

Przegląd metod, środków formalnych i narzędzi programowych wspomagających modelowanie pojęciowe informacji geograficznej, cz. II

METODY, ŚRODKI, NARZĘDZIA

Nawet najlepsze metodyki i narzędzia nie zapewnią dobrej jakości projektów informatycznych. Kluczem do sukcesu są zespoły doświadczonych i kompetentnych osób, które metodyki, notacje i narzędzia jedynie wspomagają w procesie tworzenia oprogramowania.

AGNIESZKA CHOJKA

Jak już wyjaśniałam w poprzedniej części artykułu (GEODETA 5/2006), środki formalne i narzędzia programowe wykorzystywane są przez określoną metodę, która z kolei jest składnikiem wybranej metodyki (metodologii). Różnego rodzaju notacje służące do zapisu modelu, techniki budowy modelu (metody analizy) oraz narzędzia ułatwiające stosowanie notacji i metod to składowe inżynierii oprogra-

mowania wykorzystywane m.in. w fazie analizy.

• METODYKI I METODY

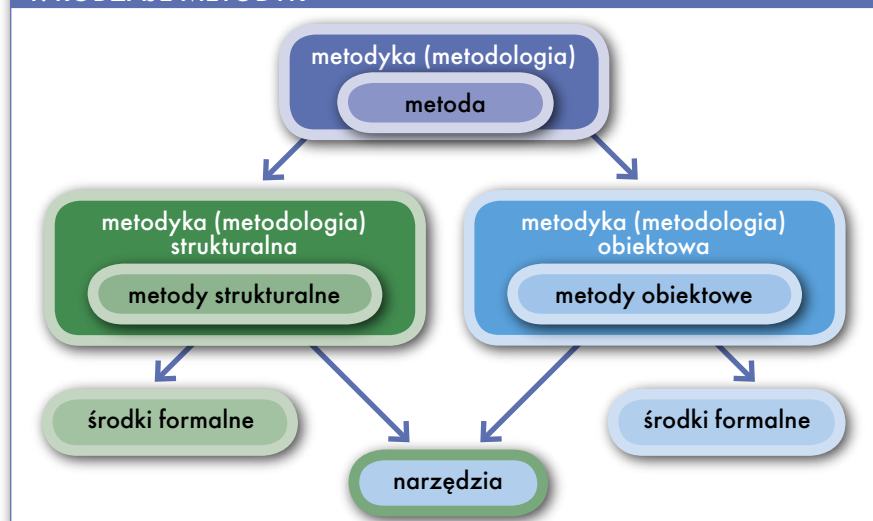
Można wyróżnić dwie główne grupy metodyk, a co za tym idzie metod analizy (rys. 1):

• **strukturalne** (*structured methodologies, structured methods*) – opierają się na wyróżnianiu w analizowanym systemie dwóch rodzajów składowych: pasywnych (odzwierciedlających przechowywanie w systemie pewnych danych) oraz aktywnych (odzwierciedlających

PODSTAWOWA TERMINOLOGIA

- **Metodyka** – zespół zasad, sposobów, metod wykonania jakiejś pracy, badań nad jakimś zagadnieniem.
 - **Metodologia** – nauka o metodach (...), określony scenariusz postępowania.
 - **Metoda** – technika, sposób postępowania dla uzyskania określonego celu [Mały Słownik Języka Polskiego 2000].
- Metodyka (metodologia), precyzując powyższą definicję pod kątem tworzenia systemu informacyjnego, to zestaw pojęć, notacji, modeli, języków, technik i sposobów postępowania służący do analizy dziedziny stanowiącej przedmiot projektowanego systemu oraz do projektowania pojęciowego, logicznego i/lub fizycznego (modelowania pewnej dziedziny problemowej). Ponadto metodyka ustala:
- fazy projektu, role uczestników projektu,
 - modele tworzone w każdej z faz,
 - scenariusze postępowania w każdej z faz,
 - reguły przechodzenia od jednej fazy do następnej,
 - techniki i notacje, których należy używać,
 - dokumentację powstającą w każdej z faz [Subieta 2002].

1. RODZAJE METODYK



wykonywanie w systemie pewnych operacji, procesów),

• **obiektywne** (*object-oriented methodologies, object-oriented methods*) – podstawą tych metod jest wyróżnienie w systemie składowych, które łączą w sobie

możliwość przechowywania danych oraz wykonywania pewnych operacji [Jaszkiewicz 1997].

• METODYKI I METODY STRUKTURALNE

Analiza strukturalna (*structured analysis*) rozpoczyna się od budowy dwóch różnych modeli systemu:

- model danych – opis pasywnej (statycznej) części systemu,
- model funkcji – opis aktywnej części systemu.

Następnie oba modele są integrowane, a wynikiem jest model przepływu danych (*data flows model*).

Metody strukturalne są szczególnie przydatne, gdy jeden z aspektów (pasywny lub aktywny) jest wyraźnie istotniejszy niż drugi. W przypadku systemów informatycznych służących niemal wyłącznie do przechowywania i wyszukiwania złożonych danych, nierealizujących żadnych skomplikowanych funkcji, najistotniejsza jest budowa modelu danych. Jeżeli natomiast system ma realizować złożone funkcje na prostych danych (np. skomplikowane obliczenia naukowe), wówczas istotniejsze jest modelowanie funkcji. Metody strukturalne sprawdzają się gorzej, gdy zadaniem systemu jest realizacja skomplikowanych funkcji na złożonych danych. Analiza i projektowanie strukturalne nie są też dobrym punktem wyjścia dla implementacji w środowisku obiektowym. Uważa się, że wadą metodyk strukturalnych są trudności w zintegrowaniu modeli danych i funkcji [Jaszkiewicz 1997].

Przykłady metodyk strukturalnych:

- Metodyka Yourdona (DeMarco i Ward/Mellor),
 - Structured System Analysis and Design Methodology (SSADM),
 - Structured Analysis and Design Technique (SADT).
- Analiza strukturalna (zgodnie z DeMarco) używa następujących technik:
- Diagramy Przepływu Danych (Data Flow Diagrams, DFD),
 - Słownik Danych (Data Dictionary),
 - Strukturalny Angielski (Structured English) -> Strukturalny Polski,
 - Tablice Decyzyjne (Decision Tables),
 - Drzewa Decyzyjne (Decision Trees, DFD).
- Dodatkowo wymienić można:
- Schemat Transformacyjny (Transformation Schema),
 - Diagram Przejść Stanów (State Transition Diagrams),

Tab. 1. RODZAJE METODYK PROJEKTOWANIA SYSTEMU INFORMACYJNEGO [Traczyk 2004]

Strukturalne	Obiektowe
• dane i procesy modelowane osobno, • wykorzystują tylko proste typy danych, • dobrze dostosowane do modelu relacyjnego danych, • podstawowe techniki: • diagramy związków-encji (ERD), • hierarchie funkcji (FHD), • diagramy przepływu danych (DFD), • modele macierzowe.	• dane i procesy modelowane łącznie, • wykorzystują złożone typy danych, • dostosowane do obiektowego modelu danych, • w przypadku realizacji opartej na relacyjnej bazie danych, wynikowy obiektowy model danych musi być odpowiednio przekształcony, • podstawowe techniki: • diagramy klas UML, • przypadki użycia i modele dynamiczne UML.

- Lista Zdarzeń (Event List),
- Schemat Danych (Data Schema),
- Pre- i post-warunki (Pre- i post-Conditions),
- Diagramy Encja-Związek (występują w SSADM),
- Historie Życia Encji (występują w SSADM) [Subieta 2002].

Na potrzeby modelowania pojęciowego informacji geograficznej z powodzeniem wystarcza Metoda Związków-Encji (Diagramy Encja-Związek) dla opisu statycznej części systemu oraz Diagramy Przepływu Danych i Diagramy Przejść Stanów dla zobrazowania procesów zachodzących w systemie informacji geograficznej.

• METODYKI I METODY OBIEKTOWE

Obiektowe metody analizy są wygodnym narzędziem analizy skomplikowanych systemów, a w szczególności systemów o złożonej stronie pasywnej i aktywnej. Stosowanie obiektowych metod analizy niekoniecznie musi pociągać za sobą implementację w środowiskach obiektowych (korzystanie z obiektowych języków programowania, obiektowych baz danych czy też obiektowych bibliotek). Metodyka obiektowa charakteryzuje się tym, że:

- wykorzystuje pojęcie obiektowości dla potrzeb modelowania pojęciowego oraz analizy i projektowania systemów informatycznych,
- podstawowym jej składnikiem jest statyczny diagram klas, będący zwykle wariantem notacyjnym i pewnym rozszerzeniem diagramów encja-związek z metodyki strukturalnej,
- diagram klas zawiera m.in. klasy (a w ich ramach specyfikacje atrybutów i metod), związki generalizacji, asocjacji i agregacji, liczności tych związków oraz różnorodne ograniczenia,

• uzupełnieniem diagramu klas są np. diagramy dynamiczne uwzględniające stany i przejścia pomiędzy tymi stanami, diagramy interakcji ustalające zależności pomiędzy wywołaniami metod, diagramy funkcjonalne (będące zwykle pewną mutacją diagramów przepływu danych) itd.,

• koncepcja przypadków użycia (*use case*) zakłada odwzorowanie struktury systemu z punktu widzenia jego użytkownika.

Przykłady metod obiektowych oraz notacji i języków formalnych:

- EXPRESS,
- OODA (Booch),
- OMT (Rumbaugh),
- OOSA (Shaler-Mellor),
- Objectory (Jacobson),
- MOSES/OPEN,
- OOA/OD (Coad/Yourdon),
- notacja UML,
- RUP [Subieta, 2002].

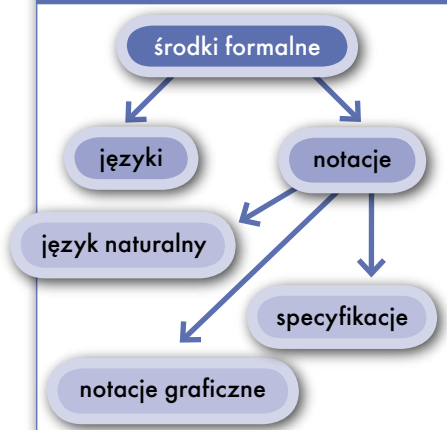
Normalizacja w dziedzinie informacji geograficznej zakłada wykorzystanie notacji UML dla potrzeb modelowania pojęciowego, w tym budowy schematów pojęciowych. Zaleca również stosowanie środków formalnych w postaci języka XML i GML. XML (eXtensible Markup Language) to metajęzyk będący podstawą do zdefiniowania języka GML (Geography Markup Language).

Zestawienie porównawcze dwóch wiodących grup metodyk przedstawia tab. 1.

• ŚRODKI FORMALNE

Metodyka jest powiązana z określonym środkiem formalnym (np. notacją, językiem) służącym do dokumentowania wyników faz projektu (pośrednich i końcowych). Zadaniem takiego środka jest wspomaganie ludzkiej pamięci i wyobraźni podczas procesu projektowania, jak również zapewnienie sprawnej komunikacji w zespołach projektowych

2. PODZIAŁ ŚRODKÓW FORMALNYCH



oraz pomiędzy projektantami i klientem. Do zapisu modelu stosuje się następujące środki formalne (rys. 2):

- języki,
- notacje (rodzaj języka), a wśród nich język naturalny, notacje graficzne oraz specyfikacje (język sformalizowany, częściowo ustrukturalizowany zapis tekstowy i numeryczny).

Środki formalne wykorzystuje się również w pozostałych fazach tworzenia systemu informacyjnego. Na etapie modelowania pojęciowego szczególną rolę odgrywają notacje graficzne. Nie bez powodu mówi się, że jeden obraz wart tysiąca słów. Łatwiej zrozumieć prosty schemat graficzny, niż szukać sensu w gąszczu słów. Diagramy graficzne umożliwiają szybkie przekazywanie podstawowych informacji. Jednak nie są w stanie pokazać wszystkich szczegółów. Zbyt wysoka szczegółowość diagramów może znacznie obniżyć ich czytelność. Stąd też notacje graficzne uzupełniane są przez tzw. specyfikacje, czyli dodatkowe informacje (tekstowe, numeryczne) umieszczane na diagramach.

Wymienione powyżej środki formalne mogą pełnić różne funkcje:

- Są podstawowym narzędziem pracy analityka i projektanta systemu. Inżynier oprogramowania zapisuje i analizuje swoje pomysły, posługując się wymienionymi środkami. Muszą więc one ułatwić pracę nad złożonymi systemami. Powinny być więc przejrzyste, łatwo zrozumiałe, powinny ułatwić modelowanie złożonych zależności.

- Środki wykorzystywane w fazie analizy są często wykorzystywane do współpracy z użytkownikiem. Muszą więc być dość proste, przejrzyste i zrozumiałe.

- Praca inżyniera oprogramowania jest praktycznie zawsze pracą zespołową.

Środki formalne są podstawowym narzędziem współpracy pomiędzy członkami zespołu. Służą szybkiemu i precyzyjnemu przekazywaniu idei.

- Notacje wykorzystywane w fazie analizy i projektowania są podstawą implementacji oprogramowania. Powinny być więc przejrzyste i precyzyjne.

- Notacje te służą także do zapisu dokumentacji technicznej [Jaszkievicz 1997].

• PRZYKŁADY

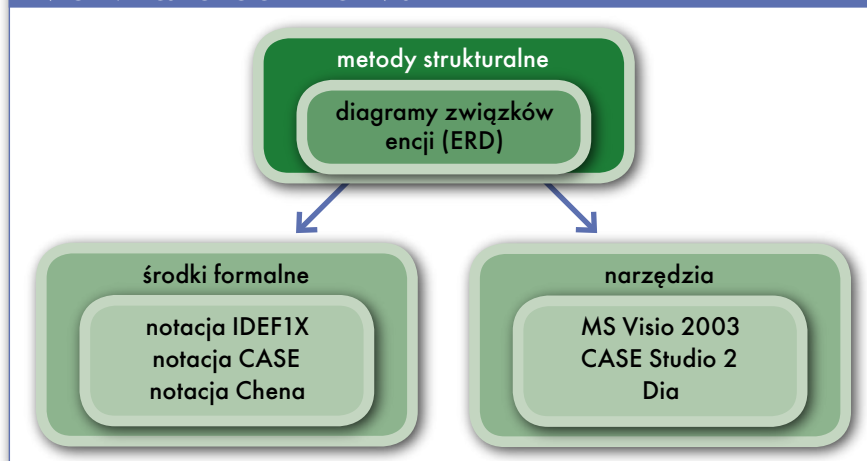
Poniższe schematy przedstawiają przykłady środków formalnych i narzędzi programowych charakterystycznych dla wybranej metodyki: strukturalnej bądź obiektowej.

• NARZĘDZIA

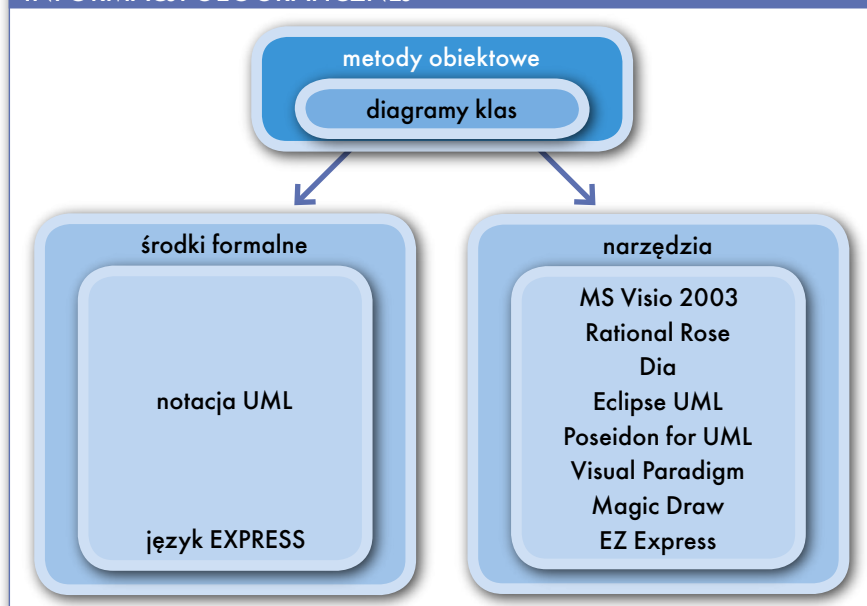
Znając już konkretną metodykę oraz metody i środki formalne, którymi posługuje się dana metodyka, możemy już stworzyć interesujący nas model. Przydatne w tym momencie będą specjalizowane narzędzia typu CASE stosowane w inżynierii oprogramowania.

Termin CASE (Computer Assisted/Aided Software/System Engineering) oznacza komputerowo wspomaganą inżynierię oprogramowania/systemów. Już sama nazwa sugeruje, iż pod tym hasłem kryją się wszelkiego rodzaju narzędzia komputerowe wykorzystywane w trakcie prac nad oprogramowaniem, a więc m.in. edytory graficzne,

3. METODY STRUKTURALNE – PRZYKŁADY ŚRODKÓW FORMALNYCH I NARZĘDZI WSPOMAGAJĄCYCH MODELOWANIE POJĘCIOWE INFORMACJI GEOGRAFICZNEJ



4. METODY OBIEKTOWE – PRZYKŁADY ŚRODKÓW FORMALNYCH I NARZĘDZI WSPOMAGAJĄCYCH MODELOWANIE POJĘCIOWE INFORMACJI GEOGRAFICZNEJ



FUNKCJONOWANIE MIASTA STAJE SIĘ PROSTSZE.

DZIĘKI OPROGRAMOWANIU FIRMY BENTLEY

Tak, chciałbym...

- ✓ Posiadać nieograniczony dostęp do wszystkich rodzajów oprogramowania inżynierskiego, przeznaczonego do projektowania, kartografii i zarządzania infrastrukturą techniczną* – wszystko w ramach jednej opłaty.
- ✓ Przewidywać i sprawnie zarządzać kosztami oprogramowania oraz planować budżety na ten cel.
- ✓ Regulować tylko jedną fakturę rocznie.
- ✓ Zrzucić ciężar konieczności administrowania licencjami oprogramowania.
- ✓ Wyeliminować czasochłonne procedury przetargowe.
- ✓ Ujednolicić dane zasilające środowisko prowadzenia projektów.
- ✓ Szybciej inicjować zadania projektowe i szybciej zbierać potrzebne dane.
- ✓ Przystąpić do programu, który będzie się dostosowywał do rozwoju mojej firmy.
- ✓ Uzyskać swobodę wpływu na rozbudowę otaczającego mnie świata.



X

MIEJSCE NA PODPIS

TAK, TO MOŻE BYĆ TAKIE PROSTE.

Przez wprowadzenie nowego programu – Municipal License Subscription (MLS) – przez firmę Bentley, oferującą zaawansowane rozwiązania GIS dla infrastruktury.

Program MLS oferuje urzędom administracji publicznej i samorządowej nieograniczony dostęp do pełnego zestawu, światowej klasy oprogramowania dla wszelkich zastosowań związanych z ewidencją gruntów, kartografią, projektowaniem infrastruktury technicznej, pracą grupową, zarządzaniem dokumentacją oraz sieciowym publikowaniem danych – wszystko w ramach jednej, ujednoliconej opłaty rocznej.

Wspomniana opłata roczna kalkulowana jest na podstawie ilości ludności zamieszkującej dany obszar i może być mniejsza niż 50 gr. na jednego mieszkańca. To najprostszy sposób na wygospodarowanie oszczędności.

Jak sprawić by funkcjonowanie miasta stało się prostsze? Odpowiedź znajduje się pod adresem www.bentley.com/pl/MLS

* MLS zawiera kompleksowe rozwiązania wspierające: kartografię, pomiary i modelowanie terenu, aplikacje katastralne, wizualizacje urbanistyczne, projektowanie sieci telekomunikacyjnych i światłowodowych, zarządzanie sieciami wodociagowymi i kanalizacyjnymi, architekturę i budownictwo, zarządzanie majątkiem, projektowanie i wykonawstwo instalacji przemysłowych, projektowanie dróg, linii kolejowych i tramwajowych, inżynierię lądową, pracę grupową i zarządzanie dokumentacją techniczną, sieciowe publikowanie informacji i wiele innych.

Tab. 2. CHARAKTERYSTYKA METODYK, METOD I NARZĘDZI [Traczyk 2004]

Metodyki	Metody	Narzędzia
● postulują przebieg procesu projektowania, ● wyznaczają zadania i kolejność ich wykonywania (co, kiedy i dlaczego powinno być wykonane), ● wskazują zastosowanie odpowiednich metod.	● określają sposób wykonania konkretnych zadań projektowych, ● dostarczają modeli formalnych, ● zwykle mogą być zastosowane „ręcznie” lub z użyciem odpowiednich narzędzi.	● wspomagają zastosowanie konkretnych metod, ● umożliwiają efektywną pracę, ● wykonują czynności niewymagające podejmowania decyzji (np. przekształcenia modeli), ● wspomagają podejmowanie decyzji projektowych, ● zapewniają powtarzalność czynności projektowych.

kompilatory, debugery, edytory tekstu, narzędzia harmonogramowania przedsięwzięć, arkusze kalkulacyjne. Są to zintegrowane pakiety oprogramowania, które realizują zbiory metod projektowania, dostosowanych do konkretnych metodyk.

Tradycyjnie jednak pod pojęciem CASE rozumie się narzędzia, które:

- wspomagają ogólnie rozumiane wytwarzanie oprogramowania,

- koncentrują się na fazach analizy i projektowania oraz bezpośrednim wykorzystaniu wyników tych faz w implementacji.

Narzędzia CASE pełnią w pracy twórcy systemu – inżyniera oprogramowania podobną rolę, jak narzędzia CAD/CAM w pracy inżynierów takich dziedzin, jak np. mechanika, elektrotechnika, architektura czy budownictwo. Tradycyjnie narzędzia CASE dzieli się na:

- **Upper-CASE** – narzędzia wysokiego poziomu, koncentrują się na wspomaganie wstępnych etapów budowy oprogramowania, tj. określeniu wymagań, modelowaniu i projektowaniu systemu realizującego te wymagania. Narzędzia takie nie są związane z konkretnym środowiskiem implementacji i nie wspomagają bezpośrednio implementacji systemu. Jednym z istotnych zadań narzędzi Upper-CASE w fazie analizy jest wspomaganie posługiwania się notacjami graficznymi. Stąd jednym z podstawowych modułów tych narzędzi są właśnie edytory notacji graficznych, specjalizowane pod kątem poszczególnych notacji. Narzędzia takie powinny również umożliwiać kontrolę poprawności i kompletności tworzonych modeli.

- **Lower-CASE** – narzędzia niskiego poziomu, koncentrują się na fazach projektowania i implementacji. Punktem wyjścia dla ich rozwoju była chęć ułatwienia programistom pracy nad złożonym oprogramowaniem. Narzędzia te oparte są na obserwacji, z której wyni-

ka, że notacje graficzne są bardziej naturalnym sposobem prezentowania dużych programów niż tradycyjny zapis tekstowy. Narzędzia te z reguły są ściśle związane z konkretnym środowiskiem implementacji.

- **Integrated-CASE** – narzędzia zintegrowane I-CASE, łączą w sobie możliwości narzędzi Upper-CASE i Lower-CASE. Efektywnie wspomagają wykonywanie praktycznie wszystkich faz produkcji oprogramowania [Jaszkiewicz 1997].

Do podstawowych funkcji narzędzi CASE należy zaliczyć m.in. możliwość zapanowania nad złożonością tworzonego oprogramowania, usprawnienie pracy zespołowej twórców przedsięwzięcia, tworzenie i zapisywanie modeli, jak również zależności między nimi, analizowanie zależności i wpływu zmian, kontrolę zgodności między modelami, automatyzowanie niektórych czynności projektowych, wspomaganie podejmowania decyzji projektowych i kontroli jakości projektu, kontrolę formalną poprawności i spójności opracowywanego systemu, przechowywanie historii i dostęp do dokumentacji procesu projektowego [Jaszkiewicz 1997, Traczyk 2004, Subieta 2002].

Narzędzia CASE wspomagają tzw. inżynierię w przód, czyli przejście od modelu do implementacji oraz tzw. inżynierię wstecz (reverse engineering), czyli możliwość odtworzenia modelu z poziomu kodu źródłowego.

Przykłady narzędzi CASE zostały wymienione na rys. 3 i rys. 4. Większość z nich to jedynie proste edytory graficzne, wspomagające tworzenie modeli w wybranej notacji, zgodnej z przyjętą metodyką. Tylko nieliczne (z wyżej wymienionych przykładów) programy narzędziowe zapewniają automatyzację procesu budowy profesjonalnego systemu informacyjnego, np. MS Visio Enterprise for Architects, Eclipse UML.

PODSUMOWANIE

Na zakończenie krótkie zestawienie charakterystycznych cech metodyk, technik oraz narzędzi analizy i projektowania (tabela 2).

Kilka uwag końcowych:

- metody analizy nie są prostymi algorytmami, których stosowanie przez różne osoby zapewni osiągnięcie tych samych wyników,

- proponowane metody są raczej zbiorem ogólnych wskazówek i rad,

- modele tego samego systemu stworzone przez różne osoby stosujące te same notacje i metody mogą się różnić, jednak z drugiej strony modele te nie powinny być zupełnie odmienne,

- nie ma metodyk uniwersalnych – analitycy i projektanci wybierają kombinację technik i notacji, która w danym momencie jest najbardziej użyteczna,

- notacje nie są nierozzerwalne z metodyką (np. notację UML można zastosować w metodyce Coad/Yourdon),

- narzędzia CASE nie narzucają metodyki, określają tylko notację,

- jedno narzędzie może wspomagać zarówno metodyki strukturalne, jak i obiektowe (różne notacje właściwe dla danej metodyki, rys. 1),

- nawet najlepsze metodyki i narzędzia CASE nie zapewnią jakości projektów. Klucz do sukcesu w tym przypadku stanowią zespoły doświadczonych i kompetentnych osób, dla których metodyki, notacje i narzędzia CASE służą jedynie jako istotne wspomaganie w procesie tworzenia oprogramowania, który jest procesem twórczym, zależnym od subiektywnych preferencji projektanta [Jaszkiewicz 1997, Subieta 2002].

AGNIESZKA CHOJKA

jest pracownikiem Katedry Geodezji Szczegółowej Uniwersytetu Warmińsko-Mazurskiego w Olsztynie

Bibliografia została zamieszczona wraz z I częścią artykułu (GEODETA 5/2006)