STUDIA INFORMATICA 2003
Volume 24 Number 1 (52)

Jacek FRĄCZEK
Politechnika Śląska, Instytut Informatyki

PROJEKTOWANIE EDYTORÓW METADANYCH GEOPRZESTRZENNYCH

Streszczenie. W pracy opisano obszary problemowe związane z projektowaniem edytorów metadanych geoprzestrzennych. Obok określenia wymaganej funkcjonalności edytora metadanych przeanalizowano zagadnienia dostępu do różnych źródeł metadanych, pracy edytora w *n*-warstwowej architekturze systemu metadanych oraz zagadnienia związane z replikacją, wielodostępem i systemem użytkowników. Rozważania teoretyczne uzupełniono przeglądem dostępnych systemów tworzenia i edycji metadanych.

Słowa kluczowe: metadane geoprzestrzenne, edytor, systemy informacji przestrzennej, SIP.

THE DESIGN OF GEOSPATIAL METADATA EDITORS

Summary. The research describes the problem areas relevant to the design of geospatial metadata editors. It defines the required functionality of a metadata editor as well as it analyses the issues of accessing various metadata sources, working in *n*-tier architecture, replication, multi-access and system's users. The review of existing metadata creation and edition systems supplements the theoretical considerations.

Keywords: geospatial metadata, editor, geographic information systems, GIS.

1. Wstęp

Metadane towarzyszące systemom informacji geoprzestrzennej są zestawami danych, które podlegają tworzeniu, odczytowi, aktualizacji i usuwaniu [1]. Globalny dostęp do systemów informacji geoprzestrzennej wymaga zapewnienia dużej efektywności procesu wyszukiwania i udostępniania danych. Stąd też w obecnie istniejących systemach duży nacisk położono na funkcjonalność aplikacji (serwisów WWW) wykorzystywanych do

wyszukiwania informacji oraz na wydajność wykonywania operacji odczytu danych. Równolegle prowadzone prace nad torem zasilania i zarządzania informacjami przechowywanymi w bazach metadanych doprowadziły do utworzenia stosunkowo prostych narzędzi wspomagających te procesy [4]. Podstawowymi narzędziami służącymi do tworzenia, aktualizacji i usuwania informacji w bazach metadanych są edytory metadanych.

Rozbudowane edytory metadanych, oprócz prostych zadań edycji informacji, często wspomagają (lub też w pewnej części automatyzują) procesy tworzenia metadanych, pozwalają utrzymać odpowiednią jakość danych, a niektóre z nich mają wbudowane mechanizmy wyszukiwania (zazwyczaj uproszczone w stosunku do dedykowanych aplikacji wyszukujących dane).

Tworzenie wydajnego edytora metadanych związane jest nie tylko z określeniem i implementacją rozbudowanej funkcjonalności edytora metadanych, ale wymaga rozwiązania wielu problemów wynikających z zastosowanej architektury podsystemu metadanych. Problemy te obejmują m.in.:

- zagadnienia dostępu do danych (pliki, bazy danych, serwisy),
- zagadnienia włączenia edytora w *n*-warstwową architekturę systemu metadanych,
- zagadnienia wielodostępu,
- zagadnienia związane z systemem użytkowników,
- zagadnienia automatyzacji procesu tworzenia i edycji metadanych (w tym i zagadnienie przeniesienia danych z systemów źródłowych).

Wymienione obszary problemowe – charakterystyczne dla systemów bazodanowych [2,3] – w przypadku projektowania edytorów metadanych nabierają specyficznych cech, opisanych w dalszej części pracy.

Prezentowana praca powstała jako rozwinięcie wyników badań i prac wdrożeniowych przeprowadzonych na Wydziale Automatyki, Elektroniki i Informatyki Politechniki Śląskiej w ramach projektu celowego KBN pod tytułem: „Wdrożenie Regionalnego Systemu Informacji Przestrzennej w Województwie Śląskim" na zlecenie Instytutu Systemów Przestrzennych i Katastralnych SA.

2. Funkcjonalność edytora metadanych

Efektywna obsługa metadanych wymaga posiadania odpowiedniej aplikacji edytora metadanych. Wymaganie to jest oczywiste w przypadku metadanych przechowywanych w bazie danych, ponieważ manualne tworzenie zapytań przez użytkownika:

- wymaga znajomości języka zapytań wykorzystywanego źródła danych (z reguły *SQL*),
- wymaga znajomości schematu bazy metadanych,
- jest procesem powolnym (pisanie na klawiaturze),
- często prowadzi do błędów (syntaktycznych, semantycznych i logicznych).

Wymaganie istnienia edytora metadanych dotyczy również metadanych przechowywanych w formacie plików tekstowych (zazwyczaj XML). Edycja dużych zbiorów danych, o złożonej strukturze, z użyciem prostego edytora tekstów jest nieefektywna i narażona na błędy (wymaga m.in. wykorzystania zewnętrznych programów sprawdzających strukturę utworzonego pliku wynikowego).

Celami stosowania edytora metadanych jest:

- prezentacja użytkownikowi interfejsu zgodnego z jego wiedzą biznesową i ukrycie tym samym technicznych szczegółów implementacyjnych,
- stworzenie intuicyjnego, prostego systemu pozwalającego użytkownikowi na pracę z relatywnie złożoną strukturą metadanych,
- zapewnienie odpowiedniej jakości wprowadzanych danych,
- przyśpieszenie procesu wprowadzania danych i późniejszych zmian.

Podstawowym wynikiem działania edytora metadanych jest utworzenie nowego zbioru metadanych. Postać fizyczna nowego obiektu jest zależna od sposobu przechowywania danych (zbiór płaski, baza danych lub dostęp przez serwis). Tworząc nowe metadane użytkownik serwisu i bazy metadanych wprowadza do systemu nowy rekord danych (zgodny z wewnętrznym formatem metadanych), a użytkownik, korzystający ze zbiorów płaskich, z reguły tworzy nowy zbiór płaski.

Zaawansowane edytory metadanych zbiorów płaskich zazwyczaj pozwalają użytkownikowi na wybór standardu, w ramach którego będzie następowała edycja metadanych i zgodnie z którym zostanie utworzony plik wyjściowy. Ze względu na różnorodność wymagań stawianych przez różne standardy metadanych edytory tego typu dysponują możliwościami rekonfiguracji formatek wejściowych, zgodnie z wymaganiami wybranego przez użytkownika standardu. W najbardziej rozbudowanych edytorach tego typu możliwe jest dostosowywanie sposobu pracy edytora z pomocą zewnętrznych plików konfiguracyjnych. Plik konfiguracyjny związany jest z danym standardem metadanych i definiuje:

- wygląd formatki (rozmieszczenie elementów, sposób prezentacji elementu, ograniczenia zakresu wartości, które może przyjmować element, sposób walidacji wartości elementu),
- opcje menu,
- format pliku wyjściowego.

Z reguły użytkownik edytora (lub administrator systemu) ma możliwość przeredagowania pliku konfiguracyjnego i dostosowania go do swoich potrzeb. Zastosowanie technologii pliku konfiguracyjnego pozwala na zmianę funkcjonalności edytora (zmiana wyglądu formatki, dodanie lub usunięcie niektórych elementów), jak też zapewnia możliwość dostosowania go do przyszłych wymagań standardów metadanych, bez potrzeby reedycji tekstu źródłowego programu i ponownej kompilacji aplikacji przez jego autorów.

W przypadku edytorów baz metadanych formatka wprowadzania danych jest zazwyczaj stała, dostarczane są natomiast moduły pozwalające na eksport danych z bazy do pliku w określonym standardzie (o ile baza danych przechowuje dane wymagane przez ten standard). W bardziej zaawansowanych edytorach tego typu możliwe jest sterowanie wyglądem formatek ekranowych oraz procesami eksportu/importu danych na podstawie dodatkowych informacji konfiguracyjnych przechowywanych w samej bazie metadanych.

Edycja metadanych funkcjonalnie odpowiada edycji informacji w bazie danych. Z tego względu wymagania funkcjonalne dotyczące edytora metadanych nie odbiegają od ogólnie przyjętych wymagań nakładanych na aplikacje bazodanowe. Dodatkowa analiza komercyjnych edytorów metadanych pozwala zdefiniować uniwersalny zestaw wymagań funkcjonalnych, które powinien spełniać dobry edytor metadanych. Przy definiowaniu wymagań przyjęto następujące założenia:

- edytor będzie umożliwiał wykonanie pełnego zestawu operacji edycyjnych na obiektach metadanych,
- edytor będzie współpracował z bazą danych,
- edytor będzie umożliwiał pracę w wielodostępie,
- edytor będzie miał graficzny interfejs użytkownika.

2.1. Wymagania szczegółowe

Edycja danych:

- tworzenie, modyfikacja i usuwanie danych zgodnie z przyjętym standardem (standardami) metadanych geoprzestrzennych: ISO 19115, CSDGM:
  - obsługa pełnego zestawu elementów obowiązkowych,
  - obsługa elementów opcjonalnych,
  - obsługa dodatkowych elementów rozszerzających (obiektów i formatów zdefiniowanych przez użytkownika),
- wyświetlanie i edycja hierarchicznej struktury metadanych:
  - opcje rozwijania i zwijania gałęzi drzewa,
  - możliwość przeniesienia obiektów pomiędzy różnymi poziomami hierarchii,

- możliwość tworzenia obiektów na różnych poziomach struktury hierarchicznej,
- możliwość usunięcia obiektu (hierarchicznie podległe obiekty podrzędne można usunąć bądź przenieść np. do innej gałęzi na tym samym poziomie),
- możliwość edycji obiektów:
  - ustawienie wartości atrybutów dla wszystkich potomków,
  - opcja dziedziczenia atrybutów rodziców przez potomków,
- możliwość wykonywania ww. operacji na wielu wskazanych obiektach,

- badanie poprawności danych:
  - na poziomie pola:
    - zgodność z wymaganym formatem (typem) – edytor nie powinien pozwalać na wprowadzenie znaków niezgodnych z typem pola (np. litery do pola typu liczba),
    - zgodność ze zbiorem wymaganych wartości,
    - dla pól tekstowych – zgodność z zasadami ortografii i stylistyki języka (zastosowanie słowników),
  - badanie poprawności danych na poziomie formatki,
  - badanie poprawności danych całego obiektu (może korzystać z procedur stworzonych w bazie danych),
- wyszukiwanie metadanych:
  - ograniczone do hierarchicznej struktury metadanych (np. nazwy wyświetlanej w drzewie),
  - pełne (jak w module przeglądarki metadanych),
- opcja kreatora metadanych – ze względu na znaczną złożoność metadanych, kreator powinien umożliwiać, w przyjazny sposób, poprowadzenie użytkownika przez pełny proces tworzenia nowego obiektu metadanych (modyfikacji obiektu już istniejącego); kreator może zostać wyposażony w dodatkowy spis treści umożliwiający skokowe przechodzenie pomiędzy poszczególnymi etapami pracy kreatora,
- dostosowanie wyglądu formatek:
  - do rodzaju edytowanego obiektu, lub
  - do odpowiedniego poziomu hierarchii metadanych (przy stałej hierarchii),
- praca w środowisku wielodostępnym,
- import/eksport metadanych w formacie zgodnym z podanym standardem,
- raportowanie z bazy metadanych:
  - raporty ogólne (statystyczne, typu: liczba i rodzaj obiektów, gestorzy danych, obiekty i prawa do obiektów),

- raporty szczegółowe dotyczące wybranego obiektu(ów),

- tworzenie zagregowanych metadanych statystycznych (na podstawie metadanych już wprowadzonych),
- zachowywanie historii zmian metadanych,
- automatyczne tworzenie i aktualizacja metadanych zgodnie z bieżącymi własnościami danych.

Bezpieczeństwo:

- autoryzacja użytkowników:
  - własny system użytkowników (aplikacja, baza danych), lub
  - integracja z systemem operacyjnym, usługami katalogowymi, w celu uzyskania systemu typu *single sign-on*,
- określenie dostępnej funkcjonalności edytora dla danego użytkownika:
  - ustalenie dostępności opcji menu (opcje niedostępne powinny zostać usunięte lub nieaktywne),
  - ustalenie dostępności elementów formatek,
  - określenie zestawu autoryzowanych danych dla danego użytkownika.

Wymagania dotyczące prezentacji:

- interfejs graficzny,
- grupowanie obiektów i elementów powiązanych z sobą (np. poprzez wykorzystanie zakładek) zgodnie z przyjętym standardem metadanych:
  - identyfikacja: tytuł, obszar danych, tematy, słowa kluczowe, aktualność danych, opis, streszczenie, cel,
  - jakość danych: dokładność (atrybutów, opisu obszaru), kompletność, logiczna spójność, pochodzenie,
  - organizacja danych przestrzennych: rodzaj danych (wektorowe, rastrowe), typ elementów, liczba elementów,
  - odniesienie przestrzenne: projekcja, układ siatki, podstawa wymiarowa, układ współrzędnych,
  - zawartość danych – obiekty i atrybuty: cechy, atrybuty, wartości atrybutów,
  - dystrybucja danych: gestor, formaty, media, dostępność poprzez system komputerowy, cena,
  - informacje o metadanych: aktualność metadanych, ciało odpowiedzialne za metadane,
  - inne,
- opis obiektów i ich elementów w języku zrozumiałym dla przeciętnego użytkownika,
- system podpowiedzi:

- wskazanie elementów obowiązkowych (poprzez wyróżnienie innym kolorem, opatrzenie opisu elementów gwiazdką, itp.),
- wskazanie dozwolonego formatu danych (liczb, dat, czasu, pól tekstowych – wyświetlenie informacji na pasku statusu i w tekście podpowiedzi zwanym popularnie „dymkiem"),
- opis dozwolonych wartości - wybór wartości ze słownika nie pozwala użytkownikowi na wprowadzenie danych spoza dziedziny dozwolonej standardem.

Pozostałe wymagania dotyczące edytora metadanych:

- możliwość wskazania bazy metadanych, z którą użytkownik chce pracować,
- implementacja powinna zapewniać następujące cechy uniwersalności aplikacji:
  - możliwości uruchomienia edytora w różnych systemach operacyjnych,
  - możliwość współpracy z różnymi bazami danych,
  - ewentualna możliwość uruchomienia edytora w przeglądarce WWW,
- zabezpieczenie transmisji danych:
  - użycie bezpiecznego kanału transmisji (SSL),
  - kodowanie danych.

3. Wpływ architektury podsystemu metadanych na edytor metadanych

W wielu współcześnie tworzonych systemach operujących na zbiorach informacji, metadane są jednym z podstawowych komponentów systemu. Sposób budowy i umiejscowienie podsystemu metadanych w architekturze całego systemu ma znaczący wpływ na proces implementacji edytora metadanych. Podsystem metadanych:

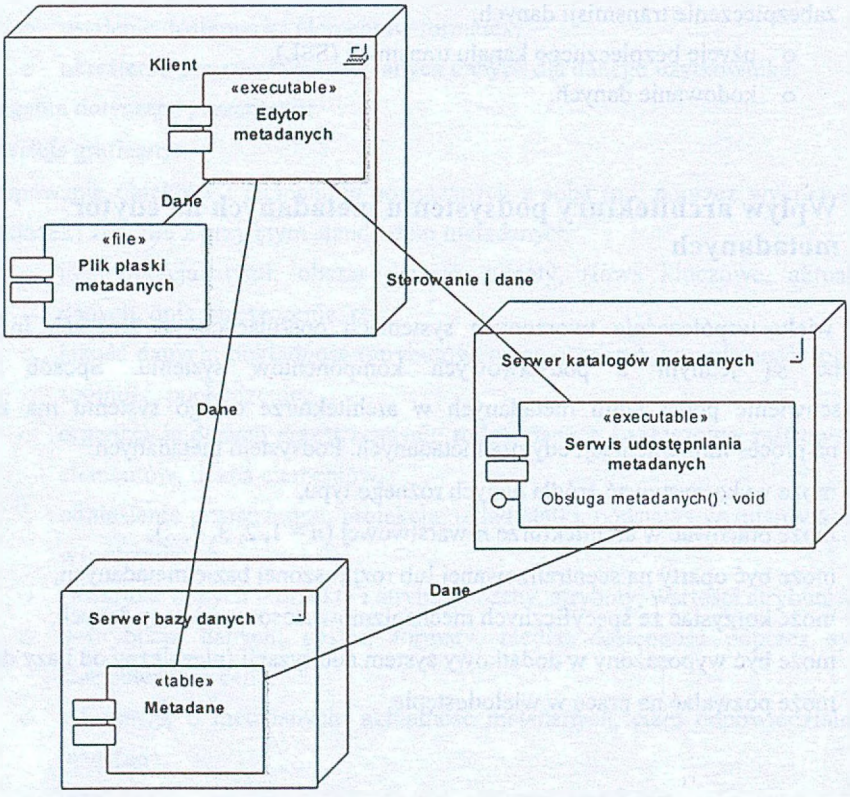
- może wykorzystywać źródła danych różnego typu,
- może pracować w architekturze n-warstwowej (n = 1, 2, 3, 4,...),
- może być oparty na scentralizowanej lub rozproszonej bazie metadanych,
- może korzystać ze specyficznych mechanizmów stosowanej bazy danych,
- może być wyposażony w dodatkowy system autoryzacji (niezależny od bazy danych),
- może pozwalać na pracę w wielodostępie.

3.1. Dostęp do źródeł danych

W praktyce spotyka się następujące sposoby przechowywania metadanych (rys.1):

- w postaci zbiorów płaskich:
  - o w formacie własnym,
  - o w formacie XML (struktury unormowane standardem),
- w postaci bazy danych:
  - o relacyjnej,
  - o obiektowej.

Wśród edytorów metadanych można wyróżnić jeszcze trzecią grupę narzędzi, które nie współpracują bezpośrednio z danymi, ale z serwisami zarządzającymi danymi (usługi udostępniania metadanych) [20]. Ten sposób dostępu do metadanych można traktować jako implementację odmiennej metody dostępu do tradycyjnych źródeł danych (zazwyczaj bazy danych) wykorzystującej architekturę wielowarstwową.



Rys. 1. Ogólny schemat współpracy edytora metadanych ze źródłami metadanych
Fig. 1. The general schema of cooperation of a metadata editor and data sources

Najodpowiedniejszym miejscem do przechowywania i zarządzania dużymi ilościami informacji są bazy danych, stąd też wykorzystuje się je jako główny sposób przechowywania metadanych. Metadane w postaci zbiorów płaskich służą głównie jako uniwersalny środek wymiany informacji między systemami. Edytory metadanych z reguły obsługują jedną z wymienionych postaci metadanych.

Między edytorami metadanych współpracującymi z różnymi źródłami danych istnieją dość istotne różnice:

- w sposobie fizycznego dostępu do danych:
  - poprzez standardowe funkcje odczytu/zapisu danych (C/C++, Pascal),
  - poprzez parser XML,
  - poprzez moduł dostępowy do bazy danych (dedykowany, ODBC, JDBC),
  - poprzez wywołanie funkcji udostępnianych przez serwis (np. poprzez usługi internetowe *web services*),
- w możliwościach wykorzystania już istniejących mechanizmów autentykacji i autoryzacji dostępu do danych,
- w możliwościach zapewnienia pracy w wielodostępie,
- w możliwościach wykorzystania mechanizmów bazy danych (np. konstrukcji optymalizujących, generatorów unikalnych identyfikatorów, procedur składowanych),
- w możliwościach efektywnej obsługi zbiorów danych dowolnego rozmiaru.

Warto również wspomnieć, że niektóre systemy zarządzania bazami danych pozwalają na przechowywanie w bazie danych dokumentów XML (metadanych) i dostęp do nich z poziomu języka *SQL* [21]. Połączenie zalet dokumentów XML z możliwościami baz danych prowadzi do stworzenia nowej generacji systemów metadanych i obsługujących je edytorów.

3.2. Edytor w *n*-warstwowej architekturze podsystemu metadanych

Ustalona przez projektantów architektura podsystemu metadanych i sposób dostępu do źródeł danych w zasadniczy sposób wpływają na implementację edytora metadanych. Wymaganie współpracy edytora ze wskazaną warstwą architektury podsystemu ma odzwierciedlenie w złożoności kodu aplikacji i możliwości samodzielnej pracy edytora.

W zależności od sposobu implementacji metadanych można rozważać następujące rodzaje architektury podsystemu metadanych:

- pliki płaskie – architektura 1- lub 2-warstwowa (z serwerem plików),
- baza danych – ogólnie architektura *n*-warstwowa,
- serwis udostępniający metadane – architektura *n*-warstwowa (n>=2).

Stosunkowo prostą implementację mogą mieć edytory metadanych współpracujące z serwisem udostępniającym metadane lub jego odpowiednikiem (serwer aplikacji) w *n*-

warstwowej architekturze z bazą danych. W tym przypadku funkcje udostępniane przez serwis mogą zapewniać całą wymaganą przez użytkownika metadanych funkcjonalność (tworzenie, usuwanie, modyfikacja, walidacja, wyszukiwanie), a sam edytor może być tzw. cienkim klientem odpowiedzialnym wyłącznie za pobieranie i odpowiednie wyświetlanie danych. W przypadku gdy używany serwis nie zapewnia pełnej wymaganej funkcjonalności, to odpowiedzialnością za jej implementację zostaje obarczony programista tworzący edytor.

Większy stopień złożoności prezentują edytory współpracujące bezpośrednio z bazą danych. W takim przypadku edytor musi implementować m.in. specyficzny dla danego systemu sposób dostępu do danych (w tym i specyficzne polecenia *SQL*).

Edytory plików płaskich są aplikacjami o stosunkowo dużym stopniu złożoności. Operacje wykonywane na plikach danych: wstawianie, usuwanie, modyfikowanie i wyszukiwanie elementów jest bardziej złożone niż w przypadku bazy danych, gdyż aplikacja sama musi zarządzać plikiem danych. Dodatkowo, trudno jest zapewnić pracę wielodostępną na pojedynczym pliku danych tekstowych. Z tego względu edytory plików płaskich są z reguły aplikacjami umożliwiającymi pracę na danym pliku tylko pojedynczemu użytkownikowi.

3.3. Rozproszenie bazy metadanych

W przypadku systemów informacji geoprzestrzennej, których architektura odpowiada złożonej strukturze organizacyjnej (np. kraj→województwo→powiat→gmina), celowe może być rozproszenie bazy metadanych. Rozproszenie danych ma służyć zwiększeniu wydajności pracy systemu na poziomach lokalnych oraz uniezależnieniu systemów lokalnych od łączności z serwerem centralnym. Niektóre konfiguracje architektur z rozproszeniem danych wpływają na konstrukcję edytora metadanych. Analizując wpływ rozproszenia danych na edytor metadanych należy wziąć pod uwagę:

- zapewnienie jednolitości słowników metadanych,
- ograniczenia związane z dostępem do danych.

3.3.1. Jednolitość słowników metadanych

Zapewnienie jednolitości słowników metadanych (definicji, opisów, słów kluczowych, itp.) gwarantuje zachowanie spójności informacji w obszarze całego systemu. Każdy węzeł hierarchii organizacyjnej powinien pracować z identycznym zestawem słowników. Jednym z możliwych rozwiązań tego problemu – w przypadku architektury rozproszonej – może być wykorzystanie mechanizmu replikacji słowników. Dodatkowo można na system narzucić wymaganie, aby modyfikacja słowników (w tym i dopisywanie nowych wartości) była realizowane wyłącznie w najwyższym węźle hierarchii, a replikacja była replikacją

jednokierunkową: od korzenia do liści struktury. Przyjmując takie założenie, należy zapewnić, aby edytor metadanych dopuszczał możliwość edycji słowników tylko we wskazanych węzłach (w tym przypadku w pojedynczym węźle – korzeniu drzewa). Żądany efekt można uzyskać implementując jedno z następujących rozwiązań:

- poprzez mechanizmy konfiguracji edytora (plik konfiguracyjny, tablice konfiguracji),
- poprzez uzależnienie funkcjonalności edytora od nazwy lub przynależności grupowej użytkownika, który loguje się do systemu (użytkownik końcowy, administrator),
- poprzez utworzenie kilku wersji edytora metadanych o różnej funkcjonalności.

Pierwsze z rozwiązań może stwarzać pewne problemy związane z ochroną pliku konfiguracyjnego i jego zawartości. W celu zapobieżenia ewentualnym zmianom w pliku konfiguracyjnym powinien on być niedostępny dla osób niepowołanych, a jego zawartość powinna być zakodowana. Z tego względu większy stopień bezpieczeństwa zapewnia zapisanie konfiguracji danej stacji w tablicy konfiguracji w bazie danych lub też uzależnienie dostępnej funkcjonalności edytora od użytkownika, który loguje się do systemu. W takim wypadku nie ma konieczności przechowywania lokalnej informacji o konfiguracji aplikacji.

W systemie bez rozproszenia danych (z pojedynczym serwerem centralnym) problem zapewnienia jednolitości słowników nie występuje.

3.3.2. Ograniczenia w dostępie do danych

W przypadku gdy aplikacje pracujące na bazie metadanych mają własny system użytkowników (niezależny od bazy danych, do której wszystkie aplikacje logują się na użytkownika o identycznej nazwie – patrz niżej), to muszą one zapewnić autoryzowany dostęp do danych. Działanie edytora metadanych powinno być zgodne z przyjętą strategią bezpieczeństwa.

W przypadku występowania hierarchicznej struktury organizacyjnej (np. kraj→województwo→powiat→gmina) rozsądnym rozwiązaniem może być przyjęcie wymagania, aby użytkownicy przypisani do danego ośrodka (węzła) mogli mieć dostęp (do odczytu i edycji) tylko do danych własnego węzła. Wymaganie to w naturalny sposób jest realizowane w systemie rozproszonym, w którym każdy węzeł przechowuje tylko własne dane. W systemie z centralnym serwerem lub też w systemie rozproszonym, w którym każdy węzeł przechowuje wszystkie dane, to edytor metadanych jest komponentem odpowiedzialnym za realizację tego wymagania. W takim przypadku metadane węzłów powinny być oznaczane znacznikiem pochodzenia, a edytor metadanych (sam lub przy współpracy z bazą danych) powinien na podstawie nazwy użytkownika udostępniać tylko autoryzowane informacje. Tabela 1 specyfikuje problemy występujące dla różnych form rozproszenia danych.

Tabela 1

Wpływ rozproszenia danych na projekt edytora metadanych

Lp.	Rodzaj rozproszenia	Problemy występujące w edytorze metadanych
1	brak - centralny serwer metadanych	• użytkownicy węzła - autoryzacja dostępu do danych własnego węzła
2	węzeł przechowuje tylko metadane własne	• zapewnienie jednolitości słowników
3	węzeł przechowuje metadane własne i metadane węzłów podrzędnych	• użytkownicy węzła - autoryzacja dostępu do danych własnego węzła (można zezwolić na dostęp do danych wszystkich węzłów podrzędnych) • zapewnienie jednolitości słowników
4	węzeł przechowuje wszystkie metadane (własne i innych węzłów)	• użytkownicy węzła - autoryzacja dostępu do danych własnego węzła • zapewnienie jednolitości słowników

3.4. Wykorzystanie specyficznych mechanizmów bazy danych

Systemy zarządzania bazami danych oferują wiele mechanizmów wspomagających pracę aplikacji. Do mechanizmów tych można zaliczyć m.in.:

- możliwość pracy w wielodostępie (system użytkowników, transakcyjność, mechanizmy blokowania danych),
- możliwość autentykacji i autoryzacji dostępu do danych (system użytkowników, prawa, role, perspektywy, wirtualne, prywatne bazy danych),
- optymalizację pracy z danymi (prowadzenie statystyk, histogramów),
- możliwość prowadzenia audytu (dzienniki historii wykonywanych działań),
- inne udogodnienia, tj.: mechanizmy tworzenia unikalnych identyfikatorów, mechanizmy kaskadowego modyfikowania danych, mechanizmy kaskadowego usuwania danych.

Wykorzystanie wyżej wymienionych mechanizmów pozwala twórcom edytora metadanych zastosować gotowe rozwiązania i uprościć projekt aplikacji oraz zapewnić dużą wydajność pracy całego systemu. Należy jednak pamiętać, że nie wszystkie systemy zarządzania bazami danych oferują pełny zakres wymienionych mechanizmów. Dodatkowo, wykorzystanie specyficznych cech systemu zarządzania bazą danych (np. sposobu generacji unikalnych identyfikatorów) ogranicza możliwości współpracy aplikacji tylko do platformy korzystającej z wybranej bazy danych.

3.5. System użytkowników

Zabezpieczenie dostępu do danych w systemach bazodanowych realizowane jest zasadniczo w oparciu o wykorzystanie jednego z dwóch mechanizmów:

- mechanizmu bezpieczeństwa wbudowanego w system zarządzania bazą danych,
- mechanizmu bezpieczeństwa wbudowanego w aplikację użytkownika.

Wymienione mechanizmy mogą być zintegrowane z mechanizmami bezpieczeństwa systemu operacyjnego.

Wykorzystanie mechanizmów bezpieczeństwa wbudowanych w system zarządzania bazą danych pozwala zbudować elastyczny i jednolity system zabezpieczeń dla wszystkich aplikacji korzystających z podsystemu metadanych. Implementacja takiego systemu z wykorzystaniem perspektyw może jednak prowadzić do znacznego spadku wydajności pracy systemu.

W przypadku istnienia pojedynczej aplikacji udostępniającej metadane możliwe jest wbudowanie mechanizmów bezpieczeństwa w samą aplikację. Aplikacja jest wtedy w pełni odpowiedzialna za kontrolę prezentowanych danych oraz czynności, które użytkownik może wykonać. W rozwiązaniu tym należy założyć ukrycie przed użytkownikiem nazwy i hasła rzeczywistego użytkownika wykorzystywanego w procesie logowania do bazy danych. Informacje te muszą zostać utajnione, aby zapobiec połączeniu się do bazy metadanych z poziomu innego narzędzia, co pozwoliłoby obejść zabezpieczenia narzucone przez aplikację edytora metadanych. W przypadku wielu aplikacji korzystających z podsystemu metadanych należy do każdej z nich dołączyć identyczny moduł zarządzający bezpieczeństwem.

Wbudowanie mechanizmów bezpieczeństwa w aplikację ułatwia sterowanie funkcjonalnością aplikacji oraz zakresem widoczności danych. Dodatkowo omawiane rozwiązanie może pozwolić na budowę systemu o większej szybkości działania w porównaniu do rozwiązania opartego na mechanizmach bazy danych.

Wymienione zalety rozwiązania wykorzystującego mechanizmy bezpieczeństwa wbudowane w aplikację edytora metadanych nie rekompensują wad tego rozwiązania:

- wymaganie stosowania identycznego modułu zarządzającego bezpieczeństwem we wszystkich aplikacjach pracujących z bazą metadanych,
- z reguły nie rozróżnia się użytkowników na poziomie bazy danych – używane są pojedyncze konta użytkowników bazy danych o pełnych prawach,
- wzrost złożoności aplikacji,
- konieczność modyfikacji aplikacji przy zmianach procedur bezpieczeństwa.

W przypadku edytora metadanych można wybrać rozwiązanie, w którym użytkownicy logują się do systemu z wykorzystaniem nazw zarejestrowanych w aplikacji. Połączenie do bazy danych jest realizowane na konto pojedynczego, dedykowanego użytkownika.

3.5.1. Użytkownicy aplikacji pracujących z podsystemem metadanych

Do podstawowych aplikacji pracujących z podsystemem metadanych zaliczamy:

- edytor metadanych,
- przeglądarkę metadanych.

Obie aplikacje posiadają bardzo wiele cech wspólnych, natomiast różnią się zasadniczo funkcjonalnością. Przeglądarka metadanych nie pozwala na edycję danych, natomiast umożliwia złożone wyszukiwanie danych. Edytor metadanych jest swego rodzaju przeglądarką metadanych, ale nie pozwala na zaawansowane wyszukiwanie danych.

W aplikacjach korzystających z bazy metadanych można wyróżnić następujących użytkowników:

- administrator węzła danych (administrator),
- użytkownicy modyfikujący dane w węźle,
- użytkownicy przeglądający metadane.

Dwaj pierwsi użytkownicy są związani z aplikacją edytora metadanych, trzeci – z przeglądarką metadanych.

Administrator węzła danych jest użytkownikiem zarządzającym danymi w określonym węźle. Użytkownik ten ma uprawnienia do:

- przeglądania wszystkich danych,
- modyfikacji metadanych (w tym i słowników): wprowadzania, usuwania i aktualizacji,
- zarządzania użytkownikami aplikacji (tworzenie, usuwanie, modyfikacja),
- modyfikacji hasła użytkownika bazy danych (nazwa i inne parametry połączenia są zapisane w konfiguracji aplikacji),

Użytkownicy modyfikujący dane w węźle mają prawo do:

- przeglądania wszystkich danych,
- wprowadzania, usuwania i aktualizacji danych z następującymi ograniczeniami:
    - ww. operacje mogą dotyczyć wyłącznie danych przypisanych do użytkownika,
    - użytkownicy nie mają możliwości modyfikacji słowników.

W przypadku systemu rozproszonego korzystne może się okazać wprowadzenie dodatkowego użytkownika – administratora słowników. Administrator słowników powinien zarządzać aktualizacją słowników metadanych i ich replikacją do pozostałych węzłów.

3.6. Wielodostęp

Jednym z wymagań, które powinien spełniać edytor metadanych, jest umożliwienie pracy w wielodostępie. W przypadku pracy z bazą danych realizację mechanizmów wielodostępu (system użytkowników, transakcyjność, mechanizmy blokowania danych) zapewnia system zarządzania bazą danych. Do zadań projektanta edytora metadanych należą określenie i implementacja efektywnej strategii blokowania danych. W rozwiązaniach komercyjnych można spotkać następujące strategie blokowania danych na poziomie rekordu danych (uszeregowane w kierunku wzrastającej długości trwania blokady) [5]:

- bez blokowania,
- blokowanie w chwili zapisu rekordu (po jego edycji),
- blokowanie w chwili rozpoczęcia edycji rekordu,
- blokowanie w chwili wyboru rekordu (po odczycie rekordu, jeszcze przed rozpoczęciem jego edycji).

Dokonując analizy problemu wielodostępu w kontekście pracy edytora metadanych można zauważyć, że uzasadniona może być rezygnacja z blokowania danych przy wykonywaniu podstawowych zadań edycyjnych wykonywanych na pojedynczych obiektach. Do edycji obiektów dochodzi bowiem rzadko, a edycja pojedynczego obiektu przez większą liczbę użytkowników będzie prowadziła co najwyżej do zgłoszenia błędu niespójności obiektu (przez procedury sprawdzające) i konieczności jego ponownej edycji. Z kolei operacje złożone – takie jak np. niektóre operacje wykonywane na grupach danych w hierarchicznym drzewie metadanych – wymagają zabezpieczenia mechanizmami transakcyjności.

3.7. Automatyzacja edycji metadanych

Automatyzacja procesu utrzymywania metadanych w stanie pełnej aktualności jest zagadnieniem trudnym. W przypadku niektórych danych geoprzestrzennych (dokładniej formatów zapisu danych) możliwe jest częściowe wnioskowanie o posiadanych danych, niemniej pełna informacja wymagana przez standardy opisu metadanych z reguły nie jest dostępna. Duża część informacji o danych nie jest wystarczająco dokładnie skatalogowana w postaci elektronicznej i często się zdarza, że wiedzę na ich temat mają wyłącznie aktualni administratorzy i użytkownicy danych. Systemy obsługi metadanych powinny się przyczynić do zmiany tej sytuacji.

Możliwość automatycznego uzyskania wyczerpującego (w stopniu wymaganym przez dany standard metadanych) opisu danych z reguły zależy od formatu zapisu obiektu. Niektóre

formaty przechowywania obiektów geoprzestrzennych obok danych zawierają w sobie także i metadane. W takim przypadku możliwa jest automatyczna ekstrakcja informacji opisowej (metadanych) przy rejestrowaniu danych w systemie. Przykładowe narzędzie *FGDCMETA.AML* potrafi ze struktury plików danych geoprzestrzennych *ArcInfo* odczytać następujące informacje:

- o wykorzystywanym systemie współrzędnych,
- o prostokątach ograniczających obiekty,
- o liczbie obiektów (wektorowych, rastrowych).

Czasami pewne dodatkowe informacje należące do metadanych można znaleźć w:

- nazwie nośnika, na którym jest przechowywany obiekt geoprzestrzenny (np. CD-ROM: *Mapy sozologiczne*),
- ścieżce dostępu do danych (np. *C:\Mapy topograficzne\Układ 1992\Skala 10000*),
- nazwie pliku (np. *topo_50_92* – mapa topograficzna, w układzie 1992, w skali 1:50000).

Zautomatyzowaną obsługę bazy metadanych można szerzej wykorzystać w przypadku, gdy baza metadanych przechowuje informacje statystyczne. Informacje takie można generować na dwa sposoby:

- na podstawie informacji zawartej w samych obiektach danych – można tu automatycznie kolekcjonować rodzaj przechowywanych obiektów, liczebność obiektów, atrybuty, jakie mają obiekty (np. z pliku w formacie *shape*),
- poprzez agregację metadanych należących do niższych poziomów drzewa hierarchii metadanych (np. sumaryczna liczba obiektów w pakiecie map).

W ogólnym przypadku informacje, które są wynikiem automatycznej analizy wskazanych zbiorów danych, często nie są wystarczające do uzyskania wymaganego, pełnego opisu metadanych. Podczas procesu synchronizacji metadanych administrator musi manualnie uzupełnić brakujące elementy opisu i ewentualnie dołączyć dodatkową dokumentację.

3.8. Import metadanych

Jedną z pożądanych cech funkcjonalnych edytora metadanych jest możliwość importu metadanych. Metadane, które użytkownicy chcą wprowadzić do systemu, mogą być zapisane w różnych formatach plików lub też mogą pochodzić z różnych zewnętrznych baz danych. Projektowany edytor metadanych można wyposażyć w funkcje importu metadanych plików płaskich uznanych standardów i automatycznej generacji metadanych dla popularnych formatów zapisu danych geoprzestrzennych.

W celu przeniesienia metadanych zapisanych w formatach własnych (plikach płaskich, bazach danych) powinno się utworzyć dedykowaną aplikację konwersji formatów danych.

Aplikacja taka powinna:

- dokonać transformacji danych ze starego do nowego systemu (formatu) metadanych,
- dokonać konwersji formatów danych (np. gdy nowy format wymaga typu logicznego *true/false*, a stary format przechowuje wartość numeryczną *1/0*),
- dokonać konwersji formatu kodowania napisów (np. z *MAZOVII* na *Unicode*),
- uzupełnić brakujące dane (w miarę możliwości),
- mieć możliwość konfiguracji procesu konwersji (np. wartości domyślne dla elementów, których nie ma w importowanym formacie).

4. Przegląd systemów tworzenia i edycji metadanych

Przegląd dostępnych na rynku narzędzi do tworzenia i edycji metadanych dostarcza wiedzy na temat stopnia rozwoju technologii edycji metadanych. Ze względu na częsty brak uznanych międzynarodowych standardów metadanych (np. w dziedzinie danych geoprzestrzennych), wielość standardów narodowych oraz stosunkowo małą złożoność realizacji podstawowych zagadnień obsługi metadanych, na rynku znaleźć można bardzo wiele prostych narzędzi do tworzenia i edycji metadanych [4]. Narzędzia te różnią się funkcjonalnością, technologią wykonania i możliwościami uruchomienia na różnych platformach sprzętowo-programowych. W zakresie funkcjonalności główne różnice dotyczą:

- możliwości obsługi różnych standardów metadanych: FGDC, Dublin Core, GILS, ANZLIC, AGLS, NZGLS, EdNA, IMS, GEM, vCard, USMARC, SOIF, IAFA/ROADS, nagłówki TEI, RDF, OLSTF, XML i inne,
- możliwości współpracy ze zbiorami płaskimi i/lub bazami danych,
- możliwości eksportu i importu danych w różnych formatach: ASCII (np. hierarchicznie wcinane), SGML, XML,
- możliwości automatycznego wspierania tworzenia metadanych (np. przy opisie zasobów stron WWW),
- możliwości edycji metadanych (niektóre narzędzia pozwalają tylko na tworzenie metadanych bez późniejszej aktualizacji),
- możliwości konfiguracji narzędzi: opisów pól i pozycji menu, formatów pól (typy i zakresy wartości), zmiany wyglądu formatek (położenie i rozmiar elementów) i menu, rozbudowy formatek o dodatkowe elementy, ograniczenia lub rozszerzenia istniejących definicji metadanych, wielojęzykowości aplikacji,
- możliwości publikacji metadanych: (strona WWW, poczta elektroniczna, przesył do centralnego/lokalnego repozytorium lub serwera metadanych),

- możliwości integracji z innymi narzędziami (np. dostęp do danych poprzez bramę ośrodka *Clearinghouse*, integracja z parserem metadanych *mp* służącym do weryfikacji syntaktycznej zgodności struktury metadanych ze standardem FGDC),
- możliwości prowadzenia dziennika historii zmian metadanych,
- możliwości wyszukiwania metadanych,
- możliwości analizy statystycznej danych.

W zakresie sposobu tworzenia i możliwości uruchamiania analizowanych aplikacji można wyróżnić następujące technologie:

- tradycyjne programy napisane w języku C/C++, z możliwością kompilacji w środowisku Unix,
- programy w języku C/C++, Visual Basic dla środowiska MS Windows,
- aplikacje Javy,
- aplety Javy uruchamiane z poziomu przeglądarki WWW,
- formularze uruchamiane z poziomu przeglądarki WWW (HTML, JavaScript, Perl).

Tabela 2

Edytory metadanych

Edytor / Cecha	*Reggie*	*DCdot*	*Xtme, Tkme*	*Anzmeta 1.1*	*Meta browser 1.5*	*SMMS*	*Arc Catalog*
Obsługiwane standardy	Dublin Core, GILS, ANZLIC, AGLS, EdNA, IMS, GEM, vCard	Dublin Core + konwersja do innych	FGDC	ANZLIC	Dublin Core DC, GILS, AGLS, EdNA, NZGLS	FGDC	ISO, FGDC, CEN/TC287 [18,19]
Źródło danych	strona WWW	strona WWW	plik	tylko użytkownik	strona WWW, baza danych	baza danych, plik	baza danych, plik XML
Import metadanych	-	-	tak	-	-	tak	tak
Publikowanie metadanych	strona WWW, e-mail	strona WWW, plik lokalny	plik lokalny	strona WWW, e-mail	plik lokalny (z publikacją WWW), baza danych	baza danych, plik lokalny (z publikacją WWW)	baza danych, plik lokalny (z publikacją WWW)
Integracja z innymi narzędziami	-	-	mp	-	serwer metadanych, słowniki metadanych, Tesaurus	serwer metadanych, MetWeb Clearinghouse	serwer metadanych, Clearinghouse, Geography Network
Konfiguracja	tak	-	tak	-	tak	tak	tak
Technologia	aplet Java	formularz WWW	aplikacja GUI (C/C++)	formularz WWW – JavaScript	aplikacja GUI	aplikacja GUI	aplikacja GUI
Inne	autom. opis stron WWW, wielojęzyko-wość	autom. opis stron WWW				automatyzacja procesów tworzenia danych	automatyzacja procesów tworzenia danych

Jako reprezentatywnych przedstawicieli narzędzi służących do edycji metadanych można wskazać (tabela 2):

- narzędzia uniwersalne służące edycji metadanych należących do różnych dziedzin: Reggie [6], DCdot [7],
- narzędzia służące do edycji metadanych geoprzestrzennych: Xtme (*X Toolkit Metadata Editor*), Tkme (*Tk Metadata Editor*) [8], ANZMETA 1.1 [9], Metabrowser 1.5 [10], SMMS (*Spatial Metadata Management System*) [11], ArcCatalog [12,13,14] wchodzący w skład pakietu *ArcGIS 8.x* [17,19,20].

5. Podsumowanie

Zapewnienie aktualności i spójności metadanych stanowi podstawę efektywnego wykorzystania posiadanych danych. Z tego względu procesy tworzenia i edycji metadanych powinny być wykonywane wydajnie, z zapewnieniem wysokiej jakości przetwarzania danych i w sposób możliwie wysoko zautomatyzowany. O ile – ze względu na wykorzystywane formaty zapisu danych geoprzestrzennych – automatyzacja procesu tworzenia i edycji metadanych jest zagadnieniem trudnym w praktycznej realizacji, o tyle proces manualny może być efektywnie wspomagany przez oprogramowanie. Zaprezentowane w pracy rozwiązania są rozwinięciem wyników badań i prac wdrożeniowych rzeczywistego edytora metadanych.

Dokonany przegląd wybranych edytorów metadanych pozwala na określenie już osiągniętego poziomu rozwoju oprogramowania. Na tej podstawie zdefiniowano, a następnie poszerzono zestaw wymagań, który powinien stanowić bazę dla projektów współczesnych edytorów metadanych. Do cech charakterystycznych takich edytorów można zaliczyć:

- możliwość obsługi kilku formatów metadanych - przede wszystkim: ISO, FGDC oraz formatów własnych,
- upowszechnienie implementacji edytora w technologii umożliwiającej dostęp do danych z sieci Internet – formularz HTML, aplet Javy,
- wykorzystanie interfejsu graficznego z systemem podpowiedzi i opisów wspomagających pracę użytkownika, przy zachowaniu względnej prostoty interfejsu,
- wykorzystanie hierarchicznego drzewa metadanych porządkującego organizację elementów struktury metadanych,
- możliwość wyszukiwania metadanych,
- współpracę edytora z bazą danych,
- możliwość pracy w wielodostępie,
- możliwość importu/eksportu metadanych zgodnych ze powszechnymi standardami.

Na projekt i sposób implementacji edytora metadanych w znaczący sposób wpływa architektura podsystemu metadanych, a zwłaszcza wybór sposobu przechowywania i dostępu do metadanych. Analiza różnych możliwości fizycznej implementacji metadanych prowadzi do wniosku, że najbardziej odpowiednie jest wykorzystanie systemu zarządzania bazą danych. Edytory współpracujące z bazą danych mogą wykorzystywać oferowane przez nią mechanizmy wspomagające pracę aplikacji i w naturalny sposób umożliwiają pracę w wielodostępie. Edytory metadanych przechowywanych w plikach płaskich znajdują zastosowanie do edycji małych zbiorów danych przekazywanych później (jako pliki lub odniesienia) do centrów zarządzania metadanymi. W przyszłości można się spodziewać rozwoju grupy edytorów współpracujących z usługami udostępniającymi metadane w sieci internet/intranet i edytorów współpracujących z bazami danych przechowującymi obiekty XML. Spośród innych czynników wpływających na projekt i pracę edytora metadanych należy wyróżnić:

- możliwość pracy ze scentralizowaną lub rozproszoną bazą metadanych,
- możliwość wykorzystania specyficznych mechanizmów używanej bazy danych,
- możliwość wyposażenia edytora w dodatkowy, własny system autoryzacji (niezależny od bazy danych).

LITERATURA

1. Nebert D.: Developing Spatial Data Infrastructures:The SDI Cookbook, GSDI, 2001. http://www.GSDI.org.
2. Elmasri R., Navathe Sh. B.: Fundamentals of Database Systems, Addison-Wesley, 2000.
3. Connoly T., Begg C., Strachan A.: Database Systems. A Practical Approach to Design, Implementation and Management, Addison-Wesley, 1999.
4. Porównanie narzędzi metadanych, http://badger.state.wi.us/agencies/wlib/sco/metatool/mtools.htm.
5. Magic eDeveloper. Reference Guide, MSE, 2001.
6. Reggie - The Metadata Editor, http://metadata.net/dstc/.
7. Formularz DCdot, http://www.ukoln.ac.uk/metadata/dcdot/.
8. FGDC, http://geology.usgs.gov/tools/metadata/.
9. Formularz ANZMETA, http://geography.anu.edu.au/anzmeta/index.html.
10. Metabrowser, http://metabrowser.spirit.net.au.
11. SMMS, http://www.intergraph.com.
12. Vermeij B.: Implementing European Metadata Using ArcCatalog, ArcUser 07-08.2001. http://www.esri.com.

13. Roberts F.: Arc/Info 8's Metadata Collection tool: ArcCatalog, A user's perspective, Coeur d'Alene Tribe GIS, prezentacja PPT.
14. Elverum K., Palmer B.: Using ArcCatalog to Create and Maintain FGDC Compliant Metadata, Western Regional Support Center, Colorado State University, prezentacja PPT.
15. Features and Functions of ArcInfo 8, An ESRI White Paper, 1999, http://www.esri.com.
16. Working with the Geodatabase: Powerful Multiuser Editing and Sophisticated Data Integrity An ESRI White Paper, 2002, http://www.esri.com.
17. What is ArcGIS™? GIS by ESRI™, ESRI, 2002, http://www.esri.com.
18. ESRI Profile of the Content Standard for Digital Geospatial Metadata, An ESRI Technical Paper, 2001, http://www.esri.com.
19. Vienneau A., Danko D.: Implementing ISO 19115, ESRI, http://www.esri.com, prezentacja PPT.
20. OpenGIS - Catalog Interface Implementation Specification (Version 1.0), http://www.opengis.org.
21. Oracle9i Application Developer's Guide – XML, Oracle Corporation 2002.

Recenzent: Dr inż. Maciej Bargielski

Wpłynęło do Redakcji 3 grudnia 2002 r.

Abstract

The importance of metadata subsystems grows steadily in modern data-processing applications. Especially in geospatial systems, the metadata is one of the most important components of the system as it provides effective searching and access to data. Many good metadata viewers have been developed so far but only a few more complex tools for creating and updating metadata have been released.

This research describes the problem areas relevant to the design of geospatial metadata editors. It defines the required functionality of a metadata editor as well as it analyses the issues of accessing various metadata sources (flat ASCII/XML files, databases, network services - pic. 1), working in *n*-tier architecture (which affects complexity of the implementation code), distribution and replication of metadata objects and dictionaries (tab. 1), multi-access and the security system (editors can use database management system capabilities to authenticate users and authorize the data access or use their own native security

system). The review of existing metadata creation and edition tools (tab. 2) supplements the theoretical considerations.

Adres:

Jacek FRĄCZEK: Politechnika Śląska, Instytut Informatyki, ul. Akademicka 16, 44-101 Gliwice, Polska, jacekf@zeus.polsl.gliwice.pl .