Open Geospatial Consortium

OGC GeoRSS
Encoding Standard

Warning

1

License Agreement

Attribution-ShareAlike
CC BY-SA

This license lets others remix, tweak, and build upon the GeoRSS work even for commercial purposes, as long as they credit you and license their new creations under the identical terms. This license is often compared to "copyleft" free and open source software licenses. All new works based on the GeoRSS standard will carry the same license, so any derivatives will also allow commercial use. This is the license used by Wikipedia, and is recommended for materials that would benefit from incorporating content from Wikipedia and similarly licensed projects. https://creativecommons.org/licenses/by-sa/4.0/ and https://creativecommons.org/licenses/by-sa/4.0/legalcode

Contents

i.   Abstract

GeoRSS is designed as a lightweight, community driven way to extend existing RSS feeds with simple geographic information. The GeoRSS standard provides for encoding location in an interoperable manner so that applications can request, aggregate, share and map geographically tag feeds.

ii.   Source of the content for this OGC document

The majority of the content in this OGC document is a direct copy of the content contained at www.georss.org. No normative changes have been made to the content. This OGC document does contain content not contained at www.georss.org. Specifically, while derived from the georss.org website, the Abstract, Keywords, Preface, Submitting Organizations, Endorsers, Terms and Definitions, and References sections in this document are not found on the georss.org website.

iii.   Validity of content

This document was created from content contained on www.georss.org in late January 2017. The content of this document has been reviewed by the original submission team and is declared to be accurate and true

iv.   Keywords

The following are keywords to be used by search engines and document catalogues.

ogcdoc, OGC document, georss, rss, feeds

v.   Preface

The original specification of GeoRSS was the result of a collaboration of Geo-IT professionals, OGC staff and members, and other highly creative individuals. This group set out to define the simplest possible geographic encoding that would still be expressive and standards-friendly enough to satisfy the professional geospatial community while at the same time being simple enough to garner quick acceptance from mainstream web and RSS developers. The first community version of GeoRSS was released in 2006.

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 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.

vi.   Submitting organizations

The following organizations submitted this Document to the Open Geospatial Consortium (OGC):

Carl Reed and Associates

Mikel Maron (as an individual)
Tumblingwalls
Galdos
IBM

vii.   Endorsers

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

Name	Affiliation
Allan Doyle	MIT Museum
Mikel Maron	Individual
Ron Lake	Galdos
Josh Lieberman	Tumblingwalls
Carl Reed	Carl Reed & Associates
Raj Singh	IBM
Satish Sankaran	Esri

1. Scope

GeoRSS is a simple approach for geo-enabling, or tagging, "really simple syndication" (RSS) feeds with location information. GeoRSS standardizes the way in which "where" is encoded with enough simplicity and descriptive power to satisfy most needs to describe the location of Web content. GeoRSS is extensible and upwardly-compatible with more sophisticated formats like the OGC Geography Markup Language (GML).

The initial goal for designing and documenting GeoRSS was to keep the encoding of geography on the Web from fracturing into various encodings the way RSS ended up, with multiple similar implementations. GeoRSS is also intended to be a lightweight way to express geography in other XML-based formats – including XHTML.

2. Conformance

Not applicable.

3. References

The following normative documents contain provisions that, through reference in this text, constitute provisions of 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.

GeoRSS: Geographically Encoded Objects for RSS feeds. www.georss.org (January 26/27 2017)

IETF RFC 4287: The Atom Syndication Format. 2005. https://tools.ietf.org/html/rfc4287 (December 19, 2016)

IETF RFC 5774: Considerations for Civic Addresses in the Presence Information Data Format Location Object (PIDF-LO) https://tools.ietf.org/html/rfc5774 (December 22, 2016)

OGC/ISO GML: Geography Markup Language Implementation Standard. 2003 http://portal.opengeospatial.org/files/?artifact_id=4700 (December 19, 2016)

4. Terms and Definitions

This document uses the terms defined in Sub-clause 5.3 of [OGC 06-121r8], which is based on the ISO/IEC Directives, Part 2, Rules for the structure and drafting of International Standards. In particular, the word "shall" (not "must") is the verb form used to indicate a requirement to be strictly followed to conform to this standard.

For the purposes of this document, the following additional terms and definitions apply.

4.1
GeoRSS
Location enabled feed.

5. Conventions

This sections provides details and examples for any conventions used in the document. Examples of conventions are symbols, abbreviations, use of XML schema, or special notes regarding how to read the document.

5.1 Identifiers
GeoRSS is identified by a namespace prefix "georss:". The model portrayed in Figure 1 below does not show, but allows, the inclusion of any other elements from other namespaces. Specifically, as an example any feed that uses an XML encoding will need to state the following:

```
xmlns="http://www.w3.org/2005/Atom"
xmlns:georss="http://www.georss.org/georss"
```

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

```
Please use version 1.1 of the GeoRSS schema. The 1.0 version of the
schema is still available for backwards compatibility.
```

5.2 GML profile and application schema for GeoRSS

All good information encodings should be backed by a formal definition. For GeoRSS GML, XML schema is used. This helps software developers--especially XML-savvy ones, know the full capabilities of the language, and program their software to handle any case, not just those they've come across in examples.

5.2.1 GML Profile

GML is a large language which is also defined by XML schema. To constraint (restrict) the size of a GML schema, OGC practice has emerged to define subsets of GML, called profiles, which contain only those elements of GML needed for the encoding job at hand.

The GeoRSS GML profile schema is here:

GeoRSS GML Profile [http://www.georss.org/xml/1.1/gmlgeorss.xsd].

There is also a UML model for this profile.

5.2.2 GeoRSS Application Schema

The application schema defines <georss:where> as the tag that signals geographic content--either in GeoRSS Simple or GML.

The GeoRSS Application Schema can be viewed here:

http://www.georss.org/xml/1.1/georss.xsd

The UML model for the application schema is provided in Annex D.

6. Overview of GeoRSS

There are currently two encodings of GeoRSS: Simple and GML:

- GeoRSS-Simple is meant as a very lightweight format that developers and users can quickly and easily add to their existing feeds with little effort. GeoRSS Simple supports basic geometries (point, line, box, polygon) and covers the typical use cases when encoding locations.
- GeoRSS GML is a formal OGC Geography Markup Language (GML) Profile, and supports a greater range of features, notably coordinate reference systems other than WGS-84 latitude/longitude.

Both formats are designed for use with Atom 1.0, RSS 2.0 and RSS 1.0, although it can be used just as easily in non-RSS XML encodings. Below are examples of the two GeoRSS point encoding. Annex B provides detailed examples of both serializations.

6.1 Simple Example

```
<georss:point>45.256 -71.92</georss:point>
```

6.2 GML Example

```
<georss:where>
  <gml:Point>
    <gml:pos>45.256 -71.92</gml:pos>
  </gml:Point>
</georss:where>
```

7. GeoRSS Encoding Rules

7.1 GeoRSS Abstract Model

Figure 1 shows the UML model for GeoRSS.

The left side of Figure 1 represents GeoRSS, the right side represents the "external" content that GeoRSS is being used to describe. In the model, **where** is an association of a **geometry** to some **content**. GeoRSS places no constraint on the type of content, nor on its format.
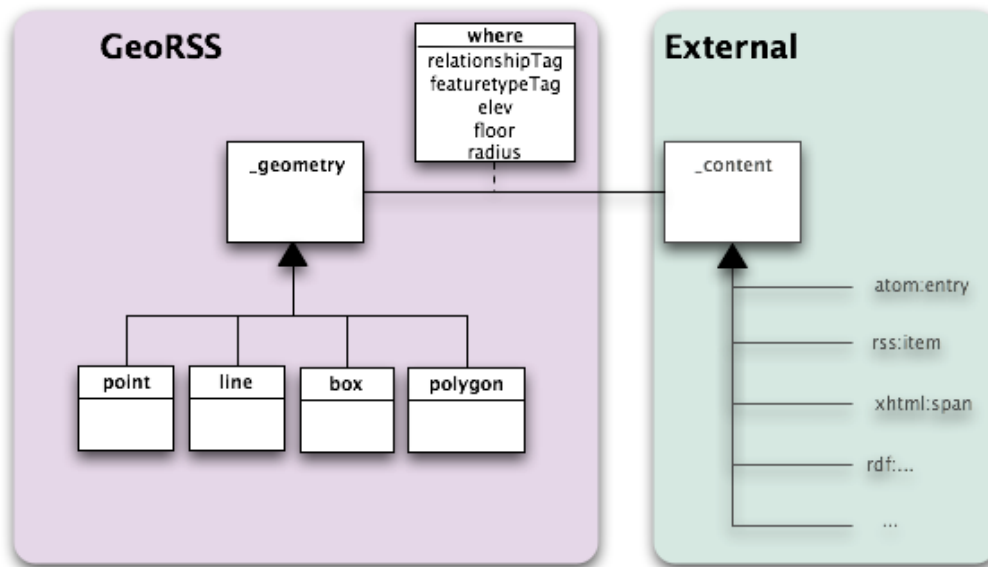

Figure 1: General GeoRSS Information Model (Source – GeoRSS web site)

The model is specified as an abstract concept. In order to use the model, it must be expressed in a concrete form such as XML, RDF, etc. In GeoRSS, these expressions are called **serializations**.

7.2 GeoRSS and Coordinate Reference Systems (CRS)

7.2.1 Default CRS for a GeoRSS Encoding

The default CRS for any GeoRSS encoding is as defined by EPSG:4326 (https://www.epsg-registry.org/ and enter code 4326). The OGC WKT for the WGS 84, GeodeticCRS (geographic 2D) is:

```
GEODCRS["WGS 84",
  DATUM["World Geodetic System 1984",
    ELLIPSOID["WGS 84",6378137,298.257223563,LENGTHUNIT["metre",1.0]]],
  CS[ellipsoidal,2],
    AXIS["latitude",north,ORDER[1]],
    AXIS["longitude",east,ORDER[2]],
    ANGLEUNIT["degree",0.01745329252],
  ID["EPSG",4326]]
```

For the default CRS, all coordinates *SHALL* be specified in decimal degrees with axis order Latitude/Longitude. The values of latitude and longitude *SHALL* be bounded by ±90 and ±180 respectively. Positive latitudes are north of the equator, negative latitudes are south of the equator. Positive longitudes are east of the Prime Meridian; negative longitudes are west of the Prime Meridian.

7.2.2 Specifying a CRS other than WGS 84 GeodeticCRS (geographic 2D)

A considerable amount of geographic data is not provided in latitude/longitude coordinates. In the United States, UTM or State Plane are very common CRSs for geographic data. In other countries, other CRS are in common use. The reason for this is that Latitude/Longitude coordinates are not ideal for small scale mapping and surveying applications. These are application where inches or centimeter accuracy is highly important. Think positioning of a well head for an oil well, designing and building roads and bridges, and measuring your property boundaries. Therefore, GeoRSS GML has the ability to support other CRSs in addition to Latitude/Longitude.

If your GeoRSS GML data is in a coordinate reference system other than lat/lon WGS84, add in the *srsName* attribute to your geometry (see examples below).

7.2.2.1 3D decimal degrees CRS example

The key aspect of clearly specifying another CRS is to specify the correct reference code for the registry being referenced. For this example, use CRS epsg:4979, then specify the srsDimension attribute in the encoding, and include a third number in your coordinate

tuple. In the example below, the urn states that a CRS definition with a code 4979 can be found in version 9.0 of the EPSG registry[1].

```
<entry>
   ...
   <georss:where>
      <gml:Point srsName="urn:ogc:def:crs:EPSG:9.0:4979"
srsDimension="3">
         <gml:pos>42.3453 -156.2342 45</gml:pos>
      </gml:Point>
   </georss:where>
</entry>
```

7.2.2.2 State Plane CRS example

26986 is the EPSG State Plane system for Massachusetts Mainland[2].

```
<entry>
   ...
   <georss:where>
      <gml:Polygon srsName="urn:ogc:def:crs:EPSG:9.0:26986">
         <gml:exterior>
            <gml:LinearRing>
               <gml:posList>
               45.256 -110.45 46.46 -109.48 43.84 -109.86 45.256 -110.45
               </gml:posList>
            </gml:LinearRing>
         </gml:exterior>
      </gml:Polygon>
   </georss:where>
</entry>
```

7.3 GeoRSS Simple Serialization

The Simple serialization of GeoRSS is designed to be maximally concise, in both representation and conception. Each of the four GeoRSS objects require only a single tag.

This simplicity comes at the cost of direct upward compatibility with GML. However, it is straightforward to devise transformations from this Simple serialization to the GML serialization through the GML model. For many needs, GeoRSS Simple will be sufficient.

---

[1] http://www.epsg-registry.org/report.htm?type=selection&entity=urn:ogc:def:crs:EPSG::4979&reportDetail=long&style=urn:uuid:report-style:default-with-code&style_name=OGP%20Default%20With%20Code&title=

[2] http://www.epsg-registry.org/report.htm?type=selection&entity=urn:ogc:def:crs:EPSG::26986&reportDetail=long&style=urn:uuid:report-style:default-with-code&style_name=OGP%20Default%20With%20Code&title=

Some publishers and users may prefer to separate latitude - longitude coordinate pairs by a comma rather than whitespace. This is permissible in Simple; GeoRSS parsers should just treat commas as whitespace.

The first example shows GeoRSS Simple within an Atom 1.0 entry. This serialization applies just as well to an RSS 2.0 or RSS 1.0 item; it can also be associated with the entire feed. The rest of the examples show only the encoding of the objects and attributes.

7.3.1 Geometry

7.3.1.1 Point

A point *SHALL* contain a single latitude-longitude pair, separated by whitespace. Below is an example of a GeoRSS point encoding as part of an Atom feed.

```xml
<?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">
   <title>Earthquakes</title>
   <subtitle>International earthquake observation labs</subtitle>
   <link href="http://example.org/"/>
   <updated>2005-12-13T18:30:02Z</updated>
   <author>
      <name>Dr. Thaddeus Remor</name>
      <email>tremor@quakelab.edu</email>
   </author>
   <id>urn:uuid:60a76c80-d399-11d9-b93C-0003939e0af6</id>
   <entry>
      <title>M 3.2, Mona Passage</title>
      <link href="http://example.org/2005/09/09/atom01"/>
      <id>urn:uuid:1225c695-cfb8-4ebb-aaaa-80da344efa6a</id>
      <updated>2005-08-17T07:02:32Z</updated>
      <summary>We just had a big one.</summary>
      <georss:point>45.256 -71.92</georss:point>
      </entry>
</feed>
```

7.3.1.2 Line

A line *SHALL* contain a space separated list of latitude-longitude pairs, with each pair separated by whitespace. There *SHALL* be at least two pairs.

```xml
<georss:line>45.256 -110.45 46.46 -109.48 43.84 -
109.86</georss:line>
```

7.3.1.3 Polygon

A polygon *SHALL* contain a space separated list of latitude-longitude pairs, with each pair separated by whitespace. There *SHALL* be at least four coordinate pairs, with the last being identical to the first (so a polygon has a minimum of three actual points).

```
<georss:polygon>
     45.256 -110.45 46.46 -109.48 43.84 -109.86 45.256 -110.45
</georss:polygon>
```

7.3.1.4 Box

A bounding box is a rectangular region, often used to define the extents of a map or a rough area of interest. A box *SHALL* contain two space separate latitude-longitude coordinate pairs, with each pair separated by whitespace. The first pair defines the lower left corner of the box and the second pair defines the upper right corner of the box.


```
<georss:box>42.943 -71.032 43.039 -69.856</georss:box>
```

7.3.2 Additional Properties

7.3.2.1 Feature Type, Feature Name, and Relationship Tags

The Feature Type, Feature Name, and Relationship tags are specified as GeoRSS elements.

```
<georss:point>45.256 -110.45</georss:point>
<georss:featuretypetag>city</georss:featuretypetag>
<georss:relationshiptag>is-centered-at</georss:relationshiptag>
<georss:featurename>Podunk</georss:featurename>
```

7.3.2.2 Elevation

Elevation, specified in GeoRSS elements, can be expressed as "elev" or "floor". "elev" is meant to contain "common" GPS elevation readings, i.e. height in meters referenced to the WGS84 ellipsoid. This is a reading that should be easy to get from any GPS device. "floor" is meant to contain the floor number of a building. In some countries the numbering is different than in other countries, but since we'll know the location of the building, it should be fairly unambiguous. The definition of "floor" as used in GeoRSS is consistent with the IETF[3] and NENA definitions as required for the Next Generation 911 system in the US.

---

[3] IETF RFC 5774: Considerations for Civic Addresses in the Presence Information Data Format Location Object (PIDF-LO)

```
   <georss:point>45.256 -110.45</georss:point>
   <georss:elev>313</georss:elev>

   <georss:point>45.256 -110.45</georss:point>
   <georss:floor>2</georss:floor>
```

7.3.2.3 Radius

"radius" indicates the size in meters of a radius or buffer around the geometry object, for example, radius of circular area around a point geometry.

```
   <georss:point>45.256 -110.45</georss:point>
   <georss:radius>500</georss:radius>
```

7.3.3 GeoRSS Application Schema

The application schema defines <georss:where> as the tag that signals geographic content - either in GeoRSS Simple or GML. Section 7.4 provides more detail on the GeoRSS schema.

[Here is the GeoRSS Application Schema](#).

7.4 GeoRSS GML

The OGC/ISO Geography Markup Language (GML) is an XML grammar written in XML Schema for the modelling, transport, and storage of geographic information. GML provides a variety object types for describing geography including features, coordinate reference systems, geometry, topology, time, units of measure and generalized values. A geographic feature is "an abstraction of a real world phenomenon; it is a geographic feature if it is associated with a location relative to the Earth. So a digital representation of the real world can be thought of as a set of features.

The following sections describe the encoding of GeoRSS' objects in a simple [GML version 3.1.1](#) profile. Each section details the construction of the GeoRSS four objects (Point, line, polygon, and box), followed by some informative use cases. As with all GeoRSS encodings, if not specified, the implied coordinate reference system is WGS84 with coordinates specified in decimal degrees.

7.4.1 Point

A point consists of a GML **<Point>** element with a child **<pos>** element. Within **<pos>** the latitude and longitude coordinate values are separated by a space.

```
<?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">
   <title>Earthquakes</title>

   <subtitle>International earthquake observation labs</subtitle>
```

```
    <link href="http://example.org/"/>
    <updated>2005-12-13T18:30:02Z</updated>
    <author>
        <name>Dr. Thaddeus Remor</name>
        <email>tremor@quakelab.edu</email>
    </author>
    <id>urn:uuid:60a76c80-d399-11d9-b93C-0003939e0af6</id>
    <entry>
        <title>M 3.2, Mona Passage</title>
        <link href="http://example.org/2005/09/09/atom01"/>
        <id>urn:uuid:1225c695-cfb8-4ebb-aaaa-80da344efa6a</id>
        <updated>2005-08-17T07:02:32Z</updated>
        <summary>We just had a big one.</summary>

        <georss:where>
            <gml:Point>
                <gml:pos>45.256 -71.92</gml:pos>
            </gml:Point>
        </georss:where>
    </entry>
</feed>
```

7.4.2 Line

A line consists of a GML **<LineString>** element with a child **<posList>** element. Within **<posList>** the coordinates of the points on the line are entered as pairs of latitude and longitude coordinate values, separated by spaces. There must be at least two pairs. No two pairs may be separated by more than 179 degrees in either latitude or longitude.

```
<entry>
    ...
    <georss:where>
        <gml:LineString>
            <gml:posList>
                45.256 -110.45 46.46 -109.48 43.84 -109.86
            </gml:posList>
        </gml:LineString>
    </georss:where>
</entry>
```

7.4.3 Polygon

A polygon consists of a **<Polygon>** element with a child **<exterior>**, **<LinearRing>** and **<posList>** elements. There must be at least four coordinate pairs with the last being identical to the first. (A boundary has a minimum of three actual points.) No two pairs may be separated by more than 179 degrees in either latitude or longitude.

```
<entry>
    ...
    <georss:where>
        <gml:Polygon>
            <gml:exterior>
                <gml:LinearRing>
```

```
            <gml:posList>
         45.256 -110.45 46.46 -109.48 43.84 -109.86 45.256 -110.45
            </gml:posList>
         </gml:LinearRing>
      </gml:exterior>
   </gml:Polygon>
</georss:where>
</entry>
```

<Exterior> specifies this shape as defining the outside of an area, and **<LinearRing>** states that the coordinates should be connected with straight lines. Within **<posList>** the coordinates of the points are entered as pairs of latitude and longitude coordinate values, separated by spaces. There must be at least four pairs with the last being identical to the first. (A polygon has a minimum of three actual points.) No two pairs may be separated by more than 179 degrees in either latitude or longitude.

7.4.4 Box

A bounding box defines a rectangular region. This object is often used to define the extents of a map or define a rough area of interest. A GML box is called an Envelope. It consists of an **<Envelope>** element with a child **<lowerCorner>** element and a child **<upperCorner>** element.


```
<entry>
   ...
   <georss:where>
      <gml:Envelope>
         <gml:lowerCorner>42.943 -71.032</gml:lowerCorner>
         <gml:upperCorner>43.039 -69.856</gml:upperCorner>
      </gml:Envelope>
   </georss:where>

</entry>
```

Annex A: Conformance Class Abstract Test Suite (Normative)

Not applicable

Annex B: Example One – Boat Trip

<u>Boat trip / hike (Atom)</u>

A good way to describe a trip that has many places of interest like a boat trip or a hike is to specify the overall trip's path with a line as a child of the <feed>. Then mark each location of interest with a point in the <entry>.

```xml
<feed xmlns="http://www.w3.org/2005/Atom"
      xmlns:georss="http://www.georss.org/georss"
      xmlns:gml="http://www.opengis.net/gml">
  <title>Dino's Mt. Washington trip</title>
  <link href="http://www.myisp.com/dbv/"/>
  <updated>2005-12-13T18:30:02Z</updated>

  <author>
     <name>Dino Bravo</name>
     <email>dbv@example.org</email>
  </author>
  <id>http://www.myisp.com/dbv/</id>

  <georss:where>
     <gml:LineString>
        <gml:posList>
           44.2706 -71.3033 44.279 -71.2872 44.274 --71.2997 44.27 -71.302
        </gml:posList>
     </gml:LineString>
  </georss:where>
  <entry>
     <title>Setting off</title>
     <link href="http://www.myisp.com/dbv/1"/>
     <id>http://www.myisp.com/dbv/1</id>
     <updated>2005-08-17T07:02:32Z</updated>
     <content>getting ready to take the mountain!</content>
     <georss:where>
        <gml:Point>
           <gml:pos> 44.2706 -71.3033</gml:pos>
        </gml:Point>
     </georss:where>
  </entry>
  <entry>
     <title>Crossing Cutler River</title>
     <link href="http://www.myisp.com/dbv/2"/>
     <id>http://www.myisp.com/dbv/2</id>
     <updated>2005-08-15T07:02:32Z</updated>
     <content>Check out the salamanders here</content>

     <georss:where>
        <gml:Point>
           <gml:pos>44.263 -71.2757</gml:pos>
        </gml:Point>
     </georss:where>
  </entry>
</feed>
```

Annex C: Example two – Calendar of events (Atom).

```xml
<feed xmlns="http://www.w3.org/2005/Atom"
      xmlns:georss="http://www.georss.org/georss"
      xmlns:gml="http://www.opengis.net/gml">
  <title>Cambridge Calendar of Events</title>

  <subtitle>Goings on around Cambridge</subtitle>
  <link href="http://example.org/"/>
  <updated>2005-12-13T18:30:02Z</updated>
  <author>
     <name>Arty "the one man party" Collins</name>
     <email>apcollins@cambridgema.gov</email>
  </author>
  <id>http://www.cambridgema.gov/calendar</id>
  <georss:where>
     <gml:Polygon>
        <gml:exterior><gml:LinearRing>
            44.256 -71.16 44.46 -71.48 44.84 -71.86
            44.8 -71.2 44.256 -71.16
        </gml:LinearRing></gml:exterior>
     </gml:Polygon>
  </georss:where>

  <entry>
     <title>City Council Meeting</title>
     <link href="http://www.cambridgema.gov/calendar/2005/09/09/evt01"/>
     <id>http://www.cambridgema.gov/calendar/2005/09/09/evt01</id>
     <updated>2005-08-17T07:02:32Z</updated>

     <content>Regular weekly meeting</content>
     <georss:where>
        <gml:Point>
           <gml:pos>42.372 -71.119</gml:pos>
        </gml:Point>
     </georss:where>
  </entry>
  <entry>
     <title>NoCa Arts Open House</title>
     <link href="http://www.cambridgema.gov/calendar/2005/09/12/evt02"/>
     <id>http://www.cambridgema.gov/calendar/2005/09/12/evt02</id>
     <updated>2005-08-15T07:02:32Z</updated>
     <content>North Cambridge arts festival featuring local artists</content>

     <georss:where>
        <gml:Point>
           <gml:pos>42.395 -71.3953</gml:pos>
        </gml:Point>
     </georss:where>

  </entry>
</feed>
```
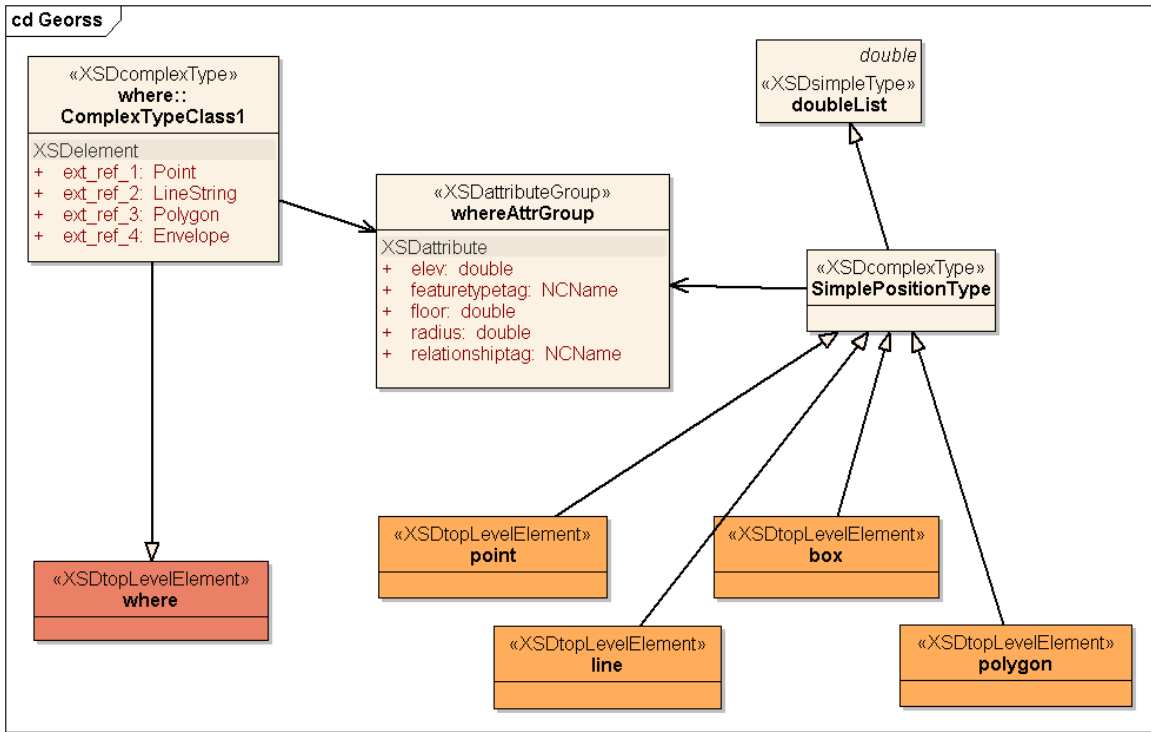
Annex D: GeoRSS Application Schema

Annex E: Additional notes on using Atom

Location is specified using Atom 1.0's official extension mechanism (see Extending Atom). We add a **<georss:where>** element as a child of the **<entry>** element. Within **<georss:where>** you may specify geography. We recommend that the GML encoding is used as follows:

- point **<gml:Point>**
- line **<gml:LineString>**
- polygon **<gml:Polygon>**
- box **<gml:Envelope>**

GeoRSS Simple could also be used:

- point **<georss:point>**
- line **<georss:line>**
- polygon **<georss:polygon>**
- box **<georss:box>**

Atom processing software is required to either read or ignore this type of "foreign markup"--not fail, so there is no danger of breaking someone's RSS reader (or publisher) by including this <where> element in your feed.

NOTE: GML allows you to specify a coordinate reference system (CRS) other than WGS84 decimal degrees (think lat/long). If you have a need to express geography in a CRS other than this, we recommend that you specify the geographic object multiple times, one in WGS84 and the others in your other desired CRS's.

Annex F: Revision history

Date	Release	Author	Paragraph modified	Description
1/30/2017	1.0	C. Reed		Put GeoRSS into OGC document template
4/26/2017	1.0r1	C. Reed	Various	Respond to public comments