QGIS Application - Feature request #3483
Unified topological editing

2011-02-11 06:12 AM - Paolo Cavallini

Status:	Feedback		
Priority:	Low		
Assignee:			
Category:	Digitising		
Pull Request or Patch supplied:	No	**Resolution:**	
Easy fix?:	No	**Copied to github as #:**	13543

Description

Currently QGIS has two different methods for editing:
- one (optionally) pseudotopological, to be used with [[SimpleFeature]] layers (shp, postgis, spatialite etc.)
- one fully topological, limited to GRASS layers.

IMHO it would be good to unify the two, digitizing always (optionally?) with full topology, and writing (a)topologically when on a topological format (in future also e.g. GML3-topo, [[PostGIS]]-topo), (b) non topologically when on SF.
This would be both more powerful and clean, and less confusing for users.
See also http://grass.osgeo.org/grass64/manuals/html64_user/wxGUI.Vector_Digitizer.html
http://grass.osgeo.org/wiki/WxGUI_Vector_Digitizer for missing (topological) features.

History

#1 - 2011-10-16 02:48 AM - Sandro Santilli

- Assignee deleted (nobody -)

- Pull Request or Patch supplied set to No

See http://strk.keybit.net/blog/2011/10/14/postgis-topology-iso-sqlmm-complete/comment-page-1/#comment-7521 for some design ideas about PostGIS-topology specific editing toolbar.

#2 - 2011-10-19 06:58 PM - Luís Ferreira

Some ideas [[http://strk.keybit.net/blog/2011/10/14/postgis-topology-iso-sqlmm-complete/comment-page-1/#comment-7638]]
(1) topology layer (in memory or else), where we can define the intervening layers/postgis tables;
(2) rules (overlaps, gaps, covers, must contain point, touch line, ...);
(3) a list of errors in face of choosen rules;
(4) tools to merge/subtract/create feature/mark as exception or error.

#3 - 2011-10-20 07:17 AM - Martin Dobias

The main issue we are facing here is that the topological models of various data sources (grass, postgis-topo, osm) differ quite a lot between each other. From what I know:

- PostGIS. Primitives: nodes, edges, faces - they correspond to the elements of a graph from graph theory: node is a point, edge is a polyline connecting two nodes, faces are formed by enclosing edges. Planar graph is enforced.

- GRASS. Primitives: point, line, boundary, centroid. Points and (poly)lines are standalone objects, boundaries are polylines that should touch at the endpoints to form areas. Centroids are special type of points that assigns a layer to the area enclosed by the boundaries. Only boundaries are required to form a planar graph (?)

- OSM. Primitives: nodes, ways. Ways are constructed strictly from nodes. There is no explicit notion of areas: a closed way is usually considered as an area, although that is not always true (e.g. roundabout is a closed way, but it represents a linear ring). The situation with polygons with holes and multipolygons is difficult: relation elements may connect individual ways. Planar graph is not enforced.

Please correct me if I am wrong.
The question is how to handle these differences:

- Use a minimal topology model that makes only few assumptions? E.g. there may be points and polylines, the editing tools would be able to edit them and individual topological models would implement the logic - decide what is valid and what is not etc.
- Build a topology abstraction that is able to handle a superset of these individual models? The individual topological models would only turn on/off some features/rules.

The former approach sounds much better

#4 - 2011-10-20 09:48 AM - Sandro Santilli

I'd like to avoid too much abstraction on the PostGIS side actually.
All I can tell is that for postgis topology you have these operations:

- Add an isolated node
- Move an isolated mode
- Remove an isolated node
- Add an edge connecting two existing nodes (automatically creates a face if closes a ring)
- Move an edge vertices except its endpoints
- Split an edge by a node (add a "splitting" node)
- Remove an edge (automatically removes a face if opens a ring)
- Heal two edges removing the connecting node

Note that some of the above operations (those creating new edges or faces or removing them)
also have an option to decide whether or not to save pre-existing faces/edges or always replace
them with the new ones.

Seems all very postgis-specific (or should I say ISO SQL/MM specific).
Unless someone from GRASSland sees an unification opportunity (beside icons?) I'd move this into another ticket.

#5 - 2011-10-21 10:20 AM - Sandro Santilli

FYI: I've started a project on github for postgis topology editing. At least will serve for brainstorming.

https://github.com/strk/qgis_pgis_topoedit/wiki

#6 - 2012-10-06 02:23 AM - Pirmin Kalberer

- Target version changed from Version 2.0.0 to Future Release - Nice to have

#7 - 2013-09-26 01:24 AM - Markus Neteler

Sandro Santilli wrote:

> *FYI: I've started a project on github for postgis topology editing. At least will serve for brainstorming.*
>
> *https://github.com/strk/qgis_pgis_topoedit/wiki*

Sandro, do you continuously develop this plugin?

#8 - 2013-09-26 01:32 AM - Sandro Santilli

Markus, I'm not developing it, no. But I'm maintaining it (unfortunately needed as QGIS API changes more often than I'd like).
Mind you: that plugin is postgis-specific.

#9 - 2017-05-01 12:50 AM - Giovanni Manghi

- Easy fix? set to No

#10 - 2018-01-29 07:09 PM - Paolo Cavallini

- Status changed from Open to Feedback

- Description updated

Still true in QGIS3. Unclear status: it's a lot of work, and possibly we should better concentrate in native QGIS tools.