



**POLSKIE
TOWARZYSTWO
INFORMACJI
PRZESTRZENNEJ**

ROCZNIKI 2012 **GEOMATYKI**

**Modele danych przestrzennych w UML
i ich transformacja do schematów GML
i struktur baz danych**

**Tom X
Zeszyt 1(51)
Warszawa**

Agnieszka Chojka

7. Transformacja schematu aplikacyjnego UML do schematu aplikacyjnego GML – wymagania, ograniczenia i wybrane narzędzia

Rozdział prezentuje metody transformacji schematu aplikacyjnego UML na odpowiadający mu schemat aplikacyjny GML oraz aplikacje komputerowe wspomagające takie przekształcenia. W rozdziale opisano również różnice w schematach aplikacyjnych GML otrzymanych przy zastosowaniu różnych metod.

7.1. Metody transformacji UML do GML

Można wyróżnić dwie metody przekształcania schematu aplikacyjnego UML na schemat aplikacyjny GML: metodę ręczną i metodę automatyczną. W tabeli 7.1 zestawiono wymagania dla obu metod, jak również ich wady i zalety.

Tabela 7.1. Porównanie metod transformacji schematu aplikacyjnego UML na schemat aplikacyjny GML

Metoda	Wymagania	Zalety	Wady
Ręczna	<ul style="list-style-type: none">– dobra znajomość normy ISO 19136, w szczególności załącznika E (reguła kodowania schematów aplikacyjnych UML-GML)– narzędzie (darmowe lub komercyjne) wspomagające tworzenie plików XSD (XML Schema), np. <i>XMLSpy</i>	<ul style="list-style-type: none">– pełna kontrola procesu transformacji	<ul style="list-style-type: none">– metoda żmudna i pracochłonna– nietrudno o pomyłkę
Automatyczna	<ul style="list-style-type: none">– podstawowa znajomość normy ISO 19136– oprogramowanie (komercyjne) <i>Rational Rose</i> lub <i>Enterprise Architect</i> umożliwiające właściwe przygotowanie schematu aplikacyjnego UML (plik profilu UML z metkami)– oprogramowanie (bezpłatne) umożliwiające transformację UML-GML, np. <i>ShapeChange</i> lub <i>FullMoon</i>	<ul style="list-style-type: none">– metoda szybka– gwarancja zgodności z normami ISO serii 19100	<ul style="list-style-type: none">– konieczność instalacji wielu aplikacji– skomplikowana konfiguracja oprogramowania (zwłaszcza <i>FullMoon</i>)– ustawienie odpowiednich metek w UML

7.2. Metoda ręczna

Metoda ręczna wymaga przede wszystkim bardzo dobrej znajomości norm ISO serii 19100, w szczególności ISO/TS 19103 i ISO 19109, które określają zasady budowy schematów aplikacyjnych UML oraz normy ISO 19136, załącznik E, w którym zdefiniowano regułę kodowania schematów aplikacyjnych UML do GML.

W metodzie tej wystarczy opracować schemat aplikacyjny UML w dowolnym narzędziu wspomagającym tworzenie diagramów klas UML (równie dobrze taki schemat można narysować na kartce papieru), a następnie korzystając z normy ISO 19136 napisać kod GML odpowiadający schematowi aplikacyjnemu UML. Najlepiej skorzystać tutaj z oprogramowania (bezpłatnego lub komercyjnego), które wspomaga tworzenie plików XSD (XML Schema) i dokonuje ich walidacji (kontrola poprawności składniowej). Przykładem takiego oprogramowania (komercyjnego) jest *XMLSpy* firmy Altova (rys. 7.1). Aplikacja ta, to najbardziej wszechstronne narzędzie do projektowania zaawansowanych aplikacji XML:

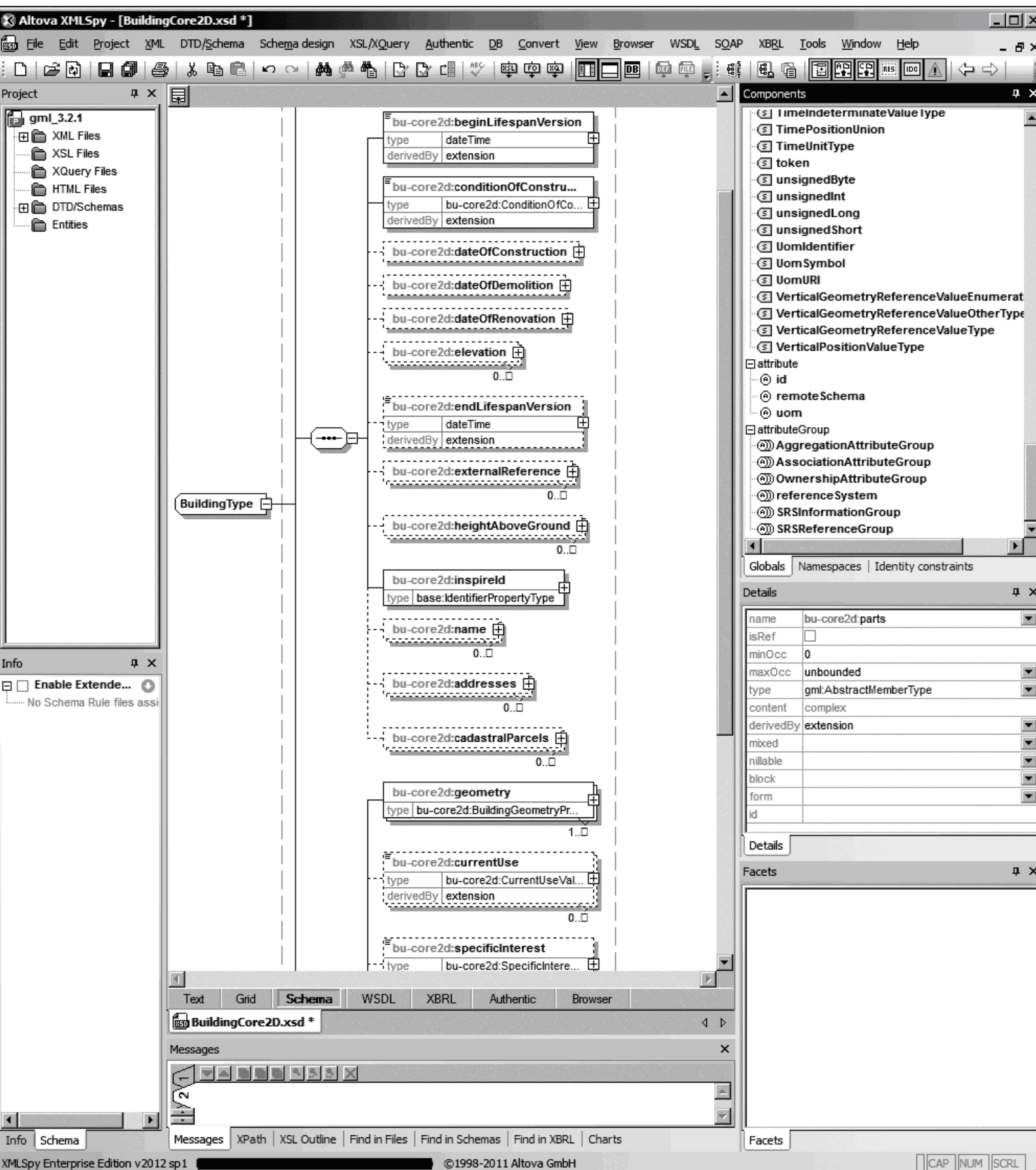
- stanowi standard przemysłowy w dziedzinie środowiska projektowego dla projektowania, edytowania i debugowania zaawansowanych aplikacji wykorzystujących technologie takie jak XML, XML Schema, XSL/XSLT, SOAP, WDSL i Web services,
- umożliwia automatyczne generowanie kodu dla aplikacji w wielu językach programowania,
- rozszerza i uzupełnia technologie takie jak J2EE, .NET, Eclipse oraz jest podstawowym narzędziem dla programistów wykorzystujących najnowsze technologie XML, Web services oraz bazy danych.

W zależności od możliwości, plik GML (XSD) można również opracować w zwykłym notatniku, a następnie skorzystać z dostępnych w Internecie darmowych walidatorów webowych plików XSD:

- walidator dla XML Schema REC (20010502) version (rys. 7.2), dostępny pod adresem: <http://www.w3.org/2001/03/webdata/xsv>,
- XML Schema Validator (rys. 7.3), dostępny pod adresem: <http://schneegans.de/sv/>.

Ręczne opracowanie schematu aplikacyjnego GML (np. w *XMLSpy*) mogłoby przebiegać w następujących krokach:

- 1) utworzenie nowego pliku XSD,
- 2) przypisanie odpowiedniej przestrzeni nazw.
- 3) zaimportowanie schematu gml (gml.xsd) z <http://schemas.opengis.net/gml/3.2.1/gml.xsd>,
- 4) utworzenie typów prostych i typów złożonych,
- 5) dodanie elementów (odpowiedniki atrybutów w UML) do typów złożonych,
- 6) ustawienie liczności elementów będących odpowiednikami atrybutów w UML,
- 7) dodanie elementów do typów właściwości,
- 8) ustawienie atrybutów powiązania (roli),
- 9) zdefiniowanie zawiązków między elementami (klasami UML) – dodanie atrybutów-ról do właściwych klas i przypisanie im odpowiednich typów, ustawienie krotności,
- 10) opcjonalnie dopisanie `<gml:reverseProperty>` w powiązaniach dwukierunkowych,
- 11) przeprowadzenie walidacji utworzonego schematu GML (tak naprawdę powinna odbywać się po każdej operacji, ponieważ łatwiej wtedy wychwycić błędy w schemacie),
- 12) zapisanie pliku projektu.



Rys. 7.1. Środowisko pracy w XML Spy

Validator for XML Schema REC (20010502) version, as amended

XSV version: XSV 3.1-1 of 2007/12/11 16:20:05

NOTICE: This is a service for a W3C **approved recommendation**. This version is for schema documents with the namespace URI <http://www.w3.org/2001/XMLSchema> and is being actively developed: see [XSV for XML Schema 20000922 version](#) for the no longer maintained previous version, for schema documents with the namespace URI <http://www.w3.org/2000/10/XMLSchema>, and [XSV for XML Schema 200004007 version](#) for the no longer maintained even earlier version, for schema documents with the namespace URI <http://www.w3.org/1999/XMLSchema>.

Use this form for checking a schema which is accessible via the Web, and/or schema-validating an instance with a schema of your own.

Address(es):

Show warnings Keep Going Check as complete schema

Default output is now text/xml with an XSLT stylesheet. Select fallbacks for browsers which don't support <http://www.w3.org/1999/XSL/Transform> stylesheets:

- text/xml + official XSLT (suits IE5 and greater, Mozilla)
- text/xml + early MS XSL support (suits vanilla IE5)
- text/html (styled server-side: suits Netscape, older IE)
- xml, but labelled text/plain (works for any browser, but hard to read)

Get Results

Rys. 7.2. Walidator dla XML Schema REC (20010502) version

XML Schema Validator

This service lets you validate XML documents such as XHTML against the appropriate [schemas](#). It performs a more accurate validation than the W3C validator. If you send XHTML documents as `text/html`, you should also test them in [XHTML Proxy](#).

This service does **not** perform generic XML Schema validation. While it can verify the *well-formedness* of any XML document, *validation* is only available with the built-in schemas. This validator does not download any DTD or schema files, but relies on its own ones.

You can validate XML documents by [URL](#), by [file upload](#) or by [direct input](#). Furthermore, you can validate [fragments](#) of XML documents. You can easily validate the page that is currently displayed in your browser with [this bookmarklet](#).

Validate by URL

Address:
Schema:
Encoding:
HTML compatibility:

Validate by file upload

File:
Schema:
Encoding:
HTML compatibility:

Rys. 7.3. XML Schema Validator opracowany przez Christpha Schneegansa

7.3. Metoda automatyczna

Metoda automatyczna wymaga przede wszystkim odpowiedniego oprogramowania, które umożliwi automatyczne wygenerowanie schematu aplikacyjnego GML ze schematu aplikacyjnego UML. Obecnie dostępne są dwa takie narzędzia: *ShapeChange* i *FullMoon*, oba bezpłatne. Jednak wymagają one dodatkowo wykorzystania komercyjnego oprogramowania *Enterprise Architect* lub *Rational Rose* pozwalającego na właściwe przygotowanie wyjściowego schematu aplikacyjnego UML (plik profilu UML z odpowiednimi metkami). Dla programu *Rational Rose* odpowiednie pliki konfiguracyjne są dołączone do oprogramowania *ShapeChange* (w katalogu *config*).

Program ShapeChange

Oprogramowanie pozwalające na generowanie poprawnych schematów aplikacyjnych GML (GML 3.2.1) ze schematów aplikacyjnych UML, przy czym modele UML należy opracować zgodnie z zasadami określonymi w dokumencie „UGAS Guidelines and Encoding Rules” (Portele, 2008a). *ShapeChange* wymaga zapisania modeli UML w formacie XMI (ang. *XML Metadata Interchange*), który pełni rolę pośrednika w wymianie modeli między różnymi narzędziami. Jest to aplikacja Java, więc może być uruchomiona z poziomu wiersza poleceń. Różne parametry poleceń pozwalają na dostosowanie reguł kodowania schematu aplikacyjnego UML na GML.

ShapeChange jest usługą kodowania zgodnie z normą ISO 19118 i implementuje operacje generowania XML Schema (*generateXMLSchema*).

Aktualnie dostępną publicznie wersją tego oprogramowania to 1.0. Została opracowana na podstawie następujących dokumentów ISO/TC 211 i OGC:

- GML 3.2.1 (ISO/TC 211, 2007a lub Portele, 2007),
- ISO/TS 19139:2007,
- ISO 19118 rev. 1 (draft),
- ISO/TS 19103:2005 and ISO 19103 (draft),
- ISO 19109:2005 (Portele, 2008b).

HollowWorld i FullMoon

HollowWorld jest środowiskiem, które umożliwia ekspertom (w danej dziedzinie) wykorzystującym informację przestrzenną (geograficzną) na tworzenie schematów pojęciowych dla ich dziedzin zastosowań, które z kolei odpowiadają normom międzynarodowym w zakresie interoperacyjności informacji przestrzennej. Tak opracowane modele mogą zostać łatwo przekształcone na odpowiadające im schematy GML (XML Schema), które określają format dokumentu na potrzeby przesyłania danych (z danej dziedziny) jako standardowy dokument XML, zgodny ze specyfikacją WFS opracowaną przez OGC.

HollowWorld umożliwia wykorzystanie specjalnego szablonu UML, który zawiera komponenty opisane w normach ISO serii 19100, głównie z ISO/TC 211 „Harmonized Model” (ISO/TC 211, 2012).

Ponadto dla użytkowników oprogramowania *Enterprise Architect* (firmy Sparx Systems) opracowano profil UML zawierający standardowe stereotypy i metki określone przez normę ISO 19136. Implementacja modelu w XML Schema może zostać wygenerowana automatycznie, przy czym model UML musi stosować się do reguł opisanych w załączniku E do normy ISO 19136, a w szczególności musi zawierać:

- określone kierunki nawigacji i nazwy ról dla powiązań asocjacyjnych,
- tylko standardowe stereotypy,
- wymagane metki.

FullMoon XML Processing Framework to oprogramowanie narzędziowe, które umożliwia łatwe przekształcenie schematu aplikacyjnego na jego reprezentację w XML Schema. Zostało opracowane na potrzeby przetwarzania dużych modeli UML według reguł kodowania określonych w normach ISO 19118, ISO 19136 oraz 19139. *FullMoon* przetwarza model UML zapisany w formacie XMI, tworząc XML Schema (Githaiga, 2010; Cox, 2011).

Oprogramowanie Enterprise Architect (firmy Sparx Systems)

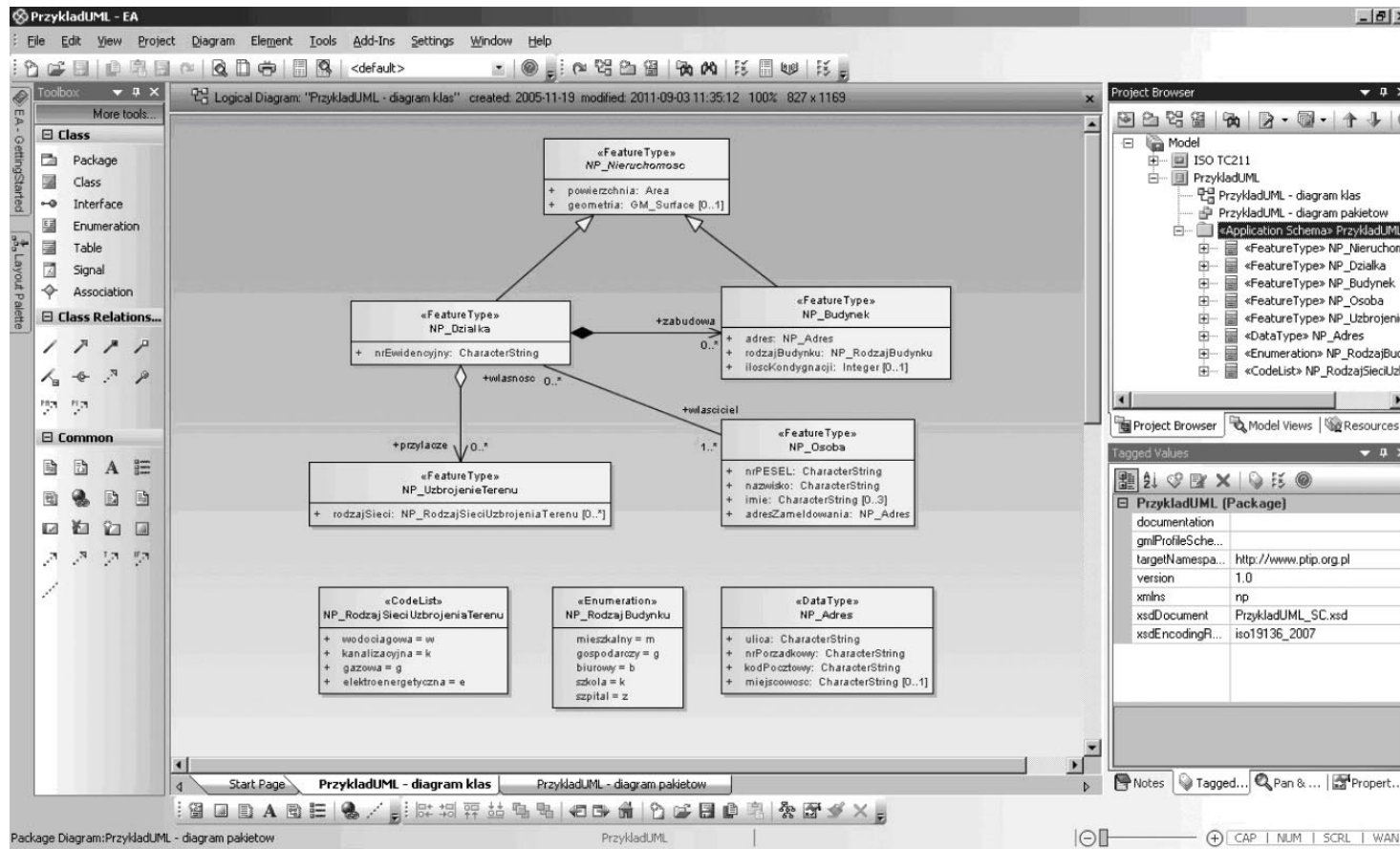
Zaawansowane narzędzie do modelowania systemów za pomocą UML (rys. 7.4). Charakteryzuje się:

- pełnym wsparciem dla specyfikacji UML 2.0 (wszystkie 13 rodzajów diagramów),
- intuicyjnym i ergonomicznym interfejsem użytkownika (duża liczba pasków narzędzi, dokowalnych okien i stylów wizualizacji),
- wsparciem dla technologii MDA,
- łatwością tworzenia dokumentacji.

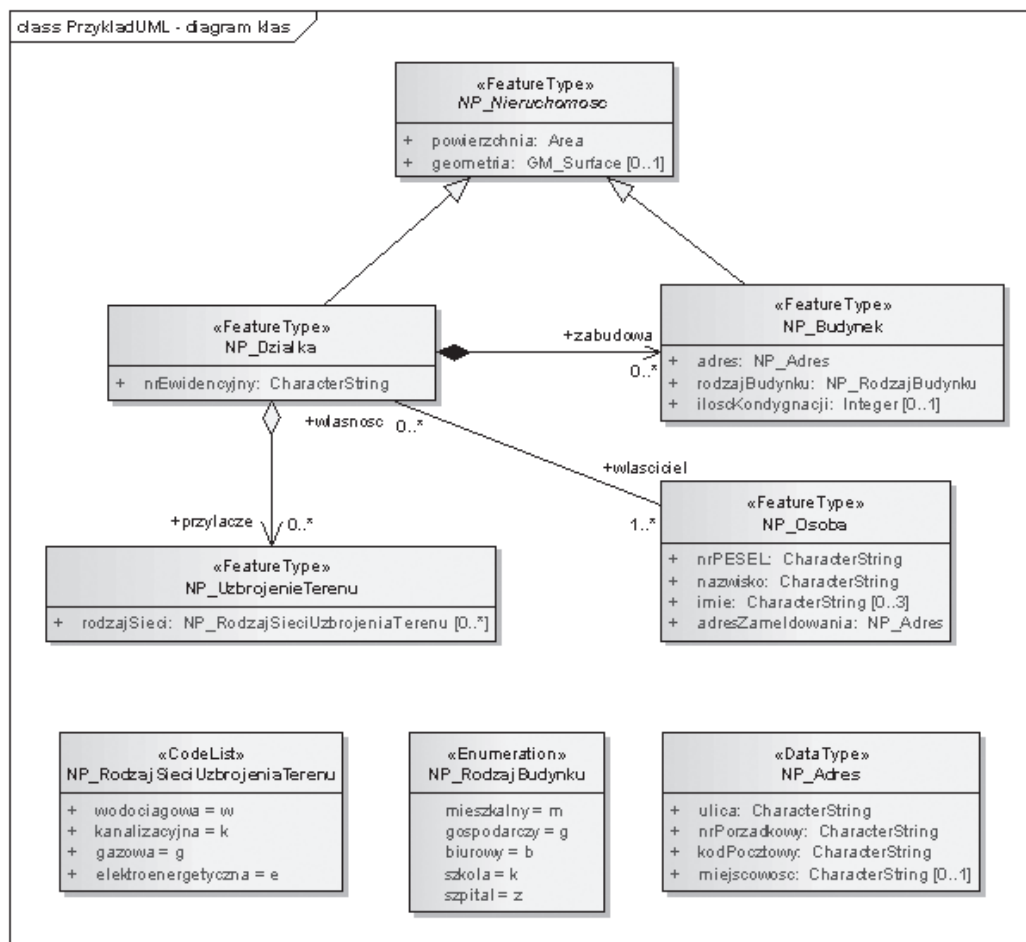
W metodzie automatycznej kluczowym etapem jest odpowiednie przygotowanie schematu aplikacyjnego UML w środowisku *Enterprise Architect*. Przede wszystkim należy skorzystać z odpowiedniego repozytorium – modeli ISO (<https://www.seegrid.csiro.au/mirrors/iso-harmonized-model/>), a w przypadku specyfikacji INSPIRE także z repozytorium INSPIRE (<https://inspire-twg.jrc.ec.europa.eu/svn/inspire-model/branches/public/>) oraz ze specjalnego profilu UML dla GML. Jest to plik XML (dostępny pod adresem https://inspire-twg.jrc.ec.europa.eu/svn/inspire-model/branches/public/uml_profile/), w którym każdemu elementowi UML przypisano właściwe metki ułatwiające proces transformacji UML-GML.

Opracowanie schematu aplikacyjnego UML w *Enterprise Architect* mogłoby przebiegać według następujących kroków:

- 1) utworzenie nowego projektu,
- 2) wczytanie całego repozytorium ISO/TC211 lub tylko wybranych norm ISO (np. 19103, 19107, 19109),
- 3) wczytanie pliku profilu UML (dostępny pod adresem https://inspire-twg.jrc.ec.europa.eu/svn/inspire-model/branches/public/uml_profile/),
- 4) nazwanie pakietu i diagramu,
- 5) dodanie diagramu pakietów,
- 6) przeciągnięcie z *UML Profiles* pakietu «application Schema» i przypisanie mu nazwy,
- 7) przeciągnięcie z *UML Profiles* potrzebnych klas («featureType», «dataType», itd.),
- 8) przypisanie nazw klasom,
- 9) dodanie atrybutów,
- 10) określenie liczności, widoczności i typu danych (ze specyfikacji ISO/TC 19103) dla atrybutów,
- 11) utworzenie związków między klasami,
- 12) opisanie związków rolami i krotnościami,
- 13) ustawienie nawigowalności związków,
- 14) ustawienie odpowiednich metek, zwłaszcza dla pakietu «application Schema»,
- 15) Zapisanie gotowego modelu (rys. 7.5).



Rys. 7.4. Środowisko pracy w *Enterprise Architect*

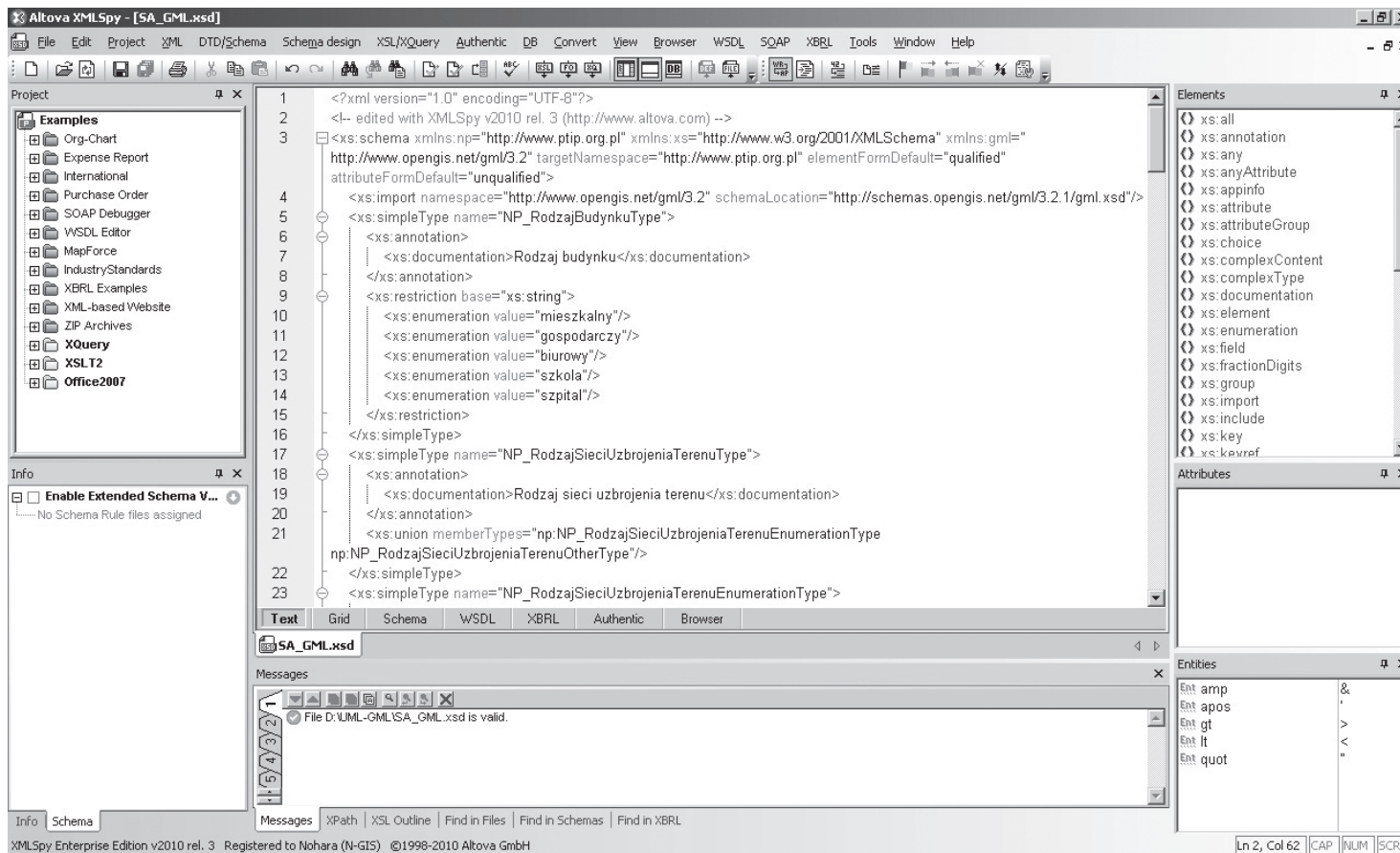


Rys. 7.5. Przykład schematu aplikacyjnego w postaci diagramu klas UML

Automatyczne wygenerowanie schematu aplikacyjnego GML ze schematu aplikacyjnego UML za pomocą aplikacji *ShapeChange* lub *FullMoon* mogłoby przebiegać następująco:

- 1) otwarcie wcześniej opracowanego schematu aplikacyjnego UML w *Enterprise Architect*,
- 2) eksport pakietu «application Schema» do formatu XMI (XML),
- 3) wprowadzenie odpowiednich ustawień dla pliku XML (różne dla obu aplikacji, m.in. rodzaj XMI – dla *ShapeChange* XMI 1.0, dla *FullMoon* XMI 1.1),
- 4) uruchomienie aplikacji *ShapeChange* lub z *FullMoon* wiersza poleceń.

Na rysunku 7.6 przedstawiono gotowy plik XSD dla GML.



Rys. 7.6. Przykładowe przekształcenie wcześniej utworzonego schematu aplikacyjnego UML na odpowiadający mu schemat aplikacyjny GML

7.4. Podsumowanie

Różnice w wygenerowanych schematach aplikacyjnych GML

Schematy aplikacyjne GML otrzymane przy użyciu różnych metod różnią się nieznacznie między sobą. Główne różnice widoczne są w sposobie kodowania elementów typów właściwości oraz związków między klasami.

Należy również zwrócić uwagę na to, iż w schematach aplikacyjnych utworzonych za pomocą narzędzia *FullMoon* brakuje mapowań typów danych pochodzących ze schematów znormalizowanych (np. `CharacterString`, `GM_Surface`). Aplikacja ma ustawione nieaktualne ścieżki do dokumentów zawierających odpowiednie tablice mapowań, a aktualizacja tych ścieżek powoduje niestety wadliwe funkcjonowanie oprogramowania.

Kodowanie typów właściwości

Zgodnie z normą ISO 19136, dla klas UML ze stereotypem «`dataType`» lub «`featureType`» w XML Schema należy utworzyć nazwane typy złożone, których nazwa zawiera nazwę danej klasy UML oraz przyrostek `PropertyType`. Aby w GML wyrazić związek „posiadania” między klasami, do takiego typu właściwości należy dodać grupę atrybutów `gml:OwnershipAttributeGroup`. Przez „posiadanie” należy tu rozmieść „posiadanie” jednej klasy przez drugą, to znaczy że jedna klasa UML posiada atrybut typu „inna klasa” UML. Natomiast, gdy zachodzi potrzeba odwzorowania zwykłego powiązania (asocjacji) między klasami UML, wtedy do typu właściwości należy dodać grupę atrybutów `gml:AssociationAttributeGroup`.

W schemacie aplikacyjnym GML wygenerowanym przez *ShapeChange* dla klasy ze stereotypem «`dataType`» do typu właściwości nie dodano żadnej grupy atrybutów, natomiast w schemacie otrzymanym z aplikacji *FullMoon* typ właściwości miał przypisaną grupę `gml:OwnershipAttributeGroup`. W przypadku klas ze stereotypem «`featureType`», niezależnie od zastosowanej metody transformacji UML do GML, każdy typ właściwości miał przypisane obie grupy atrybutów.

Kodowanie związków między klasami UML

Role w powiązaniach między klasami UML w GML są przekształcane na elementy lokalne z tą samą nazwą typu złożonego, który definiuje model zawartości typu obiektu. Jeśli powiązanie jest opisane z dwóch stron rolami, obie są przekształcane na GML i dodatkowo do ich opisu stosowany jest element `gml:reversePropertyName`, umieszczony wewnątrz elementów `annotation` i `appinfo`, który wskazuje nazwę roli na drugim końcu powiązania.

W przypadku powiązań z jedną nazwaną rolą, zarówno *ShapeChange*, jak i *FullMoon* poprawnie (zgodnie z normą ISO 19136) kodują takie elementy (pod warunkiem, że są to role klas „źródłowych”, tzn. klas od których rysowane jest powiązanie w *Enterprise Architect*). Natomiast w przypadku powiązań opisanych dwiema rolami, *ShapeChange* koduje tylko rolę klasy „docelowej” (tzn. klasy, do której rysowane jest powiązanie w *Enterprise Architect*) i bez elementu `gml:reversePropertyName`. Z kolei aplikacja *FullMoon* koduje obie role, również bez dodatkowego elementu `gml:reversePropertyName`, ale niestety z błędem. Asocjacja kodowana jest poprawnie od strony klasy „docelowej”, ale błędnie nazwana jest rola od strony klasy „źródłowej” (nazywa się tak samo jak rola klasy „docelowej”) i ma ona również błędnie przypisany typ danych.

Poniżej przedstawiono odpowiednie fragmenty kodu GML obrazujące powyższe różnice.

Przykład zapisu metodą ręczną

```

<xs:complexType name="NP_DzialkaType">
.....
  <xs:sequence>
    .....
    <xs:element name="wlasiciel" type="np:NP_OsobaPropertyType" maxOccurs="unbounded">
      <xs:annotation>
        <xs:appinfo>
          <gml:reverseProperty>np:wlasnosc</gml:reverseProperty>
        </xs:appinfo>
      </xs:annotation>
    </xs:element>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="NP_OsobaType">
.....
  <xs:sequence>
    .....
    <xs:element name="wlasnosc" type="np:NP_DzialkaPropertyType" minOccurs="0" maxOccurs="unb
      <xs:annotation>
        <xs:appinfo>
          <gml:reverseProperty>np:wlasiciel</gml:reverseProperty>
        </xs:appinfo>
      </xs:annotation>
    </xs:element>
  </xs:sequence>
</xs:complexType>

```

Przykład zapisu oprogramowaniem *ShapeChange*

```

<complexType name="NP_DzialkaType">
.....
  <sequence>
    .....
    <element name="wlasiciel" type="np:NP_OsobaPropertyType" maxOccurs="unbounded"/>
  </sequence>
</complexType>

```

Przykład zapisu oprogramowaniem *FullMoon*

```

<complexType name="NP_DzialkaType">
.....
  <sequence>
    .....
    <element name="wlasnosc" type="np:NP_DzialkaPropertyType" minOccurs="0" maxOccurs="unb
  </sequence>
</complexType>

<complexType name="NP_OsobaType">
.....
  <sequence>
    .....
    <element name="wlasnosc" type="np:NP_DzialkaPropertyType" minOccurs="0" maxOccurs="unb
  </sequence>
</complexType>

```

Problematyka niniejszej monografii stanowi przedmiot szerokiego zainteresowania środowisk współtworzących i współużytkujących infrastrukturę informacji przestrzennej budowaną w Polsce zgodnie z przepisami krajowymi i unijnymi. Zainteresowanie to znalazło swój wyraz w warsztatach „Modele danych przestrzennych w UML i ich transformacja do schematów GML i struktur baz danych”, które odbyły się w ramach konferencji Polskiego Towarzystwa Informatyki na temat „Informacja przestrzenna dla Polski i Europy”, Warszawa, 7–9 listopada 2011 roku. Odpowiadając na ujawnione wówczas zapotrzebowanie, zespół wykładowców podjął trud zawarcia zaprezentowanych przez siebie treści w opracowaniu o charakterze monograficznym. W rezultacie powstała publikacja, która przedstawia w sposób uporządkowany bogaty zasób wiadomości określonych tytułem warsztatów i dotyczących wybranych metod i technologii geoprzestrzennych.

Godne uznania jest, że zespół autorski w składzie: dr inż. A. Chojka, dr inż. A. Zwirowicz-Rutkowska, dr inż. Z. Parzyński i dr hab. J. Michalak, pełniący rolę redaktora naukowego, zrealizował podjęte przedsięwzięcie w stosunkowo krótkim terminie z niewątpliwą korzyścią dla potencjalnych Czytelników.

Jerzy Gaździcki

Warszawa, maj 2012 r.

Autorzy

dr hab. Janusz Michalak

Wydział Geologii, Uniwersytet Warszawski

J.Michalak@uw.edu.pl

Redakcja naukowa i rozdziały:

1. Wstęp
 2. Różnice pomiędzy językiem zapisu danych i jego dziedzinową aplikacją
 9. Najczęściej popełniane błędy w modelach UML dla schematów aplikacyjnych GML
 11. Schematy aplikacyjne tematów aneksów II i III Dyrektywy INSPIRE
 12. Podsumowanie
- Słownik podstawowych terminów stosowanych w tekście

dr inż. Agnieszka Chojka

Wydział Geodezji i Gospodarki Przestrzennej, Uniwersytet Warmińsko-Mazurski

agnieszka.chojka@uwm.edu.pl

Rozdziały:

3. Wprowadzenie do modelowania informacji przestrzennej – metodyka MDA i diagramy klas UML
6. Budowa schematu aplikacyjnego GML – reguły budowy, narzędzia i przykłady
7. Transformacja schematu aplikacyjnego UML do schematu aplikacyjnego GML – wymagania, ograniczenia i wybrane narzędzia

dr inż. Agnieszka Zwirowicz-Rutkowska

Wydział Geodezji i Gospodarki Przestrzennej, Uniwersytet Warmińsko-Mazurski

agnieszka.zwirowicz@uwm.edu.pl

Rozdziały:

4. Przegląd standardów i narzędzi wykorzystywanych do modelowania informacji geograficznej
5. Schematy aplikacyjne UML – reguły budowy i przykłady
10. Zastosowanie metodyki MDA – wybrane zagadnienia transformacji schematów aplikacyjnych UML do struktur relacyjnych baz danych

dr inż. Zenon Parzyński

Wydział Geodezji i Kartografii, Politechnika Warszawska

z.parzynski@gik.pw.edu.pl

8. Przykład zastosowania metod modelowania danych z zakresu Służby Geodezyjno-Kartograficznej

MODELE DANYCH PRZESTRZENNYCH W UML I ICH TRANSFORMACJA DO SCHEMATÓW GML I STRUKTUR BAZ DANYCH

Słowa kluczowe: geoinformacja, informacja geograficzna, model pojęciowy, UML, schemat aplikacyjny, GML, model relacyjny, transformacja

Streszczenie

Celem monografii jest przedstawienie czytelnikom podstawowych metodyk, technik i narzędzi przeznaczonych do budowy modeli pojęciowych danych przestrzennych na poziomie pojęciowym i implementacyjnym, a następnie do transformacji tych modeli do schematów XSD bazujących na języku GML i do zapisów struktur baz danych w języku DDL. Całość składa się z dwunastu rozdziałów dotyczących poszczególnych aspektów budowy modeli i ich transformacji. Wstęp wprowadza czytelników w całą przedstawianą problematykę i naświetla szerszy teoretyczny kontekst z zakresu modelowania i wykorzystania modeli w zastosowaniach praktycznych. Rozdział drugi poświęcony jest nowym metodom zapisu danych przestrzennych opartego na językach znacznikowych, a w szczególności na języku GML, objaśnia zasady takiego zapisu, zawiera krótką historię języka GML i przedstawia inne języki znacznikowe z nim powiązane. Rozdziały trzeci i czwarty stanowią wprowadzenie do modelowania informacji przestrzennej opartego o metodykę MDA z wykorzystaniem języka UML i zawierają przegląd standardów i narzędzi służących temu modelowaniu. W rozdziałach piątym i szóstym przedstawione są podstawowe zasady budowy tematycznych schematów aplikacyjnych w metodyce języka UML i języka GML zilustrowane przykładami. Rozdział siódmy poświęcony jest zagadnieniom transformacji schematów aplikacyjnych z UML do GML, a w szczególności wymaganiom i ograniczeniom, jakie muszą być spełnione, a także dostępnym narzędziom. Kolejny ósmy rozdział dotyczy modeli UML dedykowanych komponentowi infrastruktury krajowej, przeznaczonym dla Służby Geodezyjnej i Kartograficznej. W rozdziale dziewiątym dokonany jest przegląd najczęściej popełnianych błędów w budowie modeli UML przeznaczonych do utworzenia schematów bazujących na języku GML. Tematem rozdziału dziesiątego jest zastosowanie metodyki MDA do transformacji modeli UML do struktur relacyjnych baz danych. Rozdział jedenasty zawiera metodyczną analizę różnych przypadków występujących w modelach danych tematów aneksów II i III dyrektywy INSPIRE, w tym porównanie z modelami tematów aneksu I, analizę różnych typów i form danych, jakie tam występują. Dwunasty rozdział to podsumowanie, w którym zwraca się szczególną uwagę na dynamiczny rozwój metod z tego zakresu, zmiany zachodzące w zakresie stosowanej terminologii i skutki, jakie te zmiany za sobą pociągają.

UML GEOSPATIAL DATA MODELS AND THEIR TRANSFORMATION INTO GML SCHEMAS AND DATABASE STRUCTURES

Keywords: geoinformation, geographic information, conceptual model, UML, application schema, GML, relational model, transformation

Abstract

The main objective of the monograph is to present essential methodologies, technologies and software tools dedicated to building conceptual models of geospatial data on conceptual level, and implementation level, and then to be transformed into XSD schemas based on GML language and to encode data bases structures in DDL language. The whole monograph consists of twelve chapters concerning different aspects of models development and their transformation. The introduction familiarizes readers with all issues presented and clarifies broader theoretical context with regard to modeling and exploitation of models in practical applications. The second chapter is dedicated to modern methods of encoding spatial data based on markup languages, in particular on GML language; rules for that encoding are also explained. This chapter contains a short history of GML language and presents other markup languages associated with it. Chapters three and four provide an introduction to spatial information modeling based on MDA methodology with application of UML language and it contains a review of standards and tools dedicated to such modeling. In chapters five and six, essential rules of development of thematic application schemas are presented in the methodology of UML and GML languages. Examples to illustrate them are provided. Chapter seven is dedicated to issues of transformation application schemas from UML to GML, in particular to the requirements and constraints that must be fulfilled and also to available tools. The next chapter eight concerns UML models dedicated to components of the national infrastructure designated for Geodetic and Cartographic Service. In chapter nine, a review of most frequent mistakes committed in drawing up UML models dedicated to generating of schemas based on GML language are presented. The subject of chapter ten is the application of MDA methodology for transformation of UML models into relational databases structures. Chapter eleven contains methodological analysis of various cases occurring in data models for the themes defined in Annex II and III of INSPIRE Directive as well as a comparison with the models for themes defined in Annex I and an analysis of various data forms occurring there. In chapter twelve, the recapitulation is presented, in which dynamic development of methods in this area is taken in consideration. In addition, significant changes in the terminology and the effects of these changes are discussed.

Spis treści

1. Wstęp	11
2. Różnice pomiędzy językiem zapisu danych i jego dziedzinową aplikacją	15
2.1. Podstawy zapisu znacznikowego na bazie języka XML	15
2.2. Wprowadzenie do języka GML	18
2.3. Krótka historia zapisu geoinformacji	24
2.4. Języki oparte na GML i z nim powiązane	25
2.5. Przyszłość języka GML	27
2.6. Modele UML dedykowane zapisom w języku GML	30
3. Wprowadzenie do modelowania informacji przestrzennej – metodyka MDA i diagramy klas UML	35
3.1 Wprowadzenie	35
3.2. Reguły budowy schematów aplikacyjnych w UML	39
4. Przegląd standardów i narzędzi stosowanych do modelowania informacji geograficznej	43
4.1. Model dziedzinowy informacji geograficznej	44
4.2. Funkcjonalność narzędzi do modelowania pojęciowego	45
5. Schematy aplikacyjne UML – reguły budowy i przykłady	49
5.1. Pojęcie schematu aplikacyjnego, jego rola i znaczenie	49
5.2. Proces budowy schematów aplikacyjnych	50
5.3. Przykłady schematów aplikacyjnych UML	52
6. Budowa schematu aplikacyjnego GML – reguły budowy, narzędzia i przykłady	55
6.1. Reguły budowy schematów aplikacyjnych GML	55
6.2. Przykład przekształcenia schematu aplikacyjnego UML na GML	66

7. Transformacja schematu aplikacyjnego UML do schematu aplikacyjnego GML – wymagania, ograniczenia i wybrane narzędzia	69
7.1. Metody transformacji UML do GML	69
7.2. Metoda ręczna	70
7.3. Metoda automatyczna	71
7.4. Podsumowanie	76
8. Przykład zastosowania metod modelowania danych z zakresu Służby Geodezyjno-Kartograficznej	79
8.1. Założenia przyjęte w GUGiK przy opracowywaniu projektów rozporządzeń	79
8.2. Realizacja założeń	80
8.3. Przykłady schematów aplikacyjnych do projektów rozporządzeń	83
9. Najczęściej popełniane błędy w modelach UML dla schematów aplikacyjnych GML	87
9.1. UML jest cierpliwy jak papier	87
9.2. Wymagania dotyczące modeli UML dla INSPIRE	90
10. Zastosowanie metodyki MDA – wybrane zagadnienia transformacji schematów aplikacyjnych UML do struktur relacyjnych baz danych	95
10.1. Transformacja w ujęciu metodyki MDA	95
10.2. Ogólne zasady mapowania pomiędzy modelem obiektowym i modelem relacyjnym	97
10.3. Transformacja schematu aplikacyjnego UML do logicznej struktury relacyjnej bazy danych	101
11. Schematy aplikacyjne tematów aneksów II i III dyrektywy INSPIRE	107
11.1. Nietypowy przypadek – temat Geologia	115
12. Podsumowanie	121
Słownik podstawowych terminów stosowanych w tekście	125
Literatura	131

Janusz Michalak

Słownik podstawowych terminów stosowanych w tekście

Abstrakcyjny – obiekt, atrybut, typ, klasa (*abstract – object, attribute, type, class*) – **1:** Określony ogólnie, bez szczegółów związanych z określoną implementacją (uwarunkowaniami technologicznymi) lub z określoną aplikacją (uwarunkowaniami wynikającymi z dziedziny zastosowania). Na przykład wynik pomiaru w znaczeniu ogólnym jako atrybut w modelu pojęciowym nie musi mieć określonego typu. Typ będzie zależał od fizycznego charakteru mierzonego elementu i od typu przyrządu pomiarowego. **2:** Klasa abstrakcyjna w modelu danych to klasa, która nie ma własnych obiektów, a jedynie służy jako klasa bazowa dla innych klas. Użycie takiej klasy jest uzasadnione tylko gdy są (lub mogą być) wyprowadzone z nie przynajmniej dwie klasy.

Atrybut (*attribute*) – Właściwość wyróżnienia lub obiektu określona przez nazwę tej właściwości i zakres wartości, jakie mogą być przypisane tej nazwie dla określenia tej właściwości.

Atrybut geoprzestrzenny (*geospatial attribute*) – Właściwość (cecha) wynikająca z faktu, że wyróżnienie zajmuje pewne miejsca w rzeczywistości w sensie geoprzestrzennym. Najczęściej przez domniemanie przyjmuje się, że określenie geoprzestrzenny obejmuje również czas, czyli jest równoznaczne z określeniem czaso-geoprzestrzenny. Przykładami takich atrybutów są: wielkość, kształt, położenie, przynależność geoprzestrzenna (np. leży w obrębie), relacje geoprzestrzenne względem innych wyróżnień (np. odległość lub rodzaj sąsiedztwa).

Atrybut niegeoprzestrzenny (*non-geospatial attribute*) – Wszystkie pozostałe atrybuty niezwiązane z odniesieniem przestrzennym. Atrybuty te mogą należeć zarówno do wyróżnień geoprzestrzenne jak i do innych obiektów i wystąpień niegeoprzestrzennych.

Cecha (*trait*) – Kategoria klasy, której zadaniem jest (w przypadku modeli danych) dostarczenie innej klasie określonych własności (atrybutów i powiązań z innymi klasami). W tym przypadku klasa ma stereotyp «*trait*». Podobnym mechanizmem pozwalającym na uniknięcie problemów wielokrotnego dziedziczenia jest **domieszka**.

Dane (*data*, w liczbie pojedynczej: *datum*) – **1:** Jednostki informacji, czyli pojedyncze fragmenty informacji. Dane niezorganizowane nie stanowią informacji i często są bezużyteczne. Dane zorganizowane stanowią elementy informacji. Zorganizowanie danych może być jawne, na przykład w językach znacznikowych lub niejawne, na przykład miejsce umieszczenia adresu na kopercie decyduje, czy jest to adres nadawcy czy odbiorcy. **2:** Fakty, statystyki, opinie i przewidywania zebrane z różnych wewnętrznych i zewnętrznych źródeł. Dane bez kontekstu są szumem (Nowicki i Staniszkis, 2002).

Dane geoprzestrzenne (*geospatial data*) – **1:** Dane w sensie zdefiniowanym przez informatykę, ale w odróżnieniu od innych rodzajów danych są one odniesione do określonego miejsca (fragmentu przestrzeni) i w rezultacie niezbędnymi ich składnikami są dane określające położenie tego miejsca względem Ziemi. **2:** Dane przestrzenne dotyczące Ziemi i wszystkich obiektów przestrzennych z nią związanych (Gaździcki, 2004).

Domieszka (*mixin*) – **1:** Kategoria klasy, której zadaniem jest (w przypadku modeli danych) dostarczenie innej klasie określonych własności (atrybutów i powiązań z innymi klasami). Taka klasa nie ma własnych obiektów, czyli musi być abstrakcyjna. Stosowanie tego rodzaju klasy jest uzasadnione tylko w przypadkach, gdy przynajmniej dwie zwykłe klasy otrzymują w ten sposób własności. Jest to sposób na uniknięcie problemów z implementacją wielokrotnego dziedziczenia. Jedyny przypadek zastosowania klasy *mixin* do języka GML to modele dla niektórych tematów INSPIRE. **2:** Ograniczony sposób dziedziczenia ma pozwalający również na uniknięcie problemów z implementacją wielokrotnego dziedziczenia. W takim przypadku powiązanie dziedziczenia ma stereotyp «*mixin*». Porównaj: **cecha**.

Encja (*entity*) – Pojęcie z modelu encja-związek, oznaczające konkretny lub abstrakcyjny byt wyróżnialny w modelowanej rzeczywistości. W odróżnieniu od obiektu, encja nie jest kojarzona z metodami (Subieta, 1999a).

GML (*Geography Markup Language*) – Język znaczników geograficznych, aplikacja języka (metajęzyka) XML przeznaczona do zapisu geoinformacji w celu przesyłania jej pomiędzy różnymi systemami – on-line, niezależnie od platformy sprzętowo-systemowej i niezależnie od charakteru i technologii systemu geoinformacyjnego (Gaździcki, 2004).

Informacja (*information*) – **1:** Dane komputerowe, które są zorganizowane i przedstawione w usystematyzowanej formie dla zrozumiałości ich podstawowego znaczenia. Związki pomiędzy informacją i danymi wyjaśnia definicja danych. **2:** Dane interpretowane w kontekście określonego celu (Nowicki i Staniszkis, 2002). **3:** Wiedza uzyskiwana w drodze interpretacji danych, która w ustalonym kontekście ma określone znaczenie i dotyczy obiektów, takich jak fakty, zdarzenia, przedmioty, zjawiska, procesy i idee (Gaździcki, 2004).

Informacja geograficzna – patrz: **informacja geoprzestrzenna**.

Informacja geoprzestrzenna (*geospatial information*) – **1:** Informacja w sensie zdefiniowanym przez informatykę, ale w odróżnieniu od innych rodzajów informacji jest ona odniesiona do określonego miejsca (fragmentu przestrzeni) i w rezultacie niezbędnymi jej składnikami są dane określające położenie tego miejsca względem Ziemi. **2:** Informacja uzyskiwana w drodze interpretacji danych geoprzestrzennych (Gaździcki, 2004).

Instancja (*instance*) – Synonim egzemplarza stosowany w normach PN-EN ISO 19100 (Gaździcki, 2011).

Klasa (*class*) – Pojęcie klasy jest używane w trzech dość bliskich znaczeniach: **(1)** zbiór obiektów o zbliżonych własnościach; **(2)** byt semantyczny rozumiany, jako miejsce przechowywania takich cech grupy podobnych obiektów, które są dla nich niezmiennie (np. zestawu atrybutów, nazwy, metod, ograniczeń dostępu); **(3)** wyrażenie językowe specyfikujące budowę obiektów, dozwolone operacje na obiektach, ograniczenia dostępu, wyjątki, itd. Zwykle klasy wiąże się ze sobą poprzez hierarchię (lub inną strukturę) dziedziczenia (Subieta, 1999a).

MDA – 1: (*Model Driven Approach*) Podejście oparte na modelu: pojęciowym, logicznym i fizycznym. Niezależny od implementacji schemat aplikacyjny zostaje odwzorowany na różne specyfikacje (wykorzystujące różne technologie, np. usługi sieciowe, relacyjne bazy danych, XML), a te z kolei mogą zostać zaimplementowane (wdrożone) na różnych platformach sprzętowo-programowych (CEN, 2006). **2:** (*Model Driven Architecture*) Zbiór metod porządkujących proces tworzenia systemów informatycznych opartych na budowie modeli i ich transformacji. Koncepcja MDA została opracowana przez międzynarodową organizację OMG, której celem jest rozwiązywanie problemów związanych z integracją systemów informatycznych pochodzących od różnych dostawców oraz działających na różnych platformach informatycznych (OMG, 2003).

Metamodel (*metamodel*) – W założeniu, model definiujący składnię, semantykę i pragmatykę wprowadzonego modelu, notacji lub diagramu. Metamodel proponowany przez autorów UML ustala pewne elementy składni diagramów, ograniczenia typologiczne, klasyfikację pojęć oraz związki pomiędzy pojęciami (Subieta, 1999a).

Metka (*tagged value*) – Inaczej wartość etykietowana. Obok stereotypów i ograniczeń, to jeden z mechanizmów rozszerzenia semantyki języka UML. Pozwala dołączyć do elementu modelu UML dodatkowe właściwości. Metka to para *klucz=wartość*.

Metodyka (*methodology*) – Zestaw pojęć, notacji, modeli formalnych, języków i sposobów postępowania służący do analizy rzeczywistości (stanowiącej przedmiot projektowanego systemu informatycznego) oraz do projektowania pojęciowego, logicznego i/lub fizycznego. Zwykle metodyka jest powiązana z odpowiednią notacją (diagramami) służącymi do zapisywania wyniku poszczególnych faz projektu, jako środek wspomagający ludzką pamięć i wyobraźnię i jako środek komunikacji w zespołach oraz pomiędzy projektantami i klientem (Subieta, 1999a).

Model pojęciowy (*conceptual model*) – Model procesów lub model struktury danych odwołujący się do ludzkiej percepcji i wyobraźni, mający za zadanie zrozumienie problemu, udokumentowanie wyniku analizy lub projektu w czytelnej i abstrakcyjnej formie językowej oraz ułatwienie komunikacji w zespołach ludzkich (Subieta, 1999a).

Model semantyczny (*semantic model*) – Zestaw pojęć, technik i notacji mający na celu odwzorowanie semantyki danych, czyli ich znaczenia w świecie zewnętrznym. Modele semantyczne wprowadzają w tym celu pojęcia, takie jak: generalizacja, specjalizacja, asocjacja, agregacja, klasyfikacja, własności temporalne, zdarzenia, własności behawioralne, itd. Przykładem prostego modelu semantycznego jest model encja-związek. Niekiedy terminem “model semantyczny” określa się również konkretny diagram (lub inną formę językowo-graficzną) odwzorowującą rzeczywistość opisywaną przez dane (Subieta, 1999a).

Norma (*standard*) – **1:** Dokument przyjęty na zasadzie konsensu i zatwierdzony przez upoważnioną jednostkę organizacyjną, ustalający – do powszechnego i wielokrotnego stosowania – zasady, wytyczne lub charakterystyki odnoszące się do różnych rodzajów działalności lub ich wyników i zmierzający do uzyskania optymalnego stopnia uporządkowania w określonym zakresie (Ustawa, 2002). **2:** Polska Norma – jest normą o zasięgu krajowym, przyjętą w drodze konsensu i zatwierdzoną przez krajową jednostkę normalizacyjną (Polski Komitet Normalizacyjny), powszechnie dostępną, oznaczoną – na zasadzie wyłączności – symbolem PN (PKN, 2010). Zobacz: **normy ISO serii 19100, standard, standardy OGC.**

Normy ISO serii 19100 (*ISO 19100 series of International Standards*) – Rodzina norm ISO w dziedzinie informacji geograficznej. Wynik prac Komitetu Technicznego ISO/TC211, który pracuje nad wieloma projektami standaryzacji informacji przestrzennej w bardzo szerokim zakresie tej problematyki. Zobacz: **norma, standard, standardy OGC**.

Obiekt (*object*) – **1:** W teorii informacji – konkretny lub abstrakcyjny byt (wystąpienie) wyróżnialny w modelowanej rzeczywistości, posiadający nazwę, jednoznaczną identyfikację, wyraźnie określone granice, atrybuty i inne właściwości takie jak rodzaj struktury wewnętrznej lub struktury danych z nim związanych. Te składniki obiektu określają: jego stan (poprzez wartości atrybutów i powiązania) i jego zachowanie się (poprzez operatory i funkcje, czyli metody) (Subieta, 1999a). **2:** W geomatyce przyjmuje się, że obiekt jest wystąpieniem klasy i jest to oparte na paradygmacie obiektowości wywodzącym się z języka UML, który jest przyjęty do opisu modeli pojęciowych (OMG, 2001). **3:** Termin stosowany w różnych znaczeniach; dla uniknięcia wątpliwości, jeśli jego znaczenie nie wynika z kontekstu, powinien być uzupełniony dodatkowym określeniem (Gaździcki, 2004).

Rola (*role*) – W języku UML jedna z możliwości opisu powiązania. Pozostałe to nazwa powiązania oraz krotność. Każda klasa biorąca udział w powiązaniu ogrywa w nim określoną rolę. Inaczej jest to „oblicze”, które klasa przy jednym końcu powiązania prezentuje klasie przy drugim jego końcu.

Schemat (*schema*) – **1:** Opis logicznej struktury bazy danych lub innego systemu związanego z danymi, np. interfejsu wymiany danych (XML Schema). **2:** Opis atrybutów wyróżnień (*feature*), lub bardziej dokładnie – specyficzny model atrybutów dla wyróżnień określony za pomocą elementarnych typów danych i ograniczeń dotyczących tych typów (Buehler, McKee, 1996).

Schemat aplikacyjny (*application schema*) – Schemat przeznaczony dla konkretnego systemu lub dla konkretnej dziedziny zastosowań.

Schemat implementacyjny (*implementation schema*) – Schemat uwzględniający technologiczne środowisko, w którym będzie realizowana jego aplikacja. Na przykład zapisany w formie schematu XML.

Specyfikacja (*specification*) – **1:** Abstrakcyjny opis bytu programistycznego (procedury, modułu, klasy, obiektu, bazy danych, itp.) określający reguły użycia lub ustalający podstawowe założenia jego implementacji (Subieta, 1999a). **2:** Dokument lub opis, który określa w sposób kompletny, precyzyjny i sprawdzalny wymagania, projekt lub charakterystykę systemu lub jego fragmentu, a często także procedury dla określenia czy te wymagania są spełnione.

Standard (*standard*) – Wzorzec rozwiązania sprzętowego lub programowego zatwierdzony przez instytucję normalizacyjną lub przyjęty nieformalnie wskutek dużego upowszechnienia, w przypadku standardów informatycznych najczęściej o zasięgu światowym. Do najważniejszych instytucji opracowujących standardy należą: ISO, IEEE, ANSI. Przykładami standardów są: RS-232-C (fabryczny standard interfejsu sprzętowego), ANSI C++ (oficjalny standard języka programowania), POSIX (standard IEEE przenośnego systemu uniksowego), CORBA (standard obiektowych systemów rozproszonych) (Płoski, 1999). Zobacz: **standardy OGC, norma, normy ISO serii 19100**.

Standardy OGC (*OGC standards*) – Techniczne dokumenty specyfikujące interfejsy i reguły zapisu danych geoprzestrzennych. Stanowią one główne rezultaty działalności OGC (*Open Geospatial Consortium*) i są opracowywane przez zespoły złożone z członków OGC dla rozwiązywania różnorodnych problemów dotyczących interoperacyjności. Wszystkie publiczne dokumenty OGC są łatwo dostępne bez żadnych opłat. OGC ma ponad 400 członków, w tym ponad połowa to wyższe uczelnie i instytucje naukowe, także prawie połowę stanowią członkowie europejscy. Standardy OGC dzielą się na specyfikacje abstrakcyjne i standardy implementacyjne. Wiele z tych standardów zostało przyjęte przez komitet ISO/TC 211 jako normy ISO, na przykład: 19107, 19115, 19119, 19123, 19125, 19128, 19136, 19139, 19142, 19143, i 19156. Ze standardami OGC powiązane są inne oficjalne dokumenty OGC, na przykład: *OGC Reference Model (ORM)*, *Engineering Reports* lub nieoficjalne, na przykład *Best Practices Documents* i *Discussion Papers*. Zobacz: **standard, norma, normy ISO serii 19100**.

Stereotyp (*stereotype*) – W terminologii UML, klasyfikacja elementu modelu posiadająca semantyczne konsekwencje. Stereotypy mogą być predefiniowane lub zdefiniowane przez użytkownika (Subieta, 1999a).

Struktura (*structure*) – Termin w C++ (także w innych językach) na oznaczenie zestawu nazwanych wartości, w innych językach odpowiada jej zapis lub rekord (Subieta, 1999a).

Tabela (*table*) – Struktura danych implementowana w relacyjnych bazach danych, często nazywana relacją. Tabela składa się z wierszy lub inaczej krotek. Należy zwrócić uwagę, że pomiędzy relacją (w sensie matematycznym) i tabelą występują dość istotne różnice koncepcyjne. Tabela jest wyposażona w nazwy kolumn (które niosą informację semantyczną) (Subieta, 1999a).

Tożsamość (*identity*) – Tożsamość obiektu oznacza, że obiekt istnieje i jest odróżnialny niezależnie od jego aktualnego stanu (wartości atrybutów), który może się zmieniać; możliwe są dwa różne obiekty o identycznych wartościach atrybutów. Praktycznie, tożsamość oznacza istnienie unikalnego wewnętrznego (nieczytelnego dla użytkownika) identyfikatora obiektu, który nie ulega zmianie podczas życia obiektu (Subieta, 1999a).

Unia (*union*) – Typ struktury, rekordu lub obiektu, który może mieć alternatywnie dwa lub więcej zestawów atrybutów. Przykładowo, jeżeli właścicielem samochodu może być osoba lub firma, to obiekt *Samochód* może posiadać alternatywnie albo atrybut *Nazwisko Właściciela* albo atrybut *WłasnośćFirmy*. Unia może mieć związany dyskryminator (*discriminator*), tj. atrybut, którego wartość określa, z którym wariantem mamy do czynienia. Może też nie mieć dyskryminatora; wówczas odpowiedzialność za rozróżnianie wariantów spada na programistę (tak jest np. w C i C++). Brak dyskryminatora w unii podkopuje koncepcję mocnej kontroli typów i stwarza okazję do bardzo trudnych błędów (Subieta, 1999a).

Walidator (*validator*) – Program komputerowy sprawdzający poprawność dokumentu (np. XML) o określonej składni.

Wyróżnienie geoprzestrzenne (*geospatial feature*) (w literaturze polskiej termin *feature* jest często tłumaczony jako obiekt) – **1**: Podstawowy fragment (atom) informacji geoprzestrzennej. Posiada atrybuty geoprzestrzenne (geometryczne i topologiczne) np. kształt, rozciągłość, położenie, relacje z innymi wyróżnieniami. Często pojęcie wyróżnienie jest my-

lone z pojęciem obiekt, jednak wyróżnienie może być obiektem, ale też może nim nie być (Mark i in., 2001). Ponieważ w geomatyce wszystkie wyróżnienia są geoprzestrzenne, przymiotnik geoprzestrzenny jest na ogół pomijany i używa się krótszego terminu wyróżnienie.

2: Cyfrowa reprezentacja zjawiska (bytu) świata rzeczywistego lub jego abstrakcja w modelu pojęciowym. Wyróżnienie ma określone miejsce w przestrzeni i czasie jako jego atrybuty (Buehler, McKee, 1996). Przykładem wyróżnienia może być prawie wszystko co może być umieszczone w przestrzeni i czasie: stół, budynek, miasto, drzewo, fragment lasu, ekosystem, trasa przejazdu lub wyż atmosferyczny jako obszar wysokiego ciśnienia powietrza.

3: Abstrakcja zjawiska świata rzeczywistego. Termin wyróżnienie może odnosić się do typu zjawiska lub jego konkretnego wystąpienia (ISO/TC 211, 2002a), np. „rzeka” i „Wisła”.

Związek (*relationship*) – **1:** W języku UML i w konsekwencji także w normach grupy ISO 19100 – semantyczne połączenie pomiędzy elementami modelu. Przykładami związków są agregacje, kompozycje (agregacje całkowite), powiązania i uogólnienia. **2:** W modelu encji-relacji – powiązanie pomiędzy encjami (Michalak, 2005a).

Literatura

- AB ORMSC (Architecture Board ORMS), 2001: Model Driven architecture (MDA). Document number ormsc/2001-07-01.
URL: [http:// http://www.enterprise-architecture.info/Images/MDA/MDA%20Technical.pdf](http://http://www.enterprise-architecture.info/Images/MDA/MDA%20Technical.pdf)
- Altova, XMLSpy. URL: <http://www.altova.com/xml-editor>
- BGWM (Biuro Geodety Województwa Mazowieckiego), 2009: Opis koncepcji identyfikatorów, wersjonowania zmian, stosowania reguły *nil reason*.
URL: <http://www.geointegracja.gov.pl/download/file.php?id=80&sid=f2c1f79e942a2cf12ed12a99aee5eec0>
- Biron P. V., Permanente K., Malhotra A. (W3C), 2004: XML Schema Part 2: Datatypes. Second Edition. W3C Recommendation 28 October 2004. URL: <http://www.w3.org/TR/xmlschema-2>
- Boisvert E., Brodaric B., 2008: GroundWater Markup Language Specification v. 1.0.
URL: http://ngwd-bdnes.cits.nrcan.gc.ca/service/api_ngwds/en/gwml.html
- Booch G., Rumbaugh J., Jacobson I., 2002: UML – przewodnik użytkownika. Z serii: Inżynieria oprogramowania. Wydanie polskie. Wydawnictwa Naukowo-Techniczne, Warszawa.
- Brink L., Portele C., Vretanos P. A. (OGC), 2011: Geography Markup Language (GML) simple features profile (with Corrigendum). OpenGIS Implementation Standard Profile.
URL: http://portal.opengeospatial.org/files/?artifact_id=42729
- Buechler K., McKee L. (ed.), 1996: The OpenGIS Guide – Introduction to Interoperable Geoprocessing – Part I of the Open Geodata Interoperability Specification (OGIS). OGIS TC Document 96-001, Open GIS Consortium, Wayland.
- Burggraf D., 2011: Input to the GML 4 workshop.
URL: http://external.opengeospatial.org/twiki_public/GML/Gml4WorkshopInput
- CEN, 2006: prCEN/TR 15449, Geographic information – Standards, specifications, technical reports and guidelines, required to implement Spatial Data Infrastructure.
- Chojka A., 2006: Przegląd metod, środków formalnych i narzędzi programowych wspomagających modelowanie pojęciowe informacji geograficznej. Część I – Modelowanie pojęciowe. *Magazyn Geoinformacyjny Geodeta*, nr 5 (132).
- Cox S. (ed.) (OGC), 2010: Geographic Information: Observations and Measurements – OGC Abstract Specification Topic 20. URL: http://portal.opengeospatial.org/files/?artifact_id=41579
- Cox S., 2011: Hollow World: a GML application schema template. Solid Earth and Environment GRID (SEE GRID community website). URL: <https://www.seegrid.csiro.au/wiki/AppSchemas/HollowWorld>
- CTWG-O&M (INSPIRE Cross Thematic Working Group on Observations & Measurements), 2011: D2.9 Guidelines for the use of Observations & Measurements and Sensor Web Enablement-related standards in INSPIRE Annex II and III data specification development, Version 1.0.
URL: http://inspire.jrc.ec.europa.eu/documents/Data_Specifications/D2.9_O&M_Guidelines_V1.0.pdf
- DT_DS (INSPIRE Drafting Team „Data Specifications”), 2008: D2.6: Methodology for the development of data specifications, Version 3.0.
URL: http://inspire.jrc.ec.europa.eu/reports/ImplementingRules/DataSpecifications/D2.6_v3.0.pdf
- DT_DS (INSPIRE Drafting Team „Data Specifications”), 2010a: D2.5: Generic Conceptual Model, Version 3.3. URL: http://inspire.jrc.ec.europa.eu/documents/Data_Specifications/D2.5_v3_3.pdf

- DT_DS (INSPIRE Drafting Team „Data Specifications”), 2010b: D2.7: Guidelines for the encoding of spatial data, Version 3.2. URL: http://inspire.jrc.ec.europa.eu/documents/Data_Specifications/D2.7_v3.2.pdf
- EC (European Commission), 2009: Guidance Document No. 22 – Updated Guidance on Implementing the Geographical Information System (GIS) Elements of the EU Water policy. Common Implementation Strategy for the Water Framework Directive (2000/60/EC). Technical Report – 2009 – 028.
URL: http://circa.europa.eu/Public/irc/env/wfd/library?l=/framework_directive/guidance_documents/guidance-no22-_nov08pdf_1/_EN_1.0_&a=d
- EP&CEU (European Parliament and Council of the European Union), 2007: Directive 2007/2/EC of the European Parliament and of the Council of 14 March 2007 establishing an Infrastructure for Spatial Information in the European Community (INSPIRE).
- Fowler M., Scott K., 2002: UML w kropelce. Oficyna wydawnicza LTP, Warszawa.
- Gaździcki J., 2004: Leksykon geomatyczny – Lexicon of geomatics. Polskie Towarzystwo Informatyki Przemysłowej, Warszawa.
- Gaździcki J., 2011: [W:] (red.) Gaździcki J. Internetowy leksykon geomatyczny.
URL: <http://www.ptip.org.pl/auto.php?page=Encyclopedia&enc=1>
- Githaiga J., 2010: Project Overview – FullMoon. Solid Earth and Environment GRID (SEE GRID community website). URL: <https://www.seegrid.csiro.au/wiki/Siss/FullMoon>
- Huang C-H., Chuang T-R., Deng D-P., Lee H-M., 2009: Building GML-native web-based geographic information systems. *Computers&Geosciences*, no 35, 1802-1816.
URL: <http://www.iis.sinica.edu.tw/papers/trc/8843-F.pdf>
- IGW-CGI-IUGS (Commission for the Management and Application of Geoscience Information – CGI, Interoperability Working Group – IWG, International Union of Geological Sciences – IUGS), 2008: GeoSciML Cookbook – How To Map Data to GeoSciML, Version 2.
URL: http://www.geosci.ml.org/geosci.ml/2.0/cookbook/GeoSciML_Data_CookBook_V2.pdf
- ISO/TC 211 (Geographic Information/Geomatics), 2002a: ISO 19101: Geographic information – Reference model. URL: http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=26002
- ISO/TC 211 (Geographic Information/Geomatics), 2002b: ISO 19108:2002 Geographic information – Temporal schema.
URL: http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=26013
- ISO/TC 211 (Geographic Information/Geomatics), 2003: ISO 19107:2003 – Geographic information – Spatial schema. URL: http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=26012
- ISO/TC 211 (Geographic Information/Geomatics), 2005a: ISO 19103 Technical Specification, Geographic information – Conceptual schema language.
URL: http://www.iso.org/iso/catalogue_detail.htm?csnumber=37800
- ISO/TC 211 (Geographic Information/Geomatics), 2005b: ISO 19109:2005 Geographic information – Rules for application schema.
URL: http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=39891
- ISO/TC 211 (Geographic Information/Geomatics), 2006: ISO 19110 – Geographic information – Methodology for feature cataloguing.
URL: http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=39965
- ISO/TC 211 (Geographic Information/Geomatics), 2007a: ISO 19136:2007 – Geographic information – Geography Markup Language (GML).
URL: http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=32554
- ISO/TC 211 (Geographic Information/Geomatics), 2007b: ISO 19139 Technical Specification, Geographic information – Metadata – XML schema implementation.
URL: http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=32557
- ISO/TC 211 (Geographic Information/Geomatics), 2009: ISO 19104:2009 Technical Specification, Geographic information – Terminology
URL: http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=32557
- ISO/TC 211 (Geographic Information/Geomatics), 2011: ISO 19118, Geographic information – Encoding.
URL: http://www.iso.org/iso/iso_catalogue/catalogue_ics/catalogue_detail_ics.htm?csnumber=44212
- ISO/TC 211 (Geographic Information/Geomatics), 2012: Introduction: Welcome to the ISO/TC 211 Harmonized Model Web server. URL: <http://www.isotc211.org/hmmg/HTML/root.html>

- Lake R., Burggraf D., Trninc M., Rae L., 2004: Geography Markup Language: Foundation for the Geo-Web. Wiley (w znacznej części dostępna bezpłatnie)
URL: http://media.wiley.com/product_ancillary/47/04708715/DOWNLOAD/Lake.zip
- Mark D. M., Skupin A., Smith B., 2001: Features, Objects, and other Things: Ontological Distinctions in the Geographic Domain. Spatial Information Theory, Proceedings of COSIT 2001, Springer.
URL: <http://wings.buffalo.edu/philosophy/faculty/smith/articles/COSIT01MSS.pdf>
- Michalak J., 2003a: Modele pojęciowe hydrogeologicznych danych geoprzestrzennych – podstawy metodyczne. Biuletyn PIG – *Hydrogeologia*, z. V, nr 406, monografia.
- Michalak J., 2003b: Podstawy metodyczne i technologiczne infrastruktur geoinformacyjnych. *Roczniki Geomatyki*, t. 1, z. 2. PTIP, Warszawa, monografia.
- Michalak J., 2003c: Geomatics in hydrogeology. *Geological Quarterly*, 47(1): 69-76.
- Michalak J., 2005a: Terminologia polska w zakresie technologii interoperacyjnych w geomatyce. [W:] (red.) Gaździcki J. Internetowy Leksykon Geomatyczny.
URL: <http://www.ptip.org.pl/auto.php?page=Encyclopedia&enc=1>
- Michalak J., 2005b: HGLML – HydroGeoLogical Markup Language – znacznikowy język wymiany geoinformacji hydrogeologicznej. Współczesne Problemy Hydrogeologii, t. XII: 499-504.
- Michalak J., Nawalany M., Sadurski A., (red.) 2011: Schematyzacja warunków hydrogeologicznych na potrzeby numerycznego modelowania przepływu w JCWPd. Wyd. PIG – PIB, Warszawa.
URL: http://www.psh.gov.pl/plik/id,6091,v,artykul_4556.pdf
- Michalak J., 2012: Testowanie roboczych wersji specyfikacji danych tematów załączników II i III INSPIRE. *Roczniki Geomatyki*, t. 10, z. 2, PTIP, Warszawa.
- Nowicki B., Staniszkis W., 2002: Inteligentny system zarządzania wiedzą – prezentacja projektu. [W:] Mat. Konferencji eDemocracy, VI Konf. Miasta w Internecie, Zakopane.
- OMG (Object Management Group), 2001: OMG Unified Modeling Language Specification, version 1.4. OMG Document Repository. URL: <http://cgi.omg.org/docs/formal/01-09-67.pdf>
- OMG (Object Management Group), 2003: Object Management Group, Model Driven Architecture Guide Version 1.0.1 URL: <http://www.omg.org/cgi-bin/doc?omg/03-06-01.pdf>
- OMG (Object Management Group), 2010: Object Constraint Language.
URL: <http://www.omg.org/spec/OCL/2.2>
- Pachelski W., Parzyński Z., 2007: Aspekty metodyczne wykorzystania norm serii ISO 19100 do budowy geodezyjnych składników krajowej infrastruktury danych przestrzennych. *Roczniki Geomatyki*, t.5, z.3, PTIP, Warszawa.
- Peng Z. R., Zhang C., 2004: The roles of geography markup language (GML), scalable vector graphics (SVG), and Web feature service (WFS) specifications in the development of Internet geographic information systems (GIS).
URL: <http://gis.geog.uconn.edu/personal/paper1/journal%20paper/3%202004%20GeographicalSystem1.pdf>
- PKN (Polski Komitet Normalizacyjny), 2010: Informacje podstawowe o PN.
URL: <http://www.pkn.pl/informacje-podstawowe-o-pn>
- Płoski Z., 1999: Słownik Encyklopedyczny – Informatyka. Wyd. Europa, Warszawa.
- Portele C., 2007: OpenGIS Geography Markup Language (GML) Encoding Standard. OpenGIS Standard.
URL: http://portal.opengeospatial.org/files/?artifact_id=20509
- Portele C., 2008a: Mapping UML to GML Application Schemas. Guidelines and Encoding Rules. Interactive Instruments GmbH.
URL: <http://www.interactive-instruments.de/ugas/UGAS-Guidelines-and-Encoding-Rules.pdf>
- Portele C., 2008b: Mapping UML to GML Application Schemas. ShapeChange – Architecture and Description. Interactive Instruments GmbH. URL: <http://www.interactive-instruments.de/ugas/ShapeChange.pdf>
- Portele C., 2012: OGC Geography Markup Language (GML) – Extended schemas and encoding rules. OpenGIS Implementation Standard. URL: https://portal.opengeospatial.org/files/?artifact_id=46568
- Refsgaard J. C., Henriksen H. J., 2004: Modelling guidelines – terminology and guiding principles. *Advances in Water Resources* 27 (2004): 71-82.
- Schmuller J., 2003: UML dla każdego. Helion, Gliwice.
- Ustawa z dnia 12 września 2002 r. o normalizacji.
URL: http://www.pkn.pl/sites/default/files/ustawa_o_normalizacji_2.pdf

- Ustawa z dnia 4 marca 2010 r. o infrastrukturze informacji przestrzennej, Dz.U. 2010 Nr 76, poz. 489.
- Skogan D., 1999: UML as a Schema Language for XML based data Interchange. Materiały konferencji UML'99. URL: <http://xml.coverpages.org/skoganUMLpaper-pdf.gz>
- Sparx System, Enterprise Architect. URL: <http://www.sparxsystems.com.au>
- Subieta K., 1998: Obiektowość w projektowaniu i bazach danych. Akademicka Oficyna Wydawnicza PLJ, Warszawa.
- Subieta K., 1999a: Słownik terminów z zakresu obiektowości. Akademicka Oficyna Wyd. PLJ, Warszawa. URL: http://www.ipipan.waw.pl/~subieta/artykuly/sloownik_obiektowosci/hasla_sloownika.html
- Subieta K., 1999b: Wprowadzenie do obiektowych metodyk projektowania i notacji UML. Jedenasta Górńska Szkoła PTI Szczyrk.
- Tennakoon W., 2003: Visualization of GML data using XSLT. URL: http://www.itc.nl/library/Papers_2003/msc/gim/tennakoon.pdf
- TWG GE (INSPIRE Thematic Working Group – Geology), 2011: D2.8.II.4 INSPIRE Data Specification on Geology – Draft Guidelines. Version 2.9.1. URL: http://inspire.jrc.ec.europa.eu/documents/Data_Specifications/INSPIRE_DataSpecification_GE_v2.01.pdf
- TWG-CP (INSPIRE Thematic Working Group – Cadastral Parcels), 2010: D2.8.I.6 INSPIRE Data Specification on Cadastral Parcels – Guidelines, version: 3.0.1. URL: http://inspire.jrc.ec.europa.eu/documents/Data_Specifications/INSPIRE_DataSpecification_CP_v3.0.1.pdf
- WFD WG GIS (Working Group 3.1 – GIS), 2003: Guidance Document No 9 – Implementing the Geographical Information System Elements (GIS) of the Water Framework Directive. Water Framework Directive (WFD) – Common Implementation Strategy. URL: <http://www.ec-gis.org/docs/F2305/GIS-GD.PDF>
- Woolf A., 2009: Enterprise Architect instructions, STFC Rutherford Appleton Laboratory. URL: http://wiki.services.eportal.org/tiki-download_wiki_attachment.php?attId=732
- Zhang C., Peng Z-R., Li W., Day M. J., 2003: GML-Based Interoperable Geographical Databases. URL: <http://www.ucgis.org/summer03/studentpapers/chuanrongzhang.pdf>