

**ZASTOSOWANIE JĘZYKA KML ORAZ SYSTEMU BAZ DANYCH
POSTGRESQL/POSTGIS DO WIZUALIZACJI WEKTOROWYCH DANYCH
PRZESTRZENNYCH W INTERNECIE**

**THE USE OF KML AND POSTGRESQL/POSTGIS DATABASE SYSTEM TO
VISUALIZE VECTOR DATASETS ON THE INTERNET**

Krzysztof Kuśnierek

Instytut Geografii Fizycznej i Kształtowania Środowiska Przyrodniczego,
Uniwersytet im. Adama Mickiewicza w Poznaniu

SŁOWA KLUCZOWE: kartografia internetowa, wizualizacja danych wektorowych, KML, Google Earth, Google Maps, PostgreSQL, PostGIS, PHP

STRESZCZENIE: Celem artykułu jest omówienie opracowanego narzędzia geomatycznego, pozwalającego zoptymalizować prezentację zbiorów geoinformacji wektorowej w Internecie. Niniejsza praca opisuje metodę konfiguracji popularnego globusa elektronicznego GoogleEarth oraz kartograficznej aplikacji internetowej Google Maps z danymi przestrzennymi, które zebrane zostały w systemie baz danych PostgreSQL/PostGIS. Istotą zaproponowanego rozwiązania jest dynamiczna wizualizacja wektorowych danych przestrzennych, których zakres przestrzenny odpowiada obszarowi widocznemu na ekranie użytkownika. Zakres obszaru widzianego na ekranie jest generowany przez przeglądarkę kartograficzną w postaci współrzędnych geograficznych, które są następnie odczytywane przez autorski program dynpg2kml opracowany w języku PHP. Program ten komunikuje się z systemem PostgreSQL/PostGIS i zwraca pozyskane z niego dane wektorowe w postaci kodu w języku KML. Kod ten jest odczytywany, przetwarzany i wyświetlany przez aplikacje GoogleEarth oraz Google Maps. Główną zaletą zaproponowanego rozwiązania jest możliwość prezentacji szczegółowych danych wektorowych, transmitowanych dynamicznie w niewielkich pakietach. W porównaniu do transmisji całości zbioru danych w jednym pakiecie, skrócony zostaje czas oczekiwania użytkownika na informację, co ma duże znaczenie w przypadku dystrybucji dużych zbiorów danych oraz w warunkach ograniczonej prędkości transmisji danych. Możliwości wykorzystywanego systemu pozwalają na dalszy rozwój zaproponowanego programu. Jako przykład, przedstawiona zostanie metoda dostosowywania szczegółów geometrii wyświetlanych obiektów wektorowych do skali prezentowanej mapy.

1. WPROWADZENIE

Informacja geograficzna jest coraz chętniej rozpowszechniana nieodpłatnie, do czego przyczyniają się producenci oprogramowania oraz administracje rządowe i międzynarodowe. Na wzrost dostępności geoinformacji wpłynęło również pojawienie się na rynku takich aplikacji jak Google Earth i Google Maps, przez co znacznemu powiększeniu uległa również liczba jej użytkowników związanych z administracją,

biznesem oraz użytkowników indywidualnych (Chen et al., 2002; Goodchild, 2007). Poprzez zastosowanie w tych aplikacjach łatwych w użyciu interfejsów graficznych wielu użytkowników, którzy nie są specjalistami w dziedzinie GIS, może w prosty sposób uzyskiwać bezpośredni dostęp do bogatej informacji przestrzennej (Skiba, 2006).

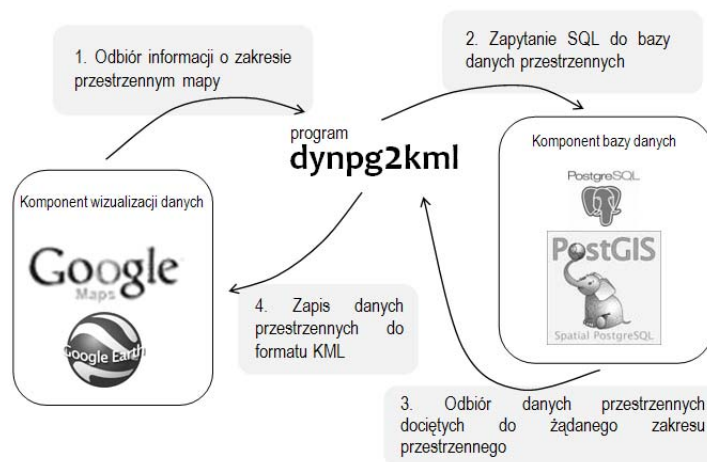
Jednym z najnowszych standardów zapisu informacji geograficznej jest format KML. Głównymi jego zaletami są: przejrzystość formy zapisu oraz popularność osiągnięta poprzez kompatybilność z powszechnie wykorzystywanymi przeglądarkami danych geograficznych. Dotyczy to nie tylko aplikacji Google Earth oraz Google Maps, ale również najważniejszych systemów informacji geograficznej. Dodatkowo, w ostatnich latach coraz bardziej rozpowszechnione jest przechowywanie wektorowych danych geograficznych w bazach danych. Rozwiązanie to umożliwia tworzenie relacji pomiędzy tabelami w bazie danych (w tym relacji przestrzennych) i wybieranie z nich tylko tych informacji, które są wymagane przez użytkownika (Goodchild, 2007). Istnieją bowiem techniczne ograniczenia dystrybucji danych, związane w szczególności z przepustowością łącza internetowego transmitującego dane geograficzne. Istnienie tego wąskiego gardła w dostarczeniu informacji geograficznej przez sieć Internet wymaga od dostawców danych oraz producentów oprogramowania stosowania technik polegających na ograniczeniu ilości transmitowanych danych lub zmniejszeniu ich jakości.

W niniejszym artykule omówiono system dynamicznej wizualizacji danych przestrzennych, który może być stosowany przez dostawców dysponujących dużymi zbiorami danych wektorowych udostępnianych przez Internet. W kolejnych etapach pracy przedstawiony jest schemat funkcjonowania systemu oraz poszczególne komponenty wchodzące w jego skład. Proponowany jest program `dynpg2kml`, umożliwiający użytkownikowi przeglądarek geograficznych otrzymywanie informacji w sposób bardziej dynamiczny. Szczegółowo omówione są elementy programu odpowiadające za dynamiczną wizualizację danych geograficznych, w postaci kodu KML, w zależności od zakresu przestrzennego wizualizacji oraz sposób dodawania wygenerowanych danych geograficznych do analizowanych przeglądarek geoinformacji: Google Earth oraz Google Maps.

2. METODY

2.1. Schemat funkcjonowania systemu dynamicznej wizualizacji geoinformacji wektorowej

Rysunek 1 przedstawia zasadę i etapy działania proponowanego rozwiązania geoinformatycznego. W jego skład wchodzi dwa główne komponenty: komponent bazy danych oparty na dostępnym na licencji BSD systemie baz danych PostgreSQL i nakładce przestrzennej PostGIS oraz komponent wizualizacji danych, w którego skład wchodzi dwie popularne przeglądarki geoinformacji: Google Maps i Google Earth. Oba komponenty są spięte programem `dynpg2kml`, opracowanym dla celów tej pracy. Charakterystyka zarówno komponentów jak i samego programu omówiona jest w dalszej części pracy. W tym miejscu należy zwrócić uwagę na interakcję pomiędzy elementami systemu (rys. 1).



Rys. 1. Schemat funkcjonowania systemu dynamicznej wizualizacji geoinformacji wektorowej

Użytkownik przeglądarki danych geograficznych wykonując interaktywną nawigację definiuje zakres przestrzenny wizualizowanego obszaru. Przeglądarka udostępnia informację o przestrzennym zakresie mapy, opisaną za pomocą współrzędnych geograficznych. Informacja ta odbierana jest przez program *dynpg2kml*, a następnie zostaje zapisana w odpowiednich zmiennych programowych. Na ich podstawie omawiany program tworzy zapytanie SQL do bazy danych przestrzennych i przygotowuje odpowiednie zestawienie w formie binarnej. Dane odebrane przez program są, za pomocą odpowiedniej funkcji, poddane konwersji do postaci wycinków kodu KML.

2.2. Komponent bazy danych

Niniejsza praca zakłada korzystanie z testowego zbioru danych wektorowych, pochodzącego z Bazy Danych Ogólnogeograficznych (BDO), zapisanego w systemie baz danych PostgreSQL, obejmującego parki krajobrazowe w Polsce. PostgreSQL jest potężnym obiektowo-relacyjnym systemem baz danych (Postgresql, 2009). Ważnym elementem komponentu bazy danych jest aplikacja PostGIS. PostGIS jest: „rozszerzeniem obiektowo-relacyjnego systemu baz danych PostgreSQL, umożliwiającym przechowywanie w bazie danych obiektów GIS” (Postgis, 2009). Baza danych stworzona w systemie PostgreSQL/PostGIS zawiera rozbudowany zestaw indeksów i funkcji przestrzennych, umożliwiającym szeroko rozumiane przetwarzanie zapisanych w niej danych przestrzennych. Dane zawarte w bazie można wywołać za pomocą standardowego języka zapytań SQL.

2.3. Komponent wizualizacji danych przestrzennych

Funkcjonowanie programu *dynpg2kml* zostało przetestowane na dwóch typach przeglądarek wizualizujących geoinformację w sieci Internet: aplikacji instalowanej na

dysku lokalnym oraz aplikacji dostępnej z poziomu przeglądarki internetowej. Głównym kryterium wyboru przeglądarek była ich popularność wśród użytkowników informacji geograficznej w Internecie w Polsce. Pod uwagę wzięto aplikacje oferowane bezpłatnie przez firmę Google Inc: Google Earth oraz Google Maps.

Google Earth jest globusem elektronicznym funkcjonującym jako aplikacja instalowana na komputerze użytkownika. Google Earth umożliwia prezentację zdjęć satelitarnych, lotniczych, panoram ulic oraz informacji geograficznej na trójwymiarowym modelu kuli ziemskiej (Google, 2009a). Google Earth jest aplikacją rozszerzalną, tj. umożliwiającą użytkownikowi dodawanie własnych danych przestrzennych, zapisanych w postaci kodu KML. Google Maps jest natomiast aplikacją internetową udostępniającą usługi i technologię kartograficzną (Google, 2009b). Google Maps również jest aplikacją rozszerzalną poprzez udostępnienie użytkownikowi interfejsu programowania aplikacji API, dzięki czemu umożliwia się tworzenie zaawansowanych aplikacji internetowych. Z punktu widzenia aplikacji opracowanej w ramach niniejszej pracy, najważniejszą cechą łączącą obie aplikacje jest możliwość importu przez nie danych zapisanych w formacie KML. Język KML (Keyhole Markup Language) jest zgodny z gramatyką i formatem języka XML. Służy on do modelowania i zapisywania obiektów geograficznych w różnych aplikacjach (Google, 2009c). Język KML stał się na tyle powszechnym sposobem zapisu danych przestrzennych, iż został przyjęty przez OGC jako międzynarodowy standard geoinformacyjny (Opengeospatial, 2009).

3. WYNIKI

3.1. Dynamiczna wizualizacja danych geograficznych w zależności od zakresu przestrzennego wizualizacji

Elementem spinającym komponent bazy danych z komponentem wizualizacji danych zawartym w systemie dynamicznego generowania geoinformacji wektorowej jest program dynpg2kml. Program dynpg2kml został napisany w języku PHP i jest umieszczony na serwerze internetowym <http://webgis.geoinformacja.pl/dynpg2kml/>.

Pierwszym elementem programu jest interfejs komunikacji z przeglądarką użytkownika. W proponowanym rozwiązaniu użytkownik, wykonując dynamiczną nawigację na mapie określa zakres przestrzenny obszaru mapy. Informacja ta jest przesyłana z przeglądarki do programu w wygenerowanym przez nią ciągu CGI (Rys. 2). Dzięki temu rozwiązaniu, do programu dostarczone są niezbędne informacje potrzebne do stworzenia nowego tymczasowego obiektu przestrzennego, wykorzystanego w dalszej części do ustalenia przestrzennej relacji z danymi zawartymi w bazie danych.

```
//przykładowy ciąg CGI
//http://webgis.geoinformacja.pl/dynpg2kml/dynpg2kml.php?south=51.80351990955801&
west=16.00433349609375&north=52.6147226179558&east=17.76763916015625
$south = $_GET['south'];
$west = $_GET['west'];
$north = $_GET['north'];
$east = $_GET['east'];
```

Rys. 2. Fragment kodu programu dynpg2kml odpowiedzialny za pozyskanie informacji o zakresie przestrzennym mapy

Kolejną procedurą programu jest zdefiniowanie połączenia z bazą danych przestrzennych zawartą w systemie PostgreSQL/PostGIS. Na poniższej rycinie przedstawiono wycinek kodu odpowiedzialny za zdefiniowanie tego połączenia (Rys. 3).

```
$username = 'admin';
$password = 'secret123';
$dbGeomName = 'natura';
$hostname = 'localhost';
$polaczenie = pg_pconnect('host='.$hostname.' port=5432
                        dbname= '.$dbGeomName.' user='.$username.' password='.$password);
```

Rys. 3. Wycinek kodu dynpg2kml odpowiedzialny za zdefiniowanie połączenia z bazą danych przestrzennych PostgreSQL/PostGIS

W dalszej części dochodzi do zdefiniowania zapytania SQL skierowanego do bazy danych przestrzennych. Przykładowe zapytanie SQL (Rys. 4), definiuje wybór 4 kolumn z tabeli 'landpark', zawierających dane o parkach krajobrazowych w Polsce. W kolumnie gid, area, naz oraz the_geom zapisano kolejno: nr parku krajobrazowego w bazie, powierzchnię parku, nazwę parku oraz jego granice opisane w formie binarnej. Na dane przestrzenne zawarte w kolumnie the_geom nałożono trzy funkcje. Funkcja ST_Intersection odpowiada za wydobycie z całego zbioru parków krajobrazowych tylko tych poligonów, które znajdują się na obszarze zdefiniowanym przez zakres przestrzenny mapy. Funkcja ST_Simplify omówiona jest w dalszej części pracy, natomiast funkcja ST_AsKml odpowiada za konwersję danych do postaci tekstowej sformatowanej zgodnie ze standardem KML. Ostatnia część zapytania SQL odnosi się do wykorzystania indeksu przestrzennego GiST przyspieszającego wyszukiwanie danych.

```
$spatialSQL = 'SELECT gid, area, naz,
              ST_AsKml(ST_Simplify(ST_Intersection(the_geom,
              .$intersectPolygon.'), '.$simplifyTolerance.')) AS geo FROM landpark
              WHERE the_geom && setSRID('\BOX3D('.$west.' '.$south.',
              '.$east.' '.$north.')\'::box3d, '.$srid.')
              ORDER BY gid DESC';
```

Rys. 4. Fragment programu dynpg2kml odpowiedzialny za zdefiniowanie zapytania SQL do bazy danych

Końcowa część programu odpowiada za wygenerowanie kodu tekstowego w formacie KML. Polega ono na prostym wydruku kodu w formie przewidzianej przez standard oraz odpowiednim umieszczeniu wycinków danych wygenerowanych przez zapytanie SQL. Poniżej przedstawiono fragment programu dynpg2kml odpowiadający za dynamiczne tworzenie kodu KML dotyczącego obiektów przestrzennych (Rys. 5).

```
while ($wiersz = pg_fetch_row($wynik)) {
echo "<Placemark>\n"
$namein = "<name>".$wiersz[2]."</name>\n";
echo "iconv('ISO-8859-2', 'UTF-8', $namein)";
$descin = "<description>Park krajobrazowy nr ".$wiersz[0]."
o nazwie ".$wiersz[2]." zajmuje obszar ".$wiersz[1]."
m2</description>";
echo "iconv('ISO-8859-2', 'UTF-8', $descin)";
echo "<styleUrl>#testowy</styleUrl>\n";
echo "$wiersz[3]";
echo "</Placemark>\n";}
```

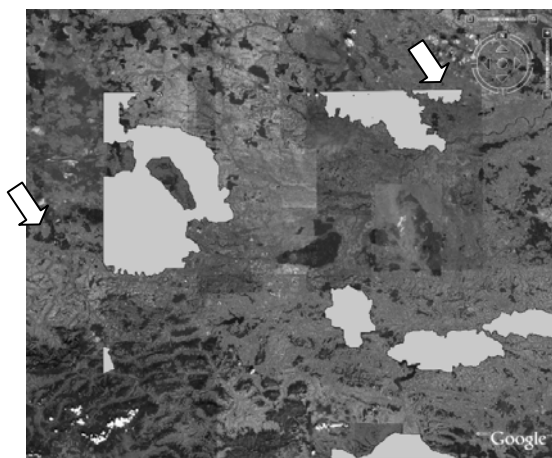
Rys. 5. Fragment programu dynpg2kml odpowiadający za dynamiczne tworzenie kodu KML dotyczącego obiektów przestrzennych

3.2. Dynamiczne dodawanie geoinformacji w postaci kodu KML do aplikacji Google Earth

Dodawanie danych w postaci kodu KML do Google Earth realizowane jest za pomocą elementu języka KML, jakim jest <NetworkLink> oraz zawarty w nim element <Link>. W omawianej procedurze dynamicznego ładowania danych przestrzennych do globusa Google Earth zdecydowano się stworzyć odrębny plik KML – landparks.kml służący do nawiązania połączenia z programem dynpg2kml. Plik landparks.kml znajduje się na stronie internetowej <http://webgis.geoinformacja.pl/dynpg2kml/>. Poniżej przedstawiono fragment tego pliku, odpowiedzialny za dynamiczne ładowanie obiektów wektorowych do aplikacji Google Earth (Rys. 6).

```
<NetworkLink>
<name>Parki krajobrazowe</name>
<Link>
<href>http://webgis.geoinformacja.pl/dynpg2kml/dynpg2kml.php?
</href>
<viewRefreshMode>onStop</viewRefreshMode>
<viewRefreshTime>2</viewRefreshTime>
<viewFormat>west=[bboxWest]&amp;south=[bboxSouth]&amp;east=
[bboxEast]&amp;north=[bboxNorth]</viewFormat>
</Link>
</NetworkLink>
```

Rys. 6. Fragment pliku landparks.kml odpowiedzialny za dynamiczne ładowanie wektorowych obiektów do aplikacji Google Earth



Rys. 7. Zrzut ekranowy aplikacji Google Earth z umieszczonymi w niej dynamicznie wygenerowanymi obiektami wektorowymi (zrzut wykonany w momencie przesuwania widoku mapy w kierunku oznaczonym strzałkami)

Element <NetworkLink>, o nazwie „Parki krajobrazowe”, zawiera podelement <Link>, definiujący bezwzględny adres do programu dynpg2kml oraz sposób odświeżania obrazu, który ustawiono na 2 sekundy po zatrzymaniu przesuwania globusa Google Earth. Należy zwrócić uwagę na element <viewFormat> określający zmienne ciągu URL dołączane do adresu do programu dynpg2kml. Za ich pośrednictwem tych zmiennych, dane są pozyskane z serwera internetowego i zainstalowanego na nim systemu zarządzania bazami danych przestrzennych PostgreSQL/PostGIS.

Zamieszczono również zrzut ekranowy aplikacji Google Earth, z umieszczonymi w niej dynamicznie wygenerowanymi obiektami wektorowymi (Rys. 7). W celu zademonstrowania dynamicznego charakteru obiektów przestrzennych dodawanych do mapy, zrzut ekranowy wykonano w momencie nawigacji wykonywanej przez użytkownika globusa Google Earth. Kierunek nawigacji przedstawiony jest za pomocą białych strzałek. Na rysunku widać jak dane zebrane w bazie danych przestrzennych były dopasowane do zakresu przestrzennego poprzedniego widoku mapy.

3.3. Dynamiczne dodawanie geoinformacji w postaci kodu KML do Google Maps

Jak wcześniej wspomniano, aplikacja Google Maps udostępnia użytkownikowi interfejs programowania aplikacji API będący zbiorem funkcji i właściwości umożliwiających rozszerzenie funkcjonalności tej aplikacji internetowej. Dzięki temu umożliwia realizację zaawansowanej interakcji użytkownika z mapą. Google Maps udostępnia również rozszerzenie umożliwiające dodawanie danych zapisanych pod postacią kodu KML (Rys. 8).

```
map.addOverlay(geoXml);  
geoXml = new GGeoXml(url);
```

Rys. 8. Fragment skryptu odpowiedzialnego za dynamiczne dodawanie danych przestrzennych zapisanych w formacie KML do mapy Google Maps

Parametr „url” jest tutaj odwołaniem do miejsca przechowywania danych geograficznych zapisanych w formacie KML. W standardowym rozwiązaniu jest nim bezwzględny adres URL, wskazujący na plik o rozszerzeniu kml zamieszczony na określonym serwerze danych. W proponowanym rozwiązaniu kod KML tworzony jest dynamicznie. Zatem, zamiast do pliku z danymi, stworzono tutaj odwołanie do programu dynpg2kml, który na podstawie informacji przesłanych w parametrach ciągu CGI, dynamicznie generuje kod w formacie KML. Fragment skryptu odpowiedzialnego za komunikację Google Maps z programem dynpg2kml w postaci KML jest przedstawiony na poniższym rysunku (Rys. 9).

```
var url =  
  "http://webgis.geoinformacja.pl/dynpg2kml/dynpg2kml.php?  
  south="+south+"&west="+west+"&north="+north+"&east="+east;
```

Rys. 9. Fragment skryptu odpowiedzialnego za komunikację Google Maps z programem dynpg2kml dynamicznie generującego dane w postaci KML

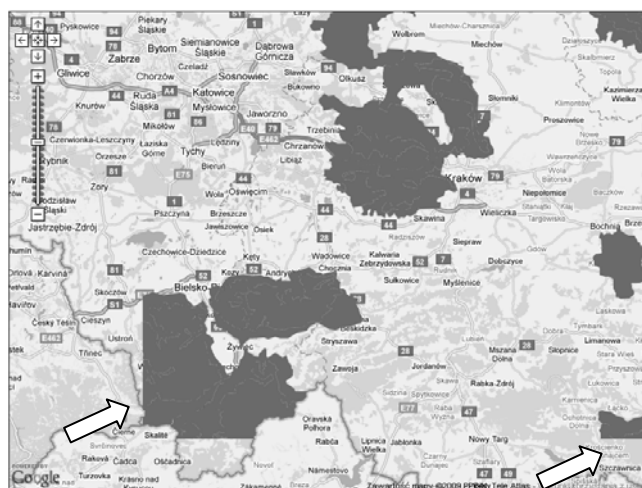
Dane zawarte w wygenerowanym kodzie KML wynikać mają z określonego przez użytkownika zakresu przestrzennego. Aplikacja internetowa Google Maps definiuje skrajne współrzędne geograficzne prezentowanego obszaru. Dostęp do nich jest uzyskiwany poprzez zastosowanie funkcji Google Maps API takich jak `getBounds()`, `getSouthWest()` oraz `getNorthEast()`.

Następnie zapisywane są współrzędne określające obszar wizualizacji do zmiennych „south”, „West”, „North” oraz „east”. Zmienne te są dodawane do ciągu URL. Za ich pomocą dane pozyskiwane są z serwera internetowego i zainstalowanego na nim systemu zarządzania bazami danych przestrzennych PostgreSQL/PostGIS. Na rysunku 10 przedstawiono zrzut ekranowy przeglądarki internetowej wyświetlającej powyższą stronę internetową <http://webgis.geoinformacja.pl/dynpg2kml/>, z zamieszczoną w niej aplikacją Google Maps. Podobnie jak w przedstawionym powyżej przykładzie dotyczącym aplikacji Google Earth, zrzut ekranowy wykonano w momencie nawigacji wykonywanej przez użytkownika. Kierunek nawigacji przedstawiony jest za pomocą białych strzałek. Na rysunku widać, jak dane zebrane w bazie danych przestrzennych były dopasowane do zakresu przestrzennego poprzedniego widoku mapy.

3.4. Dalszy rozwój projektu

Prędkość przesyłania danych jest jednym z podstawowych czynników decydujących o przystępności opracowywanego narzędzia w warunkach wysokiej jakości prezentowanej treści lub ograniczonej prędkości transferu danych. Podstawowym wymogiem użytkownika jest niewielki czas oczekiwania na odpowiedź. W celu optymalizacji transferu danych można dokonać ograniczania ilości przesyłanych danych, co polega na dostosowaniu aplikacji do poziomu uogólnienia mapy. Dla map w małej skali należy generować uogólnione granice, natomiast przy prezentacji obszaru w dużej skali aplikacja internetowa powinna wysyłać dane w postaci szczegółowej. W ramach rozwoju programu dynpg2kml zaproponowano metodę redukcji szczegółów wyświetlanej geometrii poprzez wykorzystanie funkcji PostGIS: `ST_Simplify`. Funkcja ta transponuje obiekty, wykorzystując algorytm generalizacji Douglas'a-Peucker'a (Douglas i Peucker, 1973). Funkcja `ST_Simplify` składa się z dwóch parametrów: zbioru obiektów geometrycznych

mających zostać poddanych uogólnieniu oraz parametru uogólnienia. Wykorzystanie omawianej funkcji w opracowanym programie wpłynęło znacznie na czas przesyłania danych wektorowych z serwera do przeglądarki użytkownika. Innym możliwym kierunkiem rozwoju projektu jest dodatkowa optymalizacja ilości danych poprzez redukcję dokładności lokalizacji wyświetlanych obiektów przestrzennych wyświetlanych w mniejszych skalach.



Rys. 10. Zrzut ekranowy przeglądarki internetowej wyświetlającej stronę internetową z zamieszczoną na niej aplikacją Google Maps, w której umieszczono dynamicznie wygenerowane obiekty wektorowe (zrzut wykonany w momencie przesuwania widoku mapy w kierunku oznaczonym strzałkami)

4. PODSUMOWANIE

W niniejszym artykule przedstawiono komponenty bazy i wizualizacji danych wchodzące w skład omawianego systemu. Szczególną uwagę zwrócono na zasadę działania programu dynpg2kml jako elementu spinającego oba komponenty. Program dynpg2kml działający w otoczeniu wektorowych danych przestrzennych zebranych w bazie danych w systemie PostgreSQL/PostGIS oraz aplikacji Google Earth i Google Maps umożliwia dynamiczne wyświetlanie geoinformacji w postaci wektorowej. Dzięki zastosowanemu narzędziu geomatycznemu, użytkownik ma możliwość otrzymania szczegółowych danych wektorowych, transmitowanych dynamicznie w niewielkich pakietach, co w porównaniu do transmisji całości zbioru danych w jednym pakiecie, skraca czas oczekiwania na informację. Zastosowanie wybranych komponentów umożliwia dalszy rozwój przedstawionego narzędzia np. poprzez dostosowanie uogólnienia wyświetlanej geometrii w zależności od skali wizualizowanego obszaru. Metoda ta wymaga optymalizacji i wykonania szeregu testów, niemniej jej wykorzystanie w obecnej postaci poprawia działanie programu dynpg2kml.

5. LITERATURA

- Chen A.J., Li Q., Xu G.Y., 2002. Geospatial information layer sharing model for digital earth, Ruan Jian Xue Bao/Journal of Software, v 13, n 8, August, s. 1436-1440.
- Douglas, D.H., Peucker, T.K., 1973. Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. The Canadian Cartographer, v. 10, s. 112-123.
- Goodchild, M.F., 2007. Citizens as Voluntary Sensors: Spatial Data Infrastructure in the World of Web 2.0, International Journal of Spatial Data Infrastructures Research, 2007, Vol. 2, s. 24-32.
- Google, 2009a: http://earth.google.com/userguide/v4/ug_kml.html
- Google, 2009b: <http://code.google.com/intl/pl-PL/apis/maps/>
- Google, 2009c: <http://code.google.com/apis/kml/documentation/>
- Opengeospatial, 2009: <http://www.opengeospatial.org/standards/kml/>
- Postgis, 2009: <http://postgis.refractor.net/documentation/manual-svn/>
- Postgresql, 2009: <http://www.postgresql.org/>
- Skiba, D.J., 2006. Web 2.0: next great thing or just marketing hype? Nursing education perspectives 27(4), s.212-214.

THE USE OF KML AND POSTGRESQL/POSTGIS DATABASE SYSTEM TO VISUALIZE VECTOR DATASETS ON THE INTERNET

KEY WORDS: the Internet cartography, vector dataset visualization, KML, Google Earth, Google Maps, PostgreSQL, PostGIS, PHP

SUMMARY: The goal of the paper is to describe an original geomatic tool for the optimized visualization of vector datasets on the Internet. This paper covers a configuration method of Google Earth popular digital globe and Google Maps cartographic Internet application with spatial data collected in PostgreSQL/PostGIS database system. The essence of the proposed solution is the dynamic visualization of vector data fragments, whose spatial range corresponds to the area of the user's screen. The spatial range on the user's screen, recorded in geographic coordinates, is generated by the cartographic browser and is read afterwards by the author's original programme dynpg2kml written in PHP. The programme communicates with PostgreSQL/PostGIS database system and returns vector datasets in KML code. The code is read, parsed and visualized by the Google Earth and Google Maps software. The main advantage of this solution is the possibility of presentation of detailed vector datasets, transmitted dynamically in small packets. In comparison to the transmission of whole data collection in one packet, the user's anticipation time decreases. It is considerably important in case of the distribution of large datasets or under low data transfer conditions. The possibilities of the system used here allow its further development. A method of adapting the geometry details to the presentation scale is described here, too.

Mgr Krzysztof Kuśnierek
e-mail: kus@amu.edu.pl
tel. +48 691183887
fax +61 8296230