Multi-source Cartography in Internet GIS

Peter van Oosterom, Theo Tijssen, Ingrid Alkemade, Marian de Vries
Delft University of Technology,
Department of Geodesy, Section GIS-technology
Thijsseweg 11, 2629 JA Delft, The Netherlands
Tel (+31) 15 2784268
Fax (+31) 15 2782745
Email: m.e.devries@geo.tudelft.nl

Abstract

A new development in Internet GIS is the use of distributed data sources: the user chooses data from different Web sites and combines these sources into integrated maps. These new possibilities in Web Mapping also have their drawbacks: combining different geo-data sources may result in conflicting presentation styles and consequently in confusing maps. Colors or symbol-sets of different feature layers may be the same or too much alike. The intended presentation scale of one layer might not combine with that of other layers, etc. The supplier of the data does not have prior knowledge of the possible combinations of the data in different (map-)products and contexts by the 'remote' users. The user on the other hand may not have enough cartographic or domain expertise to be able to see and resolve visualization conflicts. Some kind of intelligent cartographic presentation software is needed to solve these problems. In this paper some of the issues of multi-source cartography are discussed. A prototype for a Cartographic Expert System (CES) is presented. The focus in the example will be on the resolution of conflicts of color.

1. Introduction

Although Internet GIS is a relatively new development, this does not mean that required research has to begin from scratch. Within the cartographic community there is a long-standing discussion on issues of automated map design. The focus has primarily been on generalization as a way to solve problems of scale. Multi-source cartography and generalization have many aspects in common: both are driven by the tasks and requirements of the user, both need understanding of the meaning of the geo-information (its semantics). Sometimes the same type of solutions may be used (changing, displacing or not showing certain objects or even complete themes); similar tools may be used in the implementation of solutions (expert systems, agent technology, spatial algorithms).

Generalization has been studied with two purposes in mind (Muller 1995): the first purpose was to find ways to derive several maps, each of a different scale, from the same data source. The second purpose was to use generalization as a means to produce 'readable' digital maps at different scale levels. It is this second purpose of generalization that is relevant in the area of Web-mapping and other forms of end-user-mapping. The possibility of ‘zooming’ - one of the most common functions in GIS-applications - means that scale continually changes. Generalization-algorithms could be very helpful in Internet GIS to secure that on-the-fly maps conform to some basic principles of cartography like: ‘avoid confusion’, ‘do not overwhelm with details’.

Recently the AGENT-project, funded by Esprit, has produced interesting insights in generalization procedures, problem-solving methodology etc. (Lamy 1999). The acronym ‘AGENT’ stands for ‘Automated Generalization New Technology’. An agent is a relatively autonomous piece of software, which tries to find a good solution for its own client. This in collaboration with other agents, each trying to solve a certain (relatively) small problem. Important is the fact that there is no single monolithic software program trying to solve everything. In the AGENT-project both conflict detection and measurement and conflict solution is handled by ‘agents’.

Another branch of research that is relevant for Internet GIS is that into the use of knowledge-based systems in cartography. These systems have also been applied in generalization, but the DESCARTES-project is one example of another application area: analysis and exploration of statistical information (Andrienko 1999). Main goal of this cartographic expert system is to guide the non-expert user in choosing presentation styles for statistical information related to spatial units, in the form of pie-charts, choropleths, etc. In this respect the CommonGIS-project should also be mentioned (Andrienko 1999b): it combines the functionality of DESCARTES with a Java-product for Internet GIS: Magma/Lava. The Magma/Lava software has also been used in our prototype (see Section 5).

Complicating factor in Web-mapping is the unpredictable nature of the map design process by end-users via the Web: there is no predefined set of geographic data that is displayed, the user can be anybody, with unknown task and purpose to use the geo-data, with or without cartographic or domain knowledge. Our research has been not so much on scale but more on problems of color in ‘on-demand’ mapping in a multi-source environment. We tried to set up a knowledge-based system not for data analysis and exploration purposes but for the detection and resolution of conflicts in presentation styles in situations where data sources from different data providers are combined in Web-mapping. Of the graphic variables mentioned by Bertin (Bertin 1967) we initially focused on color and size.

Figure 1: Multi-source Internet GIS

In Section 2 the different aspects of Internet GIS are discussed together with a short description of three types of Internet GIS architecture (and their relationship). The problem of the visualization conflicts and several solution approaches are discussed in Section 3. Special attention to color models is given in the subsequent section. The prototype is described in Section 5.

2. Internet GIS architecture

The success of Internet in general has shown the power of open standards. This will also be true for the use of geo-information over the Internet. In order to create the Geo-Information Infrastructure based on the Internet, the following aspects have to be covered (between brackets some relevant OpenGIS standards are mentioned):

1. consensus on the geometric model, both raster and vector data

2. how to describe the geo-data sets (and geo-processes), that is metadata

3. how to access and query the metadata (OpenGIS catalog service specifications)

4. how to select the geo-data itself (OpenGIS web-mapping server specifications)

5. how to format (and transfer) the resulting geo-data set (OpenGIS GML)

Many different Internet GIS architectures do currently exist. Nearly every GIS vendor supports at least one or two of these approaches. The OpenGIS Consortium has developed a model to compare these different approached to Internet GIS (see Figure 2):

1. Thin clients (only raster images JPEG and PNG),

2. Medium clients (graphic primitives WebCGM and SVG) or

3. Thick clients (data in the form of simple features XML, that is GML, processed at the client side).

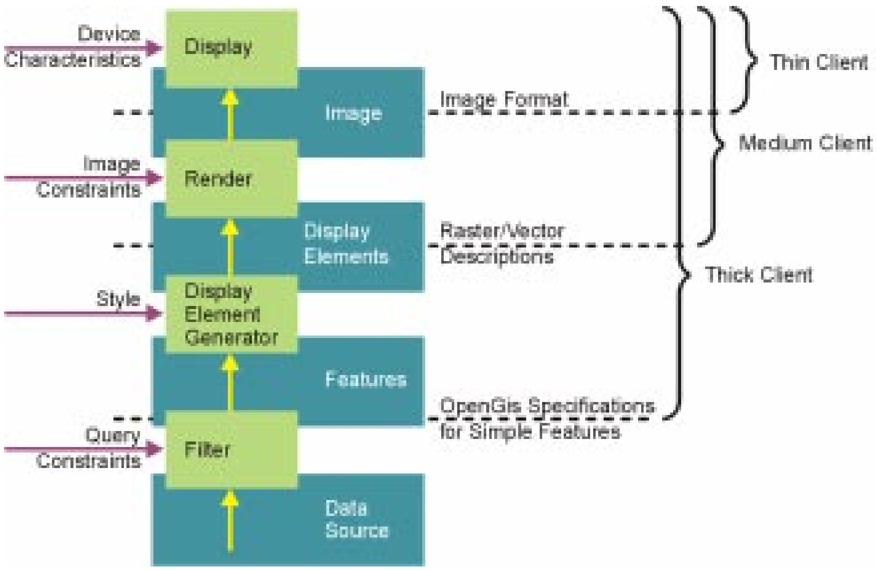

Figure 2: Internet GIS architectures

When a certain process is not done at the client side, the work has to be done at the server side. E.g. generate graphic primitives from GML data and/or convert graphic primitives into images (rendering). In order to detect and solve visualization conflicts, the 'thick client' architecture is the most suitable, because presentation styles have not yet been applied to the data.

3. Visualization conflicts

Apart from the risk of too much detail in Web-maps (problem of scale) other problems must be tackled in this kind of on-demand mapping. Some examples of conflicts that can occur when several map layers from different data providers are combined:

- two or more feature layers end up with the same color, size and/or pattern. Only users that are familiar with legends and classification will notice;

- features are not visible at all because they overlap (intersect) while having the same color;

- colors or symbology of the features are not exactly the same, but are still not distinguish able enough.

Whether or not these kinds of conflicts are a problem for the quality and readability of the map depends on a number of factors:

- are the features of the same geometry type – or in cartographic terms: graphic symbol type

- is there overlap between the features somewhere in the map extent?

When two feature layers of the same geometry-type have (almost) the same color there always is a conflict, whether the layers overlap (intersect) or not, because in all cases the classification in the legend would be confusing. On the other hand, when two feature layers are of a different geometry-type (polygons and point-symbols, or line strings and polygons) there only is a problem when the layers overlap somewhere in the map extent: a green point-symbol for a prehistoric burial ground (to name an example) will not be visible in a green forest, a green symbol for a very ancient tree in a (red) town-square will be visible.

These are rather obvious considerations for cartographers, but when this kind of 'knowledge' has to be formalized in order to be incorporated into a knowledge base, even simple rules can be hard to quantify. In the remainder of this section a number of approaches to solve (or avoid) the visualization conflicts are described.

3.1 The semantic approach

In an ideal world visualization conflicts in Web-maps could perhaps be avoided. Suppose there is a worldwide geo-information-thesaurus that lists all the possible object classes, their names (in different languages), their meaning, the domain of their attributes, the preferred presentation variables (color, texture/pattern, symbol) that should be used in their classification and presentation. The Open GIS Consortium uses the notion of Information Communities, groups of domain experts that should set standards on these semantic aspects of geo-information. To be able to 'translate' data models and thesauruses between the different Information Communities a system of semantic 'translators' is proposed (OGC 1996).

Even if such an (international) thesaurus could ever be made, the amount of different object classes plus their classifications would surpass the amount of distinguishable colors and/or symbols that can be used in map design. Consequently methods for conflict detection and conflict resolution for on-demand mapping are still necessary.

3.2 The metadata approach

When it is not possible (yet?) to work with worldwide thesauruses, other mechanisms are needed. Geographic data is currently delivered as the data itself and metadata like format, projection, attribute-definition, actuality and accuracy (quality of the data). The metadata enables the user to decide whether the data is suitable for his or her purpose. A possible extension to the metadata currently supplied would be a recommended presentation style plus a second best and a third best choice of e.g. color and pattern. This information can be used in a knowledge-based cartographic system to solve conflicts in the presentation of the distinct data sources. In this light it is useful to make a distinction between the Digital Landscape Model (DLM) where the data itself is present, and the Digital Cartographic Model (DCM) which is the graphical presentation of the geographic data (Harbeck 1995). Note that different DCM's for the same DLM are possible depending on the task and context of the user. For the use of an Internet GIS, it is important that the data supplier provides not only the DLM, but also the specifications to produce the DCM.

Cartographic metadata (preferred presentation parameters like color, pattern, symbol) should be stored together with the geo-data in the spatial database (other words for metadata could be: 'graphic model' or visualization constraints). Important is that these constraints are defined at object class and maybe even at attribute level (since a geo-object or feature can sometimes be classified on more than one attribute). The way the cartographic metadata is transferred to the client-software depends on the way the geo-data itself is distributed. In a download situation exchange formats like INTERLIS (Keller 1998) could be extended into specifying more than one graphic model per data set, with an indication of the preferred model, the second best and third best model. In the hydrographic world the standards for the data model (S57) and associated presentation styles (S52) are separated (IHO 1996). It should be noted that up to four sets of presentation styles are given, each one suited for a certain situation (varying from a dark night to a bright day situation).

Lehto (Lehto 2000) has suggested a combination of the eXtensible markup Language XML (in the case of geo-information, the Geography Markup Language GML) for the data itself and XML Style Transformation (XSLT) for the visualization metadata. With the XSLT-file the XML (GML) document can be transformed into for example the Scalable Vector Graphics format (SVG) of the W3 Consortium. This format can be displayed in most web-browsers (sometimes using a plugin) or Java-viewers. Although not all geometry can be depicted correctly yet, SVG could be a step towards interoperability as far as the visualization of the DCM (described with graphic primitives) is concerned (Neumann 2000).

The Open GIS Consortium is at the moment working on a standard for the specification of presentation variables. Just recently a discussion paper has been released, as an extension of the Web Map Server interface specification, with a proposal for a 'Styled Layer Descriptor' (SLD) standard syntax. With SLD (XML-tags) it is possible to describe the symbolization of feature data that is transferred by the Map Server (OGC 2001). See the XML in figure 3.

```xml
<?xml version=”1.0" encoding=”UTF-8"?>
<!DOCTYPE StyledLayerDescriptor SYSTEM “SLD.dtd” >
<StyledLayerDescriptor>
  <UserLayer>
    <FeatureTypeConstraint typeName=”main-river”/>
    <UserStyle title=”Blue river”>
        <LineStringSymbol>
            <Geometry>
                <FetchFeatureProperty name=”center-line”/>
            </Geometry>
            <StrokeColor>#aaaaff</StrokeColor>
            <StrokeWidth>5.0</StrokeWidth>
        </LineStringSymbol>
    </UserStyle>
  </UserLayer>
</StyledLayerDescriptor>
```

Figure 3: Styled Layer Descriptor example (in XML format)

After each NamedLayer or UserLayer tag a NamedStyle or UserStyle tag is defined. The SLD can be included in the answer to the Map-request itself; but it is also possible to use an URL-reference to a separate SLD-file (also in XML format). The names of the style parameters (like StrokeWidth, StrokeColor) are drawn from the SVG-specification of the W3-Consortium. In this respect this proposal would be a step forward in standardization. Again, also in this new initiative of OGC we see a separation of the geo-data itself (DLM) and the visualization parameters (DCM-specifications). It's implementation fits perfectly into the ‘thick client’ architecture with separate DLM's and DCM's (see Section 2).

The Styled Layer Descriptor-specification could be useful for the resolution of visualization conflicts in multi-source Web-mapping if it would be extended with the possibility to describe not one but several presentation styles for each layer. Like ‘candidate keys’ in data modeling-syntax, these could be named ‘candidate styles’ to be used when layers are combined and alternative colors have to be assigned.

3.3 The constraints approach

In the AGENT-project, a geo-object has knowledge of itself and its surroundings. At class level the knowledge consists of, among other aspects, a set of constraints that should be obeyed. At instance level the specific local situation is available, enabling agents to detect conflicts (according to the constraints and the local situation of a geo-object) and to solve the conflicts. For example, when two objects are too close, it is tried to solve this problem locally; e.g. by merging the objects, or moving the objects or not displaying one of the objects at all, etc.

Meso-agents are necessary to look at possible conflicts at a higher level: after each iteration of changes in size, shape, location of features during the automated generalization process, the meso-agents must again evaluate the new situation. It is possible that one problem has been solved (e.g. hedges and borders beside roads have been replaced by just borders) and another one is created (e.g. the width of the borders is too large in relation to the line-width of the roads).

3.4 The expert system approach

The core element of the previous approach is the notion that objects themselves contain methods to detect and solve presentation conflicts. In the expert system approach this role of problem detector and solver is played by a cartographic knowledge-based (expert) system. When a combination of several DLM's is made, the specifications from the multiple individual DCM's could be used by the Cartographic Expert System (CES) to determine the optimal presentation of these sets of data: the integrated DCM. This is done at the class level (and not yet looking at individual instances) by taking into account the cartographic rules and, if possible, the priorities of the user related to his or her (unknown) task. Generic cartographic rules are stored in an expert database. This knowledge base could be a separate expert system or an integral part of the spatial database (at the server side). Oracle 8i for example offers the possibility of Java 'stored procedures'.

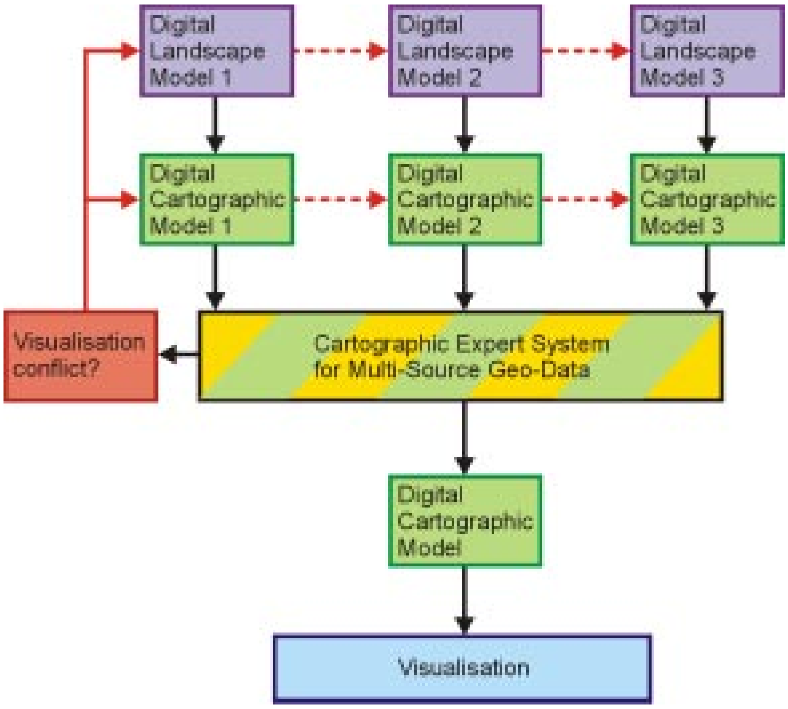

Figure 4: CES (3 DLM's and DCM's -> Integrated DCM)

A CES uses cartographic rules in the form of an IF [applicable condition] THEN [recommendation] where the applicable conditions are facts and procedural knowledge. A recommendation is a task or a reference to another rule. The development of a CES has two major parts. The first part is to transform cartographic knowledge into rule-based knowledge. Some points of attention can be distinguished in the graphical model that can be seized into rules. These points of attention are scale and scale domain, generalization, semantics, text placement, importance of themes, graphic variables like color, texture/pattern, and size.

The second part is to implement this expert system in (client-) software. A method has to be developed to detect conflicts in e.g. color or pattern. Furthermore a user-friendly interface has to be designed for the expert system to help the (non-professional) user in producing a map. This can be implemented by offering the user some kind of optimizer, e.g. an optimizer-button. Optimization is based on the rules in the Cartographic Expert System in combination with the original DCM-specifications that come with the data and the priorities (wishes) of the user. The user should still have the possibility however to manipulate the legend, thereby overruling the proposed visualization.

4. Color models

When no cartographic metadata (presentation styles) are supplied with the geo-objects by the data provider, but only 'hard coded' values of the presentation parameters are available, existing knowledge of computer graphics and color models could be used. A basic principle (rule of thumb) could be in case of a conflict, to start with changing the Value (or lightness) and keep the Hue (the color itself) the same. In this way the risk of choosing an (implied) semantically confusing color is minimized. For example, a layer that depicts 'Water' will stay blue (although a lighter or darker blue) and not turn green.

Even when preferred presentation styles in the form of cartographic metadata is provided by the data source, rules based on color models are important. Existing knowledge of computer graphics and human vision (interpretation) could provide constraints for the 'change color'-algorithm. Minimum and maximum Value per color could be specified and incorporated in the CES-rules to avoid undistinguishable color-combinations (see also: Kraak 2001).

5. A Java-client example

In order to test some of the issues of multi-source cartography a CES-prototype is being developed at the Delft Department of Geodesy, Section GIS-technology as part of a number of MSc-thesis projects (Alkemade 2000). The prototype was based on three design considerations:

- for each data source cartographic metadata is available in the form of preferred presenta tion styles and two or more 'candidate' presentation styles,
- the software intelligence is provided at the client-side of the Web-mapping configuration,
- the data are in vector format.

The prototype was developed in Magma/Lava, Internet GIS software of PGS. Magma/Lava exists of two parts (Van der Berg 1997):

- Lava: the browser software implemented as Java applet, for the presentation of geo-data in a multi-layered map view,
- Magma: a C++ Application Server that translates Web-server requests to the spatial data base engine (Oracle 8i in our prototype-setting, but in general able to deal with heteroge neous environments; e.g. other databases, such as Ingres or Informix, or file formats such as DXF).

The way the different layers are visualized in Lava is handled by a Java-class (called LavaShape). In the class-hierarchy of Lava an unlimited amount of subclasses can be defined to present different geometry types / graphic primitives. The most obvious are classes to visualize polygons, points and line strings. A LavaShape-object contains all properties and methods that are necessary for the graphic rendering of the geodata-object:

- the geometric data itself (the coordinates of the vertices)
- attributes like color, fill, line-width
- links to other objects (topological relations)

- event handlers (what to do when a user clicks on an object-instance etc.)

For the purpose of the CES-prototype the LavaShape-class has been adapted: instead of one list of graphic attributes per layer three property-lists are defined and filled after the cartographic metadata has been transferred from the source to the client. At the source the cartographic metadata is currently stored in a vendor specific format (see figure 5).

```
Layer (
  name = “HuisnummerAdressen”,
  relation = lookup(“www.gdmc.nl/cgi-bin”,”tilburg”,”HUISNUMMERSMGM”),
  lavaShapeSpec=(
    type = “PointLavaShape”
    arguments = (“HUISNUMMERSATTR”)
    properties = PropertyList(“color=0,Shape=4”),
    propertiesTwo = PropertyList(“color=3,Shape=4”),
    propertiesThree = PropertyList(“color=2,Shape=4”),
    attributes = AttributeList(“‘prefix:’,
    ‘http://www.gdmc.nl:81/info/vraagadrea?adres=’,’links:’,ADRESNR,
    ‘target:’,’main’”)
    ),
  visible = false,
  scaleRange=(from=0.01,to=2.5)
  colorMap = lookup(“standard”)
  )
```

Figure 5: Magma/Lava ‘configuration file’ example

In the example in figure 6 four feature layers have been selected by the user: buildings, railway-tracks, parcel-boundaries and main roads. The railway-layer and the parcels-layer have the same color (black), the same is true for the buildings-layer and the main roads (both red).

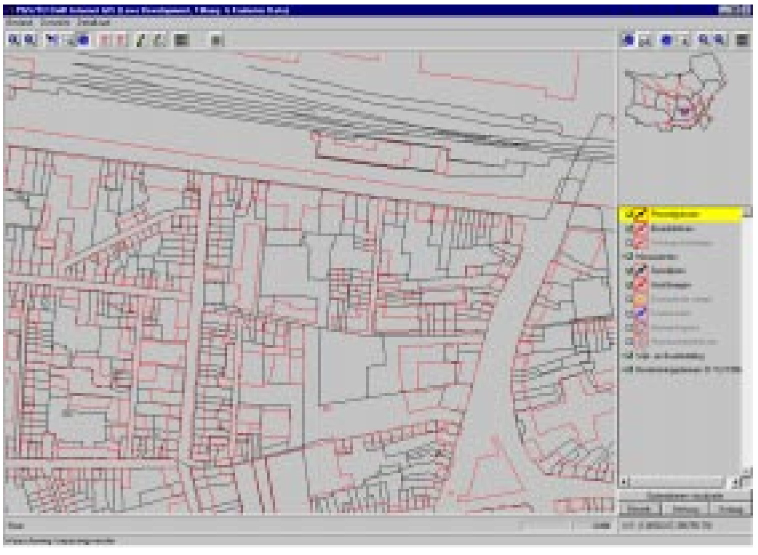

Figure 6: before optimization

When the user presses the optimizer-button (lower right, below the Legend-frame) the optimizer-Java-class is activated. After optimization two of the four colors have been changed: the parcel boundaries are now yellow, the main road has been changed both in color and in line-width (figure 7).

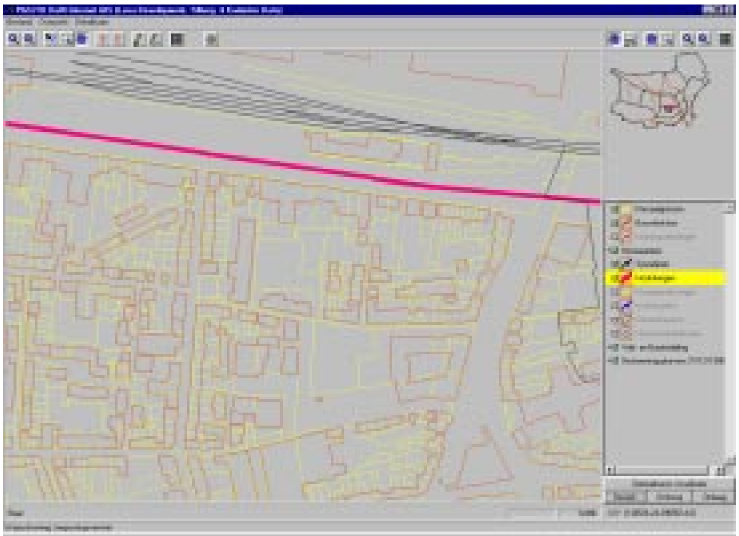

Figure 7: after optimization

5.1 Discussion related to the prototype

1. Does one feature layer have more ‘right’ to keep its first choice presentation style? If the user explicitly indicates the (relative) importance of the classes, this would be possible. In our first prototype we tried to avoid putting this burden on the user by giving more weight to features according to the ordering of the layers in the Table Of Content of the map view. The layer on top of all the others (drawn as the last layer) always kept the original presentation style, because the user decided to put this on top of the other layers (indicating that it might be the most important layer). In case of a color conflict layer number two, three etc had to change their presentation parameters. A refinement would be to give more weight to the feature layers that are graphically more prominent, i.e. that occupy the largest ‘area’ in the combined map. In this approach polygons would go before lines, lines before point-symbols. When there are conflicts between features of the same geometry-type the total area (in the case of polygons), the total length (in the case of line strings) or the number of feature-instances (in the case of points) could be used as criteria to establish the sequence in which the presentation style of the feature layers is changed by the CES.

2. What if the iterative process of color adjustment of conflicting themes exhausts the preferred presentation styles for each layer? Should the optimization-algorithm choose random colors instead until there are no color conflicts any more? It is good to see the analogy with the generalization problem. ‘Solutions’ could be: decide not to display a certain class (or instance) at all (because conflicts can not be solved at all, this should be made clear to the user; e.g. by marking this in the legend) or decide to change geometry/displace the geo-object to avoid conflict (again this must also be made clear to the user).

3. Must the software also detect whether geo-objects from multiple sources are semantically similar and how should these be treated? One approach could be only to display the geo-objects with the best (geometric, temporal, thematic) quality. Another approach could be trying to derive integrated geo-objects inheriting the best of the multiple sources and displaying this best geo-object according the rules of the integrated DCM.

6. Conclusion

The use of digital maps on Internet will become even more widespread in the future. Knowledge-based systems could help to secure cartographic quality when geo-data is combined in on-demand maps. Important issues to look into are:

- the concept of cartographic metadata: more than one preferred presentation style
- transfer of the metadata: in separate files (for example XML for the data and XSL(T) for the transformation from DLM to DCM) or in the same files (Graphic Model in INTERLIS)
- iterative process of conflict detection and solution: how to evaluate the results of color adjustments. Has the map improved?

Acknowledgments

We would like to thank the municipality of Tilburg for the permission to use their data and PGS for making the Magma/Lava Internet GIS software available for this research.

References

Alkemade, I (2000): Beeldschermkartografie ten behoeve van multi-bron Internet GIS, MSc-thesis Department of Geodesy, Delft University of Technology, December 2000

Andrienko, G. and N. Andrienko (1999): Knowledge Engineering for Automated Map Design in DESCARTES. In: C.B.Medeiros (ed.): Advances in Geographic Information Systems. Proceedings of the 7th International Symposium ACM GIS'99, Kansas-City, November 5-6, 1999 NY: ACM Press, pp. 66-72

Andrienko, G. and N. Andrienko (1999b): Making a GIS Intelligent: CommonGIS Project View AGILE'99 Conference, Rome, April 15-17, 1999, pp. 19-24

Bertin, J. (1967): Semiology of Graphics, reprinted by University of Wisconsin Press, Madison, WI, 1983

Harbeck, R. (1995): Überblick über Konzeption, Aufbau und Datenangebot des Geoinformationssystems ATKIS. In: Kophstahl, E.& Sellge, H. (Hrsg.): Tagungsband zum AdV-Symposium ATKIS am 27. und 28. Juni in Hannover. S. 19 - 37

IHO (1996): IHO Publication S-52 -Specifications for Chart Content and Display Aspects of ECDIS, 5th Edition, December 1996

Keller, S.F. and H.Thalmann (1999): Modeling and Sharing Graphic Presentations of Geospatial Data. In: A. Vckovski, K. E. Brassel, H.-J. Schek: Interoperating Geographic Information Systems. Proceedings of the 2nd International Conference, INTEROP'99, 10-12 March 1999. Lecture Notes in Computer Science Vol. 1580. 1999. Springer Verlag, Berlin, pp.135-146

Kraak, M.J. (2001) and A. Brown (eds.): Web Cartography: Developments and Prospects. London: Taylor & Francis

Lamy, S., A. Ruas, Y. Demazeau, M. Jackson, W. Mackaness and R. Weibel (1999): The Application of Agents in Automated Map Generalization. Proceedings 19th Int. Cartographic Conference, Ottawa

Lehto, L. (2000): XML in Web-based Geospatial Applications. 3rd AGILE Conference on Geographic Information Science, Helsinki/ Espoo, May 25–27, 2000, pp. 162–167

Muller, J.C., J.P. Lagrange and R. Weibel (1995): GIS and Generalization: Methodology and Practice. London: Taylor & Francis

Neumann, Andreas and Andreas M. Winter (2000): "Vector-based Web Cartography: Enabler SVG", in: http://www.carto.net/, Zurich/Vienna

OGC (1996): OGIS Information Communities Model - Managing Data Heterogeneity, in:

The OpenGIS™ Guide, Introduction to Interoperable Geoprocessing, Part I of the Open Geodata Interoperability Specification, Open GIS Consortium, 1996

OGC (2001): OpenGIS Styled Layer Descriptor Draft Candidate Implementation Specification v0.7.0, OGC Document Number 01-028, 7 February 2001

Scalable Vector Graphics (SVG) 1.0 Specification W3C Candidate Recommendation, *2 November 2000*

Van der Berg, C., F. Tuijnman, T. Vijlbrief, C. Meijer, H. Uitermark and P. van Oosterom (1997): Multi-server Internet GIS: Standardization and Practical Experiences, International Conference on Interoperating Geographic Information Systems, INTEROP'97, Santa-Barbara, CA, December 3-4, 1997