



**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 infrastruktur	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

6. ROZWIJANE I PLANOWANE TECHNOLOGIE GEOINFORMACYJNE

Rozdział ten jest przeglądem nowości technologicznych z zakresu geoinformacji. Większość z nich jest obecnie na etapie wstępnych koncepcji, roboczych specyfikacji lub pierwszych eksperymentów. Do głównych kierunków prac można zaliczyć: modelowanie i symulacje w zakresie geoinformacji, wspomaganie podejmowania decyzji z uwzględnieniem geoinformacji i usługi związane z położeniem – OpenLS (*Open Location Services*).

OGC w ramach zakończonego już programu OWS-2 ustaliło listę zagadnień technologicznych, które będą rozwijane w najbliższych latach. Część z nich to problemy ściśle należące do geomatyki:

- usługi w zakresie nazw geograficznych (*Gazetteer Service Specification*),
- usługi transformacji współrzędnych w środowisku WWW (*Web Coordinate Transformation Service*),
- WMS w postaci 3D – *OGC Web Terrain Server (WTS)*,
- geoparser – system udostępniający usługę wyszukiwania w tekstach nazw geograficznych i określania ich położenia.

Inne z tej listy nie należą bezpośrednio do zagadnień informacji geoprzestrzennej, lecz są potrzebne do prawidłowego realizowania usług geoinformacyjnych:

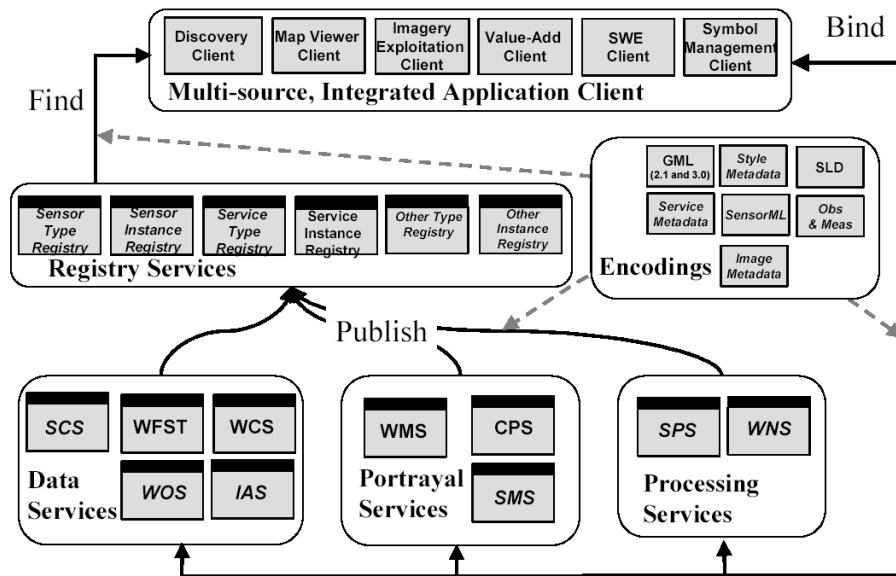
- usługi związane z rejestrowaniem zbiorów geoinformacji w katalogach,
- usługi zawiadamiania – *WNS (Web Notification Services)*,
- *OWS Messaging Framework (OMF)*,
- usługi dotyczące obiektów w WWW (*Web Object Services*),
- zastosowanie SOAP (*Simple Object Access Protocol*) w OWS,
- usługi związane z zarządzaniem stylami – *SMS (Style Management Service)*,
- język opisu modeli sensorów – *SensorML (Sensor Model Language)*,
- usługi związane z zamówieniami i opłatami w zakresie geoinformacji (WPOS) – (*XML Configuration & Pricing Format (XCPF) Specification*),
- aplikacje XML dla opisu obrazów i map (*XML for Imagery and Map Annotations*).

Na końcu tego rozdziału przedstawione są technologie gridowe przeznaczone do współdziałania systemów komputerowych w bezpiecznych powiązaniach za pośrednictwem internetu. Obecnie rozpatrywana jest możliwość zastosowania gridów do rozproszonego przetwarzania geoinformacji.

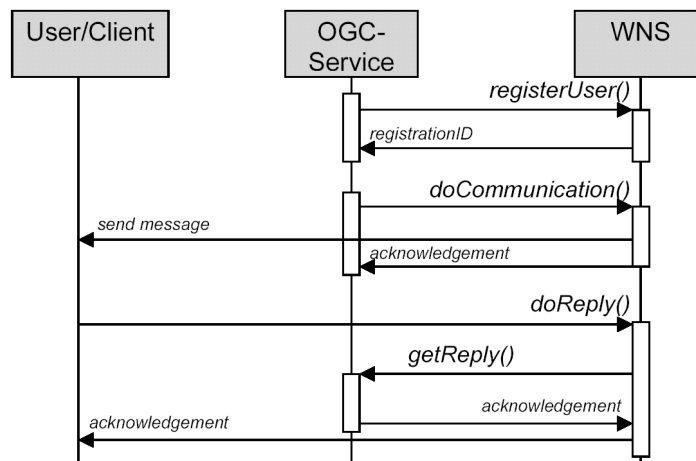
6.1. Integracja usług geoinformacyjnych

Obserwowaną obecnie tendencję w rozwoju technologii geoinformacyjnych można określić jako „interoperacyjność zorientowaną na usługi” (*service-oriented interoperability*). Pojęcie usługi jest wyjaśnione w rozdziałach 2.1.1 i 3.2. Tu będzie przedstawiona złożoność takiego podejścia wynikająca z faktu, że poszczególne usługi są najczęściej ze sobą powiązane, czasami wielokrotnie i w rezultacie można mówić o „łańcuchach usług” jako o pewnym ciągu

złożonych operacji. Rysunek 85 przedstawia wzajemne zależności pomiędzy usługami, a rysunek 86 zawiera diagram sekwencji opisujący kolejność działań tworzących łańcuch usług.



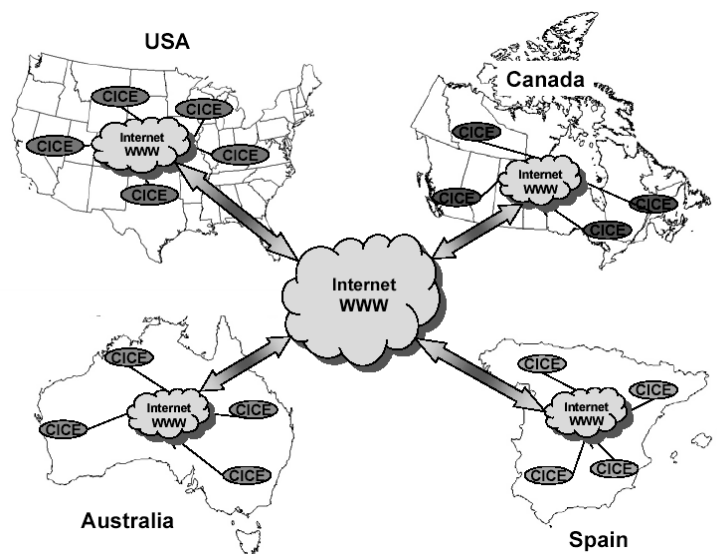
Rys. 85. Opis realizacji usług i ich wzajemne powiązania. [Źródło: Archiwum OGC]



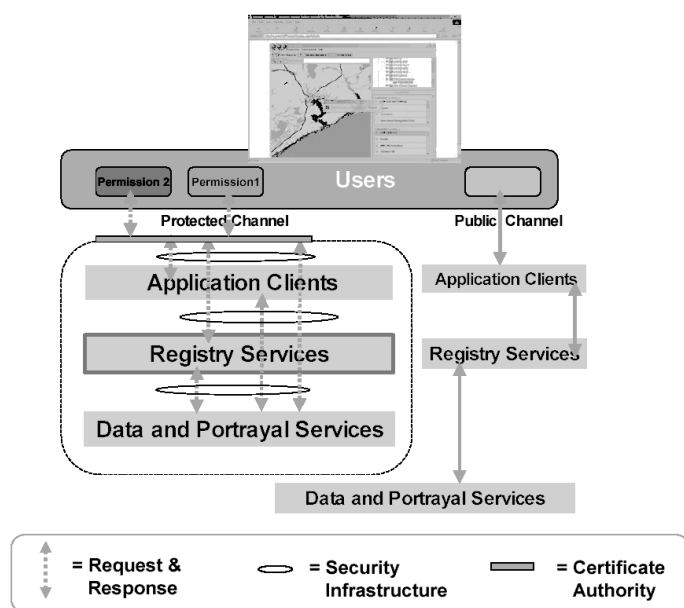
Rys. 86. Powiązanie łańcuchowe usługi powiadamiania (WNS) z innymi usługami OpenGIS – Diagram sekwencji języka UML. [Źródło: Archiwum OGC]

6.2. CICE – środowisko współdziałania w sytuacjach krytycznych

CICE (*Critical Infrastructure Collaborative Environment*) to nowy program OGC ukierunkowany na zintegrowane wykorzystanie usług geoinformacyjnych dla potrzeb bezpieczeństwa narodowego w sytuacjach szczególnych zagrożeń. W ostatnim okresie prace w OGC



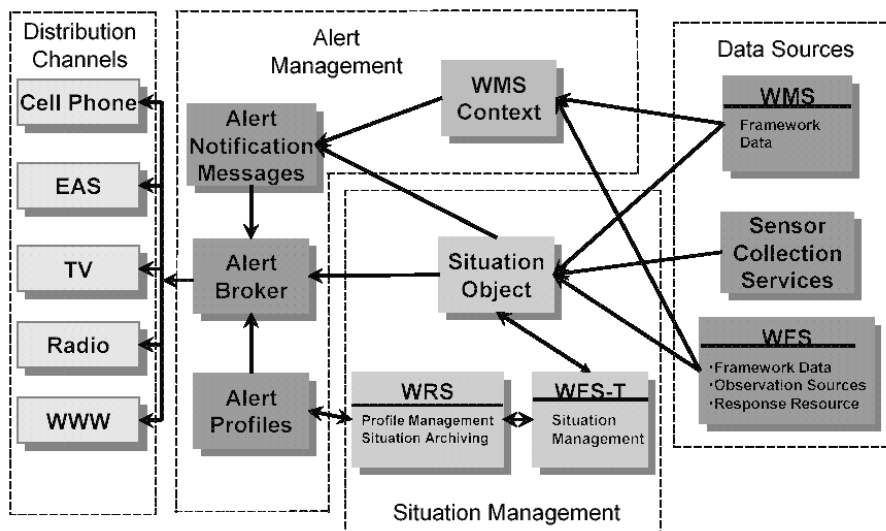
Rys. 87. Schemat struktury poziomej przepływu geoinformacji w infrastrukturze projektu CICE. [Źródło: Archiwum OGC]



Rys. 88. Podsystem ochrony dostępu do danych i usług w infrastrukturze projektu CICE. [Źródło: Archiwum OGC]

koncentrują się głównie nad rozwinięciem serwisów webowych o nowe usługi związane z sytuacjami zagrożeń publicznych. Na podstawie doświadczeń zebranych w projektach GFST (*Geospatial Fusion Services Testbed*) i GFSPP (*Geospatial Fusion Services Pilot Project*) powstało szereg roboczych specyfikacji implementacyjnych dla różnych typów internetowych usług geoinformacyjnych.

Projekt przewiduje, że infrastruktura CICE będzie miała zasięg międzynarodowy (rys. 87), ponieważ w obecnych czasach główne zagrożenia społeczne również mają charakter międzynarodowy. Ze zrozumiałych powodów istotnym problemem jest w tym przypadku wyjątkowo staranne opracowanie podsystemu ochrony dostępu do danych i usług (rys. 88). Inna, również ważna część infrastruktury CICE to podsystem powiadamiania, czyli powiązanie elementów dotyczących usług geoinformacyjnych z publicznymi kanałami i środkami przekazu informacji (rys. 89).



Rys. 89. Podsystem powiadamiania o zagrożeniach projektowany w ramach programu CICE. [Źródło: Archiwum OGC]

6.2.1. Lista projektów specyfikacji usług w ramach CICE

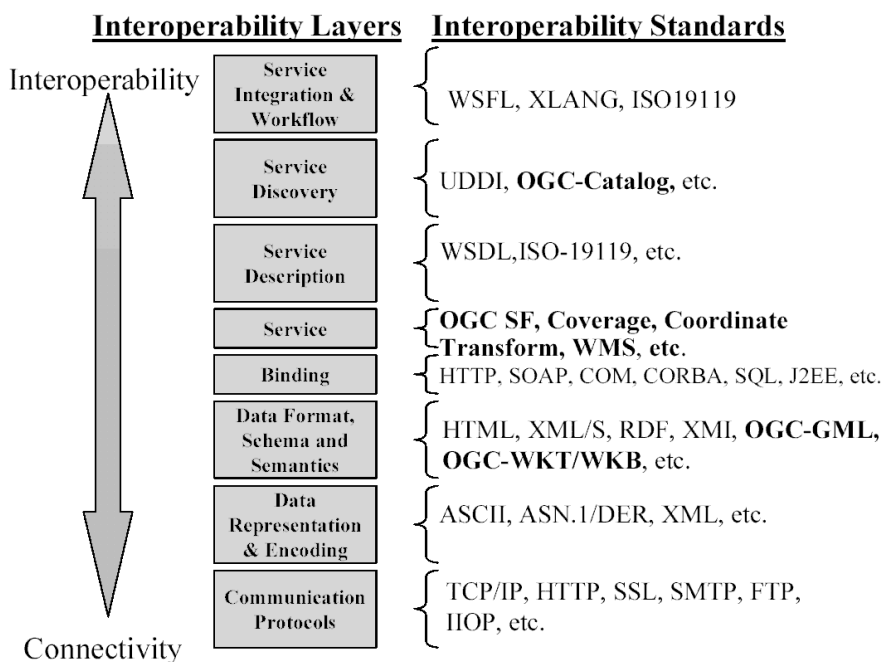
Aktualne prace zespołów roboczych programu CICE koncentrują się nad nowymi specyfikacjami implementacyjnymi dotyczącymi poszczególnych usług i powiązań między tymi usługami. Najważniejsze z nich to następujące usługi związane z siecią WWW::

- WNS (*Web Notification Service*) – usługa zawiadamiania,
- WCS (*Web Coverage Server/Service*) – serwer/usługa pokryć,
- WFS (*Web Feature Server/Service*) – serwer/usługa wyróżnień,
- WRS (*Web Registry Server*) – serwer rejestrów (usług i danych),
- WGCS (*Web GeoCoder Service*) – usługa geokodowania,
- WGPS (*Web GeoParser Service*) – usługa wyszukiwania w tekstach pośrednich odniesień geograficznych (np. nazwy geograficzne, adresy i kody pocztowe),
- LOF (*Location Organizer Folder*) – folder organizujący dane związane z położeniem,
- WTS (*Web Terrain Server*) – serwer obrazów terenu,

- WCTS (*Web Coordinate Transformation Service*) – usługa przeliczania współrzędnych,
- WGTS (*Web GazeTteer Service*) – usługa zamiany nazw geograficznych na współrzędne,
- WPOS (*Web Pricing & Ordering Service*) – usługa opłat i zamówień,
- SMS (*Style Management Service*) – usługa zarządzania stylami (zobrazowania geoinformacji).

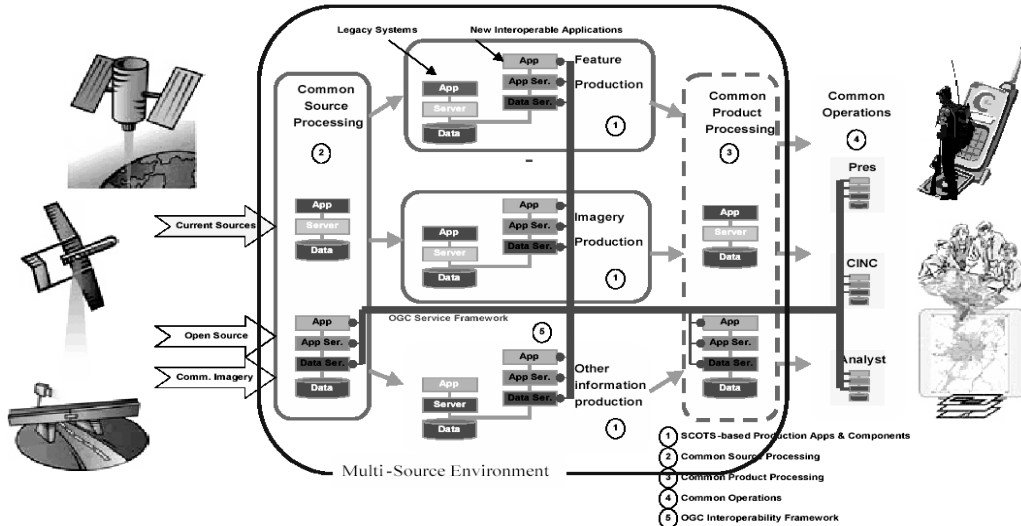
6.2.2. Problemy technologiczne integracji usług geoinformacyjnych

Nowy, inny sposób patrzenia na interoperacyjność oparty na usługach wymaga modyfikacji dotychczasowych ustaleń w zakresie zasad modelowania pojęciowego. Obok już wcześniej stosowanych diagramów klas i pakietów UML w nowych specyfikacjach pojawiają się inne diagramy UML przeznaczone do zapisu dynamicznych aspektów systemów geoinformacyjnych, jak na przykład diagramy sekwencji (rys 86 i 96) i kolaboracji (współdziałania). Nowe podejście jest wyrażone między innymi ośmiowarstwowym modelem interoperacyjności przedstawionym na rysunku 90.



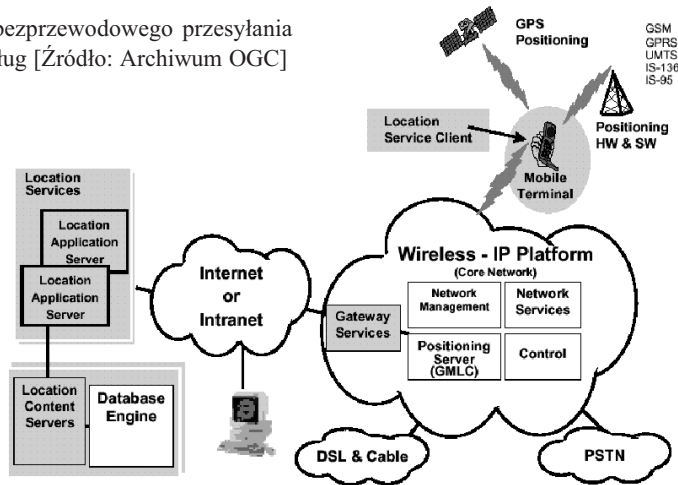
Rys. 90. Model interoperacyjności w zakresie geoinformacji obejmujący 8 warstw określony w nowych specyfikacjach OGC [Źródło: Archiwum OGC]

Dynamiczny charakter procesów informatycznych, jakie będą realizowane w tej infrastrukturze wynika głównie z włączenia nowych elementów, które będą zbierały dane na bieżąco i łączyły je z danymi zgromadzonymi w bazach. Te nowe elementy to różnego typu sensory, na przykład automatycznie działające czujniki lub kamery umieszczone na ziemi, w powietrzu i w



Rys. 91. Schemat struktury przepływu geoinformacji (archiwalnej i bieżącej) w infrastrukturze CICE [Źródło: Archiwum OGC]

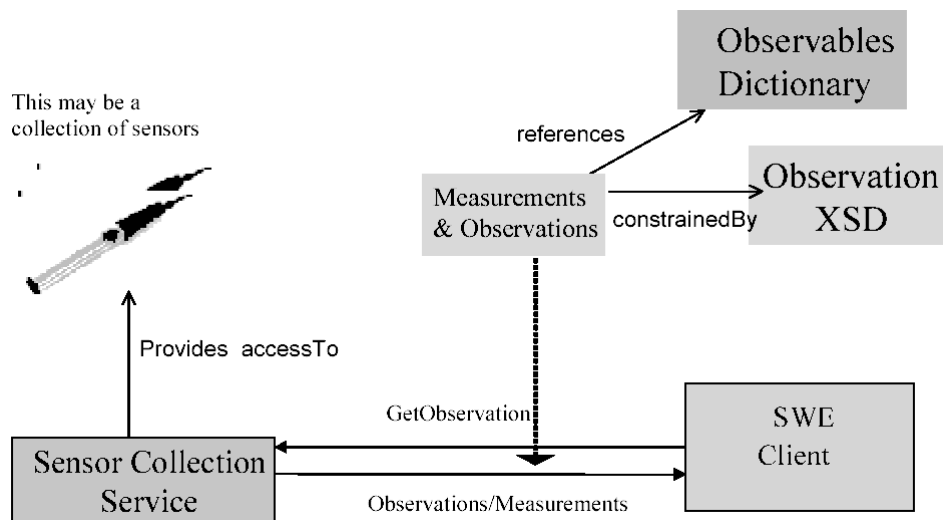
Rys. 92. Podsystem bezprzewodowego przesyłania danych i realizacji usług [Źródło: Archiwum OGC]



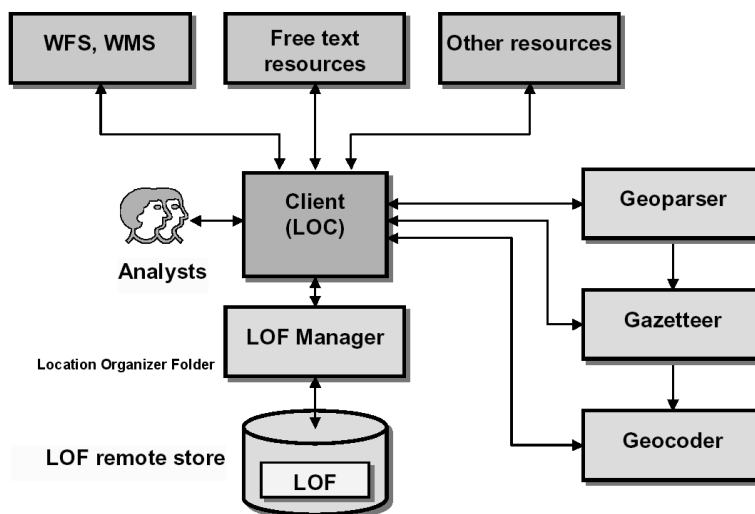
przestrzeni kosmicznej. Dynamiczny przepływ informacji z różnych źródeł jest przedstawiony na rysunku 91. Inny istotny problem to dostosowanie specyfikacji dotyczących geoinformacji do różnego typu systemów komunikacji bezprzewodowej (rys 92).

W warunkach, gdy znaczna ilość informacji będzie pochodziła ze zdalnych automatycznie sterowanych sensorów, powstaje potrzeba opracowania specyfikacji obsługi tych sensorów, a także zbierania i przetwarzania danych, które będą przez te sensory wysyłane. Ogólne założenia tego podsystemu przedstawia rysunek 93.

Wielka różnorodność i złożoność geoinformacji, a także jej ilość wymaga pomocniczych usług w zakresie selekcjonowania i dostarczania jej użytkownikom. Jednym z rozwiązań tego problemu jest podsystem LOF (*Location Organizer Folder*) – system utrzymujący indywidual-



Rys. 93. Schemat struktury usług w zakresie pomiarów i obserwacji dokonywanych przy pomocy zdalnych sensorów. [Źródło: Archiwum OGC]

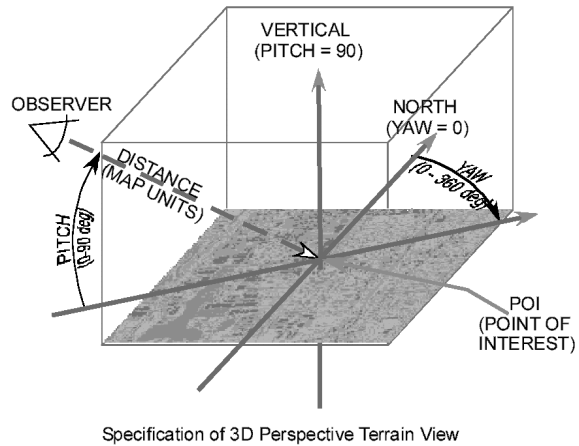


Rys. 94. Schemat struktury powiązania LOF (folder organizujący dane związane z ołożeniem) z innymi usługami OpenGIS [Źródło: Archiwum OGC]

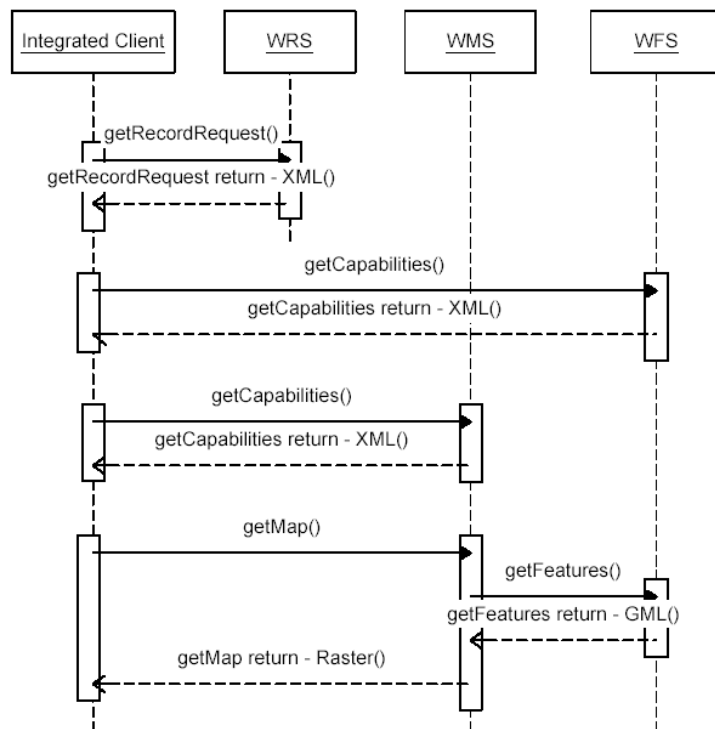
alne foldery organizujące dane związane z położeniem pod kątem widzenia potrzeb określonego użytkownika. Schemat funkcjonowania systemu LOF zawiera rysunek 94.

Inna nowa usługa dotycząca bezpośrednio użytkownika infrastruktury CICE to generowanie widoków powierzchni ziemi na podstawie geoinformacji zebranej dla wyznaczonego fragmentu tej powierzchni. Przesyłany komunikat zawierający zamówienie tej usługi zawiera między innymi współrzędne katowe i odległość określające punkt obserwacji (rys. 95).

Rys. 95. Określanie parametrów widoku ziemi w ramach usługi WTS – serwer WWW widoków powierzchni ziemi [Źródło: Archiwum OGC]



Złożoność operacji wykonywanych po stronie klienta związanych z wieloma różnymi i powiązаныmi ze sobą usługami geoinformacyjnymi wymaga, aby oprogramowanie to było w stanie komunikować się z różnymi podsystemami infrastruktury i funkcje te są określone w definicji tak zwanego „zintegrowanego klienta” (rys. 96).



Rys. 96. Diagram sekwencji UML przedstawiający „łańcuchowanie” usług OpenGIS na przykładzie webowych usług rejestru, map i wyróżnień realizowanych dla tak zwanego „zintegrowanego klienta”. [Źródło: Archiwum OGC]


6.2.3. Przykłady rozwiązań – GNS serwer nazw geograficznych

Wiele z przedstawionych powyżej koncepcji nowych usług jest obecnie już w fazie eksperymentów. Do tej grupy należy także usługa w zakresie WGTS (*Web GazeTteer Service*) polegająca na przyporządkowaniu współrzędnych nazwom geograficznym. Rysunek 97 przedstawia fragment okna przeglądarki komunikującej się z eksperymentalnym serwerem słownikowym podającym lokalizację wyróżnienia o nazwie Warszawa.

Gazetteer Search Results		
Name	Country	Designation
Warszawa	POLAND	seat of a first-order administrative division
Warszawa	POLAND	capital of a political entity

[New search](#)

The map




zoom in
 zoom out
 pan
 [Back to List](#)
[New search](#)

Rys. 97. Prototypowy serwer słownikowy nazw geograficznych (WGTS – Web GazeTteer Server) firmy IonicSoftware. Okno przeglądarki pokazuje wynik wyszukiwania nazwy Warszawa.

NIMA: GNS Public Page - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Address <http://www.nima.mil/gns/html/> Go Links

www.nima.mil **GONet Names Server (GNS)** 

Click Button to Query GNS

(Database last updated on 14 August 2003)

Through the month of August 2003, GNS can be accessed by either of the following links:

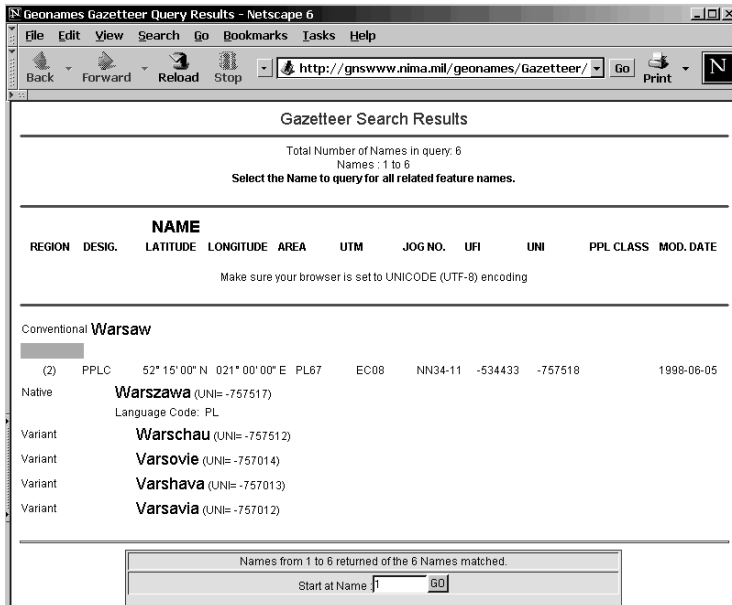
<http://gnswww.nima.mil/geonames/GNS/index.jsp> or
<http://gnpswww.nima.mil/geonames/GNS/index.jsp>

On 1 September 2003, the only valid link to access GNS will be
<http://gnswww.nima.mil/geonames/GNS/index.jsp>

If you have us bookmarked in your browser, please remember to implement this change.

Download Names Files for Countries and Territories (HTTP) Download Names Files for Countries and Territories (FTP)

Rys. 98. Operacyjny serwer nazw geograficznych GNS prowadzony przez NIMA.
 [Źródło: <http://www.nima.mil/gns/html/>]



Rys. 99. Wynik wyszukiwania w serwerze GNS nazwy Warszawa. [Źródło: <http://www.nima.mil/gns/html/>]

Native Warszawa Wschodnia										
(2)	RSTN	52° 15' 00" N	021° 03' 00" E	PL67	EC08	NN34-11				
Native Warszawa Główna										
(2)	RSTN	52° 14' 00" N	021° 00' 00" E	PL67	EC08	NN34-11				
Native Warszawa Gdańska										
(2)	RSTN	52° 16' 00" N	021° 00' 00" E	PL67	EC09	NN34-11				
Native Warszawa Dworzec Zachodni										
(2)	RSTN	52° 14' 00" N	020° 59' 00" E	PL67	DC98	NN34-11				
Native Warszawa Language Code: PL										
(2)	PPLC	52° 15' 00" N	021° 00' 00" E	PL67	EC08	NN34-11				
Native Warszawa										
(1)	PPLA	52° 15' 00" N	021° 00' 00" E	PL78	EC08	NN34-11				

Rys. 100. Nazwy geograficzne w serwerze GNS są zapisane w Unicode (pozwala to na używanie liter narodowych z całego świata), są wielocłonowe i dotyczą różnych typów wyróżnień – także dworców kolejowych.

[Źródło: <http://www.nima.mil/gns/html/>]

Na rysunkach 98, 99 i 100 są pokazane kolejne etapy przekazywania danych lokalizacyjnych (w tym przypadku także dla wyróżnień o nazwie Warszawa) przez operacyjnie funkcjonujący serwer GNS (GEOnet Names Server) prowadzony przez amerykańską agencję NIMA (National Imagery and Mapping Agency). Serwer ten nie przedstawia wyniku w formie graficznej, jednak ilość informacji dotyczących podanej nazwy jest znacznie większa niż w przypadku innych serwerów – oprócz współrzędnych podawane są wersje językowe tej nazwy (rys. 99), a także wielowyrzowe nazwy wyróżnień, w których kluczowe słowo (Warszawa) występuje (rys. 100).

6.3. Systemy programowe OpenSource dla geoinformacji

Przedstawiony w rozdziale 2.1.2 raport techniczny inicjatywy INSPIRE dotyczący architektury i standardów infrastruktury ESDI podkreśla potrzebę stosowania w systemach geoinformacyjnych oprogramowania mającego charakter otwarty, to znaczy takiego, którego kod źródłowy jest powszechnie dostępny i może być modyfikowany i rozwijany w zależności od konkretnych potrzeb aplikacyjnych. Trzeba tu zaznaczyć, że pewne podsystemy infrastruktury istotne dla spraw bezpieczeństwa lub aspektów finansowych są z tego wymogu wyłączone.

Oprogramowanie takie jest określane nazwą „OpenSource”, co jest tłumaczone na język polski jako „Wolne Oprogramowanie”. Wyjaśnienie tego, czym jest Wolne Oprogramowanie zawiera zamieszczony poniżej cytat z witryny RWO (Ruchu na rzecz Wolnego Oprogramowania) o adresie: <http://www.rwo.pl/owo>:

Czym jest Wolne Oprogramowanie?

Wolne Oprogramowanie jest udostępniane na warunkach pozwalających użytkownikowi na jego swobodne studiowanie, rozpowszechnianie i modyfikację. Licencje na korzystanie z Wolnego Oprogramowania nie zawierają restrykcji, których celem byłaby maksymalizacja zysku właściciela praw autorskich przy jednoczesnym skrajnym ograniczeniu praw użytkownika.

Przeciwieństwem Wolnego Oprogramowania są programy określane jako „własnościowe” (...). Większości własnościowych aplikacji nie wolno pod żadnym pozorem modyfikować. Zazwyczaj jest to zresztą praktycznie niemożliwe, gdyż producent programu nie dostarcza jego kodu źródłowego (a jedynie jego postaci wynikowej – wykonywalnej, zrozumiałej dla maszyny).

(...)Twórcy Wolnego Oprogramowania udostępniają je bez ograniczeń, gdyż cenią sobie te wartości, które są fundamentem rozwoju nauki: współpracę, wzajemną krytykę i wymianę idei. Sądzą, że satysfakcja z dzieła cenionego i użytecznego dla innych jest warta więcej niż można kupić za pieniądze. Dzieło doskonalone przez użytkowników ma szansę rozwijać się szybciej niż jego komercyjne odpowiedniki i przetrwać pierwotnego twórcę. Chęć zysku nie jest dla wspomnianych ludzi wystarczająco dobrym powodem, by ograniczać przepływ informacji. Z tego powodu dzielą się swoim dorobkiem z innymi. Są również przeciwnikami patentowania oprogramowania i algorytmów, uważając je za wspólny dorobek ludzkości.”

Wiele systemów programowych dla geoinformacji ma status OpenSource. Można tu podać dwa przykłady: OpenMap firmy BBN i Deegree opracowywane w Uniwersytecie w Bonn we współpracy z firmą LatLon.

6.3.1. OpenMap firmy BBN

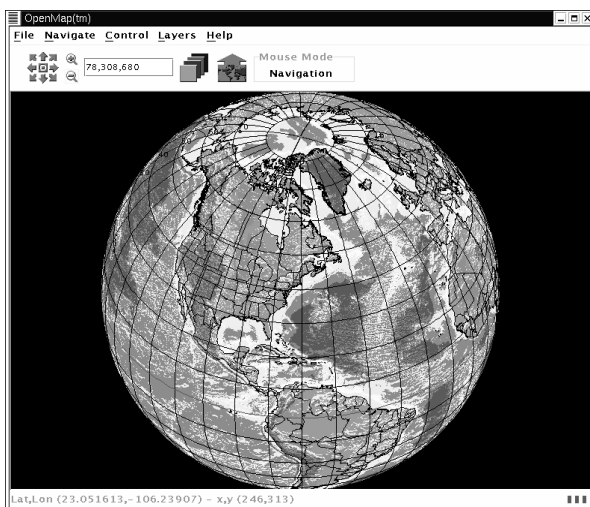
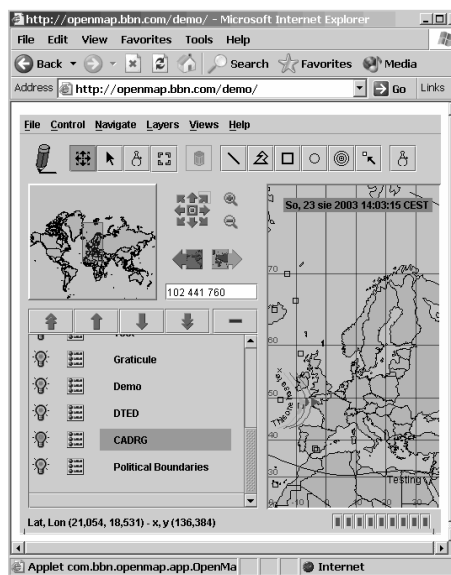


Ciemna plama w logo OpenMap to Antarktyda.

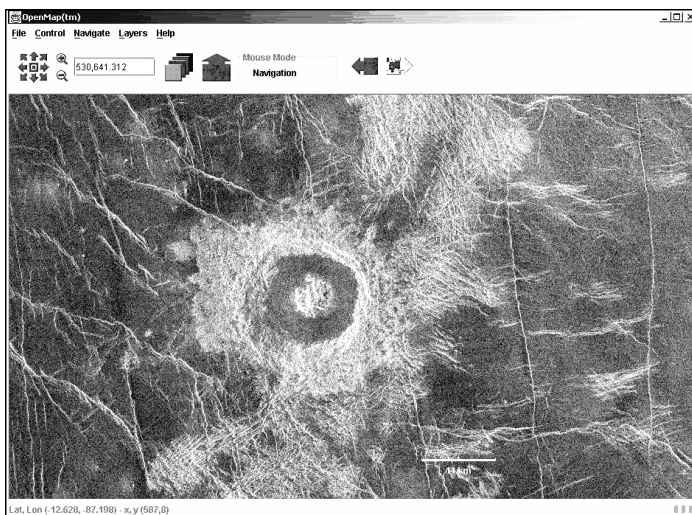
OpenMap to oprogramowanie aplikacyjne w języku Java z licencją Open-Source przeznaczone do budowy skomplikowanych przeglądarek geoinformacji w różnych formach i standardach.

Biblioteka klas Open Map była opracowana w amerykańskiej firmie BBN z przeznaczeniem do budowy aplikacji dla celów wojskowych. Wiele z tych aplikacji jest nadal w użyciu. Uniwersalność zastosowanych rozwiązań (rys. 101 i 102) i otwarty charakter tego oprogramowania (OpenSource) sprawiły, że jest ono obecnie podstawą wielu aplikacji opracowanych dla różnych zastosowań w różnych krajach. Przykłady tych aplikacji zawierają rysunki 103 i 104.

Rys. 101. Okno przeglądarki zbudowanej na bazie pakietu klas OpenMap w języku Java.
[Źródło: <http://www.openmap.org>]



Rys. 102. Przeglądarka OpenMap może po przeliczeniu pokazywać dane w różnych układach odwzorowania. W tym przypadku mapa Ziemi jest zobrazowana w odwzorowaniu azymutalnym.
[Źródło: <http://www.openmap.org>]



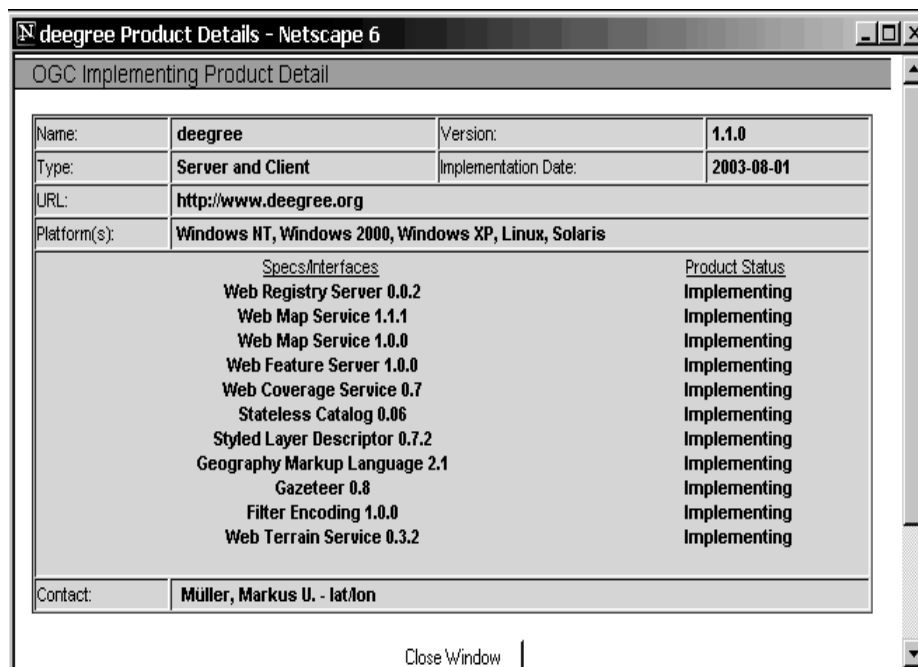
Rys. 103. Zastosowania pakietu OpenMap w geologii: zdjęcie satelitarne wykorzystane do analizy budowy skorupy ziemskiej.
[Źródło: <http://www.openmap.org>]

Rys. 104. Zastosowania pakietu OpenMap: Przeglądarka Nest opracowana w ramach projektu MARE (Uniwersytet Sztokholmski), którego celem jest analiza zjawisk związanych z ekologią basenu Morza Bałtyckiego.
[Źródło: <http://www.openmap.org>]



6.3.2. Deegree – Uniwersytet w Bonn

Na Wydziale Geografii Uniwersytetu w Bonn we współpracy z firmą LatLon prowadzone są prace badawcze i projektowe nad biblioteką klas w języku Java i nad szeregiem aplikacji opartych na tej bibliotece zgodnych ze specyfikacjami OpenGIS. W wyniku tych prac powstało oprogramowanie o nazwie Deegree objęte licencją OpenSource.



Rys. 105. Certyfikat zgodności oprogramowania Deegree ze specyfikacjami implementacyjnymi OpenGIS. [Źródło: <http://www.opengis.org>]

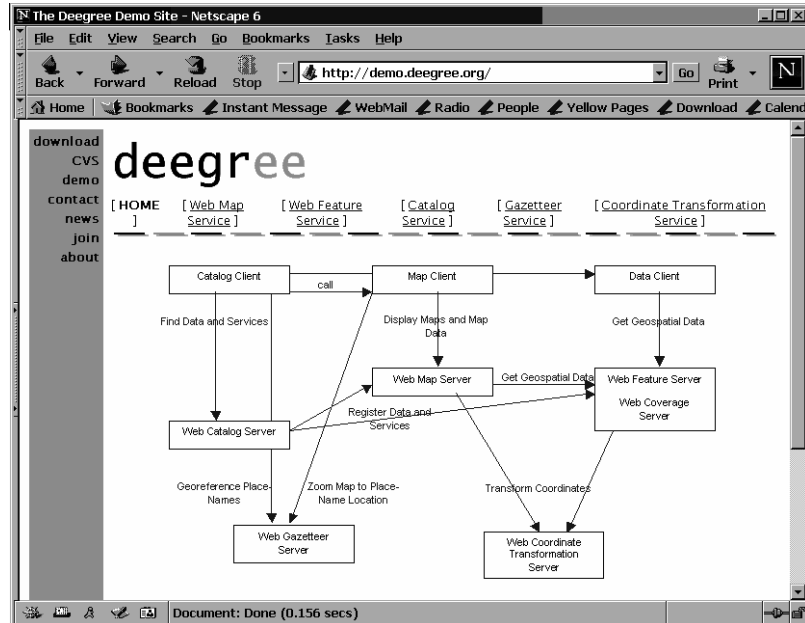
Na podkreślenie zasługuje fakt, że pod względem zgodności ze specyfikacjami OpenGIS oprogramowanie to znajduje się w ścisłej czołówce i tylko nieliczne produkty komercyjne mogą mu pod tym względem dorównać. Rysunek 105 zawiera elektroniczną wersję certyfikatu zgodności wystawionego przez OGC. Komponenty pakietu Deegree spełniające wymagania specyfikacji implementacyjnych:

- serwer map zgodny ze specyfikacją WMS 1.1.1,
- serwer wyróżnień (*features*) zgodny z WFS 1.0.0,
- serwer pokryć (*coverages*) zgodny z WCS 0.7,
- serwer katalogowy zgodny z WRS 0.0.2,
- serwer słownika geograficznego (*gazetteer*) zgodny z Gazetteer 0.8,
- serwer przeliczania współrzędnych (zgodnie z WMS 1,0.0).

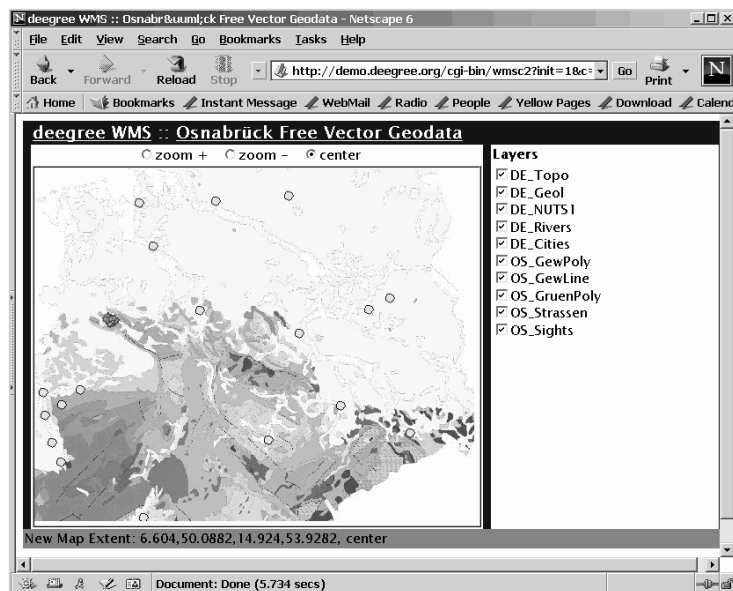
Deegree spełnia także inne standardy OGC:

- Stateless Catalog 0.06,
- Styled Layer Descriptor 0.7.2,
- Geography Markup Language 2.1,
- Filter Encoding 1.0.0,
- Web Terrain Service 0.3.2

Złożoność usług geoinformacyjnych specyfikowanych w nowych projektach OGC pociąga za sobą złożoność systemów, które mają te usługi realizować. W większości Degreee spełnia te wymagania poprzez powiązania pomiędzy poszczególnymi jego częściami, co jest widoczne na rysunku 106. Przykład zastosowania Deegree przedstawia rysunek 107.



Rys. 106. Schemat przedstawiający wzajemne powiązania modułów oprogramowania Deegree. [Źródło: <http://www.deegree.org>]

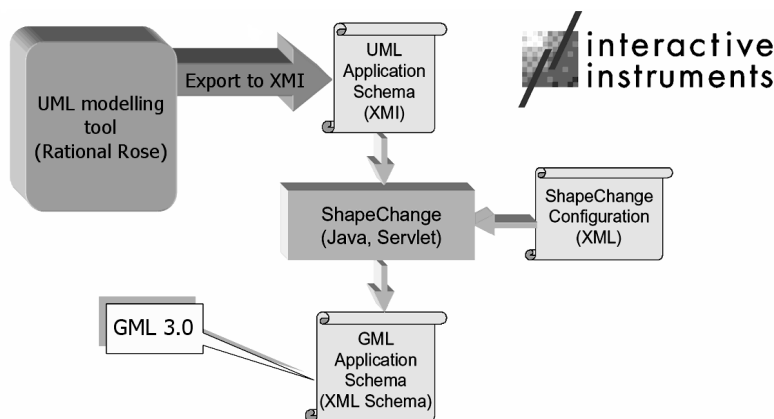


Rys. 107. Jedno z zastosowań pakietu Deegree: mapa geologiczna obszaru położonego na południu Niemiec. [Źródło: <http://www.deegree.org>]

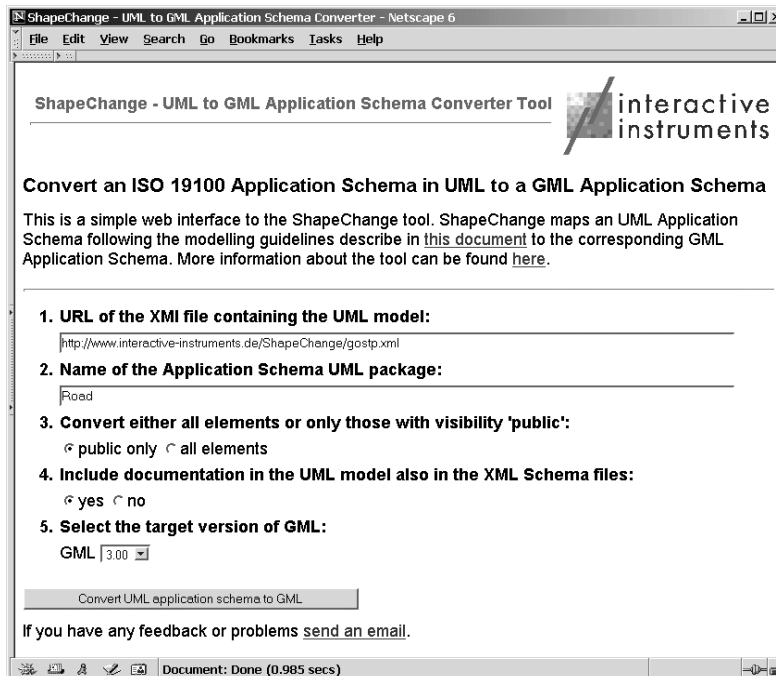
6.4. Harmonizacja i konwersja do XML modeli standardu ISO 19100

W ośrodkach badawczych związanych z OGC są obecnie prowadzone prace nad metodami przeniesienia tego, co określają normy grupy ISO 19100, na poziom zastosowań praktycznych. W wielu przypadkach jest to proste zadanie, ponieważ szereg standardów ISO jest adaptacją specyfikacji implementacyjnych OpenGIS, które najczęściej były opracowywane przy jednoczesnym weryfikowaniu ich z zastosowaniem systemów eksperymentalnych. W pozostałych przypadkach jest to proces bardziej złożony i głównie sprowadza się do harmonizacji modeli pojęciowych i do konwersji tych modeli na języki implementacyjne – ostatnio głównie do XML.

Etapem pośrednim tego przejścia, jak to jest przedstawione w rozdziale 3.3, jest język XMI. Z tego względu jest tu często stosowany program narzędziowy HyperModel firmy Ontogenics Corp. (rozd. 3.4.2), który jest realizacją koncepcji pozwalającej przy pomocy języka XMI na dokonywanie konwersji modeli pojęciowych pomiędzy różnymi środowiskami. Język XMI stanowi ogniwo pośrednie pomiędzy modelami i schematami danych kilku najważniejszych platform implementacyjnych. Równolegle do zastosowań programu HyperModel rozwijane są trzy inne metody przejścia z UML za pośrednictwem XMI do XML i w tym do GML. Pierwsza jest stosowana w NIMA do modelu metadanych (ISO 19115). Drugą rozwija międzynarodowy zespół badawczy „Grupa Nordycka” realizujący *Joint Nordic Implementation Project*. Trzecia metoda została opracowana w niemieckiej firmie *Interactive Instruments*. Dwie pierwsze metody przedstawiają rozdziały 6.4.1 i 6.4.2. Tu jest opisana metoda trzecia, której realizacją jest program ShapeChange pozwalający na konwersje modeli aplikacyjnych opartych na ISO 19100 i zapisanych w języku XMI do języka GML 3.0. Schemat przepływu danych podczas tej konwersji zawiera rysunek 108, a interfejs tego programu przedstawia rysunek 109.



Rys. 108. Schemat przedstawiający konwersję modeli ISO 19100 do języka GML 3.0 za pomocą programu ShapeChange. [Źródło: dokumentacja programu]



Rys. 109. Interfejs programu narzędziowego ShapaChange [Źródło: dokumentacja programu]

6.4.1. Projekt NIMA dotyczący standardu ISO 19115 – Metadane

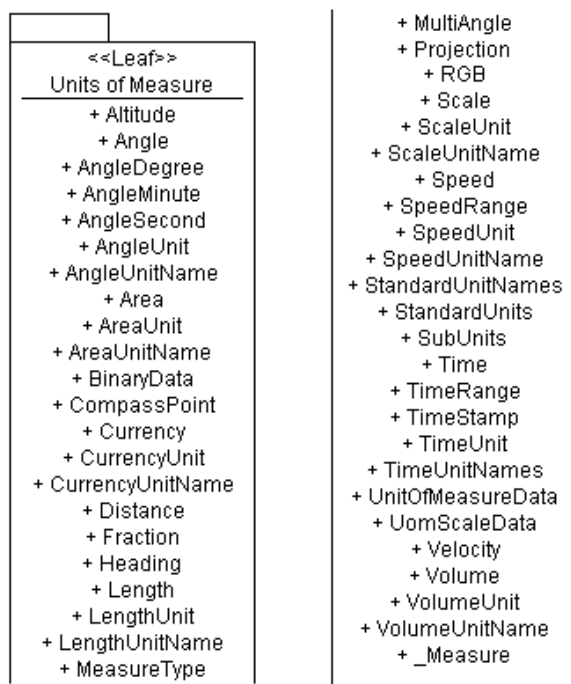
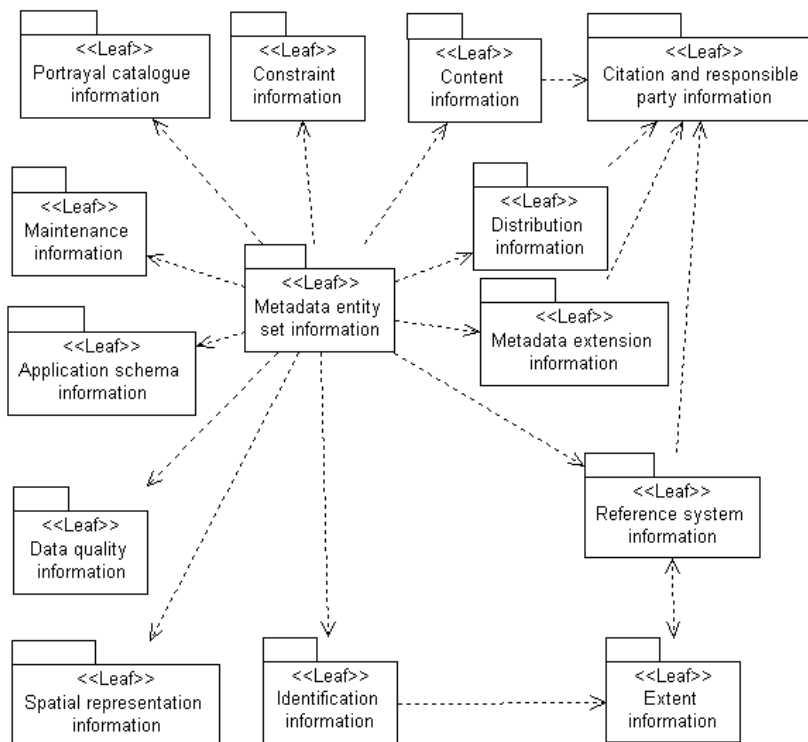
Zespół pracujący nad modelem metadanych w agencji NIMA był pierwszym, który zajął się problematyki konwersji modeli pojęciowych UML do języka XML. Prowadzone tam prace są oparte na rozwiązaniach technologicznych zawartych w programie Rational Rose i jego rozszerzeniach, a głównie na języku skryptowym tego programu RRSL (*Rational Rose Scripting Language*) i pomocniczych schematów XML definiujących reguły mapowania.

Konwersja modelu jest podzielona na 5 faz:

- odtworzenie przy pomocy programu Rational Rose modelu ISO 19115,
- eliminacja wszystkich niezgodności i harmonizacja modeli cząstkowych dla uzyskania wewnątrznie niesprzecznego modelu sumarycznego,
- zapisanie modelu w języku XML,
- konwersja modelu z XMI do XML przy pomocy skryptów,
- weryfikacja i poprawianie schematów XML opisujących model metadanych z jednoczesnym poprawieniem odpowiednich fragmentów skryptów dokonujących tej konwersji.

Rysunek 110 przedstawia pakiety modelu metadanych podlegające konwersji. Jednak ze względu na powiązania (odwołania do elementów innych modeli) dla poprawności tej operacji trzeba uwzględnić szereg innych pakietów, na przykład dotyczących jednostek miar stosowanych w metadanych (rys. 111).

Rys. 110. Diagram pakietów UML przedstawiający wzajemne powiązania pomiędzy podstawowymi pakietami modelu metadanych [Źródło: Archiwum NIMA]

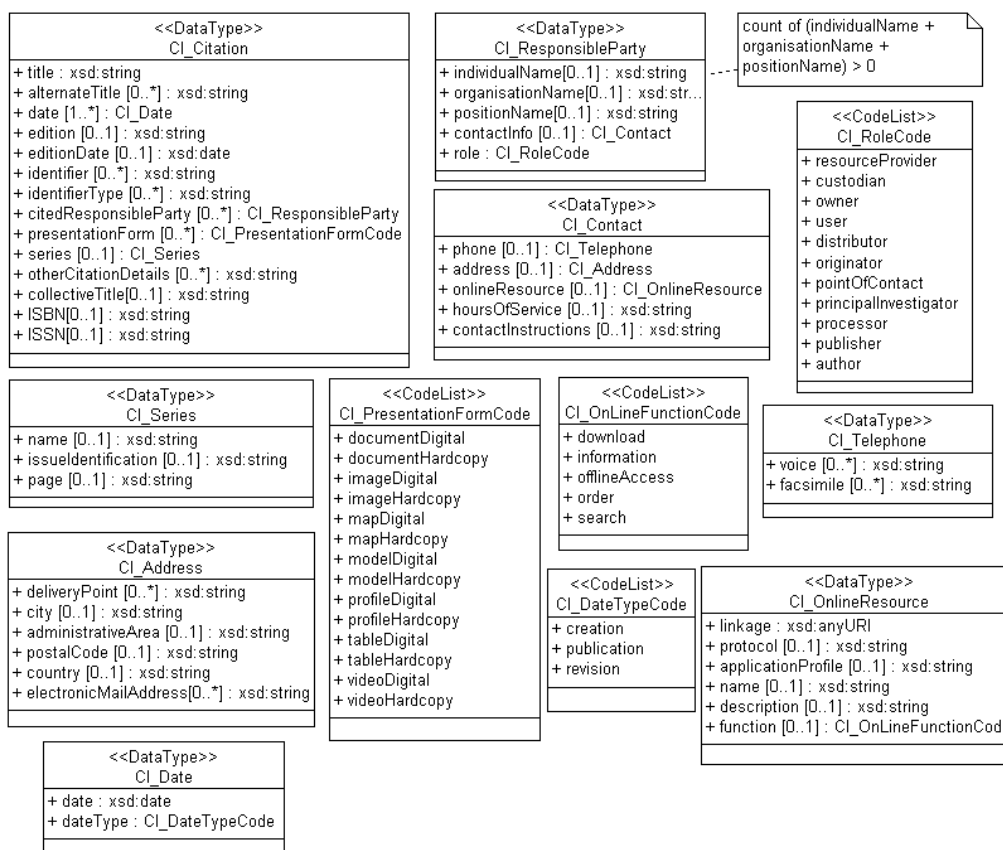


Rys. 111. Lista wszystkich klas pakietu Units of Measure potrzebnego dla odwołań z innych pakietów modelu. [Źródło: Archiwum NIMA]

Pakiet „Units of Measure” (rys. 111) nie zawiera klas dotyczących metadanych - jednak jest potrzebny, ponieważ definiuje jednostki, które są używane w innych pakietach tego modelu. Przykład ten ilustruje rozległość zagadnień, jakie trzeba rozwiązać podczas opracowywania modelu pojęciowego dotyczącego tylko jednego wąskiego problemu – w tym przypadku metadanych. Jednostki miar to zagadnienie ogólno-fizyczne, jednak bez pakietu klas, które je jednoznacznie definiują model metadanych nie może być poprawny formalnie.

Rysunek 112 zawiera diagram klas, który jest fragmentem modelu definiującego typy danych występujących w innych diagramach modelu metadanych. Użyte tu stereotypy mają charakter ogólny (abstrakcyjny) i często w innych, również abstrakcyjnych, modelach używa się w takich przypadkach innych stereotypów traktując je jako synonimy, na przykład: <<DataType>> to <<Type>>, a <<CodeList>> to <<Enumeration>>.

Skrypty Rational Rose dokonujące konwersji modeli są wspomagane zapisami (dokumentami) w języku XSL (rozd. 5.1). Poniższy przykład zawiera początkowy fragment zapisu określającego reguły transformacji:



Rys. 112. Diagram klas UML z pakietu CI_Citation definiujący klasy podstawowych typów danych modelu ISO 19115. [Źródło: Archiwum NIMA]

Przykład 33.

[Źródło: Archiwum NIMA]

```

<?xml version="1.0" encoding="UTF-8"?> <xsl:transform version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<xsl:output method="xml" indent="yes" encoding="UTF-8"/>
<!-- NEWLINE = &#xA; -->
<xsl:template match="/">
  <xsl:text>&#xA;</xsl:text>
  <xsd:schema targetNamespace="http://www.opengis.net/NIMA"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xlink="http://www.w3.org/1999/xlink" xmlns="http://www.opengis.net/NIMA"
elementFormDefault="qualified" version="2.0">
  <xsl:call-template name="SchemaDefinition"/>
</xsd:schema>
</xsl:template>
<xsl:template name="SchemaDefinition">
  <xsl:call-template name="Separator"/>
  <xsl:element name="xsd:annotation">
    <xsl:element name="xsd:documentation">
      Generated from ISO 19115 Implementation UML model by
      'UMLSchema_to_XMLSchema.xslt' Version 4.0 TMTaylor at DateTime
      <xsl:value-of select="//date"/>
    </xsl:element>
  </xsl:element>
  <xsl:call-template name="Separator"/>
  <xsl:comment>
    ===== Imports =====
  </xsl:comment>
  <xsl:call-template name="Separator"/>
  <xsl:element name="xsd:import">
    <xsl:attribute name="namespace">http://www.w3.org/1999/xlink</xsl:attribute>
    <xsl:attribute name="schemaLocation">xlinks.xsd</xsl:attribute>
  </xsl:element>
  <xsl:call-template name="Separator"/>
  <xsl:comment>
    ===== Classes =====
  </xsl:comment>
  <xsl:for-each select="classList">
    <!--===== class START =====>
  </xsl:for-each>
<!-- ... -->
<!-- koniec przykładowego fragmentu -->

```

Rysunek 113 zawiera diagram przedstawiający fragment schematu mapowania składników klas – w tym przypadku atrybutu.

W wyniku konwersji powstaje szereg schematów XML (zapisów w języku XSD). Fragment takiego zapisu jest przedstawiony poniżej:

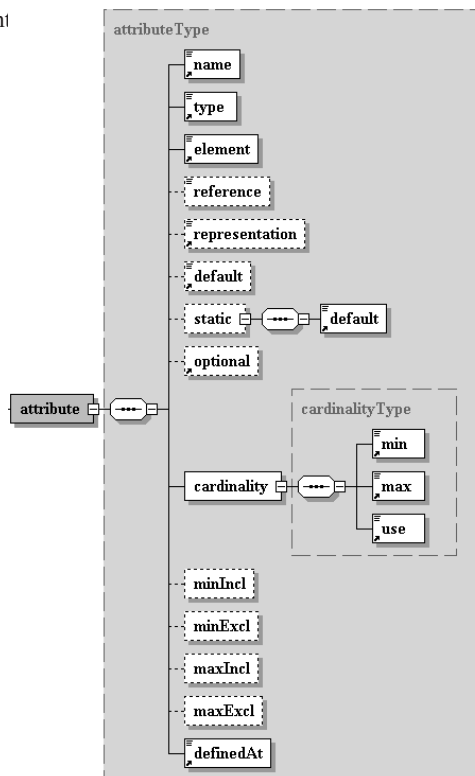
Jeden z wielu diagramów XML końcowej wersji schematów wynikowych przedstawia rysunek 114.

Przykład 33.

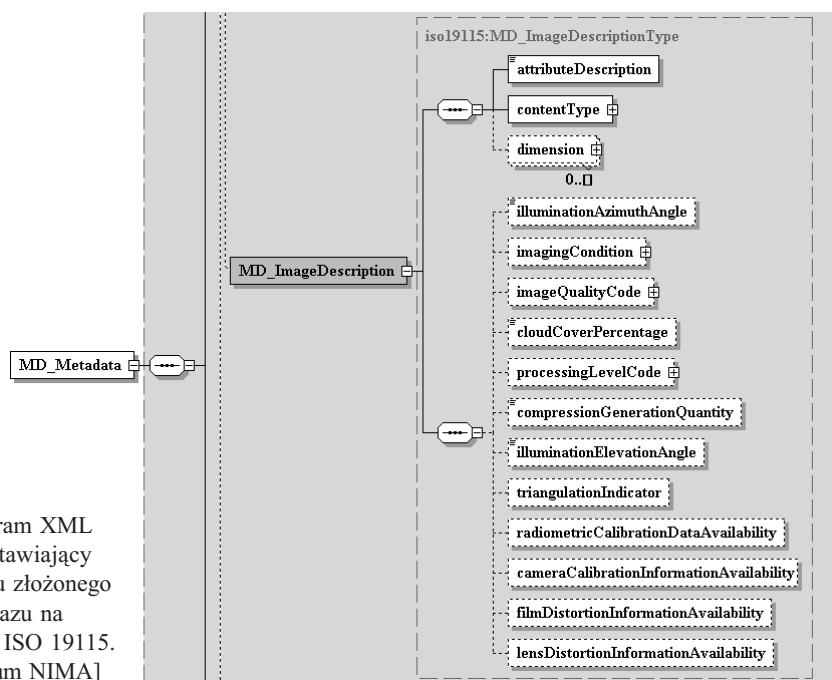
[Źródło: Archiwum NIMA]

```
<xsd:schema targetNamespace="http://www.isotc211.org/iso19115/"
  xmlns:iso639-2="http://www.isotc211.org/iso639-2/"
  xmlns:iso4217="http://www.isotc211.org/iso4217/"
  xmlns:iso19109="http://www.isotc211.org/iso19109/"
  xmlns:iso19103="http://www.isotc211.org/iso19103/"
  xmlns:gml="http://www.opengis.net/gml"
  xmlns:xlink="http://www.w3.org/1999/xlink"
  xmlns:iso19115="http://www.isotc211.org/iso19115/"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema" version="0.6">
  <xsd:annotation>
    <xsd:documentation>
      This schema was directly generated from the ISO 19115 Implementation UML
      model via Version 0.6 of the Rational Rose Script 'XMLout4XMLSchema.ebs' and
      via 'UMLSchema_to_XMLSchema.xslt' by Harry S Truman at
      DateTime 07-12-2002 12:24:57</xsd:documentation>
    </xsd:documentation>
  </xsd:annotation>
  <!-- ===== Imports ===== -->
  <xsd:import namespace="http://www.opengis.net/gml"
    schemaLocation="gml211_geometry.xsd"/>
  <xsd:import namespace="http://www.isotc211.org/iso19103/"
    schemaLocation="19103.xsd"/>
  <xsd:import namespace="http://www.isotc211.org/iso639-2/"
    schemaLocation="iso639-2.xsd"/>
  <xsd:import namespace="http://www.isotc211.org/iso4217/"
    schemaLocation="iso4217.xsd"/>
  <xsd:import namespace="http://www.isotc211.org/iso19109/"
    schemaLocation="19109.xsd"/>
  <!-- ===== Classes ===== -->
  <xsd:complexType name="MD_MetadataType">
    <xsd:sequence>
      <xsd:element ref="iso19115:_MD_Identification" maxOccurs="unbounded"/>
      <xsd:element name="metadataConstraints" type="iso19115:MD_ConstraintsType"
        minOccurs="0" maxOccurs="unbounded"/>
      <xsd:element name="dataQualityInfo" type="iso19115:DQ_DataQualityType"
        minOccurs="0" maxOccurs="unbounded"/>
      <xsd:element name="metadataMaintenance"
        type="iso19115:MD_MaintenanceInformationType" minOccurs="0"/>
      <xsd:element ref="iso19115:_MD_SpatialRepresentation" minOccurs="0"
        maxOccurs="unbounded"/>
      <xsd:element name="referenceSystemInfo"
        type="iso19115:MD_ReferenceSystemType" minOccurs="0"
        maxOccurs="unbounded"/>
    </xsd:sequence>
  </xsd:complexType>
  <!-- (...) -->
  <xsd:element name="dataSet" type="iso19103:URI"/>
  </xsd:sequence>
  <xsd:attributeGroup ref="iso19103:IM_ObjectIdentification"/>
</xsd:complexType>
<!-- (...) -->
```

Rys. 113. Diagram XML przedstawiający fragment pomocniczego schematu mapowania atrybutów
[Źródło: Archiwum NIMA]



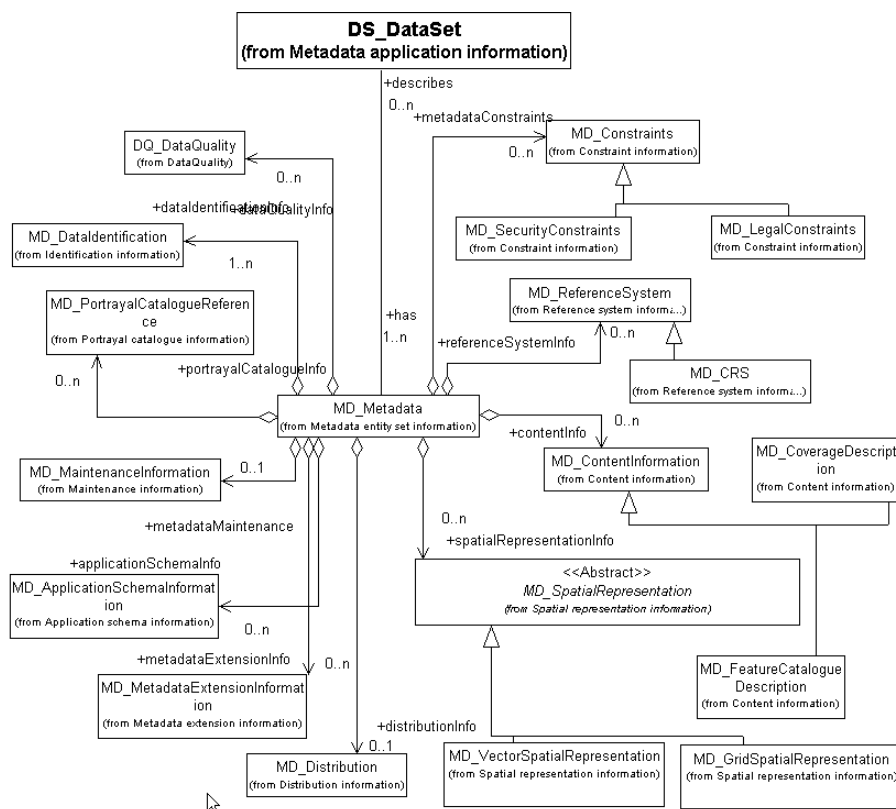
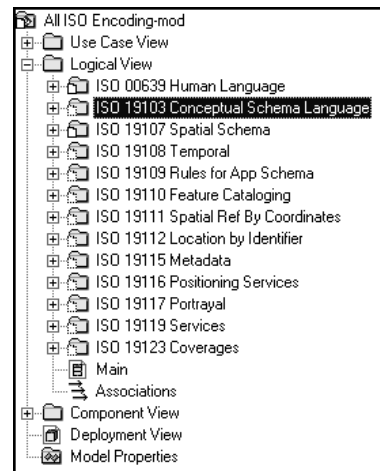
Rys. 114. Diagram XML przedstawiający składniki elementu złożonego dla opisu obrazu na podstawie modelu ISO 19115.
[Źródło: Archiwum NIMA]



6.4.2. Projekty Grupy Nordyckiej

Jednym z zadań projekt „Grupy Nordyckiej” jest pełna implementacja w XML wszystkich modeli UML zawartych w podstawowych standardach grupy ISO 19100 (rys. 115). Ponieważ nad wybranymi zagadnieniami z tego zakresu pracują także inne ośrodki, wyniki tych prac są wykorzystane w tym projekcie – przykładem jest model metadanych opracowany przez NIMA (rys. 116). Ogólna koncepcja wykorzystania tych modeli jest przedstawiona na rysunku 37 (rozd. 3.6).

Rys. 115. Okno katalogowe programu Rational Rose pokazujące listę pakietów zawierających zaimplementowane standardy ISO
[Na podstawie: raportów „Grupy Nordyckiej”]



Rys. 116. Główny diagram klas UML pakietu ISO 19115 Metadata w „Projekcie Nordyckim”
[Na podstawie: raportów „Grupy Nordyckiej”]

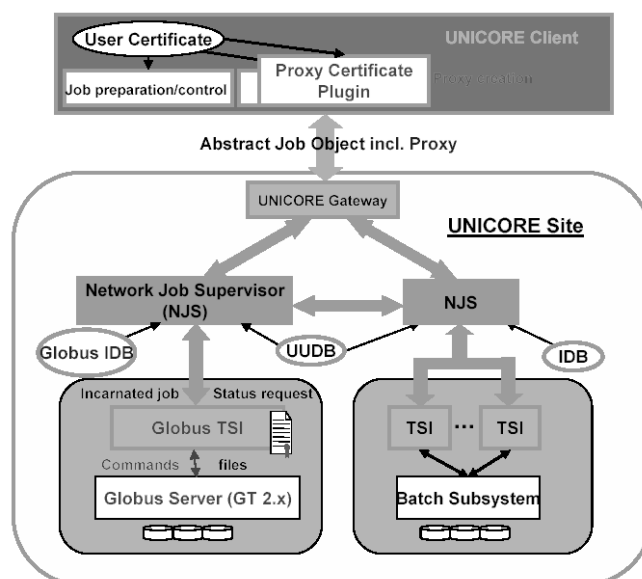
6.5. Technologie gridowe

Technologie gridowe mogą rozwiązać wiele problemów dotyczących rozproszonego przetwarzania i udostępniania danych w infrastrukturze geoinformacyjnej. Pojęcie „grid” i związana z nim technologia powstały w wyniku prac nad zwiększeniem wykorzystania superkomputerów w środowiskach rozproszonych. Grid to zorganizowana i wydzielona struktura w internecie oparta na technologii WWW i przeznaczona do przetwarzania i przesyłania informacji. Gridy znalazły zastosowanie głównie w łączeniu komputerów wielkiej mocy dla interoperacyjnego realizowania wspólnych zadań. W takich przypadkach na pierwszym miejscu stawiana jest niezawodność współpracy i ochrona przed nieuprawnionym dostępem.

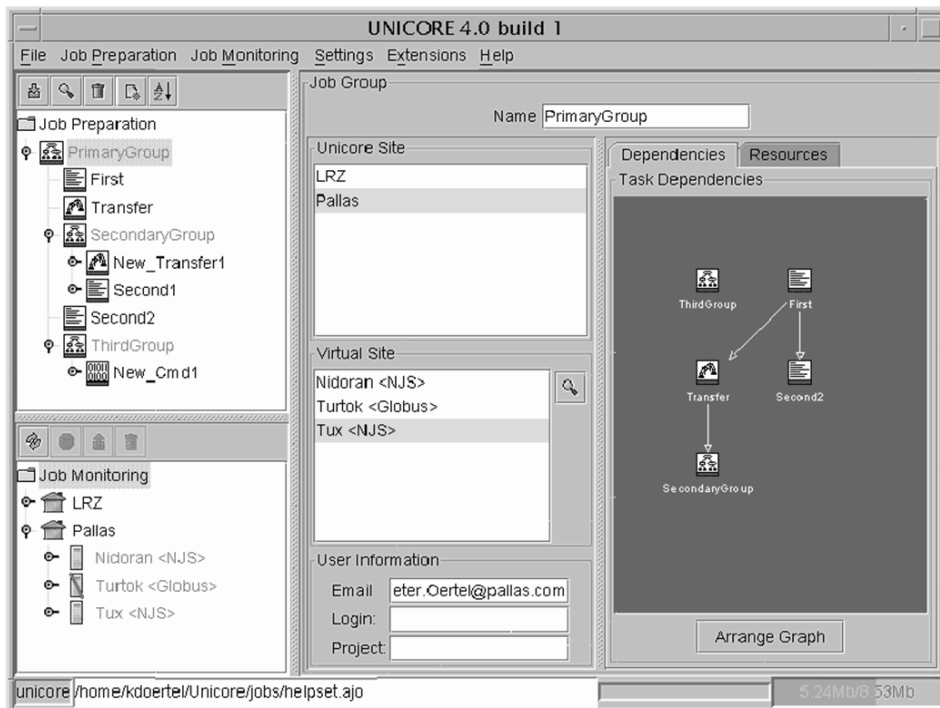
Rozwiązania technologiczne gridów i ich zastosowania ilustrują projekty:

- UNICORE – system obsługi gridu z możliwością współpracy z innym gridami,
- DataGRID – projekt ukierunkowany bardziej na przesyłanie danych, niż na współdzielenie mocy obliczeniowej i innych zasobów technicznych.

Struktura węzła gridu opartego na technologii UNICORE jest przedstawiona na rysunku 117. Użytkownik posiadający certyfikat dostępu do gridu może przy pomocy oprogramowania (rys. 118) przesłać zlecenie i następnie monitorować jego realizację. Zlecenie takie może wymagać udziału wielu komputerów udostępniających swoją moc i wielu systemów baz danych.



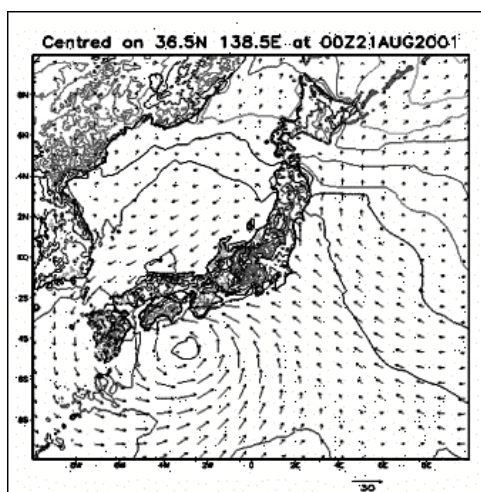
Rys. 117. Schemat architektury UNICORE. [Źródło: Raport projektu UNICORE]



Rys. 118. Okno programu przeznaczonego do przygotowywania zlecenia usługi i monitorowania jej realizacji w środowisku UNICORE. [Źródło: Raport projektu UNICORE]

6.5.1. MeteoGRID – zastosowanie UNICORE do geoinformacji

MeteoGRID jest przykładem zastosowania technologii gridowej UNICORE do prognozowania pogody. Przykład przedstawiony na rysunku 119 pozwala zauważyć, że także w tych zagadnieniach mamy do czynienia z geoinformacją.

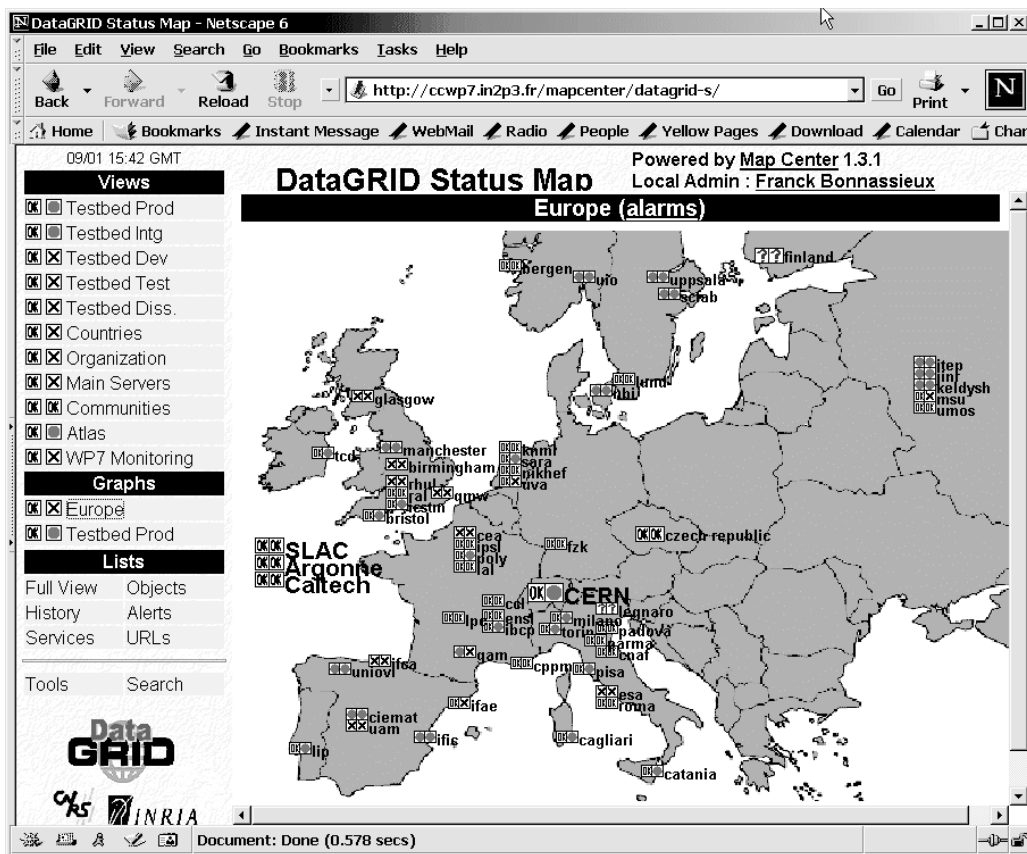


Rys. 119. Przykład aplikacji gridowej opartej na technologii UNICORE: mapa pogody opracowana w środowisku MeteoGRID. [Źródło: Raport projektu UNICORE]

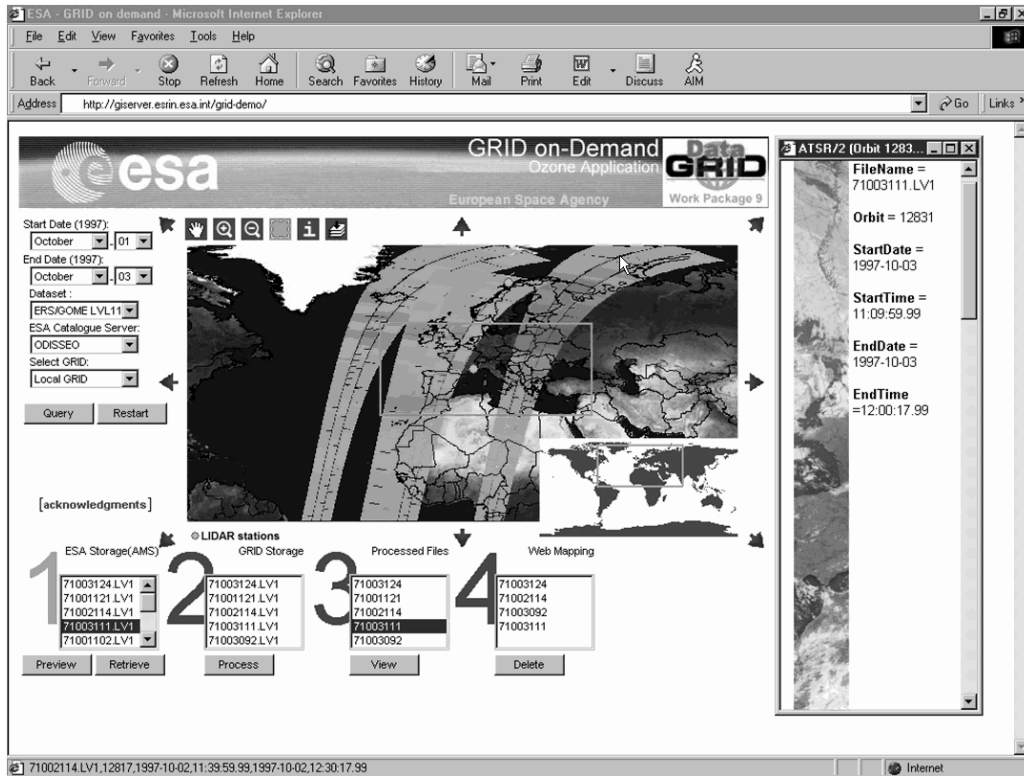
6.5.2. Przykłady zastosowania DataGRID do geoinformacji

Celem europejskiego projektu badawczego DataGRID jest zbudowanie infrastruktury komputerowej nowej generacji umożliwiającej dokonywanie obliczeń i analiz w oparciu o wielkoskalowe bazy danych – od setek terabajtów do petabajtów. Infrastruktura ta jest dedykowana rozproszonym środowiskom badawczym.

Wiele rozwiązań technologicznych zastosowanych do DataGRID może być bezpośrednio zastosowane w infrastrukturze geoinformacyjnej bez rozróżniania, jakiego poziomu ma dotyczyć. Przedstawione na rysunkach 120 i 121 przykłady potwierdzają przydatność technologii gridowych w zastosowaniach geoinformacyjnych.



Rys. 120. Monitorowanie stanu połączeń i pracy węzłów gridu projekt DataGRID
[Źródło: Raport projektu DataGRID]



Rys. 121. Geoinformacyjna aplikacja technologii DataGRID w Europejskiej Agencji Kosmicznej przeznaczona dla satelitarnych pomiarów zawartości ozonu w atmosferze
[Źródło: Raport projektu DataGRID]