europeana
connect

Milestone 3.3.2 – KML Specification

This Document (M3.3.2) specifies a subset of KML for display in the Europeana.4D (WP3.3) interface. Furthermore several recommendations and requirements for the handling of KML from external sources, that need to be fulfilled by the implementation.

Please note that this milestone was renamed.

co-funded by the European Union

The project is co-funded by the European Union, through the ***e*Content*plus*** programme

http://ec.europa.eu/econtentplus

Österreichische Nationalbibliothek

EuropeanaConnect is coordinated by the Austrian National Library

Distribution

Version	Date of sending	Name	Role in project
0.3	15.10.2010	UGOE – Christian Mahnke	Task 3.3 developer
0.4	10.12.2010	UGOE – Christian Mahnke	Task 3.3 developer
1.0	15.12.2010	Liferay	

Approval

Version	Date of approval	Name	Role in project
1.0	30.11.2010	Vassilis Tzouvaras	Developer WP1

Revisions

Version	Status	Author	Date	Changes
0.1	Draft	UGOE – Christian Mahnke	01.07.2010	Initial Version
0.2	Draft	UGOE – Christian Mahnke	15.07.2010	• Added Example XML fragments • Added Footnotes and links
0.3	Draft	UGOE – Christian Mahnke	13.10.2010	Final draft
0.4	Draft	UGOE – Christian Mahnke	09.12.2010	Added a section describing the requirements to be fulfilled by the Europeana search Engine. (based on comments by Vassilis Tzouvaras)
1.0	Final	ONBV – VPZ	15.12.2010	Minor layout changes

Table of Contents

Introduction

To enhance the reusability of the Europeana.4D implementation we looked for a well known and broadly accepted standard for data interchange. We've chosen KML, which was created by Keyhole, which was acquired later by Google. It's the basis for all geo related products by Google. In 2007 version 2.2 became a standard recognized by the Open Geospatial Consortium[1].

This decision was accepted during the last ASM in Berlin. Thus the milestone 3.3.2 was renamed to "KML Specification".

This document describes the different needed tags, optional tags and several format related requirement that the implementation has to fulfil. Please note that this specification only defines a whitelist of elements that the implementation of Europeana.4D should recognize. For the handling of any other KML (including extension namespaces) please refer to the Section "Requirements". In general the elements in this specification only describes elements required by Europeana.4D, nothing more.

Since this specification describes only a subset of KML a full description is omitted. Please refer to the official specification (OGC Number 07-147r2[2]). **Please note that the official document is normative.** Every confusing or misleading part should be addressed in the scope of this document. References to the original specification are included; XML elements are referenced with the namespace prefix(es) used in the original specification(s), they are omitted in the examples for reasons of readability.

Some features may in future rely on the Google KML extensions[3]. This may affect mainly the handling of connection data. Currently these are encoded in a `kml:lineString` element, which isn't completely in the range of the original KML specification. Using the more appropriate `gx:Track` element and it's children certainly would impose restrictions if third party tools except Google Earth are used.

This and other uses of this extension namespace are omitted for this reason.

Europeana.4D KML Profile

Root element and basic structure

The basic structure of a KML file is described in detail in the official specification. It basically consists of a root element (`kml:kml` element) and can host several grouping elements inside. The root element is described in section 7 of the KML specification. The simplest form is a list of place marks (`kml:Placemark` element), these should be grouped using a `kml:Document` (see below).

[1] Open Geospatial Consortium : http://www.opengeospatial.org/

[2] OGC 07-147r2: http://portal.opengeospatial.org/files/?artifact_id=27810

[3] Google KML extension namespace specification: http://code.google.com/apis/kml/documentation/kmlreference.html#kmlextensions

Content elements

The KML specification defines several XML abstract data types for content elements. This way it's either possible to just define a list of `kml:Placemark` elements below the `kml:kml` top level element or to use several elements to group points (see below). The model of KML is described in section 6 of the specification.

Groups of data points

Multiple data points may be grouped into one parent element. This element is `kml:Document`. Grouped data points are also loadable in tools like Google Earth. It's also possible to omit this element but this may break compatibility with third party tools. Extended Grouping functionality like `kml:Folder` isn't supported.

Structure of a data point

A single point on the map and / or time line is represented by a `kml:Placemark` element. This container groups several sub elements, which are described in the following section. Please note that the usage of these elements inside Europeana.4D is described here.

Example – The Structure of a very simple kml:Placemark

```
<Placemark>
     <name />
     <description />
     <Point/>
     <TimeStamp />
     <Link id="ID" />
</Placemark>
```

General information about time

A single point in time

In general time is encoded according to the `kml:dateTimeType` type as described in 16.10 of the KML specification.

A single point in time is represented by the `kml:TimeStamp` element, in it's simplest form it only contains a `kml:when` child element.

Example – A simple point in Time

```
<TimeStamp>
     <when>1771</when>
</TimeStamp>
```

A time span

A timespan is represented by the `kml:TimeSpan` Element (15.2). This Element contains zero or one of the following two child elements: `kml:begin` and `kml:end`. Additional possible children

Example - A time span

```
<TimeSpan>
     <begin>1771</begin>
     <end>1779</end>
```

```
</TimeSpan>
```

Implementation note:

The current implementation uses the contents of the begin element as the point in time for display, since another feature already uses a similar visualisation. To enable the selection of time ranges, the display of ranges inside the timeline is shortened to a single point in time.

Encoding of uncertain / partial dates.

In KML dates can be encoded in any format as defined by XML Schema[4]. These data types (`xsd:gYear, xsd:gYearMonth, xsd:date, xsd:dateTime`) can be used to represent partial and uncertain dates.

Implementation note:

The granularity of display of partial dates depends of the rage of date in the whole data set.

General information about places

Simple coordinates (points)

The simplest possible for of a geo encoded data point is a single point on a map. This point is represented by the `kml:Point` element and it's child element `kml:coordinates`

Example – A singe point

```
<Point>
     <coordinates>12.35,51.3,0</coordinates>
</Point>
```

Encoding of uncertain places / areas

Areas can be used to describe uncertain places. Areas can be represented in KML by the `kml:Polygon` (which is a sub type of the abstract `kml:Geometry`) element.

Implementation note:

The implementation is not able to display areas, since there are two other features that are using the implied visual semantics. The first feature is the selection mode, it lets user create custom selections of multiple data points. The second feature is the aggregation function, which uses big circles to represent more then one point. Both features would provoke a "visual clash" with areas, since it would be hard to distinguish between the different meanings of regions on the map. To overcome these limitations the implementations picks the first point of a area definition.

Encoding constructed place names

The KML specification has several fields for the encoding of place names. We've chosen the `kml:address` element as container for place names that are reconstructed from the coordinates.

Example – A constructed Place name

```
<kml:address>Leipzig</kml:address>
```

Implementation note:

[4] http://www.w3.org/TR/xmlschema-2/#duration

Constructed place names are used to represent the names of places for creation of aggregated data points. The name of the points is used to identify several places with slightly different coordinates. This might happen if the coordinates are from different sources (gazetteers) or reconstructed for example from a area definition.

They currently encoded by using the `kml:address` element, please note that this approach isn't completely identical to semantic meaning of the original specification

Internationalisation issues

A common problem for the display of place names in the locale of a specific user is the fact that one place can have different names in different languages. To overcome this problem it's possible to use a `xml:lang`[5] attribute for the different names in future implementations.

Implementation note:

Since there currently isn't any metadata record holding this information and third party implementations (like Google Earth) support this, the language switch isn't part of the prototype. Another approach can be the usage of an `kml:ExtentedData` element.

Descriptions and other metadata (for display)

Name

Each data point can have a single `kml:name`, this should be a human understandable (as opposed to an abstract identifier) short name. It may be the title of a work.

Example – The `kml:name` element

```
<name>Die Mitschuldigen</name>
```

Snippet

It's possible to define a short textual preview of an object using the `kml:Snippet` element, this element also requires the attribute `maxLines`, which can be used do limit the length of the displayed text.

Example – A description snippet

```
<Snippet maxLines="1">
     This fragment may contain title, creator and source
institution.
</Snippet>
```

Implementation note

The current implementation can't handle rich content inside a `kml:Snippet` element using a CData section. This shouldn't be a problem since several implementations (like Google Earth) have these limitations as well.

Description

The `kml:description` may hold a more elaborate description of the data point. This may be a simple text or rich text encoded in (X)HTML mark up..

[5] http://www.w3.org/TR/REC-xml/#sec-lang-tag

Please note that it's possible to include a so-called XML CData (character data) section inside the description field to use HTML formatting tags for formatting purposes. Another use of this technique is to include references to images and links this way.

Example – A "complex" description:

```
<description><![CDATA[
     <a href='http://flickr.com/photos/33917831@N00/4297186102'
title='Giant Evil Clown Head' target='_blank'>
          <img alt='Giant Evil Clown Head'
     src='http://farm3.static.flickr.com/2740/4297186102_314dce
     bbf8_t.jpg' />
      </a>
]]></description>
```

Links

Links should be included inside the narrative description (see `kml:description` above) of a singe data point using a simple `html:a` element.

Implementation note:

The `kml:Link` element can be used to reference external KML files, but since the is currently no use case for this feature, it isn't supported by Europeana.4D.

Icons and small preview images

KML allows the use small preview images or icons for each `kml:Placemark` element (as part of the `kml:Icon` element). Additionally `html:img` elements can be used inside the description (see `kml:description` above) of a data point.

Example

See the example for a complex description.

Implementation note:

Additional icons are currently only supported by the part of the Open Layers Library (map functionality), not by the data table.

Advanced features

Connections and Trails

Connections and Trails can be represented with the `kml:LineString` element.

They could also have been represented by the help of the Google KML namespace extension (see introduction), but for compatibility this approach wasn't taken.

The `kml:tessellate` element is used to pin the selection to the ground level, it's included for reference only.

Example – A connection (simplified)

```
<LineString>
     <tessellate>1</tessellate>
     <coordinates>
```

```
          1.949057931391283,47.50867638651865,0
2.089607152800692,47.84437381872774,0
     </coordinates>
</LineString>
```

Implementation note:

Since connections are created during run time, depending on the user input, they can't be loaded. *They are only serializable.*

Selections

Selections may be encoded by geometric shapes like circles or polygons. In technical terms all shapes are represented using a `kml:Polygon` element (and its children, see example below). The `kml:tessellate` element is used to pin the selection to the ground level, it's included for reference only.

Example – A selection represented as polygons (simplified)

```
<Polygon>
     <tessellate>1</tessellate>
     <outerBoundaryIs>
          <LinearRing>
               <coordinates>
                    21.12674752439695,50.45904731364891,0
21.07456391351959,50.34026176935406,0
8.214235333732416,49.17002918741375,0
               </coordinates>
          </LinearRing>
     </outerBoundaryIs>
</Polygon>
```

Implementation note:

Since selections are created on runtime of the application they are currently not loadable. *They are only serializable.*

Requirements

This section describes the requirement to be fulfilled by the implementation of the Europeana.4D interface on the one hand and by the Europeana search engine on the other.

Behaviour of the implementation

Encoding of the KML files

The implementation expects UTF-8 as file encoding for the KML files. The actual behaviour of the implementation relies on the JavaScript engine of the browser. Using UTF-8 a wide rage of characters can be supported

Unknown elements and attributes

The implementation should ignore elements and attributes that aren't covered in this specification. For example data exported from Google Earth include `kml:StyleMap`,

`kml:Style` and `kml:LookAt` elements and their children. This information's aren't displayable inside the Europeana.4D interface.

Please note that the current (as of October 2010) prototype might not meet this requirement yet.

Missing elements and attributes

Where possible the implementation should be fault tolerant, this also includes the handling of missing, malformed or plain wrong data. Whenever possible the user should be able to work with the dataset but also be aware that there might be problems. Where possible warnings should be handled as in other implementations like Google Earth, which silently ignores unknown extension namespaces for example.

User generated Data

The Implementation features a tool called "Manual Metadata editing interface" (Europeana.Connect M3.3.4). This Tool can be used to generate KML fragments (see `kml:Placemark`) of KML files including points in time, points on the map and descriptions of both. These fragments can be aggregated into a singe KML file (see `kml:Document`).

Serialisation requirements

The implementation is able to serialise data point created at runtime, this includes points in time, places, connections and selections. These different features are encoded as KML as well. The used elements are described in the section "Advanced features".

Behaviour of the Europeana search engine

To be able to use the Europeana.4D implementation the search engine needs to generate KML files according to this specification. As discussed during the All Staff Meeting in Berlin this will be done by adding a KML template to the templating engine.

The enrichment of the Europeana metadata is out of scope for this document.

Short reference

Element name	Short description	KML Specification
kml:Placemark	Parent element for a single data point.	9.11
kml:name	A short name of a single data point.	9.1.3.1
kml:description	A longer description of a data point.	9.1.3.10
kml:Snippet	Element to represent preview text for a data point.	16.19
kml:Point	Parent element for a definition of a place.	10.3
kml:coordinates	Coordinates of a point.	10.5.3.4
kml:TimeStamp	Parent element of a point in time.	15.3
kml:when	Element to represent a single point in time.	15.3.3.1
kml:TimeSpan	Parent element for a time span.	15.2
kml:begin	Element to represent the start date of a time stamp.	15.2.3.1

kml:end	Element to represent the end date of a time stamp.	15.2.3.2
kml:Document	Parent element for multiple place marks.	9.7
kml:tesselleta	This elements describes the position relative to the ground level.	10.6
kml:Polygon	Parent Element for polygons.	10.8