

Solutions for Limitations in Label Placement in OGC Symbology Encoding (SE) Specification

Maxim RYLOV, Alexander ZIPF
*Chair of GIScience Department of Geography,
Heidelberg University, Germany*

Abstract. We are currently witnessing a rapid development of automated cartography through the web mapping technologies. And one of the most important aspects of this cartography is a map labeling. Most modern platforms that can publish spatial data to the web use international standards provided by Open Geospatial Consortium (OGC). Among the other specifications these standards have specific parameters for map labeling. As the current specifications allow only basic parameters to control label placement process, we hereby propose an extension to an international standard provided by OGC that is called Symbology Encoding (SE). The objective of this work is to extend the existing label placement types as well as to describe new ones. This extension was implemented within a new web mapping platform (<http://OpenMapSurfer.uni-hd.de>) that is based on several OGC specifications.

Keywords. label placement, automated cartography, OGC, Symbology Encoding (SE)

Introduction

At present it is impossible to imagine modern cartography without computers and special software that represents a web mapping. Currently we can observe a rapid development of automated cartography through the platforms that can publish spatial data to the web using the standards of Open Geospatial Consortium (OGC) such as Web Map Service (WMS) [1], Styled Layer Descriptor (SLD) [2], Symbology Encoding (SE) [3] and so on.

New technologies as well as new standards try to solve the same basic problem of cartography as topographers and cartographers 200 years ago tried to cope with. This problem is positioning labels on maps in relation to the various aspects of the map production and the types of geometric features. The citation of Eduard Imhof [4] “Good form and placing of type make the good map” fully describes this problem. To get a good looking map using modern technologies one should use some specification that can help to determine user styles for label placement. To solve this task one can use special language that is called Symbology Encoding (SE) specification from OGC. This language can be used to portray output of Web Map Servers [1], Web Feature Servers [5] and Web Coverage Servers [6]. Symbology Encoding gives us a fine-grained control of the graphical representation but it still has some limitations. Earlier

some authors [7, 8, 9, 10] indicated that in SLD and SE in particular for thematic cartography.

There are three different label placement tasks (designations, Imhof [4]) identified in cartography: position designations (such as cities or mountain peaks), linear designations (such as rivers or roads), and areal designations (such as parks or lakes). Currently SE has a specification to define and control only two of them: position and linear. For areal designations it is assumed that they can be generalized to the point or to the linestring [3].

In order to allow more advanced control over the process of map labeling it is important to extend current SE specification in terms of label placement extension.

Moreover we have some ideas how to extend SE in the areas of text presentation and label overlapping, that have a relationship to this paper. For example, there is an extension in the *TextSymbolizer* [3] by applying more typographic features and support for international text [11], [12] in styling and rendering of text labels. Another extension in *TextSymbolizer* concerns the control of overlapping of text and symbol labels to each other as well as to the border of the output area. All mentioned above extensions are related to another work and could be discussed in future papers in more details.

The objective of this work is to propose an extension to one of the current label placement types as well as to define two new ones.

1. Types of Label Placement

In cartography three different types of designations (label placement) are indicated: position, linear, and areal designation. Within this paper several possible extensions of SE that give advanced control over map labeling are presented. In proposed extensions some of designations are presented in two different elements of SE. The necessity of such approach will be described below.

1.1. Label Placement Extension (LPE) Specification

In order to propose an extension of the current OGC Symbology Encoding (SE) specification we are using a similar technique that was done for the 3D extension to the SE specification by Neubauer & Zipf [9]. For that purpose we defined an XML-schema based on SE. It is called the *Symbology Encoding Label Placement Extension (SELPE)* schema (see Figure 1.).

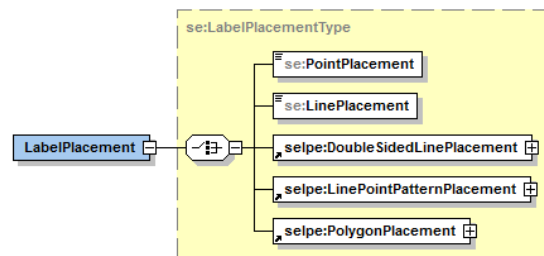


Figure 1. The element LabelPlacement.

This approach allows differentiating already existing elements and the new elements. The new XML namespace is “selpe”. In this paper we propose two new label placement types: *se:DoubleSidedLinePlacement* and *se:PolygonPlacement*, where the parent element *se:LabelPlacement* is an element of the *se:TextSymbolizer* that is used for styling text labels.

1.2. LinePlacement

The *se:LinePlacement* is a part of SE Specification and its behavior is to draw of the label along the line. For this element we propose some additional elements to have more control over the labeling process, to increase label density by drawing more labels and at the same time not disturbing the legibility and clarity of the map. Please note (see Figure 2) that the new elements have a namespace “selpe”. The detailed description of all new elements is cited below; for the elements from namespace “se” (see [3] chapter 11.4.4). The *LinePlacement* has the following XML schema definition:

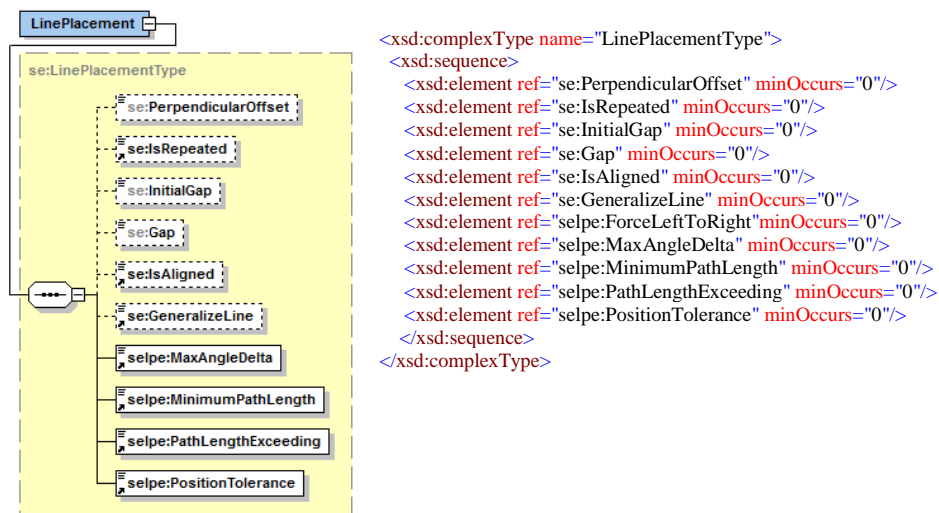


Figure 2. The element LinePlacement.

The *selpe:ForceLeftToRight* defines a parameter that can control whether to draw a label in such way that it can be read, meaning the label does not always follow the orientation of the line, but sometimes it can be flipped by 180° instead to allow normal reading. In some cases it is necessary to suppress flipping, for example when the label is a directional arrow that shows one way direction of the street.

The *selpe:MaxAngleDelta* describes the maximum angle, in degrees, between two subsequent characters in a curved label. Large values can lead to appearing of disconnected words or overlapping characters.

The *selpe:MinimumPathLength* allows to set minimum length of a linestring that is required for the label placement. In other words if the length of a linestring is less than required minimum length then labeling of the linestring is skipped.

The *selpe:PathLengthExceeding* represents a parameter that defines an maximum exceeding value on which label can exceed a linestring. This parameter can be used in cases when the length of a linestring is less than the length of a label.

The *selpe:PositionTolerance* parameter controls the displacement of the label along a line from the point A that was computed according to the values of *se:IsRepeated*, *se:InitialGap* and *se:Gap*. If the value of this parameter is greater than 0 then the next possible label position is searched within *selpe:PositionTolerance* pixels from the point A.

Please note that the described above elements such as *selpe:MinimumPathLength*, *selpe:PathLengthExceeding* and *selpe:PositionTolerance* are measured in units of measure (*uom*). More information about units of measure can be found in [13].

It was not possible to include all possible examples for new elements in this paper, but we were able at least to present one. This example can be seen in Figure 3. In this example we tested only two new elements *selpe:PathLengthExceeding* and *selpe:PositionTolerance*. Using these elements we successfully labeled one more street (street Mönchbergsteige, see (b)) that had the length less than the length of the label.



Figure 3. (a) – PathLengthExceeding = 0, PositionTolerance = 0;
 (b) – PathLengthExceeding = 4, PositionTolerance = 2,
 where the units of measure are pixels.

Example in figure 3 was made with Volunteered Geographical Information [14, 15] data that were provided by OpenStreetMap project [16] contributors.

1.3. DoubleSidedLinePlacement

The element *selpe:DoubleSidedLinePlacement* was partially derived from *se:LinePlacement*. This element describes label placement that it is needed to label a linestring with two different text values on both sides of it (see Figure 4.). It consists of several supplementary elements such as *selpe:MaximumTextGap*, *selpe:ValuesSeparator* and following elements *se:PerpendicularOffset*, *se:IsAligned*, *se:GeneralizeLine* were omitted. The structure of *selpe:DoubleSidedLinePlacement* can be seen in the following Figure 5.



Figure 4. Example of DoubleSidedLinePlacement.

The proposed new label placement type can be used to label, for example, country or region boundaries.

The *selpe:MaximumTextGap* element defines the maximum space between two text labels that is computed as two distances from the center line (violet line in Figure 4) to the label text. The center line of the linestring segment can be computed using linear regression model [17]. Through varying the values of this element we can control the degree of intersection of the label text with the line. The *DoubleSidedLinePlacement* has the following XML schema definition:

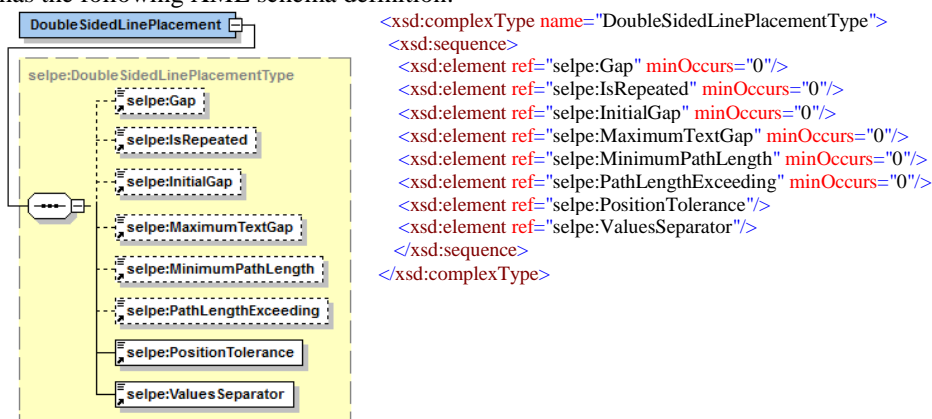


Figure 5. The element DoubleSidedLinePlacement.

The *selpe:ValuesSeparator* element specifies the string that is used to split text-label content of the element *se:Label* (see SE specification 11.4.2) into two text-strings. Here is a usage example:

```

<Label>
  <ogc:Add>
    <ogc:PropertyName>country_left</ogc:PropertyName>
    <ogc:Add>
      <ogc::Literal>^$^</ogc:PropertyName>
      <ogc:PropertyName>country_right</ogc:PropertyName>
    </ogc:Add>
  </ogc:Add>
</Label>

```

and *selpe:ValuesSeparator* has a value of “ \wedge ”.

The description of other elements of *selpe:DoubleSidedLinePlacement* can be found in 1.3.

1.4. *LinePointPatternPlacement*

The element *selpe:LinePointPatternPlacement* was partially derived from two label placement types: *se:PointPlacement* and *se:LinePlacement*. This element describes label placement that it is needed to label a linear geometry according to the predefined pattern. For example one of the scopes of this type is the labeling of geological formations (see Figure 7). This type consist of two new elements: *selpe:Pattern* and *selpe:PatternScale*. Please note that this type of placement was designed to work only with symbol or graphic based labels. The *LinePointPatternPlacement* has the following XML schema definition:

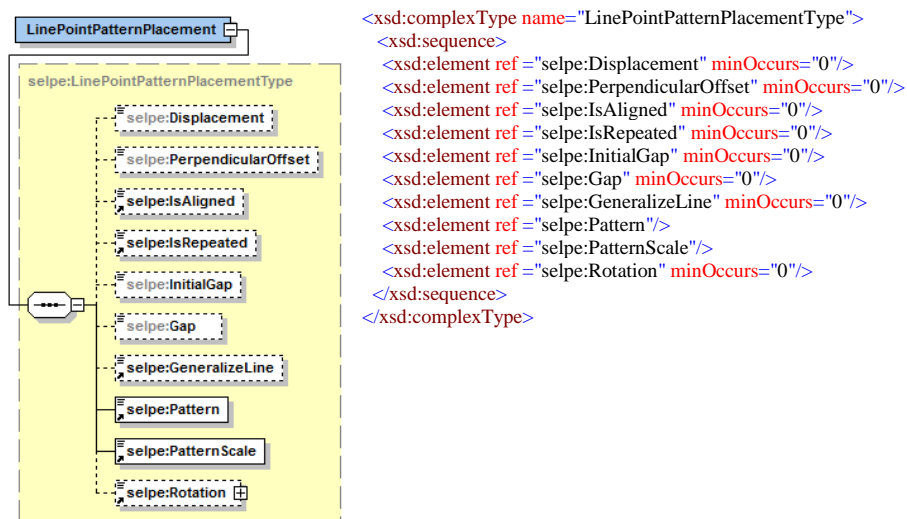


Figure 6. The element *LinePointPatternPlacement*.

The *selpe:Pattern* element represents an array of floats specifying the distance between label symbols within one group. *selpe:Gap* gives the distance between groups. Every element of *selpe:Pattern* is given in uoms.

The *selpe:PatternScale* element defines a scale of the pattern. Such approach is useful when it is just needed to change the scale of the group without changing relative distance between each of them.

Two last elements are offering a large variety of different designs. The usage of these elements can help to cope with complex labeling tasks.

The *selpe:Rotation* element gives the rotation of a label symbol (graphic) in the clockwise direction about its center point in decimal degrees. Negative values mean counter-clockwise rotation. The default value is 0.0 (no rotation). If *selpe:IsAligned* value is true and *selpe:Rotation* is not 0.0 then the result rotation angle will be a sum of the value of *selpe:Rotation* and the angle of the line slope to which belongs current pattern point.

All other elements of *LinePointPatternPlacement* have the same meaning as described in [3].

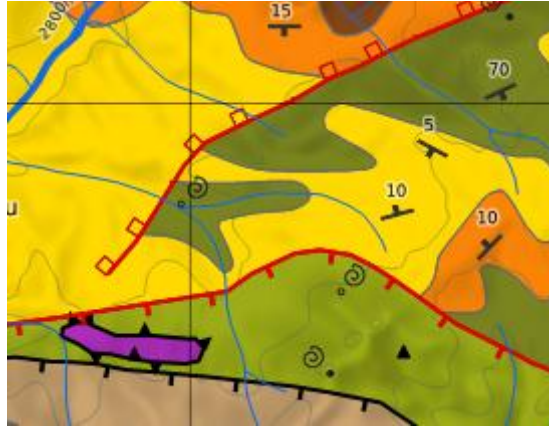


Figure 7. A map of geological formations using a `LinePointPatternPlacement` (see the red squares along the red line).

2. Conclusions and Future Work

In spite of the rapid development of automated cartography through the web mapping technologies, it appears that the international standards proposed by Open Geospatial Consortium still have some limitations for example in the fields of thematic mapping [8, 9] or label placement. In this paper we have proposed an extension to solve the limitations in the field of label positioning. More precisely, we have proposed an extension of `se:LabelPlacement` element of the Symbology Encoding specification (see [3] chapter 11.4.4).

There are two main reasons of this extension. The first one is increasing the label density on the map by giving more control over the label positioning with the help of new options and the second one is differentiation of the existing label placement types into new ones. This approach gives new possibilities to label the same geometric feature with different strategies and with different text-label content.

Due to the limited amount of pages it was not possible to describe some other possible label placement types:

selpe:PolygonPlacement – it is assumed that this element will give more options to label polygons in many ways, instead of using `se:GeneralizeLine` in `se:LinePlacement`. The behavior of `se:LinePlacement` is to draw of the label along the line. It would be great to have such placement type that could cope with labeling of the polygon using as much as possible the form and the extent of the mapped area.

At the moment all extensions to Symbology Encoding specification concerning label placement still have open issues that have to be discussed and deeper researched further.

In this paper we proposed an extension to SE that was implemented within a new platform for publishing spatial data to the web. It is called MapSurfer.NET. One of the purposes of this platform is solving some basic problems of non-manual cartography. In order to meet the design goal of a platform independent application, the software has been implemented in a component-based fashion. The platform was implemented in C#, which is a modern, object-oriented programming language and is the core language of the Microsoft .NET framework. This language was chosen for a reason. Metadata

annotations in C# provide a powerful way to extend the capabilities of a programming language and the language runtime. In this work metadata and new features of modern programming language have been used for implementing the described extension to Symbology Encoding. The platform also gives a set of interfaces for interaction with other GIS platforms and systems, and it can be easily extended in the future.

The result of using of some proposed extensions of SE within MapSurfer.NET can be seen in the web. A tiled map service based on OpenStreetMap data with detailed styling was successfully published on the Internet (<http://OpenMapSurfer.uni-hd.de>).

Future plans is to carry out the implementation of *selpe:PolygonPlacement* as well as the analysis of the relationship between the elements of different label placement types.

References

- [1] J. de la Beaujardiere, OpenGIS Web Map Server Implementations Specification, OpenGIS Implementation Specification, 06-042, Open Geospatial Consortium Inc., 2006.
- [2] M. Lupp, Styled Layer Descriptor profile of the Web Map Service Implementations Specification, OGC Implementation Specification, 05-078r4, Open Geospatial Consortium Inc., 2007.
- [3] M. Müller, Symbology Encoding Implementation Specification, OpenGIS Implementation Specification, 05-077r4, Open Geospatial Consortium Inc., 2006.
- [4] E. Imhof, Positioning names on maps. *The American Cartographer*, **2(2)** (1975), 128-144.
- [5] P.A. Vretanos, OpenGIS Web Feature Service 2.0 Interface Standard, OpenGIS Implementation Standard, 09-025r1, Open Geospatial Consortium Inc., 2010.
- [6] P. Baumann, OGC WCS 2.0 Interface Standard – Core, OpenGIS Interface Standard, 09-110r3, Open Geospatial Consortium Inc., 2010.
- [7] T. Brinkhoff, Towards a Declarative Portrayal and Interaction Model for GIS and LBS, 8th Conference on Geographic Information Science (AGILE 2005), Estoril, Portugal, (2005), 449-458.
- [8] L. Dietze, A. Zipf, Extending OGC Styled Layer Descriptor (SLD) for Thematic Cartography – Towards the ubiquitous use of advanced mapping functions through standardized visualization rules, 4th Int. Symp. on LBS and Telecartography, 2008.
- [9] S. Neubauer, A. Zipf, Suggestions for Extending the OGC Styled Layer Descriptor (SLD) Specification into 3D – Towards Visualization Rules for 3D City Models. Proc. Of. Urban Data Management Symposium, Stuttgart, Germany, UDMS, 2007.
- [10] I. Iosifescu, M. Hugentobler, L. Hurni, Cartographic extensions to OGC Web Map Services – Specifications for a Map and Diagram Service. International Cartographic Conference, 2007.
- [11] M. Davis, Unicode Bidirectional Algorithm, Unicode Standard Annex #9, Unicode Consortium, 2010.
- [12] A. Heninger, Unicode Line Breaking Algorithm, Unicode Standard Annex #14, Unicode Consortium, 2010.
- [13] J. Bobbitt, Units of Measure Use and Definition Recommendations, OpenGIS Recommendation Paper, 02-007r4, Open GIS Consortium Inc., 2002.
- [14] S. Elwood, Volunteered geographic information: future research directions motivated by critical, participatory, and feminist GIS, *GeoJournal*, **72** (2008), 173-183.
- [15] M.F. Goodchild, Citizens as sensors: the world of volunteered geography, *GeoJournal*, **69** (4) (2007), 211–221.
- [16] F. Ramm, J. Topf, S. Chilton, OpenStreetMap: Using and Enhancing the Free Map of the World, UIT Cambridge, 2010.
- [17] Å. Björck, Numerical methods for least squares problems, Philadelphia: SIAM, 1996.