Towards an automated label placement service

OGRS2009
Olivier Ertz - olivier.ertz@heig-vd.ch

Nantes, July 2009
HEIG-VD school

2000 students
9 Bachelors:
Economiste
Microtechniques
Systèmes industriels
Génie électrique
Informatique
Télécommunications
Géomatique

Ingénierie des médias
Ingénierie de gestion

200 students …

Ingénieur-e HES des médias
Medieningenieur,-ingenieurin FH
“... all required technologies within the E-content life-cycle: production, processing, analysis, search and diffusion.”

« Polypublishing-crossmedia » concept

Learning Content  Enterprise Content

Geographic Content - GeoSY SIN

Research activities concern the development of web-oriented technologies to favour cartographic content publication through the use of geographical OGC standards and those graphical from W3C”
PAL project

- Cartographic content production with labelling of geospatial features
  - Overlapping problem
  - Limits of greedy algorithms
Who are behind PAL?

- MIS-TIC
  - optimisation algorithms for large problems
  - PAL C++ implementation
- IICT-SYSIN
  - JPAL wrapper and extJPAL extension for gvSIG
  - Document study for a service-oriented approach
- G2C
  - know-how about labelling rules
  - quality evaluation
Purpose 1: PAL/JPAL library

- Push forward combinatorial optimisation approaches
  - New algorithms for a better quality of result
  - Within a satisfying laps of time to process
  - Useful even for interactive navigation
- PAL/JPAL library
  - Native for C++ tribes + JNI wrapper for Java tribes
  - More than algorithms, a ready-to-use library for Geospatial Free and Open Source communities, client-side or server-side
PAL functionalities

• Multi-layers labelling for points, lines, polygons

• Several labelling parameters:
  • Layer priority: to decide which of two conflicting labels from different layers to display
  • Obstacle: to avoid labels to overlap features
  • Labelling arrangement:
    - Point: Around point
    - Line: Around line, Over line
    - Polygon: Horizontal, Free, Over perimeter, Around centroid, Around perimeter
<table>
<thead>
<tr>
<th></th>
<th>Points</th>
<th>Lines</th>
<th>Polygons</th>
</tr>
</thead>
<tbody>
<tr>
<td>$p_{\text{point}}$</td>
<td><img src="image1" alt="Points" /></td>
<td>na</td>
<td><img src="image2" alt="Polygons" /></td>
</tr>
<tr>
<td>$p_{\text{line}}$</td>
<td>na</td>
<td><img src="image3" alt="Lines" /></td>
<td></td>
</tr>
<tr>
<td>$p_{\text{line_around}}$</td>
<td>na</td>
<td><img src="image4" alt="Lines" /></td>
<td></td>
</tr>
<tr>
<td>$p_{\text{horiz}}$</td>
<td>na</td>
<td>na</td>
<td></td>
</tr>
<tr>
<td>$p_{\text{free}}$</td>
<td>na</td>
<td>na</td>
<td></td>
</tr>
</tbody>
</table>
Library API: how to talk to PAL

• For each layer to label, PAL needs:
  • Access to spatial features (label / obstacles)
  • Precomputed bounding-box of each label
  • Priority level
  • Labelling arrangement
  • Symbol radius of each points, width of each line
• Map global properties: extent, scale, resolution
• Return a list of label position to display
Algorithm overview – step 1

• Generate candidates
Algorithm overview – step 2

- Optimization
  - POPMUSIC + TABU search + ejection chains
Purpose 2: extJPAL

- A concrete integration for GIS desktop client
  - "proof of concept": how to couple it with an underlying symbology object model
  - improvements of PAL API by iterations
- A showcase:
  - from algorithms to a concrete product
  - extJPAL, an extension for gvSIG 1.1.x
  - play a demo...
Put your hands on

- Technical documentation and source code:
  - http://geosysin.iict.ch/PAL
- extJPAL install packages for **Linux** and **Windows** from « Catalogo de productos de gvSIG »
  - www.gvsig.org/web/plugins/downloads/pal-automate
- Status: experimental
The « opensource effect »

- Strong interest at FOSS4G2008:
  - GeoServer, GRASS, MapWindow, QuantumGIS

- QGIS:
  - after a « Code sprint » at Cape Town, work has been done towards a QGIS-PAL integration *
  - currently, « Label Placement GSoC 2009 »
    - mid-term - labeling with PAL integrated into QGIS with some essential options
    - final - dynamic/static labeling, GUI for manual editing of labels, new options
QGIS integration
Purpose 3: wpsPAL

- Server-side approach 1:
  - embedded inside a web map server
  + OGC standards compliance
    - Example: GeoServer, OGC compliant server
    - Symbology Encoding (SE): rules for map rendering
SE for input parameters

- **SE:**TextSymbolizer is for labeling
  - adjust label placement (Anchor, Rotation, ...)
  - nothing to configure an underlying clever algorithm like PAL would be

- **GeoServer « SE extension »**
  - to control its internal labelling renderer

- **VendorOption parameters**: 
  - avoid overlaps
  - priority, spaceAround, group, autoWrap, forceLeftToRight
Approach 2: service-oriented

- On-demand labelling used for map production (loosely-coupled)
  - "given some features + labelling wishes, where and how do I have to draw labels for high-quality map?"
- Specific processing result from a labelling algorithm like PAL
  - how to control the service?
  - how to formalize the result?
  + OGC standards compliance
How to control wpsPAL?

- independant, no link with a web map service
- no need of SE:VendorOption

Standardized web service

- SOAP, REST, or OGC Web Processing Service

OGC WPS: "designed to standardize the way that GIS calculations are made available to the Internet."

PAL = GIS calculation: use of WPS seems relevant
wpsPAL : service chaining

Client Vector Layer

OGC Feature Server

labeling parameters

OGC GML (direct convert)

wpsPAL

OGC GML (WFS query)

OGC SE

OGC Portrayal Service

map
wpsPAL : output

- "OGC formalize" the result
- Symbology Encoding : positions and rotations of each label to draw (?)
- SE:InlineFeature, a layer of point-labels

```xml
<InlineFeature>
  <FeatureCollection>
    <featureMember>
      <Label>
        <LabelID>1</LabelID>
        <LabelName>Paris</LabelName>
        <pointProperty>
          <gml:Point>
          </gml:Point>
        </pointProperty>
      </Label>
    </featureMember>
  </FeatureCollection>
</InlineFeature>
```
Case 1: one rule for all point-labels (only positions matter)

```xml
<Rule>
  <TextSymbolizer>
    <Label><ogc:PropertyName>LabelName</ogc:PropertyName></Label>
    <Font>
      <CssParameter name="font-family">Arial</CssParameter>
      <CssParameter name="font-style">Normal</CssParameter>
      <CssParameter name="font-size">14</CssParameter>
    </Font>
    <Fill>
      <CssParameter name="fill">#000000</CssParameter>
    </Fill>
  </TextSymbolizer>
</Rule>
```
• case 2: one rule for each point-labels

```xml
<Rule>
  <Filter xmlns="http://www.opengis.net/ogc">
    <PropertyIsEqualTo>
      <PropertyName>LabelID</PropertyName>
      <Literal>1</Literal>
    </PropertyIsEqualTo>
  </Filter>
  <TextSymbolizer>
    <Label><ogc:PropertyName>LabelName</ogc:PropertyName></Label>
    <LabelPlacement>
      <PointPlacement>
        <Rotation>12.0</Rotation> // chacun sa rotation
      </PointPlacement>
    </LabelPlacement>
    <Font>
      <CssParameter name="font-family">Arial</CssParameter>
      <CssParameter name="font-style">Normal</CssParameter>
      <CssParameter name="font-size">14</CssParameter>
    </Font>
    <Fill><CssParameter name="fill">#000000</CssParameter></Fill>
  </TextSymbolizer>
</Rule>
```

... // une règle par point-label
Conclusions

- Service-oriented wpsPAL:
  - offers a label placement processing service dedicated to high-quality map production
  - inputs in respect of OGC WPS interface
- Result as a SE fragment of rules applied by the rendering application to produce the final map
  - case 2: requires a BaseSymbolizer

- Join us to give a try for an implementation ...
Thanks for your attention! Some questions?

Towards an automated label placement service

Olivier Ertz – IICT/SYSIN
olivier.ertz@heig-vd.ch
http://geosysin.iict.ch

University of Applied Sciences, Switzerland