



**POLSKIE
TOWARZYSTWO
INFORMACJI
PRZESTRZENNEJ**

ROCZNIKI **2003** **GEOMATYKI**

**Podstawy metodyczne i technologiczne
infrastruktur geoinformacyjnych**

Janusz Michalak

**Tom I
Zeszyt 2
Warszawa**

JANUSZ MICHALAK
Wydział Geologii Uniwersytetu Warszawskiego
Al. Żwirki i Wigury 93, 02-089 Warszawa
e-mail: J.Michalak@geo.uw.edu.pl
tel. (022) 55-40-529 fax (022) 55-40-001
<http://netgis.geo.uw.edu.pl>

Spis treści

1. Wstęp	11
2. Podstawowe założenia INSPIRE	12
2.1. Inicjatywa INSPIRE	12
2.1.1. Cele i zadania INSPIRE	14
2.1.2. Podstawowe pojęcia dotyczące INSPIRE	14
2.1.3. Sytuacja w okresie poprzedzającym	16
2.1.4. Koncepcja i model pojęciowy ESDI	17
2.2. Główne problemy metodyczne i technologiczne infrastruktury geoinformacyjnej	19
2.2.1. Rozwój systemów geoinformacyjnych	20
2.2.2. Interoperacyjność systemów jako podstawa infrastruktury	21
2.2.3. Interdyscyplinarność i wielopoziomowość zagadnień geomatyki	21
2.2.4. Ontologia, semantyka i obiektowość geoinformacji	23
2.2.5. Problemy geomatyki specyficzne dla poszczególnych dyscyplin	29
3. Modele pojęciowe danych, usług i interfejsów	35
3.1. Rola standardów w projektowaniu i budowie infrastruktury	35
3.2. Podstawowe pojęcia – struktura danych, interfejs i usługa	36
3.3. Modele pojęciowe dotyczące geoinformacji	38
3.3.1. Język UML i jego profil dla geomatyki	38
3.3.2. Programy narzędziowe dla UML (Rational Rose)	40
3.3.3. Modele abstrakcyjne i implementacyjne	42
3.3.4. Konwersja modeli abstrakcyjnych do modeli implementacyjnych	47
3.3.5. Modele ogólne i aplikacyjne (dziedzinowe lub tematyczne)	49
3.3.6. Stopień złożoności modeli i harmonizacja diagramów	50
3.4. Zapis modelu UML z zastosowaniem języka XML	51
3.4.1. XMI – XML dla wymiany metadanych o modelach pojęciowych	51
3.4.2. Program narzędziowy HyperModel	52
3.5. Technologie komponentowe w geomatyce	52
3.6. Rola języka XML w interoperacyjności infrastruktury geoinformacyjnej	54
4. Mapy w sieci WWW (<i>WebMapping</i>)	55
4.1. Podstawy technologiczne	56
4.2. Standard OpenGIS-WMS: interfejs i protokół (HTTP-GET)	57
4.3. Trzy podstawowe tryby komunikacji	58
4.4. Serwery kaskadowe	59
4.5. Rozbudowane przeglądarki map	61
4.6. Przykłady serwerów zgodnych z WMS	62
4.6.1. Minnesota WebMapServer	63
4.6.2. Polska aplikacja serwera Minnesota – Telkonet	66
4.6.3. Deegree WebMapServer	67
4.6.4. Oprogramowanie firmy Cubewerx	69
4.6.5. Oprogramowanie firmy Ionic Software	70

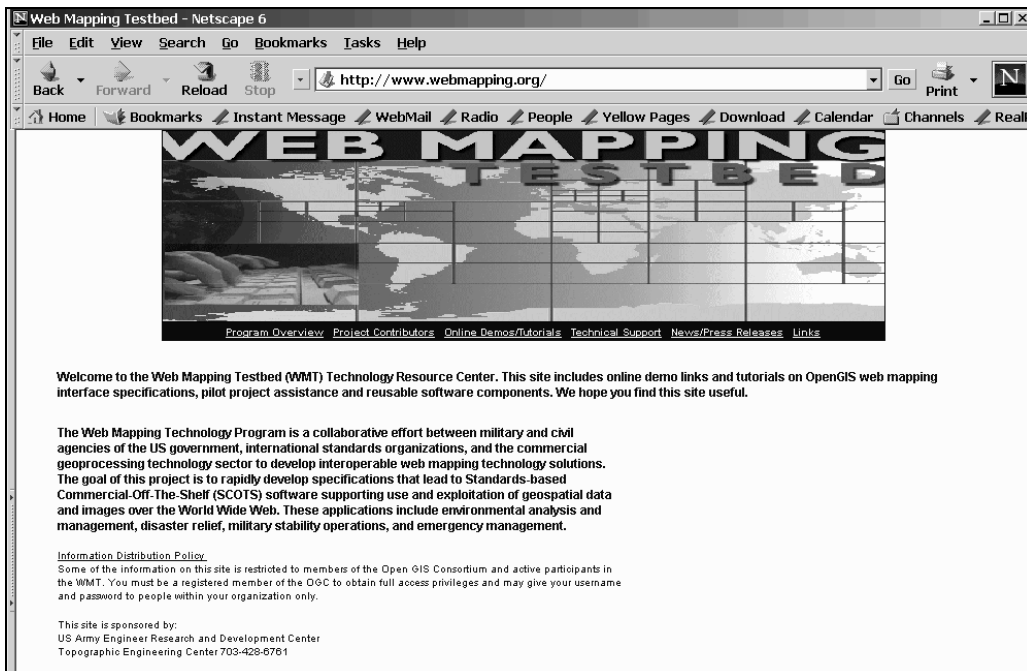
5. Język GML (<i>Geography Markup Language</i>)	73
5.1. Podstawy języka XML	73
5.2. Oprogramowanie narzędziowe XML Spy	78
5.2.1. Diagramy XML Spy	79
5.3. GML jako aplikacja XML dla geoinformacji	81
5.4. MasterMap jako przykład zastosowania GML	83
5.4.1. Projekt systemu obsługi MasterMap	92
5.5. Deegree GML Viewer/Converter	95
5.6. Lista oprogramowania implementującego GML	96
5.7. Reguły opracowywania aplikacji GML	96
5.7.1. Konwersja modeli aplikacyjnych UML do GML 3	101
5.8. Transformowanie dokumentów GML do innych języków XML	101
5.9. Zobrazowanie geoinformacji zapisanej w GML	102
6. Rozwijane i planowane technologie geoinformacyjne	103
6.1. Integracja usług geoinformacyjnych	103
6.2. CICE – środowisko współdziałania w sytuacjach krytycznych	105
6.2.1. Lista projektów specyfikacji usług w ramach CICE	106
6.2.2. Problemy technologiczne integracji usług geoinformacyjnych	107
6.2.3. Przykłady rozwiązań – GNS serwer nazw geograficznych	111
6.3. Systemy programowe OpenSource dla geoinformacji	113
6.3.1. OpenMap firmy BBN	114
6.3.2. Deegree – Uniwersytet w Bonn	115
6.4. Harmonizacja i konwersja do XML modeli standardu ISO 19100	118
6.4.1. Projekt NIMA dotyczący standardu ISO 19115 – Metadane	119
6.4.2. Projekty Grupy Nordyckiej	125
6.5. Technologie gridowe	126
6.5.1. MeteoGRID – zastosowanie UNICORE do geoinformacji	127
6.5.2. Przykłady zastosowania DataGRID do geoinformacji	128
Słownik terminów używanych w tekście	131
Literatura	137

4. MAPY W SIECI WWW (*WebMapping*)

Usługa udostępniania map w internecie przy pomocy protokołu http (utożsamianego z WWW) stosując polecenie GET była pierwszą usługą, dla której w OGC opracowano specyfikację implementacyjną i w zasadzie można uznać, że ten problem jest już rozwiązany, jeżeli ograniczy się go tylko do możliwości, jakie daje ten protokół. Cechą charakterystyczną tej usługi jest to, że wynik otrzymywany przez klienta jest zawsze obrazem rastrowym możliwym do wyświetlenia przez zwykłą przeglądarkę WWW. Z powyższych powodów portal tego projektu przedstawiony na rysunku 38 ma już raczej znaczenie historyczne. Trzeba tu dodać, że obecnie dla rozszerzenia jego możliwości powstała nowa specyfikacja wykorzystująca do komunikacji także polecenie POST.

Koncepcja usługi WebMapping nie jest nowa, lecz dawniejsze rozwiązania (niestandardowe) zakładały, że jeden klient łączy się z jednym serwerem bez standardowych rozwiązań umożliwiających interoperacyjność.

Obecne prace dotyczące tych zagadnień koncentrują się na przesyłaniu do klienta czegoś więcej niż tylko rastrowy obraz mapy – zastosowanie języka XML pozwala na przesyłanie obrazów w formie wektorowej (SVG) lub danych nie zobrazowanych przy pomocy GML. Jednak w takich przypadkach nie wystarczy sama przeglądarka WWW – potrzebne jest po stronie klienta dodatkowe oprogramowanie często dość skomplikowane.



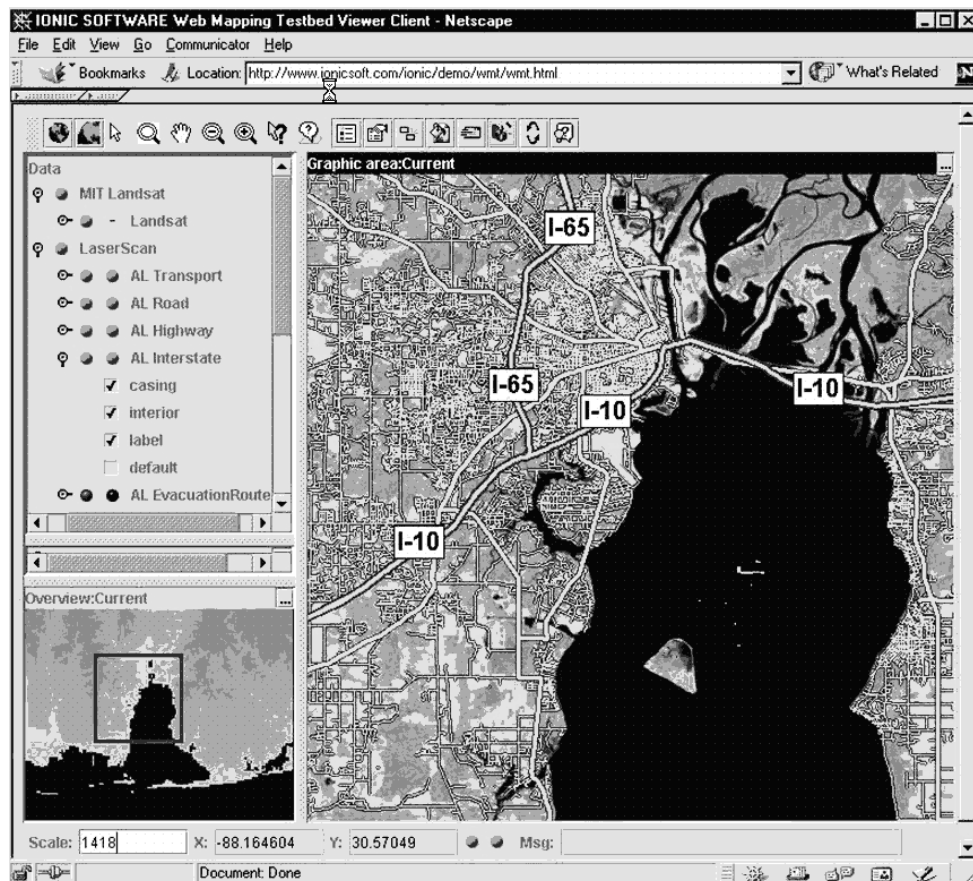
Rys. 38. Główny portal programu testowania technologii OpenGIS dla udostępniania map w WWW – WMT (Web Mapping Testbed). [Źródło: <http://www.webmapping.org>]

Innym obecnie rozwiązywanym problemem jest powiązanie usługi WebMapping z innymi nowymi usługami, które są przedstawione bardziej szczegółowo w rozdziale 6. Do tych usług należą między innymi: usługa udostępniania wyróżnień (WFS) i usługa udostępniania pokryć (WCS).

Rozdział niniejszy przedstawia, obok podstaw technologicznych tej usługi, także najczęściej stosowane oprogramowanie ze szczególnym uwzględnieniem rozwiązań niekomercyjnych, a także szereg przykładów zastosowań.

4.1. Podstawy technologiczne

Protokół HTTP (*HyperText Transfer Protocol*) będący podstawą WWW i przez to także usługi WebMapping pozwala klientowi (przeglądarce WWW) wysłać komunikat do serwera zawierający zlecenia. Po stronie serwera treść tego komunikatu jest kierowana do programu



Rys. 39. Jeden z pierwszych eksperymentów w zakresie interoperacyjnego WebMapingu – dane geoprzestrzenne z różnych serwerów są jednocześnie wyświetlane w oknie przeglądarki.
[Źródło: <http://www.laser-scan.com>]

obsługującego WebMapping, następnie jest przez ten program interpretowana i zawarte w nim zlecenie zostaje spełnione pod warunkiem, że jest poprawne i możliwe do zrealizowania. Specyfikacja serwera (WMS) określa dopuszczalną treść i format komunikatów – przykłady ich są przedstawione w dalszej części tego rozdziału (przykłady 4, 5 i 6). Znając obowiązujące tu zasady komunikat taki można wypisać „ręcznie” w pasku adresowym przeglądarki lub przekopiować go z edytora do tego paska, a następnie wysłać do serwera. Jest to jednak trudne i łatwo jest popełnić błąd – w rezultacie wysłany komunikat może być niepoprawny i serwer nie odpowie lub również przyśle odpowiedź niepoprawną. Z tego względu w usłudze WebMapping przeglądarka jest wspomagana przynajmniej przez mały program w języku Java (applet), który pomaga nam w sformułowaniu komunikatu poprzez umożliwienie wybierania dostępnych opcji przy pomocy widocznych na ekranie przycisków.

Rysunek 39 przedstawia jeden z pierwszych eksperymentów w zakresie WebMappingu, w którym na ekranie jednego klienta są wyświetlone dane z kilku serwerów.

4.2. Standard OpenGIS-WMS: interfejs i protokół (HTTP-GET)

Specyfikacja implementacyjna WMS wersja 1.1.1 określa szczegółowo sposób komunikowania się klienta z serwerem (lub kilkoma serwerami) w ramach protokołu HTTP przy pomocy komunikatów typu GET i POST. Pierwszy z nich jest powszechnie stosowany i z tego względu stanowi podstawę WMS. Obecnie opracowywane rozszerzenia WMS wykorzystujące SLD (*Styled Layer Descriptor*) i w tym przypadku stosuje się także typ POST.

W ramach typu GET specyfikacja WMS określa trzy podstawowe komunikaty (zlecenia):

- GetCapabilities – w odpowiedzi serwer podaje swoje możliwości i zasoby;
- GetMap – serwer przesyła obraz rastrowy zbudowany z wybranych warstw, w wybranej skali, w wybranym odwzorowaniu i dla wybranego obszaru;
- GetFeatureInfo – w odpowiedzi przesyłane są informacje nieprzestrzenne (opisowe, tematyczne).

W przypadku wersji rozszerzonej z SLD są używane także komunikaty związane ze sposobem zobrazowania geoinformacji: DescribeLayer, GetLegendGraphic, GetStyles i PutStyles, ale te komunikaty są przedmiotem innej specyfikacji.

Składnię komunikatu GetMap ilustrują przykłady 4 i 10, a listę parametrów, jakie muszą i mogą w tym komunikacie wystąpić zawiera tabela 2.

Przykład 4. (teoretyczny)

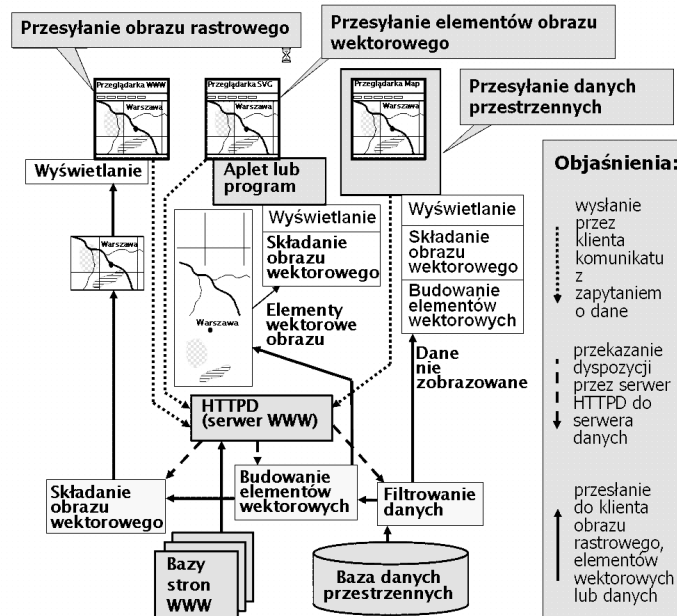
```
http://a-map-co.com/mapserver.cgi?VERSION=1.1.0&REQUEST=GetMap&
SRS=EPSG:4326&BBOX=-97.105,24.913,78.794,36.358&
WIDTH=560&HEIGHT=350&LAYERS=AVHRR-09-27&STYLES=&
FORMAT=image/png&BGCOLOR=0xFFFFFFFF&TRANSPARENT=TRUE&
EXCEPTIONS=application/vnd.ogc.se_inimage
```

**Tabela 2. Zestawienie parametrów komunikatu GetMap w według specyfikacji OpenGIS-WMS v. 1.1.1.
[Źródło: (Kolodziej, 2003)]**

Parametr komunikatu wymagany = R opcjonalny = O		Opis
VERSION=version	R	Wersja komunikatu
REQUEST=GetMap	R	Nazwa komunikatu
LAYERS=layer_list	R	Oddzielona przecinkami lista zawierająca jedną lub wiele warstw mapy. Gdy parametr SLD jest obecny to ten parametr jest opcjonalny.
STYLES=style_list	R	Oddzielona przecinkami lista stylów interpretacji (zobrazowania) – jeden styl dla każdej żądanej warstwy. Gdy parametr SLD jest obecny to ten parametr jest opcjonalny.
SRS=namespace:identifier	R	Układ odniesienia przestrzennego.
BBOX=minx,miny,maxx,maxy	R	Współrzędne prostokąta ograniczającego w jednostkach układu odniesienia (boki: dolny, lewy, górny i prawy).
WIDTH=output_width	R	Szerokość wynikowego obrazu rastrowego w pikselach.
HEIGHT=output_height	R	Wysokość wynikowego obrazu rastrowego w pikselach.
FORMAT=output_format	R	Format wynikowego obrazu rastrowego.
TRANSPARENT=TRUE/FALSE	O	Przezroczystość mapy wobec tła (domyślnie = FALSE).
BGCOLOR=color_value	O	Heksadecymalna wartość koloru dla tła (w układzie RGB, domyślnie = 0xFFFFFF).
EXCEPTIONS=exception_format	O	Format zgłaszania wyjątków przez serwer WMS (domyślnie = SE_XML).
TIME=time	O	Współrzędna czasowa żądanej warstwy.
ELEVATION=elevation	O	Współrzędna wysokościowa żądanej warstwy.
Other sample dimension(s)	O	Wartości ewentualnych innych współrzędnych.
Vendor-specific parameters	O	Dodatkowe parametry eksperymentalne.
Poniższe parametry są używane tylko dla usług WebMapping wykorzystujących specyfikację deskryptora warstwy stylów (SLD – Styled Layer Descriptor).		
SLD=styled-layer_descriptor_URL	O	URL określający deskryptor warstwy stylów (zgodnie z definicją specyfikacji SLD).
WFS=web_feature_service_URL	O	URL określający usługę udostępniania wyróżnień dla zobrazowania przy pomocy SLD.

4.3. Trzy podstawowe tryby komunikacji

Specyfikacja implementacyjna WMS dotyczy tylko przypadku zwracania przez serwer wyniku w formie obrazu rastrowego (GIF, JPG lub PNG). Przyszłość usługi WebMapping należy jednak upatrywać w obrazach wektorowych z zastosowaniem języka SVG, a także przesyłania map w postaci niezobrazowanych danych przy pomocy języka GML. Te trzy sposoby są przedstawione na rysunku 40.



Rys. 40. Schematy przesyłania geoinformacji w internecie pomiędzy serwerem (systemem udostępniającym dane) i klientem (systemem poszukującym danych i odbierającym wyniki).
[Na podstawie: materiałów archiwum OGC]

Zastosowanie SVG i GML w WebMappingu jest obecnie na etapie eksperymentów, ale wstępna specyfikacja dla tych technologii już istnieje. Rysunek 41 przedstawia obraz uzyskany z takiego serwera.

4.4. Serwery kaskadowe

Ponieważ zwykła przeglądarka w danym momencie może łączyć się tylko z jednym serwerem WWW, dla wyświetlenia obrazów z kilku serwerów przy użyciu takiej przeglądarki bez dodatkowego oprogramowania potrzebny jest serwer kaskadowy, który występuje dla tej przeglądarki w roli serwera i jednocześnie łącząc się z innymi serwerami źródłowymi, wobec nich występuje w roli klienta.

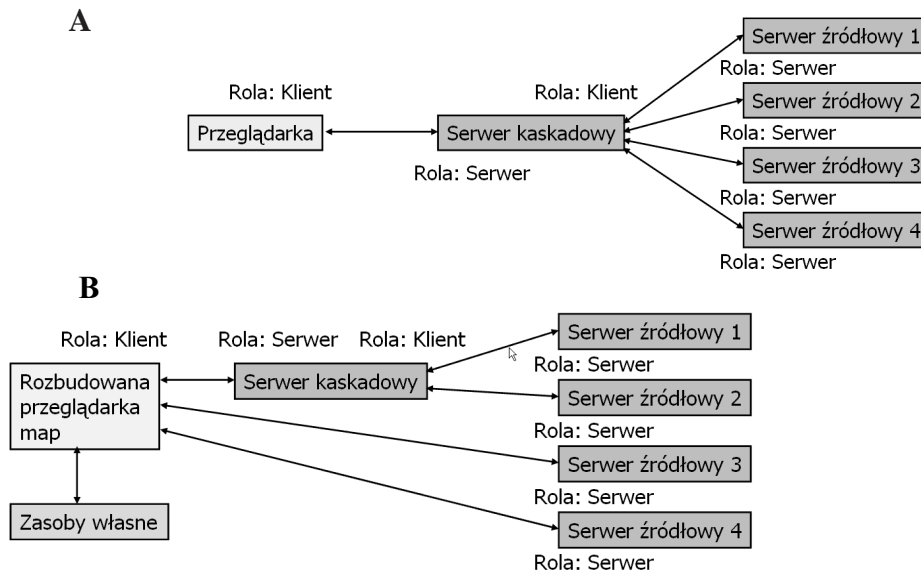
Dodatkowymi funkcjami serwera kaskadowego mogą być:

- przeliczanie współrzędnych (zmiana układu odniesienia),
- zobrazowanie danych geoprzestrzennych (budowa obrazu rastrowego mapy),
- serwer nazw geograficznych (zamiana nazwy na współrzędne),
- usługi dostępu i opłat, a także inne usługi.

Schemat przedstawiający rolę serwera kaskadowego zawiera rysunek 42. Serwer ten może współpracować zarówno ze zwykłą (A) jak i z rozbudowaną (B) przeglądarką.



Rys. 41. Przykład mapy zapisanej przy pomocy języka SVG po dokonaniu konwersji z języka GML. [Źródło: <http://demo.ionicsoft.com>]



Rys. 42. Serwer kaskadowy wobec innych serwerów jest klientem, a wobec klienta jest serwerem.

4.5. Rozbudowane przeglądarki map

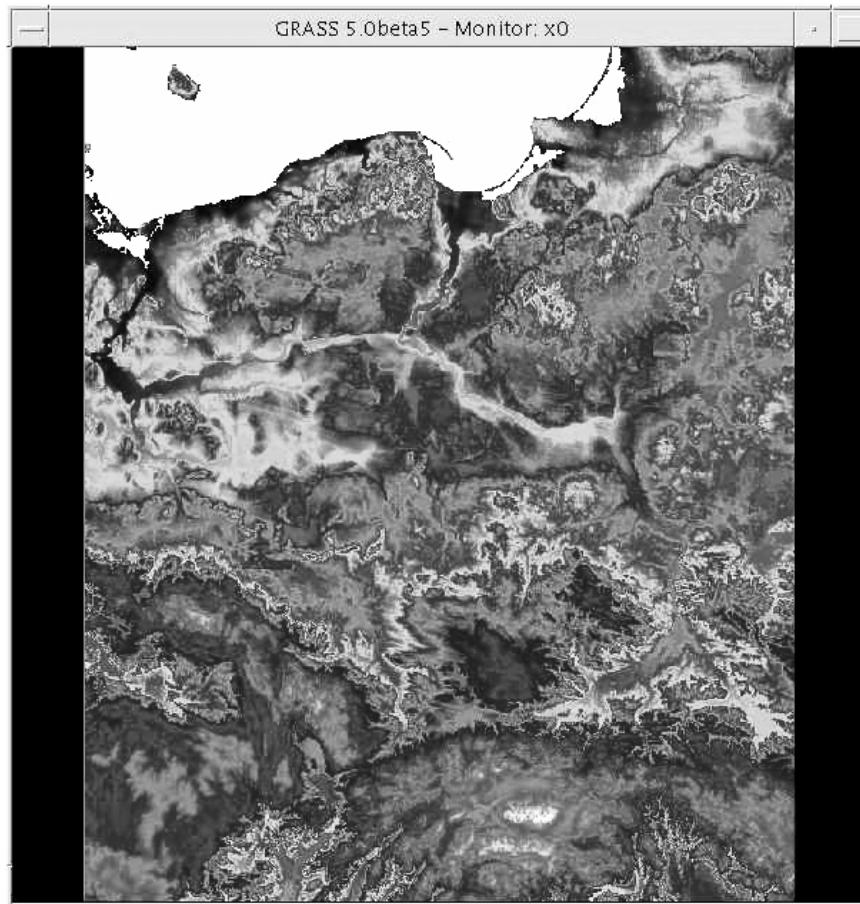
Bardziej zaawansowane sposoby udostępniania map przez WWW wymagają zastosowania po stronie klienta specjalnego oprogramowania, którego możliwości w zakresie budowy obrazów map są znacznie większe niż zwykłej przeglądarki WWW. Obecnie jest dostępnych wiele programów mogących pełnić takie funkcje – są to programy zarówno komercyjne jak i bezpłatne lub objęte licencją OpenSource. Możliwości tych programów są różne, ale najczęściej spełniają one podstawowe wymagania:

- może jednocześnie łączyć się z wieloma serwerami,
- może korzystać z własnej bazy danych,
- może składać obraz z elementów lub tworzyć te elementy z danych,
- także może korzystać z usług serwera kaskadowego dla innych operacji.

Przykłady niekomercyjnego oprogramowania dla rozbudowanych przeglądarek map (o różnych możliwościach):

- OpenMap (biblioteka klas Java i różne aplikacje)
- Deegree (także biblioteka klas Java i różne aplikacje – w tym serwery)

Oprogramowanie to jest bardziej szczegółowo przedstawione w rozdziale 6. Stanowi ono dobrą podstawę do różnych eksperymentów w zakresie WebMappingu, zarówno w aspekcie technologicznym jak i aplikacyjnym. Jednak koniecznym warunkiem powodzenia tych eksperymentów jest odpowiedni zestaw danych testowych o dużej różnorodności i dostatecznej dokładności. Przykład jednego ze składników takiego zestawu dla obszaru Polski przedstawia rysunek 43.



Rys. 43. DEM dla obszaru Polski jako dane testowe dla eksperymentów w zakresie WebMapping.

4.6. Przykłady serwerów zgodnych z WMS

Lista oprogramowania zgodnego ze specyfikacją implementacyjną WMS (v. 1.1.0 i 1.1.1) zawiera ponad 90 pozycji, w tym 58 to serwery a 42 to oprogramowanie klientów. Aktualną pełną listę można znaleźć pod adresem <http://www.opengis.org/testing/product/index.php>. Czołowe firmy w tym zakresie to: Cadcorp Ltd., Compusult Ltd., CubeWerx Inc., ESRI, Social Change Online Pty Ltd. i SICAD Geomatics GmbH. Wiodącymi ośrodkami niekomercyjnymi są: Uniwersytet Minnesota, Uniwersytet Bonn i MIT.

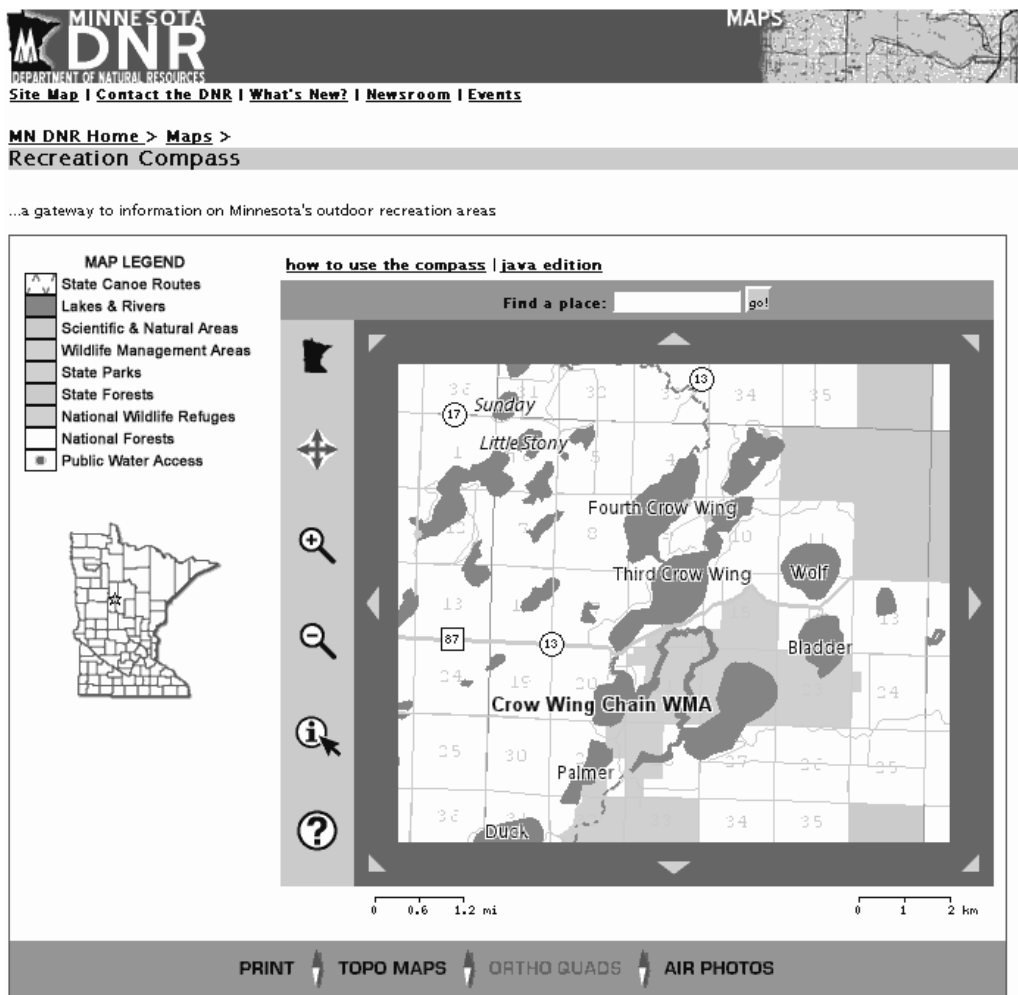
W rozdziale tym przedstawione są przykłady aplikacji serwerów: Minnesota WebMapServer (a w tym polska aplikacja firmy Telkonet), Deegree WebMapServer i oprogramowania komercyjnego firm Ionic Software i Cubewerx.

4.6.1. Minnesota WebMapServer

Niekomercyjne oprogramowanie dla serwera map opracowane w Uniwersytecie Minnesota jest zgodne ze standardem OGC – WMS v. 1.1.0. Oprogramowanie to ma bardzo wiele aplikacji w różnych dziedzinach i z tego względu jest dobrym przykładem możliwości, jakie daje ta technologia. Rysunek 44 przedstawia okno przeglądarki współpracującej z jego aplikacją na Wydziale Zasobów Naturalnych (DNR) tego uniwersytetu. Obraz ten został uzyskany po wysłaniu z przeglądarki komunikatu zawartego w przykładzie 5.

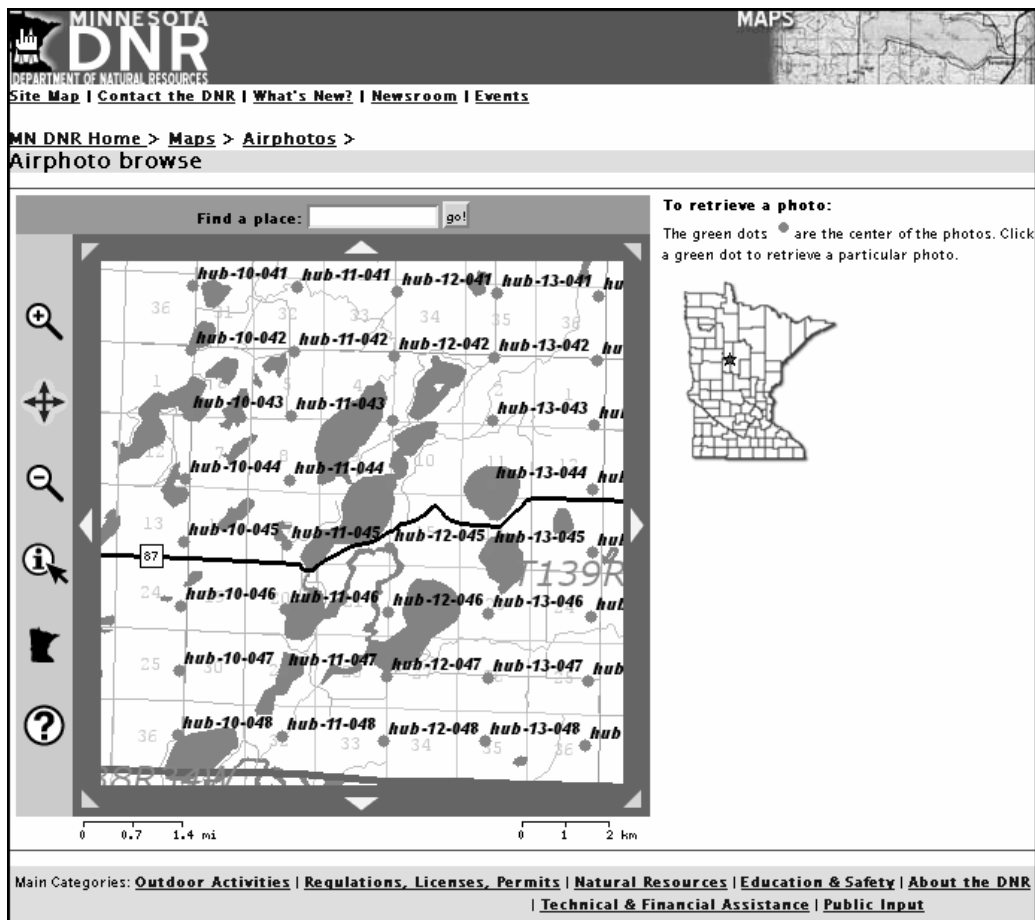
Przykład 5.

```
http://www.dnr.state.mn.us/maps/compass.html?map=COMPASS_MAPFILE&imgext=59874.874702+4793122.838717+853624.446078+5481039.133908&mode=browse&zoomdir=1&zoomsize=3&layers=&imgxy=187.0+162.0&imgbox=-1+-1+-1+-1&img.x=169&img.y=162
```



Rys. 44. Przykład zastosowania oprogramowania Web Map Server Minnesota na Wydziale Zasobów Naturalnych (DNR) tego uniwersytetu. [Źródło: <http://www.dnr.state.mn.us>]

Aplikacja ta ma także możliwość udostępniania zdjęć lotniczych i wybór zdjęcia jest dokonywany z wykorzystaniem WMS. Aby uzyskać zdjęcie dla wybranego fragmentu obszaru przedstawionego na rysunku 45, trzeba wysłać komunikat z przykładu 6. Rezultat tego polecenia przedstawia rysunek 46.



Rys. 45. Fragment okna przeglądarki podczas wybierania zdjęcia lotniczego. W tym trybie pracy można wysłać komunikat przedstawiony w przykładzie 6. [Źródło: <http://www.dnr.state.mn.us>]

Przykład 6.

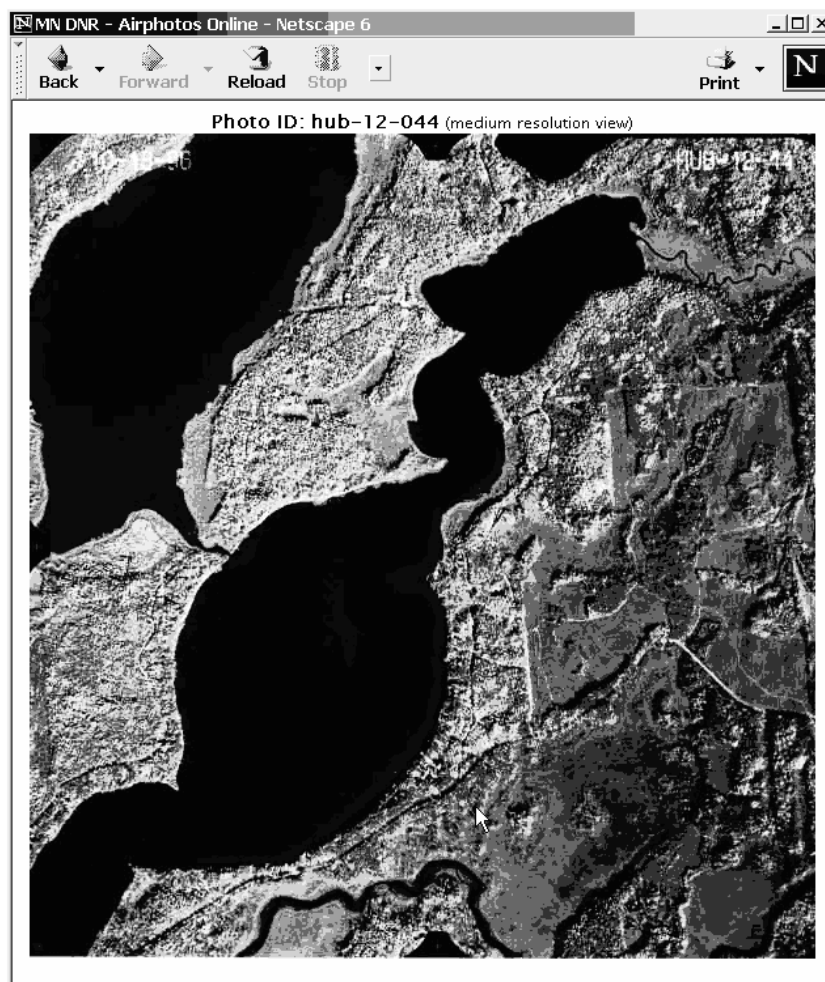
```
http://www.dnr.state.mn.us/airphotos/browse.html?map=photos/etc/photos.map&mapext=
352313.143907+5185743.584010+364219.387484+5196062.328441&zoomsize=3&
zoomdir=0
```


Ten sam komunikat podzielony na logiczne fragmenty:

Przykład 7.

```
http://www.dnr.state.mn.us/airphotos/browse.html?
map=photos/etc/photos.map&
mapext=      352313.143907
              +5185743.584010
              +364219.387484
              +5196062.328441&
zoomsize=3&
zoomdir=0
```

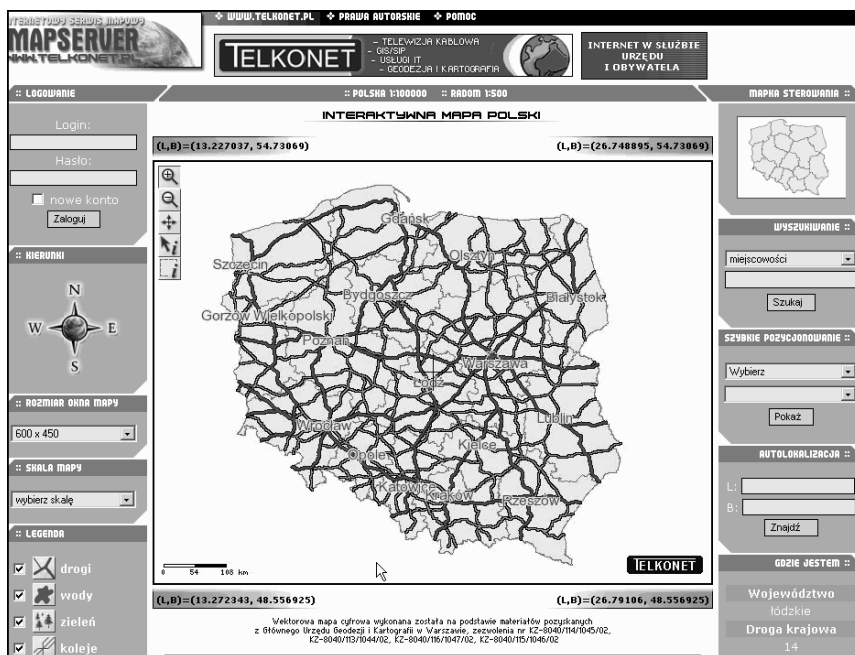
W rezultacie wysłania tego komunikatu określającego zdjęcie, serwer przesłał do przeglądarki wynik przedstawiony na rysunku 46.



Rys. 46. Zdjęcie lotnicze w średniej rozdzielczości przesłane przez serwer standardu WMS do współpracującej z nim przeglądarki. [Źródło: <http://www.dnr.state.mn.us>]

4.6.2. Polska aplikacja serwera Minnesota – Telkonet

Przykładem dla obszaru Polski jest aplikacja serwera Minnesota zrealizowana przez firmę Telkonet z Radomia (rys. 47). Po stronie klienta jest instalowany automatycznie applet Rosa w języku Java pozwalający zwykłej przeglądarce wysyłać komunikaty zawierające zlecenia klienta.



Rys. 47. Okno przeglądarki podczas sesji komunikacji z serwerem map dla obszaru Polski prowadzonym przez firmę Telkonet.

Komunikat wysłany z przeglądarki przez applet Rosa do serwera Telkonet (podzielony na fragmenty) przedstawia przykład 8.

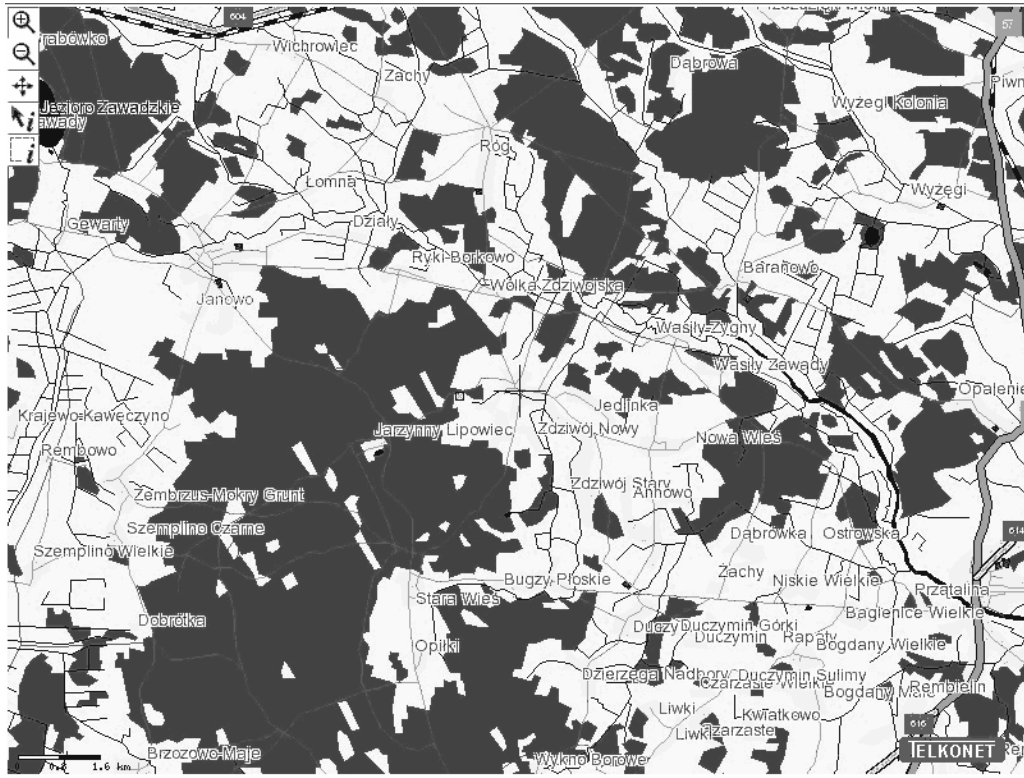
Przykład 8.

```

http://mapserver.telkonet.pl/?
PHPSESSID=579100545d47faa4f14a96c3c80a68da
&r_maptype=1
&r_password=
&MapSize=600%2C450
&grp_drogi=Y
&grp_zielen=Y
&grp_cmentarze=Y
&grp_granice=Y
&minx=480799.427495
&maxx=487813.142449
&imagewidth=600
&CMD=
&INPUT_COORD=
&KEYMAPXSIZE=119
&combo_list_to_find=1
&map_ext_meid=
&wspy=
&r_login=
&runningmapserver=yes
&skala=7
&grp_wody=Y
&grp_koleje=Y
&grp_miasta=Y
&PREVIOUS_MODE=0
&miny=5902305.379142
&maxy=5907556.790867
&imageheight=450
&INPUT_TYPE=
&theurl2=
&KEYMAPYSIZE=90
&findobiekt= &map_ext_drvid=-1
&wspix=

```


Po wysłaniu tego komunikatu serwer odpowiada obrazem rastrowym przedstawionym na rysunku 48.



Rys. 48. Obraz na przeglądarce jako wynik wykonania polecenia przez serwer (fragment mapy topograficznej w skali 1 : 100 000 z okolic Chorzel) [Źródło: <http://mapserver.telkonet.pl>]

4.6.3. Deegree WebMapServer

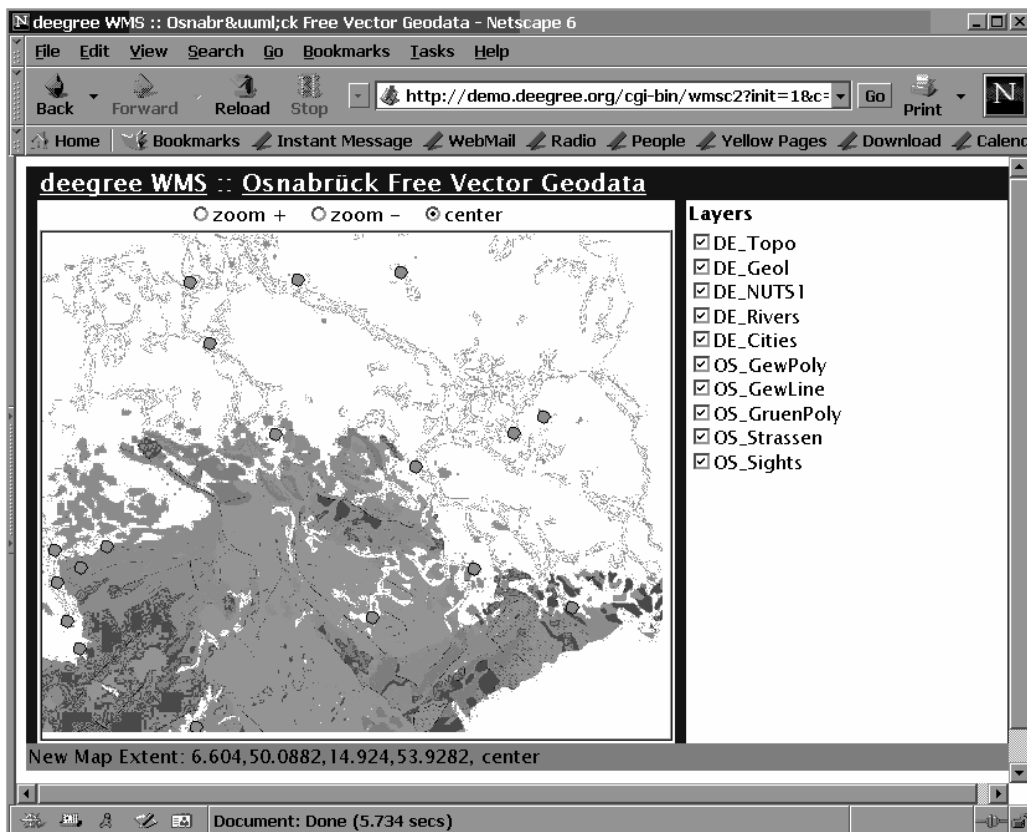
Serwer WMS Deegree jest jednym z komponentów pakietu oprogramowania zgodnego z wieloma specyfikacjami implementacyjnymi usług OpenGIS. Z tego względu zasługuje ono na szczególną uwagę, tym bardziej, że jest objęte licencją OpenSource, czyli może być wykorzystywane bezpłatnie. Postać komunikatów odbieranych przez ten serwer pokazują przykłady 9 i 10. Porównanie komunikatów wysyłanych do serwerów WMS przedstawionych w tym rozdziale (przykłady od 5 do 10) z tabelą 2 pozwala zauważyć, że tylko komunikat z przykładu 10 spełnia wymagania określone w tej tabeli. Na tej podstawie można przypuszczać, że zdecydowana większość serwerów map dostępnych w internecie nie spełnia wymagań specyfikacji OpenGIS, chociaż oprogramowanie to ma certyfikat zgodności. Jedynym wytłumaczeniem tej sytuacji może być to, że są to starsze wersje oprogramowania, a certyfikaty dotyczą nowych wersji.

Przykład 9.

```
http://demo.deegree.org/cgi-bin/wmsc2?init=1&c=& bBox=7.93%2C52.22%2C8.19%2C52.34&
mapRequest=zoom+%2B&
on=DE_Topo&on=DE_Geol&on=OS_Strassen&x=254&y=180
```

Przykład 10

```
http://127.0.0.1:8080/deegreewms/wms?SERVICE=WMS&
Request=GetMap&
LAYERS=physical,europe:country&
format=png&
bbox=-
20.0,35.0,20.0,90.0&
styles=default&
VERSION=1.1.1&
Height=300&
Width=400&
SRS=EPSG:4326
```



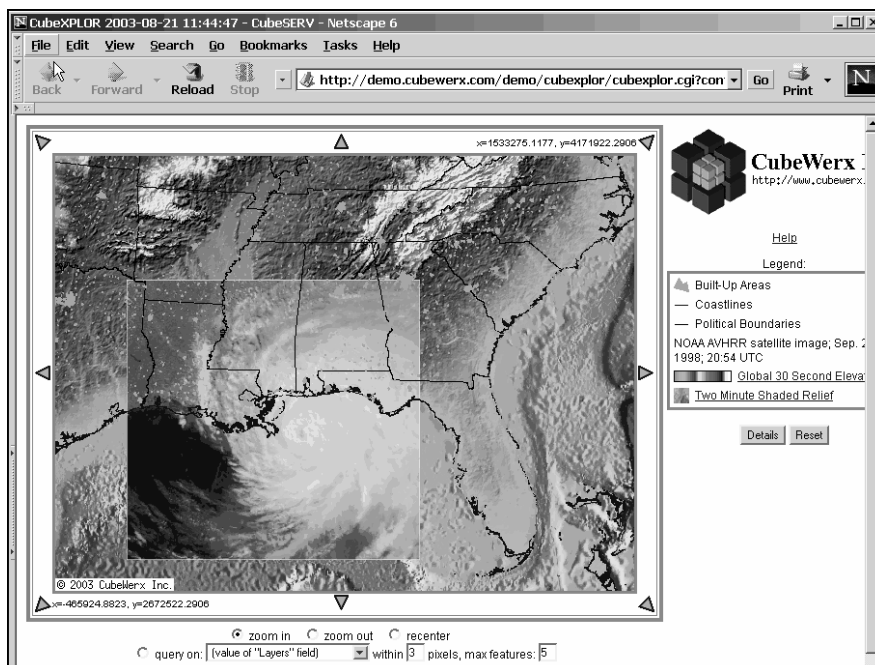
Rys. 49. Okno przeglądarki przedstawiające mapę geologiczną przesłaną z serwera Deegree.

4.6.4. Oprogramowanie firmy Cubewerx

Firma Cubewerx zyskała sobie wysoką pozycję w rozwoju usług typu WebMapping dzięki pionierskim rozwiązaniom w zakresie serwerów kaskadowych. Koncepcja tych serwerów została wyjaśniona w rozdziale 4.4. Tu przedstawione są przykłady funkcjonowania oprogramowania tej firmy. Wysłanie do serwera Cubewerx przez przeglądarkę komunikatu z przykładu 11 daje w wyniku obraz przedstawiony na rysunku 50.

Przykład 11

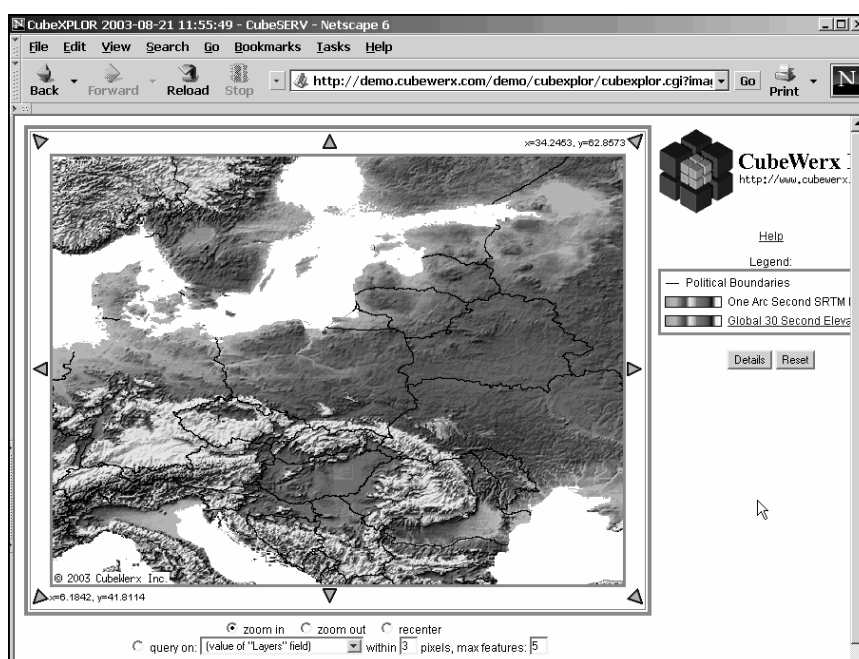
```
http://demo.cubewerx.com/demo/cubexplor/cubexplor.cgi?imageClick.x=291&
imageClick.y=220&clickMode=zoom+in&
layerToQuery=%28value+of+%22Layers%22+field%29&pixelSensitivity=3&
featureCount=5&dataStores=Foundation&themes=Boundaries&
layers=Barrier+Line+Features&styles=%28default%29&
newScale=11.9M&newX=533675.11768&newY=3422222.2906&newImageWidth=600&
newImageHeight=450&newImageType=image%2Fpng&newQuality=MEDIUM&
zoomFactor=2&newSrs=42001+%28WGS+84+%2F+Auto+UTM%29&sldUrl=&
layersToPlot=Built-Up+Areas%2CCoastlines%2CPolitical+Boundaries%2CNOAA
+AVHRR+satellite+image%3B+Sep.+27%3B+1998%3B+20%3A54+UTC%2CGlobal
+30+Second+Elevations%2CTwo+Minute+Shaded+Relief&currentY=3422222.2906&
currentX=533675.117680001&currentScale=11.9M&
currentImageWidth=600&currentImageHeight=450&
currentSrs=AUTO%3A42001%2C9001%2C-86.6475031300162%2C30.93293099497698&
currentImageType=image%2Fpng&currentQuality=MEDIUM&CONFIG=george
```



Rys. 50. Różne dane geoprzestrzenne z różnych źródeł złożone w jeden obraz rastrowy. [Źródło: <http://demo.cubewerx.com>]

W tym przypadku również można zauważyć duże rozbieżności pomiędzy listą parametrów z tabeli 2 a parametrami tego komunikatu. Jedną z przyczyn tych rozbieżności jest prawdopodobnie użycie wielu parametrów typu „vendor-specific parameters”, których używanie dopuszcza specyfikacja WMS, lecz które sprawiają, że serwer musi przesłać do klienta dodatkowe oprogramowanie interpretujące niestandardowe parametry. W rezultacie interoperacyjność takiego rozwiązania zostaje zawężona.

Obecne zasoby publicznie dostępnych eksperymentalnych serwerów map są już bardzo duże i można uzyskać z nich wiele pożytecznych danych geoprzestrzennych. Rysunek 51 przedstawia dane z obszaru Europy Centralnej udostępniane przez serwer Cubewerx.

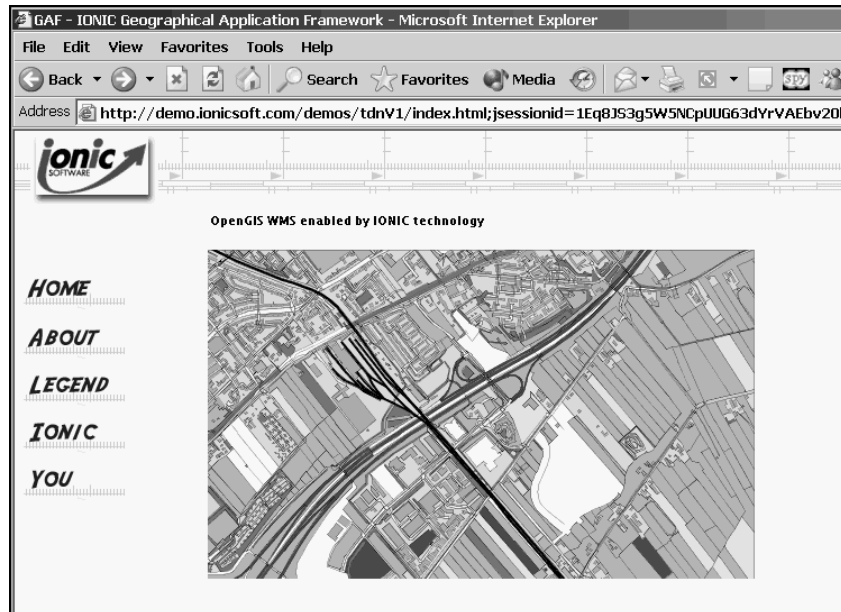


Rys. 51. Przykład danymi z obszaru Europy dostępnymi w serwisie map prowadzonym przez Cubewerx. [Źródło: <http://demo.cubewerx.com>]

4.6.5. Oprogramowanie firmy Ionic Software

Belgijska firma Ionic Software jeszcze bardzo niedawno składała się z czterech osób, a dziś jest jedną z czołowych firm w zakresie oprogramowania serwerów usług geoinformacyjnych. Pozycję tą zawdzięcza udanym eksperymentom w zakresie zaawansowanych rozwiązań opartych na XML, a w szczególności stosujących grafikę wektorową w SVG i przesyłanie danych w języku GML. Rysunek 52 zawiera obraz mapy uzyskany z zapisu wektorowego.

W rozwiązaniach technologicznych stosowanych przez firmę Ionic Software oprogramowanie po stronie klienta wysyła komunikaty do serwera samodzielnie – nie korzystając z pośrednictwa przeglądarki WWW. Z tego powodu w pasku adresowym przeglądarki nie jest umieszczany tekst komunikatu, a jedynie adres serwera map, jak to pokazuje przykład 12.



Rys. 52. Wektorowy obraz mapy zapisany w języku SVG, w przeciwieństwie do obrazu rastrowego, może być powiększany wielokrotnie i jedynym ograniczeniem jest dokładność współrzędnych.
[Źródło: <http://demo.ionicsoft.com>]

Przykład 12.

<http://demo.ionicsoft.com/gisd-c/Mainframe.jsp>

Tekst komunikatu można zobaczyć, gdy korzysta się z bardziej zaawansowanych rozwiązań, na przykład po wysłaniu zlecenia, aby otrzymać pełne dane geoprzestrzenne dotyczące wybranego wyróżnienia zapisane w GML. Tekst takiego komunikatu zawiera przykład 13, a otrzymany wynik przedstawia przykład 14.

Przykład 13.

```
http://mapserv2.esrin.esa.it/ionicweb/wfs/ATSRFIRE?&REQUEST=map&
SRS=EPSG%3A4326&
BBOX=20.67320760434113,20.490283444953547,41.20679239565887,32.929716555046454&
WIDTH=723&HEIGHT=438&LAYERS=ESA_FIRE&STYLES=default&
FORMAT=GML.2&
BGCOLOR=0xffff&TRANSPARENT=FALSE&
EXCEPTIONS=INIMAGE&
FILTER=%3CFilter%3E%3CAnd%3E%3CPropertyIsGreaterThanOrEqualTo%3E%3CPropertyNam
e%3EDATE%3C/PropertyName%3E%3CLiteral%3E1999-01-
01%3C/Literal%3E%3C/PropertyIsGreaterThanOrEqualTo
%3E%3CPropertyIsLessThanOrEqualTo%3E%3CPropertyName%3EDATE
%3C/PropertyName%3E%3CLiteral%3E1999-01-07%3C/Literal%3E
%3C/PropertyIsLessThanOrEqualTo%3E%3C/And%3E%3C/Filter%3E& USEBOX=TRUE
```

W tym przypadku łatwo jest zauważyć zgodność z tabelą 2 i zatem także ze specyfikacją OpenGIS. Jednak ten komunikat dotyczy usługi WFS (*Web Feature Server*), która jest objęta

inną specyfikację implementacyjną, chociaż opartą na wspólnych podstawach i tych samych regułach syntaktycznych. Jedyne zastrzeżenie, jakie można mieć do tego komunikatu, to wątpliwość dotycząca potrzeby użycia parametru TRANSPARENT w przypadku, gdy wynikiem jest zapis w języku GML – parametr: FORMAT=GML.2.

Przykład 14.

```
<?xml version="1.0" encoding="utf-8" ?>
<ogcwf:FeatureCollection
  scope="http://mapserv2.esrin.esa.it/ionicweb/wfs/ATSRFIRE?"
  xmlns:xlink="http://www.w3.org/1999/xlink"
  xmlns:ogcwf="http://www.opengis.net/wfs"
  xmlns:gml="http://www.opengis.net/gml"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:wfs="http://www.ionicssoft.com/wfs">
  <gml:boundedBy>
    <gml:Box srsName="EPSG:4326">
      <gml:coordinates>26.965,27.806 33.584,30.694</gml:coordinates>
    </gml:Box>
  </gml:boundedBy>
  <gml:featureMember>
    <wfs:ESA_FIRE fid="ESA_FIRE.323632363438">
      <wfs:NUMERO>262648</wfs:NUMERO>
      <wfs:DATE>1999-01-05</wfs:DATE>
      <wfs:HOURL>202547.375</wfs:HOURL>
      <wfs:NDVI>-.--</wfs:NDVI>
      <wfs:STATION>ESR</wfs:STATION>
      <wfs:LAT>30.694</wfs:LAT>
      <wfs:LONG>26.965</wfs:LONG>
      <wfs:Geometry>
        <gml:Point srsName="EPSG:4326">
          <gml:coordinates>26.965,30.694</gml:coordinates>
        </gml:Point>
      </wfs:Geometry>
    </wfs:ESA_FIRE>
  </gml:featureMember>
  <gml:featureMember>
    <wfs:ESA_FIRE fid="ESA_FIRE.323632373736">
      <wfs:NUMERO>262776</wfs:NUMERO>
      <wfs:DATE>1999-01-06</wfs:DATE>
      <wfs:HOURL>195328.061</wfs:HOURL>
      <wfs:NDVI>-.--</wfs:NDVI>
      <wfs:STATION>ESR</wfs:STATION>
      <wfs:LAT>27.806</wfs:LAT>
      <wfs:LONG>33.584</wfs:LONG>
      <wfs:Geometry>
        <gml:Point srsName="EPSG:4326">
          <gml:coordinates>33.584,27.806</gml:coordinates>
        </gml:Point>
      </wfs:Geometry>
    </wfs:ESA_FIRE>
  </gml:featureMember>
</ogcwf:FeatureCollection>
```