Proceedings of the Federated Conference on
Computer Science and Information Systems pp. 291–296

ISBN 978-83-60810-22-4

# Geospatial presentation of purchase transactions data

Maciej Grzenda, Krzysztof Kaczmarski, Mateusz Kobos, and Marcin Luckner
Faculty of Mathematics and Information Science
Warsaw University of Technology
Warsaw, Poland
Email: {m.grzenda, k.kaczmarski, m.kobos, m.luckner}@mini.pw.edu.pl

***Abstract*—This paper presents a simple automatic system for small and middle Internet companies selling goods. The system combines temporal sales data with its geographical location and presents the resulting information on a map. Such an approach to data presentation should facilitate understanding of sales structure. This insight might be helpful in generating ideas on improving sales strategy; consequently improving revenues of the company. The system is flexible and generic – it can be adjusted to process and present the data within different levels of administrative division areas, using different hierarchies of sold goods. While describing the system, we also present its prototype that visualizes the data in an interactive way on a three-dimensional map.**

## I. Introduction

COMPANIES selling goods have many possible ways to devise strategies of improving their business model and adjusting it to changing business conditions. One of them is to use business intelligence tools to automatically gather, process, analyze and visualize data that is important for the company in hope of obtaining useful insights that can be used to improve company's functioning. One of the most promising and simple approaches to this problem is to combine company's private data with publicly available data in order to obtain a useful synthesis of these two. An independent problem is how to handle and integrate different dimensions of company's data. One of the dimensions is the temporal one: the business conditions change over time and the company's decision-makers have to be able to follow changing trends in order to e.g. predict future behavior of the market. Another important dimension is the spatial one: different administrative regions have different business environments, and different business strategies might be more or less suitable for different sales areas (e.g. some regions might need more billboard advertisements while others might need more on-line advertisements).

In this paper, we describe an idea for an automatic system that combines private and publicly-available data of spatial and temporal type and visualizes it on a map. The main goal of the system is to present sales data of a company in an useful and interactive way. The system is simple but generic – it can be adjusted to process and present data within different levels of administrative division areas, using different hierarchies of goods sold by the company. Apart from describing the general idea, we also present a prototype of such a system that uses data from one of the Polish companies. The company is one of the largest Internet sellers of tires in Poland.

An overall process of data acquisition and transformation in the system is presented in Fig. 1. Our system automatically combines purchase transaction records data with information about spatial placement of administrative division areas of region of interest to locate an approximate place where each purchase was delivered. To be more precise, we use the information about delivery town and zip code of a purchase to determine which administrative division area the buyer is situated in. Each area has GPS coordinates assigned to, so the data related to this area can be easily placed on a map. The data is saved in a form of a relational database. Next, a geographic data visualization application is used to present the data in an interactive and user-friendly way. Since there are many mature applications which can be used as a visualization engine, we decided not to implement our own in the prototype. Public and free tools, although not perfect, are mature enough to be used in a professional solution. Their displaying capabilities are not limited to any particular area and can usually show different geographical regions all over the Earth. One of the most popular tools of this type, i.e. spatial data viewer equipped with ability to load user data, is Google Earth [1]. It proved to meet our requirements and we used it in our prototype (the prototype allows also sharing the visualization on-line via Google Maps).

### A. Related Research

Problem of storing and presenting spatiotemporal data is generally addressed by dedicated systems: SOLAP (Spatial On-Line Analytical Processing) being *a visual platform built especially to support rapid and easy spatiotemporal analysis and exploration of data following a multidimensional approach comprised of aggregation levels available in cartographic displays as well as in tabular and diagram displays* [2]. As all OLAP-based solutions, they require wide knowledge of data processing and data mining, in this case often combined with expertise in cartography. Another drawbacks of these systems are high licence fee and maintenance costs and therefore low return on investment values which are not acceptable for small companies.

Our lightweight data processing components try to answer spatiotemporal data analysis demands in much simple and cheaper way.

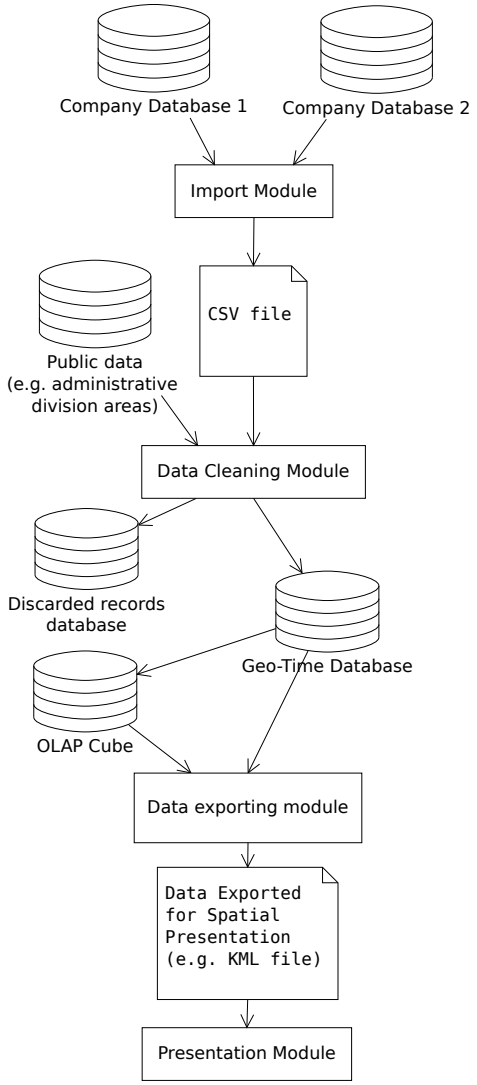


Fig. 1. Data processing phases

An important research was done to allow for fast and precise spatiotemporal aggregation calculation. This task is not easy due to imprecise querying and spatial selection criteria. Especially R-Tree structure [3] with many improvements is used to store spatial information [4]. Adding the time dimension was analyzed in many works i.e.: [5], [6], [7], [8]. Another path in research is devoted to streaming data systems and calculation of incremental spatial aggregates [9].

In our research, we focus mostly on Internet transaction data cleaning and coupling it with spatial information leaving an effective data storage and querying methods as open topics. Usage of R-Trees is still possible and could improve performance for large datasets. However, currently we do not consider this to be an important problem since our data comes from small companies and does not exceed volumes that can be effectively processed by a simple relational database management system. We tend towards simplicity for users processing Internet transactions.

Available tools for modeling and visualization of spatial data can be categorized as a stand-alone and web-based [10]. Typical stand-alone commercial products are ArcGIS and MapInfo. Both are expensive and dedicated for advanced users. As an alternative, PyNGL and PyNio applications, developed using Python programming language, are available. These applications generate 2D visualizations in several formats.

Among web-based visualization applications there are also solutions, which are based on commercial products (Bentley Map, ESRI) but many of these systems work only with their own datasets [11]. The prototype proposed in this paper is based on free software and allows presentation of user's data against a background of a third-source data.

## II. Data Extraction, Transformation, and Loading

In this section we describe an algorithm which is used to process input data and prepare presentation layer in our system.

To create the final presentation, we gather data from different sources:

1) company's private database of transactions,
2) administrative areas with zip codes,
3) statistical data for administrative regions.

The most important data comes from a transactional database of a company. Obviously, this information contains quantities, products, categories, clients, prices, values, etc. An initial data transformation module processes the data and prepares it to be imported into our tool. Possibly, the most simple way to import this data is to use a `*.csv` file where each row of the file describes a single purchase transaction. Each transaction in such a file is described by: time of the purchase, delivery zip code, delivery town, price of the purchase, quantity of the ordered product, and localization of a product in a hierarchy of types of products (see Fig. 2, *Purchase* table). In case of the data used by our prototype, we have two levels of type of the product. The product is the tire in this case. The top-level type is a brand of the tire while second-level type is the is a name of the tire, unique within the bounds of a single brand.

Each purchase can be approximately located on a map, and as such it is presented with respect to different levels of administrative division. Our system allows defining custom hierarchy of levels suitable for given application domain (see Fig. 2, *Administrative division hierarchy* group of tables). For example, in case of a company selling products in the USA, the hierarchy might look as follows: "state" $\rightarrow$ "county" $\rightarrow$ "city, town, or village". In case of our prototype, the data comes from a company selling products exclusively on the territory of Poland. Thus, while visualizing the spatial information, we use the information about Polish administrative territorial division. Polish territory consists of 16 voivodeships or provinces. Each voivodeship, called "wojewodztwo" in Polish, consists of a number of second level of local government administration areas, each one called "powiat". There is a total number of 379 powiats in Poland. For each considered administrative area, we

have obtained map coordinates of its center. A center for an administration region is calculated automatically as a centroid of administration area border taken from public government database [12]. For each powiat, we have also gathered a list of zip codes belonging to the powiat and town names connected with each zip code (see Fig. 2, *ZipCodeTown* table).

Because some town names can be written in a number of different but equivalent ways, we also use a simple text file in a `*.csv` format to store information about alternative spelling of names of some of the towns. In case of our prototype, two sample entries in this text file are: `12-220, Ruciane Nida, Ruciane-Nida` and `80-299, Gdansk-Osowa, Gdansk` where the first element is the zip code, the second one is the alternative spelling, and the last one is the canonical name (see Fig. 2, *TownNameAlias* table).

Our main goal in processing the above-mentioned data is to assign suitable administrative division areas of each level to each purchase transaction (i.e. delivery destination). To achieve this goal, we clean the data (described in Section II-A), then we transform and combine it (described in Section II-B), and finally we load into a final database used by an application that visualizes the data (see Fig. 3). In the description of the data processing, we concentrate mainly on the spatial dimension, but the temporal information is still present in the data, although its processing is limited mainly to generating the final summary statistics from selected time interval.

The last source of data is the statistical data of administrative division regions important for particular business. This could include for example population, climate, or number of high schools. If we posses an information connected to given administration area, it can be imported and presented together with statistical transaction information. It can also be used to normalize presented data, like for example displaying number of sold bottles of water per person.

### A. Data Cleaning

Due to characteristics of the input data i.e.: large influence of the human factor, possible mistakes, uncertainties, and ambiguity, the imported data has to be cleaned. The general approach is to accept correct records, repair the records that we know how to repair, and discard all others. The discarded data is saved in an auxiliary database along with information why each record was discarded. By inspecting this auxiliary database, we can check if the cleaning process improperly throws out useful records, and if it is the case, we can try to improve the cleaning algorithm.

Among all of the fields in the input records, the zip code and the town name have to be given a special care since normally they are entered by hand by each buyer using a web order form, and as a result there might be many possible versions of the same information entered. In case of our prototype, we deal with Polish zip code. It has a format of `XX-XXX`, where `X` is a single digit. While cleaning the zip code value, we: 1) remove all the spaces; 2) replace $\star$, _, =, / symbols with hyphen; 3) remove textual zip code suffix (if any) consisting of e.g. town name; 4) replace letters "o" and "O" with zero; 5) add hyphen in appropriate place; 6) add leading zero and a hyphen in a four-digit zip code without hyphen.

Next, while cleaning the town name value, we: 1) remove excessive spaces; 2) convert the name to a title format (a capital letter at the beginning of each word); 3) convert the name to the canonical name if it is in the table of the names with alternative spelling. In case of our prototype, the name of the analyzed town is sometimes "test" which is not a real name, but just a marker of a record created for test purposes. In such situations, the analyzed record is discarded.

Additionally, if type hierarchy of a purchased product is not fully specified in the record, the record is discarded. In case of our prototype, we discard the record if either brand or type name of a tire is absent.

### B. Combining Geospatial and Time Information

After doing the basic cleaning of the data, we try to assign administrative division area of the lowest level to each purchase record. It is worth noting that the higher-level administrative areas do not have to be assigned explicitly since each lower-level area is assigned to a single higher-level area. In case of our prototype, this task is done in two steps. In the first step, we look for a powiat identified uniquely by the given zip code only. If it fails, we proceed to the second step and look for the powiat identified uniquely by the purchase's (zip code, town name) pair. A more precise description of this process is presented as a pseudocode below:

$p \leftarrow$ {get powiat names associated with given zip code}
**if** $|p| = 1$ **then**
  {assign powiat to the given record}
**else if** $|p| = 0$ **then** {Given zip code was not found in the database}
  {discard record}
**else** {There was more than one powiat found for given zip code}
  {We were unable to uniquely identify the powiat using zip code only, so we will try to do it using also the town name}
  $p \leftarrow$ {get powiat names associated with given zip code and town name}
  **if** $|p| = 1$ **then**
    {assign powiat to the given record}
  **else if** $|p| = 0$ **then** {powiat for given (zip code, town name) pair was not found}
    {discard record}
  **else** {There was more than one powiat found for given (zip code, town name) pair}
    {discard record}
  **end if**
**end if**

As can be seen, the data is discarded in various points of the cleaning and transformation process. Since we are storing discarded data in an auxiliary database, we can easily generate some high-level statistics showing how much data was discarded and what was the reason of the rejection.

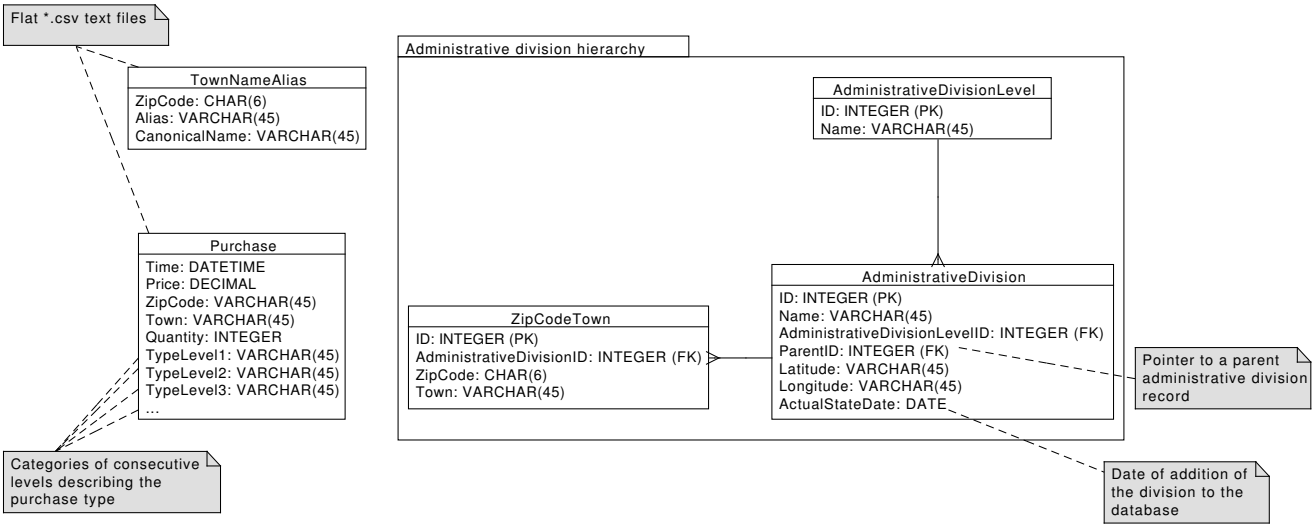


Fig. 2. A schema of the input data that we use to create the final database. Each box represents a logical database table. The following structures are presented: *Purchase* – a table of purchase transactions, *TownNameAliast* – a table of alternative spelling of names of selected towns, *Administrative division* – group of tables describing the administrative division of the area.

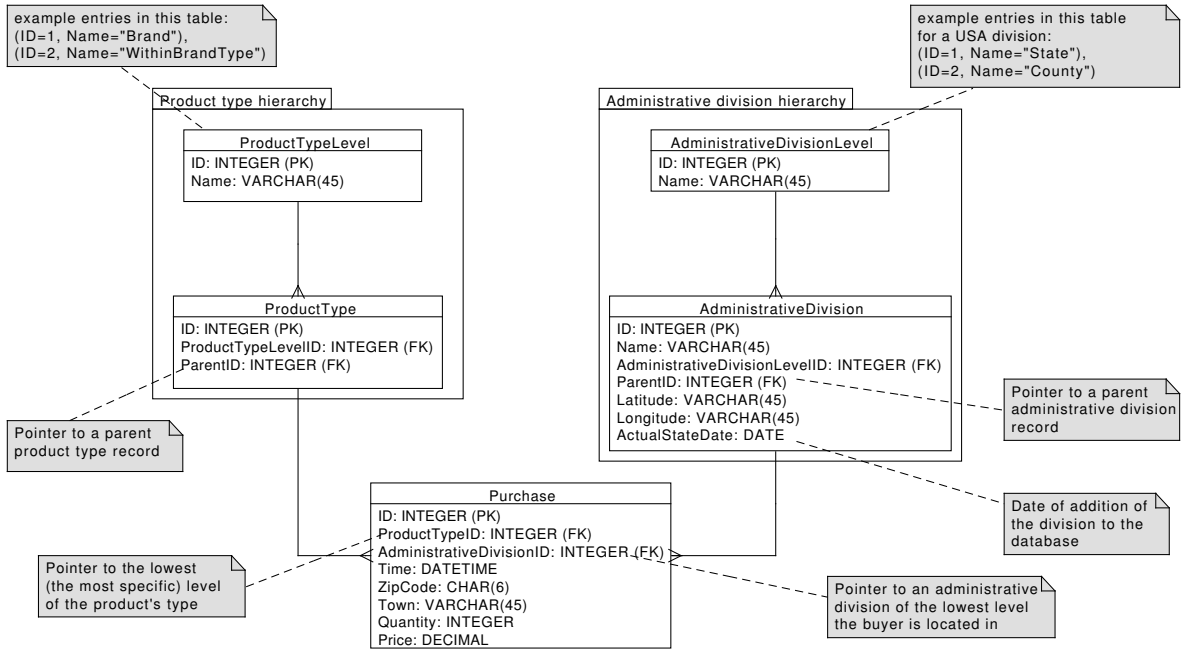


Fig. 3. A schema of the final database used by the application.

Overall, just a small percentage of the data is discarded in the cleaning process.

### C. Statistical Processing and Data Output

After the transactions data is cleaned and stored in a database, we can start extracting statistical information. In many cases, for most of the small companies, this step will just calculate simple aggregates like sum of transactions or average purchase value. A set of simple queries may be used to process this information and send it to the presentation layer. However, in various situations a more complicated statistics like growth of value per product category may be needed. In these cases an additional OLAP system may be used to store temporal aggregates.

A map feeder module uses all available information to create the presentation layer. It combines area, transactions,

and statistics to automatically produce results. Due to possibly large volume of data to be processed it should work offline and in a batch mode. Output data, depending on a time window involved and length of analysed period, may have from tens to hundreds of megabytes.

The output data is prepared in a format acceptable by the presentation module (see the next section for details).

## III. Presentation Layer of Working Prototype

The final database (see Fig. 3) comprising of integrated data from different sources is used as a basis for producing input data for the visualization module. Although any tool can be used to display the data, it should have at least basic capabilities required for user-friendly operations:

- zooming in and out with administration areas appearing automatically,
- panning around the map,
- displaying of user data values,
- time axis and ability to move in time and display data for given time window,
- ability to divide user data into layers or provide data grouping.

The Google Earth (GE) application has all the above-mentioned characteristics, that is why we use it as a visualization engine in our prototype. The data accepted by the GE is described in an XML file in a format called OpenGIS KML Encoding Standard (abbreviated simply as "KML"). This format is specifically designed to describe a way of visualizing geographic data. We use its basic capabilities to visualize sales data bars as three-dimensional polygons on a three-dimensional map of Poland.

After loading the `*.kml` file generated from our database into the GE, the user sees sales performance bars placed on each administrative area. There are four bars per area, each one corresponds to sales performance in one of four consecutive months (see Fig. 4). The user can utilize many options implemented in GE to navigate and manipulate data:

- change viewed time frame,
- run month-by-month animation showing changes in sales performance (see Fig. 5),
- change point of view,
- select subset of the data to visualize,
- get detailed information about a selected sale (number of transactions, total value, etc.).

One of the most important limitations of GE as a visualization tool in our system is that data subsets may only be defined as disjoint groups in XML format. Therefore data must be repeated in many so-called folder structures in order to achieve visualization of the same property in different layers or areas. This could result in a huge KML files if one would like to see different products divided into different areas. Also, adding time dimension multiplies the file size by the number of time steps. We observed in our prototype a file size growth of two orders of magnitude when using 24 time steps (each corresponding to a single month) instead of using a single aggregated step.

However, the system should also work with larger datasets. KML-based models can manage datasets with millions of records [13]. This is especially true when a network links technique is used [14].

## IV. Conclusions and Future Works

We presented a lightweight automatic system for combining, processing and presenting sales-related data. The presented prototype of the system relies on batch processing for data analysis and on Google Earth application as a viewing module. Our solution, although very simple, could be used by most of Internet sellers providing them a simple and convenient way to observe spatial and temporal relationships in sales data. Future work on the system involves a more thorough incorporation of the statistical data of the administrative regions into the system. Another idea is to provide visualization of results of some basic data mining processing of the analyzed data (e.g. showing clusters of regions that are similar in some specified way).

## Acknowledgment

The authors would like to thank one of the largest Internet sellers of tires in Poland: ORZEŁ S.A., Ćmiłów ul. Willowa 2-4, 20-388 Lublin, Poland. The company made available approximately two years of Internet purchase transactions data to us.

## References

[1] G. Corporation, "Google earth," www.google.com/earth.
[2] Y. Bédard, S. Rivest, and M. josée Proulx, "Spatial on-line analytical processing (solap): Concepts, architectures, and solutions from a geomatics engineering perspective," in *Data Warehouses and OLAP: Concepts, Architecture, and*. Press, 2006, p. 298319.
[3] A. Guttman, "R-trees: A dynamic index structure for spatial searching," in *International Conference on Management of Data*. ACM, 1984, pp. 47–57.
[4] Y. Theodoridis and T. Sellis, "A model for the prediction of r-tree performance," 1996, pp. 161–171.
[5] Y. Tao, J. Sun, and D. Papadias, "Analysis of predictive spatio-temporal queries," *TODS*, vol. 28, pp. 295–336, 2003.
[6] M. Hadjieleftheriou, G. Kollios, V. J. Tsotras, and D. Gunopulos, "Efficient indexing of spatiotemporal objects," 2002, pp. 251–268.
[7] Y. Theodoridis, M. V. . T. Sellis, M. Vazirgiannis, and T. Sellis, "Spatio-temporal indexing for large multimedia applications," 1996, pp. 441–448.
[8] Y. Tao, G. Kollios, J. Considine, F. Li, and D. Papadias, "Spatio-temporal aggregation using sketches," in *In ICDE*, 2004, pp. 214–226.
[9] J. Zhang, "Spatio-temporal aggregation over streaming geospatial data," in *In Proceedings of the 10th International Conference on Extending Database Technology Ph.D. Workshop*, 2006.
[10] D. Kannangara, N. Fernando, and D. Dias, "A web based methodology for visualizing time-varying spatial information," in *Industrial and Information Systems (ICIIS), 2009 International Conference on*, dec. 2009, pp. 233 –238.
[11] J. K.P. and W. N.T.S, "Product development for presentation of temporal gis results for non gis specialists, engineer," *Journal of the Institution of Engineers*, vol. 51, no. 5, pp. 44 –50, 2008.
[12] Surveyor General of Poland, "geoportal.gov.pl," geoportal.gov.pl.
[13] J. Wood, J. Dykes, A. Slingsby, and K. Clarke, "Interactive visual exploration of a large spatio-temporal dataset: Reflections on a geovisualization mashup." *Visualization and Computer Graphics, IEEE Transactions on*, vol. 13, no. 6, pp. 1176 –1183, nov.-dec. 2007.
[14] U. Dadi, C. Liu, and R. Vatsavai, "Query and visualization of extremely large network datasets over the web using quadtree based kml regional network links," in *Geoinformatics, 2009 17th International Conference on*, aug. 2009, pp. 1 –4.

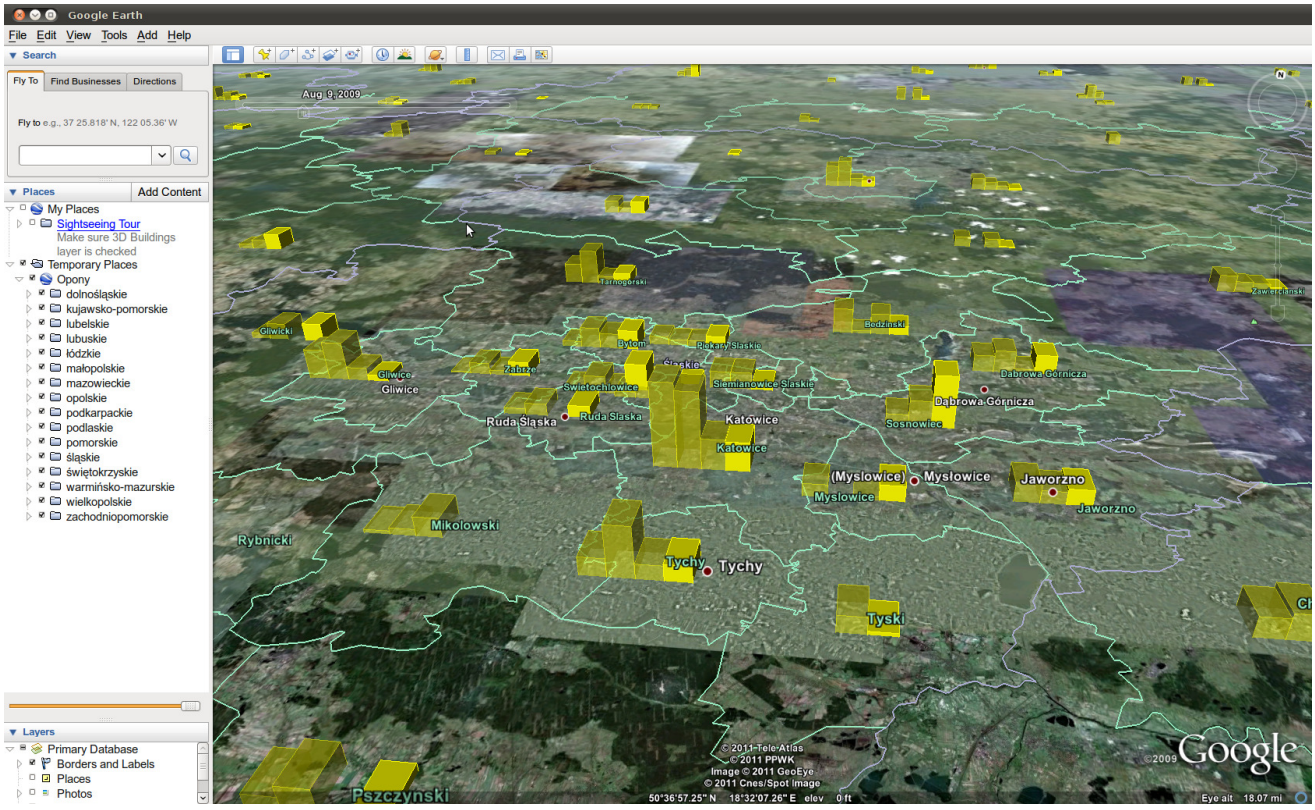


Fig. 4. A sample screenshot of the presentation layer. We can see a Google Earth's satellite image of a part of Poland with borders of powiats marked. Three-dimensional bars on the territory of each powiat correspond to sales performance in four consecutive months with the brightest bar on the right side corresponding to the most recent month.

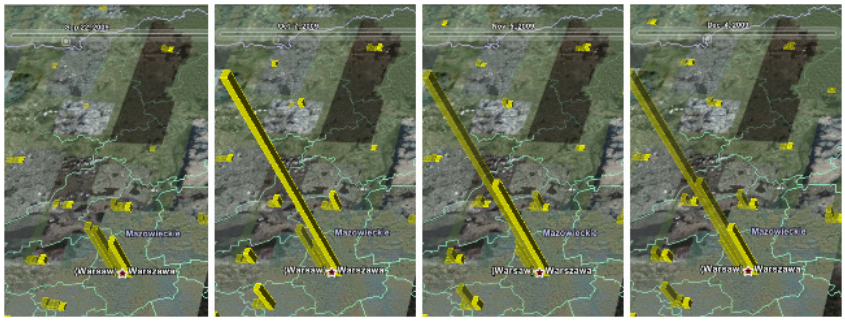

Fig. 5. Subsequent frames of animation showing how the sales performance is changing during four consecutive months. The brightest bar on the right side corresponds to the most recent month.