Redesign of OGC Symbology Encoding standard for sharing cartography

Erwan Bocher[1], Olivier Ertz[2]

[1] Lab-STICC CNRS UMR 6285, University of South Brittany, Vannes, France

[2] HEIG-VD Media Engineering Institute, University of Applied Sciences and Arts Western Switzerland, Yverdon-les-Bains, Vaud, Switzerland

Corresponding Author:

Olivier Ertz[2]

Avenue des Sports 20, Yverdon-les-Bains, Vaud, CH-1401, Switzerland

Email address: olivier.ertz@heig-vd.ch

Abstract

Despite most Spatial Data Infrastructures are offering service-based visualization of geospatial data, requirements are often at a very basic level leading to poor quality of maps. This is a general observation for any geospatial architecture as soon as open standards as those of the Open Geospatial Consortium have to be applied. To improve this situation, this paper does focus on improvements at the (inter)operability side by considering standardization aspects. We propose two major redesign recommendations. First to consolidate the cartographic design knowledge at the core of the OGC Symbology Encoding standard. Secondly to build the standard in a modular way so as to be ready to host upcoming cartographic requirements.

Thus, we start by defining the main portrayal interoperability use cases that frame the concept of sharing cartography. Then we bring to light the strengths and limits of the relevant open standards to consider in this context. Finally we paint a set of recommendations to overcome the limits so as to make these use cases a reality.

Even if the definition of a cartographic-oriented standard is not able to act as a complete cartographic design framework by itself, we argue that pushing forward the standardization work dedicated to cartography is a way to share and disseminate good practices and finally to improve the quality of the visualizations.

1. Introduction

The constant evolution of information and communication technologies and the parallel development in geospatial technologies led to new methods and techniques to access, transform and analyze shared geospatial data. Based on these technologies, many regional, national and international initiatives have setup well-defined access policies to promote the arrangement of Spatial Data Infrastructure (SDI) as a new paradigm in geospatial data handling to encourage data sharing across different communities and various applications. It extends desktop GIS (Craglia, 2010), where data collected by other organizations can be searched, retrieved and manipulated for several usages (Tóth et al., 2012).

Currently, several SDI initiatives are particularly well implemented for the harmonization of data discovery and simple viewing (also recently processing services). Nonetheless, despite service-based visualization of geospatial data is part of the SDI components, in the case of INSPIRE, the infrastructure for spatial information in Europe, requirements are often at a very basic level according to (INSPIRE Drafting Team, 2014), in section 16. Even with the few dedicated recommendations for portrayal rules by (INSPIRE Drafting Team, 2008) in section A.11, the importance of visualization for transforming geospatial data into useful geographical information is still relatively a concern of second zone. Indeed, as (Field, 2014) points out, the data is simply placed on a map and the viewer is supposed to make sense. “The demand at the moment however is for quantity, not quality and the internet, not the discipline of cartography, is reacting to the demand”. As the official European statistics portal demonstrates, contemporary maps exhibit a serious lacks of knowledges of the language of cartography and many map-makers repeat some basic mistakes. Such as (“Eurostat / Regional Statistics Illustrated”) where the population by NUTS is represented as a choropleth map (e.g. Population on 1 January by NUTS 2 region).

For SDI users, it is somewhat a frustrating cartographic experience given that in parallel hacker cartography (McConchie, 2015) is on the fast-track by means of the openness of geodata and easy to use technologies (e.g. OpenStreetMap-based platforms like (“CartoDB”) and the related open source tools like (“CartoCSS”), etc) showing obvious desire and need of cartography for all types of users: cartographer, map-makers, data artists, journalists, coders (Field, 2014).

(Hopfstock & Grünreich, 2009) underlines that poor map design results in SDI are the consequence of a “too technology- and/or data-driven approach” and proposes improvements by making the cartographic design knowledge explicit and operational. Beside such a relevant proposition at the design level, this paper does focus on improvements at the implementation level by making cartographic portrayal interoperability operational through the improvement of the dedicated open standards. Any SDI is composed of four components – spatial data, organisational framework, access and standards. Indeed, interoperability is key for SDI as interconnected computing systems that can work together to accomplish a common task and the presence of open standards is required to allow these different systems to communicate with each other without depending on a particular actor (Sykora et al., 2007).

The common task in question for this paper does concern the authoring of the symbology of a map. In other words, the common and shared task of a community of users interconnected by interoperable systems which requires the ability to share cartography. Behind the scene, the problematic can be summarized with the following general use case: “Tom does create a map with his X cartographic rendering system; Tom does share it with Jerry; Jerry shall visualize with his Y rendering system the exact same portrayal Tom does see; Jerry shall be able to re-work Toms' map”. Other use cases that come out from this general one are described in the following sections. We argue that it is possible to solve them by means of a common and standardized cartographic model allowing syntactic interoperability between rendering systems that need to share and interpret some "symbology code".

Considering SDI as open participation platforms, we can figure out how such an ability would participate to empower relevant user types with solutions to gain useful geographical information by means of cartographic visualisations. Maps are powerful tools enabling sharing of spatial information and knowledge, but also collaboration through shared creativity and the transfer of skills between producers for better decision-making based on maps of high quality.

Albeit some SDI policies laying down recommandations to make use of standards from Open Geospatial Consortium (OGC) like the Styled Layer Descriptor (Lupp, 2007) and Symbology Encoding (Müller, 2006) standards specifications, it seems they did not bring to reality the above principle and vision, at least in their current states. We might blame the facts that the moving from closed monolithic applications to open distributed systems is still undergoing (Sykora et al, 2007) and that cartography must take effect providing a methodology with a user-oriented approach (Hopfstock & Grünreich, 2009). This paper wants to show how it is also important to have syntactic portrayal interoperability operational with a mature open specification able to standardize the symbology code. We show that the current Symbology Encoding standard does offer limited capabilities for describing cartographic symbolizations and which represent another reason of poor quality maps. Then, while we develop some recommendations to improve this situation through more capabilities to customize the map symbology, we also propose some good practices to favor adoption of the standard by implementors so as to make it really operational for long term. We believe that these propositions should lead to full cartographic portrayal interoperability, going further than basic styles because there is no reason SDI users have to be satisfied with often unsuitable maps.

2. Principles of map design

A map is an intellectual construction that is based on the experience and knowledge of the cartographer to manipulate data input according initial hypotheses and its capacity to play with graphic signs (Slocum, 2009; Tyner, 2010). Therefore many types of map exist and definitions. As (Tyner, 2010) writes "We all know what a map is, but that definition can vary from person to person and culture to culture". However, even if the definition is hard to settle, cartographers have worked to formalize a map syntactics by developing typologies of symbol categories and rules to combine them.

Graphic variables is the first level that are individually manipulated or pile up by the cartographer to create a map (MacEachren, 2004; Slocum, 2009). Applied to elementary geometry (points, lines, polygons), these graphic variables called also visual variables form the cartographic language.

There is no broadly accepted list of visual variables but several cartographic authors agree with a set of commons variables (Carpendale, 2003; MacEachren, 2004; Tyner, 2010): shape, size, hue (color), value, texture, orientation. These visual variables were largely based on the (Bertin & Berg, 2010) classification.

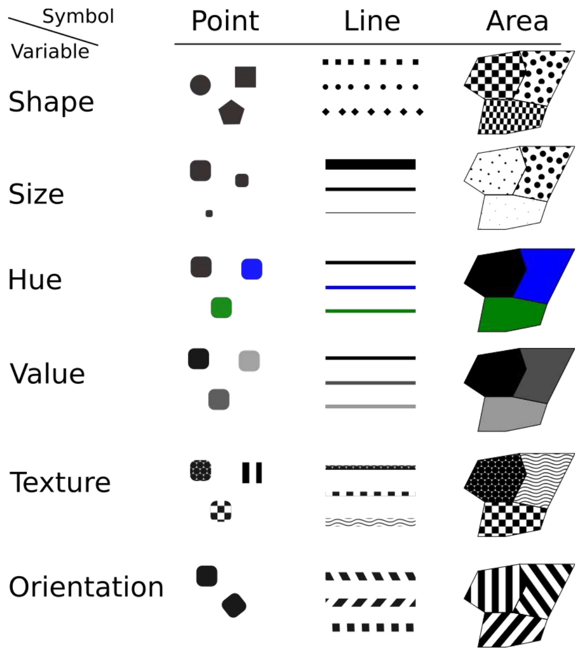

Figure 1: The visual variables of symbols.

The visual variables are used to encode information about point, line and area features (Fig. 1). They are manipulated by cartographers as design embellishments to improve the aesthetic quality of the map or visualization and express efficiently a message (Wood & Fels, 1992). But as Bertin explains the choice of the "right" visual variable entirely depends on the statistical nature of the data (qualitative, quantitative) and on the type of geographical object it is to describe (MacEachren, 2004; Lambert & Zanin, 2013).

These possibilities offer guidelines for cartographer to consider the appropriate map representation. They have permitted the construction of mainstream cartography theory, e.g. a raw data must be represented with proportional symbols and a density of values by an areal classification (ie a choropleth). Indeed, even if creating map is aesthetical exercise it's also a science that must respect best practices to make sure that the representation is accurate.

A de facto set of best practices or map design guidelines based on visual variables has been accepted by the academy of cartographers (McMaster & McMaster, 2002; Montello & Robinson, 2002). This corpus of knowledges is based on a functional and logical map design approach. As (Garlandini & Fabrikant, 2009) remind “Unfortunately, there is very little empirical evidence on the effectiveness and efficiency of these visual variables”.

If this subject is an important issue to understand how and why certain displays are more successful for spatial inference and decision making than others, nonetheless the use of visual variables to compose map symbols is more adapted to define a standard that must be functionally designed and implemented into a SDI or a GIS software.

In this context, we consider a functional definition of a map: a composition of graphic elements (either geometric primitives or pictorial elements) to serve topographic or thematic representation. As Fig. 2 shows, this definition is at the core of the third stage of the cartographic pipeline dedicated to the style rendering. It is also closed to (MacEachren, 2004) and consolidated by (Iosifescu Enescu & Hurni, 2007) that proposes the concept of cartographic domain ontology. Indeed they underlines that the main “types of thematic maps

PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.2415v1 | CC BY 4.0 Open Access | rec: 4 Sep 2016, publ: 4 Sep 2016

(choropleth maps, graduated symbol maps, multi-variable graduated symbol maps, dot density maps, etc.) can be defined only based on basic map semiotics, which in turn are defined with visual variables and/or patterns". This approach matches the aim of the OGC Symbology Encoding standard which is the main standard in question here: it offers a common set of symbol elements to build a style that will be applied on a dataset to render a map. As a consequence, layout design considerations of the map like title, legend, north arrow, scale bar, etc (Peterson, 2009) are out of the scope of this paper.

3. Portrayal interoperability use cases

Given that cartography is able to produce suitable maps as powerful analytical tools for visual thinking to reveal insight in spatial patterns and relations (Kraak, 2006) through human geospatial information processing, we expect cartographic portrayal interoperability to help in making this power operational. It is about to favor the discovery, the sharing and the creation of map products in the context of SDI:

- at first we consider the ability to share prepared ready-to-visualize static maps according to predefined cartographic rules (which drives the map rendering of spatial data offered by the SDI);
- then, we consider the ability to share not only spatial data offered by the SDI but also the cartographic rules to allow the user to manipulate/customize/adapt the map in relation to her(his) requirements.

While the first consideration does match use cases about classical geoportal applications offering the user to discover and explore prepared maps (e.g. map.geo.admin.ch), the second

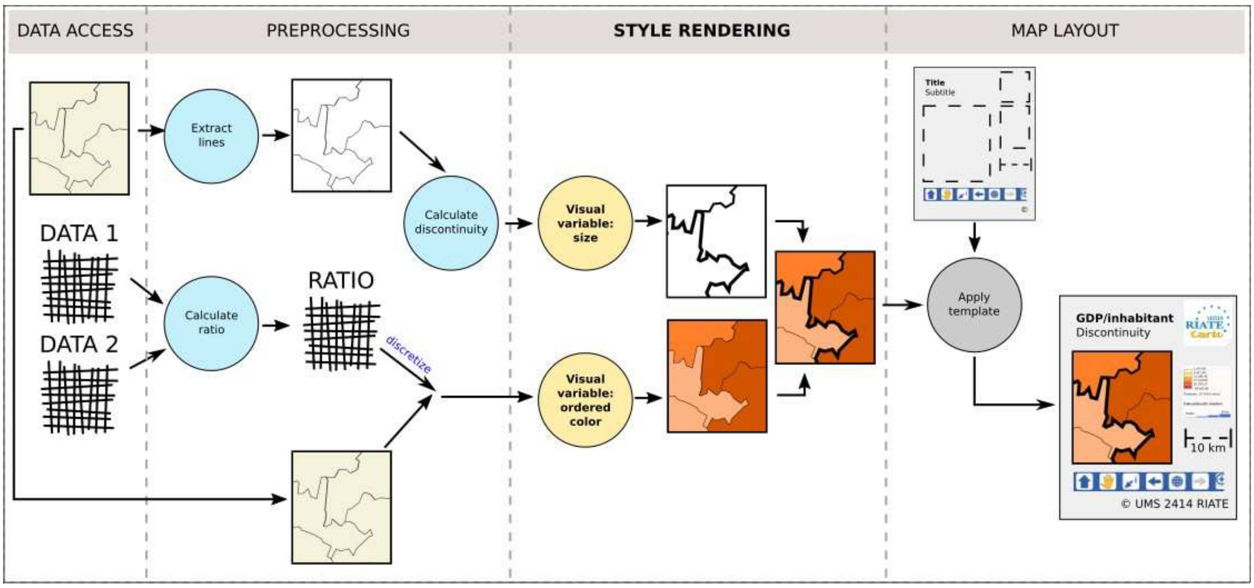

Figure 2: the four stages of the cartographic map design, inspired from (Lambert & Zanin, 2013).

is rather expanding the usage of cartography towards use cases related to the authoring of maps by the SDI users themselves.

Indeed, under the conditions of portrayal interoperability, several other use cases empowering the usage of cartography in the context of SDI can be deduced by extension of the previous second consideration:

- the appropriation use case:
  The classical webmapping use case does offer to the user the resulting image of a map rendering process which is a first step of appropriation of geographical information. This use case does push the appropriation into the core of the everyday users' cartographic tools by getting the underlying spatial data together with the several styles revealing some of their cartographic facets. The cartography is ready to be re-authored by the user inline with her(his) analytical process.

- the reuse-and-combine use case:
  By extension of the previous one, it is about reusing and combining data from different sources hence increasing the production of maps and allowing infinite visual spatial analysis possibilities.

  Typically, the use case does match with the story of the fictive SDI user Mr Tüftel in the Web Portrayal Services book (Andrae et al., 2011). Mr Tüftel wants to unify on the same map the dirty water pipes from his municipality but also the pipes from the municipalities in the neighborhood. These are different data sources he wants to reuse and combine. But also, given the default styles are not the same, he has to re-author the styles or author a common style so as to produce a suitable map ready to be used to decide where to build a shared water treatment plant.

- the do-not-reinvent-the-wheel use case:
  It is about having at disposal style catalogs (or libraries) offering ready-to-use styles, often tailored for specific thematics, e.g. noise mapping color palettes (EPA, 2011). The ability to import such a specialized symbology into users' tool just avoid to reinvent the wheel in the sense of re-creating the style from scratch.

- the participatory use case:
  Finally, this use case is about a wider context which involves several SDI users into a collaborative authoring process. Several users contribute to the creation of a common map, each user having different but required specialized skills to work the analysis but also each using her(his) own software (Ertz et al., 2012).

  We may notice the educational capacity of this use case. Considering a team of people with different levels of skills in cartography, there are offered the chance to share cartographic skills.

Putting together all these sharing cartography use cases does illustrate what we call full cartographic portrayal interoperability. The next part does focus on the technical aspects about how open standards are able to fully meet the conditions of such an interoperability.

4. Limits of open standards for sharing cartography

One of the core concepts for SDIs is interoperability with standards which play key roles to bridge the heterogeneity between the communicating systems. As underlined by (Iosifescu-Enescu, Hugentobler & Hurni, 2010), the use of standardized exchange languages is commonly considered as the most practical solution for interoperability especially when it is required to collate resources, like data, from various systems.

Consequently, (Iosifescu-Enescu, Hugentobler & Hurni, 2010) describes the first level definition of cartographic portrayal interoperability by pointing out that the Web Map Service (WMS) standard (Beaujardiere, 2006) from the Open Geospatial Consortium (OGC) is currently the only widely accepted open standard for map visualization which standardizes the way for Web clients to request maps with predefined symbolization. This ability does match the above use case about the sharing of prepared ready-to-visualize maps.

Just send this simple GetMap request to the Swisstopo WMS server to get a predefined colored map layer to overlay in your webmapping application (Fig. 3)

```
https://wms.geo.admin.ch/?SERVICE=WMS&VERSION=1.0.0&REQUEST=GetMap&FORMAT=image/png
&LAYERS=ch.swisstopo.pixelkarte-pk25.metadata-kartenblatt&SRS=EPSG:3857&STYLES=
&WIDTH=2285&HEIGHT=897&BBOX=174582,5648084,1571851,6196597
```

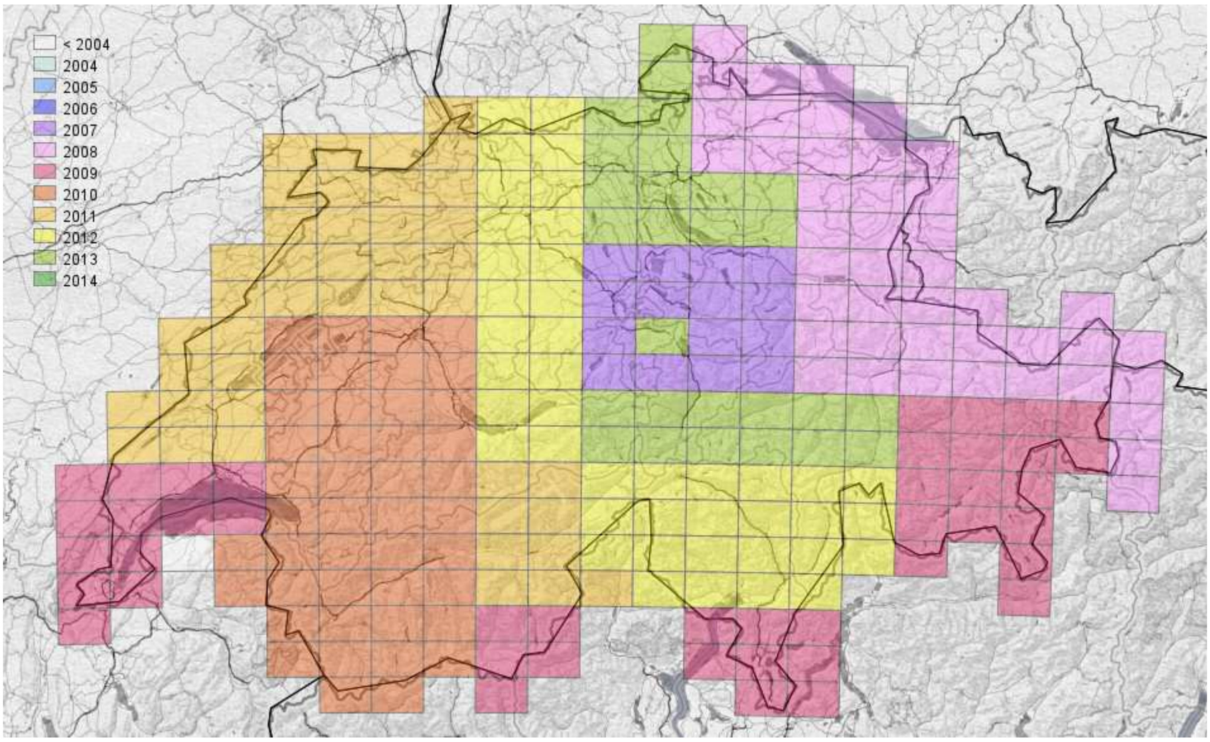

Figure 3: visualization of the grid of map sheets of Switzerland (1:25000) through a default cartographic style showing a choropleth symbology based on the year of edition of the sheet.

Moreover, Iosifescu-Enescu does also underline that the WMS standard combined with the Styled Layer Descriptor profile (SLD) and the Symbology Encoding standard (SE) do establish an open framework able to facilitate decision making through maps.

Just send the WMS request below attached with a SLD style file including below SE instructions to get a visualisation of the grid of map sheets through a symbology authored by the user client (Fig. 4):

```
https://wms.geo.admin.ch/?SERVICE=WMS&VERSION=1.0.0&REQUEST=GetMap&FORMAT=image/png
&LAYERS=ch.swisstopo.pixelkarte-pk25.metadata-kartenblatt&SRS=EPSG:3857&STYLES=
&WIDTH=2285&HEIGHT=897&BBOX=174582,5648084,1571851,6196597&SLD=http://my.server/style.sld
```

```
<FeatureTypeStyle>
   <Rule>
      <PolygonSymbolizer>
         <Fill>
```

```
            <CssParameter name="fill">#FF0000</CssParameter>
         </Fill>
         <Stroke>
            <CssParameter name="stroke">#00FFFF</CssParameter>
            <CssParameter name="stroke-width">1</CssParameter>
         </Stroke>
      </PolygonSymbolizer>
      <TextSymbolizer>
         <Label>
            <fes:PropertyName>Blattnummer</fes:PropertyName>
         </Label>
         <Fill>
            <CssParameter name="fill">#00FFFF</CssParameter>
         </Fill>
      </TextSymbolizer>
   </Rule>
</FeatureTypeStyle>
```

Figure 4: Visualization of the grid of map sheets of Switzerland (1:25000) through another cartographic facet showing labels based on the sheet number.

Indeed, this ability to drive remotely from the user client side (or map viewer) the WMS rendering server does open interesting doors to bring to life the other use cases described in the previous section.

Going further than using a simple WMS GetMap request to get a ready-to-visualize map layer, the deprecated implementation specification (version 1.0, released in 2002) of the SLD standard (Lalonde, 2002) does offer style management requests like GetStyles. So you get also the underlying symbology instructions that has been predefined and used by the server to show a cartographic facet of the spatial data. Thus, the retrieved style is ready to be reworked by the user client within a cartographic tool.

To format the symbology, GetStyles uses a set of instructions that are precursory of the SE standard (released in 2005). To complete the SLD and SE story, in 2005, the WMS/SLD 1.1 profile has been released in particular with the aim to extract the symbology instructions into the dedicated Symbology Encoding standard. The latter gets the 1.1 version so as to be inline with the versioning of SLD. As a consequence, while the SLD profile stays strongly related to WMS service, it is no longer the case for the symbology instructions which can now be used by any styling software component, not only by WMS/SLD. Nonetheless, it is to notice that the newest SLD 1.1 release does not have anymore the style management requests which is then a step back for the appropriation use case.

But we keep in mind that the SLD 1.0 style management, which offers GetStyles and PutStyles requests, is also a good start for the do-not-reinvent-the-wheel use case. The WMS service is then the storage point to discover, import and export ready-to-use styles to share with other SDI users through a style catalog.

But most importantly, the ability to drive remotely the customization of visualisations is the fundamental base for the reuse-and-combine use case so as to fulfill the cartographic requirements of Mr Tüftel. He does not want to download the spatial data, he just wants to adjust the visualization according to his specific needs.

Finally, the SE standard is also a good base for the participatory use case that has been experimented by (Bocher et al 2012) in the frame of the SOGVILLE/SCAPC2 projects. Indeed, they use SE instructions encapsulated into map documents that different users share and work together in the frame of a collaborative cartographic authoring process. The concept of map document is strongly inspired from the ("OGC OWS Context") standard which is able to carry styles related to the layers that compose a map. For this experiment, the authoring was done with the OrbisGIS open source software as the cartographic tool compliant with an experimental revision of the SE standard (see From prototype to reference implementation section). In theory, the idea is for any SDI user to be able to rework the shared map by using her(his) prefered cartographic tool, as soon as it is compliant with a common symbology standard.

So, SLD 1.0 or SE 1.1 symbology instructions are currently the more advanced open standards for sharing cartography. These are quite largely adopted at server-side rendering systems. It can be explained because SLD is a profile of WMS which is web service oriented. It is the web interface to take control of the rendering engine behind the WMS service (Fig. 5).

At the desktop-side there are some software which implement these standardized symbology instructions. These tools can be used in a WYSIWYG mode to reuse and rework a predefined style hosted by a WMS/SLD server or to share cartography among the users working to author a common map. Nonetheless, according to (Bocher et al. 2011). many implementations have a conformance that is often not fully observed leading to interoperability defects in term of rendering quality. Apart from inherent bugs and dysfunctions of a tool, several reasons can explain this general situation: (1) due to a partial implementation - see ("MapServer") implementation, there are unimplemented symbology instructions, e.g. linejoin and linecap of LineSymbolizer; (2) due to the existence of two versions of symbology instructions between SLD1.0 and SE1.1, these tools may not check this correctly which causes parsing problems of the XML encoding; (3) due to divergent reading of what the standard tries to specify which may result in different graphical visualizations (it means there are ambiguous text in the specification) (4) related to the previous point, there is currently no substantial testsuite within the OGC Compliance and Interoperability Testing Initiative (“CITE”) to help to disambiguate and test the graphical conformance of the implementation.

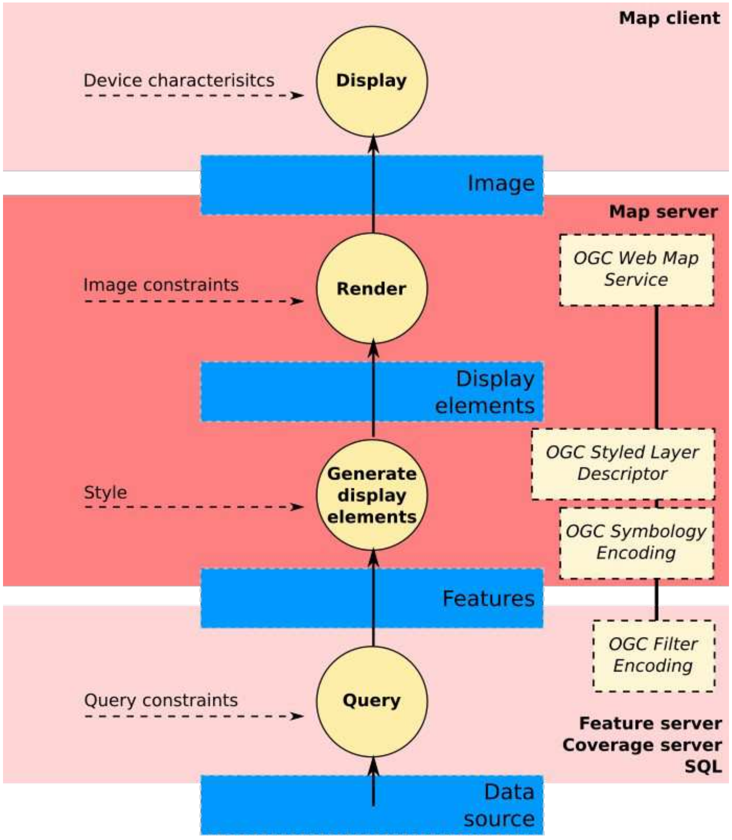

Figure 5: OGC portrayal model.

Also, both server-side and client-side, many implementations did not move to SLD/SE 1.1 although SLD1.0 is marked as deprecated by the OGC since 2006 when the revision 4 of the SE standard has been released. From that moment on, there has been many research works (Cooper, Sykora & Hurni, 2005; Teixeira, Cuba & Weiss,2005.; Sae-Tang & Ertz, 2007; Dietze & Zipf, 2007; Sykora et al, 2007; Schnabel & Hurni, 2007; Mays, 2008; Iosifescu-Enescu, Hugentobler & Hurni, 2010; Rita, Borbinha & Martins, 2010., Bocher et al, 2011; Bocher & Ertz, 2015) that were sharing a common claim about enhancing SE with new symbology capabilities. It seems the communities of users were frustrated because no substantial new symbology capabilities have been introduced with the release of SE1.1. Indeed, (Iosifescu-Enescu, Hugentobler & Hurni, 2010) underline that environmental management requires these standards to be cartographically enriched to fulfill complex visualizations with “cartographic features such as various diagram types (e.g. pie diagrams, bar diagrams), definition of complex point symbols, etc”.

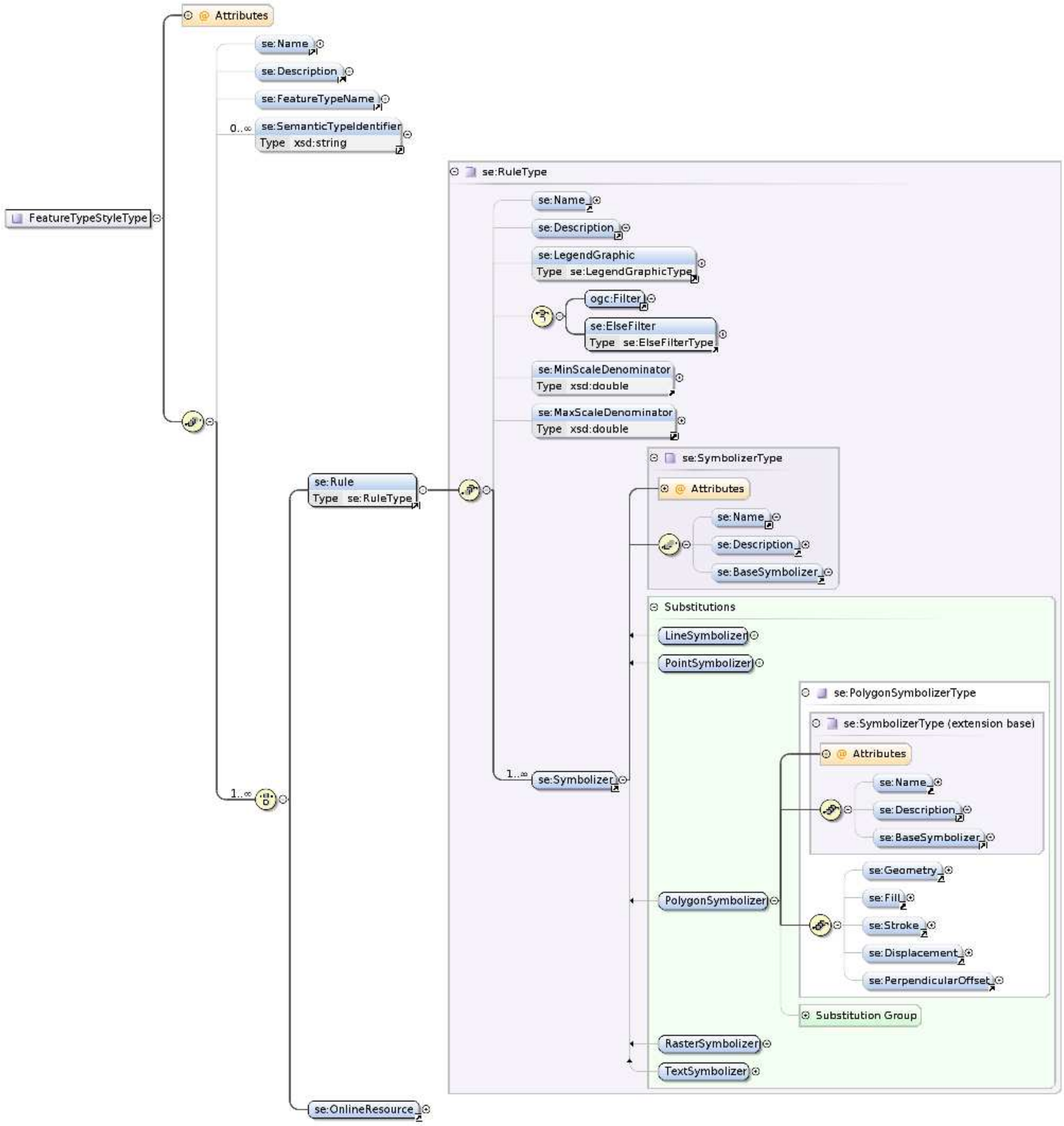

Figure 6: The physical symbology model of SE formalized with XSD, see also ("Schema documentation for FeatureStyle.xsd").

These works do also underline how it is important to have at disposal a common cartographic language that facilitates the sharing of map styles based on collaborative systems and catalogs (Chen et al., 2011; Ertz, Le Glaunec & Bocher, 2012; Rautenbach, Coetzee & Iwaniak, 2012; Wu et al., 2016). Most of these issues can also be found as pending requests at the OGC (Iosifescu-Enescu, 2007; Bruce, 2009; Holmes, 2011) which have still to be approved and integrated in a revision of standards like SLD and SE. Actually, the next section wants put forward the work with several recommendations for such a revision.

Beside these valid claims, hereinafter we want to point out some additional quite fundamental considerations so as to have a complete scene of the limits using OGC standards for sharing cartography (Bocher & Ertz, 2015).

One first aspect is about extensibility of the standard. While (Iosifescu-Enescu, Hugentobler & Hurni, 2010) are relieved by considering that fortunately these standards can be cartographically enriched to fulfill the complex visualization requirements coming from environmental management, he states this according to the extensible nature of XML being used for SE. In the same way, some application developers (e.g. "GeoServer") introduce specific VendorOption elements within the XSD schemas. Indeed, it is convenient that XML does offer extensibility to add capabilities in the cartographic language and allows at the same time to implement them in the software just like other capabilities, although it may then create some temporary non-interoperable defects.

The second aspect is that currently the SE standard does only offer one XML-based encoding. It is understandable that XML is traditionally considered as the default encoding for many OGC standards, because XML can be combined with DTD, XML Schema Definition, Relax NG or Schematron so as to strictly formalize the structure of a physical model. As a consequence, it may be difficult for cartographic communities and developers having different encoding preferences (e.g. CSS-like or JSON-based) to get a chance to observe conformance with a standard like SE which is strongly linked to XML modeling principles (Fig. 6).

5. Proposals

Given the substantial amount of deficiencies while using OGC SLD and SE standards, this section aims at describing some recommandations able to make the use cases described in section 3 a reality by solving the limits described in section 4. The overall purpose is to make these standards more attractive by turning them into "a really useful (cartographic) engine", quoting the nod to *Thomas the Tank Engine* alluded by the OGC "Specification Model — A Standard for Modular specifications" document (OGC Policy SWG, 2009), called the modular spec in below.

Before compiling all the CR related to SE, one question does arise: how to plug a new requested ability in the standard? One first and fundamental recommendation is then to consider the modular spec whose release 1.0 has been edited in 2009, at the time the SE standard was already released and thus not in compliance with. Indeed, the modular spec specifies generic rules to organize the internal logical structure of the standard in a modular way so as to strengthen the guarantee of useful and worth standards which are easy to implement but also to extend.

A) Modular structure: one symbology core, many symbology extensions

Considering the latter point, the modular spec fittingly suggests requirements for modularity with the idea of a standard built of one simple core and many extensions which expand the functionality of a specification. Applied to a new revision of the SE standard, the definition of a symbology core requires first to "reverse design" the underlying symbology model of SE1.1. After which, the concrete symbology capabilities have to be extracted and split into many relevant extensions while taking care of the dependencies. A minimal symbology core (Fig. 7) that is partially abstract and defined could be proposed by the following concepts:

- the Style, in charge of the cartographic portrayal of a collection of features[1] stored within a Layer by applying at least one symbology Rule;
- the rendering does run feature per feature using a "one drawing pass" engine;
- each Rule may be scale filtered and does hold at least one Symbolizer;
- each Symbolizer does describe the graphical parameters for drawing the features (visual variables);
- the Style, Rule and Symbolizer hold parameters which are literal values.

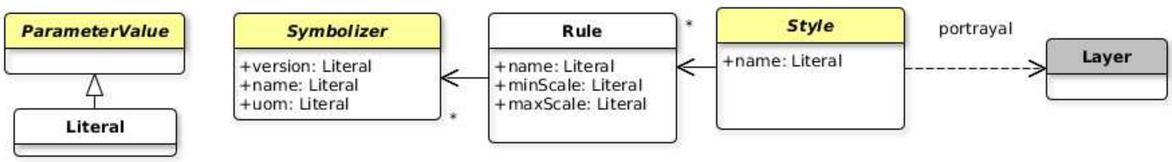

Figure 7: Recommendation for a minimal symbology core.

Some of the concepts are defined as abstract (in yellow and italic names) so as to be considered as extension points. Indeed, SE1.1 has never been designed with modularization and extensibility in mind and there is no explicit extension points defined in the underlying symbology model (except by considering XML as natively extensible). Actually, regarding this, we may notice that (Bruce, 2009) does request a better use of XSD concepts, especially by the use of XML abstract elements which may than be considered as extension points.

Now that the core is ready, some surrounding extensions have to be defined so that the engine is really able to perform a rendering. Indeed, alone, the core doesn't concretely "do" anything. Let's introduce the simple and classical symbolizer which describes the graphical parameters for drawing polygonal features with outlined and filled surface areas (called the AreaSymbolizer concept below). The aim of the below explanations is to illustrate with a simple example such a core and extensions mechanism.

So the renderer has really something to do concretely, extension points have to be expanded. At first, the FeatureTypeStyle extension does introduce:

- the portrayal of a Layer built of N instances of GML **AbstractFeatureType** (Portele, 2007);
- the ability to access features according to **Simple Feature SF-2** (Brink, Portele & Vretanos, 2012);
- the **FeatureTypeStyle** specialization of the Style core concept;
- the geometry parameter to each Symbolizer extension that depends on this extension (e.g. the below AreaSymbolizer).

Notice that the geometry parameter has a dependency on the ValueReference extension which hold the **ValueReference** specialization of the ParameterValue core concept. In a general way, when a parameter has to be assigned with a value, ValueReference does introduce the ability to reference the value extracted from a data attribute of a feature. This is useful when a FeatureType does hold many geometry properties and allows to reference the one to be used by the renderer.

[1] As defined by GML standard, a feature is an abstraction of real world phenomena

PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.2415v1 | CC BY 4.0 Open Access | rec: 4 Sep 2016, publ: 4 Sep 2016

Then, the AreaSymbolizer extension is required, holding the **AreaSymbolizer** specialization of the Symbolizer core concept. Called PolygonSymbolizer in SE 1.1 and correctly renamed AreaSymbolizer by (Bruce, 2009), it does introduce:

- the symbology ability to draw a surface area according to a filling and an outline;
- the dependency on the FeatureTypeStyle, **Fill**, **Stroke** and the **Translate** extensions;
- the ability to reference the geometry data attribute to be drawn (by means of its dependency on the FeatureTypeStyle extension).

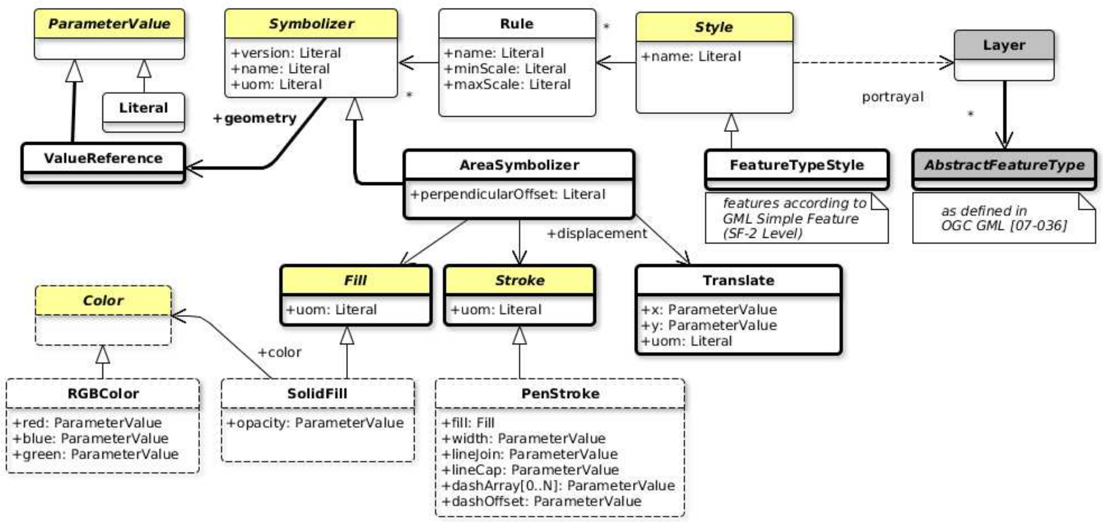

Figure 8: Concepts to implement to observe conformance with the AreaSymbolizer extension and to have the rendering engine drawing something concrete.

In summary, an implementation that wants to observe conformance with the AreaSymbolizer extension requires to implement and drive its rendering engine according to (─ outline in Fig. 8) all the concepts of the core and (━ outline in Fig. 8) the AreaSymbolizer concept with all the other concepts required by dependencies.

Nonetheless, even at this point, a rendering engine would neither concretely “do” anything. Indeed, the implementation has then to offer choices related to the filling and the outline. Concrete capabilities have to be implemented, for instance with (--- outline in Fig. 8):

- the SolidFill concept, a Fill specialization which introduces the graphical ability to define a solid color value combined with an opacity;
- the PenStroke concept, a Stroke specialization which introduces the graphical ability to draw a continuous or dashed line with or without join and cap;
- the dependent abstract Color concept (and again a concrete choice of color definition has to be done, like with the RGBColor concept which defines a color in the sRGB color space with three integer values).

Having this modularity approach for long term extensibility applied to all the concepts, past, present and future, an implementation can with ease manage step by step the evolution of the conformance level of its technical implementation of the standard.

B) One encoding-neutral conceptual model, many encodings

Currently, SE 1.1 does offer a physical model using XML Schema Definition and, at the same time, a natural encoding based on XML. The initial motivation explaining the below recommendation is related to the fact that there is not only XML, but also many other flavors of encoding, JSON-like, CSS-like, YAML-like among many others it is possible to imagine. The important for portrayal interoperability is not the encoding, it is rather the symbology model. That's why the "one encoding-neutral model / many encodings" approach is promising to favor a large adoption of the standard.

This approach has on one side the encoding-neutral model formalized using UML notations, it can be considered as conceptual. With a class diagram, it does describe the portrayal concepts, their relationships, the modular organization and the dependencies. We may notice that UML is often prefered when some work is about the design of portrayal concepts. In (Zipf, 2005), Fig. 9 does depict a simplified version of the underlying symbology model as an UML class diagram. Moreover, the author of (Bruce, 2009) does suggest to avoid the XSD attribute concept in the XML encoding so as to be more portable to other structuring languages which do not have the unusual attribute concept of XML Schema, UML in particular. These are more arguments that are in favor of defining at first a conceptual and encoding-neutral model (Fig. 9).

On the other side of this recommended approach, doors are open to offer a variety of encodings. Each encoding does format the UML notations according to mapping rules. The new symbology standard would then be released with at least one default encoding. Following the OGC tradition, XML may be this default encoding. It is up to the standard working group to define the mapping rules to translate the semantic of the conceptual model into XML Schema definitions. Indeed, as noticed by (Lonjon, Thomasson & Maesano, 2006), the translation from UML to XML requires a thoughtful analysis of the conceptual model so as to define the global mapping rules (e.g. translate a specialization relationship using static or dynamic typing? how to translate a concrete class, an abstract class, the various types of associations? when using attributes or elements? etc). Thus, UML and XML are together a winning combination two times inline with the modular specification which recommend UML "If the organizing mechanism for the data model used in the specification is an object model" and XML "for any specification which has as one of its purposes the introduction of a new XML schema".

Of course, all these questions related to the mapping rules have to be considered for each encoding offered with the standard. We may notice that the OWS Context Standard Working Group adopted a similar approach, offering the default encoding based on XML Atom and planning to provide an OWS Context JSON Encoding soon, according to ("OGC OWS Context").

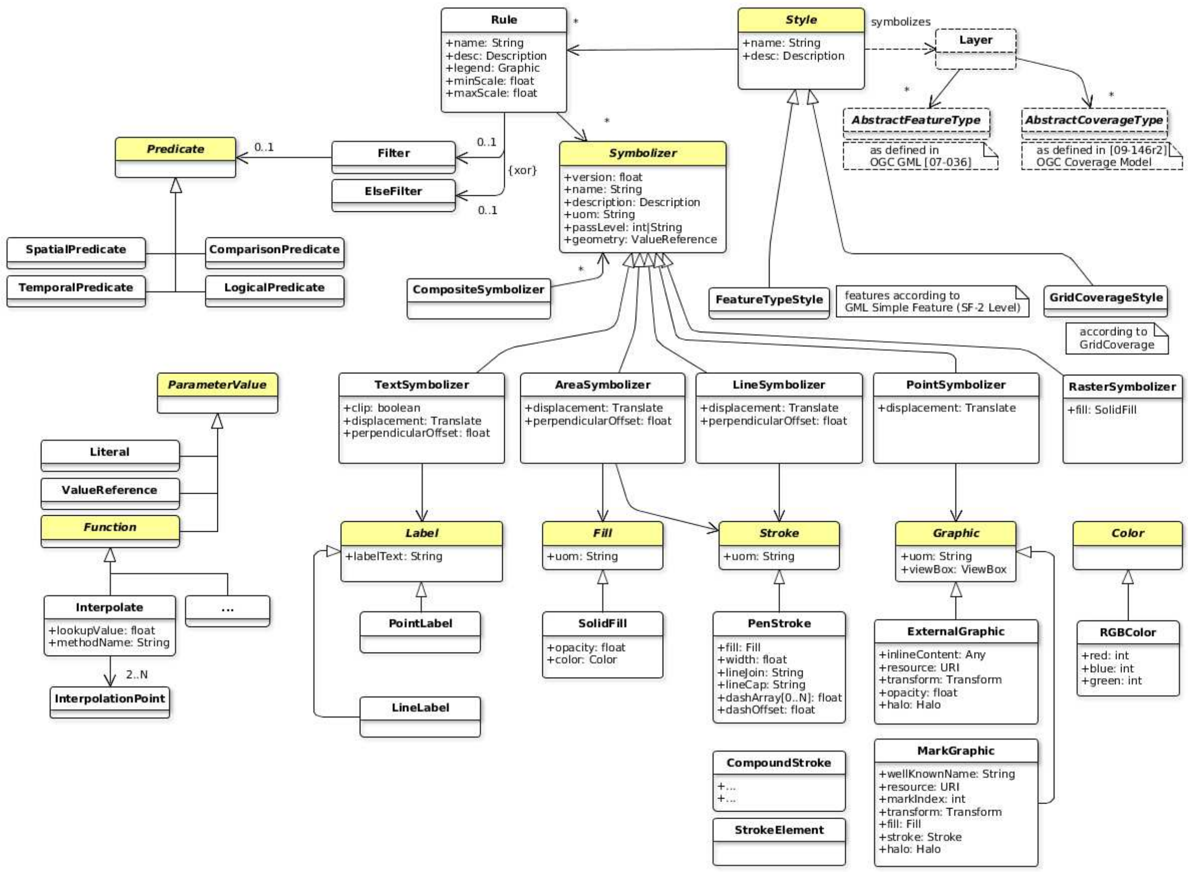

Figure 9: An extract of the recommended symbology model.

C) Style management and parametrized symbolizer

Beyond the tempting recommendation to reintroduce the WMS/SLD GetStyles and PutStyles methods, the management of a catalog of styles has to be expanded. Thus, (Bruce, 2009) does suggest the introduction of a mechanism to reference the definition of a Symbolizer hosted within a catalog. Moreover, the report does enrich the referencing with a symbolizer-parameterization mechanism so as to offer complete symbolizer reusability between different, incompatible feature types. It consists of a list of formal-parameter names and an argument list.

It is to notice that such a mechanism does fit the one specified by ("ISO 19117") in term of parameterized symbol built of dynamic parameters. Thus, in a general way, it is recommended to consider what ISO has already specified concerning the concepts of "collection of symbols and portrayal functions into portrayal catalogue".

Concerning this aspect of style management, the proposal suggests to continue the conceptual work by blending together all these recommendations: reintroduce GetStyles/PutStyles and introduce the mechanism of symbolizer-parameterization inline with ("ISO 19117").

D) New symbolizations capabilities

Among the many symbology capabilities that can be extracted from the pending requests at OGC and the research works, we can list below (non exhaustively) some relevant ones.

Considering the modular structure (see A), each capability have been designed so as to be integrated within the new symbology model as an extension and its related dependencies (e.g. HatchFill is an extension of the Fill abstract concept, just as SolidFill):

- UnitOfMeasure: adds absolute portrayal units of measure (e.g. mm). At least three additional units are added to make measurements more portable between styling representations and rendering environments: portrayal millimeters and portrayal inches as printing measurements, and portrayal (printer's) points commonly used for font sizes.";
- Affine transformations: allows to perform general affine transformations like Translate, Rotate, Scale, Matrix using homogeneous coordinates on geometries and graphics.
- Functions: updates the dependency on Filter Encoding (FE 2.0) so as to define symbology-related functions using the mechanism of FE but also to inherit geometry accessors and operators functions.
- CompoundStroke: allows multiple graphic and/or simpler strokes to be combined together along the linear path. It is interesting to produce complex stroke styles such as rendering a sequence of graphic icons along a line or drawing simple dashed lines between boat-anchor icons;
- CompositeSymbolizer: allows to manage groups of descendant symbolizers as a single unit. It does make the logical grouping more explicit, but more importantly, it allows a group of symbolizers to be remotely referenced (see C)
- HatchFill: adds cross hatching, a method of area filling which is often used and has so simple parameters that it should be established as another filling variety. It is required to allow such a filling in a way conventional in cartography, otherwise the user would be forced trying to fiddle with GraphicFill;
- DiagramSymbolizer: adds diagram symbolization of geographic features, an effective way of visualizing statistical data in a spatial context. It offers support of "Pie", "Bar", "Line", "Area", "Ring", and "Polar" charts in order to allow visualization of multiple data values using diagrams;
- ParametrizedSymbolizer: adds a symbolizer-parameterization mechanism to offer complete symbolizer reusability between different, incompatible feature types. This extension offers a way to expose a symbolizer in a remote library.
- Multiple drawing pass: adds capabilities to order the level of symbol rendering (e.g. draw nicely connected highway symbols)

6. From prototype to reference implementation

The OrbisGIS platform has been used to prototype an implementation of the symbology model all along the standardization process by iterations with tests and validations. At long term, this platform may be adopted as a reference implementation from the OGC ("Compliance and Interoperability Testing Initiative").

OrbisGIS is a Geographical Information System designed by and for research (Bocher & Petit, 2013) which is the main advantage for research communities comparing to other GIS. Indeed, OrbisGIS doesn't intend to reproduce classical GIS functionalities. It is designed to explore new issues or questions in the field of geospatial techniques and methods (such as language issues to query spatial informations and issues on cartography about standardization, semantics and user interfaces usability. To address these challenges, the OrbisGIS architecture (object and data model) and its user interface are frequently redesigned. This approach is fundamental to test the concepts and the ideas related to the ongoing standardization process of symbology standards at OGC. Furthermore, the fact that we have a common set of GIS features organised with the dynamic module system OSGi to access to the geodata, library to use simple features functions, layer model, rendering transform, etc ("Specifications – OSGi™ Alliance"). For example, it gives flexibility to plug some experimental code without breaking the platform and the user can easily switch from one to another plugin (Fig. 10). More importantly, the usage of OSGi technology does offer a way to implement the modularization principles depicted in the above (i.e. one OSGi bundle per symbology extension).

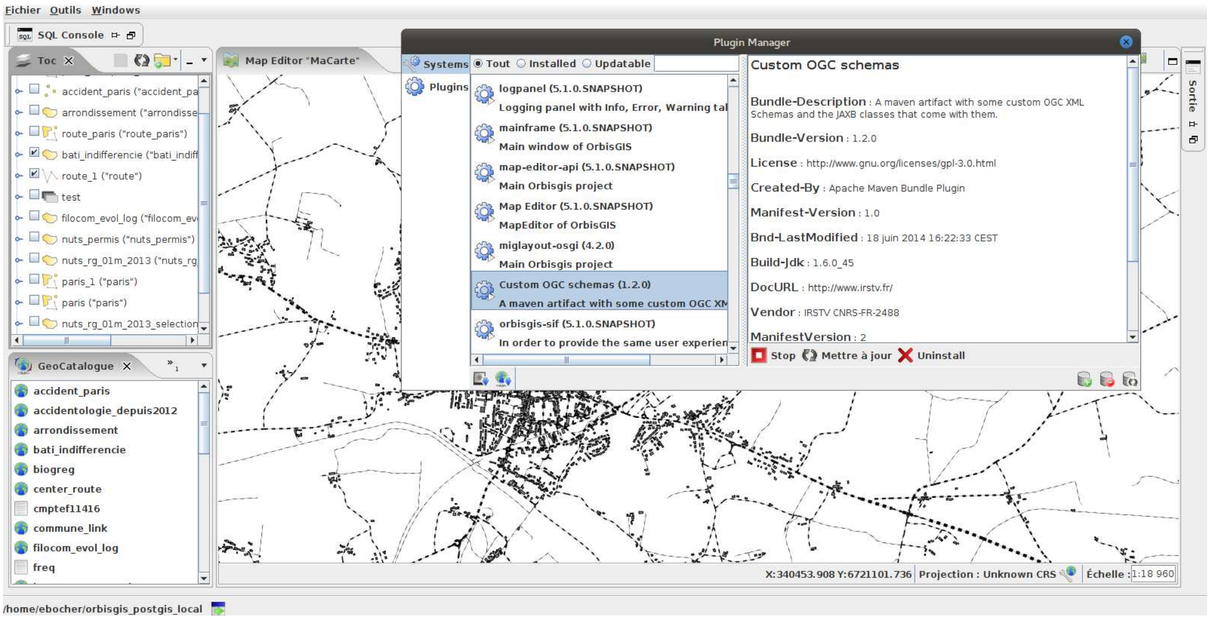

Figure 10: OrbisGIS dynamic module system with OSGi.

Another important aspect of the motivation is related to the license. OrbisGIS is an open source software, distributed under the GPL3 license and therefore grants four freedoms (1) to run the program for any purpose, (2) to study how the program works and adapt it to your needs, (3) to redistribute copies so you can help your neighbour, (4) to improve the program, and to release your improvements to the public, so that the whole community benefits (Steiniger & Bocher, 2009).

This aspect is essential in order to have a reference implementation available for the community of implementers of a standard, guiding them even more in its understanding. Given that for research, having open code source available does enable reproducibility, a core principle of science (Ertz, Rey & Joost, 2014), we argue that this is also valid for open standards. At one side, it is easy for other researchers and businesses to verify and re-use new developments and adapt them to their needs (Stefan & Bocher, 2009). Furthermore, having the code of the rendering engine, the user interfaces and all the tests fully accessible should facilitate the understanding and the dissemination of standards for portrayal interoperability while minimizing interoperability defects. In the following we describe the steps of the implementation.

Step 1 : XML encoding/decoding

In the context of a prototyping iteration, the symbology model presented in the section 5 has been transposed a first time in a XSD schema (“orbisgis/ogc-custom-jaxb”). The Java Architecture for XML Binding (JAXB) library is used to generate the XSD schema-derived Java binding classes. Finally, a Java Style Object Model is built.

Step 2 : Rendering engine

The rendering engine is a OSGi bundle. Its mechanism is divided into 12 sequences (Fig. 11):

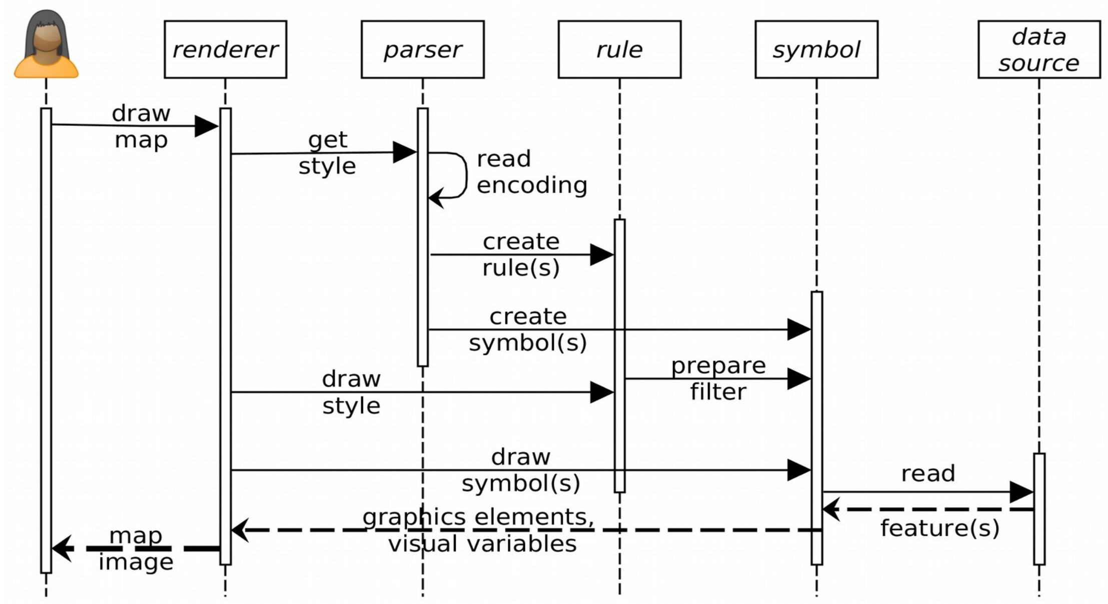

Figure 11: Main sequences of the rendering engine.

(1) A user call to draw a map in OrbisGIS;

(2) The renderer engine get the xml file that contains the style;

(3, 4 and 5) The XML file is read by the style parser to create the Java Style Object Model composed of rules and symbols

(6) The renderer engine starts to draw the style object looping over each rules.

(7) Each rule is scanned to check if a filter must be applied. The filter condition (e.g. select all values greater than…) is prepared for the symbolizer of the rule.

(8) The renderer engine starts to draw all symbols available in the Java Style Object Model.

(9) Each symbol reads the data source on which the style must be applied.

(10) A set of features according to the potential filter constraint of the symbolizer is returned (including the geometries and the data attributes)

(11) The symbols are filled with the features properties to create the graphic elements and visual variables.

(12) Finally, the renderer engine displays the style as a map image.

PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.2415v1 | CC BY 4.0 Open Access | rec: 4 Sep 2016, publ: 4 Sep 2016

Step 3 : User interfaces

OrbisGIS offers two user interfaces for configuring the map styles using the capabilities of the underlying symbology model:

- The first one is a **productivity** tool organized around a set of widgets each dedicated to a common thematic map (Fig. 12A, Fig. 12B). The possibilities are limited to what these widgets are able to configure related to what they have been built for. Nonetheless, the second tool can then be used in an expert mode to go further.
- The second one is rather intended for an expert who want to tinker and tweak (Fig. 12C). With all the possible **flexibility**, it allows to configure all elements of the style (Rule, Symbols, visual variables). This advanced style editor needs a good knowledge of the symbology model because each elements of the style must be set individually. Consequently, the user can express without any limitation (except the limits of the symbology model itself) all her(his) creativity to build complex and outstanding cartographic visualizations.

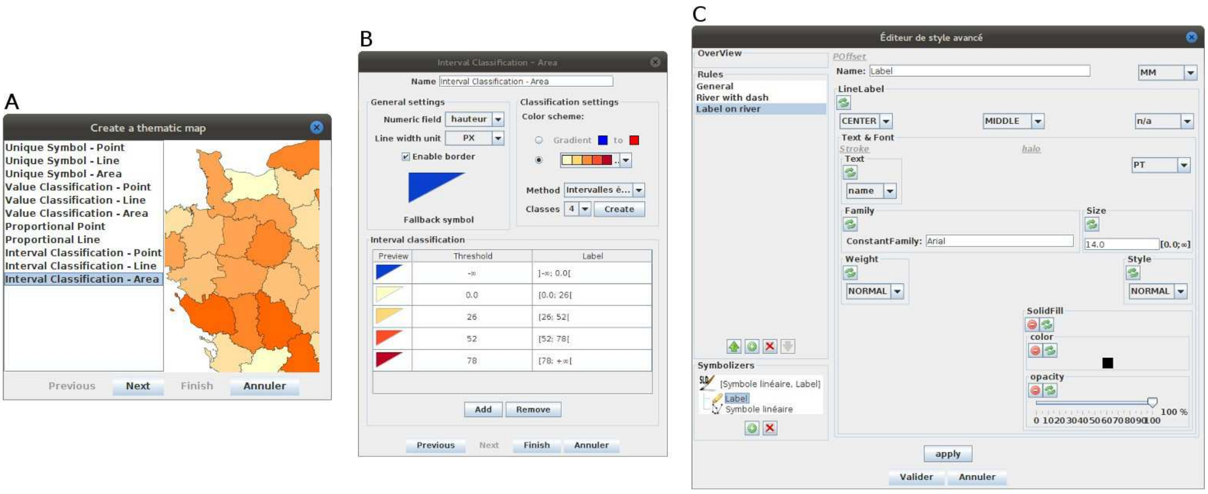

Figure 12: Style editors (A, B) productivity tools (C) advanced configurator

To illustrate some results rendered with OrbisGIS we present two maps extracted from the ("OrbisGIS Wall of Maps").

The first one shows a bivariate map to display at the same time the number of building permits in Europe in 2005 compared to 2014 (Fig. 13).

Bivariate map is a common technique to combine visual variables. The map uses the same type of visual variable to represent two values (as half circles). With an YAML-like encoding (on the right side of the Fig. 14), we list below the main symbology elements used to create this bivariate map:

- The style element contains 2 rules named A and B;
- Rule A contains one symbolizer element (AreaSymbolizer) to display the stroke of the european countries;
- Rule B defines the bivariate proportional symbol with two elements of PointSymbolizer (for readability, we present only the left half-circle variable);
- The PointSymbolizer contains several sub-elements :

  - the geometry element allows specifying which geometry attribute is to be rendered;
  - the ST_PointOnSurface is an OGC filter function (Vretanos, 2014) used to have a point geometry guaranteed to lie on the surface. This new point derived from the input geometry is the location where to anchor a MarkGraphic, otherwise the symbol might be applied on all the vertices of a geometry;
- The MarkGraphic is defined by :
  - the symbol shape identified by a well-known name, HALFCIRCLE (right side);
  - the size of the shape varies according the height of its view box;
  - to have the shape size proportional with the number of building permits in 2015:
    - an interpolate function is applied on;
    - it uses a ValueReference that points to the attribute named *permits2005;*
    - the interpolation is defined by two interpolation points chosen along a desired mapping curve (here the minimum and maximum values);
    - for each interpolation point the height of the view box is specified with a specific unit of measure;
  - because the half-circle shape is drawn to the right side, a 180-degree rotation is operated;
  - to finish, the MarkGraphic is filled with a RGB color.

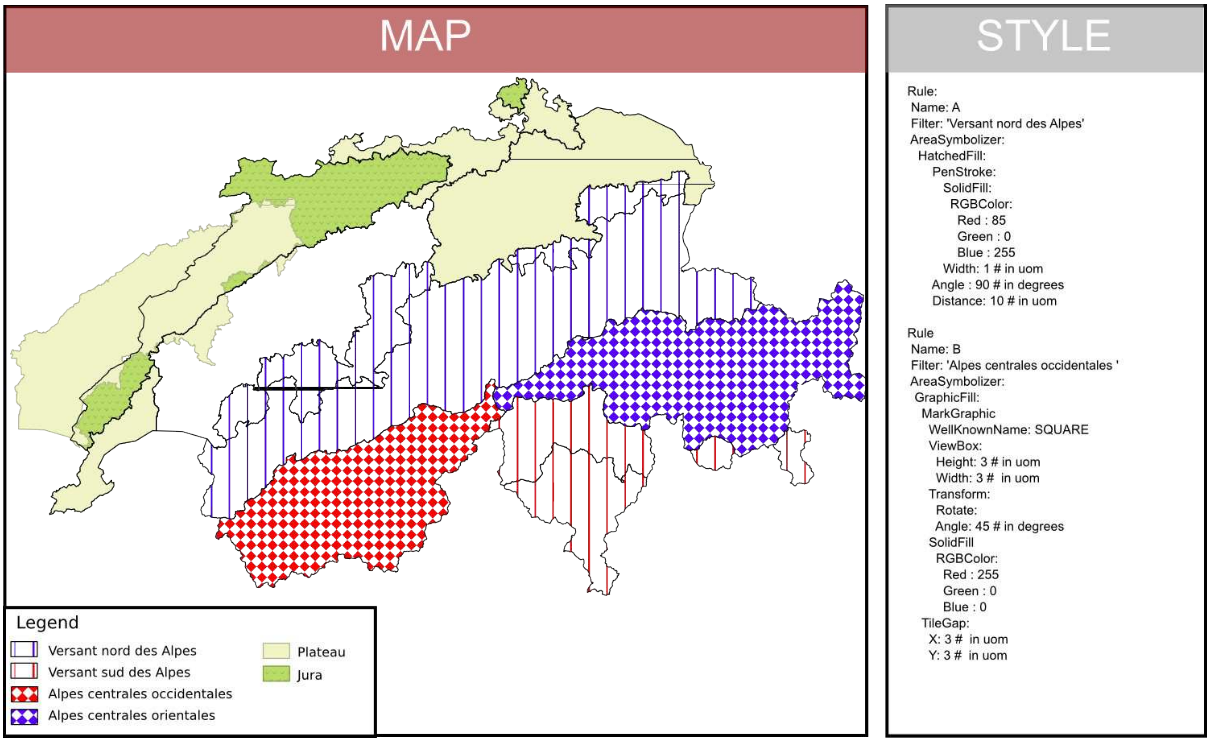

Figure 13: Combined visual variables to cartography the biogeographic regions in Switzerland.

The second map shows a combination of several visual variables: shape, size, color, patterns and orientation (Fig. 14). The style is organized around 6 filtered rules that correspond to the biogeographic regions in Switzerland. We present two Rules (A and B) that use the HatchFill and GraphicFill concepts which are extensions of the Fill abstract concept of the symbolizer model (see the proposal section for few details on HatchFill).

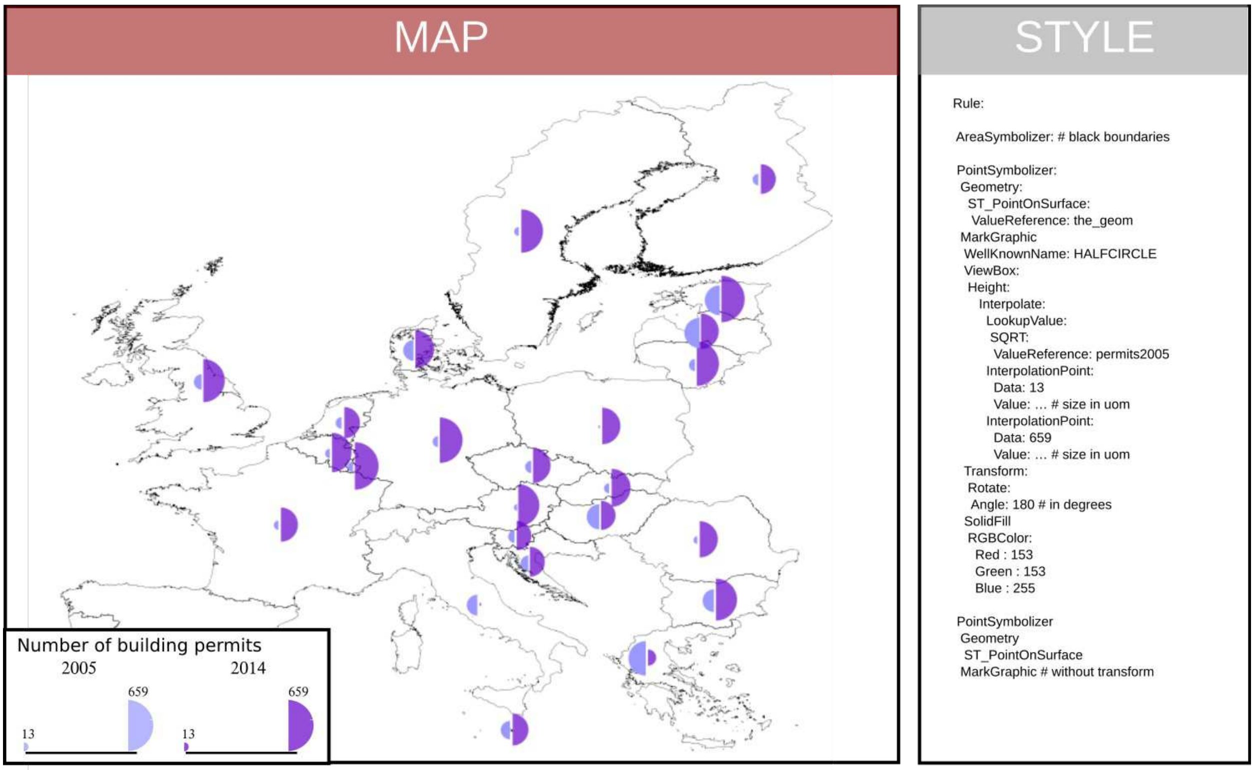

Figure 14: A bivariate proportional symbol map expressed with the Symbology Encoding model.

7. Conclusion

Considering the fundamental works of (Bertin & Berg, 2010) and their heirs, the community of map makers have constantly investigated questions about cartographic visualisations in term of design using the appropriate visual variables and in term of relevancy when these are combined together. Despite an important body of principles and practices, the community did not grasp the questions about standardization. However, given the multiplicity of software used to flood the world with maps, these questions are nowadays a strategic challenge to be considered in relation with the operational requirements.

Even if the definition of a cartographic-oriented standard is not able to act as a complete cartographic design framework by itself, we argue that pushing forward the work aiming at the creation of dedicated standards for cartography is a way to share and disseminate good practices. Indeed, too much spatial data infrastructures do merely accept the limits of the current standards and consequently poor map design. While they have to apply them, it is essential to be design them so as to be able to enrich their cartographic capabilities, to make grow up the practices and finally to improve the quality of the visualizations. In this sense, we have identified some use cases showing how it is important to make portrayal interoperability operational for sharing cartography: the appropriation use case, the reuse-and-combine use case, the do-not-reinvent-the-wheel use case, the participatory use case.

From research results in link with the dedicated SLD/SE OGC Standard Working Group ("Styled Layer Descriptor and Symbology Encoding 1.2 SWG Charter document"), this paper does extract some recommandations to form a foundation able to fulfill these goals. They invite to improve the OGC Symbology Encoding standard based on this important body of principles and practices in cartography. We start from a functional definition of a map translated into a set of visual variables which are combined to create symbols and finally a map style. The latter is then applied on a dataset itself standardized so as to render a map. This functional definition is at the heart of how SE standard has been specified by OGC and it does not change to incorporate the recommendations. These give some answers on how to organize into a symbology model the existing capabilities of SE and the requirements coming from the community of map makers (from compound stroke to charts, etc).

The design approach is driven by a conceptual definition of the model and unconstrained by specific encoding aspects. As soon as the model is ready, then a default encoding is offered (e.g. XSD/XML). Follow on from this approach of dissociation, it does allow the definition of other encodings according to the various flavors within the communities.

Given that the cartographic requirements will progress during time due to practices growing up and according to domain specific features, the offered foundation is empowered so as to be an extensible symbology model ready to host new cartographic techniques. Moreover, such a modular approach allows implementations to be compliant step-by-step. As a consequence the adoption of the standard should be favored.

Also, we claim to a testsuite within the OGC Compliance and Interoperability Testing Initiative so as to help to disambiguate and test the visual conformance of the implementations. While it shall be associated to reference implementations, having at least one opensource is also essential for the community of implementers, guiding them even more in the understanding of the standard. In this sense, OrbisGIS is a platform that has been used to prototype an implementation of the symbology model all along the standardization process by iterations with tests and validations. It may become an opensource reference implementation.

8. References

Andrae C., Graul C., Over M., Zipf A. 2011. Web Portrayal Services: OpenGIS Web Map Service, Styled Layer Descriptor, Symbology Encoding und ISO 19117 Portrayal vorgestellt und erläutert. Wichmann.

Beaujardiere J de la. 2006. OpenGIS® Web Map Server Implementation Specification (OGC 06-042).

Bertin J., Berg WJ. 2010. Semiology of graphics: diagrams, networks, maps. Redlands, Calif: ESRI Press : Distributed by Ingram Publisher Services.

Bocher E., Ertz O., Laurent M., Petit G., Rappo D. 2011. Cartographie et standard : du modèle à l'utilisateur. In : The 25th International Cartographic Conference. Paris. France.

Bocher E., Petit G., Gueganno A., Fortin N., Gourlay A., Ertz O. 2012. SOGVILLE - Système d'Observation Géographique de la Ville : Du SIG au partage des connaissances. Available at https://hal.archives-ouvertes.fr/halshs-01141548/file/Presentation_SOGVILLE_finale.pdf (accessed 1 July 2016)

Bocher E., Petit G. 2013. ORBISGIS: Geographical Information System Designed by and for Research. In: Bucher B, Ber FL eds. Innovative Software Development in GIS. John Wiley & Sons, Inc., 23–66.

Bocher E., Ertz O. 2015. Towards cartographic portrayal interoperability - The revision of OGC Symbology Encoding standard. In : 1st ICA European Symposium on Cartography. Vienna. Available at http://eurocarto.org/wp-content/uploads/2015/10/3-6.pdf (accessed 1 July 2016)

Brink L van den., Portele C., Vretanos P. 2012. Geography Markup Language (GML) simple features profile - with Corrigendum (OGC 10-100r3).

Bruce C. 2009. OGC® OWS-6 Symbology Encoding (SE) Changes (OGC 09-016).

CartoCSS. Available at https://cartodb.com/docs/cartodb-platform/cartocss/properties (accessed July 6, 2016).

CartoDB Editor. Available at https://cartodb.com/platform (accessed July 6, 2016).

Carpendale MST. 2003. Considering visual variables as a basis for information visualisation.

Chen T., Chen M., Tan S., Luo Q., Wu M. 2011. Study on Analysis and Sharing of ArcGIS Symbol and SLD Symbol. In: Zhang J ed. Applied Informatics and Communication. Communications in Computer and Information Science. Springer Berlin Heidelberg, 54–61.

CITE. Available at http://cite.opengeospatial.org/teamengine (accessed July 6, 2016).

Cooper M., Sykora P., Hurni L. 2005. The Role of Cartography within Distributed Software Systems. What can we Contribute? How can we Prosper? . In : 22nd International Cartographic Conference and 13th General assembly of the ICA. International Cartographic Society. A Coruña, Spain.

Compliance and Interoperability Testing Initiative. Available at http://cite.opengeospatial.org/reference (accessed July 7, 2016).

Craglia M. 2010. Building INSPIRE: The Spatial Data Infrastructure for Europe. ArcNews 32:1–6.

Dietze L., Zipf A. 2007. Extending OGC Styled Layer Descriptor for thematic cartography, In: Proceedings of the 4th Int. Symp. on LBS and Telecartography. Hong Kong.

EPA 2011. Guidance Note for Strategic Noise Mapping.

Ertz O., Le Glaunec J., Bocher E. 2012. Collaborative authoring and polypublication of cartographic content. In: OGRS2012 Proceedings. Yverdon-les-Bains. Switzerland.

Ertz O., Rey SJ., Joost S. 2014. The open source dynamics in geospatial research and education. Journal of Spatial Information Science. DOI: 10.5311/JOSIS.2014.8.182.

Eurostat / Regional Statistics Illustrated. Available at http://ec.europa.eu/eurostat/cache/RSI/#?vis=nuts2.population&lang=en (accessed July 6, 2016).

Field K. 2014. A Cacophony of Cartography. The Cartographic Journal 51:1–10. DOI: 10.1179/0008704114Z.000000000120.

Garlandini S., Fabrikant SI. 2009. Evaluating the Effectiveness and Efficiency of Visual Variables for Geographic Information Visualization. In: Hornsby KS, Claramunt C, Denis M, Ligozat G eds. Spatial Information Theory. Berlin, Heidelberg: Springer Berlin Heidelberg, 195–211.

GeoServer. Available at http://geoserver.org/ (accessed July 6, 2016).

Holmes C. 2011. Symbology Encoding Improvements (OGC CR11-023).

Hopfstock A., Grünreich D. 2009. A User-Oriented Map Design in the SDI Environment Using the Example of a European Reference Map at Medium Scale. In: Proceedings of the 24th International Cartographic Conference. Santiago de Chile, Chile.

Iosifescu-Enescu I. 2007. SE Implementation Specification Change Request – extensions for thematic mapping (OGC CR07-105).

Iosifescu Enescu I., Hurni L. 2007. Towards Cartographic Ontologies Or How Computers Learn Cartography. In: Proceedings of the 23th International Cartographic Conference. Moscow, Russia.

Iosifescu-Enescu I., Hugentobler M., Hurni L. 2010. Web cartography with open standards – A solution to cartographic challenges of environmental management. Environmental Modelling & Software 25:988–999. DOI: 10.1016/j.envsoft.2009.10.017.

INSPIRE Drafting Team 2008. D2.6: Methodology for the development of data specifications, Version 3.0.

INSPIRE Drafting Team 2014. D2.5: Generic Conceptual Model, Version 3.4.

ISO 19117:2012 - Geographic information -- Portrayal. Available at http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=46226 (accessed July 6, 2016).

Kraak M-J. 2006. Why Maps Matter in GIScience. The Cartographic Journal 43:82–89. DOI: 10.1179/000870406X93526.

Lalonde W. 2002. Styled Layer Descriptor profile of the Web Map Service Implementation Specification (OGC 02-070).

Lambert N., Zanin C. 2013. ESPON Cartographic Language - Mapping guide. Available at http://f.hypotheses.org/wp-content/blogs.dir/1381/files/2014/02/Task4_final_ECL_MappingGuide_dec13.pdf (accessed July 6, 2016).

Lonjon A., Thomasson J-J., Maesano L. 2006. Modélisation XML. Paris: Eyrolles.

Lupp M. 2007. Styled Layer Descriptor profile of the Web Map Service Implementation Specification (OGC 05-078r4).

MacEachren AM. 2004. How maps work: representation, visualization, and design. New York: Guilford Press.

MapServer. Avalaible at http://mapserver.org/ogc/sld.html (accessed July 6, 2016).

McMaster R., McMaster S. 2002. A History of Twentieth-Century American Academic Cartography. Cartography and Geographic Information Science 29:305–321. DOI: 10.1559/152304002782008486.

Mays J. 2008. Using SLD definitions to display charts in a deegree WMS. In : FOSS4G conference. Cape Town. Available at http://2008.foss4g.org/paper/view/332.1.html (accessed 1 July 2016)

McConchie A. 2015. Hacker Cartography: Crowdsourced Geography, OpenStreetMap, and the Hacker Political Imaginary. ACME: An International Journal for Critical Geographies 14:874–898.

Montello DR., Robinson A. 2002. Cognitive map-design research in the twentieth century: Theoretical and empirical approaches. Cartography and Geographic. In: Information Science. University Press, 283–304.

Müller M. 2006. Symbology Encoding Implementation Specification (OGC 05-077r4).

OGC OWS Context. Available at http://www.owscontext.org (accessed July 6, 2016).

OGC Policy SWG 2009. The Specification Model — A Standard for Modular specifications (OGC 08-131r3).

OrbisGIS Wall of Maps. Available at http://se.orbisgis.org/ (accessed July 12, 2016).

Peterson GN. 2009. GIS cartography: a guide to effective map design. Boca Raton: CRC Press.

Portele C. 2007. OpenGIS® Geography Markup Language (GML) Encoding Standard, Version 3.2.1 (OGC 07-036).

Sae-Tan A., Ertz O. 2007. Towards Web Services Dedicated to Thematic Mapping. OSGeo Journal. Available at http://www.osgeo.org/files/journal/v3/en-us/final_pdfs/ertz.pdf (accessed 1 July 2016)

Schnabel O., Hurni L. 2007. Diagram markup language - a new model for symbolization in internet maps. In : 23th International Cartographic Conference. Moscow. Russia.

Rita E, Borbinha J, Martins B. 2010. Extending SLD and SE for cartograms. In : In GSDI 12 World Conference. Singapore.

Rautenbach V, Coetzee S, Iwaniak, A. 2012. Orchestrating OGC web services to produce thematic maps in a spatial information infrastructure. Computers, Environment and Urban Systems. 37:107–120. DOI:10.1016/j.compenvurbsys.2012.08.001

Schema documentation for FeatureStyle.xsd. Available at http://www.mediamaps.ch/ogc/schemas-xsdoc/sld/1.1.0/FeatureStyle_xsd.html (accessed July 12, 2016).

Slocum TA. (ed.) 2009. Thematic cartography and geovisualization. Upper Saddle River, NJ: Pearson Prentice Hall.

Specifications – OSGi™ Alliance. Available at https://www.osgi.org/developer/specifications (accessed July 7, 2016).

Steiniger S., Bocher E. 2009. An overview on current free and open source desktop GIS developments. International Journal of Geographical Information Science 23:1345–1370. DOI: 10.1080/13658810802634956.

Styled Layer Descriptor and Symbology Encoding 1.2 SWG Charter document. Available at http://www.opengeospatial.org/projects/groups/sldse1.2swg (accessed July 7, 2016).

Sykora P., Schnabel O., Enescu II., Hurni L. 2007. Extended Cartographic Interfaces for Open Distributed Processing. Cartographica: The International Journal for Geographic Information and Geovisualization 42:209–218. DOI: 10.3138/carto.42.3.209.

Teixeira M. D., Cuba R. M., Weiss M. G. 2005. Creating Thematic Maps with OGC Standards Through the Web. GML And Geo-Spatial Web Services. In : GML and Geo-Spatial Web Services Conference. Vancouver, British Columbia.

Tóth K., Portele C., Illert A., Lutz M., Nunes de Lima M., European Commission, Joint Research Centre, Institute for Environment and Sustainability 2012. A conceptual model for developing interoperability specifications in spatial data infrastructures. Luxembourg: Publications Office.

Tyner JA. 2010. Principles of map design. New York: Guilford Press.

Vretanos P. 2014. OGC® Filter Encoding 2.0 Encoding Standard – With Corrigendum (OGC 09-026r2).

Wood D., Fels J. 1992. The power of maps. New York: Guilford Press.

Wu M., Zhu A., Zheng P., Cui L., Zhang X. 2016. An improved map-symbol model to facilitate sharing of heterogeneous qualitative map symbols. Cartography and Geographic Information Science:1–14. DOI: 10.1080/15230406.2015.1102083.

Zipf A. 2005. Using Styled Layer Descriptor (SLD) for the Dynamic Generation of User- and Context-Adaptive Mobile Maps – A Technical Framework. In: Li K-J, Vangenot C eds. Web and Wireless Geographical Information Systems. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 183–193.