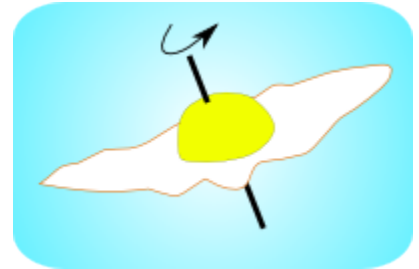


Fried Egg

The Fried Egg module provides the Geotools library with support for the creation and manipulation of the geometric objects defined by various realizations of the abstract ISO 19107 spatial schema.

Status:

- 2008.05 Design work started.
- No specification document, yet.
- No module unsupported/isogeom, yet.



Moving beyond JTS to ISO Geometry

Through 2008, Geotools 2.x has depended on an external library, the Java Topology Suite (JTS), both for the definition of the vector geometric structures and for the operations on those structures. JTS has proved sublime in meeting the immediate needs of Geotools over these past years.

JTS focuses on a simple geometric model sufficient for the representation of all objects defined in the Simple Features for SQL (SFSQL) specification of the Open Geospatial Consortium (OGC). All JTS objects live on a simple manifold of 2 dimensional, infinite, unbounded space. The coordinates are defined in an orthogonal coordinate system with equal units along each axis. All JTS objects can be decomposed into simplices and therefore all lines are comprised of segments with linear interpolation between the two vertices and polygon edges use lines of that type.

The ISO 19107 specification, "Geographic information - Spatial Schema," published by the International Organization for Standardization (ISO) and adopted as the core specification for spatial schema by the OGC, defines an abstract schema of geometric objects with greater complexity than those defined in JTS. Organizations are free to define implementation schemata based on the ISO 19107 schema which may restrict or extend the base schema.

The Fried Egg module aims to provide the facilities within the Geotools library by which third party programmers may build software applications conformant with any of these implementation schemata.

The core hurdles of ISO Geometry

1. ISO 19107 is an abstract schema, we must allow users both to subset and extend the library
2. Operations are CRS aware.
3. Object boundaries may use curvilinear interpolation between vertices.
4. Objects and the operation space may be three dimensional.
5. Complex geometries open the road for shared geometries between features.

Abstract schema and its implementation

ISO 19107 is abstract

This International Standard specifies conceptual schemas for describing the spatial characteristics of geographic features, and a set of operations consistent with these schemas. {emphasis added}

which means that Geotools cannot aim simply to implement the ISO 19107 specification and call it a day. We must instead provide users with a way to subset the full implementation to match the types and operation rules present in a particular application schema. For example, both ISO 19125 and ISO 19137 provide implementation schemas which include only 'simple' objects, however, these schemata include different objects and have different rules for the relation of an object to its boundary.

CRS aware analysis

The ISO 19107 specification says

In all of the operations, all geometric calculations shall be done in the coordinate reference system of the first geometric object accessed, which is normally the object whose operation is being invoked.

which means that operations must take place in the geometric space defined by the first object's CRS. This introduces a major new difficulty not merely in tracking the CRS of objects and converting objects to a common CRS when required but in defining the base operations in all of the geometric spaces for which geospatial CRS are commonly defined.

For example, if we start with a geographic object defined on an ellipsoid, the object's `.buffer(Distance d)` method will need to be calculated on the surface of the ellipsoid: on an unbounded, curved, finite manifold with one axis which wraps around to the value at the other extreme of the axis and a second axis where values merely decrease beyond the extreme.

Curvilinear interpolation

The ISO 19107 specification introduces object which use non-linear interpolation between vertices such as `GM_ArcString` which uses circular arcs for interpolation, `GM_GeodesicString` which uses geodesic curves as interpolation between vertices and `GM_Clothoid` which uses a parametric curve to describe the arc between vertices. The introduction of these new interpolation schemes requires a completely new design for the geometric operations. Most of the operations defined for geometries with only linear interpolation can consider only the vertices of the geometry and ignore the intervening segment. This approach fails when performed on curvilinear geometries.

Three dimensional coordinate spaces

The ISO 19107 specification introduces three dimensional objects and operations in three dimensional space. Many developers appear to believe erroneously that this is the major addition of ISO 19107. However, because the specification considers the CRS of the objects, the specification actually introduces several three dimensional object spaces!

The 3D euclidean (orthogonal, infinite axes) space is well known and sufficiently understood to not present much inherent difficulty. However, geospatial data commonly uses three other 3D coordinate systems: a plane plus a height above the surface, an ellipsoid plus an orthogonal height above the ellipsoidal surface, and a geoid model plus a gravimetric height above a reference value. Handling these alternative three dimensional systems pose major additional challenges for the module.

Indirect spatial representation

The ISO 19107 specification introduces complex geometric structures such as Triangular Irregular Network objects, Spline Surface objects, Aggregate objects, and Complex objects. Coupled with the ongoing work on so called "Complex Features" which are namespace aware and can have associations, this opens the way for Features whose spatial schema is defined by association with another feature. At its simplest, this merely allows a Feature to obtain its spatial schema from another; at its most complex, it allows a feature to act as a coverage aggregating a group of other features for which it can generate, on demand, a spatial representation.

While this issue will not arise in the immediate needs of the project, it may be worth keeping in mind as the module grows.

Past efforts

There have already been two notable efforts to integrate ISO 19107 geometries into geotools. Unfortunately, neither of the efforts yielded a clear plan for the future so it is exceedingly hard to know how to build on those past efforts.

JTS wrapper

The first effort done to implement ISO 19107 geometries in Geotools took as its goal merely to support the same operations as already existed in JTS but acting via the ISO interface. For this effort, the interfaces to the ISO Geometry objects have been implemented in GeoAPI. The JTS wrapper module's page is [here](#).

This effort has become dormant in the past year. There has been no active work on the `jts-wrapper` module and there is no clear path forward from that work to a broader library.

ISO Geometry module

The second effort to implement ISO Geometry was initially started by Sanjay Jeena and then extended by Refrations Research. This module is described in [the Geometry module's wiki page](#).

Unfortunately this work, while extensive, is hard to build upon. The module has been brought into the 'supported' plugin directory in the hierarchy of Geotools modules which means that it should not be broken from release to release making radical changes difficult. The module was not started with any concrete documentation describing its long term evolution so there is no easy way to extend the module. Finally, it appears the module is too tied to JTS to be generally applicable to the full complexity of the ISO 19107 and related specifications.

For example, while ISO 19107 clearly requires that the operations on Geometries be aware of the CRS of the objects on which they are working, we can see that the current Geometry module misses this aspect of the specification in its handling of the `Geometry` object's `distance` method. As a very simple example, we consider invoking the `.distance` method on one `Point` geometry in, say, a geographic CRS (lambda, phi) with another as argument, in say, a projected CRS (x,y in meters). The code in `GeometryImpl`, first evaluates the two geometries to see what type of geometry they are. In this process, the code casts both Geometries to their `PointImpl` (which means the library can only work with itself). The module then extracts the `DirectPostions` from the two `Point` objects (ok), but then pulls the coordinate doubles directly from the `DirectPosition` objects (oh, no!) with this code from `DirectPositionImpl`:

```
public double distance(DirectPosition p) {
    return AlgoPointND.getDistance(this.coordinate, p.getCoordinates());
}
```

with no consideration that the objects could be in different CRS's, that the axes could be in different units (or that the units could be non metric like longitude). Finally that method of `AlgoPointND` is implemented by the code

```
public static double getDistance(double[] p0, double[] p1) {
    // OK
    double sum = 0.0;
    for (int i=0, n=p0.length; i<n; ++i) {
        double a = p1[i] - p0[i];
        sum += a*a;
    }
    return Math.sqrt(sum);
}
```

which presumes that distance is a simple euclidean measure rather than a measure defined on the manifold of the first object. So we have lost all the CRS awareness implied by ISO 19107 and are comparing apples (long-lat coordinates) to oranges (UTM coordinates). Unfortunately, this is not merely a bug of this method, it's an oversight of design.

The ISO Geometry module also suffers from several other shortcomings. A notable limitation, mentioned in passing above, is that the module only supports working against objects of its own construction rather than being able to work against any object which implements the GeoAPI geometry interfaces. The module has not yet considered the issue of different boundary rules in different realizations of the ISO 19107 abstract specification.

Overall, the assessment of this effort is that it may have some good code that can be absorbed into a new module but the costs of working directly on the module outweigh the benefits. The Fried Egg module hopes to grab what it can from the earlier ISO Geometry module but Fried Egg will be a new effort centered around a more complete design.

Plans for the module

Work on this module has been started at the beginning of 2008 in a collaboration between the satellite unit located in Arles, of the Université de Marseille, France and the

First goal

The working aim of this effort comprises

1. a full design specification for the module,
2. a skeleton code layout of the full library,
3. the implementation of several examples of complex geometries, and
4. a complete implementation of either ISO 19125-1 or ISO 19137.

A complete design specification

The Fried Egg module will be developed against a concrete specification document. This use of a formal design document should help all those working on the module collaborate and help other developers understand the internal design of the module so they can use its public interface, fix errors in the implementation which they encounter and implement the pieces which will remain missing over the near and medium term future.

All of the initial work on this module is being done to develop this design document. It is intended that a full first draft of the design specification be published and linked to from this section prior to the start of work revising existing code and creating new code.

The design needs to consider both high level questions and internal details. A high level issue involves how to allow a user of an application based on Geotools to override the default analytic pathways of the geometric operations and force every operation to occur in a particular space or with a particular boundary rule. Another high level issue involves enabling a way for a programmer developing an application based on Geotools to restrain the geometry system to use and operate with only the objects defined in a particular concrete specification drawn on the abstract ISO 19107.

A skeleton library

The skeleton of the full library aims to provide the space for future code and show the expected control flow which will occur when the module handles complex structures which will not be implemented in the initial days of the project. For example, intersections of three dimensional objects will not be in the initial aim of the project but the library must be designed so that anyone who wants to implement that functionality has a clear place to do so. The skeleton will need to cover the design of the logging system, the progress monitoring and control system (allowing a user to cancel a long lived operation), the exceptions which will be thrown by illegal or merely unimplemented methods being called, the system for instantiation of new data structures (geometry objects), and various other similar issues.

A ISO 19137 implementation for Cartesian objects

The next step will be to handle cartesian ISO 19137 objects retracing the steps of JTS. Since the boundary rule changes between ISO 19137 and JTS this will give us a chance to isolate that from the rest of the library.

Beyond

The two obvious next steps would be to handle ISO 19125 objects (opposite boundary rules) and then the really fun step of handling objects with geographic CRS and then 3D beyond.

But one step at a time, shall we?