Symbology Encoding Repository

Miloš Bogdanović, Danilo Vulović, Leonid Stoimenov
Faculty of Electronic Engineering, University of Niš, Serbia
{milos.bogdanovic, leonid.stoimenov}@elfak.ni.ac.rs, danilo.vulovic@gmail.com

INTRODUCTION

In GIS solutions, the importance of visual identification of geographic data is impossible to ignore (Open Geospatial Consortium, 2007) (Open Geospatial Consortium, 2006). Compared to solutions that visualize raw data, GIS applications have the ability to display geographic data in different ways depending on their thematic attributes. Within the community of GIS application users, the ability to control detailed graphical representation of geographic objects is of great importance. Therefore, it is important for categorized geographic data to be depicted in as legible form as possible. One of the ways to achieve this is to integrate graphic semiotics into vector information. This can be performed by applying styles to both object being visualized on the map and the map itself. One of the ways to define the styling rules that will be used to display geographic objects is the usage of language developed for this purpose (Baer, 2003) (Schnabel and Hurni, 2007) (Kovačević et al., 2009) (Mathiak et al., 2004) (Tennakoon, 2003).

Some of the widely spread languages used for creation of styles are implementations of specifications proposed by Open Geospatial Consortium (OGC). OGC proposed Styled Layer Descriptor (SLD) language (Open Geospatial Consortium, 2007) which was used to develop Symbology Encoding (SE) language (Open Geospatial Consortium, 2006). As the most of styling languages, both styling languages are XML-based languages which can lead to difficulties in users understanding the styling documents. What would be helpful in these situations is the existence of a set of predefined, on-line available, easy-to-understand styles, i.e. styling documents (Maldonado et al., 2006). This set would help users in understanding rules used for development of styling documents. If possible, generated styling documents should be available for further modification and usage. Styling documents set should be coupled with document editor that would follow styling rules and aid development of styling documents. In this paper we will present a solution that meets all previously enumerated demands. It consists of repository and editor of styling documents developed using Symbology Encoding styling language.

SYMBOLOGY ENCODING REPOSITORY SERVICE

Symbology Encoding Repository is implemented as an on-line service. Repository capabilities are defined through ISERService interface shown in Figure 1.

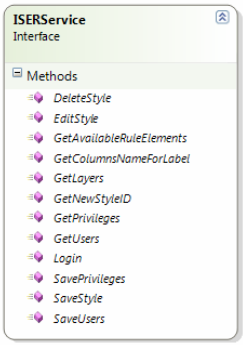


Figure 1: ISERService interface

This service provides users with ability to create, update and delete Symbology Encoding documents, create and delete repository users, and grant and revoke privileges upon owned Symbology Encoding documents.

Symbology Encoding Repository functionalities can be divided into two groups:

1. Symbology Encoding document manipulation functionalities: Create Style, Edit Style, Delete Style, Use Available Rule Elements and Save Style
2. Repository users and privileges manipulation functionalities: Login, View all users, View granted privileges for styles, Edit and save privileges for styles

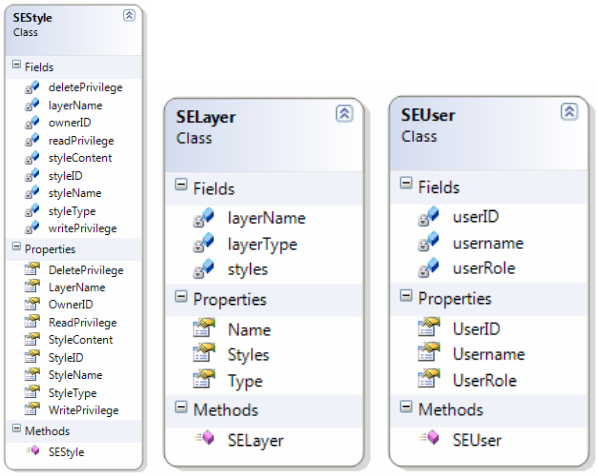


Figure 2: Basic classes of Symbology Encoding repository

SEStyle, SELayer and SEUser, shown in Figure 2, are the basic classes that contain information necessary for repository functioning. SEStyle class captures information about the style stored in the database and is responsible for creating Symbology Encoding document on the file system. SELayer class contains all information for a particular layer. SEUser contains data about repository users.

Symbology Encoding Repository uses Postgre RDBMS, namely PostGIS. Repository database schema is shown in Figure 3 and consists of the following tables: User, Role and Style.

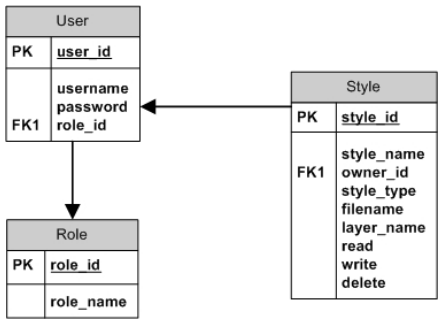


Figure 3: Repository database schema

SYMBOLOGY ENCODING DOCUMENT EDITOR (SEEDITOR)

Both Symbology Encoding repository service and the underlying database are used by Symbology Encoding document Editor (SEEditor). SEEditor allows users to visually create and edit SE documents. SEEditor uses separate controls for each type of geometry of the data layer that new style is created for. SEEditor uses three separate controls for three types of geometry supported in PostGIS RDBMS:

- *SEEditor_LineSymbolizer* control – Used for development of styles for MULTILINESTRING geometry layers.
- *SEEditor_PolygonSymbolizer* control – Used for development of styles for MULTIPOLYGON geometry layers.
- *SEEditor_PointSymbolizer* control – Used for development of styles for MULTIPOINT geometry layers.

Since SE Graphic element may occur multiple times within a single Symbology Encoding document, special control was developed for this element – SEEditor_Graphic. Created Graphic element may have only one Graphic element inside its Fill and Stroke elements. SEEditor enables development of Graphic elements for LegendGraphic, Stroke, Fill and Halo SE elements.

Symbology Encoding Editor has two types of users: Registered user and Administrator. The main page of SEEditor is shown in Figure 4. This page is the login page of the repository. If the user is not registered, it is possible to create account for repository access.

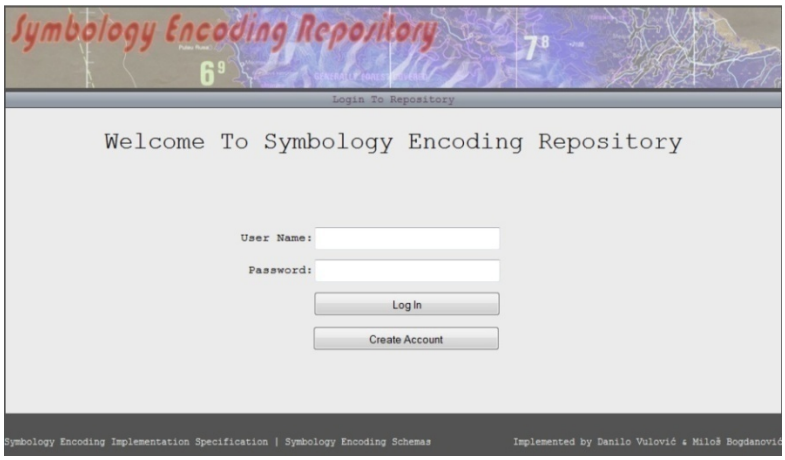


Figure 4: Repository main page

After successful registration, SEEditor main page will be displayed to user, as shown in Figure 5. If the user has administrator role, all available styles from repository will be shown in the style tree on the left – each layer is shown in a tree node. Styles that were created for a particular layer are displayed as a child node of the layer node. Styles created by the currently logged in user are emphasizes by being marked in bold font (Figure 6).

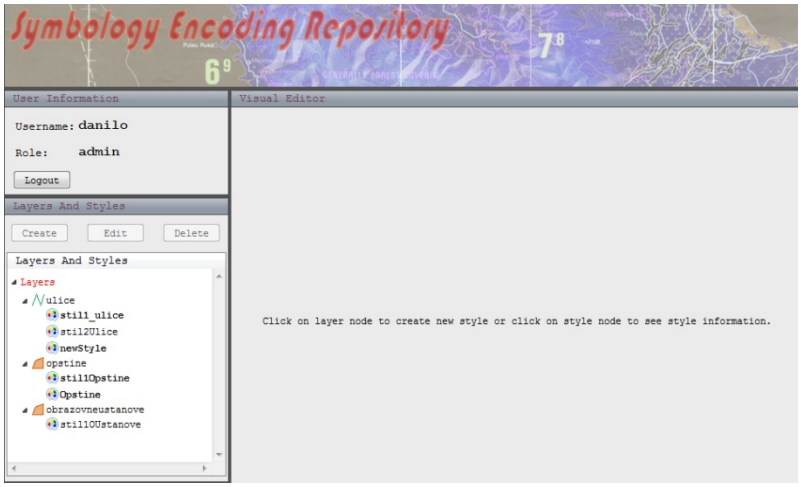
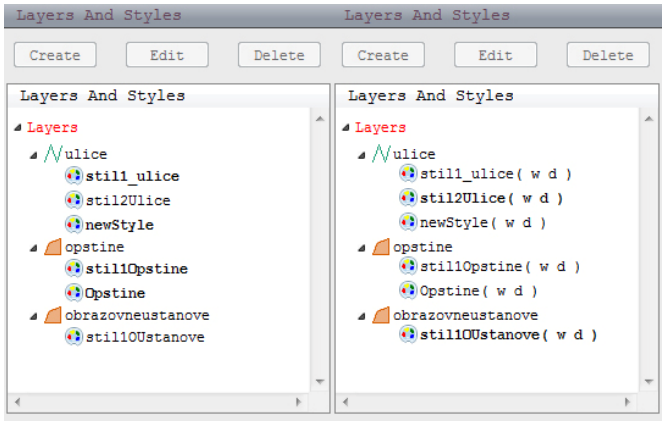


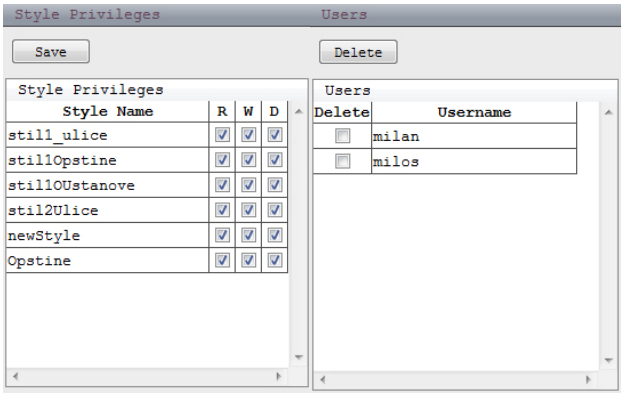
Figure 5: Repository and SEEditor appearance after successful login

If the user does not have administrator role, style tree will display only the styles that are available for reading. Privileges for each of the styles are shown on the right from each style node (Figure 6). Registered user who does not have administrator role can change the privileges only for styles he created (Figure 7). Registered users with no administrator privileges can edit styles that have editing privileges upon.



a) b)

Figure 6: Repository layer tree when a) administrator has logged in b) registered user has logged in



a) b)

Figure 7: Repository privileges for a) administrator b) registered user

Basic SEEditor Repository functionalities are: Create new style, Update style, Save style, Delete style, Export style and Change style privileges.

CONCLUSION AND FUTURE WORK

If Symbology Encoding was chosen as a styling language for creating styling documents, solution presented in this paper could solve a significant part of the problems considering visualization of geographic objects. In the future, users will be provided with the ability to create styles for geographic data that is stored in any relational DBMS that has an extension for the storage of geographic data.

REFERENCES

Baer H.R. (2003) iMap Documentation. Institute of Cartography, ETH Zurich, Zurich, http://www.ika.ethz.ch/statlas/imap/index.htm

Kovačević M., Milosavljević A., Rančić D. (2009) Implementing Complex Polylines for use in GIS, ICEST 2009, 25-27 June, Veliko Tarnovo, Bulgaria

Maldonado A., Bernabé M.A., Manso M.A., Muñoz M.C., Manrique M. (2006) STYLEDCAT: Definition of a SLD Catalogue, 12th EC-GI & GIS Workshop, 21-23 June, Insbruk, Austria

Mathiak B., Kupfer A., Neumann K. (2004) Using XML languages for modeling and Web-visualization of geographic legacy data, GeoInfo, Campos do Jordão. Sociedade Brasileira de Computação

Open Geospatial Consortium, Inc. (2006) OpenGIS Symbology Encoding Implementation Specification, 05-077r4, Version 1.1.0, http://www.opengeospatial.org/standards/symbol

Open Geospatial Consortium, Inc. (2007) OpenGIS Styled Layer Descriptor Profile of the Web Map Service Implementation Specification, 05-078r4, Version 1.1.0

Schnabel O., Hurni L. (2007) Diagram Markup Language - A New Model for Symbolization in Internet Maps, Proceedings of the International Cartographic Conference, Theme 12, Oral 2, Moscow

Tennakoon W.T.M.S.B. (2003) Visualization of GML data using XSLT, Msc Thesis, ITC, Emschede, 60 pp., http://www.itc.nl/library/papers_2003/msc/gim/tennakoon.pdf