Advances in Web-based GIS, Mapping Services and Applications – Li, Dragićević & Veenendaal (eds)
© 2011 Taylor & Francis Group, London, ISBN 978-0-415-80483-7

Vector data formats in internet based geoservices

Franz-Josef Behr
Faculty of Geomatics, Computer Science and Mathematics, Stuttgart University of Applied Sciences, Stuttgart, Germany

Kai Holschuh
Marketing and Intercultural Management, Karlshochschule International University, Karlsruhe, Germany

Detlev Wagner
Faculty of Geomatics, Computer Science and Mathematics, Stuttgart University of Applied Sciences, Stuttgart, Germany

Rita Zlotnikova
Blom Romania, I.H. Radulescu, Jud Dambovita, Romania

ABSTRACT: A diverse range of formats and standards are used to exchange and present 2D location awareness information on the internet. The emphasis of this chapter is on the eXtensible Markup Language (XML), the basis of many (web) geographic information system (GIS) standards and services. The background, structure and elements of the related languages are introduced, and their relationships to new tools, application programming interfaces (APIs) and associated areas of application are also described. Throughout the text examples are used to explain data interchange formats. The illustrations of code are mostly in XML, the rest in WKT, JSON and GeoJSON. The chapter concludes with a summary of usefulness of the different standards to data interoperability on the internet.

Keywords: Internet, web GIS, interoperability, data structures, standards, spatial infrastructure, representation, format

1 INTRODUCTION

In recent years, interest in distributed web services has increased within information technology circles. Internet-based geospatial applications are an integral element of the Web 2.0 developments. Web-based geographic information systems (Web GIS) provide the navigational tools on which many distributed web services are based and enhance the appeal and usefulness of countless others.

The aim of this chapter is to give an overview of the diverse formats and standards used to exchange and present 2D location awareness information on the Internet. Emphasis is on the eXtensible Markup Language (XML), the basis of many Web GIS standards. The background, structure and elements of these languages are introduced, and their relationship to new tools, application programming interfaces (APIs) and associated areas of application are also described.

While XML plays a leading role in the creation of geospatial web services today, especially in the establishment of open geospatial standards, such as the Web Map Service (WMS) and the Web Feature Service (WFS) of the Open Geospatial Consortium (OGC), other related standards and formats have also been authorized by the World Wide Web Consortium (W3C, http://w3.org/) and are regularly used in Web GIS. The definitions and applications

Li ISPRS.indb 349 4/12/2011 4:19:32 PM

of these meta-languages, including XML Schema Definitions (XSD), XML Namespaces, XML Linking Language (XLink), XML Pointer, Extensible Stylesheet Language with XSL Transformations, XML Path Language and XSL Formatting Objects, in combination with Cascading Stylesheets (CSS), are reviewed in this chapter as well. Moreover, since the combination of Extensible Hypertext Markup Language (XHTML) with complementary technologies, such as Scalable Vector Graphics (SVG), has become an essential part of Web GIS as well, it is also covered.

Throughout the text examples are used to explain data interchange formats. The illustrations of programming code are mostly in XML, the rest in WKT, JSON and GeoJSON. The chapter concludes with a summary of usefulness of the different standards to present geographic information on the Internet.

2 EXTENSIBLE MARKUP LANGUAGE

The eXtensible Markup Language (XML Bray et al. 2006, Bray et al. 2008) is fundamental to geoinformatics. It is used to define data objects, structure data, provide metadata, assign attributes and other information, as well as store configuration parameters.

2.1 *Introduction to XML elements and objects*

Before considering data objects, let us look at the parts that make up an XML element. All types of alphanumerical data can be identified and put together as an *element* which is simply a way to structure information within a pre-defined namespace. Figure 1 shows the basic three-part structure of an XML element: *start tag*, *element content* and *end tag*. Together, the three parts form a "nested" structure. The element name appears in both tags, in the latter preceded by a slash character. These two tags bracket the element content. The content can be either empty, contain text, and / or contain other XML elements.

For the sake of brevity, an empty element may be written in the following abbreviated form:

```
<name />
```

Element content, however, is subject to some restrictions. It may not contain certain characters reserved for the markup language, such as < or &. These must then be coded as *entities*, for example, `<` instead of <.

In addition to being part of the element content, information may also be provided in the form of attributes in the start tag. Here is an example:

```
<text x="15" y="135">AbcDef</text>
```

Some XML-based languages, such as Scalable Vector Graphics (SVG, explained in section 3.8.2) often include information as attributes in start tags.

Attributes are made up of pairs of keys and values (KVPs) separated by an equal sign. Keys are alphanumeric symbols (letters and numerals) chosen to label aspects of the

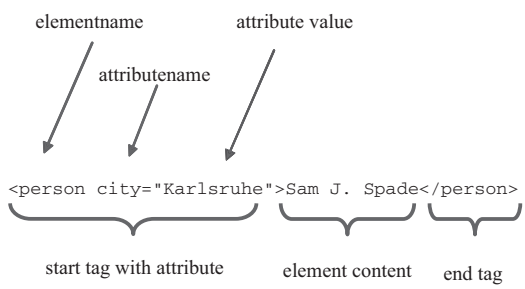

Figure 1. Basic structure of an XML element.

Li ISPRS.indb 350 4/12/2011 4:19:32 PM

properties of an element, and values are the specific names or numbers the keys represent. Values must be embedded in single or double quotation marks. In Figure 1, `"city"` is the name of an attribute corresponding to the element `"person"` and with the attribute value `"Karlsruhe"`.

Another important feature of XML is that one element may itself contain text and other elements. This characteristic enables elements to support additional and more interesting tasks, but it also increases the complexity of revising and correcting compositions. Hence, it is essential that so-called "nesting" be carried out methodically. Inconsistency in structuring elements not only leads to errors at inception, but inhibits the ability to process the XML instance later.

Elements are principal building blocks of XML data objects. The element that encloses the other elements used to document a data object is called the *root element*. In other words, the code which introduces and finalizes a particular document is actually just one element that surrounds all the other elements in a document.

XML objects *per se* are difficult to define in a concise manner. Comprehension requires familiarity with the nuances of their formation and application. Therefore it is all the more important to gain an understanding of the basic aspects of XML data objects in order to begin to understand their usage in geodata interchange.

In general, when an XML data object is coded according to the conventions used to define elements as indicated above, it is considered "well-formed" and called an *XML document*. More specifically, an XML object consists of:

- a prolog, the so called XML declaration;
- a root element and embedded child elements;
- a linked or embedded Document Type Declaration (DTD) or, alternatively, a linked XML Schema defining the elements, their attributes and hierarchical structure (these are optional and not a prerequisite of an XML object; see section 2.4);
- processing instructions with information on programs used to process the XML object (optional);
- further comments (optional);
- CDATA sections, containing blocks of information, such as JavaScript code, hidden from XML parsers (optional).

The lines of code shown below illustrate the various parts of an XML object. In line 1 the XML declaration (prolog) is given. This indicates the version of XML employed, here 1.0, and the specific character encoding used, in this case the ISO-8859-1 codification of the Latin alphabet. The declaration may also indicate if the document can be considered "stand alone", i.e. is valid without further declarations.

```
 1: <?xml version="1.0" encoding="ISO-8859-1" standalone="yes"?>
 2: <?xml-stylesheet href="point.xsl" type="text/xsl" ?>
 3: <!DOCTYPE point [
 4: <!ELEMENT point (easting, northing, d)>
 5: <!ELEMENT easting (#PCDATA)>
 6: <!ELEMENT northing (#PCDATA)>
 7: <!ELEMENT id (#PCDATA)>
 8: ]>
 9: <point>
10:   <easting>35.0</easting>
11:   <northing>54.0</northing>
12:   <id>1</id>
13: </point>
```

In lines 3 through 8 the *Document Type Declaration* is defined, indicating how an XML instance of a point element is laid out. In the example, the association to the root element, as defined by the contents of elements easting, northing, and id, is given. Here, these represent

Li ISPRS.indb 351 4/12/2011 4:19:32 PM

geographic coordinates, and identify the element with a unique id. In terms of the syntax used here, they are PCDATA, i.e. character data[1].

Lines 9 through 13 contain the root element, indicating a specific geographic spot. The contents of this element, as specified in line 4, are the child elements containing the actual values of the geographic coordinates and the id.

Instructions on how to process information (*processing instruction*, given in line 2) bring additional programs into play. Such applications can handle tasks involving the style of graphical presentation or to execute an XSL Transformation (XSLT, to be discussed below).

An important part of programming and markup languages are the comments written in source code. These are ignored by a parser. In XML-based languages, a comment may look like the following:

```
<!-- This is a comment -->
```

Any alphanumeric character or other symbol may be employed between the divergent tags (`<--` and `-->`) that frame a comment except the string double-hyphen (`"--"`).

2.2 *XML related languages and concepts*

The term XML stands for a large group of different markup languages and concepts, the vast majority developed or coordinated by the World Wide Web Consortium. Of these, the most important concepts and standards used in the field of geoinformatics are:

- XML Namespaces,
- Document Object Model (DOM; World Wide Web Consortium 2005),
- Document Type Declaration and XML Schema,
- XLink – XML Linking Language and XML Pointer,
- Scalable Vector Graphics (SVG),
- Cascading Stylesheets (CSS) and the Extensible Stylesheet group of languages.

XML is the basis of additional standards like the Geography Markup Language and other specific application extensions used in geoinformatics. XML is also employed in communication interfaces for geo-web services, such as Web Map Service (Open Geospatial Consortium 2004), Web Feature Service (Open Geospatial Consortium 2002) and Coordinate Transformation Service (Open Geospatial Consortium 2001).

2.3 *XML namespaces*

In general, the designation of element names in XML is unrestricted, allowing multiple ambiguous uses of the same name. For instance, the element "title" could be used to identify persons within an organization as well as indicate songs on a CD recording.

XML namespaces (http://www.w3.org/TR/REC-xml-names/) provide the means to specify element and attribute names in a simple and unequivocal manner. Names are connected to namespaces (vocabularies) by a prefix through unique Internationalized Resource Identifier (IRI) references[2]. Thereby, even if they belong to different schemas, through the definition and use of prefixes, elements can have *qualified names* and can be used simultaneously. For example, the namespaces `"tkfd"`, `"gml"` and `"xsi"`, indicated below, are given for the TKFD root element[3] and following elements. The URI reference "http://www.lv-bw.de/tkfd" identifies the namespace tkfd which, in turn, connects to the elements `"TrainStation"`, `"objectType"` and `"id"`. Similarly, the elements `"centerOf"`, `"Point"` and `"pos"` are derived from the namespace `gml` which is typically used for the Geography Markup Language (discussed in section 3).

---

[1] An acronym derived from "parsed character data."
[2] formerly called Unified Resourxce Identifier (URI).
[3] TKFD: Akronym for Thematische Kartenfachdaten (Thematic Cartographic Data, Graf 2004).

Li ISPRS.indb 352 4/12/2011 4:19:32 PM

```
<?xml version="1.0" encoding="UTF-8"?>
<tkfd:TKFD xmlns:tkfd="http://www.lv-bw.de/tkfd"
  xmlns:gml="http://www.opengis.net/gml"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.lv-bw.de/tkfd">
  <tkfd:TrainStation>
  <tkfd:objectType tkfd:id="EZ00VPK">9201</tkfd:objectType>
    <gml:centerOf>
      <gml:Point>
        <gml:pos>3515955.37 5409276.28</gml:pos>
      </gml:Point>
    </gml:centerOf>
  </tkfd:TrainStation>
</tkfd:TKFD>
```

A namespace without declared namespace prefix can be used as a *default namespace*, associated to all elements without qualifying prefix.

2.4 *Document type declaration and XML schema*

A *Document Type Declaration* (DTD) defines elements according to their attributes and hierarchical structure. DTDs were used in the past in HTML, SVG and GML 1.0, but today this form of element declaration is considered inadequate and is being replaced by the *XML Schema Definitions* (XSD, http://www.w3.org/XML/Schema) which permit significantly more detailed classifications of objects. As compared to DTD however, developing an XML Schema is more complex, principally due to the ability to build on other, previously defined elements.

An XML schema essentially operates on two levels. At the core, the root element of the schema contains namespace declarations and the attribute of the target namespace. This root element is then supported by a structure of geo-objects designated through their own specific elements. Thereby, whether simple and pre-defined or complex and self-defined, data types can be used on the basis of the same well-established concepts as used in object-oriented programming languages.

An XML object, once it is defined by an XML schema, is called an *instance document* or an *XML instance*. An XML instance is usually a file, but can also be a stream of data emitted by a network service or a specific value as indicated in a designated field of a given database. Schema definitions can be found in many kinds of geospatial standards and data models and they play an essential role in all types of services designated by the Open Geospatial Consortium (OGC).

If an XML object is well-formed and complies with the constraints expressed in its DTD or its XML Schema it is considered "valid".

2.4.1 *Data types*

The basic data types used in XML Schema are the same as those used in other programming languages. These data types include string, decimal, those used for date, time, Uniform Resource Locators (URLs), and so on (http://www.w3.org/TR/xmlschema-0/#CreatDt). Elements are then easy to define. For example, an element named Easting can be indicated as having a data type double defined in the namespace xs:

```
<xs:element name="Easting" type="xs:double"/>
```

With *facets* it is possible to create new data types by restricting existing ones. Emphasis is on those aspects which differ from other data types. A new data type called "featureClassIdType", which supports only the integer values 9401, 9402, and 9403, would be documented as follows:

Li ISPRS.indb 353 4/12/2011 4:19:33 PM

```
<xs:simpleType name="featureClassIdType">
  <xs:restriction base="xs:integer">
    <xs:enumeration value="9401"/>
    <xs:enumeration value="9402"/>
    <xs:enumeration value="9403"/>
  </xs:restriction>
</xs:simpleType>
```

Or, as shown below, geometryType can be designated as a string limited to three values:

```
<xs:simpleType name="geometryType">
  <xs:restriction base="xs:string">
    <xs:enumeration value="Point"/>
    <xs:enumeration value="Line"/>
    <xs:enumeration value="Area"/>
  </xs:restriction>
</xs:simpleType>
```

New complex types may be developed based on previously defined types. The data type `"PointType"` below, equivalent to the DTD in section 2.1, has two properties, which describe the location of a point. The third element indicates the `id` used to identify the object:

```
<xs:complexType name="PointType">
  <xs:sequence>
    <xs:element name="Easting" type="xs:double"/>
    <xs:element name="Northing" type="xs:double"/>
    <xs:element name="id" type="xs:nonNegativeInteger"/>
  </xs:sequence>
</xs:complexType>
```

2.4.2 *Deriving types by extension*

In GML it is common that complex data types are defined by extending previously defined data types. Below, the data type `"TrainStationType"` extends `AbstractFeatureType` (obtained from the `gml` namespace) by adding the properties `"stationName"` and `"centerOf"`.

```
<xs:complexType name="TrainStationType">
  <xs:complexContent>
    <xs:extension base="gml:AbstractFeatureType">
      <xs:sequence>
        <xs:element name="stationName" type="xs:string">
        <xs:element ref="gml:centerOf"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

Moreover, it is possible to define the kinds of derivations to be used in instance documents. Similar to polymorphism in object-oriented programming languages, derivated features can be used instead of the original ones, indicated by the substitutionGroup attribute value. Below, first the element `"Station"` with the data type `"StationType"` is declared. Then, according to the attribute `"substitutionGroup"`, this element can be used as a substitute whenever the abstract element `"gml:_Feature"` is used[4]:

```
<xs:element name="TrainStation" type="tkfd:TrainStationType"
substitutionGroup="gml:_Feature"/>
```

[4]The underlined space in `"gml:_Feature"` indicates that the element is abstract.

Li ISPRS.indb 354 4/12/2011 4:19:33 PM

The specification of "BathingPond" below provides a more detailed example of XML Schema. As usual, the first element is made up of two parts: the declaration and the type definition. The declaration includes the name (BathingPond) and data type (BathingPondType). The type definition, also called *content model*, describes the structure of how BathingPond is specified. The value "gml:_Feature" of "substitutionGroup" indicates that occurrences of gml:_Feature can be substituted by BathingPond. Specifically, the data type "tkfd:BathingPondType" is derived from the abstract data type "AbstractFeatureType", which provides general properties for GML features. In this example, "gml:_Feature" is an element of this data type.

```
<xs:element name="BathingPond" type="tkfd:BathingPondType"
substitutionGroup="gml:_Feature"/>
<xs:complexType name="BathingPondType">
  <xs:complexContent>
    <xs:extension base="gml:AbstractFeatureType">
      <xs:sequence>
        <xs:element name="objectType" type="xs:string">
        </xs:element>
        <xs:element name="tkn" type="xs:string">
        </xs:element>
        <xs:element ref="gml:centerOf"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

2.5 *XML linking language and XML pointer language*

XML Linking Language (XLink; http://www.w3.org/XML/Linking) allows elements to be added to XML documents by connecting them to other information resources. XLink does more than attach a clickable hyperlink, as HTML does. In SVG documents, for instance, its href attribute can reference individually defined symbols such as map signatures or coordinates and fix them onto a map as shown in the example:

```
<symbol id="symbol3067" viewBox="0 0 50 50">
  <circle cx="25" cy="25" r="24"/>
</symbol>
<!-- insert the symbol at the position in the map canvas -->
<g transform="translate(145.7, 5.2)
  scale(0.004)"
  <use x="0" y="0" width = "50" height = "50"
    xlink:href="#symbol3067" />
</g>
```

The XLink schema document xlinks.xsd is a fundamental component of GML 3. It allows elements and data types to be defined by indicating references to other declarations where the descriptions of these elements are actually spelled out. When only a specific part of a document needs to be accessed the application of XLink is enhanced by the XML Pointer Language (XPointer; http://www.w3.org/TR/xptr-framework/). In XML Pointer, the symbol "#" initiates the address of the document fragment to be referenced, as shown in the examples provided in section 3.3. Furthermore, it is possible to address single child elements in the document's element hierarchy.

2.6 *Cascading Stylesheets and extensible Stylesheet language*

Cascading Stylesheets (CSS, http://www.w3.org/Style/CSS/) provide a straightforward means of presenting XML documents on output devices such as computer screens. The use of CSS

Li ISPRS.indb 355 4/12/2011 4:19:33 PM

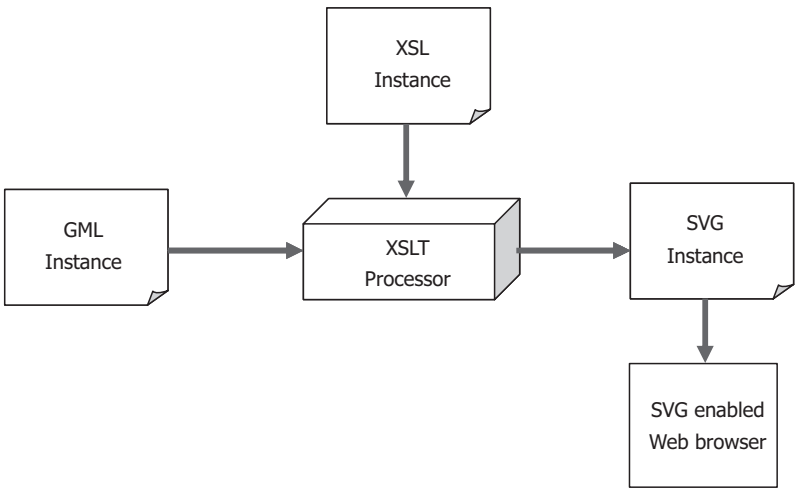

Figure 2. Transformation of geodata encoded on GML into an SVG instance.

enables document presentation to be separated from content, an essential principle in the work with geospatial data. In SVG, for example, CSS enable the presentation style of all aspects and features of an entire geographic layer to be set at once as well as separately from other layers (Watt & Lilley 2001:151). In the following example style properties like line color (stroke), line widths, etc. for the border of a parcel are defined:

```
<path id="parcel114"
  style="fill:none;opacity:1;stroke:#000000;stroke-width:1px;"
  d="M2294.96 1366.67 l -1.88 -46.52 l -29.49 0.63 l -29.75 1.11
  l 2.33 47.64 l 58.78 -2.86"/>
```

The value of the `d` attribute comprises the coordinates of the line stroke to draw on the output device.

As shown in Figure 2, XML documents can be transformed into other XML documents (like SVG instances), into data streams or even into other output formats (like PDF) through the application of *The Extensible Stylesheet Language Family* (XSL; http://www.w3.org/Style/XSL/). This group of languages includes XSL Transformations (XSLT), XML Path Language (XPath) and XSL Formatting Objects (XSL-FO). XSLT is the tool most often used to transform XML documents, because it supports both the structural transformation of XML objects and schema definitions.

Of particular importance to geoinformatics is the ability of specialized programs (XSLT processors) to transform GML instances into SVG instance documents (see Figure 2). XSLT, for example, enables the British Ordinance Survey (Ordnance Survey 2010) to visualize building permits and cadastral data.

XPath (World Wide Web Consortium 2007) is another related concept widely used in the process of transforming XML documents. It allows parts of XML documents to be addressed using a separate syntax. It simply indicates where to look for the document fragment by following the nested hierarchical structure of the XML elements. In the example below the first coordinate value of a point feature is extracted:

```
<xsl:value-of select="substring-
before(gml:centerOf/gml:Point/gml:pos,' ')"/>
```

Here, through the use of a so called *substring-before* function, the first part of the element content (e.g. the first coordinate value) is extracted from the `"gml:pos"` element.

Li ISPRS.indb 356 4/12/2011 4:19:33 PM

3 GEOGRAPHY MARKUP LANGUAGE

Geography Markup Language (GML) is the only broadly recognized markup language used for the modeling, transport and storage of geographic information (Lake et al. 2004, Open Geospatial Consortium 2007). The basic definitions for elements, attributes and data types of the most recent version 3 of GML are provided in more than 30 XML schema documents, called basis schemas.

3.1 *Application domain extension*

For many essential tasks that involve GML an application specific schema, or *Application Domain Extension* (ADE), is created. An ADE provides a formal description of all relevant features and relationships used for that particular task or field of work. Such a schema can define separate geographic objects or features (as shown in the BathingPond example below), with their corresponding elements and properties. The use of AFIS-ALKIS-ATKIS-Application schemas[5] published by the Surveying Authorities in Germany is a case in point (Seifert 2005).

3.2 *GML features and properties*

The essential building blocks of GML are *geo-objects* or *features*. Features are made up by the many data types used to specify non-geometric and geometric characteristics. They are meant to represent objects in the real world, such as streets, buildings, rivers or survey markers, all anchored within the specific context of a given application (Open Geospatial Consortium 2002).

The following shows how the instance of the geo-object `"BikePath"` could be coded in GML:

```
<BikePath gml:id="EZ02LLB"> ... </BikePath>
```

The instance is attached to the mandatory attribute `"id"` which, in turn, is a part of the gml namespace. This explicit marker is required for all object instances of a GML document. Further, features become child elements that more closely define object instances. In the case of BikePath, `"objectType"`, `"category"` and `"mapSheetNumber"` can be specified in the following manner:

```
<BikePath gml:id="EZ02LLB">
  <objectType>9102</objectType>
  <category>1470</category>
  <mapSheetNumber>l7120</mapSheetNumber>
</BikePath>
```

By convention, names of features begin with a capital letter and names of specific feature traits begin with a small letter. Words that are part of trait names are attached to each other, and the subsequent words (within the continuous string of words) start with a capital letter.

Trait values, as shown in the example above, can be declared within object instances. By applying XLink, however, traits can also be specified in another part of the document or in any other document that can be referenced by a URI.

```
<objectType xlink:href="http://www.anyserver.de/objType.xml#9401" />
```

[5] Theses schemas describe the data model for cadastral and topographic data of the Surveying Authorities of the States of the Federal Republic of Germany (AdV, http://adv-online.de). Core of NAS, its data exchange format, is the Geography Markup Language (GML).

Li ISPRS.indb 357 4/12/2011 4:19:33 PM

Above, the element objectType is an empty element. Its content is referenced by an attribute in the start tag—actual specifications are in fragment 9401 of file "objType.xml" on server "www.anyserver.de".

3.3 *Relationships between features*

Features can have properties which themselves are features. If a bathing pond is near a hiking trail the relationship can be denoted as follows:

```
<BathingPond>
  <objectType> 9401</objectType>
  <nearBy>
    <tkfd:HikingTrail gml:id="trail001" />
  </nearBy>
</BathingPond>
```

As mentioned before, a feature defined elsewhere can be referenced using XLink. Below, the GML instance of BathingPond indicates other sources addressable by an URL:

```
<BathingPond>
  <nearBy xlink:href="#trail002" />
</BathingPond>
```

3.4 *Geometrical properties*

While the previous standards for GML provided only a limited number of simple geometrical properties the current GML 3 standard supports almost all kinds of two- and three-dimensional properties, as well as coordinate reference systems, time properties, dynamic features, topology, spatio-temporal coverages, observations, units of measure and some rules on presentation style derived from SVG (Open Geospatial Consortium 2007:49).

GML features may have several geometric properties, each one embedded in a child element, which specifies the data type responsible for specific geometric properties (e.g. centerLineOf, curveProperty, surfaceProperty). The child element of a property element is one geometric element (e.g. Point, LineString, Polygon, etc.) or an XLink reference to a remote element. The following example shows the geometry of a linear feature specified by three vertices, each defined by a `"gml:pos"` element:

```
<gml:centerLineOf>
  <gml:LineString>
    <gml:pos>3512126.69 5411713.94</gml:pos>
    <gml:pos>3512219.35 5411712.56</gml:pos>
   <gml:pos>3512748.51 5411740.24</gml:pos>
  </gml:LineString>
</gml:centerLineOf>
```

A second example shows the locational description of a point feature:

```
<gml:centerOf>
  <gml:Point>
    <gml:pos>3512280.93 5410246.16</gml:pos>
  </gml:Point>
</gml:centerOf>
```

As mentioned above, geometrical properties can be referenced using an XLink link to another feature which is identified by its id, such as:

```
<gml:centerLineOf xlink:href="#0002CFV"></gml:centerLineOf>
```


Li ISPRS.indb 358 4/12/2011 4:19:33 PM

The value of this centerLineOf property is the resource returned by traversing the link (Open Geospatial Consortium 2007:25).

3.5 *Spatial reference systems*

Spatial geometry is determined by the system of coordinates used. In GML, the specific system of coordinates chosen must be explicitly designated through a *Coordinate Reference System* (CRS) or *Spatial Reference System* (SRS) (Lake 2004:192). A spatial reference system can in turn be designated by simply using a Universal Resource Identifier (URI) to link the document to a classification system developed by the OGP Geomatics Committee, formerly called European Petrol Survey Group (OGP 2010). Within this system, each spatial reference system is identified by a unique code. For example, "4326" is the code for the World Geodetic System 1984 used in many fields like GPS measurements and Internet mapping APIs:

```
<tkfd:TrainStation gml:id="b00214">
  <stationName>Aguas Calientes</stationName>
  <gml:centerOf>
    <gml:Point
srsName="http://www.opengis.net/gml/srs/epsg.xml#4326">
      <gml:pos>-72.525 -13.155</gml:pos>
    </gml:Point>
  </gml:centerOf>
</tkfd:Station>
```

3.6 *Feature collections*

Features are commonly aggregated into *feature collections*, in a manner similar to the layer concept used in GIS. A feature collection can be empty or contain an unlimited number of feature members. A collection may have spatial properties as well. Furthermore, a collection can itself also be a feature. It is even possible to have feature collections of feature collections!

The following example shows features as members of the feature collection `"SportLocations"`:

```
<tkfd:SportLocations gml:id="ID000001">
  <gml:featureMember>
    <tkfd:BathingPond gml:id="ID000002">
      ...
    </tkfd:BathingPond>
  </gml:featureMember>
  <gml:featureMember>
    <tkfd:BoatRent gml:id="ID000004">
      ...
    </tkfd:BoatRent>
  </gml:featureMember>
...
```

3.7 *Assessment*

GML is a highly developed, yet complex set of standards. The complexity is probably why "…the uptake of GML in the mainstream Web community has been slow" (Open Geospatial Consortium 2006b:11). The richness of GML is also its weakness. Usage of GML often results in very large and complex files, which can greatly hinder the ability to transfer data across networks and the further processing. The application specific schema is another problem because, in contrast to KML (see next section), for example, it does not allow data encoded in GML to be shared in an interoperable way directly among arbitrary applications.

Li ISPRS.indb 359 4/12/2011 4:19:33 PM

3.8 *Additional formats*

3.8.1 *KML*

The Keyhole Markup Language (KML) is an XML based grammar, originally created by Keyhole, Inc. Its use has increased significantly since Google purchased the company in 2004, especially since it was incorporated into the Google Earth Browser. The list of applications using KML today is extensive, including Internet mapping APIs, desktop GI systems, and virtual globes. KML received further recognition after it was published as a standard by the OGC (Open Geospatial Consortium 2008).

KML is a relatively concise standard, mixing data and presentation, which makes it easy to use and to learn. It is suitable for representing objects on pre-existing maps; however, it is not rich enough to be able to define a complete, complex map. For that a case more complete languages, like GML, must be used.

KML is effective at programming geographic explanations, but is limited in other ways. It uses only one coordinate reference system (WGS84) and altitude values are defined by EGM96 (Geoid Vertical Date). Furthermore, many KML viewers are two-dimensional and do not render three-dimensional models or provide a means to represent altitude. In addition, some standard aspects, such as animation, are usually not supported. Below is an example of KML code taken from http://www.mygeoposition.com/:

```
<?xml version="1.0" encoding="iso-8859-1"?>
<kml xmlns="http://earth.google.com/kml/2.1">
  <Placemark>
    <name>Stuttgart, Germany</name>
    <description></description>
    <Point>
      <coordinates>9.180769,48.777106,0</coordinates>
    </Point>
  </Placemark>
</kml>
```

3.8.2 *Scalable vector graphics*

The XML-based markup language *Scalable Vector Graphics* describes two-dimensional vector and mixed vector/raster graphics (World Wide Web Consortium 2009). The image is brought to the screen by so-called *User Agents*. These are either common browsers, separate SVG-viewers, or SVG-enabled Web browsers.

SVG offers a number of advantages over other common graphic formats for Web pages, such as GIF, JPEG and PNG (Watt & Lilley 2001, Ueberschär & Winter 2006):

- it is an open, non-proprietary, standardized language,
- it is directly supported in modern Web browsers, and hand-held devices,
- it supports high-resolution graphics that, among other things, retain the same graphical quality when zooming-in on an object,
- it supports a greater variety and depth of colors,
- it supports animation without increasing file size,
- it supports the Document Object Model, which allows elements written in SVG to be controlled and modified by ECMAScript, or other object-oriented programming languages,
- it facilitates filter effects, such as shading.

The above features are all especially useful when programming maps and other forms of geographic information. Hence, SVG has become the required format for outputs made for the Web Map Service (Kettemann 2005). Some programs automatically generate SVG code; other programs include SVG converters as an extension. In short, the application of SVG is widespread. For instance, SVG graphics are supported by Wikipedia. In the field of geoinformatics, it is used by the Geographical Survey Department of the German State of

Li ISPRS.indb 360 4/12/2011 4:19:33 PM

Baden-Württemberg (Graf 2004) and by the British Ordnance Survey (Ordnance Survey 2010). SVG plays an essential role in the XSL-based generation process of map tiles for OpenStreetMap (http://wiki.openstreetmap.org/wiki/SVG).

SVG is a standard in many browsers and is used in several APIs to overlay vector data on top of grid tiles. The following example (taken from Zerndl 2009) shows the integration of SVG elements (i.e. polyline, closed polygon, circle, and text) in XHTML. The background image of a topographic map of the Geographical Survey Department of the German State of Bavaria, is derived from its WMS server.

```
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <title>HTML and SVG with XHTML</title>
  </head>
  <body>
    <div style="position:absolute; left:50px; top:50px;">
      <img
src="http://www.geodaten.bayern.de/ogc/getogc.cgi?REQUEST=GetMap
&amp;VERSION=1.1.1&amp;  LAYERS=TK50&amp;  SRS=EPSG:31468&amp;BBOX=
4516489.753,5323730.381,4518264.567,5325505.195&amp;FORMAT=image/png
&amp;WIDTH=500&amp;HEIGHT=500&amp;TRANSPARENT=TRUE&amp; STYLES="/>
    </div>
    <svg xmlns="http://www.w3.org/2000/svg" width="660" height="440"
style="position:absolute; left:50px; top:50px; z-index:100">
    <polygon points="10,10 220,20 220,250 220,296 10,110"
             style="stroke:blue; fill:yellow; fill-opacity:0.5" />
    <circle class="flst" cx="300" cy="350" r="80"
            style="stroke:blue; stroke-width:2px; fill:grey;
            fill-opacity:0.5"/>
    <polyline points="300,30 350,50 200,200 150, 180 280, 50"
            style="stroke:blue; stroke-width:5px;
            fill:none; opacity:0.5" />
    <text x="50" y="400" style="font-family:Arial;
            font-size:36px;
            stroke:#FF0000;">SVG on top of XHTML</text>
    </svg>
  </body>
</html>
```

The positioning and the styling of the geometric elements are done by CSS which allows the precise overlay of XHTML's div element with the SVG element. The result is shown in Fig. 3.

3.8.3 *GeoRSS*

RSS (usually defined as Really Simple Syndication) is essentially an XML-based grammar used to syndicate news or weblogs (Open Geospatial Consortium 2006).

Normally the RSS feed structure contains a channel (feed), which has a title, a link and a description. The channel includes items (entry), each with a title, a short description and a link to a Web page that contains the complete article.

GeoRSS builds on RSS, to be able to describe the various ways to encode the location of news sites or weblogs (Worldkit 2006). Furthermore, GeoRSS differs from RSS by introducing several namespaces based on XML schema files. Due to its simplicity and versatility GeoRSS has the ability to make systems interoperable. It is used in both popular commercial and for-free APIs used in making maps.

Two versions of GeoRSS currently exist, GeoRSS GML and GeoRSS Simple. GeoRSS GML is intended to be a bonafide GML Application Profile. It is not only the smallest but probably the most "atomic" GML profile (Open Geospatial Consortium 2006:3). According

Li ISPRS.indb 361 4/12/2011 4:19:33 PM

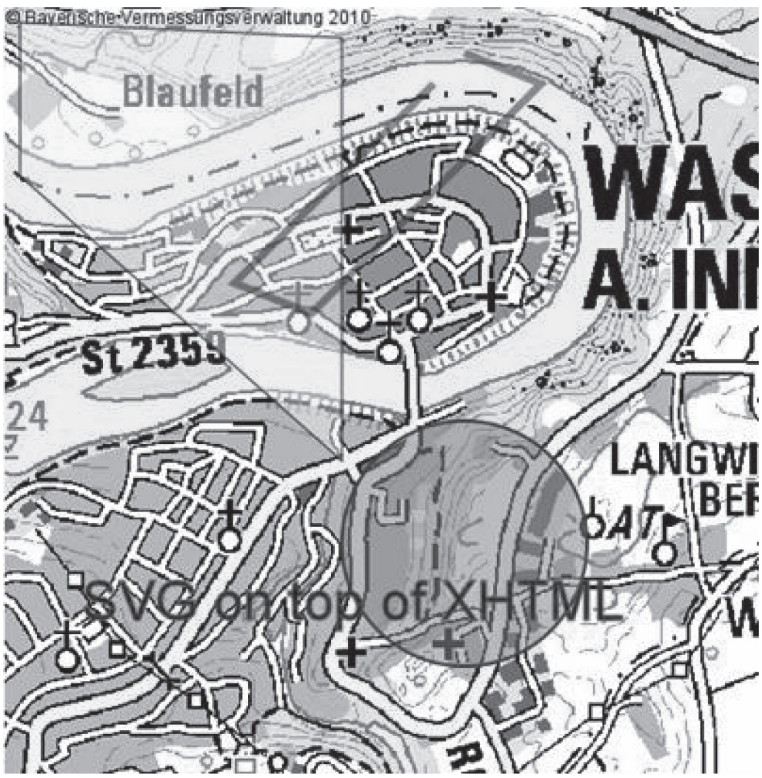

Figure 3. XML allows the integration of different XML languages, demonstrated in this example by overlaying SVG data to an image derived from a WMS server (Geobasisdaten © Bayerische Vermessungsverwaltung 2010, http://www.geodaten.bayern.de).

to the GML specifications, it supports a greater range of features than GeoRSS Simple. Notable is the ability of GeoRSS GML to support different coordinate reference systems.

Below is an example of how a point feature is coded in GeoRSS GML taken from the GeoRSS Web site (http://georss.org/gml):

```
<entry>
  <title>Crossing Muddy Creek</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>45.256 -110.45</gml:pos>
      </gml:Point>
  </georss:where>
</entry>
```

GeoRSS Simple supports basic geometries (point, line, box, polygon) and covers the typical use cases; the spatial reference system is restricted to WGS84. The following example, taken from the USGS Web site (http://earthquake.usgs.gov/earthquakes/catalogs/1 day-M2.5.xml), demonstrates how point features are coded in GeoRSS Simple. In addition to RSS elements and information on location HTML code is included in a CDATA section.

Li ISPRS.indb 362 4/12/2011 4:19:33 PM

```
<entry>
  <id>urn:earthquake-usgs-gov:nc:40237628</id>
  <title>M 2.6, Northern California</title>
  <updated>2009-06-04T12:49:49Z</updated>
  <link rel="alternate" type="text/html"
      href="/eqcenter/recenteqsww/Quakes/nc40237628.php"/>
  <link rel="related" type="application/cap+xml"
      href="/eqcenter/catalogs/cap/nc40237628"/>
  <summary type="html"><![CDATA[<img
      src="http://earthquake.usgs.gov/images/globes/40_-120.jpg"
      alt="38.191&#176;N 121.878&#176;W" hspace="20" />
      <p>Thursday, June 4, 2009 12:49:49 UTC
      <br>Thursday, June 4, 2009 05:49:49 AM at epicenter</p>
      <p><strong>Depth</strong>: 20.80 km (12.92 mi)</p>]]>
  </summary>
  <georss:point>38.1910 -121.8778</georss:point>
  <georss:elev>-20800</georss:elev>
  <category label="Age" term="Past hour"/>
</entry>
```

3.8.4 *GPS eXchange format (GPX)*

GPX is an open XML-based format especially designed for the interchange of GPS data. The latest XML schema version 1.1 has been available since August 2004 on Topgrafix's website (http://www.topografix.com/gpx/1/1/gpx.xsd) along with sample files and a validator. GPX is used to describe tracks (records of points along already traveled paths) or waypoints (descriptions of turning points, landmarks or other points along a user's path to a given destination). Each waypoint may also be given a timestamp, in order to record time and place of a visited position for later evaluation.

```
<?xml version="1.0" encoding="UTF-8" standalone="no" ?>
<gpx xmlns="http://www.topografix.com/GPX/1/1"
  creator="MapSource 6.11.6" version="1.1"
  xmlns:xsi=http://www.w3.org/2001/XMLSchema-instance
  xsi:schemaLocation="http://www.topografix.com/GPX/1/1
http://www.topografix.com/GPX/1/1/gpx.xsd">
  <wpt lat="42.998545" lon="6.400465">
    <ele>193.109131</ele>
    <name>VIGIE DE P</name>
    <cmt>04-JUN-09 13:48:46</cmt>
    <desc>04-JUN-09 13:48:46</desc>
    <sym>Flag, Blue</sym>
  </wpt>
  <trk>
     <name>03-JUN-09</name>
     <trkseg>
       <trkpt lat="-40.943039" lon="172.891352">
         <ele>731.219971</ele>
       </trkpt>
       <trkpt lat="-40.942966" lon="172.891563">
         <ele>727.855225</ele>
       </trkpt>
...
```

As is common for most GPS coding, the geographic coordinates are given in decimals based on WGS 84 datum.

The format is supported by a wide range of desktop applications. In addition, mobile phone platforms and handheld GPS devices support this format. An complete list can be found on http://www.topografix.com/gpx_resources.asp.

Li ISPRS.indb 363 4/12/2011 4:19:34 PM

3.8.5 *JSON and GeoJSON*

Another established format for transferring information is JavaScript Object Notation (JSON, http://json.org/). JSON is applied extensively throughout the Web and uses conventional JavaScript notation to define objects. For example, several of Google's web-based applications and services provide data feeds in JSON.

JSON-coded information can be executed as a JavaScript statement by using the eval() function. With this function, objects defined in a JSON string can be instantiated. A note of caution here, however. A JSON string may contain malicious code intended to hack into or harm programs. Therefore only JSON code obtained from a trusted server should be executed.

A variation of JSON, GeoJSON, is commonly used in the geoinformatics community (Schaub et al. 2008). It is especially useful for combining different geometrical features and shapes. GeoJSON standardizes the way spatial data is represented in JSON by structuring different geographic data according to the Simple Features Specification (Open Geospatial Consortium 2006). The following geocoding of `Accra, Ghana` is encoded using GeoJSON:

```
{ "name":"Accra, Ghana",
  "Status":
  {"code":200,
   "request":"geocode"},
   "Placemark":[
     {"id":"p1",
      "address":"Accra, Ghana",
      "AddressDetails":
        {"Country":
          {"CountryNameCode":"GH",
           "Locality":
           {"LocalityName":"Accra"}
          },"Accuracy": 4
        },
      "Point":
      {"coordinates":[-0.20738,5.54009,0]
      }
     }
   ]
}
```

3.8.6 *WKT format*

WKT (well known text) is a format for encoding simple features according to the specification of OGC (Open GIS Consortium 2010). It offers a light-weight solution to encode geometrical features like Point, Linestring, Polygon, MultiPoint, MultiLineString, MultiPolygon and GeometryCollections.

In the latest specification, version 1.2.1 (Open Geospatial Consortium 2010), more complex geometries are supported, like PolyhedralSurface, TIN and 3D Points. A simple Polygon shows the basic structure of the encoding:

```
POLYGON ((0 0 0, 0 0 1, 0 1 1, 0 1 0, 0 0 0))
```

Coordinate reference systems can be defined in WKT (Open Geospatial Consortium 2010:73). This is necessary because no default reference system is given, either for GML or GeoRSS GML.

The WKT standard is supported by most major applications and widely used with spatially enabled Data Base Management Systems (DBMS). Due to its compactness and standardization it is used for importing and exporting geodata. Some systems support the

Li ISPRS.indb 364 4/12/2011 4:19:34 PM

Well-known Binary Representation for Geometry (WKB format, Open Geospatial Consortium 2010:62).

3.8.7 *CSV format*

Comma Separated Values (CSV) format, an implementation of a delimited text file, is a widely supported light-weight format. Because this format is quite simple and supported by almost all spreadsheets and database management systems, it has become a pseudo standard even for GI systems.

The following example shows how CSV is used as part of a two-step coding process. First, the geometrical properties are converted to WKT format. Then, then resulting string is embedded in a CSV encoded string together with information about Road ID, road name, layer name, and geometry type.

```
Road_0001;Aimin_Jie;Road;Polyline;LINESTRING(116.375686
39.931674,116.375683 39.931593,116.375747 39.93026,116.375806
39.926574)
```

It is important to note that coordinate reference systems can not be defined using CSV format and no information about the data structure can be included.

Due to its simplicity CSV formatted data can be efficiently transferred through the web, easily parsed and evaluated. It is well suited for bulk loading of geo-data into databases.

4 CONCLUSIONS

In this chapter, the major standards used for the interoperable sharing of geodata in web applications have been presented and their most important properties have been defined. Basic applications of each standard have been illustrated by examples of code written in XML and other grammars.

GML, in particular, provides almost limitless capability to define geographic objects. However, it also has some drawbacks. It is not easy to learn quickly—the documentation of GML 3.2.1 is more than 420 pages long. In addition, it can be extended according to the user's needs. The size of files can become very large, especially when extensive XML tags surround the actual data. This can make the transfer of and access to GML instances through the Internet difficult. Finally, GML format is inappropriate for presenting images.

KML, on the other hand, can be very useful to represent geographic features in 3D. The downside of this format is, however, that it can not support descriptions of complex features, a task easily handled in GML. KML is supported by commercial mapping APIs such as Google Maps or BING, and also open source mapping APIs such as OpenLayers.

GeoRSS, in turn is the simplest of the three XML-based formats. It is not difficult to learn, can be quickly implemented, and files can be easily transferred through the Internet. Yet, GeoRSS is more limited than the other standards. In contrast to GML, it cannot display most forms of information about geographic objects. Therefore, as in the case of KML, commercial and open source mapping APIs support GeoRSS, primarily as an import data format. GeoRSS is used by most of the web-mapping applications, including OpenLayers, Yahoo! Maps, Google Maps, Virtual Earth and others.

In general, XML-based formats differ mostly in capability and complexity, the usual trade-off when choosing between different markup languages.

WKT and GeoJSON provide other possibilities to present geographic information on the Internet. They generally follow the simple feature model of OGC, which makes the task of system coordination easy. Both are "lean" formats, without the baggage of many tags wrapped around the essential data. Compared to fully-fledged GML, they are also well suited to decrease network load. Both are supported by many applications. WKT, in particular,

Li ISPRS.indb 365 4/12/2011 4:19:34 PM

can be understood intuitively and is easy to read. It excels in describing the geodetic datum, geoid, coordinate system and map projection of geospatial objects. For these reasons it is extensively used in many GIS programs. Also it is used by web-mapping applications, e.g. OpenLayers. GeoJSON, in turn, does not need a special parser, since JavaScript can be parsed by any browser directly. GeoJSON format is supported by some open source APIs, including Yahoo's FireEagle and OpenLayers. At present the commercial APIs such as Google Maps or Virtual Earth do not support this format. GeoJSON is also used by some desktop applications, such as FME.

Overall, WKT, GeoJSON and GeoRSS offer the best compromise between capability and complexity. They support almost every type of geometrical feature, and can be implemented quickly without much training or prior experience.

Actually, no vector data format for Internet based geoservices is perfect, and each has its advantages and disadvantages. That is why it is so important to be familiar with their different capabilities and limitations, in order to decide which is most appropriate for the task on hand.

In the future, coding in Web GIS standards will become even more widespread than today. Other non-geographic formats are also likely to increasingly incorporate information about geographic locations.

REFERENCES

Bray, T., Paoli, J., Sperberg-McQueen, C.M., Maler, E. & Yergeau, F. (2008) Extensible Markup Language (XML) 1.0 (Fifth Edition). [Online] Available from: http://www.w3.org/TR/2008/REC-xml-20081126/, [Last accessed 1st June 2010].

Bray, T., Paoli, J., Sperberg-McQueen, C.M., Maler, E., Yergeau, F. & Cowan, J. (2006) Extensible Markup Language (XML) 1.1 (Second Edition). [Online] Available from: http://www.w3.org/TR/2006/REC-xml11-20060816/, [Last accessed 1 st June 2010].

Graf, G. (2004) *Vektordatenausgabe im Format SVG am Beispiel der Ausgabe von Thematischen Kartenfachdaten (TKFD)*. [Online] Available from: http://www.lv-bw.de/LVShop2/produktinfo/wir-ueber-uns/links/svg/VektordatenAusgabeImFormatSVG_110105.pdf, [Last accessed 2nd June 2010]

Kettemann, R. (2005) GIS im Intra-/Internet und Web-Dienste für Geoinformationssysteme. In: Kettemann, Rainer, Coors. & Volker, (2005) *Aktuelle Entwicklungen in der Geoinformatik*. Tagungsband 5. Vermessungsingenieurtag, Hochschule für Technik, Stuttgart.

Lake, R., Burggraf, D.S. & Trninic, Rae, L. (2004) *GML, Geography Mark-Up Language: Foundation for Geo-Web*. Wiley.

OGP, (2010) *OGP Geomatics Committee*. [Online] Available from: http://www.epsg.org, [Last accessed 16th December 2010].

Open Geospatial Consortium. (2001) *OpenGIS® Implementation Specification: Coordinate Transformation Services*. [Online] Available from: http://portal.opengeospatial.org/files/?artifact_id=999, [Last accessed 22nd May 2009].

Open Geospatial Consortium (2002) *Web Feature Service Implementation Specification*. de la Vretanos, P.A., (eds.), [Online] Available from: http://portal.opengeospatial.org/files/?artifact_id=8339, [Last accessed 10th June 2009]

Open Geospatial Consortium. (2004) *OGC Web Map Service Interface*. de la Beaujardiere, J., (eds.), [Online] Available from: http://portal.opengeospatial.org/files/?artifact_id=4756, [Last accessed 10th June 2009

Open Geospatial Consortium. (2006) *An Introduction to GeoRSS: A Standards Based Approach for Geo-enabling RSS feeds*. Reed, C., (eds.), [Online] Available from: http://www.opengeospatial.org/pt/06-050r3, [Last accessed 2nd June 2010].

Open Geospatial Consortium. (2007) *OpenGIS® Geography Markup Language (GML) Encoding Standard*. Portele, C., (eds.), [Online] Available from: http://portal.opengeospatial.org/files/?artifact_id=20509, [Last accessed 1st June 2010].

Open Geospatial Consortium. (2008) *OGC® KML*. Wilson, T., (eds.), [Online] Available from: http://portal.opengeospatial.org/files/?artifact_id=27810, [Last accessed 1st June 2010].

Li ISPRS.indb 366 4/12/2011 4:19:34 PM

Open Geospatial Consortium. (2010) *OpenGIS® Implementation Standard for Geographic information - Simple feature access - Part 1: Common architecture*. [Online] Available from: http://portal.opengeospatial.org/files/?artifact_id=25355, [Last accessed 7th September 2010]

Ordnance Survey. (2010) What is the default OS MasterMap® style? [Online] Available from: http://www.ordnancesurvey.co.uk/oswebsite/products/osmastermap/faqs/data009.html, [Last accessed 1st June 2010]

Schaub, T. Doyle, A., Daly, M., Gillies, S. & Turner, A. (2008) GeoJSON draft version 6. [Online] Available from: http://wiki.geojson.org/GeoJSON_draft_version_6, [Last accessed 10th June 2009]

Seifert, M. (2005) Das AFIS-ALKIS-ATKIS-Anwendungsschema als Komponente einer Geodateninfrastruktur. *Zeitschrift für Vermessungswesen*, 2/2005.

Ueberschär, N. & Winter, A.M. (2006) *Visualisieren von Geodaten mit SVG im Internet*. Wichmann, p. 256.

Watt, A. & Lilley, C. (2001) *SVG Unleashed*. SAMS, ISBN 0-672-32429-6. p. 1117.

Worldkit. (2006) WorldKit easy web mapping user manual. http://worldkit.org/doc/rss.php, [Last accessed 2nd June 2010].

World Wide Web Consortium. (2005) *Document Object Model (DOM)*. http://www.w3.org/DOM, [Last accessed 10th June 2009].

World Wide Web Consortium. (2007) *XML Path Language (XPath) 2.0.* [Online] Available from: http://www.w3.org/TR/xpath20, [Last accessed 27th August 2010].

World Wide Web Consortium. (2009) *Scalable Vector Graphics (SVG) 1.1 Specification*. Ferraiolo, J., Fujisawa, J. & Jackson, D., (eds.), [Online] Available from: http://www.w3.org/TR/2003/REC-SVG11-20030114, [Last accessed 2nd June 2010].

Zerndl, M. (2009) Personal Communications, January 2009.

Li ISPRS.indb 367 4/12/2011 4:19:35 PM