Geographic Information Systems 2018/19
Week 3, Topic 1

Mark Foley
School of Computing
DIT, Kevin Street

mark.foley@dit.ie

In this session...

Representation of spatial features

- “Simple features” model

Geometry: points, lines and polygons

Data structures: shapefile, spatialite, PostGIS, etc.

Interchange formats: JSON/GeoJSON, WKT, GML etc.

Representation of spatial features

Representation of spatial features:

- are needed to convey information
- fit information into a standard form or model
- almost always simplify the truth that is being represented
- can rarely (never?) be perfect

Details can be irrelevant, or too expensive and voluminous to record

- It’s important to know what is missing in a representation

Representations can leave us uncertain about the real world

The Fundamental Problem

Geographic information links a place, and often a time, with some property of that place (and time)

- "The temperature at 34 N, 120 W at noon local time on 12/2/99 was 18 Celsius"

The potential number of properties is vast

- In GIS we term them *attributes*
- Attributes can be physical, social, economic, demographic, environmental, etc.

The number of places and times is also vast

- Potentially infinite

The more closely we look at the world, the more detail it reveals

- Potentially *ad infinitum*
- The geographic world is infinitely complex

Humans have found ingenious ways of dealing with this problem

- Many methods are used in GIS to create representations or *data models*

Discrete Objects and Continuous Fields

- Two ways of conceptualizing geographic variation
  - The most fundamental distinction in geographic representation
- Discrete objects
  - The world as a table-top
  - Objects with well-defined boundaries
- Continuous Fields
  - Properties that vary continuously over space

Discrete Objects and Continuous Fields (cont'd)

- Discrete objects
  - Points, lines, and areas
  - Countable
- Continuous fields
  - Properties that vary continuously over space
    - Value is a function of location
    - Property can be of any attribute type, including direction
  - Elevation as the archetype
    - A single value at every point on the Earth's surface

Vector data

Points

- Single point (0 dimension), (x,y)

Lines

- Drawn between two point locations (1 dimension)
- Polyline or linestring – multiple line segments – list of point locations –
  $(x_1y_1, x_2y_2 \ldots x_ny_n)$

Polygons (Areas)

- As polyline but first & last points are equal thus ensuring closure
- Represents area (2 dimensions)

Vector model often (though not always) associated with discrete object view

Note that the chosen representation can vary with scale

- A city could be represented by a point at 1:1,000,000 scale and as a polygon at 1:50,000 scale

Storing vector data in a computer

For the moment we'll consider the *simple features* model, i.e. no *topological* relationships considered

- **Simple**: every *feature* has its own complete geometry and we don't store the relationship between different features. For example, two adjacent polygons will each store their shared boundary

Geometries aren't that interesting on their own. We need to store *attributes* to give them context.

We usually use some variant of the **georelational** model. This is based on the relational model where we store each feature as a row of related attributes including geometry.

- Geometry and other attributes can be stored separately (as in the *shapefile* format) or together, with the geometry serialized into a binary format (as in *PostGIS*)

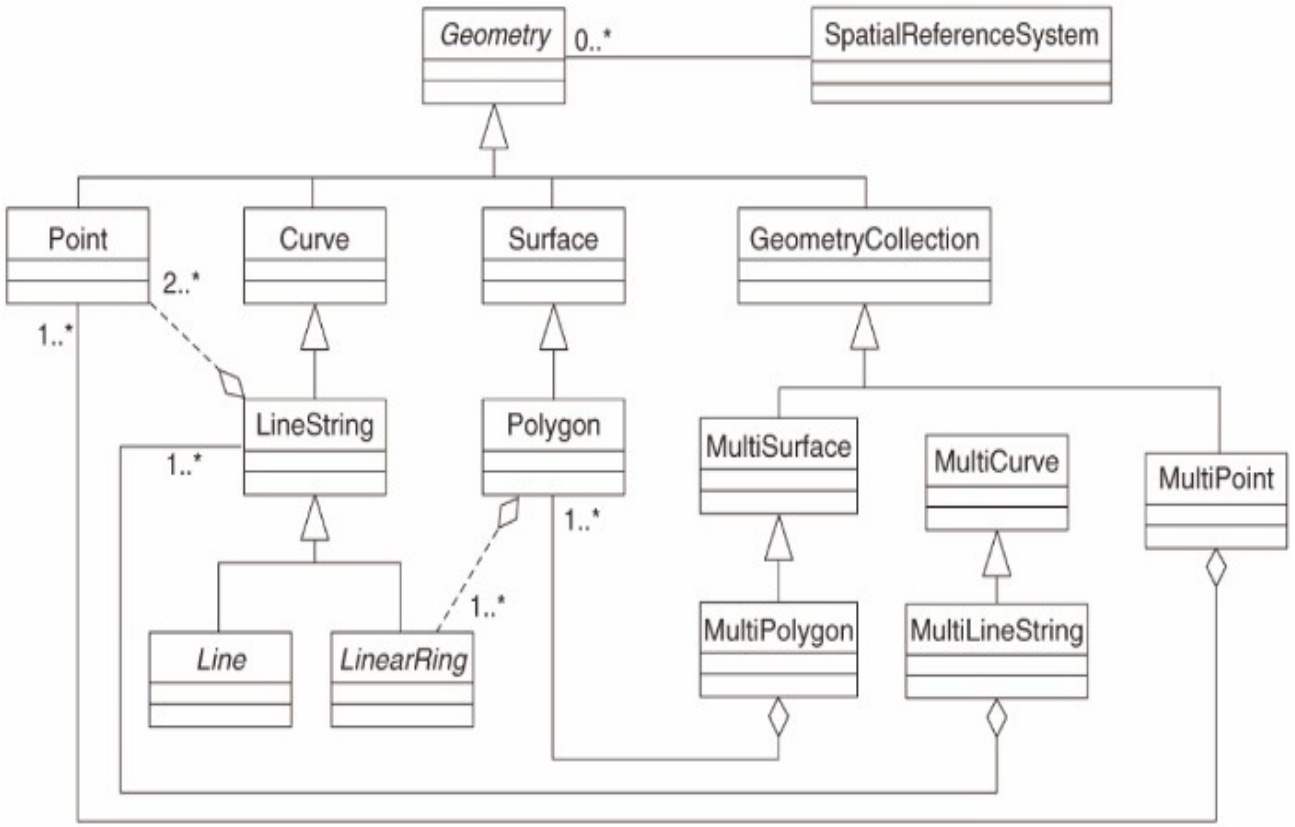
OGC SFS Model

What is OGC?

The Open Geospatial Consortium (OGC) is an international standards organization. In the OGC, organizations worldwide collaborate in an open process encouraging development and implementation of standards for geospatial content and services.

What is the SFS Model?

The Simple Feature for SQL (SFS) Model is a non-topological way to store geospatial data in a database and defines functions for accessing, operating, and constructing these data.



Vector data storage formats

Shapefile

- Common file-based format
- Splits data into several files with different suffixes
  - Geometry - ***.shp*** - binary format consists of a single fixed-length header followed by one or more variable-length records
  - Attribute data - ***.dbf*** - columnar attributes for each shape, in dBase IV format
  - Shape index- ***.shx*** - a positional index of the feature geometry to allow seeking forwards and backwards quickly
  - CRS - ***.prj*** - the coordinate system and projection information, a plain text file describing the projection using well-known text format
- May be others, note that the projection information is not mandatory but is usually present
- Always consider the collection of files with the same filename prefix e.g. **name_of_shapefile.*** as a single unit

Vector data storage formats

PostgreSQL/PostGIS

PostgreSQL, often simply Postgres, is an object-relational database management system (ORDBMS) with an emphasis on extensibility and standards compliance. It can handle workloads ranging from small single-machine applications to large Internet-facing applications with many concurrent users. It is available for all platforms, MS Windows, macOS and Linux (supplied in most distributions).

PostgreSQL is ACID-compliant and transactional.

PostGIS is an open source software program that adds support for geographic objects to the PostgreSQL object-relational database. PostGIS follows the Simple Features for SQL specification from the (OGC).

PostGIS

Sample queries

select location, st_astext(location), name, latitude, longitude from geonames_pop_5000 limit 2;

Yields...

```
location                  |       st_astext        |    name     | latitude | longitude
--------------------------------------------------+------------------------+-------------+----------+-----------
 0101000020E6100000C780ECF5EE4F1AC03524EEB1F4994A40 | POINT(-6.57806 53.20278) | Eadestown   | 53.20278 |  -6.57806
 0101000020E6100000B28009DCBA7B18C043FF04172B9A4A40 | POINT(-6.12083 53.20444) | Little Bray | 53.20444 |  -6.12083
(2 rows)
```

select a.countyname from ctygeom a, ctygeom b where st_touches(a.geom, b.geom) and b.countyname like '%ongford%';

Yields...

```
     countyname
------------------
 Westmeath County
 Cavan County
 Leitrim County
 Roscommon County
(4 rows)
```

Vector data storage formats

SQLite/SpatiaLite

SpatiaLite is a spatial extension to SQLite, providing vector geodatabase functionality. It is similar to PostGIS, Oracle Spatial, and SQL Server with spatial extensions, although SQLite/SpatiaLite aren't based on client-server architecture: they adopt a simpler personal architecture. i.e. the whole SQL engine is directly embedded within the application itself: a complete database simply is an ordinary file which can be freely copied (or even deleted) and transferred from one computer/OS to a different one without any special precaution.

SpatiaLite extends SQLite's existing spatial support to cover the OGC's SFS specification.

Vector data storage formats

Other (this is just a random selection)

AutoCAD DXF – contour elevation plots in AutoCAD DXF format (by Autodesk)

Cartesian coordinate system (XYZ) – simple point cloud

Digital line graph (DLG) – a USGS format for vector data

GeoMedia – Intergraph's Microsoft Access based format for spatial vector storage

ISFC – Intergraph's MicroStation based CAD solution attaching vector elements to a relational Microsoft Access database

MapInfo TAB format – MapInfo's vector data format using TAB, DAT, ID and MAP files

National Transfer Format (NTF) – National Transfer Format (mostly used by the UK Ordnance Survey)

SOSI – a spatial data format used for all public exchange of spatial data in Norway

Spatial Data File – Autodesk's high-performance geodatabase format, native to MapGuide

TIGER – Topologically Integrated Geographic Encoding and Referencing

Vector Product Format (VPF) – National Geospatial-Intelligence Agency (NGA)'s format of vectored data for large geographic databases

Interchange formats

GeoJSON

GeoJSON is an open standard format designed for representing simple geographical features, along with their non-spatial attributes. It is based on JSON, the JavaScript Object Notation.

The features include points (therefore addresses and locations), line strings (therefore streets, highways and boundaries), polygons (countries, provinces, tracts of land), and multi-part collections of these types.


```
{
  "type": "FeatureCollection",
  "features": [{
  "type": "Feature",
  "id": "geonames_pop_5000.fid--351590fd_1661776951f_-275e",
  "geometry": {
  "type": "Point",
  "coordinates": [-9.05026, 51.58666]
  },
  "geometry_name": "location",
  "properties": {
  "geonameid": 2964820,
  "name": "Derry",
  "latitude": 51.58666,
  "longitude": -9.05026,
  "countrycode": "IE",
  "population": 10000,
  "timezoneid": "Europe/Dublin",
  "modificationdate": "2010-08-29Z",
  "bbox": [-9.05026, 51.58666, -9.05026, 51.58666]
  }
  }],
  "crs": {
  "type": "EPSG",
  "properties": {
  "code": "4326"
  }
  },
  "bbox": [-9.05026, 51.58666, -6.10028, 53.27833]
}
```

Interchange formats

GML

The Geography Markup Language (GML) is the XML grammar defined by the Open Geospatial Consortium (OGC) to express geographical features. GML serves as a modeling language for geographic systems as well as an open interchange format for geographic transactions on the Internet. Key to GML's utility is its ability to integrate all forms of geographic information, including not only conventional "vector" or discrete objects, but coverages (see also GMLJP2) and sensor data.

```
<?xml version="1.0" encoding="UTF-8"?>
<wfs:FeatureCollection xmlns:wfs="http://www.opengis.net/wfs/2.0" xmlns:dit="http://gis.dit.ie/dit" xmlns:gml="http://www.opengis.net/gml/3.2"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" timeStamp="2018-10-01T14:51:53.804Z" xsi:schemaLocation="http://www.opengis.net/gml/3.2
http://mf2.dit.ie:8080/geoserver/schemas/gml/3.2.1/gml.xsd http://www.opengis.net/wfs/2.0 http://mf2.dit.ie:8080/geoserver/schemas/wfs/2.0/wfs.xsd
http://gis.dit.ie/dit http://mf2.dit.ie:8080/geoserver/dit/wfs?service=WFS&amp;version=1.0.0&amp;request=DescribeFeatureType&amp;typeName=dit
%3Ageonames_pop_5000">
<wfs:boundedBy>
   <gml:Envelope srsDimension="2" srsName="http://www.opengis.net/gml/srs/epsg.xml#4326">
   <gml:lowerCorner>-8.18052 51.91526</gml:lowerCorner>
   <gml:upperCorner>-8.18052 51.91526</gml:upperCorner>
</gml:Envelope>
</wfs:boundedBy>
<wfs:member>
   <dit:geonames_pop_5000 gml:id="geonames_pop_5000.fid--351590fd_1661776951f_-272a">
      <gml:name>Midleton</gml:name>
      <gml:boundedBy>
         <gml:Envelope srsDimension="2" srsName="http://www.opengis.net/gml/srs/epsg.xml#4326">
         <gml:lowerCorner>-8.18052 51.91526</gml:lowerCorner>
         <gml:upperCorner>-8.18052 51.91526</gml:upperCorner>
      </gml:Envelope>
   </gml:boundedBy>
   <gml:location>
      <gml:Point srsDimension="2" srsName="http://www.opengis.net/gml/srs/epsg.xml#4326">
      <gml:pos>-8.18052 51.91526</gml:pos>
   </gml:Point>
</gml:location>
<dit:geonameid>2962630</dit:geonameid>
<dit:asciiname>Midleton</dit:asciiname>
<dit:latitude>51.91526</dit:latitude>
<dit:longitude>-8.18052</dit:longitude>
<dit:featureclass>P</dit:featureclass>
<dit:featurecode>PPL</dit:featurecode>
<dit:countrycode>IE</dit:countrycode>
<dit:admin1code>M</dit:admin1code>
<dit:admin2code>4</dit:admin2code>
<dit:population>8891</dit:population>
<dit:dem>10</dit:dem>
<dit:timezoneid>Europe/Dublin</dit:timezoneid>
<dit:modificationdate>2011-05-28Z</dit:modificationdate>
</dit:geonames_pop_5000>
</wfs:member>
</wfs:FeatureCollection>
```

Coming next...

Topology

Topological data structures - general overview

Advantages/disadvantages vis a vis simple features