

ANALIZA EFEKTYWNOŚCI WYTWARZANIA OPROGRAMOWANIA UŻYTKOWEGO GIS W ŚRODOWISKU GEOBA

EFFICIENCY ANALYSIS OF CREATION OF GIS APPLICATION PROGRAMS IN GEOBA ENVIRONMENT

Edward Kołodziński¹, Ireneusz Wyglądała²

¹Regionalne Centrum Informatyczne, Wydział Matematyki i Informatyki
Uniwersytet Warmińsko-Mazurski w Olsztynie

²Przedsiębiorstwo Projektowo-Wdrożeniowe INFOKART S.A.

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

Wprowadzenie

Posiadanie wiedzy oraz racjonalne jej wykorzystanie jest podstawowym czynnikiem rozwoju cywilizacyjnego. Człowiek do swoich działań potrzebuje wiedzieć nie tylko, co i dlaczego ma miejsce ale również kiedy i gdzie. Dawniej precyzyjne wskazywanie miejsca naręczało poważne trudności. Dynamiczny rozwój informatyki, a zwłaszcza w zakresie systemów klasy GIS, w coraz większym stopniu eliminuje te trudności.

Systemy informatyczne klasy GIS charakteryzują się z reguły ponadprzeciętną złożonością. Dotyczy to oprogramowania, sprzętu, a przede wszystkim bazy danych. Wytwarzanie i wdrażanie ich rodzi wiele problemów realizacyjnych, które można przezwyciężyć poprzez dobór właściwych modeli wytwarzania i programowych narzędzi do tworzenia systemów. Wyjściem na przeciw tym problemom jest opracowanie uniwersalnego szkieletu oprogramowania, który może być wielokrotnie wykorzystywany podczas wytwarzania wielu dedykowanych systemów, dla różnorodnych zastosowań. Dla potrzeb konkretnego zastosowania, szkielet ten może być obudowywany dodatkowymi modułami zapewniającymi specyficzne funkcje, niezbędne z punktu widzenia docelowego użytkownika. Takie podejście zostało zastosowane przy opracowywaniu krajowego środowiska programowego o nazwie GEOBA do wytwarzania oprogramowania użytkowego systemów informatycznych klasy GIS. Jednym z czynników decydującym o jego upowszechnieniu jest efektywność wytwarzania i wdrażania oprogramowania użytkowego GIS wykonanego z zastosowaniem tego środowiska.

Problem ten jest przedmiotem niniejszego artykułu.

Analiza podstawowych modeli wytwarzania oprogramowania użytkowego systemów informatycznych

System informatyczny (SI) – najkrócej można zdefiniować jako zbiór powiązanych ze sobą elementów przetwarzających dane przy użyciu techniki komputerowej. Na system informatyczny składają się: sprzęt, oprogramowanie, baza danych, ludzie oraz elementy organizacyjne. Szczególnie w przypadku systemów klasy GIS konieczne jest wyodrębnienie **bazy danych** jako równoprawnego składnika systemu. Każdy z nich jest równie ważny dla poprawnego funkcjonowania całości, jednak uwaga wykonawcy SI z reguły skupiona jest na oprogramowaniu użytkowym jako najbardziej pracochłonnej i podstawowej dla niego części przedsięwzięcia. Z kolei użytkownik GIS traktuje oprogramowanie użytkowe jako jedno z narzędzi do zarządzania i wykorzystywania zawartości bazy danych, która ma dla niego rolę pierwszoplanową. Sukces przedsięwzięcia informatycznego zorientowanego na GIS zależy więc, w głównej mierze, od właściwości użytkowych jego dwóch składowych: bazy danych i oprogramowania użytkowego.

W procesie tworzenia oprogramowania użytkowego GIS można wyróżnić cztery podstawowe fazy:

- określenie pożądanej funkcjonalności – wymagań,
- projektowanie i tworzenie oprogramowania na podstawie wymagań,
- testowanie oprogramowania na zgodność z wymaganiami
- doskonalenie właściwości funkcjonalnych oprogramowania podczas jego użytkowania.

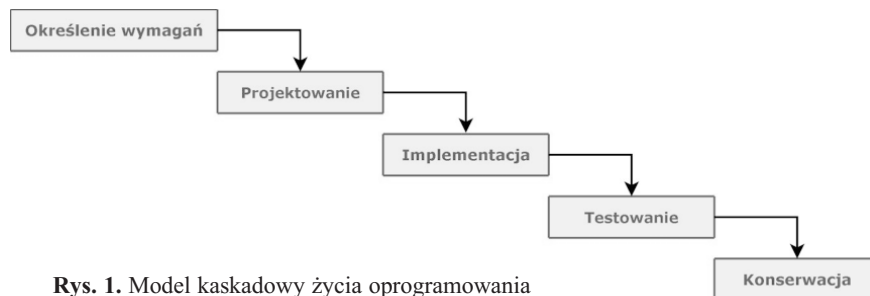
Dalszy podział powyższych faz na bardziej szczegółowe części zadaniowe zależy zarówno od przyjętej metodologii postępowania – **modelu wytwarzania** – jak i charakteru tworzonego oprogramowania. Stosowana metodologia jest determinowana przez wiedzę i doświadczenie wykonawcy, a przede wszystkim przez środowisko programowe i narzędzia wspomagające zastosowane do jego wytwarzania. W literaturze, np. (Jaszkiewicz, 1997; Somerville, 2003), wyróżnia się następujące podstawowe modele wytwarzania oprogramowania SI, które w pełni mogą być odniesione do GIS:

- model kaskadowy,
- metodyka HIPO,
- wytwarzanie oprogramowania kierowane dokumentami,
- prototypowanie,
- model przyrostowy,
- model ewolucyjny,
- model spiralny.

Model kaskadowy jest klasycznym i najprostszym podejściem do wytwarzania oprogramowania. Wprowadza on systematyczne, sekwencyjne podejście do tworzenia oprogramowania użytkowego rozpoczynając od określenia wymagań, a kończąc na opiece nad gotowym produktem (rys. 1).

Jest to idealne podejście w przypadku, gdy wymagania użytkownika są jasno sprecyzowane i w pełni zrozumiałe. Praca podzielona jest w przejrzysty sposób na niezależne, kolejne etapy, co znakomicie ułatwia zarządzanie zasobami ludzkimi.

Metodyka HIPO (*Hierarchy-Input-Process-Output*) jest podejściem hierarchicznym do realizacji SI i polega na integracji prac projektowych i programowych. Cały proces dzielony jest na procesy składowe, a każdy z nich jest odpowiednio dokumentowany. Wymusza to kaskadowy



Rys. 1. Model kaskadowy życia oprogramowania

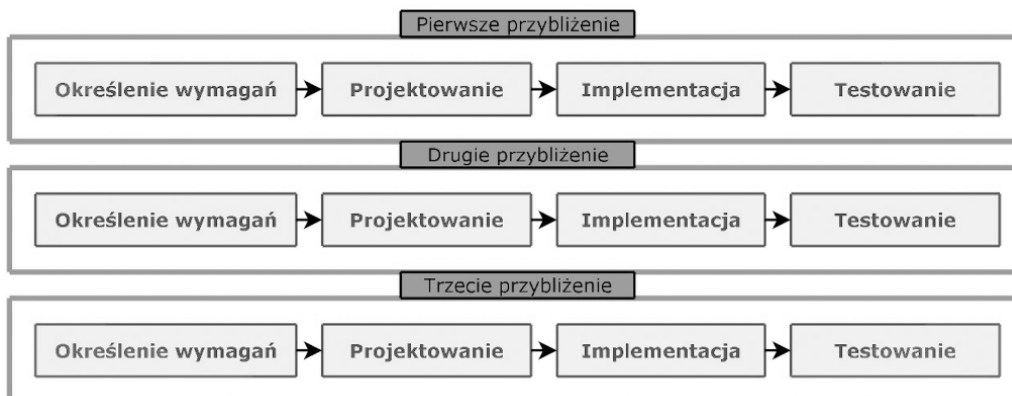
sposób realizacji w kilku wymiarach. Realizowane bloki działań, są rozwijane kaskadowo „od ogółu do szczegółu”, tworząc odpowiednią kaskadę na kolejnych poziomach szczegółowości.

W modelu **wytwarzania oprogramowania kierowanego dokumentami** za podstawę przyjmuje się model kaskadowy wzbogacony o szczegółową dokumentację prac. Każda faza kończy się opracowaniem dokumentacji w pełni ją opisującej. Dokumenty udostępniane są klientowi i po jego akceptacji rozpoczyna się kolejna faza prac. Niedogodnością takiego podejścia jest bardzo duży nakład pracy na opracowanie dokumentacji i „przestoje” w realizacji prac projektowo-programowych, konieczne dla weryfikacji dokumentów przez zamawiającego.

Prototypowanie jest w pewnym sensie receptą na prowadzenie prac nad wytworzeniem oprogramowania użytkowego SI w przypadku braku precyzyjnych wymagań użytkownika. Na podstawie niepełnych wymagań tworzona jest pierwsza wersja systemu (prototyp) – z reguły nie obejmująca funkcjonalności docelowej. Prototyp służy do weryfikacji i konkretyzacji wymagań. Jest poprawiany iteracyjnie – aż do momentu pełnej identyfikacji oczekiwań użytkownika. Następnie przechodzi się do fazy tworzenia właściwego programu, z reguły zgodnie z zasadami modelu kaskadowego.

Model przyrostowy wytwarzania oprogramowania użytkowego SI stanowi połączenie modelu kaskadowego z modelem prototypowania. Istotą tego podejścia jest dostarczanie użytkownikowi kolejnych wersji oprogramowania systemu, o rozszerzanych właściwych użytkowych. Użytkownik po zapoznaniu się z kolejną wersją formułuje wymagania (ich przyrost) na następną wersję (rys. 2).

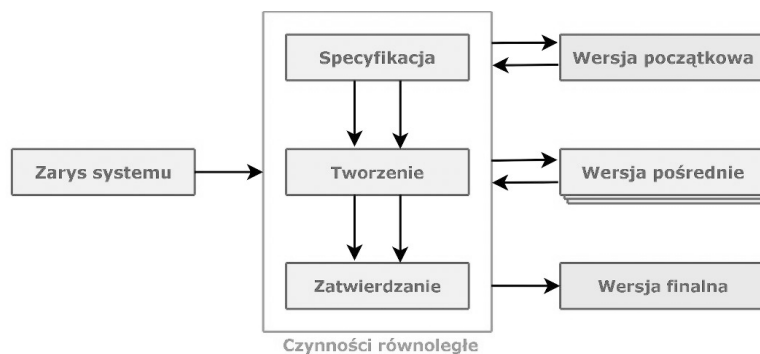
W praktyce w modelu przyrostowym klienci identyfikują w ogólnym zarysie usługi, które system ma oferować. Wskazują, które z nich są dla nich najważniejsze, a które najmniej



Rys. 2. Model przyrostowy życia oprogramowania

ważne. Użytkownik dzięki temu otrzymuje możliwość odłożenia decyzji o szczegółowych wymaganiach do czasu, aż zdobędzie pewne doświadczenia w pracy z systemem.

Model ewolucyjny wytwarzania oprogramowania użytkowego SI polega na opracowaniu wstępnej implementacji, pokazaniu jej użytkownikowi z prośbą o komentarze i udoskonalaniu jej w kolejnych wersjach aż do powstania odpowiedniego systemu (rys. 3).



Rys. 3. Model ewolucyjny życia oprogramowania

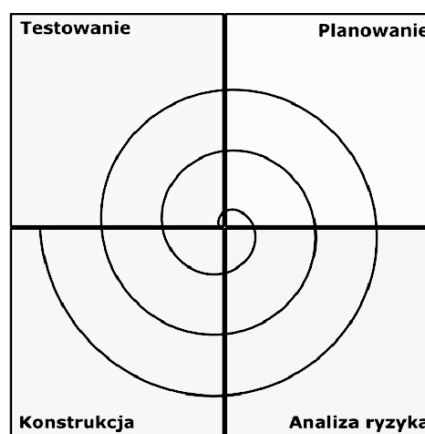
W podejściu ewolucyjnym procesy specyfikacji, tworzenia (implementacji) i weryfikacji poprawności przeplatają się wzajemnie. Pierwsza wersja systemu stworzona jest bardzo szybko na podstawie pewnej abstrakcyjnej specyfikacji. Następnie wersja ta jest modyfikowana na podstawie uwag użytkownika, by w wersji finalnej spełnić jego oczekiwania. Wśród modeli ewolucyjnych wyróżniane są:

- **programowanie odkrywcz**, w których wymagana jest bezpośrednia współpraca z klientem celem sprecyzowania wymagań. Pierwsza wersja systemu jest dostarczana na podstawie częściowych wymagań, a następnie „dokładane” są dodatkowe cechy na podstawie życzeń klienta,
- **prototypowanie z odrzuceniem**, w którym kolejne wersje oprogramowania użytkowego służą wyłącznie do uszczegóławiania wymagań na następne wersje oprogramowania SI.

W **modelu spiralnym** wyróżnione są cztery główne fazy cyklu życia oprogramowania wykonywane cyklicznie (rys. 4):

- **planowanie** – ustalenie celów produkcji kolejnej wersji systemu,
- **analiza ryzyka** – szczegółowa analiza rozpoznanych zagrożeń realizacyjnych,
- **konstrukcja** – wybór modelu tworzenia systemu. Z reguły mamy do czynienia z modelem kaskadowym, choć możliwe są inne podejścia do poszczególnych modułów,
- **testowanie** – ocena użytkownika, jeśli pozytywna rozpoczyna się kolejny cykl.

Każda pętla spirali reprezentuje jedną proces i kończy się stworzeniem prototypu. Model spiralny kładzie nacisk na wielokrotne prototypowanie powtarzające model kaskadowy.



Rys. 4. Model spiralny życia oprogramowania

Wytwarzanie oprogramowania użytkowego dedykowanych GIS w środowisku GEOBA

Geneza opracowania środowiska GEOBA i jego podstawowe elementy składowe

Systemy klasy GIS znajdują zastosowanie przede wszystkim w obszarach:

- analiz przestrzennych,
- zarządzania infrastrukturą,
- zarządzanie przedsiębiorstwami sieciowymi.

Rosnące zapotrzebowanie na GIS implikuje rozszerzanie wymagań i zapotrzebowania na środowiska narzędziowe do wytwarzania ich oprogramowania użytkowego. Jednym z podstawowych wyznaczników jakości środowiska jest *szybkość i łatwość tworzenia systemów dedykowanych*. Stało się to genezą i podstawową ideą opracowania środowiska programowego Geoba.

Prace nad środowiskiem rozpoczęto w późnych latach osiemdziesiątych XX wieku, w których określono jego pożądane właściwości. Początkowo prace prowadzone były przez firmy PBPW CYBER sp. z o.o. i PPW INFOKART S.A. we współpracy z Instytutem Automatyzacji Systemów Dowodzenia WAT. Obecnie kontynuowane są w firmach PPW INFOKART S.A. i PBPW CYBER sp. z o.o.

Programowe Środowisko GEOBA składa się z trzech zasadniczych części (Kołodziński, Betliński, 2004b). Są to (rys. 5):

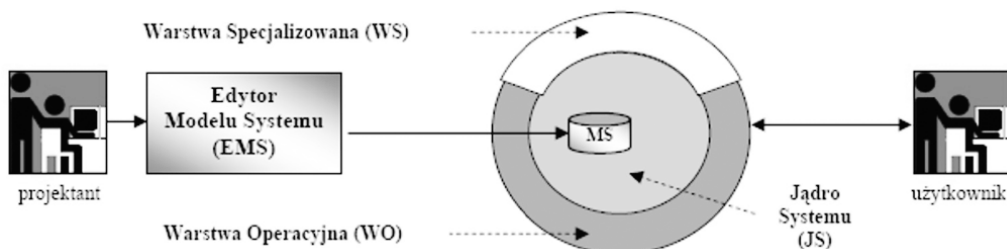
1. Edytor Modelu Systemu (EMS),
2. Jądro Systemu (JS) złożone z Modułów Jądra (MJ),
3. Warstwa Operacyjna (WO) obejmująca Moduły Operacyjne (MO).

Edytor Modelu Systemu jest modulem typu CASE wspomagającym projektowanie i konfigurowanie dedykowanych wersji PŚ GEOBA. Rezultatem jego działania jest:

- Model Systemu – definicje obiektów i parametry konfiguracyjne,
- Schemat Bazy Danych.

Jądro Systemu jest pakietem wykonywalnych komponentów (DLL, OCX, COM+) z interfejsami programowymi obejmującymi niskopoziomowe funkcje udostępniające zasoby systemu.

Warstwa Operacyjna środowiska GEOBA implementuje większość najważniejszych funkcji zaawansowanego GIS wraz z odpowiednim dla ich realizacji graficznym interfejsem użytkownika.



Rys. 5. Schemat środowiska programowego GEOBA

W trakcie procesu wytwarzania oprogramowania, przeznaczonego dla określonego użytkownika, powstaje dodatkowa warstwa oprogramowania, tzw. **Warstwa Specjalizowana** obejmująca wszystkie dodatkowe moduły programowe, które są niezbędne dla zabezpieczenia realizacji tych wszystkich funkcji systemu, które nie są udostępniane przez samo środowisko, a które są niezbędne z punktu widzenia użytkownika. Powstaje tym samym tzw. dedykowana wersja PŚ GEOBA.

Edytor Modelu Systemu – podstawowe narzędzie do wytwarzania oprogramowania użytkowego dedykowanych GIS

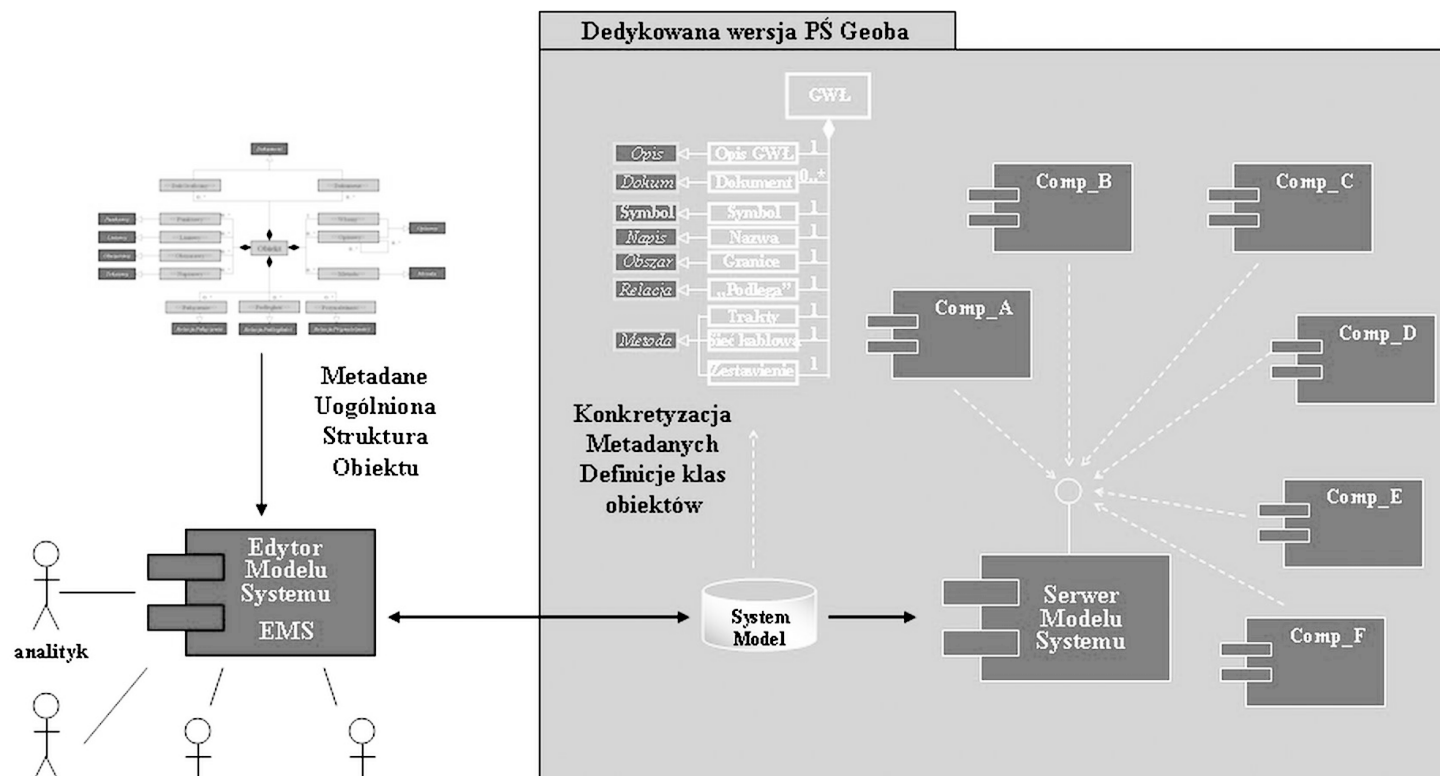
Podstawę działania EMS stanowią metadane, których zasadniczym elementem jest uogólniona klasa obiektów (Kołodziński, Betliński, 2006). W rezultacie prac analityczno-projektowych nad dedykowanym oprogramowaniem użytkowym powstaje cyfrowa konkretyzacja metadanych dla projektowanego systemu GIS, tzw. Model Systemu (MS), który za pośrednictwem specjalnie dla tego celu przeznaczonego modułu (Serwer Modelu Systemu) udostępniany jest wszystkim pozostałym modułom systemu. Ze względu na to, że działanie wszystkich modułów programowych systemu oparte jest na definicjach zawartych w MS, w zasadzie może być on traktowany jako integralna część oprogramowania dedykowanej wersji GEOBY. Utworzenie MS dedykowanej wersji GEOBY w istocie oznacza jednocześnie utworzenie pełnowartościowego, wielodostępnego systemu GIS, eksploatacja którego może być bezzwłocznie rozpoczęta.

Edytor Modelu Systemu z założenia jest narzędziem programowym, które może być wykorzystywane przez wszystkich uczestników procesu wytwarzania GIS, a więc przez użytkowników, analityków, projektantów i programistów. Za pomocą Edytora Modelu Systemu określone są zarówno dane konfiguracyjne (interfejs użytkownika, wykorzystywane układy współrzędnych, obszar działania systemu i jego podział, rodzaje map itp.) jak i klasy obiektów będących właściwym przedmiotem działania tworzonego systemu (rys. 6).

Edytor Modelu Systemu stanowi zamkniętą, pod względem funkcjonalnym, część środowiska GEOBA umożliwiającą projektantowi sprawne tworzenie systemów dedykowanych. Podstawową zasadą działania EMS jest maksymalne wykorzystanie istniejących repozytoriów. Dotychczasowe prace koncepcyjne i projektowe stanowią bazę dla wszelkich nowych projektów. Zawsze punktem wyjścia staje się przetestowany i sprawdzony model – rozbudowywany o dodatkowe elementy. Tworzona definicja systemu jak i samo narzędzie pozwalają na wykorzystanie zalet obiektowego opisu rzeczywistości. Istotna jest przede wszystkim, wynikająca z koncepcji Uogólnionej Klasy Obiektów (Kołodziński, Betliński, 2006) pełna unifikacja obsługi i definicji modelowanych obiektów. Opierając się na pewnym zbiorze bazowych klas kształtowane są kolejne typy obiektów.

Jeśli chodzi o walory ściśle użytkowe, EMS zapewnia pełną integracją z interfejsem bazodanowym (np. Oracle), uwalniając projektanta od konieczności jakiegokolwiek ingerencji w strukturę bazy danych. Wszystkie operacje edycyjne dokonywane na modelu systemu są automatycznie implementowane w schemacie bazy danych. Dotyczy to zarówno prostych działań typu dodanie, usunięcie atrybutu lub komponentu, ale również bardziej zaawansowanych jak zmiana typu atrybutu, co jest związane z konwersją danych. Jest to szczególnie istotne w przypadku modyfikacji modelu systemu w fazie produkcyjnej.

W przypadku środowiska GEOBA świadomie używamy określenia **model systemu**, co w odróżnieniu od modelu danych, oznacza również pełną parametryzację środowiska pracy.



Rys. 6. Edytor Modelu Systemu – podstawowe narzędzie do wytwarzania oprogramowania użytkowego dedykowanych SIP

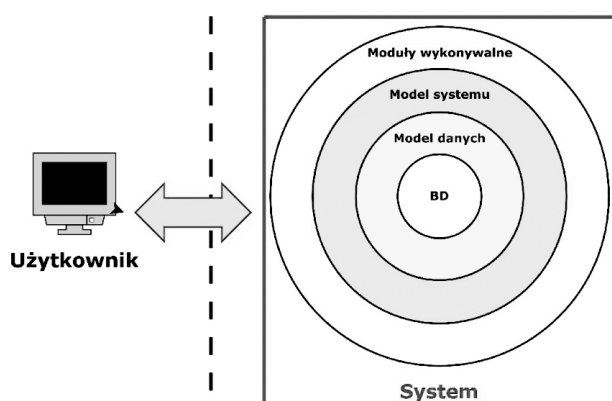
Edytor Modelu Systemu umożliwia znaczące modyfikacje interfejsu środowiska GEOBA dla potrzeb opracowywanego systemu dedykowanego. Co istotne, wszelkie dane konfiguracyjne obsługiwane są również przez właściwe im klasy obiektów. Mamy, więc do czynienia z klasycznym przykładem architektury systemu sterowanej modelem.

Wytwarzanie architektury oprogramowania sterowane modelem

GIS charakteryzują się dużymi nakładami na pozyskiwanie danych. W praktyce oznacza to, iż pełną funkcjonalność systemu otrzymuje się dopiero po wypełnieniu bazy danych. Wiąże się z tym zasadniczy problem – utrzymania niezmienności wymagań na system. W trakcie tak wydłużonego cyklu życia systemu w jego opisie (wymaganiach, założeniach, projekcie itp.) pojawia się szereg zmian, które znacznie utrudniają doprowadzenie oprogramowania użytkowego GIS do pomyślnego wdrożenia. Spostrzeżenie to ma przełożenie na wymóg dotyczący architektury systemu. Budowany w niej system musi zacząć działać jak najszybciej, (aby można było wprowadzać dane) oraz powinna istnieć możliwość dokonywania znacznych modyfikacji definicji przetwarzanych w nim obiektów (reagowanie na zmiany wymagań). Analizując historię rozwoju inżynierii oprogramowania, a wraz z nią wsparcia narzędziowego i metodologicznego przy jednoczesnym braku zauważalnych efektów w postaci zwiększenia liczby projektów SI zakończonych sukcesem, można pokusić się o syntezę problemów dotyczących procesu wytwarzania systemów:

- problem czasu – skrócenie czasu wytwarzania w dotychczasowej praktyce jest niemożliwe albo osiąganego poprzez istotne zwiększenie kosztów na produkcję półproduktów,
- problem komunikacji – zapewnienie odpowiedniej komunikacji między uczestnikami procesu tworzenia systemu (użytkownikami, projektantami i programistami) skutkuje koniecznością utrzymywania skomplikowanych, rozbudowanych a nierzadko i nadmiarowych opisów modelu systemu (w postaci diagramów, opisów słownych itp.),
- problem złożoności – nadmiernie rozbudowana metodyka projektowania wymaga specjalistycznych narzędzi a także profesjonalnych szkoleń.
- problem zmian – dotyczy zmian wymagań powodowanych zmianami zachodzącymi w otoczeniu SI (przepisów, procedur, zestawu danych itp.) oraz możliwości ich uwzględniania w systemie.

Zaproponowanie rozwiązania uwzględniającego wszystkie wyszczególnione problemy wy-



Rys. 7. Architektura oprogramowania sterowana modelem

ma radykalnego przededefiniowania procesu tworzenia systemu informatycznego. Taką zmianą jest wykorzystanie **architektury oprogramowania GIS sterowanej modelem**, rozumianej jako: zbiór odpowiednio sparametryzowanych i współpracujących ze sobą modułów programowych.

Wytwarzanie architektury oprogramowania sterowanej modelem (rys. 7) opiera się na wprowadzeniu bufora (w postaci Modelu Systemu) między bazą danych a moduły wykonywalne. Model Syste-

mu zawiera definicję systemu oraz metadane przetwarzanych w nim obiektów. W efekcie zachowanie systemu, a także zakres przetwarzanych w nim informacji jest zdefiniowany w Modelu Systemu. Moduły wykonywalne nie operują bezpośrednio na danych rzeczywistych, a system „widzą” przez definicje dostarczane z warstw Model Systemu i Model Danych. Praktycznie może to wyglądać w ten sposób, iż ten sam moduł wykonywalny może pracować w różnych systemach, albo jeszcze inaczej – może przetwarzać dane z różnych systemów (dla różnych Modeli Systemu). Wprowadzenie warstwy pośredniej zmienia zasadniczo percepcję systemu przez wszystkich aktorów zaangażowanych w jego powstawanie: analityków/projektantów, programistów i użytkowników. Powyższa metoda projektowania systemów informatycznych stosowana jest przy wytwarzaniu oprogramowania użytkowego GIS w środowisku narzędziowym GEOBA.

GEOBA jest spójnym środowiskiem umożliwiającym jednoczesną pracę wszystkich twórców systemu informatycznego (projektantów, programistów i użytkowników) (rys. 8). Model systemu jest buforem, który pozwala jednocześnie rozwijać i użytkować system. Dzięki architekturze oprogramowania sterowanej modelem każdy uczestnik procesu wytwarzania systemu skupia się na innych, istotnych dla niego, aspektach systemu.

Wytwarzanie oprogramowania użytkowego GIS w środowisku GEOBA

Wytwarzanie GIS w środowisku GEOBA polega na ciągłym prototypowaniu systemu poprzez modyfikacje Modelu Systemu i dodawanie kolejnych warstw specjalizowanych otaczających podstawową funkcjonalność. Jest to metodyka tworzenia systemu bazująca na architekturze sterowanej modelem, przy następujących założeniach:

- maksymalne zaangażowanie użytkownika w proces wytwarzania,
- maksymalnie skrócony czas uruchomienia systemu (prototypowanie - kolejne przybliżenia systemu),
- maksymalne wykorzystanie już istniejących rozwiązań (definicji obiektów, modułów wykonywalnych),
- maksymalna elastyczność – możliwość ciągłej modyfikacji bez utraty zdolności operacyjnej systemu,
- zmiany definicji obiektów powodują automatyczne modyfikacje schematu BD bez utraty zawartości.

System zaczyna działać od momentu zdefiniowania jego zakresu (identyfikacji oraz wstępnej definicji przetwarzanych w nim obiektów) systematycznie i stale ulegając modyfikacji i dostosowaniu do wymagań użytkownika. W omawianej architekturze Model Systemu może być traktowany jako:

- dokumentacja projektowa (techniczna),
- platforma współdzielonej informacji o systemie,
- element parametryzujący działanie modułów wykonywalnych.

Proces prototypowania ma charakter ciągły i nie ma potrzeby określania sztywnych jego faz. Jednakże z uwagi na możliwość porównania z innymi rozwiązaniami, sekwencje czynności wykonywanych w trakcie budowy systemu w środowisku GEOBA podzielono na następujące fazy (które są miarą stopnia dopasowania systemu do informatyzowanej organizacji):

- identyfikacji,
- analizy,
- konfiguracji,
- eksploatacji.

Faza identyfikacji. Analityk wraz z użytkownikiem decydują o zakresie właściwości funkcjonalnych systemu (rys. 9). Posiłkując się repozytorium Modeli Systemu, wybierają do nowotworzonego produktu definicje obiektów.

Użytkownik może obserwować (kontrolować) zawartość informacyjną systemu za pomocą uniwersalnych interfejsów użytkownika. Etap służy określeniu granic systemu oraz wstępnemu przybliżeniu użytkownikowi definicji przetwarzanych w nim obiektów. Już od tego momentu użytkownik może gromadzić dane w systemie oraz rozpocząć jego użytkowanie w zakresie ewidencji i raportowania w wybranym przez siebie obszarze.

Faza analizy. Wybrane z repozytorium definicje obiektów są dostosowywane do wstępnie zidentyfikowanych procesów zachodzących w systemie (rys. 10). Definiowane są nowe obiekty specyficzne dla tworzonego systemu. Następuje wstępne określanie funkcjonalności oraz parametryzowanie interfejsów, które są udostępniane w systemie.

Faza konfiguracji. Powstają kolejne warstwy otaczające i specjalizujące podstawową funkcjonalność systemu (rys. 11). W proces tworzenia zaangażowani są wszyscy aktorzy. W tej fazie system nabiera cech indywidualnych. Dopasowywanie systemu, projektowanie i implementacja automatyzacji realizacji procesów.

Faza eksploatacji. Powstają kolejne warstwy wynikające z rozszerzenia zakresu komputerowego wspomaganie realizacji zadań przez użytkownika, optymalizacji ich realizacji i zwiększenia komfortu pracy w systemie (rys. 12). System działa stabilnie i poprawnie, ale pojawiają się nowe wymagania i potrzeby zmiany definicji obiektów oraz procesów zmiany akcentów, dodatkowe interfejsy użytkownika, reagowanie na zmiany otoczenia systemu.

Efektywność wytwarzania systemów informatycznych klasy GIS w środowisku GEOBA

Czynniki określające efektywność tworzenia systemów informatycznych

Wytwarzanie systemu informatycznego traktowane jest jako rodzaj projektu biznesowego. Przyczyną przedsięwzięcia informatycznego jest pewna, częstokroć nie do końca zdefiniowana, potrzeba poprawy efektywności funkcjonowania przedsiębiorstwa bądź instytucji. Rolą wykonawcy dedykowanego SI jest zaspokoić ją przy jednoczesnej minimalizacji poniesionych kosztów. Systemy klasy GIS stanowią specyficzną grupę wśród SI. Podstawowe etapy wytwarzania tzn. analiza wymagań, projekt, implementacja i wdrożenie są zasadniczo podobne. Główna różnica uwidacznia się przy całościowym spojrzeniu na przedsięwzięcie informatyczne. Pierwszoplanową rolę, w sensie kosztowym i czasowym, ma faza wypełniania bazy danych. GIS można uznać za w pełni wdrożony dopiero w momencie osiągnięcia gotowości do jego użytkowania.

Efektywność najprościej można zdefiniować jako relację pomiędzy oczekiwanym przyrostem przychodów (lub zmniejszeniem dotychczasowych kosztów realizacji zadań) organizacji a nakładami na ich osiągnięcie. Jest to najbardziej jednoznaczny sposób oceny przedsięwzięcia. Oczywiście nie można pomijać wielu „niematerialnych” korzyści takich jak zgromadzone doświadczenie i zadowolenie klienta, które z pewnością zaprocentują w przyszłości.

Ogół wszystkich czynników wpływających na końcową efektywność przedsięwzięcia (przy ustalonych pożądanym właściwościach użytkowych), zarówno dla wykonawcy jak i klienta, można praktycznie zagregować do dwóch wielkości:

- kosztu wytworzenia i wdrożenia systemu,
- czasu wdrożenia.

Analiza efektywności budowy systemów klasy GIS przy zastosowaniu środowiska programowego GEOBA

Elementem determinującym zarówno przebieg prac nad systemem jak i w konsekwencji jego efektywność jest wybór środowiska pracy aplikacji użytkowej. Niżej zostanie dokonana analiza wpływu środowiska narzędziowego GEOBA na efektywność tworzenia systemów klasy GIS.

Na poprzedniej stronie przedstawiono zarys procesu wytwarzania oprogramowania użytkowego w środowisku GEOBA. Należy przede wszystkim podkreślić fakt dostępności gotowego do użycia systemu już w momencie rozpoczęcia fazy analizy wymagań. Z istniejącego repozytorium modeli systemów wybierany jest najbardziej zbliżony do opracowywanego. Na jego podstawie tworzony jest wzorzec nowego systemu, który jest gotowy do użycia zarówno przez projektanta jak i użytkownika końcowego. W procesie prototypowania środowisko programowe GEOBA obudowywane jest kolejnymi warstwami specjalizowanymi. Takie podejście oczywiście nie wyklucza zastosowania klasycznego, np. kaskadowego modelu wytwarzania aplikacji, jednak w przypadku systemów GIS natychmiastowa dostępność do użycia systemu jest nie do przecenienia. W początkowej fazie budowy oprogramowania z reguły nie są znane wszystkie wymagania odnośnie pożądanego funkcjonalności systemu, natomiast dość szybko ustalana jest pożądana zawartość informacyjna bazy danych. Model wzorcowego systemu jest rozbudowywany zgodnie z projektem obiektów bazy danych i od razu jest gotowy do wprowadzania danych. Krótkie szkolenie użytkowników z zakresu wykorzystania środowiska pozwala na rozpoczęcie procesu ładowania bazy danych, który jest elementem krytycznym podczas budowy SIP. Możliwość równoległych prac programistycznych i bazodanowych znacząco skraca czas wdrożenia systemu. Wiąże się z tym również korzyści innego rodzaju. Użytkownik eksploatując system zapoznaje się z jego technologią funkcjonowania, nabiera pożądanym nawyków oraz zgłasza swoje uwagi pomocne w doskonaleniu właściwości użytkowych oprogramowania dedykowanego GIS. Dodatkowo, następuje proces samokształcenia znacznie redukujący nakłady na szkolenia.

Na szczególne podkreślenie w środowisku GEOBA zasługuje w pełni obiektowy interfejs programowy. Programista tworzy kolejne moduły specjalizowane obudowujące niezmiennie dla niego jądro środowiska. Ich działanie jest całkowicie uzależnione i sterowane przez Model Systemu. Mamy więc do czynienia, podobnie jak w przypadku sfery bazodanowej, z podejściem przyrostowym. Każdy przyrost jest realizowany całkowicie bezkonfliktowo zarówno w fazie wdrożenia, jak i eksploatacji systemu.

Innym, niezwykle korzystnym aspektem, tym razem dla wytwórcy środowiska GEOBA, jest ciągły przyrost jego funkcjonalności niejako przy okazji tworzenia systemów. Budowa oprogramowania użytkowego wymusza działania optymalizacyjne również w jądrze środowiska. Architektura oparta na Uogólnionej Klasie Obiektów umożliwia modyfikacje wewnętrznych struktur środowiska z zachowaniem kompatybilności wszystkich powstałych syste-

mów dedykowanych. Następuje ciągły rozrost funkcjonalności podstawowych możliwych do wykorzystania w nowych produktach. Dotyczy to również zasobów projektowych i dokumentacyjnych.

Podsumowanie

Budowa systemów informatycznych klasy GIS jest niezwykle skomplikowanym i kosztownym przedsięwzięciem. Praktycznie nie jest możliwe wytworzenie w pełni powielarnego produktu gotowego do dystrybucji na zasadzie „toważ z półki”. Kaądy system musi uwzględniać specyfikę użytkownika i być w maksymalny sposób dostosowany do jego wymagań. Rozwiązanie polegające na każdorazowym tworzeniu oprogramowania użytkowego GIS od podstaw nie wchodzi w grę ze względów ekonomicznych – produkt ma właściwie tylko jednego, dedykowanego odbiorcę. Dla zwiększenia efektywności przedsięwzięcia konieczne jest wykorzystanie środowiska narzędziowego jako jądra systemu dedykowanego, rozszerzonego o charakterystyczne dla niego funkcje i moduły. Podstawowe czynniki warunkujące wysoką efektywność produkcji systemu GIS, to koszt wytworzenia oprogramowania oraz jego wdrożenia.

Budowa GIS w środowisku GEOBA odbywa się na zasadzie rozbudowy systemu wzorcowego o nowe, specjalizowane moduły. Już w fazie analizy gotowy jest system o podstawowej, ale wystarczającej dla rozpoczęcia wypełniania bazy danych, funkcjonalności. Wykorzystanie Uogólnionej Klasy Obiektów pozwala na pełną unifikację obsługi elementów systemu. Z kolei, dzięki zastosowaniu metody wytwarzania architektury oprogramowania sterowanego modelem możliwa jest jednoczesna praca wszystkich twórców systemu informatycznego – projektantów, programistów i użytkowników. Kaądy z nich może skupić się na istotnych dla niego aspektach systemu.

Literatura

- Adamczewski P., 2000: Zintegrowane systemy informatyczne w praktyce. MIKOM, Warszawa.
- Jaszkiewicz A., 1997: Inżynieria oprogramowania. HELION, Warszawa.
- Kołodziński E., Betliński G., 2002: Środowisko programowe Geoba do wytwarzania zautomatyzowanych systemów dowodzenia. Właściwości i możliwości. X Konferencja Naukowa Automatyzacja Dowodzenia.
- Kołodziński E., Betliński G., 2003: Środowisko programowe Geoba do wytwarzania ZSyD – Zunifikowane stanowisko Funkcyjne. XI Konferencja Naukowa Automatyzacji Dowodzenia, Pieczyska.
- Kołodziński E., Betliński G., Popławski R., Ożarowski M., Kapłański P., 2003: Technologia wytwarzania zintegrowanego systemu informatycznego o architekturze oprogramowania sterowanej modelem danych. XI Konferencja Naukowa Automatyzacji Dowodzenia, Pieczyska.
- Kołodziński E., Betliński G., 2004a: Model systemu GIS w środowisku programowym GEOBA. Materiały XII Konferencji Naukowej „Automatyzacja dowodzenia”, Gdynia - Jurata.
- Kołodziński E., Betliński G., 2004b: Programowe środowisko GEOBA wspomagające wytwarzanie systemów GIS. Opracowanie wewnętrzne, Warszawa.
- Kołodziński E., Betliński G., 2006: Modelowanie systemów informacji przestrzennej z zastosowaniem uogólnionej klasy obiektów. *Roczniki Geomatyki* t. V z. 2, PTIP, Warszawa.
- Sommerville I., 2003: Inżynieria oprogramowania, WNT.
- Wrycza S., Marcinkowski B., Wyrzykowski K., 2005: Język UML 2.0 w modelowaniu systemów informatycznych. Wydawnictwo Helion S.A.

Summary

The root cause of building an information system is, often not fully defined, the need to improve efficiency of functioning of a company or institution. Creation of an information system should be treated as a business project. The goal of the entity responsible for the creation is to fulfill expectations of a customer keeping the costs as low as possible.

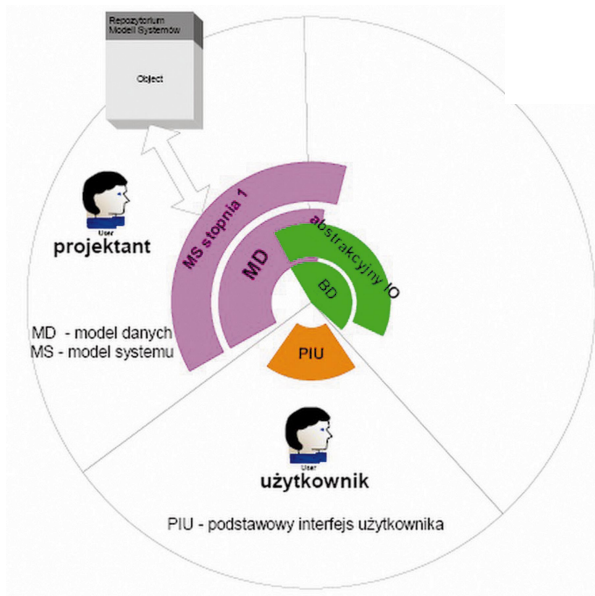
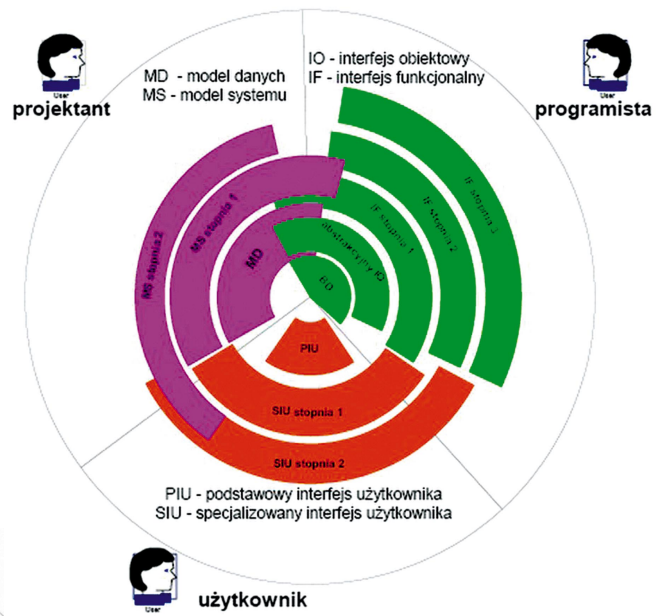
Systems of GIS class make a special group of IS. Basic stages of their creation i.e. Requirements analysis, design, application and implementation are similar to other IS. The main difference shows up in a comprehensive view on the information ventures of the organization. The main role, as regards costs and time, takes the stage of filling out databases. The GIS system may be considered fully implemented only when its readiness for use is achieved.

An ultimate advantage of GEOBA programming environment is the possibility to generate a fully cooperating prototype already on the stage of requirements analysis. The prototype made available at the time to the user allows him to establish correctly functional expectations for the dedicated GIS software. Additionally, after setting up contents of a database of such a GIS, there is a possibility (practically at the same time) to fill it out with data and to develop the system. Especially important property of the GEOBA environment is the possibility for a simultaneous work of all different parties engaged in the creation of the system – designers, programmers and users. Each group of them may focus on aspects of the system essential for them. Additionally, thanks to the General Class of Objects and the complete list of edition operations connected with it, there is a possibility of instant creation of a standard system. It doesn't require any additional programming, only the proper definition of the data model.

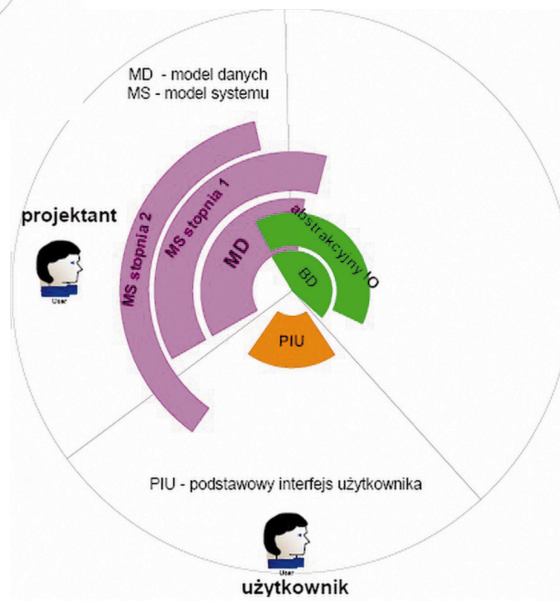
dr hab. inż. Edward Kołodziński, prof. UWM
ekolodzinski@wp.pl

mgr inż. Ireneusz Wyglądała
irekw@infocorp.com.pl
tel.: (0-22) 638 41 78

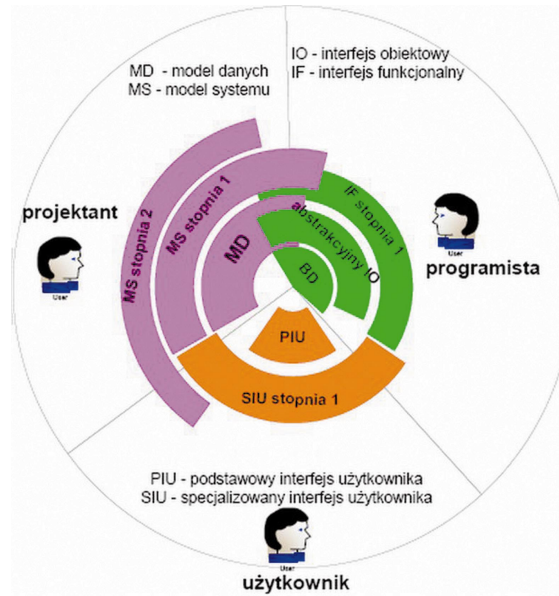
Rys. 8. Dostęp Aktorów wytwarzania systemu do oprogramowania w architekturze Geoba



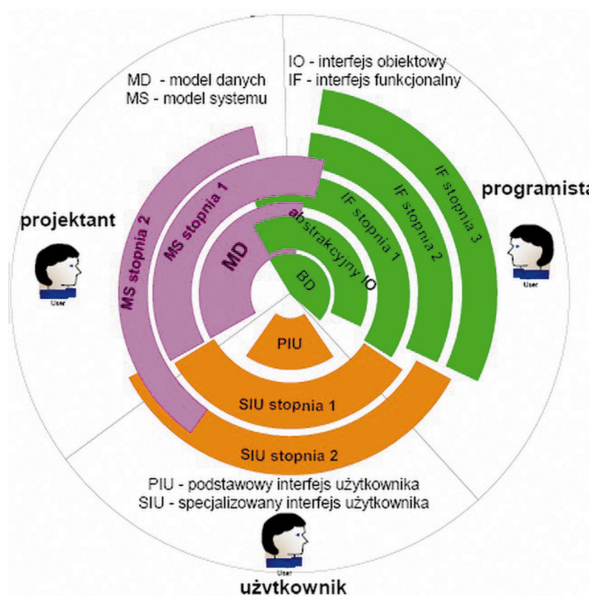
Rys. 9. Stan systemu w fazie identyfikacji



Rys. 10. Stan systemu w fazie analizy



Rys. 11. Stan systemu w fazie konfiguracji



Rys. 12. Stan systemu w fazie eksploatacji