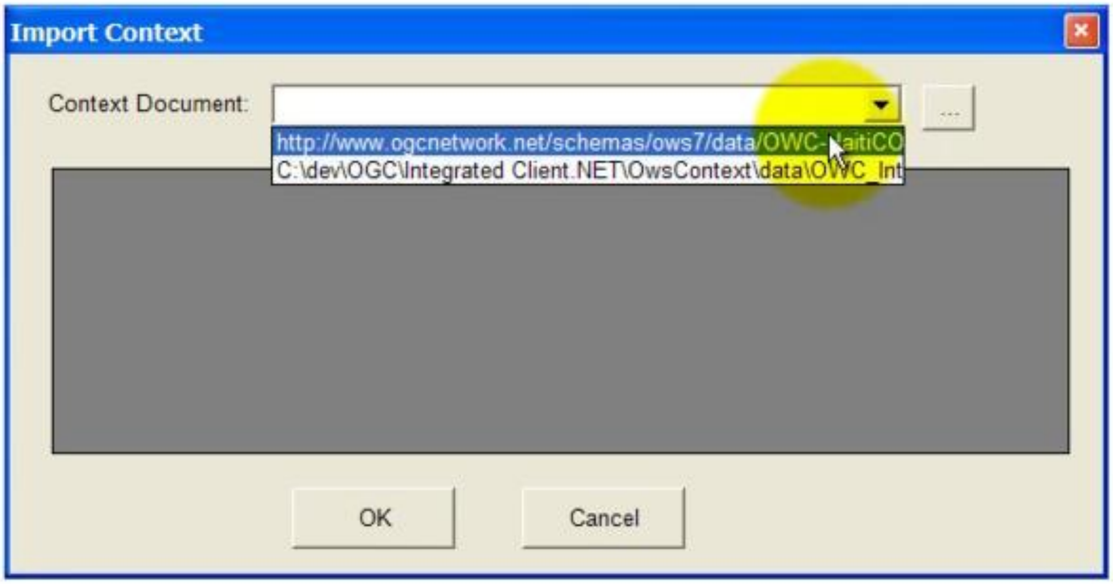


**Figure 4 — Selecting an OWS Context document**

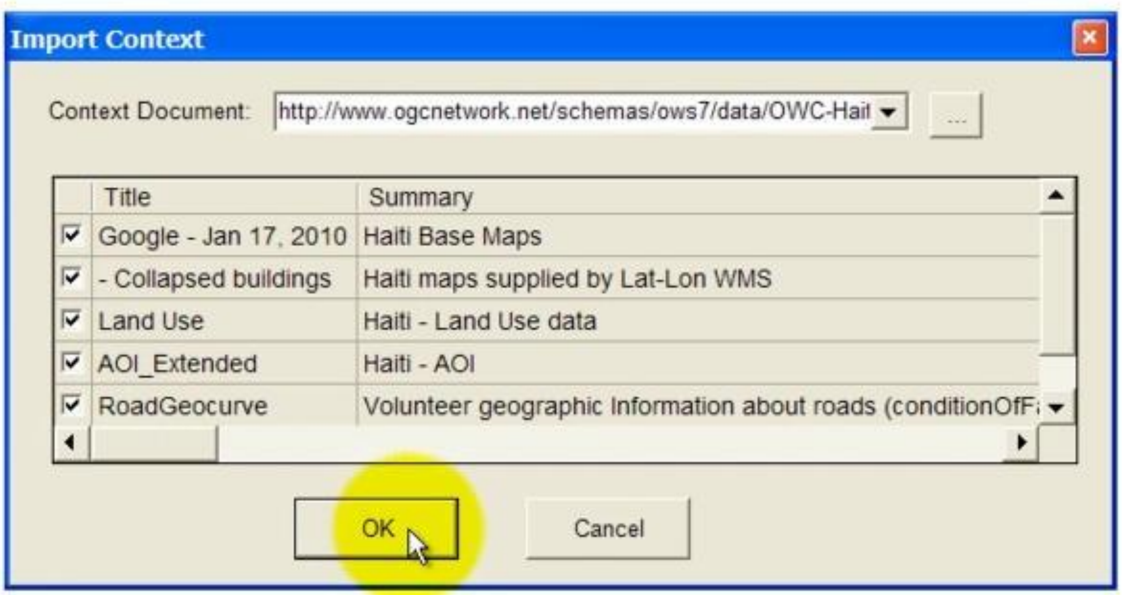


**Figure 5 — Inspecting the content of an OWS Context document**

 


**Figure 6 — Connecting to the resources in the OWS Context document**

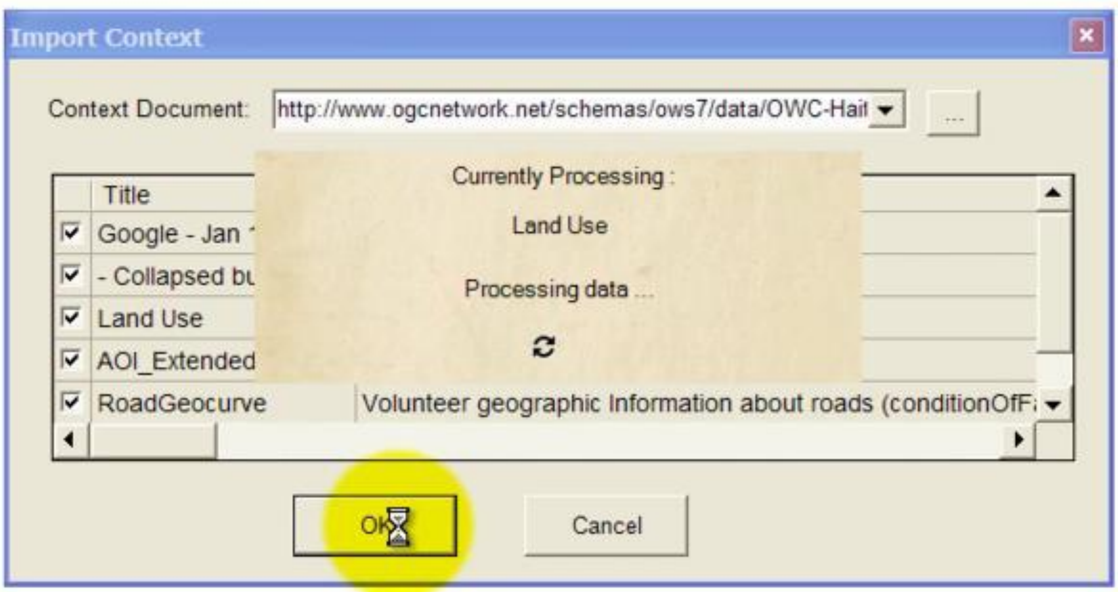


**Figure 7 — Connecting to the resources in the OWS Context document (continued)**

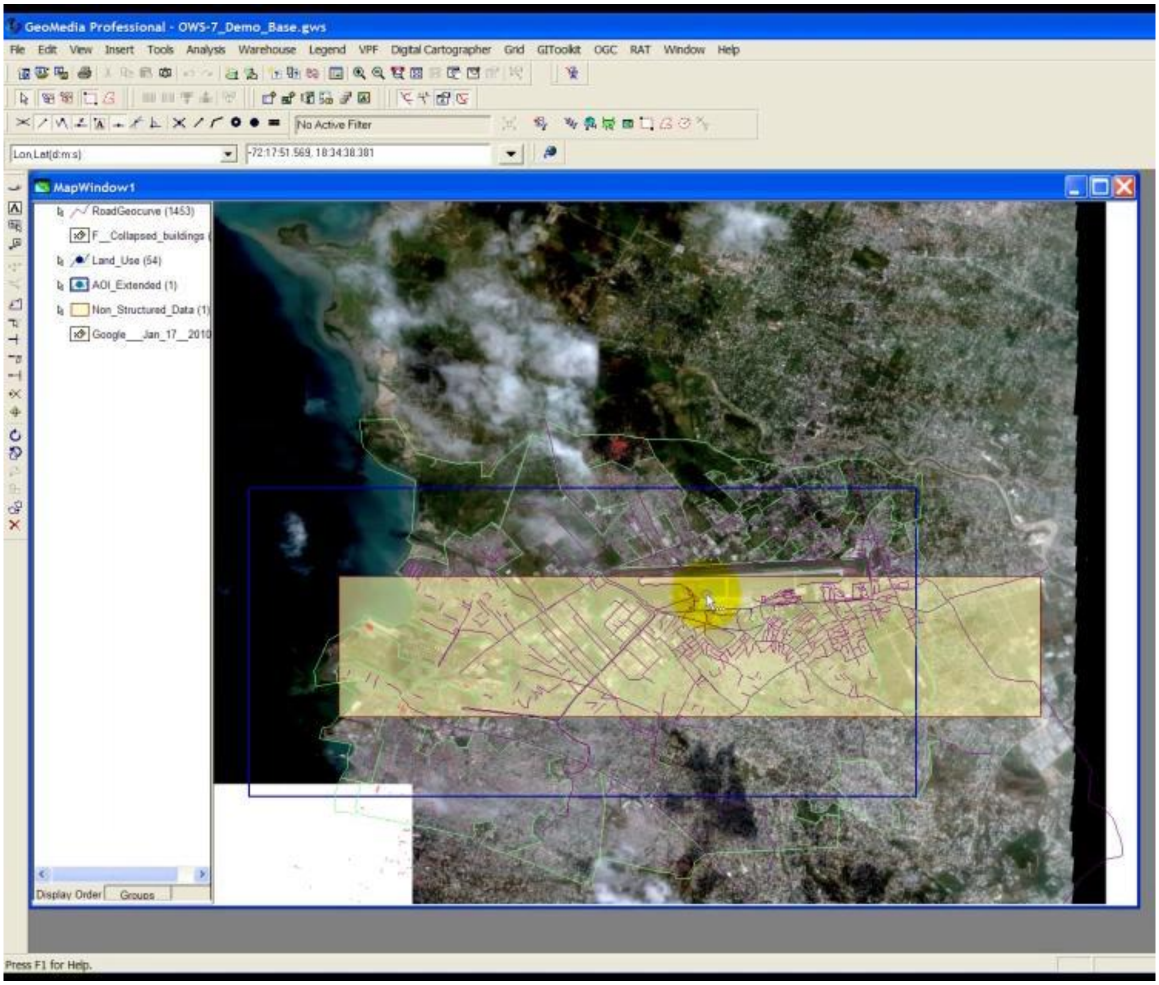


**Figure 8 — The Legend and Map View in the Integrated Client after loading the OWS Context document**

## 7 Future work

By its very nature, the rapid prototyping that occurs in a testbed such as OWS-7 does not allow time to explore every facet of an area of interest. The participants in the OWS-7 Information Sharing subthread believe the following topics merit further study to improve the viability of a future OWS Context specification:

– Is Atom a suitable encoding for an OWS Context document? Is Atom adequate for storing all of the information content desired in an OWS Context document? (See subclause 5.6 for pros and cons noted in OWS-7.)

– Revisit the encoding of service metadata to see that it is adequate for supporting XML/Post and SOAP.

  The experiments in OWS-7 focused on using the service metadata to construct “Get” binding requests. However in theory, by encoding the service metadata as individual parameters, a request payload should be able to be constructed with the desired encoding. The OWS Context encoding as outlined in this report does not specify a request method (e.g., Get, Post, SOAP); that is left up to the client. But, would there be a sound reason for doing so?

- The handing of additional resource types such as Sensor Observations and WPS process execution results should be explored. (Such a future investigation would also likely incorporate SOAP.)

- OWS-7 did not have the time to prototype the OWS Context packaging mechanism outlined in subclause 6.7. A prototyping exercise would be useful to test its viability.

- Evaluate the potential for harmonizing the encoding of the OWS Context document with the encoding of the query response from the OpenSearch interface that is under development in the Catalog 3.0 SWG. (An OpenSearch interface to OGC catalogs was also explored in the ADSD subthread in OWS-7.) If harmonizing were to prove feasible and desirable:

  - Catalog 3.0 search results (via OpenSearch) could potentially be a close sibling to a Context Document. That would mean that an Integrated Client would be able to bind to the resources returned in search results as easily as it can when processing resources contained within an OWS Context document.

  - Catalog 3.0 search results (obtained via OpenSearch) could potentially be referenced in an OWS Context document as described in subclause 6.6.

# Annex A

## XML Schema Documents and Examples

In addition to this document, this report includes several XML schema documents and examples of OWS Context documents encoded using the Atom format. These XML schema documents and examples are bundled in a zip file with this document.

# Annex B

## Mapping from OWS Context 0.3.1 Schema to Atom

The table in this annex was created during an early exploratory exercise in OWS-7 when the Information Sharing subthread was first considering the possibility of moving the OWS Context document format to Atom.

Atom Publishing Format reference links

- Specification: http://atompub.org/rfc4287.html
- Specification, but a bit friendlier: http://www.atomenabled.org/developers/syndication/
- Validation: http://www.validome.org/rss-atom/validate
- GeoRSS: http://georss.org

Key
n/n = not needed
n/p = not provided (so add custom XML to <content>)

**Table 1 — Mapping from OWS Context 0.3.1 schema to Atom**

| OWSContext | Atom Elements | Notes |
|---|---|---|
| OWSContext | feed | |
| @id | feed/id | |
| General<br>General/BoundingBox<br>General/Title<br>General/Abstract<br>General/Keywords<br>General/LogoURL<br>General/DescriptionURL<br>General/ServiceProvider<br>General/Extension | n/n<br>not there, keep ows:BoundingBox<br>title<br>subtitle<br>category<br>log<br>link/@rel<br>author<br>n/n (general Atom extension mechanism) | contains information about the Context document such as bounding box and CRS and information describing the Context document itself such as title, abstract, etc. |
| General/OpenSearch stuff | TODO: find OpenSearch inclusion in Atom. This exists | |

| | | |
|---|---|---|
| | somewhere already. | |
| ResourceList | n/n | |
| Resource | entry | |
| Resource@name<br>Resource@id<br>Resource@mime-type<br>Resource@group<br>Resource@hidden<br>Resource@opacity<br>Resource@queryable | entry/title<br>entry/id<br>entry/content/@type<br>n/p<br>n/p<br>n/p<br>n/p | why do we need Resource@name and Resource/Title? |
| Resource/Title | entry/title | why do we need Resource@name and Resource/Title? |
| Resource/Abstract | entry/summary | |
| Resource/Keywords | entry/category | |
| Resource/Identifier | n/p | |
| Resource/BoundingBox | n/p | |
| Resource/OutputFormat | n/p | |
| Resource/AvailableCRS | n/p | |
| Resource/Metadata | n/p | |
| Resource/DataURL | link/@rel | |
| Resource/MetaDataURL | link/@rel | |
| Resource/MinScale-Denominator | n/p | |
| Resource/MaxScale-Denominator | n/p | |
| Resource/Extension | extensionElement | |
| Resource/Related | entry/link/@rel="related" | XPath reference to a related resource. Primarily to relate image annotations to their related imagery within the OWC, but could also deep-link into XHTML |

| | | |
|---|---|---|
| | | or XML content in the OWC or out in the general Web. |
| Resource/Server | entry/content (use src for binding URL, and add attributes for version, service type, queryable, hidden) | the content element can be empty and have a src attribute instead, making it linked Content instead of inline. |
| Resource/DimensionList | n/p | |
| Resource/ResponseCRS | n/p | |
| Resource/ParameterList | n/p | |
| Resource/MaxFeatures | n/p | |
| Resource/Filter | n/p | |
| Resource/InlineGeometry | entry/content/@type="app/gml" | Feed/entry/content/@type can be "text", "html", or "xhtml" OR a mime type. |
| Resource/kml:Document | entry/content/@type="app/kml" | |
| Resource/VendorExtension | extensionElement | |
| Resource/StyleList | n/p | |
| Resource/InlineMedia | entry/content/@type="[mime type info] " | |
| Resource/InlineText | entry/content/@type="text" | |
| Resource/Resource | n/p | |

# Bibliography

[1] OGC Web Services Context Schema version 0.3.1, http://www.ogcnetwork.net/schemas/owc/0.3.1/

[2] Discussion on distributing KML files in a “zip” archive, KML Wikipedia page, http://en.wikipedia.org/wiki/Kml

[3] OGC Geospatial Fusion Services Test Bed documents (provided on request from techdesk@opengeospatial.org)

[4] Guidelines for Successful OGC Interface Standards, OGC document 00-014r1

[5] Atom validator, http://www.validome.org/rss-atom/validate

 