Jakub OTRZĄSEK, Wojciech MUELLER, Krzysztof KOSZELA
Uniwersytet Przyrodniczy w Poznaniu, Instytut Inżynierii Biosystemów
ul. Wojska Polskiego 50, 60-627 Poznań
e-mail: koszela@up.poznan.pl

SQL SERVER GEOGRAPHY AND GEOMETRY TYPES IN SUPPORT OF AGRICULTURAL ANALYSIS

Summary

In general, geospatial analysis, so essential in agriculture, is conducted using GIS systems. Considering the current state of information technology, alternative methods may and do involve dedicated applications that use geography and geometry data types, available on the SQL Server 2008 level. Their creation would not be possible without the types being implemented in Visual Studio 2008 and 2010 development environment as well. Using these technologies, the authors have created an information system that supports agricultural insurance market analysis based on vegetal production potential of farms. This analysis primarily uses geographical data.
***Key words**: computer systems, geospatial analysis, databases*

TYPY GEOGRAFICZNE I GEOMETRYCZNE SQL SERVER W PROCESIE WSPOMAGANIA ANALIZ DOKONYWANYCH W ROLNICTWIE

Streszczenie

Generalnie analizy geoprzestrzenne, tak potrzebne w rolnictwie, dokonywane są przy użyciu systemów informatycznych typu GIS. Alternatywne sposoby ich przeprowadzenia, z perspektywy aktualnie dostępnych technologii informatycznych, są i mogą być wykonywane przez dedykowane aplikacje, wykorzystujące typy geograficzne i geometryczne oferowane na poziomie SQL Server 2008. Ich powstanie nie byłoby możliwe bez implementacji tych typów również w środowisku programistycznym Visual Studio 2008 i 2010. Posługując się tymi technologiami, autorzy wytworzyli system informatyczny, wspomagający analizę rynku ubezpieczeń rolnych w oparciu o potencjał produkcji roślinnej gospodarstw rolnych. Analiza ta w głównej mierze bazuje na danych typu geograficznego.
***Słowa kluczowe**: systemy informatyczne, analizy geoprzestrzenne, bazy danych*

1. Introduction

Currently, when speaking of geospatial data management systems, commonly used in agriculture, we mean the Geography Information Systems (GIS). They are usually programs based on client-server architecture (window applications) that have wide functionality. These technologies involve certain costs (licensing) and require workstations of adequate processing power, but most of all they are not a very convenient solution for the mobile manager or for the owner of a scattered holdings.

Other solutions exist for managing geospatial data, namely relational database management systems (RDBMS). At present, the leading RDBMSs support geometry and geography type data. Their implementations in free systems (MySQL, PostgreSQL) do not always follow standards (WKT - *Well Known Text*, WKB - *Well Known Binary*, GML - *Geography Markup Language*). Microsoft database products from version 2008 onwards (2008, 2008R2, 2012) have integrated support of said data types. The situation is similar in the case of Oracle, though this upgrade may involve additional costs [3].

In this article, the authors focus primarily on solutions provided by Microsoft that include SQL Server 2008 (R2) together with geospatial data type libraries ready to use on the .Net platform. It is worth noticing that geospatial data types are also available in the free versions of SQL Server Express Edition.

2. Geospatial data types in the database environment

At present, virtually every production or organization entity makes use of one or more applications that work with databases. Thanks to the geolocation services that are currently commonly available, most addresses contained in DBMSs can be easily associated to relevant geographical coordinates. Furthermore, it is possible to purchase appropriate address bases that contain geographical coordinates of individual locations. There currently exist numerous commercial solutions that provide a variety of spatial data (soli richness, floodplains, purchasing capacity of residents, etc.).

Availability of such data allows for their integration with production data, which in turn enables more sophisticated analysis to be conducted, increasing the accuracy of decisions based on them. A spread of the aforementioned analytical methods depends of course on possession of the spatial data itself, availability of information technologies that enable the integration and processing, and on the enterprise costs.

In the case of SQL Server, and similarly other DBMSs, in order to ensure the capability to store spatial data one must create an entity with the right column type:

```
create table new_table (
id int not null primary key identity(1,1),
description varchar(255),
geo geography).
```

The first two columns store traditional (atomic) data[1], while the last one contains geography type data. Presently, geospatial data is managed by two subtypes: geographic and geometric. Each one is a complex (non-atomic) type that provides a number of methods and properties. The geographic type, tightly bound with geographic coordinates, was used in the showcased system. The aforementioned data types allow for storage of information about a point, a group of points, a line, a polyline, a polygon and a multipolygon (a polygon containing other polygons). By saving information of this kind we create an appropriate permanent object, using the T-SQL language together with proper methods available for the given type of object. This is illustrated by the following example:

```
Insert into new_table (description, geo) values (‘some
description', geography::STGeomFromText('Point(15
50)',4326));
```

The entry: geography::STGeomFromText('Point(15 50)',4326) requires explanation. It uses one of the static methods that create a specific geographic object based on information contained in the argument [4]. A characteristic feature of this method is that it expects an argument in the shape of text data. Consequently, the first argument of this function is the aforementioned string, containing both information about the created spatial data (e.g. a point or polygon) and the geographic coordinates that characterize the object being formed. The second numerical argument contains information about the assumed Spatial Reference Identifier (SRID)[5].

Acquisition of spatial data is done via the traditional SQL queries. Performing such queries on the SQL Server Management Studio level provides us with a data set both in table and graphical form, created basing on geographic data.

3. Spatial types in the programming environment

Complex harnessing of spatial data gathered in MSSQL is impossible without information systems aimed at users with varied requirements. Of course we can use the commonly available maps (google maps, bing maps), but then we are limited to satellite and road maps (or hybrid maps; satellite images overlaid with e.g. a road grid) that cannot be associated with our data without proper tools.

Using SQL Server Management Studio as a client program to display spatial data is not a fully satisfying and justified solution either. One limitation of the program is the number of objects that can be simultaneously projected in graphic form (5000 objects). It doesn’t appear as a significant drawback of the SSMS, since the interface of this application, as well as the application itself, is generally designed for advanced users creating applications that work with MSSQL, or for database administrators.

Information technologies of Microsoft are usually of complex nature, and consequently software developers have access to libraries that allow for the use of complex data types, among them spatial data. In order to be able to process this type of data, one should first add references from the .NET framework to the right library, located in one of the folders created by SQL Server. Linking a specific namespace will be of unquestionable help as well during the development of said applications. Access to data embedded in MSSQL, regardless from their complexity, can be performed by using the ADO.NET technology.

In the case of data retrieved from a database, containing spatial information of course and saved in a DataTable object, access to such data from an application can be performed as follows, using the C# language:

```
SqlGeography ng;
if (dr[spatialcolname].GetType().ToString() ==
"Microsoft.SqlServer.Types.SqlGeography")
  {
  ng = (SqlGeography)dr[spatialcolname];
       ...
}
```

In this example, spatialcolname is a string variable containing the name of the column with spatial data. SqlGeography object in .NET framework enables the same methods and properties as the Geography type in SQL Server[4]. The data retrieved can be easily integrated with Google Maps or Virtual Earth (Bing Maps)[4][5].

If we want to overlay our own (raster) maps with geographic or geometric data, putting together a proper mechanism is not trivial. The greatest difficulty laid in cartographic representation, that is in projecting the curvature of Earth to a two-dimensional surface. To transfer spatial (geographic) data, recorded in RDBMS, to a raster map, an appropriate type of representation had to be used. It had to be compatible with the representation used to create the said raster map. Another essential step was to correlate points on the map with the points retrieved from the database.

4. Software development

Due to the low complexity of the project the traditional approach, stemming from software engineering and involving a multiaspect modelling of problem field, was abandoned. Nevertheless, a portion of UML diagrams was put together, including action diagrams presented on figs. 1 and 2 [2].

One of the more difficult problems to solve that arose during the development of the discussed application was visualizing the spatial data, retrieved from DBMS as a result of a performed query, against raster maps that constituted the reference level. For this particular purpose, the authors created a library that retrieves geographic and geometric objects from data acquired from DBMS and performs an appropriate conversion, then embeds them in a list created for this purpose. Said data, retrieved from the database, is first handed to a DataTable object that in turn shares it with a Jspatial object. Consequently, it is another important layer between the database and the user interface. A method was implemented in the proposed Jspatial class that takes the name of the column containing spatial data as an argument. Said method iterates results, identifying the type of received data and simultaneously sorting geographic data from the geometric. Later, a geographic or geometric object’s complexity is recognized and the identified points, being the smallest structural elements, are efficiently embedded in a special, multilevel list. Such a way of recording allows for the correct representation of the structure of spatial objects. Visualizations of data contained in said list requires performing of several additional steps. The first is a conversion of point coordinates to the accepted cartographic

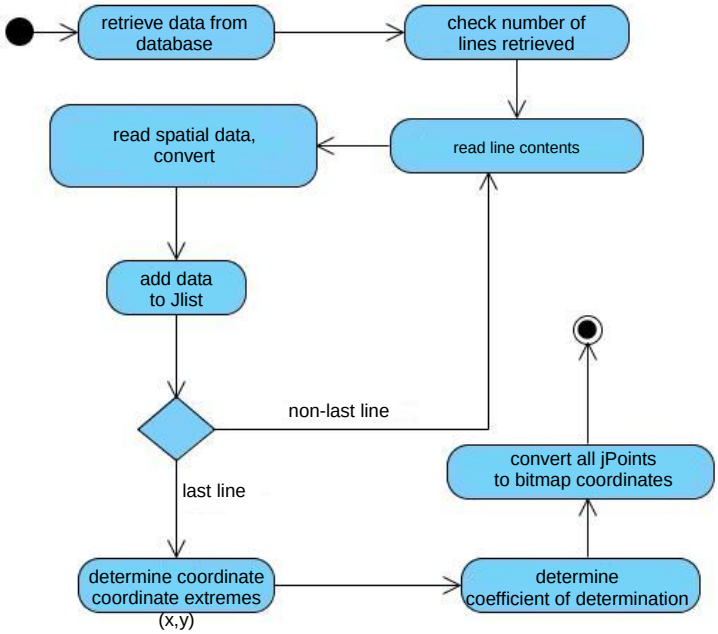

Fig. 1. Action diagram - a general algorithm of creating the spatial data presentation layer

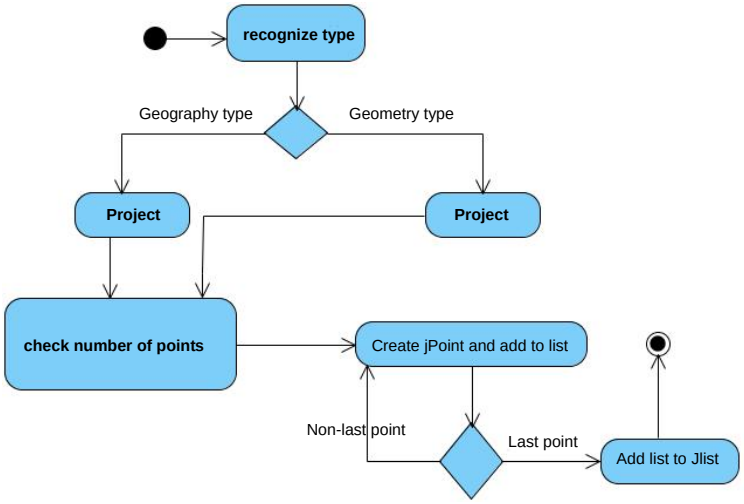

Fig. 2. Action diagram - creating a Jlist that contains JPoints

grid, which should be consistent with the cartographic grid of the raster map that forms the background/reference for the spatial data presentation. For the time being, the proposed class implements a method involving the Mercator projection. This projection was applied in the construction of the aforementioned raster map used by the authors for the background. Another necessary action was to match the scale of objects and the raster map. This operation was fairly simple and basically boiled down to determining a coefficient that was a multiplier for geographic coordinates. The basis for its determination was the proportion between the spacing of points in DBMS and their equivalents on the raster map.

The .NET framework has numerous libraries, including classes equipped with methods that allow for easy 'drawing' on bitmaps. However, their utilization had to be preceded by implementing and using two different mechanisms responsible for, respectively, scaling data to correspond with the scale of the raster map used, and for coordinating it with the background. While the mechanism mentioned above does not provide perfect precision, it does seem accurate enough for presentation and for further analysis purposes.

From the visualization point of view, the final objects, connected with the converted data, that have a graphical form, are jPoint objects, created based on the designed class that inherits from the Point class.

5. Results discussion

The application proposed by the authors, that allows for a visualization of geographic and geometric data embedded in MS SQL Server 2008 and associating it with a raster map, has been tested on data regarding vegetal production insurance. Associating insurance policies purchased by farmers protecting their vegetal production from fortuitous events seems an interesting idea both from insurance companies and state administration point of view. In the second case, thanks to a similar analysis based on a larger data pool, one can evaluate the efficiency of national policies that support

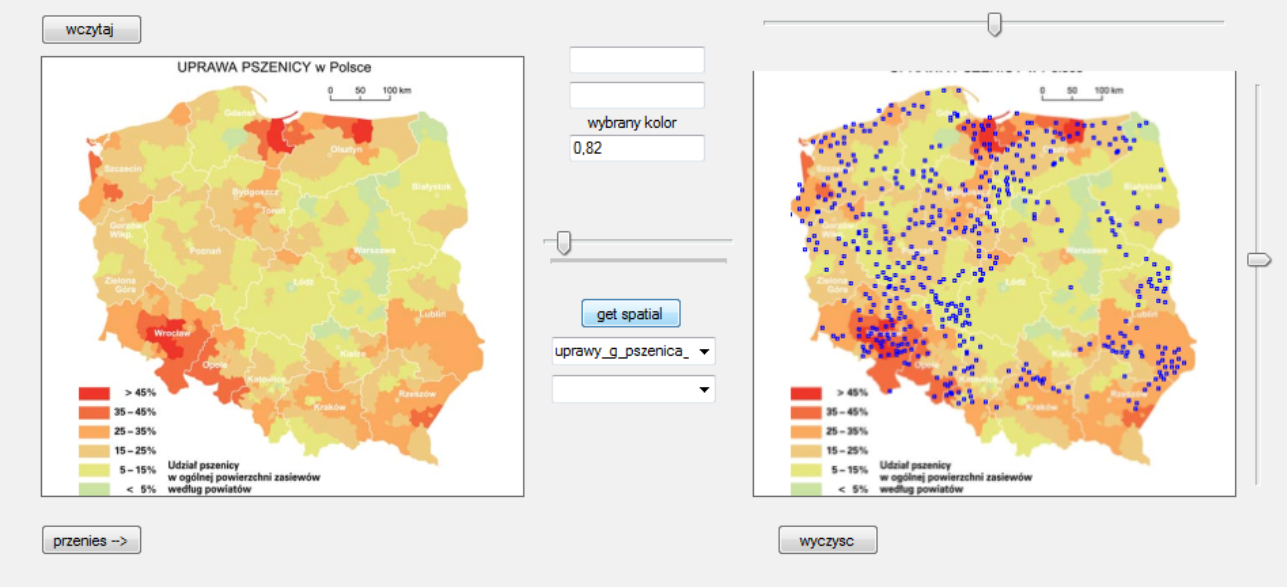

Fig. 3. Inspection of spring wheat insurance sales potential.

agricultural production profitability. Inter alia, a system of state-subsidized crop insurance has been implemented in Poland since 2006 [6]. The basis for farmers to apply for state aid in case of natural disasters is holding a policy that covers at least half of the farming area and at least two possible risk factors (e.g. flood and drought). A lack of insurance policy is grounds for denying aid by the state administration, unless the victim can produce at least two insurance denial documents from two different insurance companies [6].

All parties are interested in increasing the number of such insurance policies. The authors have only used the created information system to determine the sales potential for insurance policies related to spring wheat cultivation in Poland. Of course, it is a matter of interest for insurance companies. The choice was not accidental, but a consequence of data availability.

Key program features are illustrated by figure 3 below. It shows the main form of the information system. The left side is taken up by the reference map for the presentation of spatial data retrieved from RDBMS. Associated visualization effects are displayed in the right part of the window. Graphical display elements are surrounded by action buttons, dropdown lists and sliders allowing to synchronize data recorded in two different formats. The buttons and lists in turn allow us to decide what map is to form reference for the presentation and what spatial data from RDBMS will be visualized and analysed.

In this case we can view a map displaying what percentage of crops the spring wheat takes in individual parts of Poland, overlaid with insured locations.

The fact that the authors used raster maps for background, instead of vector maps that cause the programming problems mentioned earlier and prevent SQL Server 2008R2 from being more efficiently used, is not accidental and primarily stems from financial issues. Vector maps exist already, but due to high, poorly calculated purchase prices and inefficient organization of public administration bodies, their availability is severely limited.

6. Conclusion

1. The sharing of spatial data in DBMS and in the Visual Studio development environment allows for a convenient enough development of applications that support agriculture, where this type of information is of crucial importance. It may be done without access to GISs, which require specialist competence. Furthermore, in contrast to GIS, the presented technology is inexpensive to implement, since all the necessary tools are free.
2. It is worth to stress that the presented information technologies create further processing possibilities of said data, using analytic databases that significantly improve the managing process of not only the production entities. This requires a much larger pool of data, but considering the longer timescale of information systems usage, this does not seem to be an issue.
3. Utilizing RDBMS spatial data allows both for macro (an entire country, a whole continent or even the entire Earth) and micro (centimetre by centimetre) analysis. The analysis scale is only dependant on the data available.

7. Bibliography

[1] Beynon-Davies P.: Systemy baz danych. Warszawa: WNT, 2003.
[2] Booch G., Rumbaugh J., Jacobson I.: UML przewodnik użytkownika. Warszawa: WNT, 2002.
[3] Otrząsek J., Mueller W., Koszela K.: Methodology of comparing the performance of SQL Insert operations in selected RDBMS. Journal of Research and Applications in Agricultural Engineering, 2012, vol. 57(2).
[4] Alastair Aitchison – Beginning Spatial With SQL Server 2008, 2009.
[5] http://msdn.microsoft.com/en-us/library/bb933876%28v=sql.105%29.aspx - dostęp 01-02-2013
[6] http://isap.sejm.gov.pl/DetailsServlet?id=WDU20051501249 - dostęp 01-02-2013.