

## ASPEKTY IMPLEMENTACYJNE SCHEMATÓW APLIKACYJNYCH

### IMPLEMENTATION ASPECTS OF APPLICATION SCHEMES

Agnieszka Chojka, Agnieszka Zwirowicz

Katedra Geodezji Szczegółowej, Uniwersytet Warmińsko-Mazurski w Olsztynie

**Słowa kluczowe:** schemat aplikacyjny, implementacja, SDI

Keywords: application scheme, implementation, SDI

### Wstęp

Schemat aplikacyjny jest schematem pojęciowym dla danych wykorzystywanych przez jedną lub więcej aplikacji. Definiuje zawartość i strukturę danych oraz specyfikację operacji służących do manipulowania i przetwarzania danych przez aplikacje. Schemat aplikacyjny zapisany jest za pomocą języka schematu pojęciowego (CSL, ang. *Conceptual Schema Language*), którym powinien być, według zaleceń norm ISO serii 1900, UML. Celem schematów aplikacyjnych jest zapewnienie opisu danych oraz osiągnięcie powszechnego i poprawnego rozumienia danych.

Z każdą infrastrukturą danych przestrzennych (SDI, ang. *Spatial Data Infrastructure*) związana jest perspektywa danych (ang. *data-centric view*) i perspektywa usług (ang. *service-centric view*). Obie perspektywy wiążą się z implementacją schematów aplikacyjnych.

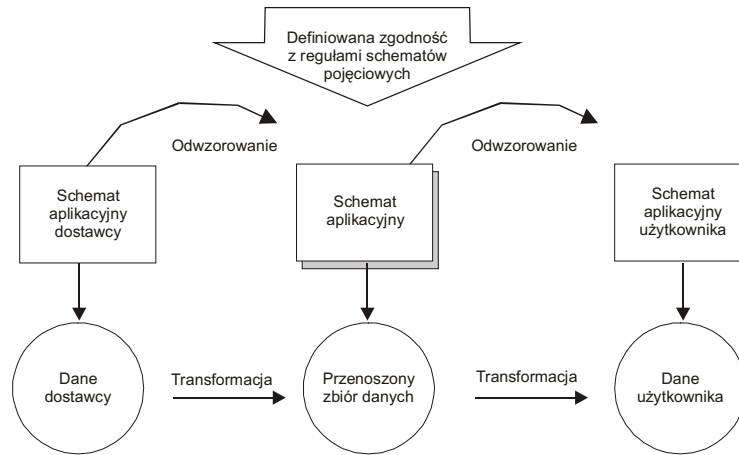
### Modele wymiany danych

W infrastrukturach danych przestrzennych realizowane są dwa sposoby wymiany danych (rys. 1 i 2) między niezależnymi systemami: transfer i transakcje.

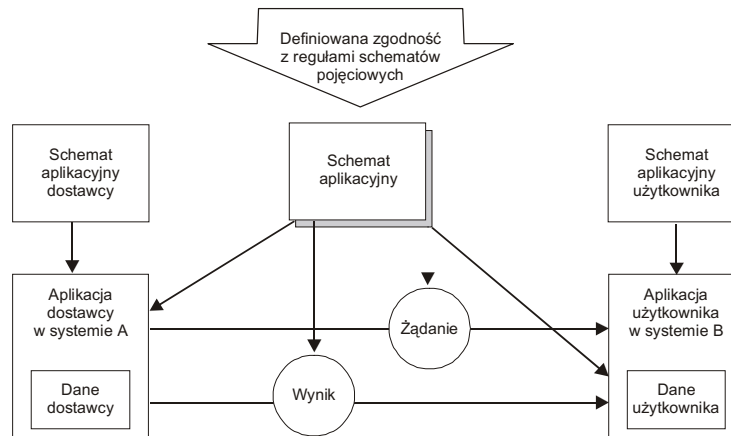
W przypadku perspektywy danych wykorzystywany jest model transferu, natomiast w przypadku perspektywy usług używany jest model transakcji.

W modelu transferu danych, dostawca danych tworzy zbiór danych, który jest przekazywany użytkownikowi. Struktura i zawartość danych są opisane w schemacie aplikacyjnym dla zbioru danych. Zbiór danych jest wysyłany w formie wymiany.

W modelu transakcji (współdziałania), aplikacja użytkownika kontaktuje się z dostawcą aplikacji poprzez powszechny protokół komunikacji. W tym przypadku użytkownik odwołu-



Rys. 1. Wymiana danych przez transfer. Źródło: PN-EN ISO 19109, 2009



Rys. 2. Wymiana danych przez transakcje. Źródło: PN-EN ISO 19109, 2009

je się do usług i w ten sposób dane są przekazywane od dostawcy usługi użytkownikowi aplikacji. Schemat aplikacyjny opisuje nie tylko strukturę i zawartość wymienianych danych, ale także strukturę interfejsów biorących udział w transakcji.

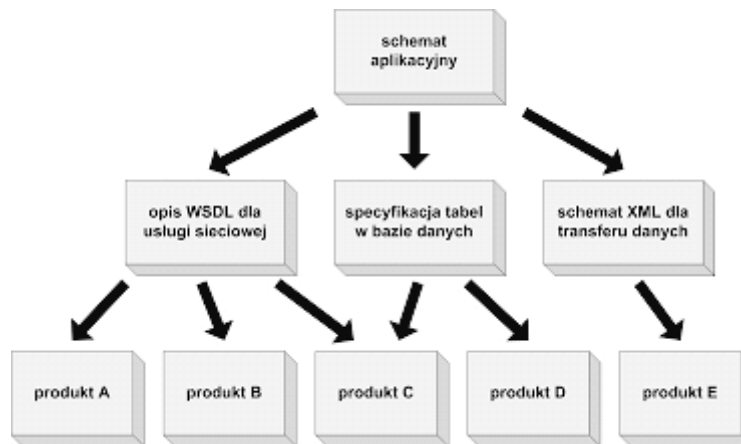
Stosując transfer danych, zbiór danych jest predefiniowany w schemacie aplikacyjnym. Zasięg przestrzenny i reguły włączania instancji obiektów są także predefiniowane. Użytkownik wysyła polecenie i otrzymuje kopię zbioru danych (albo otrzymuje ją automatycznie dzięki długoterminowej umowie dotyczącej dystrybucji zbioru danych). Wykorzystując transakcję danych, użytkownik najpierw określa kryteria wyboru, takie jak zasięg przestrzenny i reguły włączania instancji obiektów danych od dostawcy danych. Dane, które spełniają kryteria wyszukiwania są pobierane z miejsca ich składowania i dostarczane użytkownikowi.

Model transakcji jest stosowany przy dużej liczbie operacji i prostych wymianach. Model transferu jest stosowany przy małej liczbie operacji i dużych zbiorach danych.

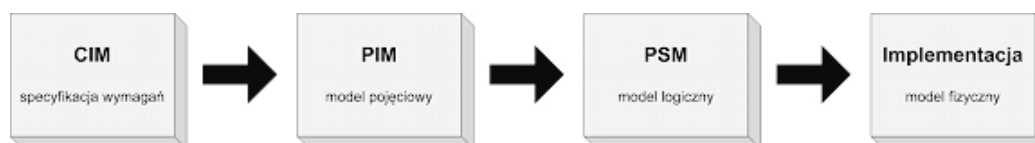
## Perspektywa danych

Z perspektywą danych związane jest podejście oparte na modelu (ang. *Model Driven Approach*) (rys. 3), w którym niezależny od implementacji schemat aplikacyjny odwzorowany zostaje na specyfikacje (pod kątem różnorodnych technik), a te z kolei zaimplementowane (wdrożone) na różnych platformach implementacyjnych. Koncepcja ta zgodna jest z architekturą opisaną w specyfikacjach OMG jako MDA (ang. *Model Driven Architecture*) (rys. 4), w której wyróżnia się cztery modele systemu:

- CIM (ang. *Computation Independent Model*) – model domenowy systemu, stanowiący specyfikację wymagań systemu,
- PIM (ang. *Platform Independent Model*) – model systemu niezależny od platformy sprzętowej i programowej (model pojęciowy),
- PSM (ang. *Platform Specific Model*) – model systemu zależny od wybranej platformy sprzętowej i programowej, odzwierciedlający model PIM dla konkretnej platformy (model logiczny),
- Implementacja (ang. *Implementation Model*) – fizycznie działający program zapisany w konkretnym języku programowania (model fizyczny).

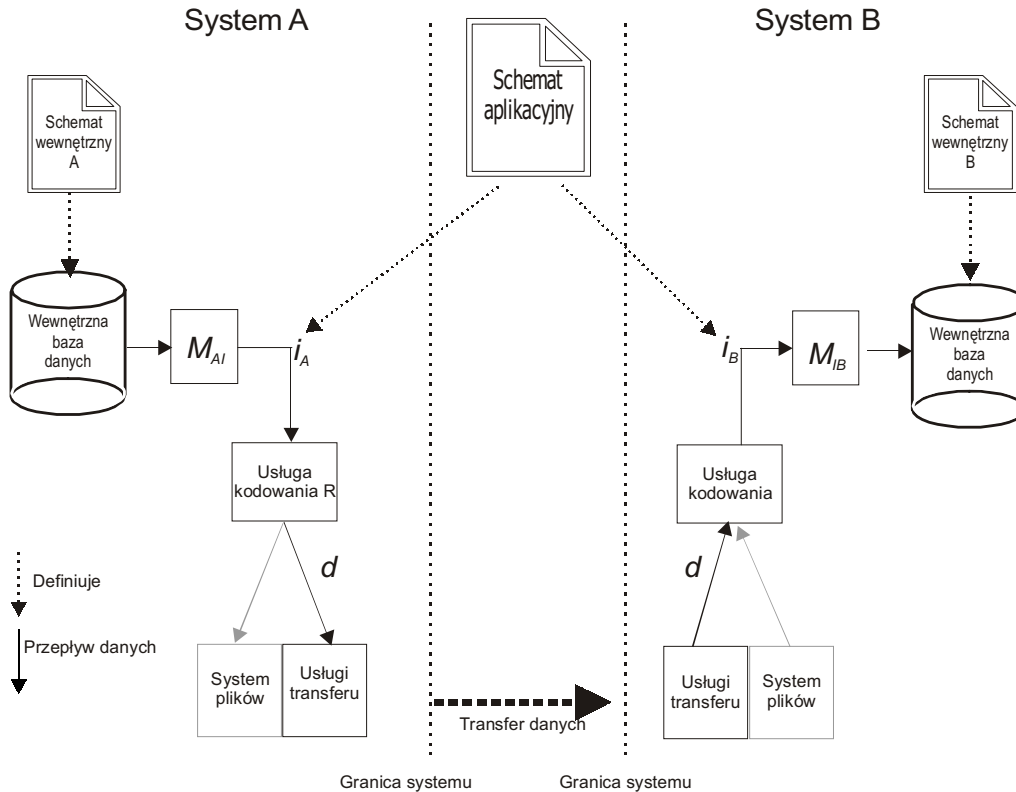


Rys. 3. Podejście oparte na modelu. Źródło: prCEN/TR 15449, 2006



Rys. 4. Modele systemu wg technologii MDA

W modelu transferu (rys. 5), zgodnie z normą ISO 19118 (PN-EN ISO, 2006), systemy przechowują dane w wewnętrznych bazach danych, które są zgodne z wewnętrznymi schematami. Pierwszym krokiem w wymianie danych z systemu *A* do systemu *B* jest przełożenie danych systemu *A* na strukturę danych, która jest zgodna ze schematem aplikacyjnym. W tym celu definiowane jest odwzorowanie  $M_{AI}$  pojęć z wewnętrznego schematu na pojęcia



Rys. 5. Model transferu danych. Źródło: PN-EN ISO 19118, 2006

zdefiniowane w schemacie aplikacyjnym, jak również wykorzystany (napisany) odpowiedni program do przełożenia instancji danych. Wynikiem jest struktura danych zgodna ze schematem aplikacyjnym  $i_A$ . Struktura danych jest przechowywana w pamięci lub w pliku pośrednim i jest zależna od systemu. Jest ona nieodpowiednia do transferu.

Kolejnym krokiem jest użycie usługi kodowania, która stosuje regułę  $R$  w celu stworzenia struktury danych, która jest niezależna i odpowiednia do transferu. Zakodowany zbiór danych  $d$  jest przechowywany w systemie plików lub przeniesiony przez usługę transferu. System  $A$  uruchamia usługę transferu w celu wysłania zbioru danych  $d$  do systemu  $B$ . Usługa transferu postępuje zgodnie z protokołem transferu (jak spakować dane, jak powinien wyglądać transport przy połączeniu on-line i off-line). Oba systemy muszą uzgodnić protokół transferu.

Usługa transferu w systemie  $B$  otrzymuje zbiór danych i zgodnie z protokołem zbiór danych jest rozpakowywany i przechowywany jako zbiór zakodowany  $d$  np. w pliku pośrednim. Aby otrzymać strukturę danych, która jest zgodna ze schematem aplikacyjnym  $i_B$  system  $B$  stosuje regułę rozkodowania  $R^{-1}$ , która interpretuje zakodowane dane. Aby wykorzystać zbiór danych, system  $B$  musi przełożyć strukturę danych  $i_B$  na wewnętrzną bazę danych. W tym celu definiowane jest odwzorowanie  $M_{IB}$  schematu aplikacyjnego na wewnętrzny schemat i wykorzystywane (pisane) oprogramowanie, które dokonuje tłumaczenia.

Zgodnie z rekomendacjami zawartymi w normach ISO serii 19100, kodowanie powinno być wykonane w XML i GML. Norma ISO 19119 (PN-EN ISO, 2006) podaje reguły konwersji schematu, które definiują jak budować dokument XML Schema (XSD) zgodnie ze schematem aplikacyjnym wyrażonym w UML. Odzworowanie schematów aplikacyjnych UML na schematy aplikacyjne GML zawarte są w załączniku E normy ISO 19136 (PN-EN ISO, 2009).

Schemat aplikacyjny zgodnie z zaleceniami norm ISO serii 19100 powinien być zapisany w języku UML, który jest notacją metodyk obiektowych. Jednak z punktu widzenia implementacji mamy do czynienia z relacyjnymi bazami danych. Ma to wpływ na różnice w plikach XSD.

Wygenerowanie pliku GML z uzgodnionego schematu aplikacyjnego UML (w danej dziedzinie, organizacji, itp.) jest możliwe, dzięki aplikacji *ShapeChange*. Stosując to narzędzie należy pamiętać, że taki schemat powinien być zgodny z normą 19109 (PN-EN ISO, 2009) i 19103 (ISO/TS, 2005). Ponadto należy go wyeksportować do formatu XMI uwzględniając wersję języka UML 1.3.

Zgodnie z rysunkiem 5, w przypadku, gdy strukturę danych wewnętrznej bazy danych systemu A zgodną z uzgodnionym schematem aplikacyjnym  $i_A$ , chcemy wyeksportować do pliku XSD, otrzymujemy pliki XSD, których struktura jest z góry określona przez dane oprogramowanie, np. MS Access. Problemem jest niezgodność otrzymanych plików XSD z plikami XSD będącymi wynikiem przekształceń UML-GML, ponieważ w istniejących narzędziach brakuje funkcjonalności pozwalającej na takie przekształcenie. Na przykład po eksporcie bazy danych (tabele i relacje) pod MS Access do XML otrzymujemy oddzielne pliki XSD dla każdej tabeli. Natomiast w wyniku przekształcenia schematu aplikacyjnego takiej bazy danych w UML (klasy/tabele i związki/relacje) do pliku XSD otrzymujemy tylko jeden plik.

Propozycją autorki jest następujące rozwiązanie:

- albo rozszerzenie funkcjonalności każdego oprogramowania bazodanowego o możliwość eksportu danych do uzgodnionego pliku XSD,
- albo opracowanie narzędzia pozwalającego na konwersję pliku XSD na uzgodniony plik XSD.

Rozwiązanie drugie wydaje się być bardziej optymalne ze względów ekonomicznych.

Podobne problemy pojawiają się po stronie odbiorcy, kiedy zachodzi potrzeba konwersji struktury z uzgodnionego XSD do wewnętrznej struktury danych  $i_B$  zgodnej z uzgodnionym schematem aplikacyjnym. Oddzielnym zagadnieniem jest również sprawdzenie poprawności danych z ustaloną strukturą (plik XSD).

Wyżej opisane podejście do implementacji schematów aplikacyjnych koncentruje się wyłącznie na bezpośrednim transferze danych między różnymi systemami. Nie uwzględnia możliwości zastosowania w tym procesie usług sieciowych – udostępniania i pobierania danych.

Pobieranie danych przestrzennych może być realizowane na trzy sposoby:

- dostęp do danych przez plik umieszczony pod adresem URL,
- transfer danych za pomocą usług sieciowych,
- transfer danych przestrzennych przez media fizyczne.

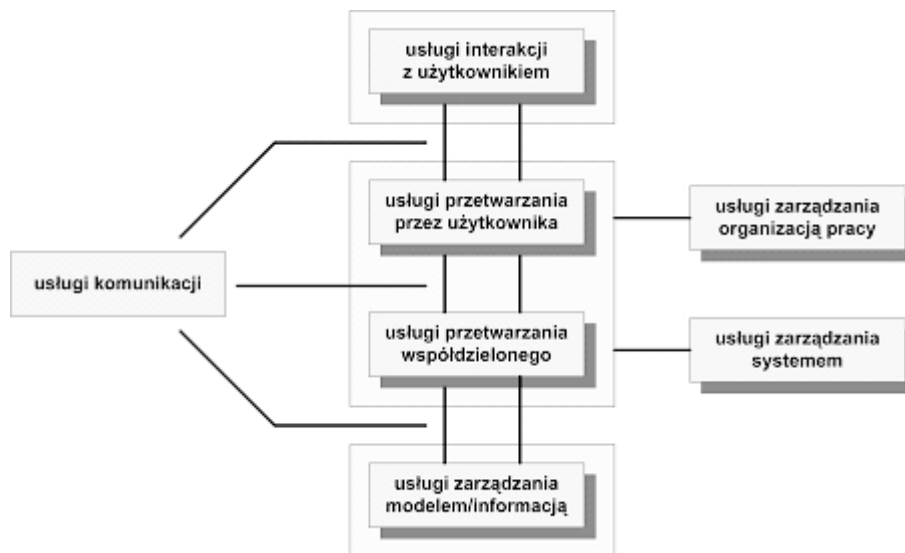
Zagadnienia sposobów pobierania danych przestrzennych w ramach INSPIRE omawia dokument *Guidelines for the encoding of spatial data* (INSPIRE Drafting Team *Data Specifications*, 2008).

## Perspektywa usług

Z perspektywy usług, dostęp do danych nie powinien się opierać na przechowywaniu danych, tylko na usługach w sieci. Transfer danych zależy od usługi. Bezpośredni dostęp do baz danych nie jest rozważany w perspektywie usług. Rozwiązanie jest rozpatrywane w perspektywie danych.

Perspektywa usług w ujęciu *Referential Model of Open Distributed Processing* (RM-ODP, model wzorcowy otwartego przetwarzania rozproszonego) jest rozpatrywana z punktu widzenia technologicznego, komputerowego, projektowego, inżynierskiego oraz informacyjnego.

Z punktu widzenia inżynierskiego architektury IT są zorganizowane w wielowarstwowe architektury dystrybucji. Logiczną architekturę tworzą usługi i związane z nimi interfejsy (rys. 6). Fizyczną architekturę tworzą komponenty i związane z nimi interfejsy, które implementują usługi.

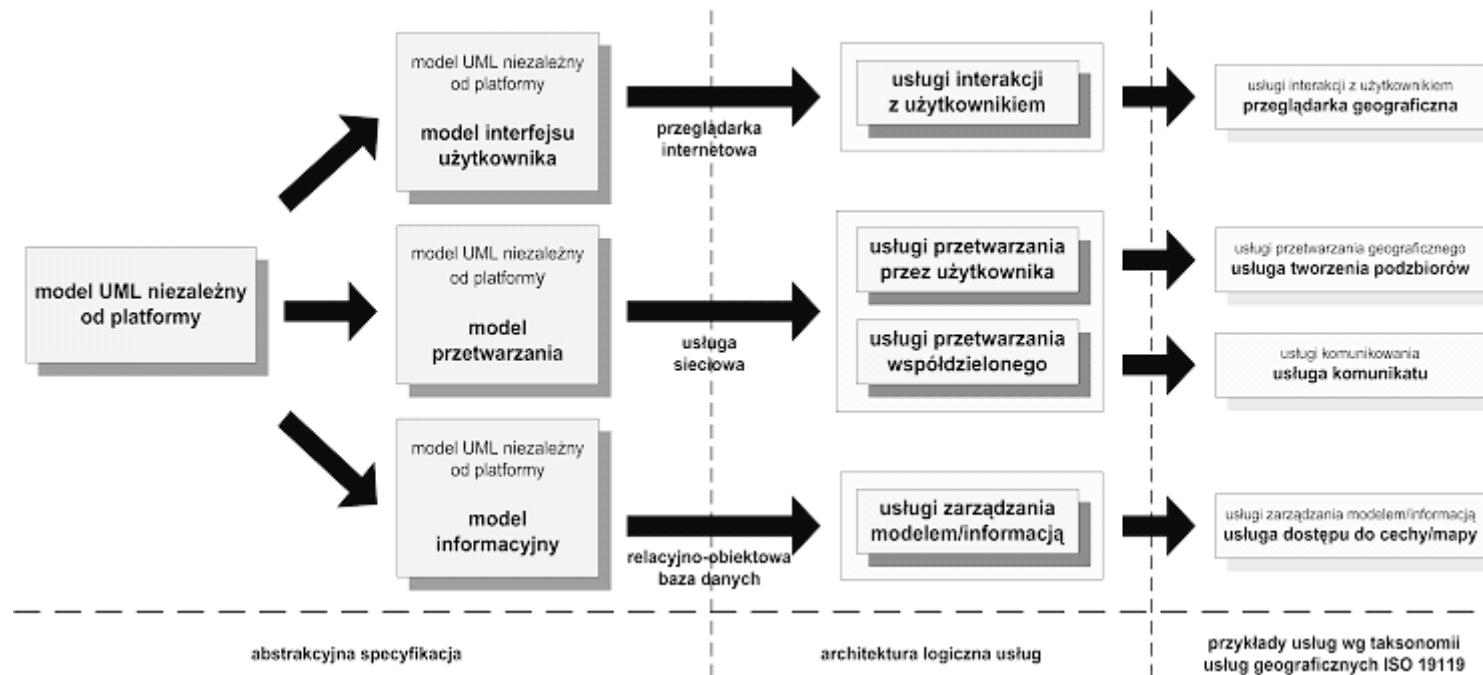


Rys. 6. Logiczna architektura usług. Źródło: PN-EN ISO 19119, 2006

Logiczna architektura może być odwzorowana na różne architektury fizyczne. Wszystkie warstwy mogą być przekształcone albo na jedną monolityczną aplikację (np. ArcGIS, GeoMedia Professional), albo wykorzystując różne fizyczne architektury klient-serwer (np. ArcIMS, GeoMedia WebMap Professional).

W modelu interakcji schemat aplikacyjny UML można podzielić na części, które będą opisywały specyfikacje dotyczące np. interfejsu użytkownika, aspektów informacyjnych. Poszczególne części są odwzorowane na różne technologie, m.in. na technologię usług sieciowych, np. WMS (ang. *Web Map Service*, prezentacja danych przestrzennych), WFS (ang. *Web Feature Service*, pobierania danych przestrzennych w formacie GML) (rys. 7).

Warstwom architektury logicznej można przyporządkować konkretne usługi geograficzne wg taksonomii usług geograficznych ISO 19119 (PN-EN ISO, 2006) (rys. 7).



Rys. 7. Przykład odwzorowania modelu UML niezależnego od platformy na technologię usług sieciowych (np. WMS, WFS)

Poniżej przedstawiono możliwość odwzorowania modelu UML niezależnego od platformy na technologię usług sieciowych na przykładzie usługi WMS i WFS.

Model interfejsu użytkownika opracowano w postaci ogólnego diagramu czynności (rys. 8) przedstawiającego komunikację użytkownika z przeglądarką internetową, która pełni rolę przeglądarki geograficznej (rys. 9) dla usługi WMS.



Rys. 8. Model interfejsu użytkownika – diagram czynności



Rys. 9. Przykład przeglądarki internetowej

Model przetwarzania przedstawiono za pomocą diagramów sekwencji obrazujących komunikację użytkownika z serwisami WMS i WFS (rys. 10 i 11). Przykład usługi tworzenia podzbiorów to możliwość wydania zapytania GetMap (przesłanie obrazu mapy spełniającej warunki zapytania) do serwisu WMS:

```

http://ngis/wmsconnector/com.esri.wms.Esrimap?SERVICE=WMS&VERSION=1.1.1
&REQUEST=GetMap&LAYERS=15,14,13,12,8,7,5,4,3,2&STYLES=&SRS=EPSG:3120
&BBOX=4556516.933854,5577507.542804,4558701.230502,5579317.874267
&WIDTH=600&HEIGHT=400&FORMAT=image/png
&SERVICENAME=UzbTerenu65_ISAM&
  
```

Innym przykładem usługi tworzenia podzbiorów jest polecenie GetFeature (przesłanie konkretnych obiektów spełniających warunki klienta podane w poleceniu) do serwisu WFS:

```

http://ngis/wfsconnector/com.esri.wfs.Esrimap/UzbTerenu65_FSaxI?request=getfeature
&typename=GOdcienkPrzewodu-4&service=wfs&version=1.0.0
  
```

Natomiast usługa komunikatu, to możliwość obejrzenia kolekcji danych jednocześnie przez wielu użytkowników, zarówno w przypadku usługi WMS jak i WFS.

Model informacyjny opracowano w postaci diagramu klas (rys. 12), a związaną z nim usługę zarządzania modelem/informacją przedstawiono na przykładzie usługi dostępu do mapy



– zapytanie *GetMap* do serwisu WMS (rys. 13) oraz usługi dostępu do cechy – zapytanie *GetFeature* do serwisu WFS (rys. 14) zwracające w wyniku dane w formacie GML.

## Podsumowanie

Schematy aplikacyjne zapewniają uzgodniony opis danych oraz umożliwiają osiągnięcie powszechnego i poprawnego rozumienia danych. Są podstawą do osiągnięcia interoperacyjności środowisk narzędziowych i swobodnego transferu danych. Poprawne działanie systemów w ramach infrastruktur danych przestrzennych jest uzależnione od odpowiedniej ich implementacji. Wdrażanie uzgodnionych schematów aplikacyjnych jest zagadnieniem złożonym, które wymaga rozpatrzenia wielu aspektów. Jednym z aspektów jest ustalenie perspektywy infrastruktury (danych, usług sieciowych). Z poszczególną perspektywą związana jest określona metodologia implementacji schematów aplikacyjnych. Wybór odpowiedniego modelu zdeterminowany jest potrzebami oraz narzędziowymi i technicznymi możliwościami danej organizacji, dla której tworzone są schematy aplikacyjne, a także przyszłym wykorzystaniem zbiorów danych.

## Literatura

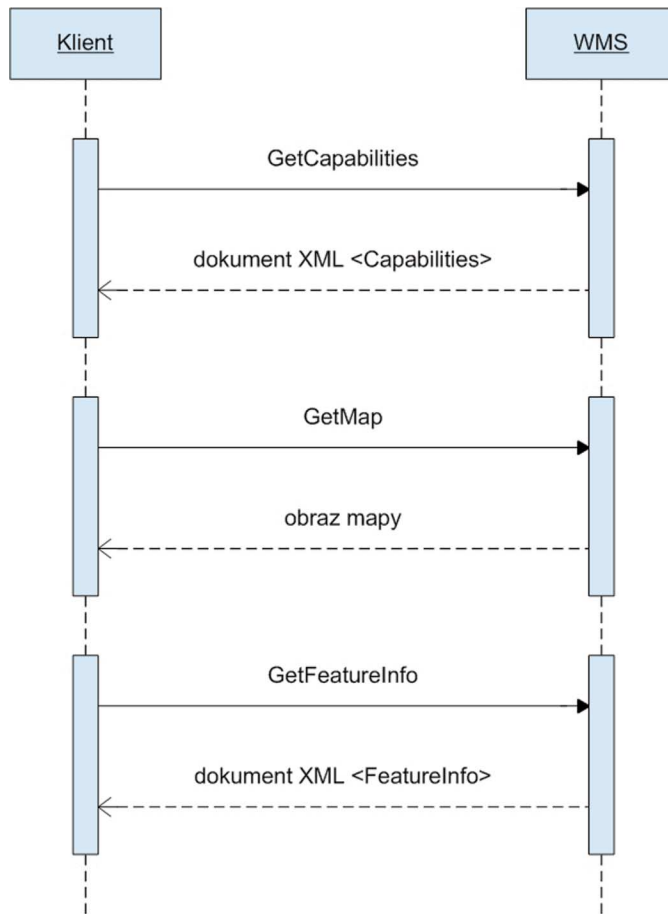
- Dyrektywa 2007/2/WE Parlamentu Europejskiego i Rady z dnia 14 marca 2007 r. ustanawiająca infrastrukturę informacji przestrzennej we Wspólnocie Europejskiej (INSPIRE), 2007. *Dziennik Urzędowy Unii Europejskiej*.
- INSPIRE Drafting Team *Data Specifications*, 2008. D2.7: Guidelines for the encoding of spatial data, Version 3.0.
- ISO/CD 19142:2006 *Geographic information – Web feature service*.
- ISO/TS 19103:2005 *Geographic information – Conceptual schema language*.
- MDA Guide Version 1.0.1, 2003. <http://www.omg.org/mda>
- PN-EN ISO 19109:2009 Informacja geograficzna – Reguły schematów aplikacyjnych.
- PN-EN ISO 19118:2006 Informacja geograficzna – Kodowanie.
- PN-EN ISO 19119:2006 Informacja geograficzna – Usługi.
- PN-EN ISO 19128:2008 Informacja geograficzna – Interfejs internetowego serwera map.
- PN-EN ISO 19136:2009 Informacja geograficzna – Język znaczników geograficznych GML.
- prCEN/TR 15449, 2006. *Geographic information – Standards, specifications, technical reports and guidelines, required to implement Spatial Data Infrastructure*.

## Abstract

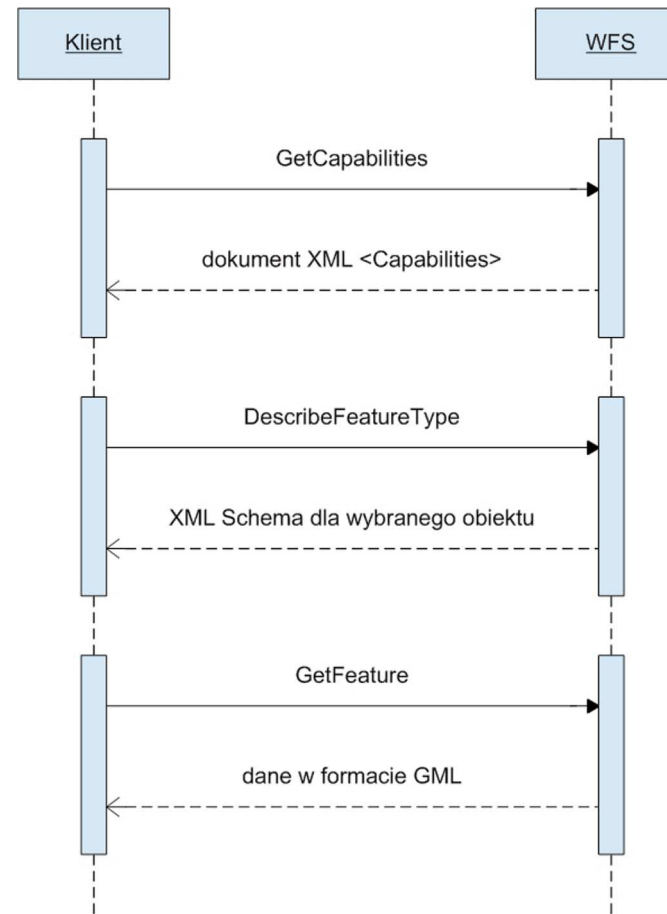
*The paper is concerned with issues connected with implementation aspects of application schemes within the framework of building spatial data infrastructures. Each SDI can be seen from both data-centric and service-centric view. Implementation of the application schemes is a complex issue. Many aspects must be considered. The way of investigating application schemes is influenced by the infrastructure view assumed. The view determines the methods of applications' implementation. The authors describes the realization of the processes of application schemes' implementation leading to spatial data interchange and to interoperability of the geoinformation systems.*

dr inż. Agnieszka Chojka  
agnieszka.chojka@uwm.edu.pl  
tel. +48 89 523 35 86

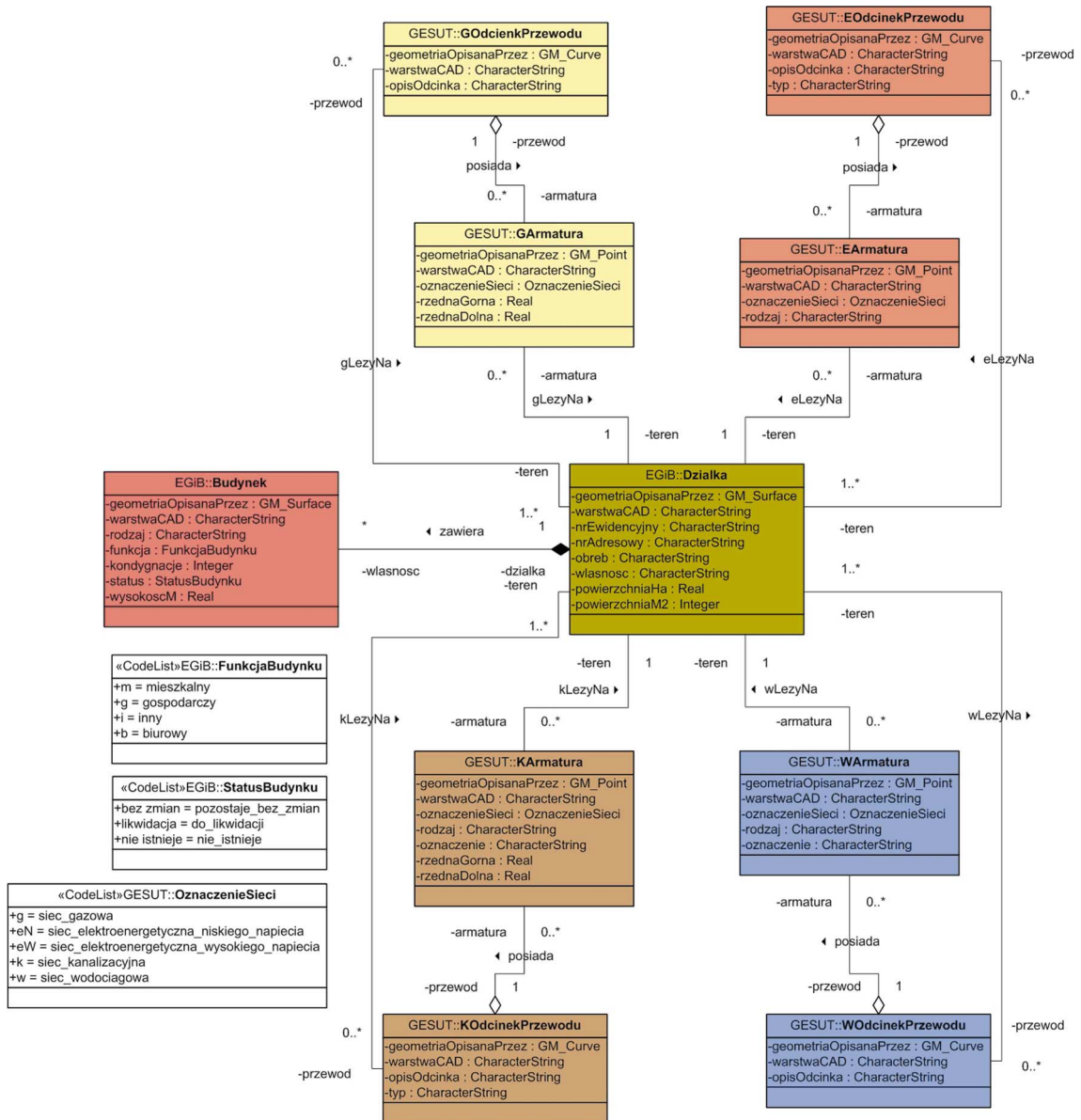
dr inż. Agnieszka Zwirowicz  
agnieszka.zwirowicz@uwm.edu.pl  
tel. +48 89 523 42 53



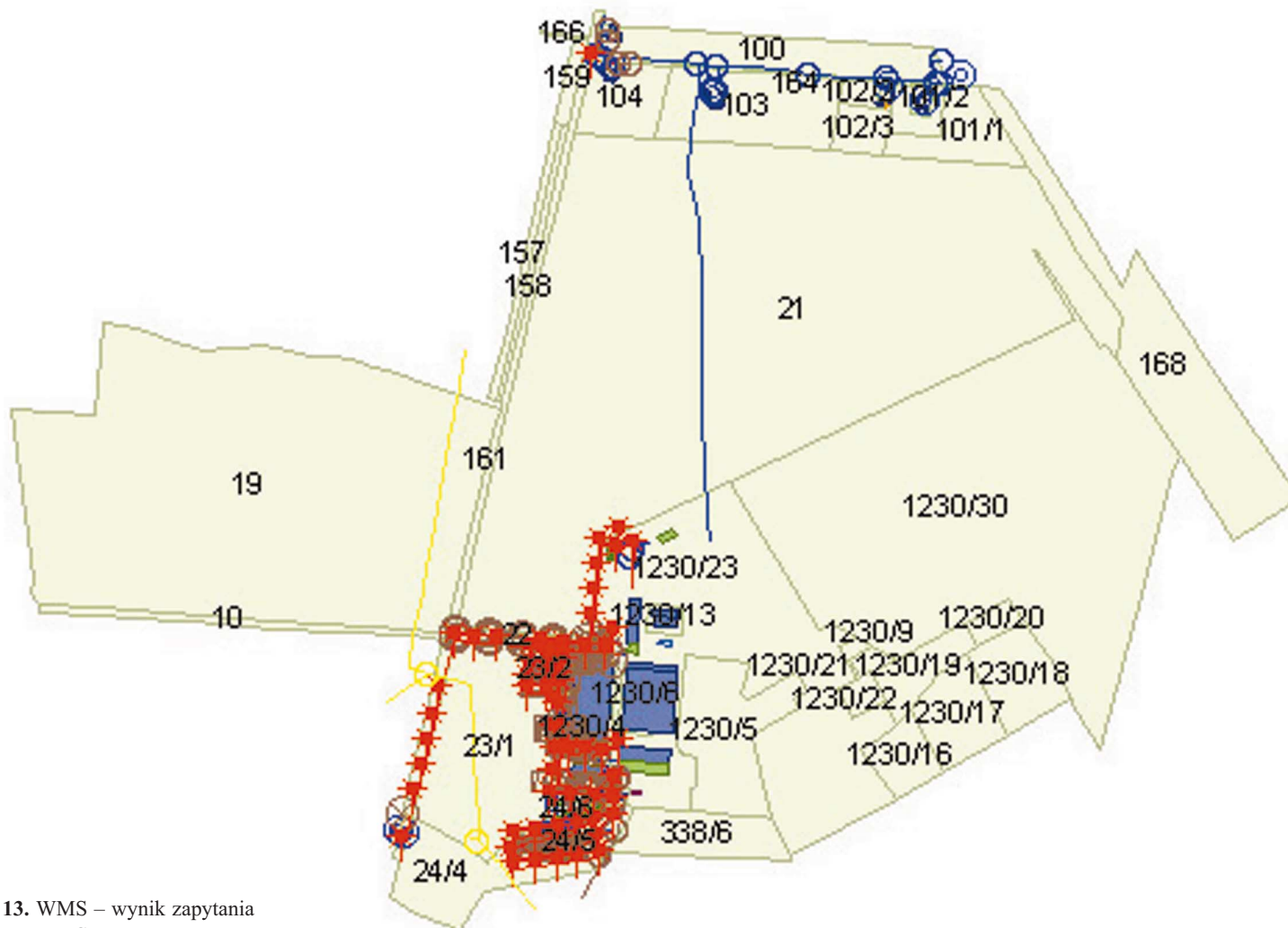
**Rys. 10.** Model przetwarzania dla usługi WMS  
– diagram sekwencji



**Rys. 11.** Model przetwarzania dla usługi WFS  
– diagram sekwencji



Rys. 12. Przykład modelu danych – diagram klas



Rys. 13. WMS – wynik zapytania  
*GetMap*

```

- <esri:G0dcienkPrzewodu-4 fid="G0dcienkPrzewodu-4.1">
  - <gml:multiLineStringProperty>
    - <gml:MultiLineString srsName="EPSG:4326">
      - <gml:lineStringMember>
        - <gml:LineString srsName="EPSG:4326">
          <gml:coordinates>4557284.28079056,5578461.57351391 4557274.34281897,5578408.91596932
            4557271.65574282,5578392.41992615 4557270.71210917,5578388.65942463
            4557264.17332371,5578353.29071538 4557257.39599877,5578315.50504209
            4557241.74185181,5578231.94964569</gml:coordinates>
          </gml:LineString>
        </gml:lineStringMember>
      </gml:MultiLineString>
    </gml:multiLineStringProperty>
    <esri:enabled>0</esri:enabled>
    <esri:warstwa_ca>Gaz</esri:warstwa_ca>
    <esri:opis_odcin>gA200</esri:opis_odcin>
    <esri:shape_leng>233.543554511</esri:shape_leng>
  </esri:G0dcienkPrzewodu-4>
</gml:featureMember>
- <gml:featureMember>
- <esri:G0dcienkPrzewodu-4 fid="G0dcienkPrzewodu-4.2">
  - <gml:multiLineStringProperty>
    - <gml:MultiLineString srsName="EPSG:4326">
      - <gml:lineStringMember>
        - <gml:LineString srsName="EPSG:4326">
          <gml:coordinates>4557241.74185181,5578231.94964569 4557232.05469722,5578178.98164916
            4557231.94419732,5578178.0502936 4557219.42964944,5578109.62810929
            4557213.0417049,5578073.16667968</gml:coordinates>
          </gml:LineString>
        </gml:lineStringMember>
      </gml:MultiLineString>
    </gml:multiLineStringProperty>
    <esri:enabled>0</esri:enabled>
    <esri:warstwa_ca>Gaz</esri:warstwa_ca>

```

Rys. 14. WFS – wynik zapytania *GetFeature* – fragment pliku GML