

Towards Web Services Dedicated to Thematic Mapping

Abson Sae-Tang, Olivier Ertz

Introduction

Open standards favor interoperability of systems, and Open Geospatial Consortium (OGC) is the group that specifies the standards that allow geographic systems to interoperate. Among most known specifications, OGC defines the Web Map Server (WMS), the Web Feature Server (WFS), the Geographic Markup Language (GML), and the Styled Layer Descriptor (SLD) standards to solve the issue of spatial data sharing and interoperability. The project described in this paper puts the emphasis on the use of SLD to favor interoperability of geographic systems for thematic mapping.

What is SLD ?

SLD stands for Styled Layer Descriptor, it is an encoding that extends the Web Map Service specification to allow user-defined symbolization of feature data. It allows users (or systems) to determine which features or layers are rendered with which colors or symbols. SLD addresses the important need for users (and software) to be able to control the visual portrayal of the geospatial data. FOSS4G applications highly rely on and respect open standards, and SLD is implemented by Geoserver, Mapserver, deegree among many other software.

Next version of this standard is known as the Symbology Encoding Specification which is still in progress. The ability to define styling rules requires a styling language that the client and server can both understand. Symbology Encoding provides this language, while the SLD profile of WMS enables application of Symbology Encoding to WMS layers using extensions of WMS operations.

Is SLD ready for thematic mapping ?

Concretely, SLD is a useful and complete specification for styling your maps. For each layer you could say "color all my line features in blue", or "make all polygon borders black, and the insides pale yellow",

or even "use little triangles for all my points". But you can also define even more complex styles. You can define the style rules based on attributes of the features in a layer. In a roads data set, you can style highways with a three-pixel red line, style four-lane roads in a two-pixel black line, and style two-lane roads in a one-pixel black line, thanks to an attribute that indicates the type of road.

But is it as good for thematic mapping? That is, for choropleth maps (Figure 1a), proportional symbols (Figure 1b), overlaid symbols (Figure 1c), juxtaposed symbols (Figure 1d), pie charts (Figure 1e), bar/histogram charts (Figure 1f), etc. These charts mainly make use of style rules based on feature attributes to transform statistical data to a graphical representation on the map. So, this project is first a study on how far we can go with SLD for thematic mapping.

With the current specification of SLD, it's more or less possible to describe thematic maps, but there are some drawbacks :

Choropleth map : one rule with a filter (class boundaries) per class, each rule having its polygon symbolizer with the fill color to apply.

Proportional symbols : a point symbolizer with a built-in graphic mark like circle, a fill color, and its size controlled by a data attribute.

Bivariate symbols : a mix of the filter and color of a choropleth map, and the point symbolizer of proportional symbols. Both size of the symbol and fill color are controlled by two data attributes.

Overlaid symbols : two rules with a filter for the rendering order (the fact that the smallest symbol has to be in front of the greatest). And two point symbolizers per rule, each with its size controlled by a data attribute.

For juxtaposed symbols, pie chart, and bar/histogram chart maps, it starts to be more complex and even unpleasant to describe them with SLD. Concretely, how do we hang two juxtaposed symbols on a unique point or centroid? Same for slices of a pie chart or bars of a histogram. One solution would be to use InlineFeature (from SLD 1.1) to draw the symbols, bars of the histogram for example. But unfortunately, InlineFeature uses GML to create temporary features, not graphics! So this is not accept-

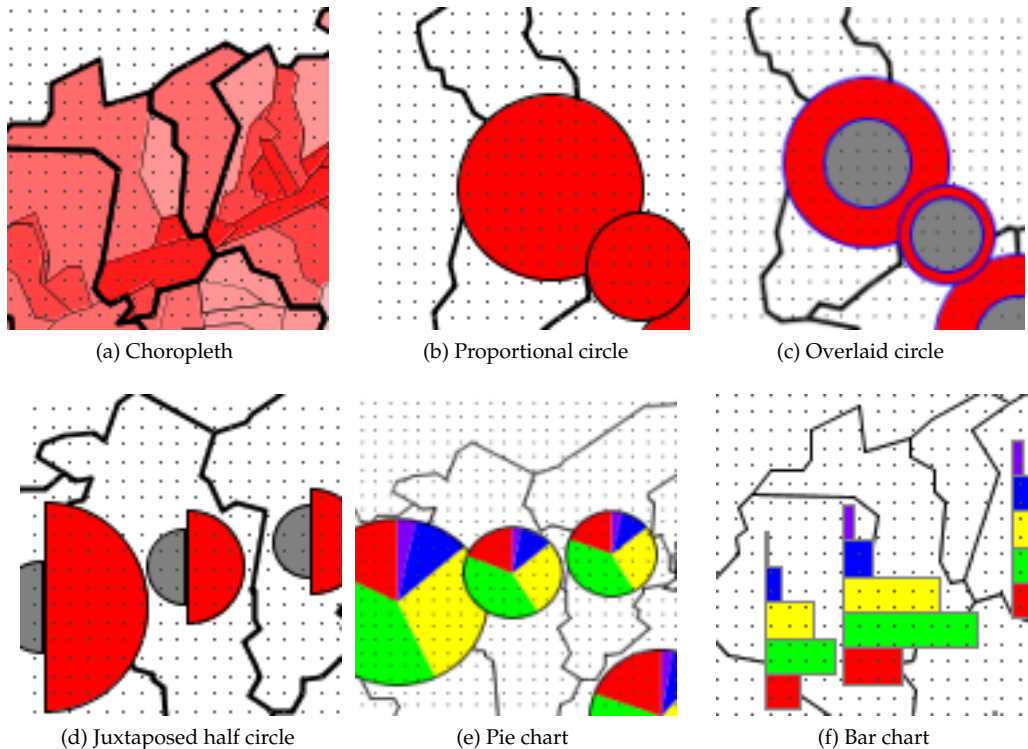


Figure 1: Thematic maps

able. The second idea would be to use a kind of third party application for producing pictures of the symbols to represent points with external graphic links. Nonetheless if this is more acceptable, it is too verbose. Because one rule with a filter per symbol is required (given 50 points, SLD will need 50 rules).

Towards an extension dedicated to thematic mapping

This initial study on how far we can go with SLD lead us to the idea of extending the symbology encoding of SLD. We call it SLD-T (even if it has nothing to do with WFS-T). Such an extension wants to extend the grammar in order to introduce specific terminology the thematic mapper is used to, ease the description of thematic maps, and reduce verbosity and redundancy.

Basically the idea is to create an abstract `ThematicSymbolizer` (like a `PointSymbolizer` is) that will be included in the SLD schema by extending the existing `Symbolizer` element from a `Rule`, with specializations for each kind of thematic map (Figure 2).

CategoryThematicSymbolizer: for maps with classifications (i.e. choropleth and bivariate map).

This symbolizer is built-on `ThematicCategory` elements to describe the classification type (by unique value, by range value, etc.).

SimpleThematicSymbolizer: for maps without classification (i.e. proportional symbols). It is a simple wrapper of “standard” symbolizers to let them inherit useful generic elements from the `ThematicSymbolizer` like symbol priority and placement (see `MultiThematicSymbolizer`).

MultiThematicSymbolizer: to depict several thematic symbols per feature (i.e. overlaid and juxtaposed symbols). A `BaseSymbolizer` is used to group common rendering element (like `Stroke` or `WellKnownName`) and avoid redundancy. For overlaid symbols, the rendering process order is managed by the `Priority` element which can be controlled by a feature attribute. For juxtaposed symbols, the `PointPlacement` (inspired from the `TextSymbolizer`) allow to define an `AnchorPoint` and a `Displacement` for each symbol.

ChartThematicSymbolizer: for chart symbols (i.e. pie and bar charts). A `ThematicMark` (following the idea of graphic `Mark`) is used to specify the chart type (pie or bar). And a `ChartParts`

element to describe the bars and slices composing the complex chart. As all ThematicSymbolizer can be rotated, one can also create a histogram.

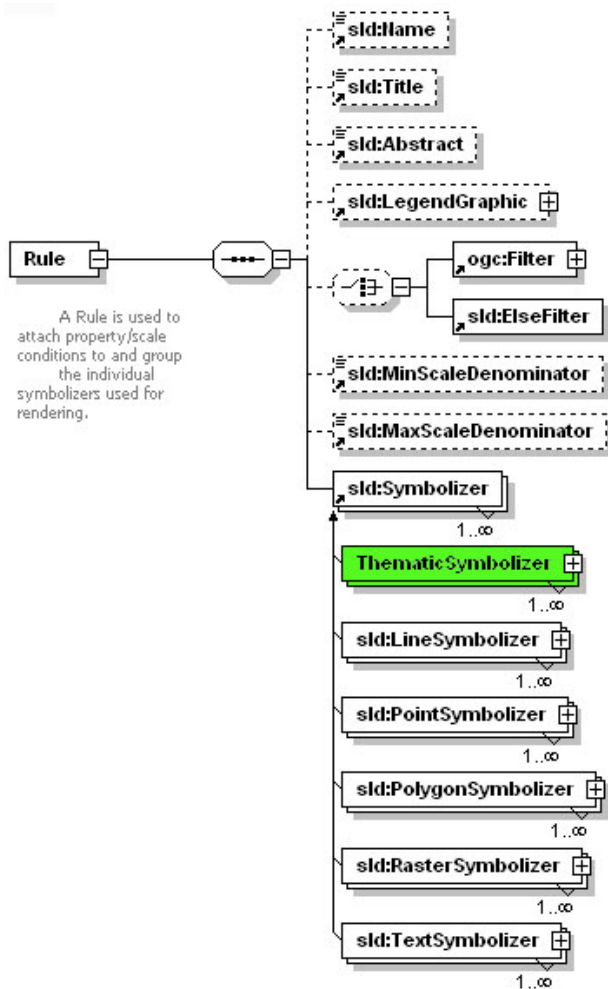


Figure 2: Rule with a new symbolizer

This is a brief summary about the extension. Complete XSD schema and document specification with examples are available online.¹²

SLD specification was originally meant for WMS. The user can define a SLD document and send it to a WMS server, and it returns the layer after applying the style you provided. But it's also useful as part of an OWS compliant desktop application. It could take a SLD file and apply it to a WFS response in GML that it receives. So, it makes sense to use such a styling specification server-side and client-side.

Consider this basic scenario: a cartographer or thematic mapper designing a nice and meaningful

map on its SLD compliant desktop application, and as soon as the map is ready, he pushes it on a WMS server to share it. He will push the data and the styling. If interoperability is first about sharing data, it is finally about sharing maps with the appropriate styling for visualization, and the user wants also to share thematic maps.

Conclusion

The initial study of this project tries to depict how far and how to use the specification for thematic mapping further than just for basic styling. Moreover, to enable SLD for complex thematic mapping, the project describe a solution with an extension of the symbology encoding. The extension has the aim to introduce specific terminology thematic mapper is familiar with, ease the description of thematic maps, and reduce verbosity and redundancy the use of "standard" SLD could produce.

As a proof of concept, a first implementation has been started on GeoTools, only about juxtaposed symbols. It was quite easy, because of an initial strong support of SLD and rendering model. The advantage of using GeoTools is that the library is used server-side for Geoserver but also client-side with uDig. But there are many FOSS platforms on which a complete implementation could be done, and no decision has been made. Future plans include an implementation but also a real use case. Notice, to have a really complete implementation, the rendering of legend graphic is mandatory.

SLD is more then ever a key element for Spatial Data Infrastructure, and its acceptance is probably crossing a step as we see more and more projects building SLD renderer and especially editor to ease user's life to create SLD (GeoServer, Mapbender, OpenLayers, etc.). Finally, at the so called "web thematic mapping" side, it is good to see MapServer now able to create complex thematic maps with pie and bar charts, and also client-side application like CarThema5 (based on gvSIG), JumpChart (based on JUMP), OrbisGIS (based on GeoTools and GDMS) are ready to go for thematic mapping. This is all good news going the right way.

Acknowledgements

The authors wish to thank their partners at [MicroGIS SA](#) and [Camptocamp SA](#) for their collaboration

¹²XSD schema and document specification: <http://geosysin.iict.ch>

on parts of the research behind this paper.

Bibliography

- [1] A. Sae-Tang, and O. Ertz, *Towards Web Services Dedicated to Thematic Mapping*, HEIG-VD, IICT/geo.SYSIN <http://geosysin.iict.ch>.
- [2] OGC Styled Layer Descriptor Specification, <http://www.opengeospatial.org/standards/sld>.
- [3] OGC Symbology Encoding Specification, <http://www.opengeospatial.org/standards/symbol>.
- [4] M.D. Teixeira, R. de Melo Cuba, and G.M. Weiss, *Creating Thematic Maps with OGC Standards Through the Web*, CPqD Telecom & IT Solutions, <http://www.gmldays.com/papers/Teixeira.html>.
- [5] M.A. Manso, A. Maldonado, R. Hernandez, D. Ballari, and J. Moya, *GEOSISMO : Visualization of Events and Seismologic Characteristics in the Internet*, Madrid Polytechnic University, http://redgeomatrica.rediris.es/ICA_Madrid2005/papers/manso.pdf.

Abson Sae-Tang
HEIG-VD IICT/geo.SYSIN
IICT/geo.SYSIN, <http://www.iict.ch>
`musy-abson.sae-tang AT heig-vd.ch`

Olivier Ertz
HEIG-VD IICT/geo.SYSIN
IICT/geo.SYSIN, <http://www.iict.ch>
`olivier.ertz AT heig-vd.ch`

Interoperability for 3D Geodata

Experiences with CityGML and OGC Web Services

Hans Plum and Dr. Markus Lupp

Summary

Storage, processing and visualization of 3D geodata are an important subject in the GIS world even before the leading search engine introduced its globe viewer. Usage of standards of the Open Geospatial Consortium (OGC) open up new possibilities for combination and usage of 3D geodata. First practical experiences show promising results.

Introduction

Processing and visualization of 3D geodata became a common subject during the last years. Some indicators for this are the number of offered software solutions but also the amount of interest for the de-

velopment of CityGML. CityGML is a GML-based exchange format for three dimensional digital city models, that is already implemented in a number of software products. With the definition of CityGML and application of OGC Web Services for access to and visualization of 3D geodata the areas of 3D geodata processing and Spatial Data Infrastructures (SDI) are converging.

This article is discussing solutions that were realized using technology from the deegree project. The mentioned projects are: "Storage and administration of 3D city models for the cities of Bonn, Berlin and Hamburg", "Visualization of digital terrain models for the Federal Agency for Cartography and Geodesy of Germany", "Realization of a transactional CityGML WFS for the Open Geospatial Consortium" are outlined.