



INSTYTUT INŻYNIERII I GOSPODARKI WODNEJ

POLITECHNIKA KRAKOWSKA im. TADEUSZA KOŚCIUSZKI

---

Piotr Tokarz

UMN MAPSERVER JAKO PLATFORMA WEBGIS  
NA PRZYKŁADZIE RZGW W KRAKOWIE

praca magisterska

studia dzienne

kierunek studiów: **informatyka**

specjalność: **informatyka stosowana w inżynierii środowiska**

promotor: **dr inż. Robert Szczepanek**

nr pracy: **2076**

data złożenia: .....

KRAKÓW 2007

## PODZIĘKOWANIA

Chciałbym podziękować wszystkim, którzy wspierali mnie w trakcie pisania tej pracy oraz motywowali do działania, a szczególnie:

**Dr inż. Robertowi Szczepankowi**, za zainteresowanie mnie tematem systemów GIS, a w trakcie pisania pracy za cenne uwagi i poświęcony czas.

**Dyrekcji RZGW w Krakowie oraz Panu Radosławowi Radoniowi** za udostępnienie danych oraz przedstawienie wymagań projektowanego systemu.

1. Wprowadzenie .....	1
2. Systemy GIS .....	2
2.1 Historia systemów GIS .....	2
2.2 Współczesne Systemy GIS .....	4
3. Przegląd Systemów GIS .....	9
3.1 Systemy komercyjne .....	9
3.2 Systemy darmowe .....	13
4. UMN MapServer .....	15
4.1 Wprowadzenie do UMN MapServer .....	15
4.2 Najważniejsze funkcje i możliwości .....	16
4.3 Struktura aplikacji opartej o MapServer .....	18
4.4 Zawartość Map File .....	20
4.5 PHP Mapscript .....	28
4.6 Najciekawsze przykłady implementacji z galerii UMN Mapserver .....	29
5. Projekt aplikacji dla RZGW w Krakowie .....	37
5.1 Założenia do projektu .....	37
5.2 Dane wykorzystane w aplikacji .....	37
5.3 Środowisko implementacji .....	38
5.4 Funkcjonalność aplikacji .....	43
5.5 Zalety i wady aplikacji .....	53
6. Podsumowanie .....	54
Literatura .....	56
Spis rysunków .....	57

## 1. Wprowadzenie

Gwałtowny rozwój technologii komputerowych i internetu, który nastąpił w ostatnich latach spowodował zwiększone zapotrzebowanie na informacje i usługi praktycznie w każdej dziedzinie, w tym także z zakresu kartografii i systemów GIS. Co za tym idzie spowodował także ich szybki rozwój, najczęściej w stronę udostępniania danych w sieci.

Większość współczesnych systemów GIS oferuje możliwość pracy na danych udostępnianych w sieci przy pomocy serwerów WMS i WFS, umożliwiając użytkownikowi łączenie danych przechowywanych lokalnie z danymi zdalnymi. Część systemów oferuje także możliwość utworzenia serwera WMS i WFS i udostępniania w ten sposób posiadanych danych, zarówno w lokalnych sieciach Intranet jak i w Internecie.

Coraz częściej aplikacje te oparte są o architekturę klient-serwer, pozwalającą na rozdzielenie warstwy prezentacji danych od ich przetwarzania. Architektura ta, oraz jej odmiany została opisana szczegółowo w części pracy dotyczącej współczesnych systemów GIS.

Celem tej pracy było pokazanie możliwości prezentacji danych przestrzennych w sieci, jakie oferują współczesne systemy GIS. W tym celu dokonany został przegląd dostępnych na rynku systemów, zarówno komercyjnych jak i darmowych. Szczegółowo opisany został także jeden z najlepszych darmowych systemów do prezentacji danych przestrzennych w sieci, a mianowicie UMN MapServer. W ostatniej części przedstawiony został sposób implementacji najważniejszych funkcji MapServera, na przykładzie aplikacji dla potrzeb Regionalnego Zarządu Gospodarki Wodnej w Krakowie.

## 2. Systemy GIS

### 2.1 Historia systemów GIS

Historia systemów GIS rozpoczęła się w latach 60-tych XX w. w USA wraz z rozwojem techniki i powstawaniem coraz szybszych komputerów. Za prekursorów wykorzystania komputera do tworzenia map i gromadzenia danych geograficznych uważa się profesora Uniwersytetu w Harvardzie Howarda T. Fishera oraz Rogera Tomlinsona. Pierwszy z nich założył w 1964 roku Harvard Laboratory for Computer Graphics and Spatial Analysis w którym wkrótce powstał SYMAP (Synagraphic Mapping System) – pierwszy program do komputerowego opracowywania map. Z kolei Roger Tomlinson w 1963 roku rozpoczął w Kanadzie prace nad projektem CGIS (Canada Geographic Information System) będącym pierwszym systemem mającym objąć swoim zasięgiem cały kraj i zawierającym dane o możliwościach wykorzystania terenu. Właśnie od tego projektu powstała nazwa systemów GIS.

Koniec lat 60-tych i początek 70-tych to okres kilku ważnych wydarzeń w historii systemów GIS, min.:

- **1968** – w USA powstają firmy M&S Computing (obecnie Intergraph) i ESRI, które do dzisiaj wyznaczają kierunki rozwoju technologii GIS. Warto wspomnieć że założyciele obu firm (David Sinton - M&S Computing i Jack Dangermond - ESRI) wywodzą się z wspomnianego wyżej Harvard Laboratory for Computer Graphics and Spatial Analysis.
- **1969** – Powstaje uogólniony język znaczników GML (Generalized Markup Language), protoplasta języków HTML i XML. W oparciu o XML powstał język GML (Geography Markup Language) będący standardem stworzonym przez OGC i wykorzystywanym przez większość współczesnych systemów GIS.
- **1970** – zakończono wprowadzanie danych do projektu CGIS, projekt rozpoczyna działanie.
- **1972** – Wystrzelono pierwszego satelitę ERTS-1 (później nazwanego Landsat). Rozpoczynają się obserwacje Ziemi za pomocą technik teledetekcji satelitarnej.
- **1973** – Powstaje model TIN (Triangular Irregular Network) jako sposób przedstawienia powierzchni terenu.

W latach 70-tych i 80-tych wraz z rozwojem techniki pojawiają się nowe możliwości dla systemów GIS. Komputery stają się szybsze, mniejsze i dostępne nie tylko dla wąskiej grupy pracowników naukowych i wojskowych ale również dla przeciętnego człowieka. Następuje również szybki rozwój technik satelitarnych (duża liczba satelitów w przestrzeni okołozemskiej, przesyłają one na ziemię także coraz dokładniejsze zdjęcia i dane) co przekłada się na większą dokładność systemów GIS oraz możliwość stosowania ich w coraz szerszym wachlarzu dziedzin.

Ważniejsze wydarzenia z tego okresu:

- **1981** – firma ESRI wprowadza na rynek pierwszą wersję programu ArcInfo, wiodącego systemu komercyjnego GIS na świecie.
- **1985** – W US Army Construction Engineering Research Laboratories powstaje GRASS (Geographic Resources Analysis Support System), oprogramowanie typu Open Source wykorzystywane (zwłaszcza w środowiskach akademickich) do dzisiaj.

Lata 90-te to okres gdy wraz z upowszechnieniem się komputerów PC systemy GIS trafiły „pod strzechy”. To również okres rozwoju internetu i technologii z nim związanych. Spowodowało to ewolucję systemów GIS w kierunku aplikacji opartych o architekturę klient-serwer. Architektura klient-serwer jest tzw. „architekturą dwuwarstwową”, polega na rozdzieleniu warstwy gromadzenia i przetwarzania danych od warstwy prezentacji tych danych użytkownikowi. Dzięki takiemu rozwiązaniu na komputerze użytkownika (zazwyczaj dużo wolniejszym i o ograniczonej pojemności) instalowany był tylko program prezentujący wyniki (*klient*) a całe przetwarzanie danych było realizowane na komputerze o bardzo dużej mocy obliczeniowej (*serwer*), na którym znajdowała się również baza danych (osiągająca czasem bardzo duże rozmiary).

W latach 90-tych nastąpił również rozwój ruchu „Open Source” opartego o idee „bezpłatności” oprogramowania oraz jawności jego kodu źródłowego. Dzięki temu takie oprogramowanie było tworzone i poprawiane przez wielu użytkowników na całym świecie i każdy człowiek posiadający pewną wiedzę z zakresu programowania oraz dziedziny, którą dany program się zajmował mógł rozbudować go o kolejny nowy fragment lub zaproponować usprawnienie już istniejących rozwiązań.

Ważne dla rozwoju systemów GIS wydarzenia lat 90-tych to min.:

- **1994** - Powstaje organizacja Open GIS Consortium (OGC), obecnie Open Geospatial Consortium, grupująca na starcie 20 firm i instytucji zainteresowanych rozwojem otwartych systemów GIS. Obecnie liczy 339 firm, instytucji i agencji rządowych.
- **1998** – Uruchomiono TerraServer, wspólne przedsięwzięcie firm: Aerial Images Inc., Microsoft i Compaq oraz amerykańskiej służby geologicznej. W 2000 roku nastąpił rozłam w projekcie TerraServer spowodowany różnicą zdań co do dalszego losu projektu pomiędzy głównymi twórcami. Microsoft chciał pozostać przy darmowym systemie – demonstracji podczas gdy Aerial Images Inc. chciał przekształcić TerraServer w projekt komercyjny (sprzedaż zdjęć z rosyjskich satelitów). Obecnie obie firmy kontynuują własne wersje projektu. Każda z nich zawiera setki terabajtów danych.
- **1998 - 2001** – Prace nad GML (Geography Markup Language) opartym na języku XML (eXtensible Markup Language), zakończone w 2001 roku przyjęciem przez OGC języka GML jako standardu kodowania, rozpowszechniania i gromadzenia informacji przestrzennej.

## 2.2 Współczesne Systemy GIS

Współczesne systemy GIS oparte są na tzw. sieciowych usługach geoinformacyjnych jak np. dystrybucja i wyszukiwanie danych przestrzennych. Dzięki temu użytkownik korzystający tylko z przeglądarki internetowej i prostych narzędzi informatycznych uzyskuje dostęp do rozbudowanych baz danych oraz zaawansowanych funkcji GIS. Zbudowane są na bazie architektury n-warstwowej, przy czym najczęściej wykorzystywany jest model 3-warstwowy, na który składają się:

- interfejs użytkownika
- serwer aplikacji
- baza danych

W zależności od stopnia złożoności poszczególnych warstw oraz funkcji spełnianych przez nie w systemie wyróżniamy 3 rodzaje aplikacji [Litwin, Myrda, 2005]:

- **Klient Cienki**

Przetwarzanie danych jest dokonywane w całości na serwerze aplikacji. Użytkownik poprzez interfejs użytkownika (najczęściej jest to przeglądarka WWW) wysyła żądanie do serwera i otrzymuje przetworzone dane, zazwyczaj w postaci wygenerowanego obrazu fragmentu mapy. Klient Cienki jest wykorzystywany najczęściej w webmappingu. Zaletą takiego rozwiązania jest jego szybkość – dane są przetwarzane na serwerze, który zazwyczaj jest komputerem dużo szybszym od komputerów użytkowników oraz prostota obsługi – użytkownik nie musi instalować specjalnej aplikacji, zazwyczaj wystarczy przeglądarka WWW. Niestety prostota obsługi skutkuje ograniczonymi możliwościami wykorzystania zaawansowanych funkcji GIS.

- **Klient Średni**

Klient Średni działa na podobnych zasadach jak klient cienki, wymaga jednak instalacji w przeglądarce specjalnego plug-inu pozwalającego rozszerzyć możliwości aplikacji o bardziej zaawansowane funkcje obsługi mapy, takie jak np. obsługa różnych formatów danych. W dalszym ciągu większość przetwarzania danych dokonywana jest na serwerze, a oprogramowanie użytkownika tylko wyświetla przetworzone dane. Główną zaletą klienta średniego są rozbudowane w porównaniu do klienta cienkiego możliwości interakcji z mapą i wykorzystywania zaawansowanych funkcji GIS. Niestety konieczność instalacji dodatkowych plug-inów powoduje czasami problemy z kompatybilnością na różnych systemach operacyjnych i w przeglądarkach różnych producentów.

- **Klient Gruby**

W kliencie grubym większość lub wręcz całe przetwarzanie danych jest dokonywane na komputerze użytkownika. Serwer jest tylko dostarczycielem danych w formie nieprzetworzonej. Ten sposób działania wymaga instalacji aplikacji klienckiej służącej do przetwarzania danych i wykorzystującej zaawansowane funkcje GIS. Zaletą takiego rozwiązania jest możliwość pobrania danych interesujących użytkownika z serwera a następnie praca na nich w trybie off-line, bez komunikacji z serwerem. Wadą natomiast są duże wymagania sprzętowe stawiane komputerowi użytkownika.



Ważnym wymogiem stawianym przed współczesnymi Systemami GIS jest ich wzajemna kompatybilność, czyli możliwość wykorzystywania danych stworzonych przy pomocy jednego systemu do pracy w innym systemie. Aby to osiągnąć Systemy GIS muszą być zgodne z pewnymi ogólnie przyjętymi standardami i specyfikacjami. Standardy i specyfikacje które powinny spełniać systemy GIS są definiowane przez utworzoną w 1994 roku organizację OGC (Open Geospatial Consortium).

„Jest to międzynarodowa organizacja utworzona w 1994 roku. Jej główna siedziba znajduje się w Wayland, Massachusetts. Celem organizacji jest pełna integracja danych geoprzestrzennych i sposobów ich przetwarzania z systemami informatycznymi, jak również upowszechnianie przez infrastrukturę informacyjną interoperacyjnego oprogramowania do przetwarzania danych geoprzestrzennych oraz produktów geoinformacyjnych.” [Piszczek, 2007]

Organizacja ta zrzesza obecnie ok. 300 firm i instytucji działających między innymi w branży systemów informacji geograficznej, telekomunikacji, informatyki czy obronności.

Standardy i specyfikacje opracowane przez OGC to między innymi:

- **Catalogue Service** – specyfikacja definiująca interfejs pozwalający aplikacjom na przeglądanie, przetwarzanie i wyszukiwanie metadanych przy użyciu różnorodnych serwisów katalogowych.
- **Coordination Transformation Service** – specyfikacja określająca sposób identyfikowania, przetwarzania i konwertowania różnych systemów koordynatów.
- **Filter Encoding** – określa sposób definiowania przy pomocy języka XML wyrażeń filtrujących obiekty w celu stworzenia ich podzbioru. Filtry umożliwiają użytkownikowi wybranie tylko interesujących go obiektów które spełniają określone kryteria.

- 
- **Geographic Objects** – specyfikacja definiująca zbiór niezależnych od języka programowania abstrakcji służących do opisywania, zarządzania i manipulowania obiektami geometrycznymi i geograficznymi.
  - **Geography Markup Language** – język oparty na XML służący do modelowania, przechowywania i transportu informacji geograficznych umożliwiający opisywanie przestrzennych i nieprzestrzennych właściwości obiektów.
  - **Grid Coverage Service** – specyfikacja określająca metody współpracy między oprogramowaniem tworzonym przez dostawców danych a oprogramowaniem służącym do analizy i przetwarzania danych rastrowych
  - **Location Service** – specyfikacja definiująca platformę dla aplikacji służących do wyszukiwania lokalizacji.
  - **Simple Feature Access** – specyfikacja opisująca wspólną architekturę dla Systemów GIS a także metody współpracy z bazami SQL.
  - **Simple Feature CORBA, OLE/COM** – specyfikacja opisująca interfejs dostępu do danych geograficznych w rozproszonych środowiskach obliczeniowych.
  - **Styled Layer Description** – określa sposoby rozszerzania usługi Web Map Service o możliwość definiowania własnych symboli i stylów wyświetlania.
  - **Web Coverage Service** – specyfikacja definiująca standard wymiany danych przestrzennych w ich oryginalnej formie.
  - **Web Feature Service** – specyfikacja określająca sposoby dostępu do danych geograficznych w języku GML przechowywanych na serwerach WFS. Definiuje interfejsy do pobierania, przekształcania i łączenia danych z kilku źródeł.
  - **Web Map Context** – specyfikacja opisująca metody zapisu widoku mapy złożonej z kilku warstw pobranych z kilku serwerów WMS. Zapis musi być przeprowadzony w taki sposób, aby było możliwe odtworzenie tego widoku w późniejszym czasie.
  - **Web Map Service** – specyfikacja określająca sposoby udostępniania danych geograficznych w sieci.

Standaryzacja formatów wymiany danych umożliwia ich łatwe współdzielenie przez aplikacje różnych producentów i ułatwia pracę, gdyż nie ma zazwyczaj potrzeby konwersji formatu danych między aplikacjami.

Pomimo technicznego zaawansowania współczesnych aplikacji, do sprawnego działania system GIS wymaga również aktualnych i wiarygodnych danych oraz „czynnika ludzkiego” czyli doświadczonych ludzi nadzorujących poprawność i spójność tych danych.

## 3. Przegląd Systemów GIS

### 3.1 Systemy komercyjne

Systemy komercyjne są zazwyczaj bardzo rozbudowanymi aplikacjami udostępniającymi użytkownikowi pełen wachlarz możliwości analizy i modyfikowania danych geoprzestrzennych. Jest to ich wielką zaletą, która jednak powoduje że dla początkującego użytkownika system taki jest zbyt skomplikowany w obsłudze. W połączeniu z zazwyczaj mało intuicyjnym interfejsem powoduje to konieczność długiego szkolenia użytkownika.

Kolejną wadą takich systemów jest ich cena, która może sięgać kilkudziesięciu lub nawet kilkuset tysięcy dolarów, co znacząco redukuje grupę użytkowników jako że tylko największe firmy i instytucje mogą sobie pozwolić na zakup takiego systemu.

Poniżej zostały przedstawione najciekawsze produkty wiodących firm z branży systemów GIS: ESRI, Intergraph oraz Autodesk. Wspólną ich cechą jest to że umożliwiają dostęp do danych geoprzestrzennych poprzez internet lub intranet.

#### 3.1.1 Systemy firmy ESRI

##### ArcGIS Server

ArcGIS Server jest korporacyjnym serwerem aplikacji GIS, umożliwiającym całej organizacji / przedsiębiorstwu dostęp do funkcji GIS i danych zgromadzonych na serwerze. Zachowuje jednocześnie centralne zarządzanie zgromadzonymi danymi i obsługę aplikacji, bardzo ważne dla spójności danych. ArcGIS Server oferuje funkcjonalności takie jak: funkcje kartograficzne, geokodowanie, zapytania przestrzenne, edycję, śledzenie obiektów i złożone analizy. Dzięki wykorzystaniu popularnych środowisk programistycznych (Java, C++, .NET, COM) oraz możliwości tworzenia specjalistycznych aplikacji po stronie serwera nie jest wymagane instalowanie dodatkowych aplikacji po stronie klienta, powoduje to też że końcowy użytkownik nie musi posiadać szczegółowej wiedzy z zakresu GIS.

## **ArcIMS**

„Oprogramowanie ArcIMS służy do dystrybucji danych i aplikacji GIS za pośrednictwem internetu. Jest platformą umożliwiającą wymianę i współużytkowanie zbiorów danych w sieci (internet lub intranet).”[www.esri.pl, 2007]

Najważniejsze cechy oprogramowania ArcIMS to: [www.esri.pl, 2007]

- Możliwość integracji danych przechowywanych w lokalnych bazach z danymi dostępnymi poprzez internet,
- Łatwość projektowania, tworzenie i zarządzanie stronami WWW,
- Duża skalowalność i elastyczność architektury serwerów, dynamiczna edycja map,
- Wysoka jakość udostępnianych opracowań kartograficznych,
- Prosta instalacja, wdrożenie i zarządzanie systemem,
- Otwarte i skalowalne środowisko, strumieniowe przesyłanie danych wektorowych.

ArcIMS znajduje zastosowanie przy tworzeniu usług i aplikacji GIS dostępnych za pomocą internetu jak i funkcjonujących w ramach organizacji lub przedsiębiorstwa i udostępnianych przez intranet. Umożliwia także zarządzanie serwisami internetowymi oferującymi dane i usługi w zakresie GIS oraz wymianę i współdzielenie danych między użytkownikami.

Usługi i dane udostępniane przy pomocy ArcIMS mogą być wykorzystywane przez różnorodne programy klienckie począwszy od innych programów z rodziny ArcGIS, poprzez niezależne aplikacje tworzone przez użytkowników a skończywszy na aplikacjach w urządzeniach mobilnych.

## **ArcSDE**

„ArcSDE jest zorientowanym obiektowo serwerem danych przestrzennych działającym na komercyjnych systemach zarządzania relacyjnymi bazami danych.” [www.esri.pl,2007]

ArcSDE poprawia efektywność działania systemu GIS poprzez rozdzielenie aplikacji pomiędzy serwer bazodanowy, klientów oraz serwer ArcSDE. Zwiększa także niezawodność systemu poprzez zaawansowane mechanizmy kontroli integralności danych, transakcji oraz ułatwia przenoszenie danych z jednej bazy do drugiej bez ryzyka straty informacji. Ułatwia także budowę aplikacji GIS poprzez udostępnienie interfejsu programowania API dla języków JAVA i C/C++/C# .

ArcSDE został wbudowany w najnowszą wersję oprogramowania (9.2) ArcGIS Server i od 31 grudnia 2006 r. nie jest dostępny jako osobny produkt.

### **3.1.2 Systemy firmy Intergraph**

#### **Geomedia WebMap**

Geomedia WebMap jest narzędziem służącym do wizualizacji map i analizy danych geoprzestrzennych oraz udostępniania ich w internecie/intranecie użytkownikowi poprzez przeglądarkę WWW. Umożliwia wykorzystywanie danych w większości popularnych formatów, między innymi:

- Oracle9i™
- Microsoft® SQL Server
- Microsoft Access
- MGE
- MGE Segment Manager
- MGE GeoData Manager
- FRAMME™
- Esri Shape files
- MapInfo
- AutoCAD
- MicroStation® CAD
- OGC GML i WFS

Dane wyjściowe mogą być prezentowane zarówno w formacie wektorowym SVG jak i w formatach rastrowych takich jak:

- jpeg
- gif
- png
- tiff / geotiff

### **3.1.3 Systemy firmy Autodesk**

#### **Autodesk MapGuide**

„Autodesk MapGuide jest rozbudowaną platformą programistyczną pozwalającą programistom szybko i łatwo tworzyć aplikacje obsługujące dane przestrzenne dla sieci WWW i intranetu. Umożliwia integrowanie danych CAD i GIS, tworzenie aplikacji, wdrażanie w systemach Microsoft® Windows® i Linux oraz publikowanie dla wielu użytkowników.” [www.procad.pl, 2007]

Podstawowe cechy programu:

- Wieloplatformowość – MapGuide dostępny jest zarówno dla systemów operacyjnych z rodziny Windows jak i pod Linuxem. Może zostać uruchomiony jako usługa systemowa (Windows) lub demon (Linux).
- Możliwość tworzenia aplikacji obsługujących dane po stronie serwera, co powoduje że w przypadku aktualizacji oprogramowania na serwerze nie ma potrzeby aktualizowania programu u klienta.
- Obsługa wielu języków programowania między innymi JAVA, .NET, C++, PHP.
- Udostępnianie danych poprzez usługi WMS i WFS innym aplikacjom
- Skalowalność – możliwość rozbudowy serwisu o kolejne serwery w przypadku konieczności obsługi większej ilości użytkowników.

### 3.2 Systemy darmowe

W ostatnich latach nastąpił gwałtowny rozwój aplikacji typu Open Source, powodując że systemy GIS stały się dostępne dla zwykłego człowieka. Wbrew powszechnemu przekonaniu że to co darmowe musi być gorsze, uboższe od rzeczy za które trzeba zapłacić, wiele darmowych systemów dorównuje rozwiązaniom komercyjnych lub wręcz je przewyższa. Co więcej, dzięki ogólnie dostępnemu kodowi źródłowemu każdy użytkownik o pewnej wiedzy programistycznej jest w stanie rozszerzyć funkcjonalność takich systemów zgodnie ze swoimi potrzebami lub usunąć zauważony błąd. Bardzo często systemy takie mają również bardziej intuicyjną obsługę od systemów komercyjnych, przez co nawet mniej zaznajomiony z komputerem i GIS użytkownik może ich używać. Poniżej przedstawione zostały 2 najpopularniejsze darmowe systemy GIS.

#### **GRASS (Geographic Resources Analysis Support System)**

GRASS jest darmowym systemem GIS przeznaczonym do tworzenia, przetwarzania i analizy danych geoprzestrzennych, stworzonym na potrzeby Armii Amerykańskiej przez US Army CERL (Construction Engineering Research Laboratory). Początkowo dostępny tylko na platformie UNIX, obecnie dostępne są wersje między innymi na systemy LINUX, MacOS, Solaris, SGI IRIX, HP UX, CRAY Unicos oraz MS Windows.

Najnowsza wersja GRASS zawiera ok.350 aplikacji i narzędzi do przetwarzania danych, jako że kod źródłowy jest ogólnodostępny każdy użytkownik może napisać swoją własną aplikację lub narzędzie najlepiej odpowiadające jego potrzebom.

Możliwości przetwarzania danych w GRASS:

- Analizy przestrzenne
- Generowanie map
- Wizualizacje danych 2D i 3D
- Generacja danych z modeli
- Współpraca z systemami baz danych (min. Oracle, MySQL, PostgreSQL, ODBC)

GRASS obsługuje większość popularnych formatów danych:



- 
- Dane rastrowe – wszystkie formaty obsługiwane przez bibliotekę GDAL, min.: JPEG, GIF, TIFF, ASCII, ARC/GRID, E00, GMT, PNG, ERDAS LAN, Vis5D
  - Dane wektorowe – wszystkie formaty obsługiwane przez bibliotekę OGR, min.: ASCII, ARC/INFO, ESRI shapefiles, BIL, DLG (U.S.), DXF, DXF3D, GMT, GPS-ASCII, USGS-DEM, IDRISI, MOSS, MapInfo MIF, TIGER, VRML

## QuantumGIS

QuantumGIS jest darmowym systemem GIS, rozpowszechnianym na licencji Open Source. Dostępny jest w wersjach na kilka różnych systemów operacyjnych, m.in. Windows, MacOS czy Linux. Pozwala na korzystanie z danych w najpopularniejszych formatach, zarówno rastrowych, np. JPEG, GIF, TIF jak i wektorowych, np. ESRI Shapefile, Mapinfo file czy SDTS. Umożliwia także wykorzystywanie danych udostępnianych przez serwery WMS.

Najważniejsze cechy QuantumGIS wg. jego autorów to:

- Obsługa PostGIS
- Obsługa formatów wektorowych poprzez bibliotekę OGR
- Obsługa formatów rastrowych poprzez bibliotekę GDAL
- Możliwość integracji z GRASS
- Narzędzie do tworzenia map, tzw. Map Composer
- Możliwość rozszerzenia funkcjonalności poprzez instalację dodatkowych plug-inów
- Możliwość eksportu danych do formatu Mapserver mapfile
- Obsługa serwerów WMS

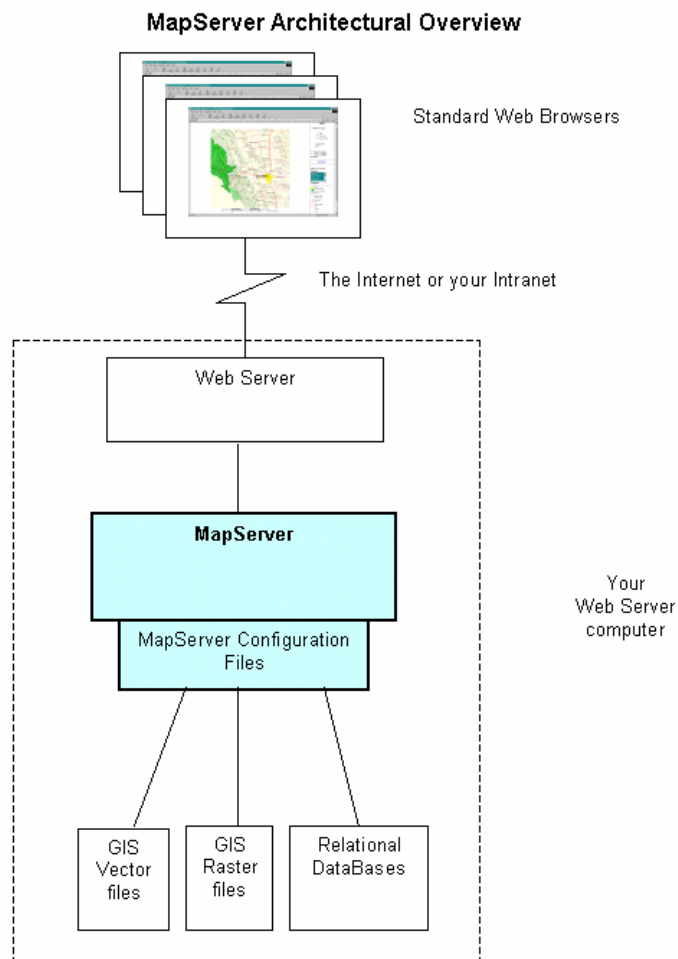
Niewątpliwą zaletą QuantumGIS jest przyjazny interfejs użytkownika i łatwość obsługi, dzięki której nawet użytkownik bez dużej wiedzy z zakresu GIS poradzi sobie z uzyskaniem potrzebnych informacji.

## 4. UMN MapServer

### 4.1 Wprowadzenie do UMN MapServer

UMN MapServer jest darmowym środowiskiem programistycznym pozwalającym na tworzenie sieciowych aplikacji przetwarzających dane przestrzenne. Rozpowszechniany jest na zasadach licencji Open Source.

MapServer nie jest pełnowartościowym systemem GIS i nie aspiruje do bycia takim. Głównym zadaniem stawianym przed MapServerem jest generowanie i udostępnianie danych przestrzennych, zarówno map, obrazów jak i danych wektorowych na potrzeby aplikacji sieciowych.



**Rysunek 1. Struktura Mapservera**  
[<http://mapserver.gis.umn.edu/>]

MapServer został stworzony na University of Minnesota (UMN). Powstał jako fragment projektu ForNet, będącego wspólnym przedsięwzięciem University of Minnesota, NASA oraz Minnesota Department of Natural Resources (MNDNR). Obecnie MapServer jest częścią projektu TerraSIP, sponsorowanego przez NASA a realizowanego przez University of Minnesota we współpracy z konsorcjum firm i organizacji.

W pracach nad rozwojem oprogramowania wchodzącego w skład MapServera bierze udział ok.20 firm i organizacji z całego świata i liczba ta stale rośnie.

## 4.2 Najważniejsze funkcje i możliwości

### Zaawansowane możliwości kartograficzne

- **Zależne od skali wyświetlanie obiektów** – MapServer pozwala na uzależnienie od skali mapy wyświetlania lub nie właściwości, obiektów i poszczególnych warstw. Pozwala to z jednej strony na przyspieszenie działania aplikacji (obiekty niewidoczne w danej skali np. z powodu swoich rozmiarów nie będą umieszczane na mapie, zmniejszając ilość przesyłanych danych) a także poprawia wygląd aplikacji (przy zbyt dużym powiększeniu obrazu rastrowe mogłyby zostać wyświetlone jako duże piksele psując wygląd graficzny aplikacji).
- **Nadawanie etykiet obiektom** – pozwala na przypisywanie etykiet do obiektów lub ich właściwości. Zapobiega także występowaniu konfliktów i kolizji pomiędzy etykietami.
- **W pełni konfigurowalne wyświetlanie** – dzięki możliwości zastosowania szablonów mechanizm wyświetlania mapy można łatwo zmodyfikować i dostosować do potrzeb użytkownika.
- **Czcionki TrueType**
- **Automatyzacja elementów mapy** – Część elementów składowych mapy, takich jak legenda, podgląd mapy czy skala może być automatycznie tworzona i dodawana przez MapServer. Jednocześnie użytkownik dysponuje dużymi możliwościami konfiguracji wyglądu i zachowania tych elementów.

- **Mapy tematyczne** – MapServer pozwala na tworzenie w prosty sposób map tematycznych w oparciu o klasy obiektów. Możliwe jest wykorzystywanie klas logicznych jak i klas opartych o wyrażenia regularne.

### **Wsparcie dla języków programowania**

MapServer oferuje poprzez swój interfejs programistyczny API wsparcie dla różnorodnych języków i środowisk programistycznych. Umożliwia w prosty sposób tworzenie aplikacji korzystających z MapServera przy użyciu języków skryptowych takich jak PHP, PERL czy PYTHON jak i języków programowania takich jak C++/C#, .NET, DELPHI czy JAVA. Zapewnia gotowe biblioteki dla tych języków ułatwiając użytkownikom tworzenie aplikacji przetwarzających dane przestrzenne.

### **Wieloplatformowość**

Mapserver może zostać uruchomiony na wielu różnych systemach operacyjnych, począwszy od systemów opartych o UNIX-a, poprzez wiele dystrybucji systemu LINUX i systemy z rodziny MS WINDOWS a skończywszy na MacOS X i SUN Solaris.

### **Mnogość obsługiwanych formatów danych**

MapServer pozwala na wykorzystywanie danych zapisanych w większości popularnych formatów, tak rastrowych jak i wektorowych.

- Obsługuje formaty rastrowe poprzez bibliotekę GDAL. Biblioteka ta umożliwia odczyt (i w przypadku niektórych formatów zapis) danych zapisanych w formatach takich jak:
  - JPEG
  - GIF
  - TIFF/GeoTIFF
  - USGS ASCII DEM
  - PNG
  - EPPL7
  - ArcInfo ASCII i Binary GRID

- 
- Dane wektorowe obsługiwane są przez bibliotekę OGR. Umożliwia ona odczyt danych zapisanych min. w formatach:
    - ESRI Shapefile
    - PostGIS
    - ESRI ArcSDE
    - Oracle Spatial
    - MySQL
    - GRASS
    - DWG / DXF

MapServer jest także w pełni zgodny ze standardami opracowanymi przez Open Geospatial Consortium (OGC) takimi jak:

- WMS (zarówno klient jak i serwer)
- WFS (bez obsługi transakcji)
- WMC
- WCS
- język GML
- systemy filtrów, szablonów i schematów

### **Zaawansowane odwzorowania map**

Dzięki wykorzystaniu biblioteki Proj.4 MapServer posiada możliwość przetwarzania w locie wyświetlania mapy na jedno z tysięcy odwzorowań kartograficznych zdefiniowanych w tej bibliotece.

## **4.3 Struktura aplikacji opartej o MapServer**

Typowa aplikacja stworzona w oparciu o MapServer składa się z kilku podstawowych komponentów:

- **Map File** – tekstowy plik, zazwyczaj o rozszerzeniu .map zawierający dane konfiguracyjne dla Aplikacji. Definiuje między innymi wyświetlany obszar, lokalizację danych wejściowych i miejsce składowania danych wyjściowych. W pliku tym zdefiniowane są także wszystkie warstwy mapy oraz klasy

---

poszczególnych warstw. Zawartość plików .map została omówiona szczegółowo w kolejnym podrozdziale.

- **Dane geograficzne** – MapServer obsługuje wiele formatów danych wejściowych poprzez biblioteki OGR i GDAL. Domyślnym formatem danych wejściowych jest ESRI Shapefile.
- **MapServer CGI** – Binarny plik wykonywalny, do którego wysyłane jest zapytanie z przeglądarki a który zwraca do przeglądarki wygenerowany na podstawie tego zapytania obraz. Zazwyczaj jest zlokalizowany w katalogu cgi-bin na serwerze http.
- **Serwer HTTP** – dostarcza strony HTML do przeglądarki użytkownika. Prawidłowo skonfigurowany i działający serwer HTTP, np. Apache lub Microsoft IIS jest wymagany do działania MapServera.
- **Strony HTML** – stanowią interfejs pomiędzy użytkownikiem a MapServerem. W najprostszym przypadku wywołanie MapServera umieszcza statyczny obraz na stronie. Aby umożliwić użytkownikowi interakcję z mapą należy obraz zwracany przez MapServer umieścić na stronie jako element formularza. Programy CGI jak MapServer nie zapamiętują swojego poprzedniego stanu i każde zapytanie przesłane do nich generuje na nowo stan aplikacji. Zachodzi zatem potrzeba ponownego przekazania wszystkich informacji na temat stanu mapy w każdym kolejnym zapytaniu. Realizowane jest to zazwyczaj poprzez umieszczanie tych informacji w ukrytych polach formularza lub przekazanie ich w adresie URL.

Prosta aplikacja może składać się z 2 stron HTML:

**Strona startowa** – korzystając z ukrytych pól formularza wysyła do MapServera i serwera http żądanie inicjujące wyświetlanie mapy.

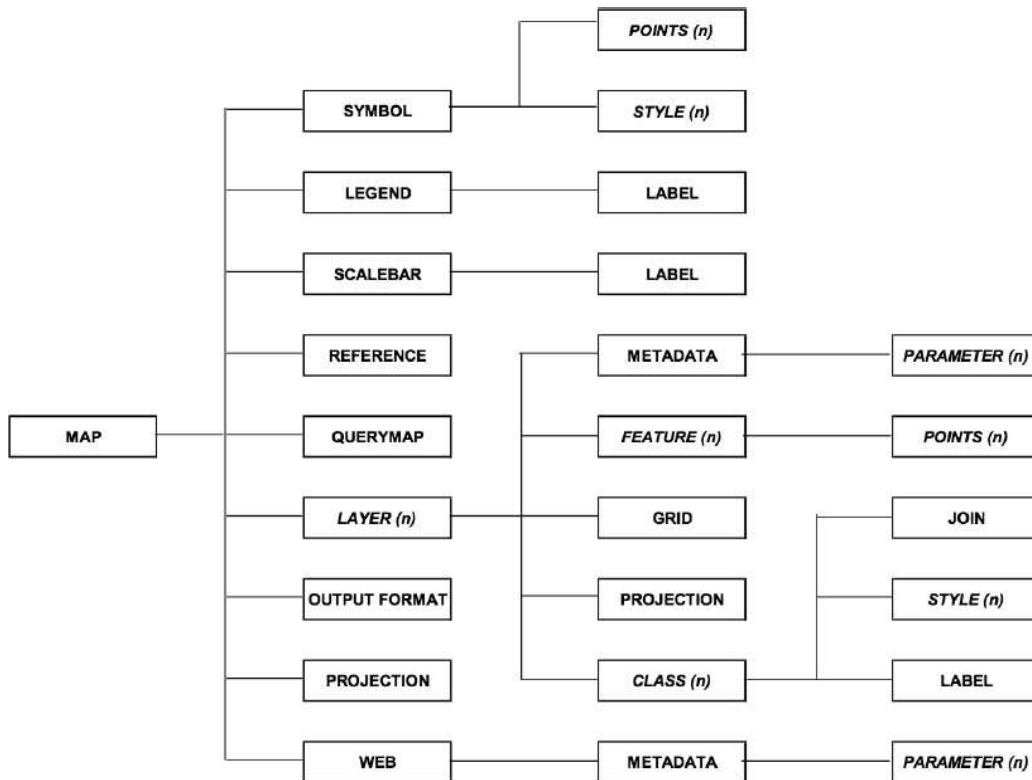
**Plik szablonu** – kontroluje sposób, w jaki mapa i legenda będą wyświetlane w przeglądarce użytkownika. Przy pomocy tej strony następuje przesłanie aktualnego stanu aplikacji (np. informacji, które warstwy są wyświetlane lub aktualnych współrzędnych wyświetlanego obszaru) do MapServera, który na tej podstawie tworzy nową zawartość mapy. Szablon definiuje także dostępne dla użytkownika rodzaje interakcji z mapą (przybliżanie, oddalanie, przesuwanie, zapytania).

## 4.4 Zawartość Map File

Pliki Map File są sercem MapServera. Zawierają definicje związków między obiektami, wskazują gdzie znajdują się pliki z danymi wejściowymi i określają sposób w jaki rezultat działania aplikacji ma być wyświetlany.

Podstawowe zasady tworzenia Map File:

- W plikach Map File nie jest ważna wielkość liter w nazwach obiektów.
- Ciągi znaków zawierające znaki inne niż alfanumeryczne muszą być umieszczane w cudzysłowach. Zaleca się umieszczanie wszystkich ciągów znaków w cudzysłowach.
- Maksymalna ilość warstw (LAYER) zdefiniowanych w jednym pliku .map wynosi domyślnie 200. Inne domyślne ograniczenia to maksymalna liczba klas (CLASS) ustalona na 250, maksymalna liczba stylów (STYLE) wynosząca 5 oraz maksymalna liczba symboli (SYMBOL) wynosząca 64. Wartości te można zmienić edytując plik map.h jednak wymaga to ponownej kompilacji MapServera.
- Ścieżki do plików mogą być podawane zarówno w postaci bezwzględnej lub względnej w stosunku do lokalizacji pliku .map. Można także wykorzystać atrybut SHAPEPATH.
- Komentarze są poprzedzane znakiem #.
- Pliki Map File oparte są na strukturze drzewiastej z obiektem typu MAP jako „korzeniem” zawierającym wszystkie inne obiekty.



Rysunek 2. Struktura Pliku Map File [opracowanie własne]

Poniżej zostały omówione typy obiektów dostępnych w plikach .map i ich najważniejsze parametry:

**MAP** – główny obiekt w pliku. Zawiera w sobie definicje pozostałych obiektów.

Parametry dostępne w obiekcie MAP:

- **ANGLE** [double] – określa kąt, o jaki ma zostać obrócona mapa. Domyślnie ustawiony na 0. Parametr ten działa tylko przy użyciu MapScriptu, nie jest obsługiwany poprzez tryb CGI.
- **CONFIG** [key] [value] – służy do ustawiania zmiennych środowiskowych, jak np. ścieżki dostępu do katalogu zawierającej pliki biblioteki PROJ.4.
- **DEBUG** [on|off] – uruchamia tryb diagnostyczny dla mapy. Ewentualne błędy trafiają do standardowego portu wyjścia lub do pliku logów MapServera jeśli taki został zdefiniowany. Użycie tego parametru wymaga skompilowania mapservera z opcją --with-debug .
- **EXTENT** [minx] [miny] [maxx] [maxy] – określa początkowy obszar wyświetlania mapy.



- 
- FONTSET [filename] – względna lub bezwzględna ścieżka do pliku zawierającego czcionkę, która ma zostać użyta.
  - IMAGECOLOR [r] [g] [b] – definiuje kolor tła mapy.
  - IMAGETYPE [gif|png|jpeg|wbmp|gtiff|swf|userdefined] – określa format, w jakim ma zostać wygenerowana mapa. Nazwa formatu musi dokładnie odpowiadać parametrowi „NAME” jednego z obiektów z sekcji OUTPUT FORMAT.
  - LAYER – rozpoczyna definicję obiektu LAYER omówionego w dalszej części rozdziału.
  - LEGEND - rozpoczyna definicję obiektu LEGEND omówionego w dalszej części rozdziału.
  - MAXSIZE [integer] – określa maksymalny rozmiar mapy w pikselach. Ustalenie tego parametru na 1000 spowoduje, że mapa będzie mogła maksymalnie mieć rozmiar 1000 na 1000 pikseli.
  - NAME [nazwa] – definiuje nazwę mapy.
  - PROJECTION – rozpoczyna definicję obiektu PROJECTION omówionego w dalszej części rozdziału.
  - REFERENCE - rozpoczyna definicję obiektu REFERENCE MAP omówionego w dalszej części rozdziału.
  - RESOLUTION [int] – określa rozdzielczość generowanego obrazu.
  - SCALE [double] – określa skalę mapy. Zazwyczaj określane przez aplikację.
  - SCALEBAR - rozpoczyna definicję obiektu SCALEBAR omówionego w dalszej części rozdziału.
  - SHAPEPATH [filename] – określa ścieżkę do katalogu zawierającego pliki z danymi.
  - SIZE [x][y] – rozmiar w pikselach zwracanego obrazu.
  - STATUS [on|off] – określa czy mapa jest aktywna.
  - SYMBOLSET [filename] – ścieżka do pliku z zestawem użytych symboli.
  - SYMBOL – rozpoczyna definicję obiektu SYMBOL, omówionego w dalszej części rozdziału.
  - UNITS [feet|inches|kilometers|meters|miles|dd] – określa jednostkę mapy.

---

**LAYER** – obiekt zawierający definicję warstwy mapy. Warstwy mapy są wyświetlane w kolejności w jakiej są zdefiniowane w pliku .map. Parametry dostępne w obiekcie LAYER:

- **CLASS** – rozpoczyna definicję obiektu CLASS omówionego w dalszej części rozdziału.
- **CLASSITEM [attribute]** – nazwa pola w tabeli atrybutów wg którego obiekty są dzielone na klasy.
- **CONNECTION [string]** – ciąg określający połączenie z bazą danych. W zależności od rodzaju bazy przyjmuje różną postać, np. dla połączenia z bazą SDE zawiera host name, instance name, database name, username i password oddzielone przecinkami.
- **CONNECTIONTYPE [local|sde|ogr|postgis|oraclespatial|wms]** – określa rodzaj połączenia z bazą danych. Domyślnie jest ustawiony na „local”.
- **DATA [filename][sde parameters][postgis table/column][oracle table/column]** – określa nazwę pliku z danymi lub w przypadku połączeń bazodanowych nazwę tabeli i kolumny zawierającej dane.
- **DEBUG [on|off]** – uruchamia tryb diagnostyczny dla warstwy. Ewentualne błędy trafiają do standardowego portu wyjścia lub do pliku logów MapServera jeśli taki został zdefiniowany.
- **DUMP [true|false]** – określa czy umożliwić MapServerowi zwracanie danych w formacie GML. Przydatne przy komendzie „GetFeatureInfo” używanej przez WMS. Domyślnie ustawione na false.
- **FEATURE** – rozpoczyna definicję obiektu FEATURE omówionego w dalszej części rozdziału.
- **GROUP [name]** – określa grupę, do której należy warstwa. Pozwala to np. na włączanie / wyłączanie całych grup.
- **MAXSCALE [double]** – maksymalna skala, przy jakiej warstwa jest wyświetlana.
- **MINSCALE [double]** – minimalna skala, przy której warstwa jest wyświetlana.
- **NAME [string]** – określa nazwę warstwy.
- **PROJECTION** – rozpoczyna definicję obiektu PROJECTION omówionego w dalszej części rozdziału.
- **STATUS [on|off]** – określa czy mapa jest aktywna.

- 
- **TYPE** [point|line|polygon|circle|annotation|raster|query] – określa sposób wyświetlania warstwy.

**CLASS** – obiekt zawierający definicję klasy dla danej warstwy. Każda warstwa musi posiadać co najmniej jedną klasę. Jeśli jedna warstwa ma więcej niż jedną klasę przynależność jest ustalana w oparciu o wartości atrybutów i wyrażenia regularne. Parametry dostępne w obiekcie **LAYER**:

- **BACKGROUND** [r] [g] [b] – określa kolor tła nieprzezroczystych symboli.
- **COLOR** [r] [g] [b] – określa kolor symboli.
- **DEBUG** [on|off] – uruchamia tryb diagnostyczny dla warstwy. Ewentualne błędy trafiają do standardowego portu wyjścia lub do pliku logów MapServera, jeśli taki został zdefiniowany.
- **EXPRESSION** [string] – określa przynależność do klasy. Wyróżniamy 4 rodzaje testów przynależności do klasy: porównanie ciągu znaków, wyrażenia regularne, wyrażenia logiczne i funkcje operujące na tekście.
- **KEYIMAGE** [filename] – określa nazwę pliku zawierającego symbol klasy wykorzystywany przy tworzeniu legendy.
- **LABEL** – rozpoczyna definicję obiektu LABEL omówionego w dalszej części rozdziału.
- **MAXSCALE** [double] – maksymalna skala, przy jakiej klasa jest wyświetlana.
- **MINSCALE** [double] – minimalna skala, przy której klasa jest wyświetlana.
- **MAXSIZE** [integer] – maksymalny rozmiar symbolu w pikselach.
- **MINSIZE** [integer] – minimalny rozmiar symbolu w pikselach.
- **NAME** [string] – nazwa klasy
- **OUTLINECOLOR** [r] [g] [b] – kolor obramowania symbolu.
- **SIZE** [integer] – wysokość symbolu w pikselach. Używany tylko w przypadku skalowalnych symboli.
- **STYLE** - rozpoczyna definicję obiektu STYLE omówionego w dalszej części rozdziału.
- **SYMBOL** [integer|string|filename] – określa symbol, który ma być wykorzystywany jako reprezentacja tej klasy. Wpisanie liczby powoduje wykorzystanie symbolu o takim numerze z pliku symboli. Wpisanie ciągu znaków

powoduje użycie symbolu o takiej nazwie. Możliwe jest także podanie ścieżki względnej do pliku gif/png zawierającego symbol.

- **TEMPLATE** [filename] – nazwa pliku zawierającego szablon prezentacji wyników zapytania dla danej klasy.
- **TEXT** [string] – określa tekst używany do nadawania etykiet właściwościom danej klasy.

**FEATURE** – obiekt umożliwiający wpisanie „inline” definicji elementu warstwy, np. punktu, linii czy polilinii zamiast tworzenia osobnego pliku shapefile.

Parametry dostępne w obiekcie **FEATURE**:

- **POINTS** – lista par współrzędnych (x,y) definiujących kolejne punkty elementu i zakończona **END**.
- **TEXT** [string] – określa ciąg znaków używany do nadawania etykiet elementowi.
- **WKT** [string] – określa geometrię obiektu zdefiniowaną poprzez wykorzystanie formatu OpenGIS Well Known Text.

**INCLUDE** – obiekt umożliwiający wczytanie innego pliku .map w miejsce wywołania **INCLUDE**.

**LABEL** – Obiekt służący do definiowania etykiet dla elementów warstwy / klasy. Parametry dostępne w obiekcie **LABEL**:

- **ANGLE** [double|auto|follow] – określa kąt, pod jakim etykieta będzie wyświetlana.
- **ANTIALIAS** [true|false] – określa czy tekst ma być wygładzany.
- **BACKGROUND**COLOR [r] [g] [b] – kolor tła.
- **BUFFER** [integer] – odstęp w pikselach pomiędzy napisem a innymi obiektami na mapie.
- **COLOR** [r] [g] [b] – kolor tekstu etykiety.
- **ENCODING** [string] – określa kodowanie tekstu
- **FONT** [name] – określa czcionkę użytą w etykiecie.
- **MAXSIZE** [integer] – maksymalny rozmiar czcionki. Domyślnie ustalone na 256.
- **MINDISTANCE** [integer] – minimalny odstęp w pikselach pomiędzy etykietami.

- **MINFEATURESIZE** [integer|auto] – minimalna wielkość elementu, przy której etykieta jest wyświetlana.
- **MINSIZE** [integer] – minimalny rozmiar czcionki. Domyślnie ustalone na 4.
- **OUTLINECOLOR** [r] [g] [b] – kolor obramowania etykiety.
- **PARTIALS** [true|false] – określa czy tekst może wychodzić poza obszar mapy. Domyślnie ustawiony na „true”.
- **POSITION** [ul|uc|ur|cl|cc|cr|ll|lc|lr|auto] – określa pozycję etykiety względem elementu mapy do którego należy.
- **SIZE** [integer][tiny|small|medium|large|giant] – określa rozmiar tekstu.
- **TYPE** [bitmap|truetype] – określa typ użytej czcionki.
- **WRAP** [character] – określa znak nowej linii.

**LEGEND** – Obiekt służący do definiowania legendy mapy. Parametry dostępne w obiekcie LEGEND:

- **IMAGECOLOR** [r] [g] [b] – definiuje kolor tła.
- **LABEL** – rozpoczyna definicję obiektu LABEL omówionego powyżej.
- **OUTLINECOLOR** [r] [g] [b] – określa kolor obramowania elementów legendy.
- **POSITION** [ul|uc|ur|ll|lc|lr] – pozycja legendy na mapie.
- **KEYSIZE** [x][y] – określa rozmiar symboli elementów legendy.
- **KEYSPACING** [x][y] – określa odstęp pomiędzy elementami legendy.
- **STATUS** [on|off|embed] – określa czy legenda jest dołączana do mapy.
- **TEMPLATE** [filename] – określa nazwę pliku z szablonem legendy.

**PROJECTION** – Obiekt definiujący system koordynat wykorzystywany do wyświetlania danych. Należy zdefiniować jeden obiekt dla całej mapy i po jednym dla każdej warstwy danych. MapServer wykorzystuje do definiowania koordynat bibliotekę PROJ.4.

Definiując sposób wyświetlania można użyć serii parametrów biblioteki PROJ.4 lub odwołać się do plików epsg, będących częścią tej biblioteki a które zawierają wcześniej zdefiniowane parametry.

**SCALEBAR** – Obiekt służący do definiowania skali mapy. Parametry dostępne w obiekcie SCALEBAR:

- IMAGECOLOR [r] [g] [b] – definiuje kolor tła.
- COLOR [r] [g] [b] – definiuje kolor elementów skali.
- INTERVALS [integer] – liczba odcinków, na które należy podzielić skalę.
- LABEL – rozpoczyna definicję obiektu LABEL omówionego powyżej.
- OUTLINECOLOR [r] [g] [b] – określa kolor obramowania elementów skali.
- POSITION [ul|uc|ur|ll|lc|lr] – określa położenie skali na mapie.
- SIZE [x][y] – określa rozmiar skali w pikselach
- STATUS [on|off|embed] – określa czy legenda jest dołączana do mapy.
- STYLE [integer] – określa styl skali. Możliwe wartości to 0 i 1.
- UNITS [feet|inches|kilometers|meters|miles] – określa jednostkę skali.

**STYLE** – Obiekt służący do definiowania styli elementów. Parametry dostępne w obiekcie STYLE:

- ANGLE [double] – określa kąt nachylenia elementu na mapie.
- ANTIALIAS [true|false] – określa czy czcionki i symbole mają być wygładzane.
- BACKGROUNDCOLOR [r] [g] [b] – definiuje kolor tła dla nieprzezroczystego elementu
- COLOR [r] [g] [b] – definiuje kolor elementu.
- MAXSIZE [integer] – maksymalny rozmiar w pikselach symbolu elementu.
- MINSIZE [integer] – minimalny rozmiar w pikselach symbolu elementu.
- OUTLINECOLOR [r] [g] [b] – kolor obramowania elementu.
- SYMBOL [integer|string|filename] - określa symbol, który ma być wykorzystywany jako reprezentacja tego elementu. Wpisanie liczby powoduje wykorzystanie symbolu o takim numerze z pliku symboli. Wpisanie ciągu znaków powoduje użycie symbolu o takiej nazwie. Możliwe jest także podanie ścieżki względnej do pliku gif/png zawierającego symbol.
- WIDTH [integer] – grubość rysowanej linii.

Dzięki prostocie swojej struktury oraz bardzo przyjaznemu opisowi poszczególnych funkcji w dokumentacji dostępnej na stronie MapServera, stworzenie pliku .map nie nastręcza żadnych trudności nawet dla użytkownika nie zaznajomionego na co dzień z technologią GIS i MapServerem. Pliki Map File można utworzyć w dowolnym edytorze tekstowym lub przy użyciu jednego z kilku dostępnych w sieci narzędzi do tego celu.

## 4.5 PHP Mapscript

Kolejną bardzo ważną częścią MapServera jest język skryptowy MapScript opisany w tym podrozdziale na przykładzie PHP MapScript. MapScript działa niezależnie od modułu CGI Mapserver. Stanowi interfejs pozwalający tworzyć samodzielne aplikacje jak i aplikacje sieciowe przy użyciu wybranego przez użytkownika języka skryptowego. W chwili obecnej istnieje możliwość używania MapScriptu między innymi w językach Php, Perl, Python, Ruby, Tcl, Java i C#. Poniżej został przedstawiony krótki opis najważniejszych klas obiektów i ich funkcji dostępnych w PHP MapScript. Klasy zostały ułożone w porządku alfabetycznym dla łatwiejszej orientacji.

- **ClassObj** – Odpowiada obiektowi CLASS w pliku Map File. Obiekt tej klasy może zostać zwrócony przez obiekt klasy Layer lub utworzony przy użyciu konstruktora.
- **ImageObj** – Obiekt tej klasy jest tworzony przy użyciu metod klasy MapObj. Zawiera aktualną mapę jako obraz.
- **LabelObj** – Obiekt tej klasy zawiera etykietę i odpowiada obiektowi LABEL w pliku Map File. Obiekt ten jest zawsze zagnieżdżony w obiekcie innej klasy.
- **LayerObj** – Odpowiada obiektowi LAYER w pliku Map File. Obiekty tej klasy są tworzone przy użyciu konstruktora lub zwracane przez obiekt klasy MapObj.
- **LegendObj** - Odpowiada obiektowi LEGEND w pliku Map File. Obiekty tej klasy są zawsze zagnieżdżone w obiekcie klasy MapObj.
- **LineObj** – Obiekt tej klasy jest reprezentacją linii na mapie / warstwie. Składa się ze zbioru punktów, przez które przechodzi linia.
- **MapObj** – Klasa główna będąca reprezentacją pliku Map File w PHP MapScript. Obiekt tej klasy zawiera w sobie obiekty pozostałych klas.
- **PointObj** – Obiekt tej klasy jest reprezentacją punktu na mapie. Zawiera 3 współrzędne punktu (x,y,z) lub w przypadku plików shapefile 4 współrzędne (x,y,z,m).
- **ProjectionObj** – Klasa tworząca obiekt definiujący system koordynatów mapy, odpowiada obiektowi PROJECTION w pliku Map File. Mapa a także każda warstwa mapy muszą mieć zdefiniowany przynajmniej jeden obiekt tego typu.

- 
- **RectObj** – Obiekt tej klasy definiuje prostokąt na mapie. Obiekty tej klasy wykorzystywane są także przy zbliżaniu/oddalaniu mapy.
  - **ScalebarObj** – Odpowiada obiektowi SCALEBAR w pliku Map File. Obiekty tej klasy są zawsze zagnieżdżone w obiekcie klasy MapObj. Zawierają zmienne definiujące sposób wyświetlania i zachowania skali mapy.
  - **ShapefileObj** – Obiekt tej klasy jest reprezentacją pliku shapefile w aplikacji. Wywołując metody tej klasy można modyfikować zawartość pliku shapefile, np. dodając lub usuwając obiekty.
  - **ShapeObj** – obiekt będący reprezentacją figury (shape'a) zawartego w pliku shapefile lub stworzonego w aplikacji. Metody tej klasy pozwalają na modyfikowanie takiej figury, np. dodawanie lub usuwanie składowych figury takich jak linia lub punkt.
  - **StyleObj** – Klasa będąca odpowiednikiem obiektu STYLE w pliku Map File. Definiuje styl wyświetlania warstwy / klasy obiektów na warstwie.
  - **SymbolObj** – Klasa obiektów wykorzystywanych do definiowania symboli użytych na mapie.
  - **WebObj** – Obiekt tej klasy odpowiada za wygląd aplikacji. Definiowane są w nim między innymi nagłówek i stopka strony, a także na jaką stronę ma przekierować w przypadku wystąpienia błędu.

## 4.6 Najciekawsze przykłady implementacji z galerii UMN Mapserver

Galeria implementacji dostępna na stronie <http://mapserver.gis.umn.edu/gallery> zawiera około 100 pozycji z całego świata. Znajdują się tam zarówno duże projekty obejmujące swym zasięgiem obszar całego kraju lub nawet kilku krajów jak i małe projekty tworzone przez entuzjastów i obejmujące zasięgiem np. miasto lub jakąś jego część. Poniżej przedstawione zostały najciekawsze z projektów dostępnych w galerii:



UMN MapServer jako platforma WebGIS  
na przykładzie RZGW w Krakowie

- **Free public access to water data in France**

<http://sandre.eaufrance.fr/geoviewer/?lang=en>

Jedna z najciekawszych implementacji UMN MapServer opisanych w galerii. Umożliwia internautom dostęp do danych hydrologicznych dotyczących obszaru Francji a także jej departamentów zamorskich jak np. Martynika lub Gwadelupa. Umożliwia eksport danych do formatu ESRI Shapefile wykorzystując WFS, dokonywanie analizy danych a także pozwala na wykorzystywanie wielu źródeł map tła, takich jak zdjęcia satelitarne czy warstwy udostępniane przez serwery WMS.



**Rysunek 3. Sandre GeoViewer**

[<http://sandre.eaufrance.fr/geoviewer/?lang=en>]

Wielką zaletą tej implementacji jest wręcz ogromna ilość informacji dostępnych w niej. Umożliwia dostęp do kompletnych danych o praktycznie każdym cieku wodnych na terenie Francji, do informacji nt. obiektów hydrotechnicznych także do innych informacji na temat np. dróg, terenów itd. Od strony technicznej projekt prezentuje się bardzo dobrze, obsługa jest prosta, rozmieszczenie narzędzi ergonomiczne, graficznie również prezentuje się dobrze. Jedyne, do

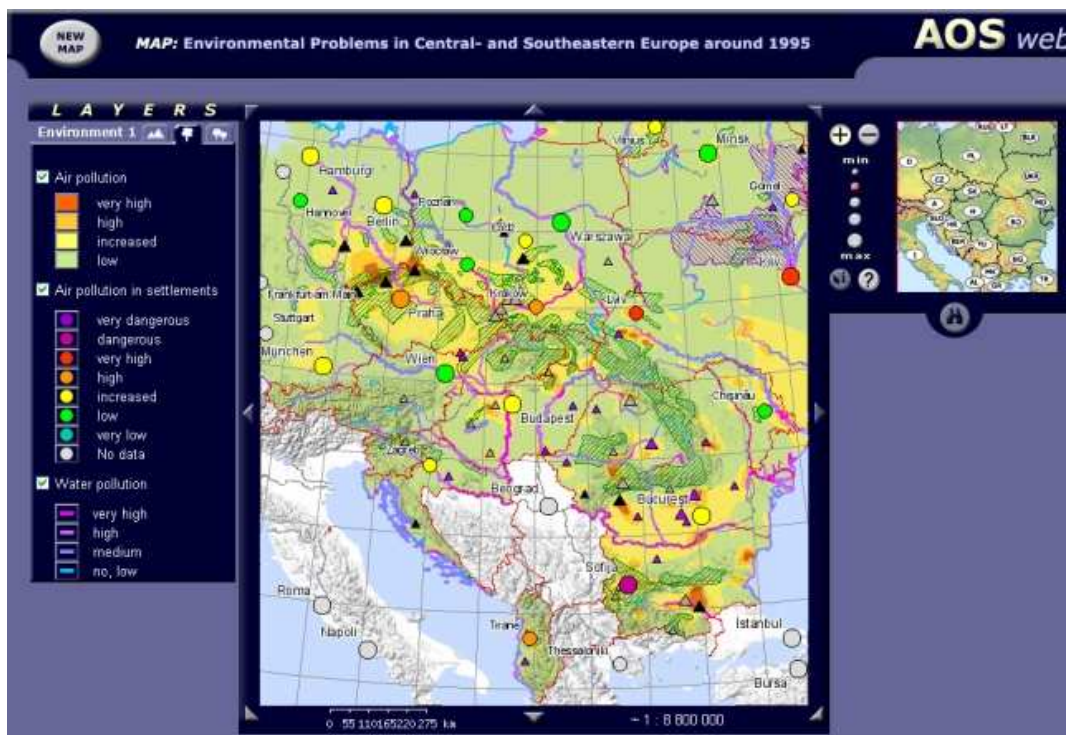
czego można mieć zastrzeżenia to brak angielskich nazw warstw pomimo przełączenia aplikacji na angielską wersję językową. Utrudnia to dostęp do informacji osobom nieznającym języka francuskiego.

- **Atlas of Eastern and SouthEastern Europe**

<http://titan.gis.univie.ac.at/project/atos/htdocs/prototyp/>

Kolejną bardzo ciekawą implementacją UMN Mapserver jest "Atlas of Eastern and SouthEastern Europe". Projekt ten jest efektem współpracy Austrian Institute of East and Southeast European Studies oraz University of Vienna. Obecnie w ramach tego projektu dostępne są dwie mapy tematyczne:

- „Environmental Problems in Central and Southeastern Europe around 1995” – mapa ukazująca stopień zanieczyszczenia środowiska w Europie południowo-wschodniej i obejmująca swoim zasięgiem obszar min. Polski.
- „International Tourism Attractions in Central and Southeastern Europe” – mapa zawierająca informacje na temat atrakcji turystycznych Europy południowo-wschodniej.



**Rysunek 4. Atlas of Eastern and SouthEastern Europe**  
[<http://titan.gis.univie.ac.at/project/atos/htdocs/prototyp/>]

Projekt jest cały czas rozwijany, na stronie są dostępne zapowiedzi kolejnych map tematycznych, między innymi na temat struktury etnicznej czy klimatu.

Z polskiego punktu widzenia bardzo ciekawa będzie mapa o tytule „Poland as a Source of Migration and Travel”. Zarówno pod względem technicznym jak i graficznym projekt stoi na wysokim poziomie, obsługa jest intuicyjna. Zastrzeżenia można mieć za to do strony merytorycznej, jako że na przykład brakuje kilku dużych polskich miast jak Tarnów, za to znajdziemy na mapie mniejsze miasteczka jak Będzin czy Pruszków.

- **Bundaberg City**

*<http://bundaberg.qld.gov.au/applications/map/>*

Projekt ten jest przykładem zastosowania MapServera do stworzenia serwisu miejskiego na potrzeby lokalnej społeczności. Stworzony na zamówienie Bundaberg City Council projekt pozwala na dostęp do praktycznych informacji na temat miasta czy dzielnicy. Wykorzystując tę aplikację użytkownik może uzyskać informacje między innymi na temat:

- stref zalewowych pobliskiej rzeki
- rodzajów dróg
- lokalizacji parków, szkół, szpitali, jednostek policji i straży pożarnej.
- sieci kanalizacyjnej i wodociągowej.
- planu zagospodarowania poszczególnych obszarów
- porządku wywożenia śmieci z poszczególnych obszarów.

UMN MapServer jako platforma WebGIS  
na przykładzie RZGW w Krakowie

Layers

**Features**

- Suburbs
- River
- Parks
- Schools
- Emergency Services
- Property Desc.
- Burnett Rv Flood Area
- Planning Scheme
- Recycling Days
- 2005 Aerial Photos

**Roads**

- House Numbers
- Major Roads
- 50K/h Zones

**Reticulation**

- Sewer Lines
- Water Mains
- Stormwater Mains

Apply

**Feed Back**

Please send any comments by e-mail to the [GIS Section](#)

We recommend viewing this page with a standards compliant web browser such as Mozilla FireFox or compatible.

Powered by MapServer

GetFirefox

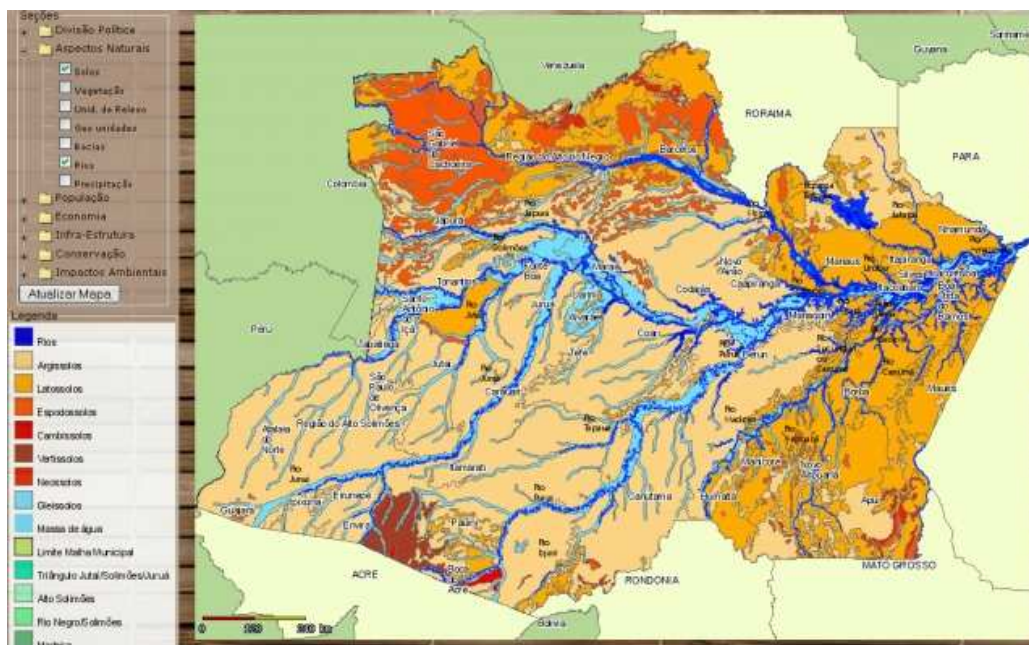
Rysunek 5. Bundaberg City [<http://bundaberg.qld.gov.au/applications/map/>]

Aplikacja jest przejrzysta i łatwa w obsłudze, jednak możliwości manipulacji skalą mapy są ograniczone do 4 z góry zdefiniowanych wartości, brakuje także możliwości zaznaczania interesującego użytkownika obszaru.

- **Atlas Amazonas**

<http://siglab.inpa.gov.br/atlasamazonas/index.php>

Atlas Amazonas jest aplikacją prezentującą informacje geograficzne na temat brazylijskiego stanu Amazonas. Informacje te obejmują podziały administracyjne, infrastrukturę, dane ekonomiczne, dane o populacji, dane o środowisku naturalnym i wiele innych.



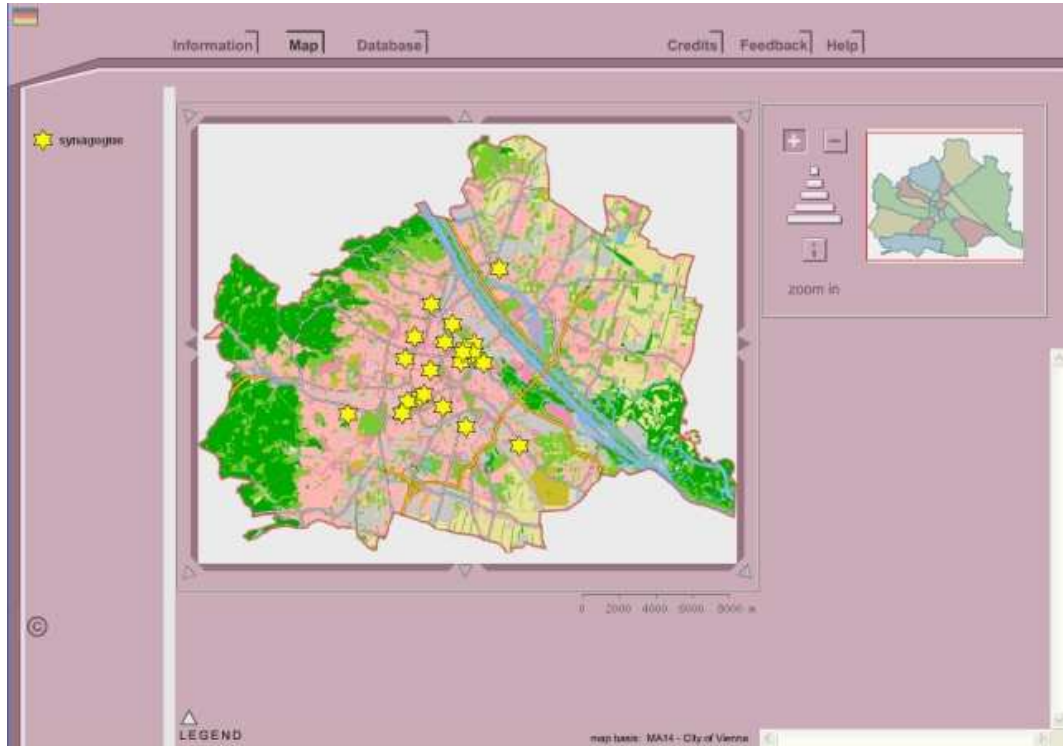
**Rysunek 6. Atlas Amazonas** [<http://siglab.inpa.gov.br/atlasamazonas/index.php>]

Aplikacja ta zbudowana jest w oparciu o UMN MapServer, Postgis oraz PHP Mapscript. Graficznie aplikacja wygląda dobrze, chociaż wydaje się, że tło strony jest niepotrzebne. Interfejs użytkownika jest przejrzysty a informacje dobrze widoczne. Problemem jest tylko brak angielskiej wersji językowej utrudniający pracę nieznaną języka portugalskiego użytkownikom.

- **Reason for Eviction - Non-Ayrian**

<http://titan.gis.univie.ac.at/project/kna-wien/>

Aplikacja ta stanowi doskonały przykład wykorzystania UMN MapServer do prezentacji danych historycznych. Umożliwia użytkownikowi zapoznanie się z historią przedwojennego Wiednia i losami jego żydowskich mieszkańców.



**Rysunek 7. Reason for Eviction - Non-Ayrian**  
[<http://titan.gis.univie.ac.at/project/kna-wien/>]

Część opisowa zawiera wyczerpujące informacje na temat większości aspektów życia w przedwojennym Wiedniu, ze szczególnym naciskiem na kwestie żydowskie. Z części opisowej można także dowiedzieć się także o przyczynach zagłady Żydów, nazistowskiej ideologii czy warunkach panujących w obozach, do których Żydzi zostali deportowani. Aplikacja składa się z 3 projektów:

- Project Reason for Eviction - non Ayrian, Jews in Hietzing – projekt prezentujący miejsca skąd deportowano żydów, zawierający także listę wiedeńskich żydów oraz historię ich dalszych losów po eksmisji i deportacji do innych miejsc.
- Project Spiegelgrund – projekt umożliwiający poznanie losów żydowskich dzieci, zawiera także mapę miejsc skąd deportowano dzieci do obozów.

---

- Project Synagogues in Vienna – projekt mający na celu przybliżenie użytkownikowi historii i lokalizacji synagog wiedeńskich. Oprócz mapy z lokalizacją synagog dostępny jest szczegółowy opis każdej z nich, w większości zawierający zdjęcia lub szkice budynków.

Zarówno od strony technicznej jak i graficznej aplikacja wygląda bardzo dobrze. Obsługa jest intuicyjna, uzyskanie interesujących użytkownika informacji zajmuje bardzo mało czasu. Dostępne są 2 wersje językowe: niemiecka i angielska.

## 5. Projekt aplikacji dla RZGW w Krakowie

### 5.1 Założenia do projektu

Celem projektu było zaprojektowanie i stworzenie na potrzeby Regionalnego Zarządu Gospodarki Wodnej w Krakowie aplikacji służącej do analizy i prezentacji danych hydrograficznych zgromadzonych w zasobach tej instytucji.

Do podstawowych wymogów, jakie aplikacja powinna spełniać w opinii potencjalnych użytkowników należały:

- Aplikacja powinna umożliwiać użytkownikom dostęp do map i danych poprzez wewnętrzną sieć Intranet oraz, przy zastosowaniu systemów autentykacji, dostępu dla uprawnionych użytkowników poprzez sieć internet.
- Powinna być dostępna dla użytkownika niezależnie od wykorzystywanego przez niego systemu operacyjnego.
- Ze względu na zróżnicowanie konfiguracji sprzętowych komputerów całe przetwarzanie danych powinno być dokonywane na serwerze a aplikacja klienta powinna tylko wyświetlić rezultat tego przetwarzania.
- Aplikacja powinna móc wykorzystywać dane w formatach wektorowych (ESRI ShapeFile) a także rastrowych (JPG,TIFF/GeoTIFF,PNG, GRID)
- Aplikacja musi mieć możliwość współpracy z systemami bazodanowymi MySQL, PostgreSQL a także Oracle
- Powinna mieć możliwość współpracy z oprogramowaniem ArcSDE.

### 5.2 Dane wykorzystane w aplikacji

W ramach testowego uruchomienia aplikacji wykorzystane zostały dane źródłowe uzyskane dzięki uprzejmości RZGW w Krakowie. Są to warstwy wektorowe w postaci plików ESRI Shapefile oraz rastrowy podkład w formacie GeoTIFF.

Warstwy wektorowe zawierają informacje na temat stref zalewowych rzeki Koszarawy:

- zasięg zalewu dla wody o prawdopodobieństwie przewyższenia 0,2%
- zasięg zalewu dla wody o prawdopodobieństwie przewyższenia 1%
- zasięg zalewu dla wody o prawdopodobieństwie przewyższenia 3,33%



- zasięg zalewu dla wody o prawdopodobieństwie przewyższenia 5%
- zasięg zalewu dla wody o prawdopodobieństwie przewyższenia 10%
- zasięg zalewu dla wody o prawdopodobieństwie przewyższenia 20%
- zasięg zalewu dla wody o prawdopodobieństwie przewyższenia 50%

Podkład rastrowy jest to mapa topograficzna obejmująca swoim zasięgiem obszar gminy Koszarawa.

Poza warstwami udostępnionymi przez RZGW wykorzystana została również wektorowa warstwa, zawierająca lokalizacje stacji monitoringu systemu ostrzegania przed podwoziami Starostwa Powiatowego w Żywcu. Warstwa została opracowana we własnym zakresie na podstawie danych udostępnianych przez Starostwo.

Docelowo aplikacja może wyświetlić dane przestrzenne w każdym formacie wektorowym oraz rastrowym obsługiwanych przez UMN MapServer.

### **5.3 Środowisko implementacji**

Część aplikacji będąca serwerem danych została uruchomiona na komputerze testowym klasy PC o następującej konfiguracji:

- Procesor AMD Athlon64 3000+
- 1024 MB RAM DDR2 667 MHz
- Dysk Twardy Seagate Barracuda 250 GB
- System Operacyjny Windows XP Professional PL
- MySQL Community Server 5.0
- Pakiet MS4W 2.2.4, w skład którego wchodzi:
  - Apache HTTP Server 2.2.4
  - PHP 5.2.3
  - MapServer CGI 4.10.2
  - MapScript 4.10.2
  - GDAL/OGR 1.4.1
  - OGR/PHP Extension 1.0.0
  - Różnorodne narzędzia do konwersji typów danych oraz ich wyświetlania.

MS4W jest pakietem aplikacji i narzędzi pozwalającym uruchomić UMN MapServer na platformie MS Windows. Zawiera wszystkie składniki wymagane do prawidłowego działania MapServera ułatwiając użytkownikowi szybkie uruchomienie stabilnej i niezawodnej platformy udostępniania map w sieciach intranet/internet. Ważną jego zaletą jest zastosowanie w nim najnowszych wersji serwera Apache, języka skryptowego PHP oraz UMN MapServer. Na serwerze, na którym uruchomiona została aplikacja-serwer zainstalowana jest wersja 2.2.4 pakietu MS4W, udostępniona użytkownikom w dniu 6 czerwca 2007.

Poniżej omówione zostały najważniejsze technologie i aplikacje wykorzystane przy tworzeniu aplikacji:

### 5.3.1 HTML

HTML (*Hyper Text Markup Language*) jest językiem programowania opartym na znacznikach, stosowanym do tworzenia stron WWW. Początkowo każdy producent przeglądarek WWW korzystał ze swojego Standardu HTML, jednak zazwyczaj były one niekompatybilne ze sobą nawzajem. Rozwiązaniem tego problemu było stworzenie organizacji WorldWideWeb Consortium, w skrócie W3C, której zadaniem jest ustalanie standardów HTML oraz rozwój tego języka. Podstawowym elementem składowym języka HTML są znaczniki, zwane także tagami. Podstawowa struktura znacznika wygląda następująco:

- Znacznik początkowy, zazwyczaj w formie <znacznik>
- Zawartość elementu, może to być tekst lub inne znaczniki
- Znacznik końcowy, zazwyczaj w formie </znacznik>

Część znaczników nie posiada znacznika końcowego, są to tak zwane znaczniki samozamykające się. Mają one postać <znacznik/>. Atrybuty znaczników, określające nazwę elementu, jego wygląd i zachowanie umieszcza się wewnątrz znacznika początkowego w postaci <znacznik atrybut="wartość" atrybut2="wartość2"... atrybutN="wartośćN">. Przykładowy znacznik, w tym przypadku komórka tabeli, może wyglądać w taki sposób:

```
<td border=1 width=100> tekst </td>
```

Zapis taki oznacza, że komórka tabeli będzie miała szerokość 100 pikseli oraz będzie obramowana.

W HTML stosowane jest „zagnieżdżanie” znaczników, tzn. umieszczanie znaczników wewnątrz innych. Doskonałym przykładem tego jest definicja tabeli w HTML:

```
<table border=1>
  <tr>
    <td> tekst w komórce pierwszej wiersza pierwszego</td>
    <td> tekst w komórce drugiej wiersza pierwszego</td>
  </tr>
  <tr>
    <td> tekst w komórce pierwszej wiersza drugiego</td>
    <td> tekst w komórce drugiej wiersza drugiego</td>
  </tr>
</table>
```

Przykładowy dokument HTML wygląda następująco:

```
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html;
    charset=UTF-8">
    <title>Tytuł strony w HTML-u</title>
  </head>
  <body>
    <table border=1>
      <tr>
        <td> tekst w komórce pierwszej wiersza
        pierwszego</td>
        <td> tekst w komórce drugiej wiersza
        pierwszego</td>
      </tr>
      <tr>
        <td> tekst w komórce pierwszej wiersza
        drugiego</td>
        <td> tekst w komórce drugiej wiersza
        drugiego</td>
      </tr>
    </table>
  </body>
</html>
```

gdzie:

- znaczniki `<html>` i `</html>` oznaczają początek i koniec kodu html
- znaczniki `<head>` i `</head>` oznaczają początek i koniec sekcji head, w której określany jest typ dokumentu, jego kodowanie, tytuł, słowa kluczowe czy w końcu dołączane są style CSS oraz skrypty JS.
- znaczniki `<body>` i `</body>` zawierają sekcję body, która jest treścią dokumentu

Obecnie większość stron WWW zawiera kod HTML generowany dynamicznie przez skrypty napisane w różnych językach programowania takich jak PHP, Java lub ASP.

### 5.3.2 PHP

PHP jest skryptowym językiem programowania wykorzystywanym do dynamicznego generowania stron internetowych. Najczęściej jest używany po stronie serwera do interpretowania skryptów i na ich podstawie generowania stron, aczkolwiek może być także wywoływany z linii poleceń systemu operacyjnego oraz może być wykorzystywany w aplikacjach pracujących w trybie graficznym poprzez bibliotekę GTK+.

„ PHP może być także użyty w większości najważniejszych systemów operacyjnych, takich jak Linux, wiele wariantów systemu Unix (włączając w to HP-UX, Solaris i OpenBSD), Microsoft Windows, Mac OS X, RISC OS i prawdopodobnie wiele innych. PHP w chwili obecnej obsługuje większość serwerów HTTP, włączając w to Apache, Microsoft Internet Information Server, Personal Web Server, serwery Netscape i iPlanet, Oreilly Website Pro, Caudium, Xitami, OmniHTTPd i wiele innych. Dla większości z nich PHP dostępne jest jako moduły serwera, dla pozostałych jako program CGI. PHP może pracować jako procesor CGI.”  
[pl2.php.net, 2007]

Zaletą PHP jest także mnogość formatów wyjściowych. Oprócz generowania stron HTML możliwe jest także tworzenie przy jego użyciu plików PDF, obrazów czy nawet

animacji flash generowanych „w locie”. Możliwe jest także zapisywanie wyniku działania skryptu na dysku zamiast zwracania go poprzez standardowe wyjście.

„Jedną z najmocniejszych i najbardziej znaczących możliwości PHP jest obsługa wielu rodzajów baz danych. Pisanie strony WWW wykorzystującej bazę danych jest niewiarygodnie proste. Obecnie obsługiwane są następujące bazy danych:

Adabas D	InterBase	PostgreSQL
dBase	FrontBase	SQLite
Empress	mSQL	Solid
FilePro (tylko do odczytu)	Direct MS-SQL	Sybase
Hyperwave	MySQL	Velocis
IBM DB2	ODBC	Unix dbm
Informix	Oracle (OCI7 i OCI8)	
Ingres	Ovrimos	

Istnieje także abstrakcyjne rozszerzenie DBX pozwalające na przezroczyste używanie dowolnej bazy danych obsługiwanych przez to rozszerzenie. Dodatkowo PHP obsługuje standard ODBC (Open Database Connection), przez co możesz połączyć się do dowolnej innej bazy danych obsługującej ten popularny standard.” [pl2.php.net ,2007]

Funkcjonowanie PHP na serwerze WWW opiera się na 4 grupach modułów:

- **Moduł Jądra** – silnik PHP, zawsze aktywny na serwerze
- **Moduły oficjalne** – elementy znajdujące się w każdej dystrybucji PHP, mogą być aktywowane i dezaktywowane przez administratora serwera.
- **Repozytorium PECL** – zbiór modułów i rozszerzeń stworzonych przez użytkowników z całego świata i rozpowszechnianych na licencji Open Source. Od wersji PHP 5 do repozytorium przeniesiono także rzadko używane lub niestabilne moduły oficjalne.
- **Repozytorium PEAR** – ujednoczony system dystrybucji darmowych rozszerzeń i modułów dla PHP. Moduły te są dystrybuowane jako tzw. Paczki ( package ) co ułatwia ich instalację i dołączenie do PHP. PEAR definiuje także zależności pomiędzy paczkami, informując użytkownika o modułach i rozszerzeniach wymaganych do pracy przez instalowany przez niego moduł.

---

Składnia języka PHP oparta jest o składnie języków C, Java oraz PERL. Przykładowy fragment kodu PHP wygląda następująco:

```
<?php
$a = 5;
$b = 3;

function suma($a, $b)
{
    return $a + $b;
}

echo suma($a, $b);

?>
```

Wynikiem wywołania powyższego kodu jest wypisanie na ekranie liczby 8.

### 5.3.3 MySQL

MySQL jest najpopularniejszym systemem bazodanowym dostępnym na licencji Open Source. Został stworzony przez szwedzką firmę MySQL AB. Głównymi zaletami tego systemu są jego wydajność, szybkość działania a także niezawodność i stabilność. Dzięki temu jest wykorzystywany do większości zadań, w których zachodzi potrzeba wykorzystania baz danych. Duża część internetowych serwisów WWW opiera się na MySQL..

## 5.4 Funkcjonalność aplikacji

Poszczególne funkcje aplikacji realizowane są przez 2 oddzielne moduły, mogące pracować niezależnie od siebie. Moduły te to:

- Moduł serwera map
- Moduł interfejsu użytkownika oparty o stronę WWW.

Dzięki takiemu podziałowi możliwe jest wykorzystywanie serwera map także przez inne aplikacje, zarówno komercyjne jak i darmowe. Możliwe jest również dostosowanie niewielkim nakładem pracy interfejsu użytkownika do pracy przy wykorzystaniu danych udostępnianych przez inne serwery.

### 5.4.1 Moduł Serwera

Moduł serwera odpowiada za przechowywanie danych źródłowych, zarówno w formacie wektorowym jak i rastrowym oraz udostępnianie ich modułowi interfejsu użytkownika. Realizowane jest to poprzez omówiony w poprzednim rozdziale UMN MapServer. Serwer odbiera od modułu interfejsu użytkownika żądanie, zawierające informacje o skali mapy, obszarze mapy, warstwach które mają się na niej znaleźć i na tej podstawie generuje mapę wynikową.

Informacje na temat lokalizacji danych źródłowych dla każdej z warstw, sposobu wyświetlania warstw (między innymi kolor warstwy, minimalna i maksymalna skala, w jakiej jest ona wyświetlana, styl czcionki etykiet obiektów), wygląd skali mapy oraz legendy a także początkowa konfiguracja mapy przechowywane są w pliku Map File.

Poniżej przedstawione zostały najważniejsze fragmenty pliku Map File wykorzystywanego w aplikacji:

```
MAP
  NAME Koszarawa
  #rozmiar w pikselach generowanej mapy
  SIZE 600 400
  #format w którym generowana jest mapa
  IMAGETYPE GIF
  #kolor tła R G B
  IMAGECOLOR 255 255 255
  #ścieżka do katalogu z danymi źródłowymi
  SHAPEPATH "/ms4w/Apache/htdocs/Koszarawa/"
  #początkowy obszar generowanej mapy
  EXTENT 514100 193981 531337 202212
  #jednostka mapy
  UNITS METRES
  .
  .
  .
  #definicja warstwy rastrowej
  LAYER
    NAME "172-343"
    TYPE RASTER
    STATUS ON
```

---

```
DATA "172-343.tif"

CLASSITEM "[pixel]"
END
.
.
.
#definicja warstwy wektorowej
LAYER
    NAME "granica_zalewu_koszarawa_1"
    TYPE LINE
    STATUS ON
    DATA "granica_zalewu_koszarawa_1"
    CLASS
        NAME "granica_zalewu_1"
        STYLE
            OUTLINECOLOR 175 54 59
            WIDTH 3
            ANTIALIAS TRUE
        END
        SYMBOL 0
    END
END
.
.
.
#definicja skali mapy
SCALEBAR
    TRANSPARENT ON
    BACKGROUNDCOLOR 254 254 254
    STATUS EMBED
    POSITION LL
    STYLE 1
    UNITS KILOMETERS
END
#koniec pliku
END
```

Za przetwarzanie żądań klienta na serwerze odpowiada jeden z modułów MapServera – PHP Mapscript. Przy każdym przesłaniu żądania przez interfejs



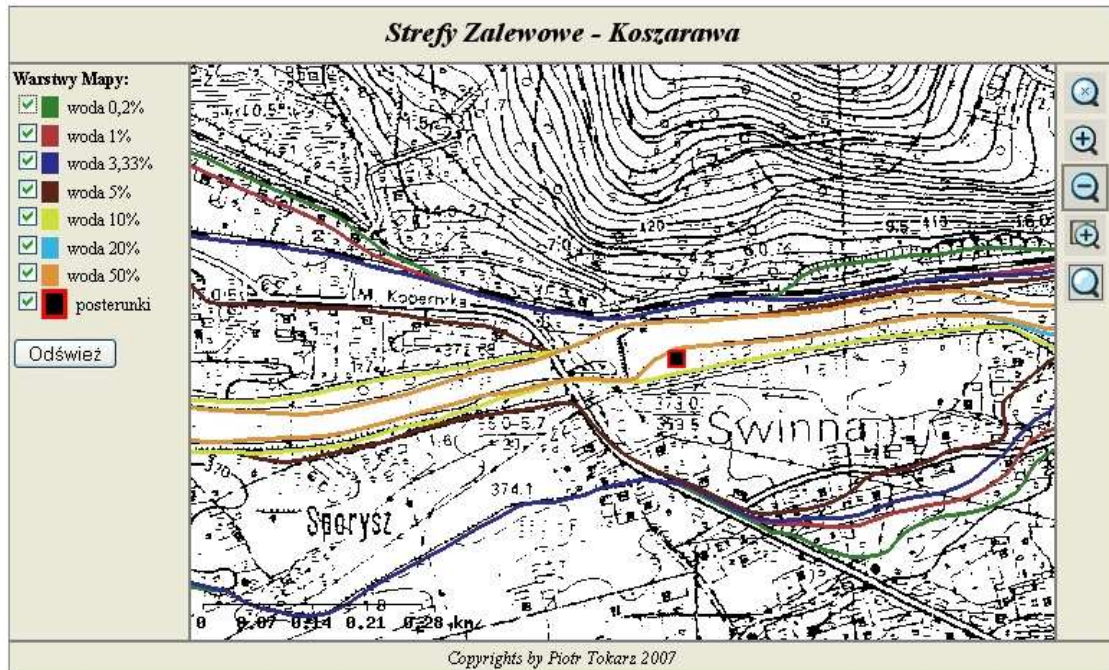
użytkownika tworzony jest na podstawie pliku Map File obiekt klasy MapObj a następnie przy użyciu metod tej klasy ustawiany jest obszar wynikowej mapy, wyłączane lub włączane są poszczególne warstwy i dokonywane są innego typu zmiany wyglądu wynikowej mapy, w zależności od zawartości żądania klienta. Wykorzystanie metod tej klasy zostało opisane szczegółowo przy omawianiu funkcji interfejsu użytkownika w dalszej części rozdziału.

#### **5.4.2 Moduł Interfejsu Użytkownika**

Interfejs użytkownika został stworzony w oparciu o stronę WWW udostępnianą w sieci przy użyciu serwera WWW Apache. Dzięki takiemu rozwiązaniu aplikacja jest niezależna od systemu operacyjnego wykorzystywanego przez użytkownika jak i konfiguracji sprzętowej jego komputera. Znacząco mniejsze są także wymagania sprzętowe aplikacji, możliwe jest uruchomienie jej nawet na komputerach wyprodukowanych przed kilkunastu laty.

## Wygląd interfejsu użytkownika

Interfejs graficzny aplikacji jest podzielony na 3 sekcje umieszczone w jednej linii. Po lewej stronie znajduje się menu z wyborem dostępnych warstw mapy, następnie zajmująca większość obszaru roboczego mapa wygenerowana na serwerze na podstawie informacji przesłanych w żądaniu z aplikacji, zaś po prawej stronie umieszczony jest pasek narzędzi służących do manipulowania obszarem mapy.



Rysunek 8. Aplikacja – widok ogólny

Takie rozmieszczenie kontrolek zapewnia dużą ergonomię interfejsu i powoduje, że aplikacja jest prosta w obsłudze nawet dla początkującego użytkownika a wszystkie funkcje aplikacji są łatwo dostępne.

## Metoda implementacji interfejsu użytkownika

Interfejs użytkownika oparty jest o funkcje języka skryptowego PHP, przy pomocy których na serwerze WWW generowany jest kod HTML i JavaScript umożliwiający wyświetlenie strony w przeglądarce użytkownika.. Aplikacja działa w oparciu o formularz HTML, którego głównym elementem jest rastrowy obraz mapy będący jednocześnie przyciskiem wysyłającym dane z formularza do serwera. Wygląd strony jest skonfigurowany poprzez użycie Kaskadowych Arkuszy Styli (CSS), co ułatwia jego ewentualne późniejsze zmiany. Do realizacji zaawansowanych operacji na formularzu wykorzystany został język JavaScript.

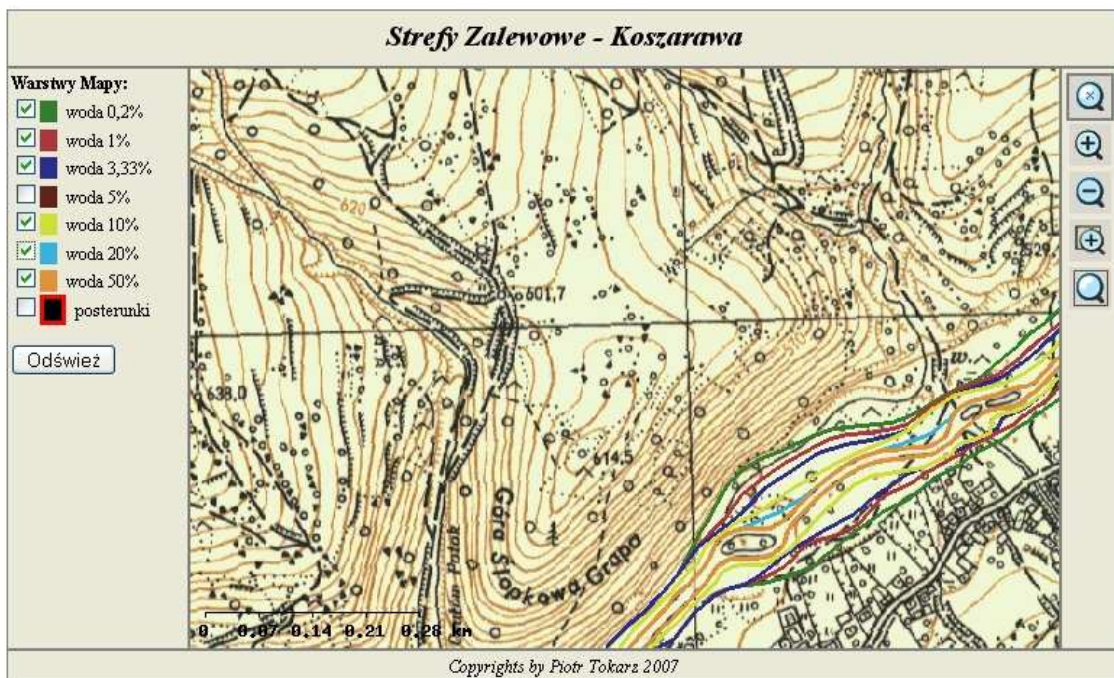
### 5.4.3 Funkcje dostępne w aplikacji

Aplikacja oferuje różnorodne możliwości manipulowania zarówno zawartością jak i sposobem prezentacji danych przestrzennych. Możliwości te zostały omówione szczegółowo poniżej.

#### Wybór warstw wyświetlanych na mapie

W lewej części okna aplikacji dostępna jest lista warstw, które mogą zostać wyświetlone na mapie. Przy nazwie każdej warstwy dostępny jest prostokąt w kolorze, w jakim jest ona na mapie wyświetlana. Kolorów tych nie można zmienić inaczej jak tylko poprzez edycję pliku Map File. Obok koloru dostępne jest także pole typu checkbox określające czy warstwa jest wyświetlana w danej chwili.

Aby włączyć / wyłączyć wyświetlanie warstwy na mapie należy zaznaczyć / odznaczyć pole checkbox przy jej nazwie a następnie kliknąć przycisk „Odśwież”. Spowoduje to wysłanie żądania do serwera i wygenerowanie mapy na nowo z warstwą włączoną / wyłączoną.



Rysunek 9. Aplikacja – wybór warstw

## Opcja „Wycentruj”

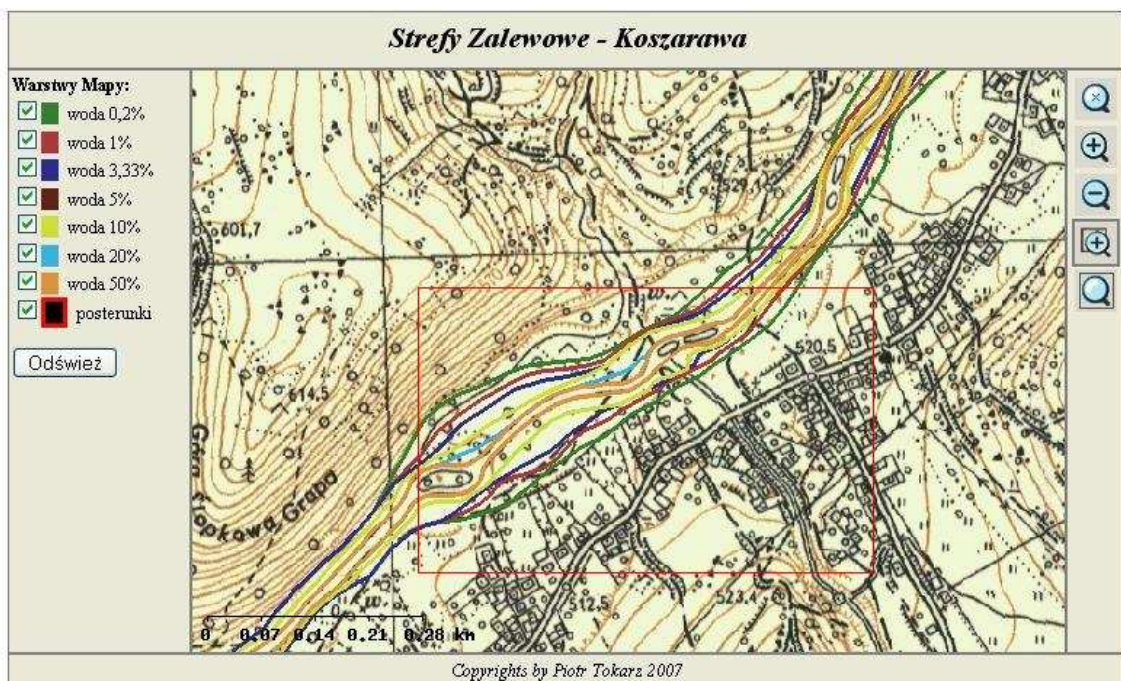
Pierwszym przyciskiem od góry na pasku narzędzi po prawej stronie okna aplikacji jest przycisk „Wycentruj”. Jest on domyślnie wybrany przy uruchamianiu aplikacji. Wciśnięcie go powoduje ustawienie aplikacji w trybie przesuwania środka wyświetlanej mapy, bez zmiany skali, w jakiej jest ona wyświetlana. W trybie tym kliknięcie na obszarze mapy powoduje przeładowanie aplikacji i wygenerowanie przez serwer mapy ze środkiem ustawionym w miejscu kliknięcia.

## Opcje „Zbliź” i „Oddal”

Kolejnymi przyciskami na pasku narzędzi są przyciski „Zbliź” i „Oddal”. Przełączają one aplikację w tryb zmieniania skali mapy. Po wybraniu któregoś z tych przycisków i kliknięciu na obszarze mapy zostanie wygenerowana nowa mapa, której skala będzie odpowiednio 2 razy mniejsza lub większa od poprzedniej a mapa zostanie wycentrowana w punkcie kliknięcia.

## Opcja „Pokaż obszar”

Czwarty z przycisków dostępnych na pasku narzędzi to przycisk „Pokaż obszar”. Wybranie go skutkuje przejściem aplikacji w tryb zaznaczania obszaru mapy. W trybie tym użytkownik może wybrać interesujący go obszar mapy poprzez zaznaczenie go na mapie. Po zaznaczeniu obszaru na mapie aplikacja wygeneruje na nowo mapę obejmującą zasięgiem tylko obszar interesujący użytkownika.



Rysunek 10. Aplikacja – funkcja „Pokaż obszar”

## Opcja „Pokaż wszystko”

Ostatnim z przycisków dostępnych na pasku narzędzi jest przycisk „Pokaż wszystko”. Po wybraniu tej opcji aplikacja generuje ponownie mapę z obejmującą swym zasięgiem cały dostępny obszar.

### 5.4.4 Techniczne aspekty implementacji

W punkcie tym przedstawione zostały rozwiązania programistyczne wykorzystane do implementacji wymienionych wyżej funkcjonalności. W większości oparte są one o wykorzystanie istniejących funkcji dostępnych w PHP MapScript.

#### Generowanie Mapy

W momencie uruchomienia lub przeładowania (poprzez wysłanie żądania do serwera związanego z kliknięciem na mapie lub zmianą listy wyświetlanych warstw) aplikacji skrypt php na serwerze wywołuje konstruktor klasy **mapObj**.

```
$map = ms_newMapObj($map_file);
```

którego argumentem jest zmienna

```
$map_file="./test2.map";
```

zawierająca względną lub bezwzględną ścieżkę do pliku Map File zawierającego informacje ustawień początkowych mapy oraz źródeł danych. Plik ten został omówiony dokładnie w punkcie 5.4.1.

Następnie dokonywane są wszelkie zmiany obiektu mapy i jej składowych, poprzez wywołanie odpowiednich funkcji omówionych w dalszej części punktu. Po dokonaniu wszystkich zmian na obiekcie mapy wywoływane są funkcje

```
$image=$map->draw();  
$image_url=$image->saveWebImage();
```

które odpowiednio generują na podstawie obiektu mapy obraz rastrowy w odpowiednim (zadeklarowanym w pliku Map File) formacie oraz zapisują ten obraz w katalogu tymczasowym serwera http skąd obraz jest wysyłany do przeglądarki użytkownika.

W przypadku początkowego uruchomienia aplikacji, gdy nie są dokonywane żadne operacje na obiekcie mapy funkcje te są wywoływane tuż po utworzeniu obiektu mapy.

## Operacje na mapie

Pomiędzy utworzeniem obiektu mapy a wygenerowaniem na jego podstawie rastrowego obrazu mapy dokonywane są wszystkie zmiany na tym obiekcie. Do zmian tych należą przede wszystkim wszelkie zmiany skali, w której mapa jest wyświetlana czy aktualnie udostępnianego użytkownikowi obszaru mapy. Wykorzystywane są do tego następujące funkcje PHP Mapscript:

**ms\_newrectObj()** – konstruktor klasy **rectObj**, klasa wykorzystywana jest do ustawienia współrzędnych obecnego / żądanego obszaru mapy. Obiekt tej klasy jest jednym z argumentów funkcji **zoompoint()** i **zoomrectangle()**.

**rectObj::setextent(double minx, double miny, double maxx, double maxy)** – funkcja klasy **rectObj**, ustawiająca współrzędne obszaru.

**ms\_newpointObj()** – konstruktor klasy **pointObj**, klasa wykorzystywana jest do wskazania punktu kliknięcia na obrazie mapy.

**pointObj::setXY(double x, double y)** – funkcja klasy **pointObj**, wykorzystywana do ustawienia współrzędnych kliknięcia (współrzędnych obrazu a nie współrzędnych mapy).

**mapObj::zoompoint(int nZoomFactor, pointObj oPixelPos, int nImageWidth, int nImageHeight, rectObj oGeorefExt)** - funkcja klasy **mapObj**, wykorzystywana do przybliżenia / oddalenia mapy ze środkiem w punkcie kliknięcia. Argumentami tej funkcji są:

**nZoomFactor** – współczynnik skali

**oPixelPos** – punkt kliknięcia.

**nImageWidth , nImageHeight** – szerokość wysokość obrazu.

**oGeorefExt** – współrzędne obecnego obszaru mapy.

---

**mapObj::zoomrectangle(rectObj oPixelExt, int nImageWidth, int nImageHeight, rectObj oGeorefExt)** - funkcja klasy **mapObj**,

wykorzystywana do przybliżenia / oddalenia mapy ze wskazaniem obszaru.

Argumentami tej funkcji są:

**oPixelExt** – współrzędne nowego obszaru mapy.

**nImageWidth , nImageHeight** – szerokość wysokość obrazu.

**oGeorefExt** – współrzędne obecnego obszaru mapy.

### Operacje na warstwach mapy

Operacje na warstwach mapy również są wykonywane pomiędzy utworzeniem obiektu mapy a wygenerowaniem rastrowego jej obrazu. Należą do nich wszelkie zmiany wyglądu warstw, ich sposobu wyświetlania i dostępności. Realizowane jest to przy użyciu następujących funkcji PHP MapScript:

**mapObj::getAllLayersName()** - funkcja klasy **mapObj**, wykorzystywana do pobrania listy warstw dostępnych na mapie.

**mapObj::getLayer(int index)** - funkcja klasy **mapObj**, która zwraca warstwę o indeksie **index** w postaci obiektu klasy **layerObj**.

**layerObj::set(string property\_name, new\_value)** - funkcja klasy **layerObj**, wykorzystywana do ustawiania wszystkich właściwości warstwy. Argumentami tej funkcji są:

**property\_name** – nazwa właściwości.

**new\_value** – nowa wartość właściwości.

W powyższym punkcie przedstawione zostały funkcje i klasy wykorzystywane w aplikacji do dokonywania operacji na zawartości mapy i danych źródłowych. Pozwalają one na modyfikacje w znacznym stopniu wyglądu i sposobu prezentacji danych źródłowych na mapie. Dzięki bogatej dokumentacji i dobrze napisanemu spisowi dostępnych funkcji, możliwa jest także stosunkowo szybka implementacja innych możliwości operowania mapą.

## 5.5 Zalety i wady aplikacji

W punkcie tym opisane zostały korzyści płynące z używania aplikacji jak i ewentualne problemy z tego wynikające.

Niewątpliwą zaletą wykorzystania aplikacji jest redukcja kosztów dostępu do danych przestrzennych, gdyż aplikacja do działania wymaga jedynie przeglądarki internetowej, nie ma potrzeby zakupu wyspecjalizowanego oprogramowania. Jest to ważne zwłaszcza przy większej liczbie użytkowników gdyż zazwyczaj w przypadku programów komercyjnych na każde stanowisko należy zakupić osobną licencję. Zaletą jest także możliwość dostępu do danych przez wielu użytkowników jednocześnie, co ważne zwłaszcza w dużych instytucjach.

Kolejną zaletą jest zlikwidowanie problemu nieaktualnych danych, oraz ich spójności gdyż wszystkim użytkownikom prezentowane są te same dane. Umożliwia to współdzielenie danych przez użytkowników w centrali instytucji oraz w jej oddziałach terenowych.

Ważnym elementem z punktu widzenia dużych instytucji jest także możliwość wprowadzenia różnych poziomów dostępu do danych, co powoduje zwiększenie bezpieczeństwa. Można wydzielić użytkownikowi dostęp tylko do danych z danego obszaru, lub tylko z dziedziny, nad którą obecnie pracuje. W przypadku aplikacji opartych o serwer WWW jest to proste w realizacji przy pomocy mechanizmów autentykacji dostępu wbudowanych w serwer WWW takich jak htaccess czy filtrowanie IP, jak również przez zastosowanie prostego mechanizmu logowania opartego o sesje czy certyfikaty. Zastosowanie takich mechanizmów pozwala także na udostępnianie przy pomocy aplikacji danych dla organizacji zewnętrznych, w przypadku RZGW mogą to być np. samorządy lokalne.

Oprócz niewątpliwych zalet aplikacja posiada także pewne wady. Do głównych wad należy duży nakład pracy wymagany do uruchomienia takiego systemu. Aby aplikacja spełniała swoje zadanie konieczne jest udostępnienie danych w aplikacji poprzez stworzenie odpowiednich plików Map File, lub, w przypadku większych instytucji jak RZGW, stworzenie systemu do zarządzania i generowania tych plików na bieżąco. Problemem może być także brak możliwości pracy z wykorzystaniem aplikacji w przypadku braku dostępu do sieci lub jej awarii gdyż całość danych znajduje się na serwerze.



## 6. Podsumowanie

Celem pracy było omówienie najciekawszych dostępnych rozwiązań z dziedziny prezentacji danych przestrzennych w Internecie, a następnie stworzenie w oparciu o jedno z nich aplikacji na potrzeby RZGW, która pozwalałaby na udostępnianie danych przestrzennych zarówno w sieci wewnętrznej Intranet oraz, przy odpowiednich modyfikacjach w zakresie autentykacji dostępu użytkowników, pozwalała na korzystanie z niej przez internet.

Na początek opisana została historia systemów GIS, począwszy od lat 50-tych a skończywszy na czasach współczesnych. Pozwoliło to ukazać ogromny postęp jaki dokonał się zarówno w dziedzinie systemów GIS jak i w informatyce. W ramach omówienia współczesnych systemów GIS przedstawiony został także opis organizacji OGC, mającej obecnie największy wpływ na kształt współczesnych systemów GIS, oraz tematów którymi zajmuje się obecnie.

Kolejnym etapem pracy było przedstawienie najciekawszych rozwiązań umożliwiających udostępnianie danych w Internecie, zarówno komercyjnych jak i darmowych. Pozwoliło to na dokonanie wyboru platformy, w oparciu o którą budowana będzie aplikacja. Wybrana została rozpowszechniana na licencji Open Source platforma UMN MapServer, która umożliwia implementację wszystkich funkcjonalności wymaganych przez projekt. Jeśli chodzi o aplikacje darmowe UMN MapServer nie ma praktycznie żadnej konkurencji w tej dziedzinie.

Ważną częścią pracy jest szczegółowy opis UMN MapServer oraz jego najważniejszych części składowych, takich jak pliki Map File i ich struktura oraz moduł PHP MapScript, umożliwiający wykorzystanie funkcji MapServera w aplikacjach opartych o język PHP. Przedstawione zostały także najciekawsze aplikacje stworzone przy użyciu MapServera.

Kolejną ważną częścią pracy było opisanie stworzonej aplikacji oraz sposobu implementacji poszczególnych funkcjonalności przy użyciu funkcji dostępnych w module PHP MapScript.

Wnioskiem płynącym z tej pracy jest to, że rozwiązanie oparte o UMN MapServer byłoby bardzo dobrym sposobem na realizację dostępu do danych

przestrzennych zgromadzonych w zasobach RZGW. Pomimo kilku wad i problemów, jakie mogą wystąpić przy wdrożeniu, UMN MapServer wydaje się rozwiązaniem wartym rozważenia i z powodzeniem mogącym zastąpić obecnie wykorzystywane rozwiązania komercyjne. Za wykorzystaniem platformy UMN MapServer przemawia również fakt, że jest ona z powodzeniem stosowana na zachodzie Europy do realizacji różnorodnych projektów.

Szeroki wachlarz możliwości platformy UMN MapServer pozwala także sądzić, że może ona z powodzeniem zostać wykorzystana do stworzenia aplikacji praktycznie z każdej dziedziny wymagającej dostępu do danych przestrzennych.

## Literatura

**Davis David E.:** *GIS dla każdego*, MIKOM, 2004

**ESRI INC.:** *Understanding GIS*. ESRI, 1995

**Laska M.:** *Systemy informacji przestrzennej*, 1999

**Litwin L., Myrda G.:** *Systemy Informacji Geograficznej*. Helion, 2005

**Longley Paul A., Goodchild Michael F., Maguire David J., Rhind David W.:** *GIS Teoria i praktyka*, PWN, 2006

**Piszczyk K.:** *WebGIS i Webmapping - technologie dla globalnych systemów informacji przestrzennej*, 2007

**Widacki W.:** *Wprowadzenie do systemów informacji geograficznej*, Wydawnictwo TEXT, 1997

**Zapart P.:** *Komputerowe Systemy Informacji Przestrzennej*, Intersoftland, 1994

## Źródła internetowe

ESRI POLSKA	<a href="http://www.esripolska.com.pl/">http://www.esripolska.com.pl/</a>
Gazeta IT	<a href="http://www.gazeta-it.pl">http://www.gazeta-it.pl</a>
GDAL Library	<a href="http://www.gdal.org/">http://www.gdal.org/</a>
GeoForum	<a href="http://geoforum.pl">http://geoforum.pl</a>
Geostrada	<a href="http://geostrada.com">http://geostrada.com</a>
Geoportal	<a href="http://www.geoportal.gov.pl/">http://www.geoportal.gov.pl/</a>
GIS MAZOWSZA	<a href="http://www.gismazowska.pl">http://www.gismazowska.pl</a>
INTERGRAPH	<a href="http://www.intergraph.pl">http://www.intergraph.pl</a>
MapTools	<a href="http://www.maptools.org">http://www.maptools.org</a>
Open Gis Consortium.	<a href="http://www.opengeospatial.org">http://www.opengeospatial.org</a>
Polskie Towarzystwo Informacji Przestrzennej	<a href="http://www.ptip.org.pl">http://www.ptip.org.pl</a>
Portal Województwa Małopolskiego	<a href="http://wrotamalopolski.pl">http://wrotamalopolski.pl</a>
PostGIS	<a href="http://postgis.refrations.net">http://postgis.refrations.net</a>
PROJ	<a href="http://proj.maptools.org">http://proj.maptools.org</a>
RZGW	<a href="http://www.krakow.rzgw.gov.pl/">http://www.krakow.rzgw.gov.pl/</a>
UMN MapServer	<a href="http://mapserver.gis.umn.edu">http://mapserver.gis.umn.edu</a>
W3C (World Wide Web Consortium)	<a href="http://www.w3.org">http://www.w3.org</a>
Wikipedia – Wolna Encyklopedia	<a href="http://pl.wikipedia.org">http://pl.wikipedia.org</a>

---

## Spis rysunków

Rysunek 1. Struktura Mapservera.....	15
Rysunek 2. Struktura Pliku Map File.....	21
Rysunek 3. Sandre GeoViewer.....	30
Rysunek 4. Atlas of Eastern and SouthEastern Europe.....	31
Rysunek 5. Bundaberg City.....	33
Rysunek 6. Atlas Amazonas.....	34
Rysunek 7. Reason for Eviction - Non-Ayrian.....	35
Rysunek 8. Aplikacja – widok ogólny.....	47
Rysunek 9. Aplikacja – wybór warstw.....	48
Rysunek 10. Aplikacja – funkcja „Pokaż obszar”.....	49