POLSKIE TOWARZYSTWO INFORMACJI PRZESTRZENNEJ
ROCZNIKI GEOMATYKI 2010 ❍ TOM VIII ❍ ZESZYT 6(42)

METODA SZYBKIEGO WYTWARZANIA DEDYKOWANYCH SYSTEMÓW INFORMACJI PRZESTRZENNEJ BAZUJĄCA NA KONCEPCJI UNIWERSALNEJ KLASY OBIEKTÓW*

THE METHOD OF QUICK PRODUCING OF DEDICATED SPATIAL INFORMATION SYSTEMS BASED ON THE CONCEPT OF GENERALIZED CLASS OF OBJECTS

Edward Kołodziński [1], Grzegorz Betliński[2]

[1] Wyższa Szkoła Informatyki i Ekonomii, TWP w Olsztynie
[2] Wydział Cybernetyki Wojskowej Akademii Technicznej

Słowa kluczowe: inżynieria oprogramowania, model systemu, architektura systemu
Keywords: software engineering, system model, system architecture

Wprowadzenie

Systemy informacji przestrzennej (SIP) bazujące na koncepcji Uniwersalnej Klasy Obiektów są to systemy przetwarzające wyłącznie jeden mechanizm klas obiektów pokazany na rysunku 1. Mechanizm ten składa się z dwóch klas obiektów, stereotypowanej jako <<obiekt>>, tzw. Uniwersalnej Klasy Obiektów i klasy *FObject*.

Zarówno w przypadku Uniwersalnej Klasy Obiektów jak i klasy *FObject* mamy do czynienia z agregacjami całkowitymi. Obiekty odpowiadające elementom składowym tych klas nie mogą samodzielnie istnieć w systemie i dlatego w ich przypadku nie używamy określenia obiekt, lecz komponent obiektowy. Wszędzie tam gdzie nie prowadzi to do nieporozumień mówimy w skrócie komponent, a nie komponent obiektowy i to nie tylko w stosunku do ich wystąpień, ale również do samych klas. Wystąpienia Uniwersalnej Klasy Obiektów określają metadane dla danych przechowywanych w wystąpieniach klasy *FObject*. Każdy obiekt Uniwersalnej Klasy Obiektów definiuje jedną, istotną z punktu widzenia użytkownika klasę, która powinna być w rzeczywistości przetwarzana w projektowanym systemie. Dane zawarte w atrybutach reprezentowanych przez wystąpienia *FAttribute* komponentów *FCom-*

* Praca naukowa finansowana ze środków na naukę w latach 2010-2012 jako projekt badawczy własny nr 0 N516313938.

ponent klasy *FObject* są interpretowane i w odpowiedni sposób przetwarzane na podstawie ich definicji dostarczanych przez odpowiadające im atrybuty odpowiednich komponentów Uniwersalnej Klasy Obiektów.

Komponent Def. Stanowi korzeń drzewa dziedziczenia, zawiera atrybuty, których wartości są identyfikatorami m.in. elementu definicji danych i obiektu bazy danych itp. Wszystkie te atrybuty wraz z działającymi na nich metodami, z założenia wykorzystywane są jedynie wewnętrznie przez system – użytkownik w sposób bezpośredni nie ma do nich żadnego dostępu.

Komponenty standardowe. Znajdują się na następnym, licząc od korzenia, poziomie drzewa dziedziczenia Uniwersalnej Klasy Obiektów. Każdy zawiera odrębne zestawy atrybutów i metod. Zależą one od znaczenia danego komponentu w definicji obiektu. I tak np. standardowy komponent liniowy zawiera takie atrybuty jak: kolor, styl czy grubość linii, zaś metody zabezpieczają takie funkcjonalności jak: dodaj, usuń, przesuń wierzchołek, dodaj, usuń, przesuń odcinek czy całą linię. Zestawy te są predefiniowane. Podczas modelowania ustala się jedynie ich właściwości i wartości domyślne.

Komponenty specjalizowane. Stanowią ostatnią grupą komponentów drzewa dziedziczenia Uniwersalnej Klasy Obiektów. W modelu systemu definiuje się ich własności, krotność, jak również ustala się związane z nimi zestawy dodatkowych, tzw. specjalizowanych atrybutów. Dla każdego takiego atrybutu definiuje się m.in. nazwę (widoczną dla użytkownika), własności, dziedzinę i wartość domyślną. Komponenty specjalizowane dzielimy na pięć niżej opisanych grup.

1. **Komponenty opisowe** (<<własny>>, <<opisowy>>) służą do przechowywania danych alfanumerycznych, które mogą być związane z obiektami danej klasy, przy czym, jak można to zauważyć na diagramie Uniwersalnej Klasy Obiektów, komponenty stereotypowane jako <<opisowy>> mogą być powiązane same ze sobą (powiązanie zwrotne), tym samym umożliwiając modelowanie złożonych, „zagnieżdżonych" struktur danych.

2. **Komponenty graficzne** (<<liniowy>>, <<punktowy>>, <<obszarowy>> i <<napisowy>>) umożliwiają wiązanie z obiektami danych przestrzennych i określają sposób zobrazowania obiektu na mapie (wyznaczają lokalizację przestrzenną i graficzną reprezentację obiektu). Obiekt może nie posiadać danych przestrzennych, a więc może nie obejmować żadnego komponentu graficznego i wówczas nazywany jest obiektem logicznym lub abstrakcyjnym.

3. **Komponenty dokumentów elektronicznych** stereotypowane są jako <<dokument>> i <<dok_graficzny>>. Zapewniają możliwość wiązania z obiektem tzw. dokumentów elektronicznych, a więc różnorodnych plików (tekstowych, graficznych, arkuszy, baz danych, multimedialnych itp.). Pliki te mogą być wytwarzane przez aplikacje zewnętrzne, takie jak np.: MS Word, MS Access, MS Power Point czy AutoCAD, ale również mogą być efektem działania specjalizowanych komponentów programowych wytworzonych specjalnie dla potrzeb budowanego SIP. Pliki elektroniczne identyfikowane przez komponenty <<dok_graficzny>> umożliwiają generowanie w budowanym SIP specjalizowanego zobrazowania graficznego (np. obszary skażenia czy powodzi), które nie może być zrealizowane efektywnie za pośrednictwem komponentów graficznych.

4. **Komponenty relacji** stereotypowane jako: <<połączenie>>, <<podległość>> i/lub <<przynależność>>. Wykorzystywane są w celu definiowania relacji międzyobiektowych, w tym również relacji topologicznych, a więc relacji uwzględniających przestrzenne dane obiektów.

5. **Komponenty specjalizowanych funkcji** – grupa ta obejmuje dwa komponenty. Pierwszy z nich, stereotypowany jako <<fun_ob_zd>>, wykorzystywany jest w celu dostosowywania procedur tworzenia, modyfikowania i usuwania poszczególnych obiektów do specyficznych potrzeb użytkownika. Natomiast drugi, stereotypowany jako <<metoda>>, w celu rozszerzania zestawu metod obiektu o wszystkie takie funkcje, które są niezbędne z punktu widzenia użytkownika budowanego systemu, a które nie są dostarczane przez standardowy system informacji przestrzennej, na bazie którego budowany jest system dedykowany.

Architektura systemu informacji przestrzennej przetwarzającego mechanizmy Uniwersalnej Klasy Obiektów

Dla celów wytwarzania SIP przetwarzających mechanizmy Uniwersalnej Klasy Obiektów przygotowane zostało specjalnie dla tego celu przeznaczone oprogramowanie narzędziowe tzw. Programowe Środowisko Geoba, którego ogólna architektura przedstawiona została na rysunku 2.

Składa się ono z narzędzia typu CASE, tzw. Edytora Modelu Systemu i binarnych komponentów Jądra Systemu i Warstwy Standardowej. Rezultatem działania Edytora Modelu Systemu jest binarny plik Modelu Systemu zawierający wystąpienia Uniwersalnej Klasy Obiektów będące w istocie definicjami klas obiektów, które mają być przetwarzane w projektowanym systemie. Z tego, że w SIP, wytworzonym na bazie PŚ Geoba, przetwarzane są mechanizmy Uniwersalnej Klasy Obiektów, wynika bezpośrednio ogólna architektura takiego systemu. Poszczególnym poziomom drzewa dziedziczenia Uniwersalnej Klasy Obiektów odpowiadają różne warstwy oprogramowania. I tak komponentom typu definicji odpowiada warstwa Jądra Systemu, metody komponentów standardowych znajdują się w Warstwie Standardowej, natomiast Warstwa Specjalizowana obejmuje oprogramowanie związane ze wszystkimi dodatkowymi komponentami, zapewniającymi realizację wszystkich specyficznych funkcji, które są niezbędne dla zabezpieczenia wszystkich tych oczekiwań docelowego użytkownika, które nie są udostępniane przez Warstwę Standardową. Zwykle są to moduły implementujące specjalizowaną (dostosowaną do potrzeb użytkownika) edycję obiektów, czy generujące zestawienia danych w niestandardowych formatach. W wymagających tego przypadkach, komponenty Warstwy Specjalizowanej dostarczają również własny specjalizowany interfejs graficzny. Warstwa Specjalizowana implementuje wszystkie specjalizowane metody, a więc metody zdefiniowane w specjalizowanych komponentach, jak również wszystkie aplikacje własne elektronicznych dokumentów, funkcje odrysowujące dokumenty graficzne i funkcje obsługi zdarzeń. Wytwarzając takie dodatkowe, specjalizowane oprogramowanie, programista korzysta z bardzo szerokiego zestawu funkcji Jądra Systemu, zapewniających możliwość realizacji wszelakich obiektowych operacji edycyjnych, takich jak np.: tworzenie/usuwanie obiektu/komponentu, ustalanie relacji międzyobiektowych, ustalanie wartości poszczególnych atrybutów itp. Na rysunku 3 pokazane są trzy stereotypy wykonywalnych komponentów, wchodzących w skład Warstwy Specjalizowanej:

1) komponenty <<application>> odpowiadają niestandardowym aplikacjom własnym dokumentów elektronicznych, a więc aplikacjom specjalnie zaprojektowanym i wytworzonym dla potrzeb budowanego SIP, np. generującym specjalizowane zestawienia danych lub specjalizowane zobrazowanie graficzne;

2) komponenty <<library>> są bibliotekami obejmującymi: specjalizowane metody, funkcje obsługi zdarzeń i funkcje odrysowujące zawartość dokumentów graficznych;

3) komponenty <<wrapper>> implementują klasy „opakowujące" mechanizm Uniwersalnego Obiektu (<<obiekt>> i FObject) za pomocą własnych definicji atrybutów i metod.

Przykład modelowania SIP w Środowisku Programowym Geoba

Założenia

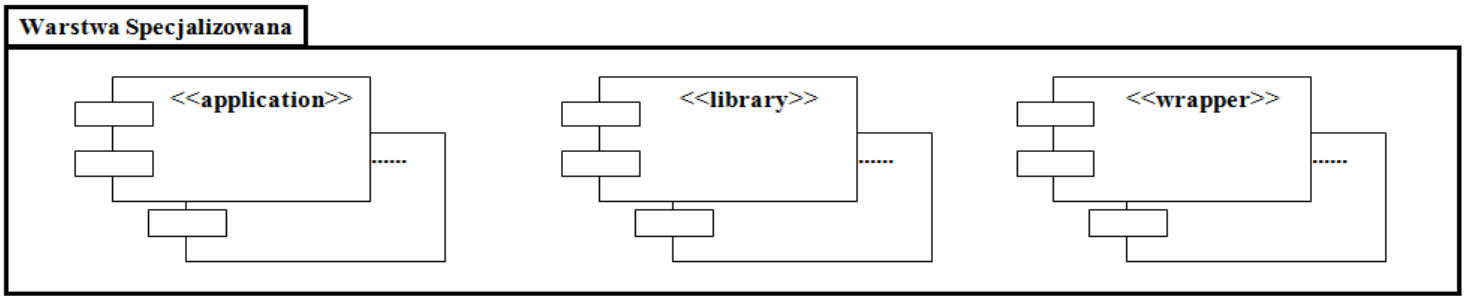
Rozpatrzmy System Obsługi Zdarzeń (SOZ). Będzie to uproszczony, o bardzo ograniczonej funkcjonalności, system reagowania na zdarzenia, których obsługa zwykle leży w gestii takich służb publicznych jak policja, straż pożarna czy ratownictwo medyczne. Charakterystyczną właściwością takich rozszerzonych funkcjonalnie systemów informacji przestrzennej jest konieczność organizowania wielu, zróżnicowanych pod względem dostępu do zasobów i zakresu funkcjonalności, stanowisk funkcyjnych. Przedmiotem działania SOZ będą obiekty opisywane zarówno przez dane przestrzenne, jak i opisowe należące do klas ZGŁOSZENIE, ZDARZENIE i POJAZD, a ich przetwarzaniem będą zajmowały się trzy stanowiska funkcyjne. Na Stanowisku Przyjmowania Zgłoszeń przyjmowane są zgłoszenia, np. telefoniczne, o zaistnieniu jakiegoś zdarzenia wymagającego reakcji ze strony właściwej służby. Zgłoszenia te odwzorowywane są w systemie informatycznym w postaci obiektów ZGŁOSZENIE. Na podstawie obiektu (obiektów) ZGŁOSZENIE, tworzone są w systemie obiekty ZDARZENIE, przy czym stanowiskiem odpowiedzialnym za ich kreację i dalszą obsługę jest Stanowisko Dyspozytora. Podstawowym zadaniem tego stanowiska jest zainicjowanie i nadzór nad przebiegiem akcji obsługi zdarzenia, której pierwszym etapem jest zwykle przydzielenie środków. W przypadku naszego systemu możemy wykorzystywać wyłącznie środki jednego typu, odwzorowane w systemie za pośrednictwem obiektów POJAZD (odpowiadają one np. radiowozom, karetkom itp.). Ewidencjonowaniem tych środków w systemie zajmuje się Stanowisko Nadzoru, którego zakres zadań obejmuje również wprowadzanie do bazy danych systemu wszelkich danych związanych z już zakończonymi akcjami obsługi zdarzeń, jak również np. generowanie bieżących i/lub okresowych zestawień danych dotyczących działania systemu.

Model Systemu Obsługi Zdarzeń

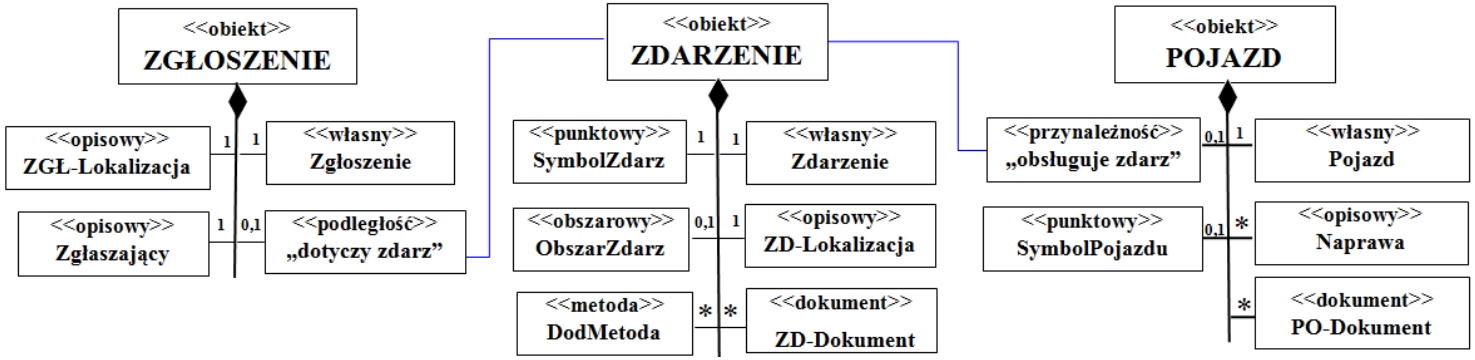
Na rysunku 4 przedstawiono definicje klas obiektów Systemu Obsługi Zdarzeń.

Dla obiektów **ZGŁOSZENIE** zdefiniowano cztery komponenty. W komponencie własnym, o nazwie identycznej jak nazwa klasy, zdefiniowane są atrybuty opisujące przedmiot zgłoszenia, a więc kiedy nastąpiło zgłoszenie, jakiego zdarzenia zgłoszenie to dotyczy (przestępstwo, pożar, uwolnienie toksycznej substancji itp.), jaka jest skala zdarzenia, czy są ofiary, itd.

Definicja komponentu *Zgłaszający* będzie zawierała definicje atrybutów umożliwiających identyfikację osoby zgłaszającej zajście zdarzenia, definicja komponentu *ZGŁ-Lokalizacja* – definicje atrybutów wykorzystywanych dla opisu lokalizacji miejsca zdarzenia, natomiast definicja komponentu relacji *dotyczy zdarz* – definicje standardowych atrybutów służących do identyfikacji obiektu nadrzędnego w tej relacji (z klasy ZDARZENIE). Dla obiektów klasy ZGŁOSZENIE przewidziany został wyłącznie jeden stan, a zatem obiekt w tym stanie może



Rys. 3. Typy komponentów Warstwy Specjalizowanej

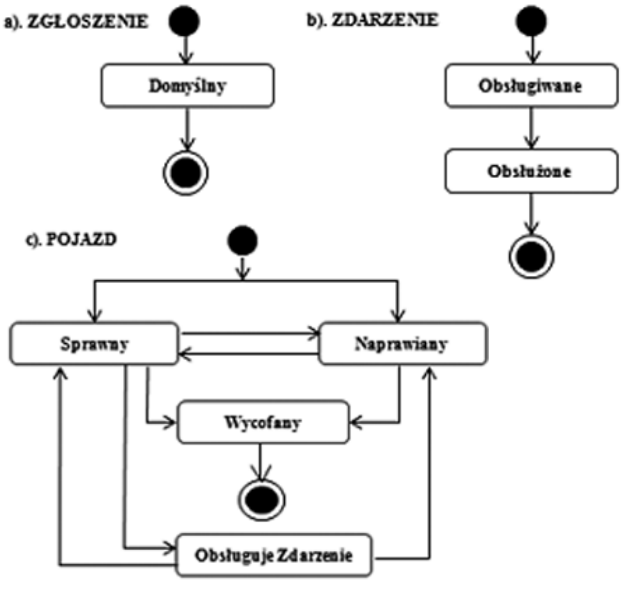


Rys. 4. SOZ – klasy obiektów i ich powiązania

zawierać wszystkie komponenty, z których żaden nie jest replikowalny (wielokrotny) i wszystkie, z wyjątkiem komponentu relacji, są komponentami obligatoryjnymi.

Dla obiektów **ZDARZENIE** zdefiniowano sześć komponentów. Komponent własny *Zdarzenie* i komponent opisowy *ZD-Lokalizacja* służą do tego samego celu co komponenty *Zgłoszenie* i *ZGŁ-Lokalizacja* w klasie ZGŁOSZENIE, a zatem zawierają analogiczne zestawy definicji atrybutów. W komponencie własnym *Zdarzenie* dodatkowo są zdefiniowane atrybuty służące do opisania kiedy została rozpoczęta akcja obsługująca zdarzenie, jak również kiedy i z jakim rezultatem została zakończona. Komponent graficzny *SymbolZdarz* opisuje reprezentację graficzną obiektu ZDARZENIE, a jego punkt wstawienia na mapie determinowany jest wartościami aktualnymi atrybutów komponentu *ZD-Lokalizacja*. Drugi komponent graficzny *ObszarZdarz* jest komponentem opcjonalnym, definującym obszar skutków zdarzenia. Kolejnym komponentem klasy ZDARZENIE jest replikowalny i opcjonalny komponent *ZD-Dokument*, na podstawie którego do istniejących obiektów tej klasy będą dołączane różnorakie pliki dotyczące zdarzenia, takie jak np. zdjęcia z miejsca zdarzenia, protokoły przesłuchań świadków itp. Ostatnim komponentem zdefiniowanym w klasie ZDARZENIE jest replikowalny i opcjonalny komponent *DodMetoda*, umożliwiający rozszerzanie zestawu metod istniejących obiektów tej klasy o dodatkowe specjalizowane metody. Dla obiektów klasy ZDARZENIE przewidziane zostały dwa stany (rys. 5): 1) obsługiwane, 2) obsłużone.

Ostatnią klasą zdefiniowaną w SOZ jest klasa obiektów **POJAZD**. W komponencie własnym zdefiniowane zostały atrybuty służące do identyfikacji obiektu, a więc takie atrybuty jak numer rejestracyjny, marka, typ, numer silnika, numer podwozia itp. Wszystkie pozostałe komponenty są opcjonalne. Replikowalny komponent *PO-Dokument* umożliwia wiązanie z obiektami klasy różnorakich plików, takich jak np. zdjęcia, dane taktyczno-techniczne, instrukcje itp., natomiast drugi replikowalny komponent *Naprawa* przewidziany został do przechowywania danych opisujących wykonywane naprawy. Ostatnim zdefiniowanym w klasie POJAZD komponentem jest opcjonalny i nie replikowalny komponent relacji *obsługuje zdarz*, zawierający definicje standardowych atrybutów służących do identyfikacji obiektu nadrzędnego w tej relacji (z klasy ZDARZENIE). Dla obiektów klasy POJAZD przewidziane zostały cztery stany: 1) sprawny, 2) obsługuje zdarzenie, 3) naprawiany, 4) wycofany z eksploatacji. Komponenty *Pojazd, PO-Dokument* i *Naprawa* występują we wszystkich stanach, natomiast komponent *Obsługuje zdarzenie* wyłącznie w ww. stanie 2) obsługuje zdarzenie.



Rys. 5. Stany obiektów przetwarzanych w SOZ

Dostosowywanie Systemu Obsługi Zdarzeń do potrzeb użytkownika

W chwili zdefiniowania Modelu Systemu Obsługi Zdarzeń, parametryzującego w pełni działanie Jądra Systemu i Warstwy Standardowej, w istocie dysponujemy już również działającym prototypem systemu. W związku z tym możemy rozpocząć prace wdrożeniowe, które w początkowym etapie sprowadzają się głównie do wypełnienia bazy danych, jednocześnie rozbudowując oprogramowanie i Model Systemu Obsługi Zdarzeń, w celu zapewnienia wszystkim jego użytkownikom (na wszystkich stanowiskach funkcyjnych) wszystkich niezbędnych im funkcji. Jednakże użytkownicy takiego prototypu systemu mogą podczas swojej pracy wykorzystywać jedynie wbudowane, standardowe mechanizmy programowe Programowego Środowiska Geoba, które z założenia są mechanizmami uniwersalnymi, i w związku z tym nie zawsze w pełni wychodzą naprzeciw oczekiwaniom docelowych użytkowników. Dobrymi przykładami, pochodzącymi z omawianego w artykule SOZ, mogą być niżej wymienione operacje, za których realizację jest w tym systemie odpowiedzialny użytkownik – dyspozytor:

- analiza zgłoszeń,
- identyfikacja zdarzenia,
- przydział środków do obsługi nowego zdarzenia,
- zakończenie obsługi zdarzenia.

Dyspozytor musi dokonać analizy zgłoszeń, chociażby w celu wypracowania decyzji, czy nowe zgłoszenie dotyczy jakiegoś już obsługiwanego zdarzenia, czy też dotyczy całkowicie nowego zdarzenia, które wymaga utworzenia nowego obiektu ZDARZENIE, i co za tym idzie, przydzielenia do jego obsługi określonych środków (pojazdów). Wykorzystując wyłącznie standardowe funkcje Warstwy Standardowej dyspozytor musiałby wygenerować zestawienie danych dotyczące aktualnie obsługiwanych zdarzeń, po czym, na podstawie danych opisujących same zdarzenia i ich lokalizację, podjąć taką decyzję, a więc bądź powiązać nowy obiekt ZGŁOSZENIE z którymś spośród już istniejących obiektów ZDARZENIE za pośrednictwem relacji *dotyczy zdarz*, bądź najpierw utworzyć nowy obiekt ZDARZENIE, a następnie utworzyć tę relację międzyobiektową. Narzucającym się rozwiązaniem, wspomagającym dyspozytora w tym aspekcie jego pracy, jest oprogramowanie funkcji, która automatycznie wypracuje propozycję powiązania nowych obiektów ZGŁOSZENIE z istniejącymi obiektami ZDARZENIE. Propozycja ta, opierająca się na porównaniu lokalizacji i opisu zgłoszeń i zdarzeń, powinna być przedstawiona w odpowiedniej formie graficznej do zatwierdzenia dyspozytorowi, który wówczas może podjąć właściwą decyzję, w oparciu o odpowiednio zaprezentowane dane przestrzenne i opisowe. Wówczas omawiana funkcja wspomagająca, przed zakończeniem swojego działania, automatycznie utworzyłaby (lub nie) relację *dotyczy zdarz* pomiędzy zaakceptowanymi przez dyspozytora obiektami ZGŁOSZENIE i ZDARZENIE.

Zgłoszenia mogą dotyczyć zdarzenia, którego identyfikacja, prognozowanie dalszego przebiegu i ewentualnych skutków nie muszą być oczywiste. Przykładem mogą być ataki terrorystyczne powodujące zdarzenia, w rezultacie których następuje uwolnienie toksycznej substancji chemicznej/radioaktywnej w powietrzu lub wodzie w obszarze skupisk ludności. W takim przypadku osoba funkcyjna powinna mieć zapewniony dostęp do specjalistycznej wiedzy i danych, które zazwyczaj pozostają w gestii właściwych systemów eksperckich, potrafiących na podstawie charakterystyki zdarzenia, w tym m.in. zarejestrowanych objawów

zatruć, zidentyfikować niebezpieczną substancję i ustalić właściwe procedury postępowania, mające na celu jej neutralizację i ratowanie zdrowia i życia ludności. Niewątpliwie najlepiej byłoby, aby właściwa funkcja takiego systemu eksperckiego mogła być aktywowana bezpośrednio z interfejsu dedykowanego SIP, bez konieczności ponownego wprowadzania danych, które istnieją już w systemie, w postaci wartości odpowiednich atrybutów powiązanych ze sobą obiektów z klas ZGŁOSZENIE i ZDARZENIE.

W chwili utworzenia nowego obiektu ZDARZENIE, dyspozytor powinien: 1) przydzielić środki do jego obsługi, 2) ponownie, wykorzystując wyłącznie funkcje udostępniane przez Warstwę Standardową, wygenerować zestawienie danych dotyczące pojazdów w stanie 'sprawny', 3) przeanalizować, które spośród nich powinny być przydzielone do obsługi zdarzenia, 4) na koniec, wybrane obiekty POJAZD powiązać relacją 'obsługuje zdarzenie' z obiektem ZDARZENIE. Funkcja wspomagająca dyspozytora w tym zakresie powinna wygenerować okno dialogowe, w którym dyspozytor miałby, analogicznie jak uprzednio, przedstawioną propozycję takiego przydziału, wypracowaną automatycznie na podstawie opisu zdarzenia (rodzaj, skala, liczba ofiar itp.) i danych technicznych dostępnych (w stanie 'sprawny') pojazdów. Oczywiście, ostateczną decyzję o przydziale podejmuje dyspozytor, a jej rezultatem powinno być automatyczne utworzenie wszystkich niezbędnych relacji 'obsługuje zdarzenie'.

Po zakończeniu obsługi zdarzenia odpowiadający mu obiekt ZDARZENIE powinien przejść ze stanu 'obsługiwane' w stan 'obsłużone', natomiast odpowiadające obsługującym zdarzenie pojazdom, obiekty POJAZD ze stanu 'obsługuje zdarzenie' w stan 'sprawny'. Oczywiście dyspozytor może te operacje zrealizować sekwencyjnie za pomocą standardowego interfejsu, jednak niewątpliwie ułatwi mu pracę dodatkowa funkcja, która uaktywniona w chwili zmiany stanu obiektu ZDARZENIE, automatycznie zmieni również stany wszystkich powiązanych z tym obiektem relacją 'obsługuje zdarzenie' obiektów POJAZD.

Takie dodatkowe funkcje, wspomagające i ułatwiające pracę użytkownikom poszczególnych stanowisk funkcyjnych, muszą być oczywiście osobno implementowane i następnie zintegrowane z oprogramowaniem całego systemu.

Dodatkowe metody w klasie ZDARZENIA

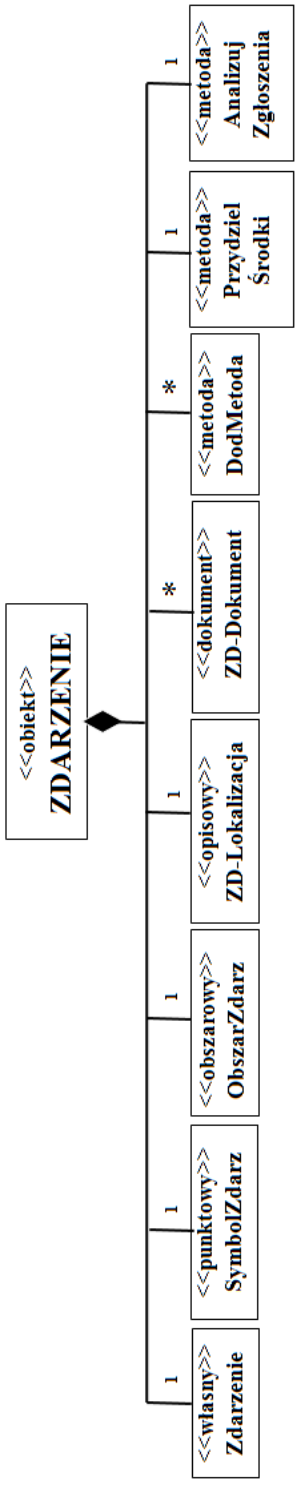
Specjalizowana funkcja (specjalizowana metoda) jest to funkcja, która może być uaktywniona na rzecz istniejącego obiektu danej klasy (jednym z jej parametrów wejściowych jest ten obiekt), natomiast specjalizowana funkcja (metoda) statyczna klasy jest to funkcja, której uaktywnienie nie jest uzależnione od istnienia jakiegokolwiek obiektu tej klasy. Zarówno metody jak i metody statyczne definiuje się w Modelu Systemu w postaci komponentów typu <<metoda>>. Atrybuty komponentu metoda wykorzystywane są do identyfikacji funkcji, która ma być uaktywniona, a więc zawierają takie dane jak: nazwa funkcji; nazwa biblioteki, z której funkcja pochodzi; sposób aktywowania (synchroniczny, asynchroniczny) itp. Warstwa Standardowa zapewnia użytkownikowi właściwy interfejs graficzny, umożliwiający aktywowanie metod zarówno standardowych, jak i specjalizowanych. Po wskazaniu elementu graficznego, reprezentującego dane przestrzenne obiektu posiadającego komponenty metod, automatycznie generowane jest kontekstowe menu zawierające dla każdego takiego komponentu odrębną pozycję, której uaktywnienie jest równoważne z uaktywnieniem funkcji (metody). W przypadku metod statycznych analogiczne menu kontekstowe generowane jest po wskazaniu każdego elementu odpowiadającego klasie, w której zostały zdefiniowane metody statyczne. Zatem, w celu zapewnienia użytkownikowi możliwości realizacji metody, należy jedynie wytworzyć jej kod programowy i odpowiednio zmodyfikować Model Systemu.

Zasadniczą różnicą pomiędzy tradycyjnym podejściem obiektowym a podejściem bazującym na koncepcji Uniwersalnej Klasy Obiektów jest to, że w koncepcji tej metody przejawiają dwoistą naturę, gdyż z punktu widzenia operacji edycyjnych obiektu, metody są w istocie danymi – wystąpieniami klasy *FComponent*, które można utworzyć, zmodyfikować i/lub usunąć. W rezultacie, w różnych okresach swojego istnienia, obiekt może posiadać różne zestawy metod, przy czym mogą z nimi być związane różne wartości parametrów, wpływające na przebieg ich realizacji.

Kolejną zasadniczą różnicą pomiędzy tradycyjnym podejściem obiektowym a podejściem bazującym na koncepcji Uniwersalnej Klasy Obiektów jest to, że w bazujących na tej koncepcji systemach informatycznych możliwe jest zastąpienie kodu funkcji związanej z komponentem <<metoda>> również podczas normalnej eksploatacji systemu. Jeżeli w Modelu Systemu standardowe atrybuty: nazwaFunkcji, nazwaModułu i typModułu komponentu <<metoda>> zostaną zdefiniowane jako edytowalne, to użytkownik będzie mógł podczas edytowania komponentu <<metoda>> ustalić ich całkowicie nowe wartości, identyfikujące całkowicie inną funkcję, np. z całkowicie innego modułu programowego, niż to zostało przewidziane w dotychczasowej wersji Modelu Systemu. Oznacza to, że funkcjonalność eksploatowanego już SIP może być w znaczącym stopniu zmieniona przez samego użytkownika, bez konieczności angażowania wytwórcy systemu.

W naszym przykładowym SOZ, mechanizm metod może być w naturalny niejako sposób wykorzystany, w celu implementacji wyżej opisanych dodatkowych funkcji analiza *zgłoszeń* i *przydział środków* wspomagających pracę dyspozytora.

Pierwsza z nich powinna, w interakcji z użytkownikiem, ewentualnie utworzyć nowy obiekt ZDARZENIE i powiązać go relacją *dotyczy zdarzenia* z jednym lub kilkoma obiektami ZGŁOSZENIE. Tak więc, funkcja ta nie będzie uaktywniana na rzecz jakiegoś istniejącego obiektu ZDARZENIE, a zatem powinna być to metoda statyczna klasy obiektów ZDARZENIE. Z kolei druga funkcja *przydział środków* dotyczy konkretnego, istniejącego już obiektu ZDARZENIE, a zatem należy ją implementować jako metodę (nie statyczną), gdyż wówczas Warstwa Standardowa uaktywni ją, przekazując jako jeden z parametrów wejściowych obiekt, na rzecz którego metoda została wywołana. Na rysunku 6 przedstawiono zmodyfikowaną o takie dodatkowe komponenty metod definicję obiektów klasy ZDARZENIE.



Rys. 6. Definicja klasy ZDARZENIE rozszerzona o dodatkowe komponenty metod

Funkcje obsługi zdarzeń w klasie ZDARZENIA

W SIP bazujących na koncepcji Uniwersalnej Klasy Obiektów, niezależnie od klasy obiektu, zawsze wykorzystywane są te same uogólnione funkcje: tworzenia i destrukcji obiektów/komponentów; zmiany stanu obiektów; zmiany wartości atrybutu itp. Oznacza to, że w tych funkcjach mogą zostać zidentyfikowane istotne punkty, po osiągnięciu których mogą być uaktywniane inne funkcje, np. zaimplementowane w zewnętrznych komponentach programowych. W skrajnym przypadku funkcje te – służące do dostosowania funkcji standardowych, dostarczonych przez producenta SIP, do specyficznych potrzeb użytkownika – mogą być przez użytkownika projektowane i implementowane. W terminologii Uniwersalnej Klasy Obiektów, osiągnięcie przez metodę obiektu punktu, w którym będzie ewentualnie wywoływana funkcja zewnętrzna, określane jest jako „zajście zdarzenia w życiu obiektu", natomiast samą tę zewnętrzną funkcję określa się mianem „funkcji obsługi zdarzenia obiektowego". Funkcje obsługi zdarzeń definiowane są w Modelu Systemu za pośrednictwem specjalizowanych komponentów stereotypowanych jako <<fun_ob_zd>>.

W naszym przykładowym SOZ mechanizm obsługi zdarzeń może być wykorzystany w celu implementacji wcześniej opisanej dodatkowej funkcji, wspomagającej pracę dyspozytora w chwili, gdy zdecyduje on, że obsługa jakiegoś zdarzenia została już całkowicie zakończona. Wówczas powinien zmienić stan obiektu ZDARZENIE z 'obsługiwane' na 'obsłużone', jak również stan wszystkich tych obiektów POJAZD, które były związane relacją 'obsługuje zdarzenie' z tym obiektem ZDARZENIE, ze stanu 'obsługuje zdarzenie' w stan 'sprawny'. Tak jak to było wcześniej powiedziane, dyspozytor mógłby wykorzystać dla osiągnięcia takiego celu standardowy interfejs Warstwy Standardowej, jednak jego praca zostanie znacznie uproszczona, jeśli zdecydujemy się na rozszerzenie oprogramowania Warstwy Specjalizowanej o dodatkową funkcję, w tym przypadku będzie to funkcja obsługi zdarzenia „zmiana stanu obiektu", która zostanie automatycznie wykonana w chwili, gdy dyspozytor aktywuje, w zwykły sposób, za pośrednictwem standardowego interfejsu, operację zmiany stanu obiektu z klasy ZDARZENIE. Funkcja ta, podczas działania, powinna jedynie zmienić stan obiektów POJAZD na „sprawny" – komponenty relacji 'obsługuje zdarzenie' nie muszą być przez tę funkcję usuwane, gdyż występują one wyłącznie w stanie 'obsługuje zdarzenie' obiektów POJAZD, a zatem z chwilą „wyjścia" obiektów z tego stanu, zostaną automatycznie usunięte.

Możliwe jest również wykorzystanie zdarzenia obiektowego występującego w chwili utworzenia obiektu ZDARZENIE. Jeżeli wartość atrybutu określającego typ zdarzenia identyfikuje zdarzenie uwolnienia niebezpiecznej substancji, to funkcja obsługi zdarzenia kreacji nowego obiektu może rozszerzyć zestaw metod takiego obiektu o dodatkową metodę, przez utworzenie wystąpienia komponentu *DodMetoda*, z odpowiednio wypełnionymi jego standardowymi atrybutami: nazwaFunkcji, nazwaModułu i typModułu, tak by identyfikowały one funkcję, w odpowiedni sposób aktywującą właściwy system ekspercki. Taką dodatkową metodę użytkownik może uaktywnić za pośrednictwem automatycznie generowanego przez Warstwę Standardową kontekstowego menu obiektu. Po jej uaktywnieniu, do systemu eksperckiego przekazywane są właściwe dane wejściowe, pozyskiwane z obiektu, na rzecz którego metoda została uaktywniona. Ostatecznie, po zakończeniu działania systemu eksperckiego, wypracowane przez ten system rezultaty wiązane są z wejściowym obiektem, np. za pośrednictwem dodatkowych komponentów *ZD-Dokument*, zawierających specyfikacje procedur postępowania, odpowiednich dla zaistniałego zdarzenia.

Podsumowanie

Fundamentalnym założeniem, leżącym u podstaw technologii wytwarzania systemów informatycznych w Środowisku Programowym Geoba, była konieczność zapewnienia takich mechanizmów programowych, które umożliwiają przyrostową rozbudowę, zarówno w aspekcie samych danych przetwarzanych w budowanym (rozbudowywanym) systemie, jak i w aspekcie jego funkcjonalności, przy czym każdy taki przyrost powinien „być realizowalny bezkonfliktowo" nie tylko podczas wdrożenia, ale również podczas eksploatacji systemu. Przykłady takich przyrostów funkcjonalności można zauważyć w naszym przykładowym Systemie Obsługi Zdarzeń. Były nimi w kolejności: prototyp, funkcja „analiza zgłoszeń", funkcja „przydział środków" i na koniec funkcja obsługi zdarzenia „zmiana stanu ZDARZENIA". Kolejnymi mogłyby być np. „wyznaczenie trasy dojazdu do miejsca zdarzenia", czy w przypadku takich służb jak policja lub straż graniczna, „wyznaczenie blokad w zadanej odległości od miejsca zdarzenia". Jednak dla implementacji tych ostatnich funkcji, w bazie danych naszego przykładowego SOZ musiałyby znajdować się dane opisowe i przestrzenne sieci drogowej, a w przypadku „wyznaczania blokad" zapewne i kolejowej. Zatem musimy rozszerzyć Model Systemu o definicje obiektów należących do tych sieci, zmodyfikować schemat bazy danych, a następnie wprowadzić rzeczywiste dane. I nic nie stoi na przeszkodzie byśmy zrealizowali nasze zamierzenia w trakcie eksploatacji SOZ. Za pomocą Edytora Modelu Systemu modyfikujemy Model Systemu i generujemy skrypt rozszerzający schemat bazy danych. Zmiana Modelu Systemu i schematu bazy danych jest niewidoczna dla pracujących w systemie użytkowników, gdyż Jądro Systemu automatycznie nadaje im „zerowe" uprawnienia w stosunku do wszystkich nowych definicji. Zatem, by móc ewidencjonować np. sieć drogową, powinniśmy utworzyć w systemie nowego użytkownika, o nazwie np. Operator Sieci Drogowej i nadać mu niezerowe uprawnienia jedynie w stosunku do związanych z siecią drogową definicji Modelu Systemu. Tym samym utworzyliśmy nowe, dodatkowe stanowisko funkcyjne w SOZ, na którym bezkonfliktowo z pozostałymi stanowiskami systemu, możemy ewidencjonować potrzebne dane. Jednocześnie z wprowadzaniem tych danych, możemy implementować, a następnie testować wyżej wspomniane dodatkowe funkcje. Na koniec, modyfikujemy definicję klasy ZDARZENIE, specyfikując w niej metody i/lub funkcje obsługi zdarzeń, zapewniając tym samym możliwość ich aktywacji z interfejsu graficznego systemu. Na przykład funkcja „wyznaczania blokad" mogłaby być implementowana jako metoda klasy ZDARZENIA, natomiast „wyznaczenie trasy dojazdu" jako funkcja obsługi zdarzenia „utworzenie obiektu" w tej samej klasie.

Literatura

Kołodziński E., Betliński G., 2004: Model systemu GIS w środowisku programowym GEOBA. Materiały XII Konferencji Naukowej „Automatyzacja dowodzenia", Gdynia-Jurata, 2-4 czerwca.

Kołodziński E., Betliński G., 2004: Modelowanie zależności topologicznych za pośrednictwem międzyobiektowych związków w programowym środowisku GEOBA. *Roczniki Geomatyki* t. 2, z. 2, 121-128. PTIP, Warszawa.

Kołodziński E., Betliński G., 2004: Resource Multi-Usage During GIS System Development Using GEOBA Programming Environment As Example Of EiZZT MON. 6th NATO Regional Conference On Military Communication And Information Systems. Zegrze, 10-12 October. 135-141.

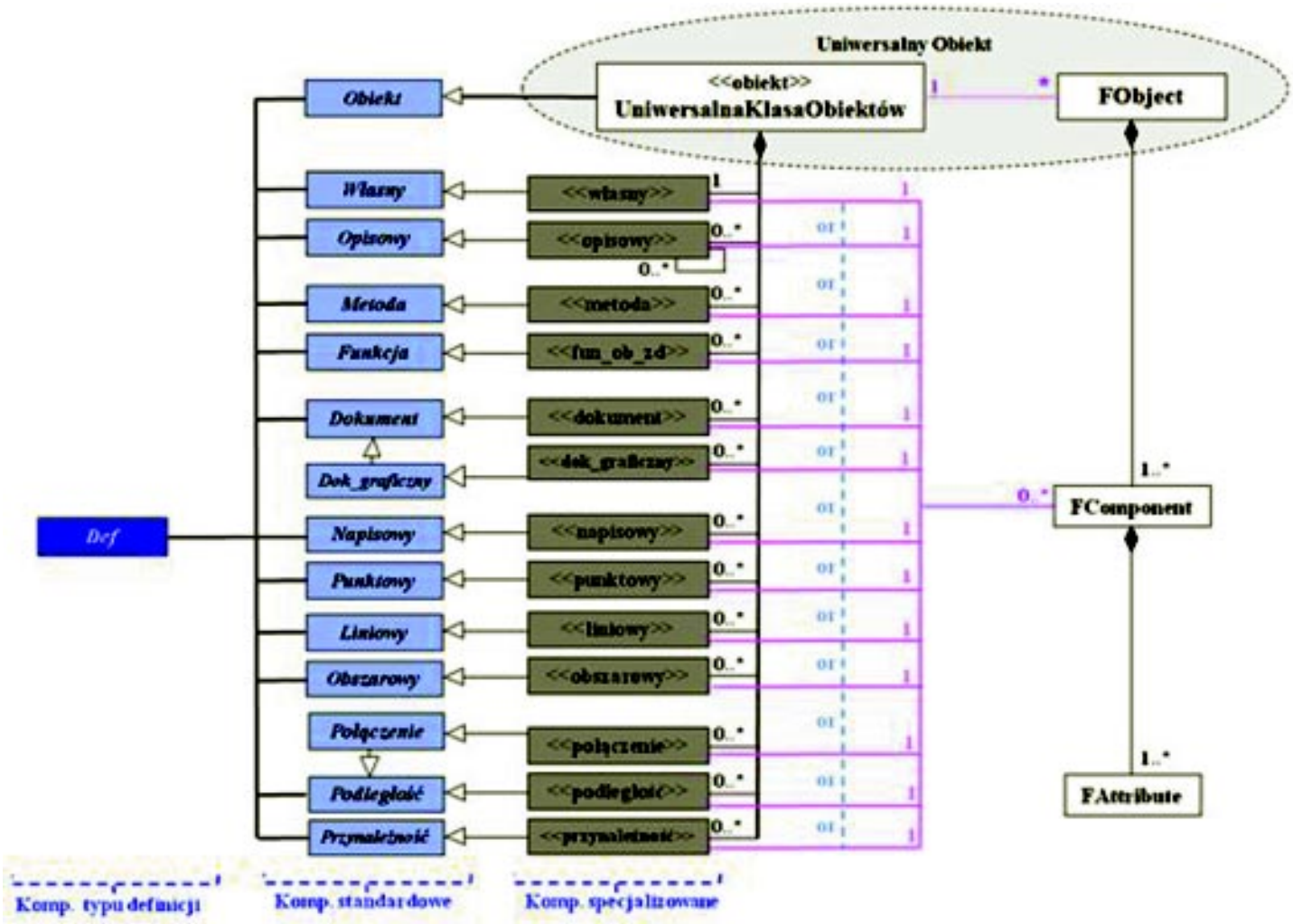
Kołodziński E., Betliński G., 2007: Modelowanie systemów informacji przestrzennej z zastosowaniem Uogólnionej Klasy Obiektów, *Roczniki Geomatyki* t. 5, z. 2, 39-49. PTIP, Warszawa.

Abstract

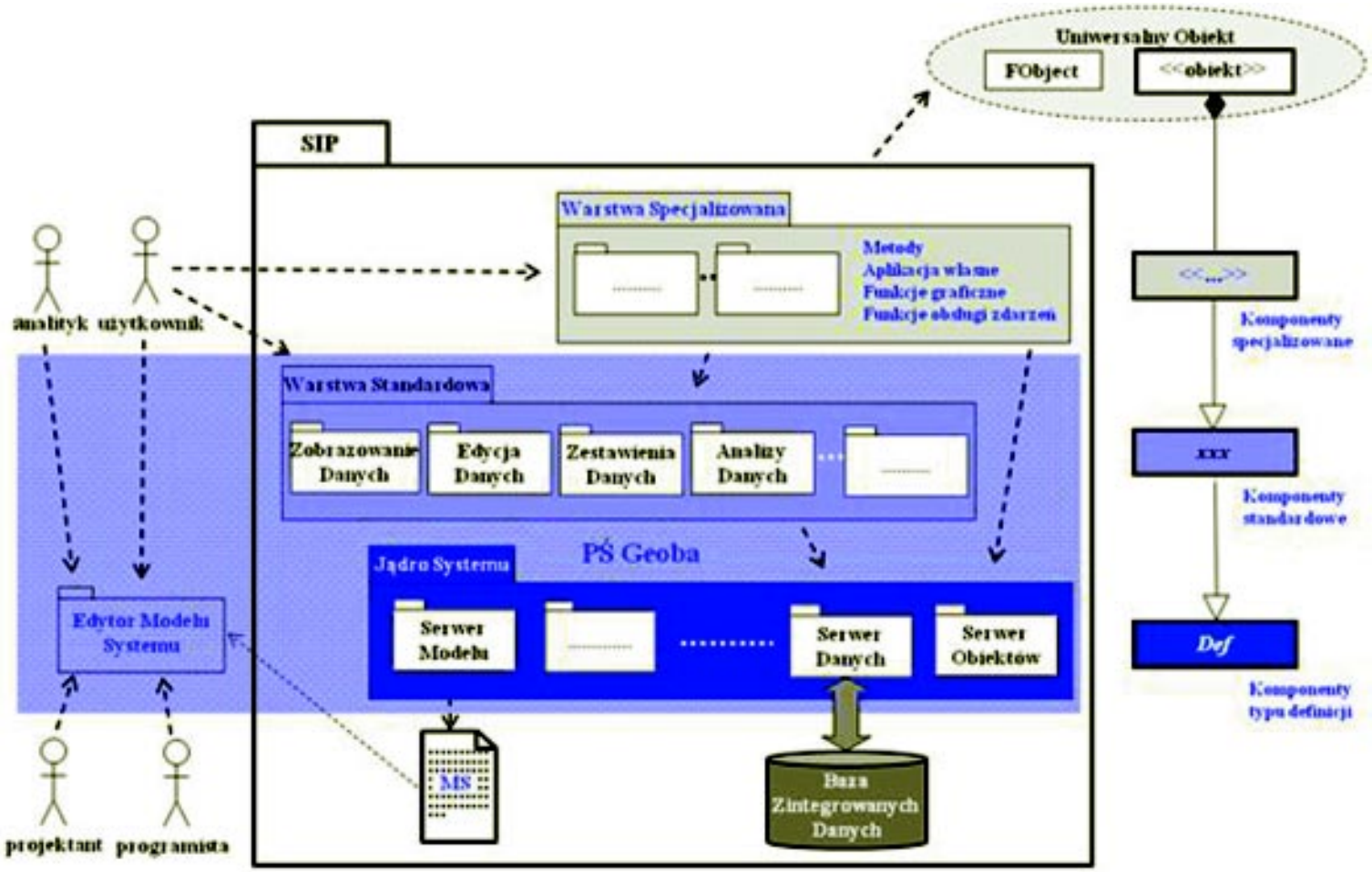
The paper describes an approach to adapting functionality of spatial information system, based on the concept of generalized class of objects, to the needs of the end user. Of essential importance in this approach are function components designed for defining specialized object methods as well as event service functions or own applications of electronic documents. These components make it possible to connect several specialized functions with individual objects. When required, such functions can be also realized by external systems, for example expert systems. There are two basic differences between the traditional object approach and the approach based on the concept of generalized class of objects. The first difference consists in the fact that in this concept object methods manifest dual nature because from the point of editing operations these methods are in fact normal data which may be created, updated and/or removed. The second basic difference consists in the fact that in the spatial information systems based on this concept it is possible to replace the code of the function tied with the component also during normal operation of the system. Thus, while editing such a component it is possible to determine completely new values entirely identifying another function, e.g. from other program module than initially foreseen. It means that the functionality of already exploited spatial information system can be changed by the end user without the need to engage the producer of the system.

dr hab. inż. Edward Kołodziński, prof. WSIiE
ekolodzinski@wp.pl

dr Grzegorz Betliński
gbetl@wp.pl



Rys. 1. Mechanizm Uniwersalnej Klasy Obiektów



Rys. 2. Ogólna architektura SIP przetwarzającego mechanizmy Uniwersalnej Klasy Obiektów