УДК 004.9:504:519.6

DEVELOPMENT OF GIS MODULE OF EUROPEAN DECISION SUPPORT SYSTEM FOR NUCLEAR EMERGENCY MANAGEMENT RODOS

Ie. Ievdin, D. Treebushny, M.I. Zheleznyak, N. Shlyakhtun
Institute of Mathematical Machine and System Problems NAS of Ukraine
e-mail: yewgen@env.com.ua

INTRODUCTION

RODOS system for nuclear emergency management in Europe has been developed under the funding of European Commission since 1992 [1]. RODOS has been designed as a complex system which includes models and data bases used for modeling, evaluation of possible nuclear emergency consequences and for the corresponding emergency planning. In 2006 the work on the RODOS system redesign using modern Java and GIS technologies was started [2-3].

Original RODOS system has its own implementation of geographic information system using internal specific data types, complicated converter raster and vector data from files in standard formats. Redesigned system – JRodos – is developed on the base of user requirements and use cases suggested by RODOS User Group [4]. According to the document JRodos should support direct import of raster or vector map data in common formats, aggregation of results, easy configuration of layers. Also, the system should suggest an easy adaptation of GIS data to the national conditions and provide access to the calculation results from external GIS applications.

GIS IMPLEMENTATION DETAILS

GIS functionality of the redesigned RODOS is based on the GeoTools library [5]. GeoTools is an open source Java code library which provides standards compliant methods for the manipulation of geospatial data. The GeoTools library implements Open Geospatial Consortium (OGC) specifications as they are developed. GeoTools code is released under the GNU Lesser General Public License (LGPL) which allows developers to adapt the library to the special needs of the decision support system, increase performance, eliminate memory leaks and correct errors.

JRodos is developed as a distributed system with Computational Engine for performing calculations, Management Server for database access, providing data exchange between system components, collecting information from other parts of the system and Client for capturing input information, visualizing results and messaging implementation. GIS module is dived into two parts, according to the correspondent functionality – Client and Computational Engine parts.

The Client part is responsible for user-friendly data presenting, facilitating decision-making process in emergency situations or analysis of model results for research purposes. Client GIS module is intended for visualization of geospatial calculated results and thematic data, including:

- Digital terrain data as background maps (elevation),
- Land use, soil, population, borders of administrative regions, urban areas and roads,
- Maps of objects of interest for the user, such as nuclear power plant, detectors, weather stations, etc.,
- Maps of meteorological and radiological situation, the results of measurements and numerical weather forecasting.

The following data sources can be used in JRodos:

- Files in ESRI .shp or GeoTIFF formats, which are loaded completely in the RAM.
- Results of spatial query to the GIS database PostGIS / PostgreSQL. In general, these are the data, which size does not allow loading them entirely into RAM.

- The results of the query to WMS, Google Maps servers, which is sent when a user changes or scales the visible map region.
- The model calculation results, measurement or numerical weather prediction data. In this case the layers are formed in memory-based objects that contain numerical data, as well as meta information about the result (name, source, substance). Time dependent data can be animated. During the animation only attribute values of the layer (values in grid cells) or the orientation of spatial objects (e.g., arrows in the measurement of wind direction) are changed without any recreation of the layer, which significantly improves the system performance.

GIS module also has the minimum necessary set of tools for working with layers. Among them is zoom in/out tool, pan tool to scroll the visible region, projection changing tool, the tool for measuring of the distance on the projection, selecting objects and visualizing attribute information of the selected feature, printing the content of the map, creating counter plot map. For time-dependent results it is possible to create the plot (values vs. time) for a particular point, export to vector format. The layer list permits the user to manipulate and customize the map: adding new layers, moving, changing a hiding layers, import layer data and export vector data to external file. Each layer in the GIS module of RODOS system is the object of a certain class type that defines a set of operations applicable to the layer. Built-in style editor allows a user to edit styles for vector layers, export and import them into .sld files.

Integrated models also can provide additional GIS tools, which are needed for capturing input parameters: creating the features (e.g. computational grid), selecting the area (e.g. different countermeasure zones), assigning attribute and value for each feature (e.g. location factor for detector point).

The map control window of client-side GIS module is presented in Figure 1.

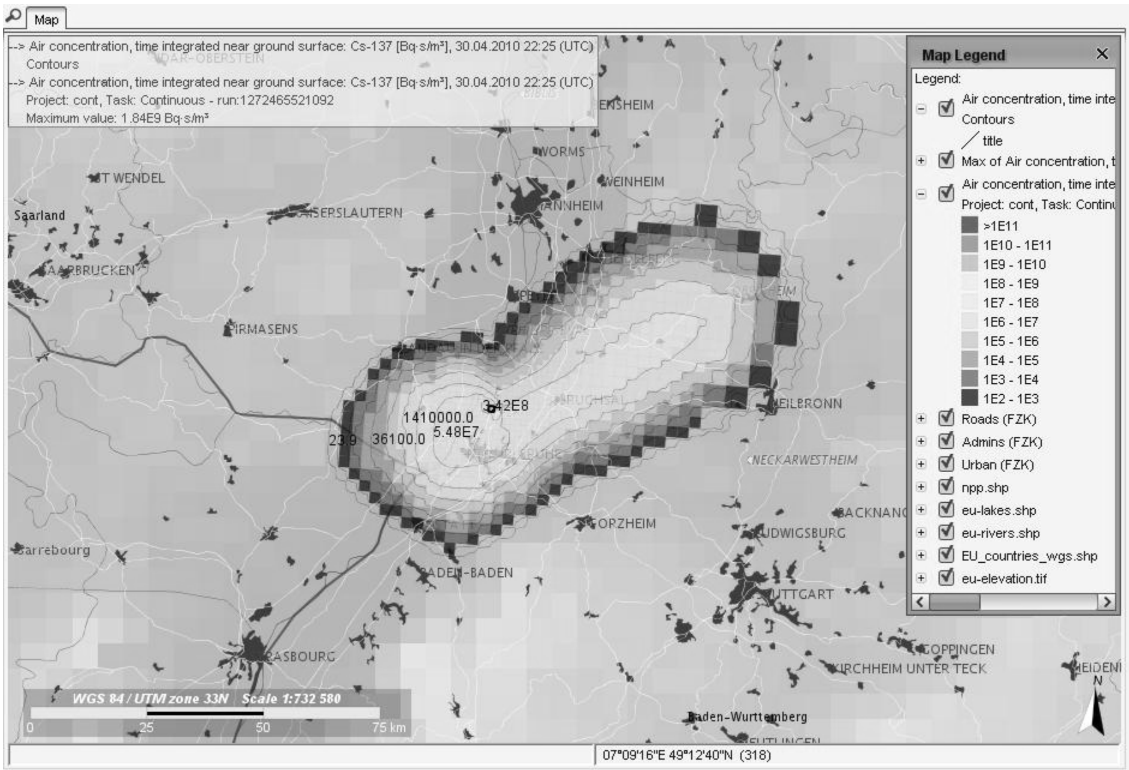

Figure 1. Client-side GIS module with different layers, map legend, concentration map, counter plots built

Computational engine part of GIS module processes the geospatial data to be used as input of mathematical models. Computational models naturally require geographic data, such as elevation maps, land use, soil, population density, agricultural production data, radiological regions etc. The original maps are supplied as raster (altitude, land use, soil, population density) or vector data (production by region).

To convert geo-referenced data into model inputs an interpolation of maps on a computational grid is required. The original RODOS system uses simple interpolation method – requesting data in the centre of computational cells. This approach obviously has drawbacks: for example, if the centre of a large computational cell reaches the mountain peak, the value of the average height of the entire cell is taken equal to the height of the mountain, which is incorrect.

The redesigned system held a correct interpolation, which "cuts" the grid source data (raster or polygon) on the computational grid. The value of computational cell is calculated as a weighted average proportional to the area of each “slice” received from the original maps in the selected cell. In the classification (e.g. land-use data), the cell value is given to the class with have the maximum area in the cell.

Several major approaches are used to accelerate the software implementation of interpolation procedures.

- All raster maps are stored as uncompressed geo-referenced tiff images. This allows the system to quickly "cut" a small sub-region which contains computational grid without the need for the reading the entire raster into the program memory. This approach especially proves its effectiveness when working with detailed maps where the file size is hundreds of megabytes.
- To provide rapid clipping of the raster on the computational cells and estimating the area of pixel slice in the cell the system uses a Sutherland-Hodgman algorithm [6]. Experience has shown that the polygon-polygon intersection algorithm from the standard JTS library [7] can be very time consuming, up to 3-5 minutes to carry out interpolation comparing to about 1 minute of model calculation itself. The usage of the Sutherland-Hodgman algorithm reduced the interpolation time down to 10-20 seconds.

The idea of Sutherland-Hodgman algorithm is shown in Figure 2. It works by extending each line of the convex clip polygon (which is raster pixel in our case) in turn and selecting only vertices from the subject polygon (computational grid cell) that is on the visible side.

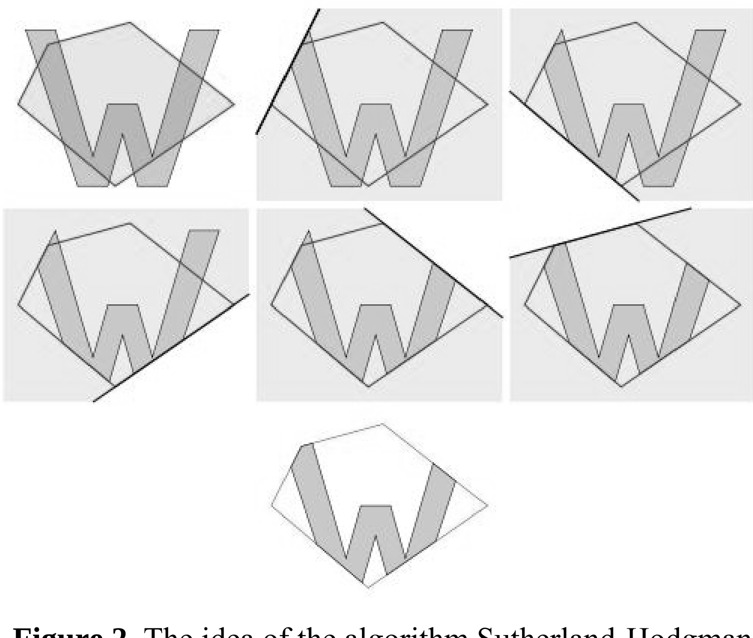

Figure 2. The idea of the algorithm Sutherland-Hodgman

- The result of the interpolation are cached with input index as a combination of MD5 code of input data file and the MD5 code of computational grid polygons, and the value – calculated data set. So the interpolation should be done only once for most of the system use cases.

JRodos is deployed with coarse European data. To satisfy user requirement of easier adaptation the system to national conditions JRodos allows a user to load its own map with finer resolution. Regions not covered by user defined maps take the default coarse data during the interpolation process.

To provide access to the model calculation results from the external GIS applications, the maps can be exported in ESRI .shp format, as well as stored in geospatial database driven by PostgreSQL with spatial extension PostGIS installed. This database contains a table with cells of the computational grid patterns of arbitrary shape. In addition the geo database contains the data necessary to identify the result: information about the project and model, which generate the result, the name, units, timestamp of the result. A user then must create a query to the database to obtain the spatial field corresponding to the desired output parameter with the particular timestamp.

CONCLUSIONS

Redesigned decision support system RODOS implements GIS module as an adaptation of the open source GeoTools library. GIS module allows the user to visualize the geospatial and temporal information, perform basic manipulation with it, importing and exporting data in standard formats, which greatly simplifies the task of making decisions and analyzing the calculated results. The implementation satisfies all requirements made by users before starting the development process.

A new method of GIS data interpolation on the computational grid improves the accuracy of calculation and gets rid of artefacts. The approaches used in the RODOS efficiently accelerate data interpolation and significantly reduced the computational time.

REFERENCES

1. Raskob W. European approach to nuclear and radiological emergency management and rehabilitation strategies (EURANOS). Kerntechnik 72 (4) (2007) 172-175.

2. Ievdin Ie, Treebushny D., Kovalets I., Guziy O., Zheleznyak M.I. Development multi-platform version of decision support system for nuclear emergency management in Europe (RODOS) on base of modern Java and GIS technologies, Зб. праць Четвертої конференції з міжнародною участю “Системи підтримки прийняття рішень: теорія і практика. СППР-2008”, червень 2008 р., Київ, С. 121-124.

3. Ievdin Ie, Treebushny D., Zheleznyak M.I. Distributed architecture of RODOS – Decision support system for nuclear emergency management in Europe, Зб. доповідей науково-практичної конференції з міжнародною участю “Системи підтримки прийняття рішень: теорія і практика. СППР-2009”, червень 2009 р., Київ, С. 181-184.

4. F. Gering, B. Gerich, T. Duranova, L. Bohun, M. Ammann, G. Fritsch, P. Hofer, C. Twenhöfe, M. de Ridder. User requirements for the re-design of RODOS in phase 2 of the EURANOS-project, EURANOS(CAT2)-TN(06)-09.

5. GeoTools. The Open Source Java GIS Toolkit – http://www.geotools.org/

6. Foley J. D., Dam A. van, Feiner S., Hughes J. Computer Graphics: Principles and Practice in C (2nd Edition) // Addison-Wesley Professional. – 1995. – Section 3.14.1. – P. 124–127.

7. JTS Java Topology Suite – http://www.vividsolutions.com/jts/jtshome.htm.