RESEARCH ARTICLE

# Automatic integration of spatial data in viewing services

Ara Toomanian[1,2], Lars Harrie[2],
Ali Mansourian[2], and Petter Pilesjö[2]

[1]Department of Cartography, University of Tehran, Tehran, Iran
[2]Department of Physical Geography and Ecosystem Science, Lund University, Sweden

**Abstract:** Geoportals are increasingly used for searching, viewing, and downloading spatial data. This study concerns methods to improve the visual presentation in viewing services. When spatial data in a viewing service are taken from more than one source there are often syntactic, semantic, topological, and geometrical conflicts that prevent maps being fully consistent. In this study we extend a standard view service with methods to solve these conflicts. The methods are based on: (1) semantic labels of data in basic services, (2) a rule-base in the portal layer, and (3) integration methods in the portal layer. To evaluate the methodology, we use a case study for adding historical borders on top of a base-map. The results show that the borders are overlaid on top of the map without conflicts, and that a consistent map is generated automatically as an output. The methodology can be generalized to add other types of data on top of a base-map.

## 1 Introduction

Many countries are working with creating and improving spatial data infrastructures (SDIs). A geoportal is a key component of an SDI, used for searching, viewing, and downloading spatial data and services [10]. Viewing services in a geoportal enable users to view spatial data from other basic services. A common use of viewing services is when application-specific data are overlaid on top of a base-map. In the visual hierarchy, application-specific data should be placed in the foreground and the base-map in the

background. One problem with the integration of application-specific and base-map data is the introduction of semantic, topological, and geometrical conflicts. These conflicts may cause the map data to be inconsistent (see [12], Figure 1). Such conflicts are caused by inaccuracy and imprecision in the data; heterogeneity in information models; different levels of detail in the geometric representation; and so forth. To solve conflicts, the relationships between the application-specific data and the base-map data must be known. These relationships can be modeled in a rule-base. Furthermore, integration methods are needed that can solve conflicts based on the rule-base. In this study we developed methods to adjust application-specific data to the base-map. The rationale of maintaining the geometry of the base-map is twofold: (1) the geometric quality of the base-map is generally better than the application-specific data; and (2) the geometry remains consistent irrespective of which application-specific data is added.
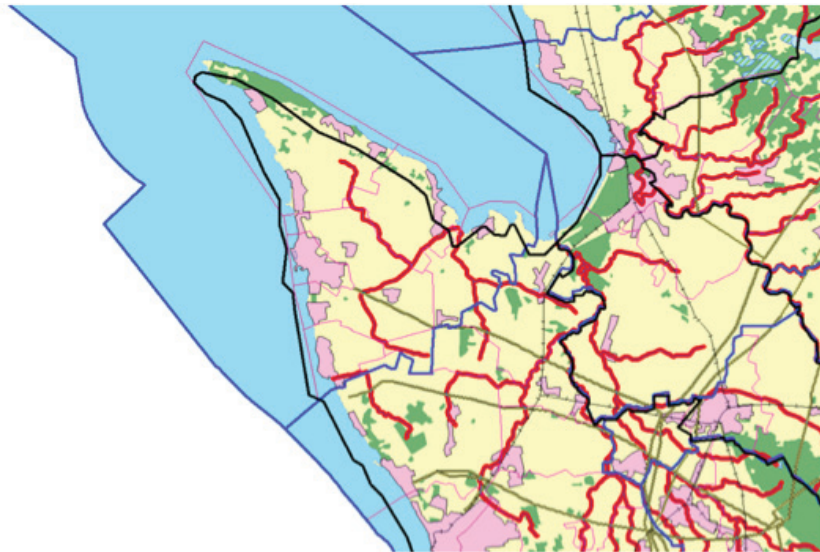
Figure 1: The shore line (black) in the foreground does not follow the shoreline in the background data: geometric conflicts cause the maps to be inconsistent.

The aim of this study is to solve the topological and geometrical conflicts of data in a viewing service. To accomplish this, we develop a method based on: (1) semantic labels of data in basic services, (2) a rule-base at the portal level, and (3) integration methods in the portal layer. The methods are implemented in a system architecture based on open source products.

The structure of the paper is as follows: first, the basic technologies and related studies are described (Section 2). In Section 3, we describe the methodology and the system architecture. Section 4 describes implementation, and Section 5 presents a case study. The paper ends with a discussion followed by conclusions (Sections 6 and 7).

# 2   Basic technology and related studies

In order to implement automated integration methods for spatial data in geoportals, solutions to several tasks are needed. First, standards for syntactic interoperability between the client, geoportals, and basic services of spatial data are required. Second, spatial data should be self-describing in such a way that the geoportal can interpret its meaning, i.e., semantic labeling of the spatial data is needed. Third, rules should be defined for the visual integration of spatial data. These rules should be included in a rule-based system that selected the best rules to apply. Fourth, an integration method should be developed to modify datasets based on rules to remove conflicts. In this section we review standards, techniques, and previous work for these tasks.

## 2.1   Syntactic interoperability of geoportals

There are a number of standards used in geoportals to satisfy syntactic interoperability. WMS (web map service) is an Open Geospatial Consortium (OGC) standard that enables a user to view a map from a client via a remote server over the Internet. The output of a WMS is a raster format (PNG, GIF, or JPEG) or a vector-based graphical format such as SVG (scalable vector graphics) [19]. WFS (web features service) is an OGC standard for distributing spatial data in vector format over the Internet in GML (geography markup language) format [23]. Finally, SLD (style layer descriptor) and SE (symbology encoding) are OGC standards that specify the feature symbolization using XML-based descriptions [18].

## 2.2   Semantic labeling of spatial data

Semantic interoperability, in this context, relates to the clear definition of spatial data in such a way that user,s or even systems, have a common understanding of the semantics of the data. The World Wide Web Consortium (W3C) has developed standard languages and formats for writing and storing semantic labeling, e.g., RDF (resource description framework) and OWL (web ontology language) [8, 20].

In recent years, there have been a number of studies conducted to perform semantic labeling. Some researchers attempted to find automatic methods to enhance the usability of distributed and heterogeneous geographic data sets. In this context, semantic interoperability is an issue that can be addressed by using semantic annotations of geodata and geospatial domain ontologies. Klien and Lutz [14] proposed a method for automating the annotation process based on spatial relations. According to their method, spatial relations have an important role for defining and identifying geospatial concepts at the domain level. They argue that spatial relations may be expressed through spatial processing methods, because relations like topology, direction, or distance between two spatial entities can be calculated at the data level. Klien and Lutz show how this potential can be exploited for automating the semantic annotation of geodata.

Semantic labeling can be stored and distributed using a specific type of ontology base-language. Using semantic labeling (e.g., in OWL), data can be interpreted by the system (e.g., a geoportal) and then further processed by rules defined on top of the semantic layer. This idea is central to creating an expert geoportal and automatic service composition (see, e.g., [9,13,17]). In this study we implement rules that utilize semantic labeling information.

## 2.3    Rule-based system that utilities semantic labeling

In general, a rule-based system consists of set of rules (rule base), a working memory, and an inference engine. Rules encode domain knowledge and business logic as condition action pairs. The working memory represents system input first, but the actions that occur when rules are fired can cause the state of the working memory to change. The inference engine runs a method to fire rules producing new data for application-specific layer adjustment.

Both rules and semantic descriptions of data can be defined based on a standard ontology language such as OWL. The rules are IF-THEN statements in which the IF clause describes the data and its relationships, and the THEN clause describes the operation(s) which should be applied to remove conflicts.

The inference engine includes algorithms that specify the way in which the rule-base is used to reach the final result. Two general methods are used in expert systems for inference [22]: forward chaining and backward chaining. In forward chaining, the inference engine starts with primary conditions and seeks rules which may be applied to these conditions (in an IF clause). In backward chaining, inference begins with the desired goal (in a THEN clause). The system then works backwards, searching for data that can be used with rules to arrive at that goal.

Recent research has investigated using semantic labeling rule-bases. Fan and Wang [11] proposed a rule-based semantic matching strategy. They argued that existing semantic web service discovery technologies focus only on keyword-based or primary semantic-based service matching. Fan and Wang first studied the rule-based service-matching algorithm in the context of a large-scale services library and the formal descriptions of semantic web services and service matching. Fan and Wang divided services' matching into different levels. Then, a set of matching rules was given. The related services set was retrieved from the service ontology-base through rule-based reasoning to determine their matching levels. Their experiment showed that the proposed service-matching strategy achieves high service discovery efficiency in comparison with a conventional global traversal strategy. Another study concerned a semantic knowledge-base for religious thangka images [25]. The authors first analyzed the basic semantic content of a religious portrait, adopting structured semantic metadata to describe domain knowledge with the aid of an ontology of thought. Then, [25] identified key semantic information from the metadata to build a knowledge-base, and utilized IF-THEN structures to store that knowledge. Finally, with the help of a decision-tree, they made use of the forward chaining inference to retrieve certain pictures. In our method, we similarly investigate a rule-based system that uses semantic labeling.

## 2.4    Integration methods

Solving geometrical and topological conflicts in a view service is similar to the general problem of combining two spatial datasets, i.e., conflation. Conflation is the process of merging two datasets in order to improve the quality of the resulting output [13]. In an early study of conflation, Jones et al. present a method for identifying homologous administrative areas in two data sets [13]. In their study a rule-based system was used that included properties of polygons, the arcs that make up those polygon, and semantic properties. Olteanu and Mustière [17] developed a conflation method based on belief theory. They select matching candidates (one-to-one or one-to-many relationships) by a weighting of geometrical, toponym, and semantic criteria.

To solve the geometrical discrepancies between the datasets in Figure1, we need to perform network matching/integration—a well-studied problem. Walter and Fritsch developed a statistical approach for integration of road data sets, where they considered statistical differences between the homologous road objects in geometric (length and angle) and topological (connectedness) aspects [24]. Several other studies have also been performed on the integration of road data sets [16, 21, 26]. It is common for these studies to use combinations of semantic, topological, and/or geometrical properties of the roads. Often, the studies use the name of roads as a first criterion, and then use matching of close nodes and arcs when considering connectedness between the arcs.

# 3 Methodology

This section starts with a description of the system architecture followed by details about the methods used. A more detailed description of the system architecture is given in [3].

## 3.1 System architecture

Our system architecture includes the following components (Figure 2):

**Client**   Our client is a WMS-client in which a user can specify if a layer is an application-specific layer or if it belongs to the base-map (Figure 3).

**Registry service**   Our registry service manages the registry of spatial services to be used by the geoportal.

**Cartographic enhanced geoportal**   The cartographic enhanced geoportal is a geoportal with added functionality to enable improved cartography. The geoportal consists of four components:

- The *cartographic core* determines the symbolization of the layers and interacts with the definition of the layers to select the type of visualization. In this component, several cartographic methods can be implemented.
- The *SLD* library contains one or more symbolization(s) for each dataset registered in the geoportal.
- The *expert system* checks the data definitions, and determines the proper visualization method according to the selected feature type. The expert system then instructs the cartographic core, based on the nature of the dataset and the predefined integration rules for different layers.
- The *semantic label library* is an OWL document in which the semantic labels of the layers and their relationship are saved.

**Basic services**   Basic services are for distributing geographic data. In this architecture, basic services are download services (WFS) and viewing services (WMS).
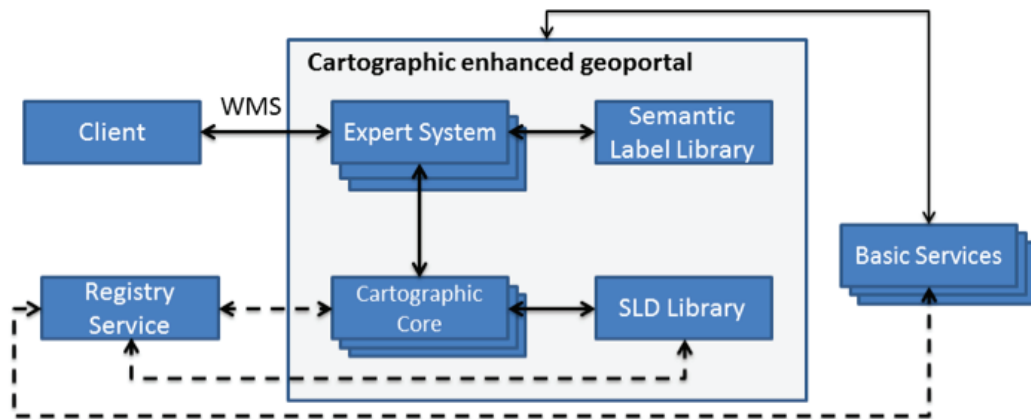
Figure 2: System architecture.



Figure 3: Graphical interface for specifying the layer in the client ("AS" stands for "application-specific").

## 3.2 Rule-based system

The rule-based system consists of rules and the relations between them. In our study, these rules are translated to IF-THEN statements and solved by forward chaining. The combination of a number of rules grouped together (IF clause) indicates a result (THEN clause). As discussed above, in this step we have a number of base-map layers that use predefined spatial rules. The result of applying these rules produces new conditions, and the system then seeks rules in which these new conditions can be applied. This iteration continues until the output of one of the rules matches that specified by the user. In the forward-chaining approach, external components (foremost the object refinement method) are used to solve the integration of the data. The integration of geographic data is based on rules

of how the data from different data sets is related (see Table 1). The rules are based on either spatial/topological relations (e.g., a layer may coincide with another one); or distance/tolerance relations, which may affect the application-specific layer displacement on the base-map layers (a layer with a distance less than a threshold may be replaced by another layer).

## 3.3 Object refinement method

The cartographic enhanced geoportal enables the user to decide which of the layers (that are registered in the portal) are application-specific data, and which belong to the base-map (cf. Figure 3). The application-specific data are then adjusted to fit topologically and geometrically to the base-map data. The rules for how to adjust the application specific data are retrieved from the rule-based system. To adjust the application-specific data, we use a procedure terms "object refinement," which is applied once for each application-specific layer. The object refinement method uses a data matching technique that borrows several components from [16, 21, 26].

In short, we can describe the object refinement method as follows. For each link in the application-specific layer we investigate if the whole or parts of the links are sufficiently close to a link in the base-map (according to the rules in the rule-base). If so, we replace the geometry of the link (or part of the link) in the application-specific layer with the geometry in the base-map. In this way, we arrive at a fast method that can be used in a real-time system. The drawback is that there are some geometric configurations that are not handled properly (e.g., if only the middle part of a link in the application-specific layer corresponds to a link in the base-map).

In more detail, the object refinement method consists of the following steps:

1. Construct a line network (denoted $network_{BM}$) to which the application layer should be adjusted. This network is constructed from a subset of the base-map layers, chosen using information from the rule-base.
2. Read the next node from the application-specific layer list, denoted $node_{AS}$.
3. Find the closest node in $network_{BM}$ to $node_{AS}$. Denote this $node_{BM}$ (cf. Figure 4).
4. Check that $node_{AS}$ is closer to $node_{BM}$ than the limit in the rule-base.
5. Find the links that are connected to $node_{AS}$ and $node_{BM}$. Perform geometric matching to find the corresponding links in the base-map (BM) for each link in the application-specific (AS) layer (that is connected to $node_{AS}$). If no link is found in the base-map, the link in application-specific layer is left unchanged, and the procedure restarted from point 2 above. This step can handle the situation where one link in the application-specific layer corresponds to several connected links in the base-map. If so, all the corresponding links in the base-map are combined and treated as a single link in the next step.
6. For each pair of corresponding links the following steps are executed. For each break point in both of the links, the distance to the other link is computed. If all of the distances are less than the threshold in the rule-base, the link in the application-specific later is assigned the same geometry as the link in the base-map. If only one part of the links is sufficiently close, only that part of the link is replaced. The other part of the link is untouched (cf. Figure 5). Continue from point 2 above.
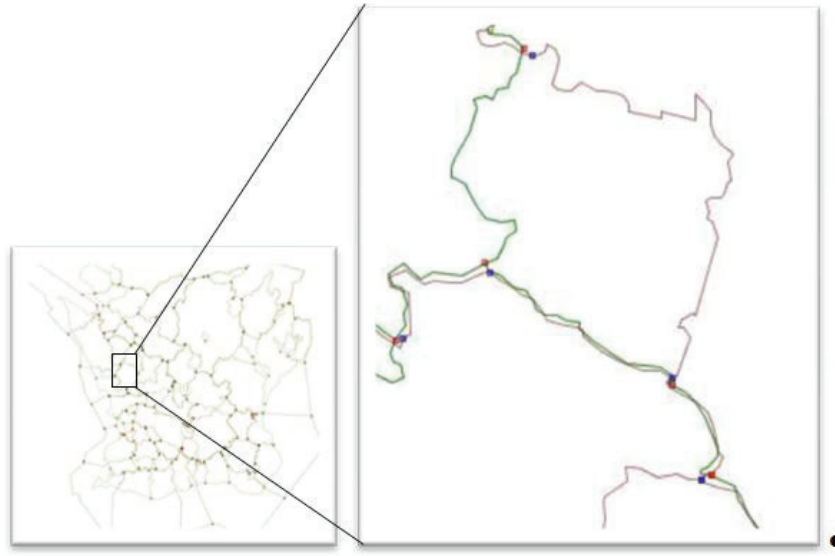7. Continue until all nodes in the application layer are read.

Figure 4: Corresponding nodes for the application-specific layer and the base-map.

# 4 Implementation

## 4.1 Components

The implementation consists of a client, a portal, and basic services with the following properties (Figure 6).

- The client is a WMS client written in Java.
- The cartographic enhanced geoportal consists of six components:

  1. a Java program that operates as a controller of all components in the portal;
  2. a standard installation of MapServer [3] that runs via CGI (common gateway interface) and Java MapScript API. The main responsibility of this component is to register the services and transfer data to the cartographic core;
  3. a cartographic core which contains an implementation of various cartographic methods. For this study the object refinement method is implemented. The implementation is based on open source tools, e.g., OpenJUMP [4] to convert GML-files to the WKT (well-known text) format, and Java topology suite (JTS) [2] for the geometric computations;
  4. a semantic label library containing OWL documents to define the content of the layers as well as the relationships between different layers;
  5. the Java expert system that applies the predefined rules in the OWL document and retrieves required information from the knowledge-base; and
  6. an SLD library consisting of SLD documents that describe the visual presentation of the layers that are registered to the portal.
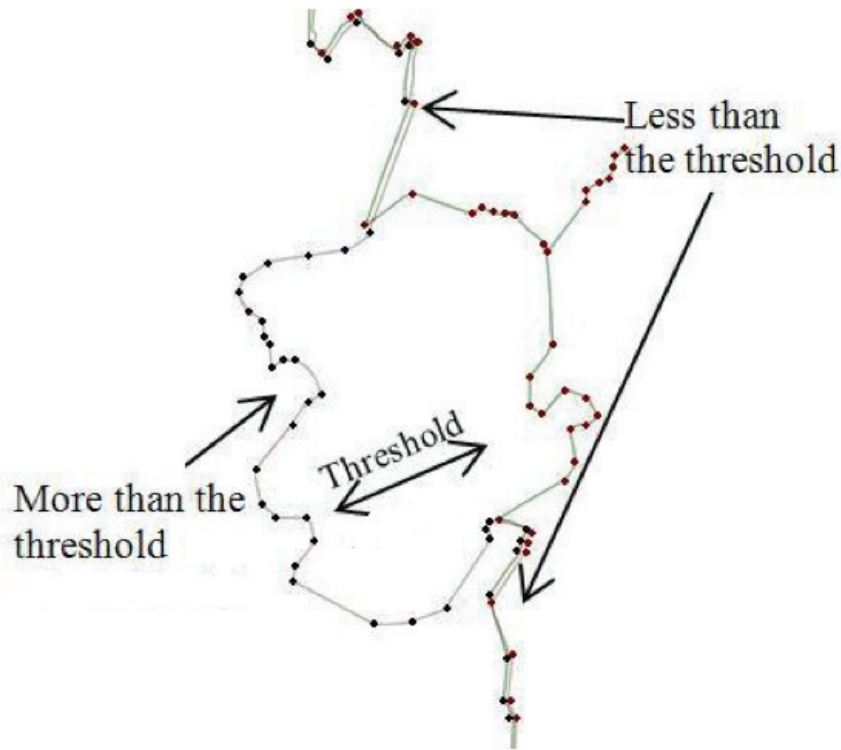
Figure 5: In comparing the thresholds with the node/vertex distances, there are two options: if the distance is less than the defined threshold, then the geometry in the application-specific layer is replaced with the geometry of the base-map; otherwise, the original geometry in the application-specific layer is kept.

- Basic services are normal OGC WFS and WMS services implemented using GeoServer [1] and Mapserver [3]. The geographic data are stored in Post-gresSQL/PostGIS [5, 6].

All basic services have to be registered at the portal. In this study all basic services were registered manually in the MapServer configuration file. For all registered layers, an SLD file was created that determines the visual presentation of that layer. The SLD file is stored in the SLD library.

The content of the layers and the relationships between the layers were defined using Protégé [7] and stored in OWL files (Figure 7). A relationship could be, for example, how an application layer should be integrated to a base-map and the geometries that should be shared. Examples of such relationships are given in Table 1 below.

## 4.2 Workflow

The workflow starts with a WMS *GetCapabilities* request from the client to fetch information about the registered layers in the portal. The request is sent as a CGI command to the MapServer CGI application. The geoportal responds to the request by returning an XML
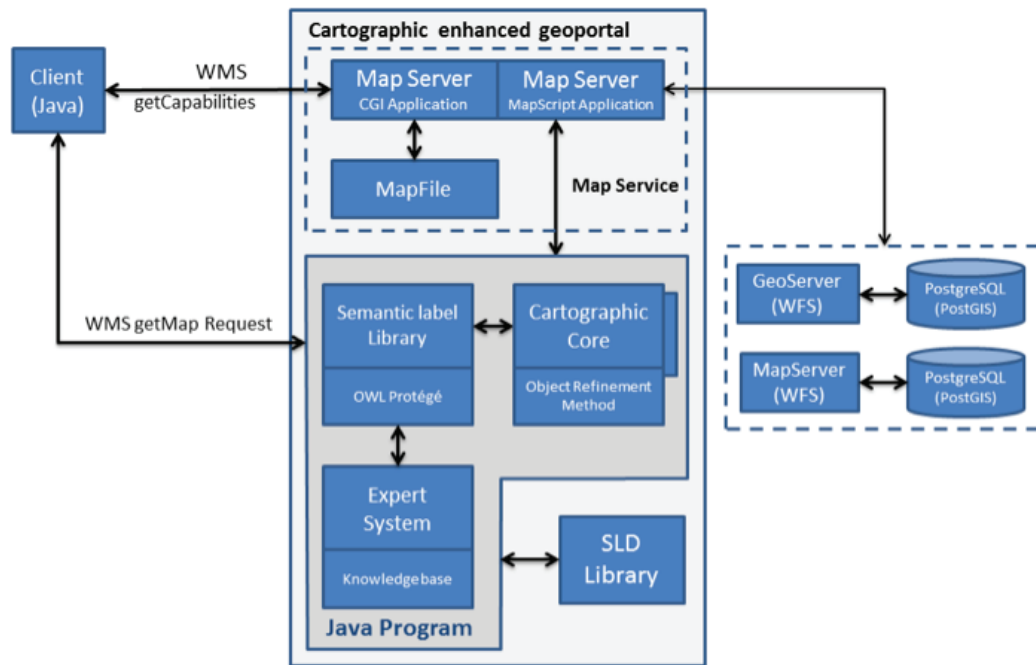
Figure 6: Implementation of the components used in the study.

document that describes the capabilities. Based on this request the system generates the information in the graphical interface (Figure 3). The user then selects layers in this interface. The main difference from an ordinary WMS client is that the user selects which layers that are application-specific layers, and which layers that belong to the base-map. This extension is based on a vendor specific parameter (VSP) [18]. When the user has defined a map request, a *GetMap* request including the vendor specific parameter is sent to the registry via TCP/IP.

The Java program transforms the incoming request to WFS requests, which are sent to the basic services via the MapServer MapScript. The data is returned to the portal as GML files, where they are converted to WKT format.

The Java program now takes control of the integration process. Using the vendor-specific parameters it finds out which layers are application-specific and should be adjusted to the base-map. From the semantic label library the Java program retrieves rules for how the application-specific layers should be integrated with the base-map layers. These rules are then triggered in the expert system. In the execution of these rules, the expert system uses the object refinement methods in the cartographic core.

The output from the integration process is one or several new application-specific layers that are adjusted to the base-map.

In the next step the new application-specific layers are made available for MapServer by the Java program. For convenience we have chosen to retrieve the base-map layers through a new WMS *GetMap* request to the basic services (rather than converting the base-map vector data in the portal). Then MapServer creates the map image according to the
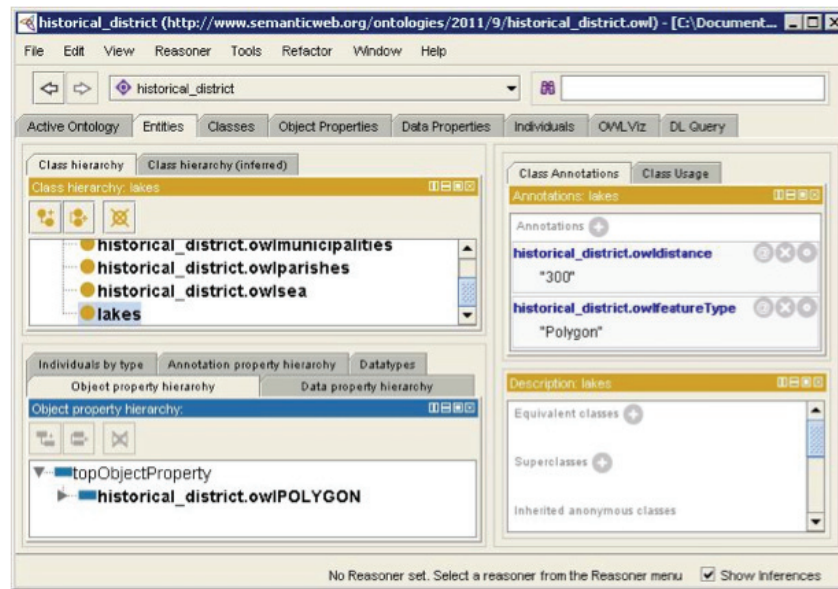
Figure 7: The semantics of each layer are defined in the protégé software.

original *GetMap* request from the client. Finally, the image is sent back to the client via the Java program.

## 5 Case study

The case study in this article is part of a project of integration of demographic data with historic and modern geographic data. The aim of the project is to visualize and analyze the living conditions for inhabitants in a region from the 17th century and onwards. A problem encountered during this project is that historic geographic data do not fit with modern geographic data. For example, old administrative boundaries do not match modern topographic data for two reasons. First, the administrative boundaries may have changed during the last centuries. In this case, the administrative units should of course not be revised. Second, and most commonly, the low geometric quality of historic geographic information causes the mismatch. In these cases, it is important to adjust the historical borders to the modern geometry to enhance both visualization and analysis.

### 5.1 Study area and data

The study area is the Skåne province, Sweden. The used data were:

- historical boundaries, digitized based on historical maps. The data were provided by Stockholm University [15].
- topographic data layers, including sea, lake, municipality, and land use at 1:250 000 scale (Röda kartan by Lantmäteriet).

## 5.2   Method

First, we registered the base-map layers and the application-specific layer in the portal (see Section 4.2). Amongst other, we specified the rules for how the historical borders are related to the base-maps (Table 1). When this pre-processing was finalized, we defined a request to the geoportal as given in Figure 3.

| | |
|---|---|
| 1 | The *historical border* coincides with *sea* shore |
| 2 | If the *historical border* area coincides with a *lake* area more than 50% then replace the *historical border* with the *lake* border. |
| 3 | The *historical border* replaces *lake* within the distance of 500m or less |
| 4 | The *historical border* is replaced by *sea* border if the distance is 500m or less |
| 5 | The *historical border* is replaced by *municipality* border if the distance is 500m or less |
| 6 | The *historical border* has to be adjusted in the order of *sea*, *lake*, and *municipality*. |
| 7 | The *historical border* cannot be on top of a *sea* layer |
| 8 | The *historical border* can be on top of a *lake* |

Table 1: The rules for integrating the historical borders in the base-map.

Based on the rules in Table 1 the expert system solved the integration problem by first moving all borders that were in the sea to lie on the sea shore. Secondly, historical regions that overlap substantially with lakes were replaced by the lake geometry. Finally, the remaining historical borders where adjusted to the sea, lake, and municipality borders with help of the object refinement method.

## 5.3   Results

The output for the whole Skåne province is illustrated in Figure 8. In this figure, the historical borders are properly integrated with the base-map.

Figure 9 shows a smaller zoomed-in portion of the area, highlighting effectiveness of the integration process. Figure 9(a) shows a map in which the original digitalized borders are presented. Figure 9(b) shows the visual output from our viewing service based on the cartographic enhanced geoportal.

The implementation of the portal layer was tested using a desktop PC with an Intel Xeon CPU with 4 Core(s), 2.0 GHz. The execution time was estimated by a Java program. The tests revealed that the total execution time was around 7 to 10 seconds for the data used in this case study (Figure 8).

## 6   Discussion

The proposed method can improve visual output from a viewing service. The cost for achieving this improved visual result is additional preparatory work. The method requires that all layers and their symbology are properly registered in the geoportal (or in an external registry service). All layers that will act as application-specific layers must be semantically labeled and their relationship to the base-map layers must be specified. This was not too time-consuming for the data used in this case study, but for more complex datasets it could be cumbersome.
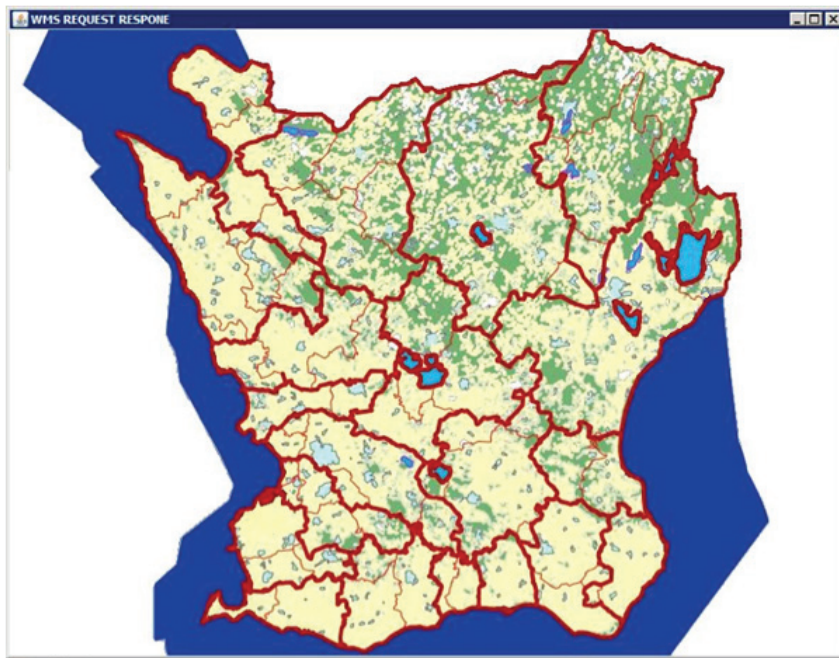
Figure 8: Output from the cartographic enhanced geoportal for the whole study area (province of Skåne). Historical borders are thick dark red lines and municipality borders are thin orange lines.

In this paper, we assumed the scale to be constant, but in reality there are many cases where the scale may change. Moving from one scale to another is an important issue that has to be taken into account. To obtain good maps, the integration needs to be complemented with generalization methods.

The quality of the base-map layer affects the output. In our case study, it was known that the base-map was of high quality. This is not the case in all situations, affecting the approach of only adjusting the application-specific data and leaving the base-map data unchanged.

## 7 Conclusions

In this paper, we developed and implemented integration methods in a cartographic enhanced geoportal with the aim of improving a viewing service. From the user perspective, the enhanced viewing service should work just as a standard viewing service, apart from requiring the user to specify the application-specific layers and the base-map layers respectively. By implementing the system architecture and applying it in a case study, we showed that one can integrate application-specific layers on top of a base-map without any geometrical or topological conflicts. In this way, we obtain clearer map presentation for the overlaying layers.
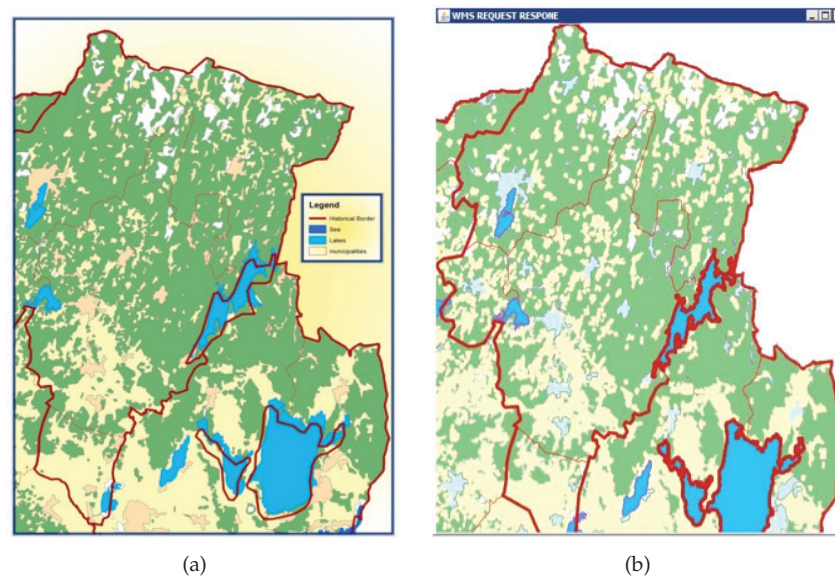
(a)           (b)

Figure 9: (a)Visualization of the original digitalized historical borders. (b) Output from the cartographic enhanced geoportal.

## Acknowledgments

## References

[1] Geoserver. http://www.geoserver.org/display/GEOS/Welcome, 2010. Accessed 20 March 2013.

[2] Java topology suite. http://www.vividsolutions.com/jts/jtshome.htm, 2010. Accessed 20 March 2013.

[3] Mapserver. http://www.mapserver.org/, 2010. Accessed 20 March 2013.

[4] OpenJUMP. http://www.openjump.org, 2010. Accessed 20 March 2013.

[5] PostGIS. http://www.postgis.org/, 2010. Accessed 20 March 2013.

[6] PostgreSQL. http://www.postgresql.org/, 2010. Accessed 20 March 2013.

[7] Protégé. http://protege.stanford.edu/, 2010. Accessed 20 March 2013.

[8] ALLEMANG, D., AND HENDLER, J. *Semantic Web for the Working Ontologist: Effective Modelling in RDFS and OWL.* Morgan Kaufmann, 2011. doi:10.1016/B978-0-12-385965-5.10003-2.

[9] CHIFU, V. R., SALOMIE, I., AND CHIFU, E. Automatic web service composition using OWL-S and fluent calculus. In *International Conferences on Computational Intelligence for Modelling, Control, and Automation* (2008), pp. 271–276. doi:10.1109/CIMCA.2008.118.

[10] CROMPVOETS, J. *National spatial data clearinghouses: Worldwide development and impact.* PhD thesis, Wageningen University, 2006.

[11] FAN, H., AND WANG, Z. Rule-based semantic web services matching strategy. In *Proc. SPIE 8006, MIPPR 2011: Remote Sensing Image Processing* (2011), vol. 800603. doi:10.1117/12.901802.

[12] HARRIE, L., MUSETIÈRE, S., AND STIGMAR, H. Cartographic quality issues of view services in geoportals. *Cartographica 46*, 4 (2011), 92–100. doi:10.3138/carto.46.2.66.

[13] JONES, C. B., KIDNER, D. B., LUO, L. Q., BUNDY, G. L., AND WARE, J. M. Database design for multi-scale spatial information system. *International Journal of Geographical Information Systems 10*, 8 (1996), 901–920.

[14] KLIEN, E. M., AND LUTZ, M. The role of spatial relations in automating the semantic annotation of geodata. In *COSIT 2005*, A. G. Cohn and D. M. Mark, Eds., vol. 3693 of *Lecture Notes in Computer Science*. Springer Verlag, Berlin Heidelberg, 2005, pp. 133–148.

[15] KULTURGEOGRAFISKA INSTITUTIONEN, STOCKHOLM UNIVERSITY. An atlas produced by GeoDataEnheten. http://130.237.186.140/_atlas1900_ny/, 2012. Accessed 20 March 2013.

[16] MUSTIÈRE, S., AND DEVOGELE, D. Matching networks with different levels of detail. *GeoInformatica 12*, 4 (2008), 435–453.

[17] OLTEANU-RAIMOND, A., AND MUSTIÈRE, S. Data matching: A matter of belief. pp. 501–519. doi:10.1007/978-3-540-68566-1_29.

[18] OPEN GEOSPATIAL CONSORTIUM. Web feature service. http://www.opengeospatial.org/standards/wfs, 2005. Accessed 20 March 2013.

[19] OPEN GEOSPATIAL CONSORTIUM. Web map service. http://www.opengeospatial.org/standards/wms, 2013. Accessed 20 March 2013.

[20] POLLOCK, J. T. *Semantic Web for Dummies.* Wiley, 2009. ISBN: 0470498161.

[21] STIGMAR, H. Matching route data and topographic data in a real-time environment. In *Proc. 10th Scandinavian Research Conference on Geographical Information Sciences (ScanGIS 2005)* (2005), H. Hauska and H. Tveite, Eds., pp. 89–107.

[22] SYDENHAM, P. H., AND THORN, R. *Handbook of Measuring System Design, Rule-based Expert Systems.* Wiley & Sons, USA, 2005.

[23] VRETANOS, P. A. Web feature service implementation specification—version: 1.1.0. Tech. rep., Open Geospatial Consortium, 2005. http://www.opengeospatial.org/standards/wfs.

[24] WALTER, V., AND FRITSCH, D. Matching spatial data sets: A statistical approach. *International Journal of Geographical Information Science 13*, 5 (1999), 445–473. doi:10.1080/136588199241157.

[25] YANG, W., LI, X., AND WANG, W. Structure of semantic knowledge database rules-based religious portrait thangka image. In *Networking and Digital Society (ICNDS) 2nd International Conference* (30–31 May 2010), vol. 2, pp. 59–62. doi:10.1109/ICNDS.2010.5479418.

[26] ZHANG, M., AND MENG, L. An iterative road-matching approach for the integration of postal data. *Computers, Environment and Urban Systems 31*, 5 (2007), 597–615. doi:10.1016/j.compenvurbsys.2007.08.008.