



Politechnika Wroclawska

Inżynieria oprogramowania

Jan Magott



Wprowadzenie

Literatura do języka UML

- G. Booch, J. Rumbaugh, I. Jacobson, UML przewodnik użytkownika, Seria Inżynieria oprogramowania, WNT, 2001, 2002.
- M. Fowler, UML w kropelce, wersja 2.0, LTP, 2005.
- M. Śmiątek, Zrozumieć UML 2.0, Metody modelowania obiektowego, Helion, 2005.
- S. Wrycza, B. Marcinkowski, K. Wyrzykowski, Język UML 2.0 w modelowaniu systemów informatycznych, Helion, 2005.



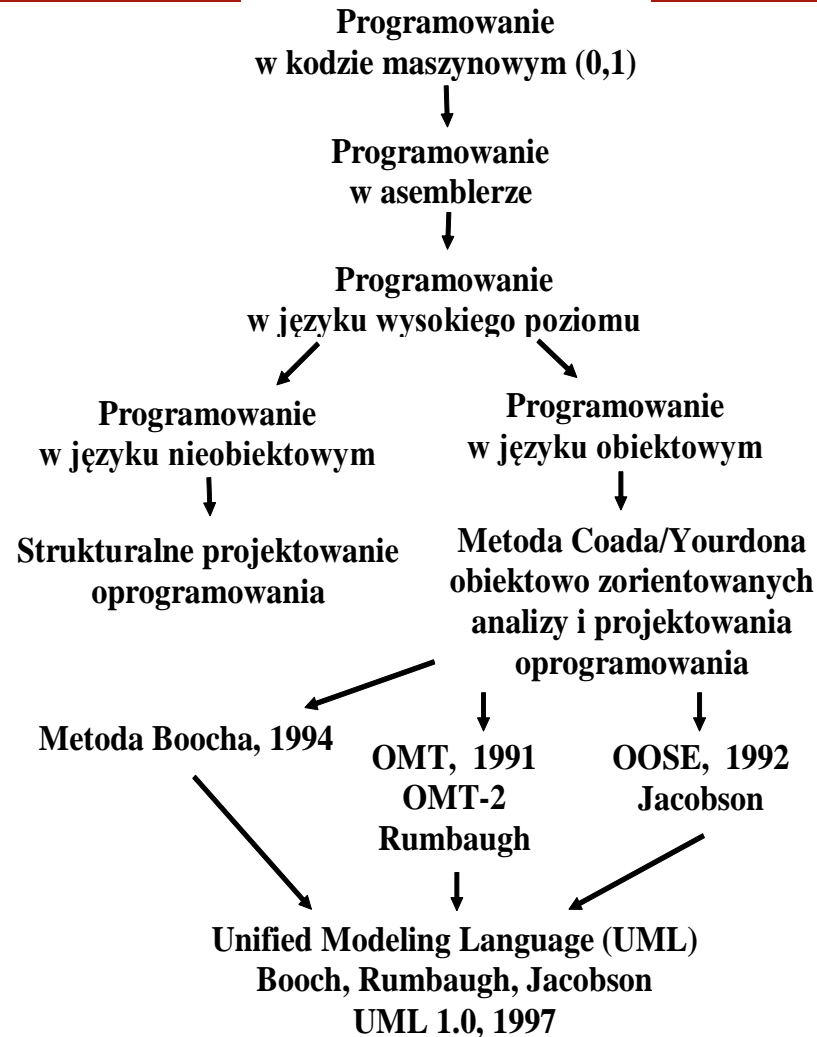
Wprowadzenie

Literatura do wzorców projektowych

- E. Gamma, R. Helm, R. Johnson, J. Vlissides, **Wzorce projektowe, Elementy oprogramowania obiektowego wielokrotnego użytku, WNT, Warszawa, 2005.**
- Shalloway, J. R. Trott, **Projektowanie obiektowo zorientowane, Wzorce projektowe, wydanie II, Helion, 2005.**



Wprowadzenie





Wprowadzenie

Inżynieria oprogramowania

UML



Inżynieria systemowa

SysML



Wprowadzenie

Znaczenie języka UML

- Język komunikacji z użytkownikiem
 - Na etapie specyfikacji wymagań stosowane są diagramy i opis w języku naturalnym.
- Język wielu perspektyw:
 - Wymagań użytkownika,
 - Struktury abstrakcyjnej,
 - Dynamiki abstrakcyjnej,
 - Architektury implementacji (komponenty programowe),
 - Architektury sprzętowo-programowej - wdrożenia (rozmieszczenia komponentów programowych w węzłach fizycznych).
- Język dokumentacji projektu
- Język projektowania poprzez modelowanie



Wprowadzenie

Źródło:

A. Bobkowska, M. Gala,

Konsekwencje zastosowania modelowania w projektach informatycznych - badanie z udziałem praktyków,

w: J. Górski, C. Orłowski, Inżynieria oprogramowania w procesach integracji systemów informatycznych, PWNT, Gdańsk, 2010, str. 49-56



Wprowadzenie

Korzyści z zastosowania metod modelowania w inżynierii oprogramowania:

- Pomoc w radzeniu sobie ze złożonością systemu, dzięki mechanizmowi abstrakcji, umożliwiającemu prezentację tylko tych elementów, które są istotne z danego punktu widzenia
- Możliwość przewidywania własności i skutków działania oprogramowania na podstawie modeli, dzięki czemu modele stosowane są do analizy, weryfikacji, walidacji, oceny i porównania różnych wersji systemu
- Wspomaganie komunikacji w zespole i z klientem dzięki wspólnym modelom koncepcyjnym
- Standardowa notacja i zwiększenie precyzji dokumentacji systemu
- Możliwość automatycznego przekształcania modeli, np. generacji dokumentacji, generacji kodu, sprawdzaniu poprawności syntaktycznej, spójności i innych własności modeli.



Wprowadzenie

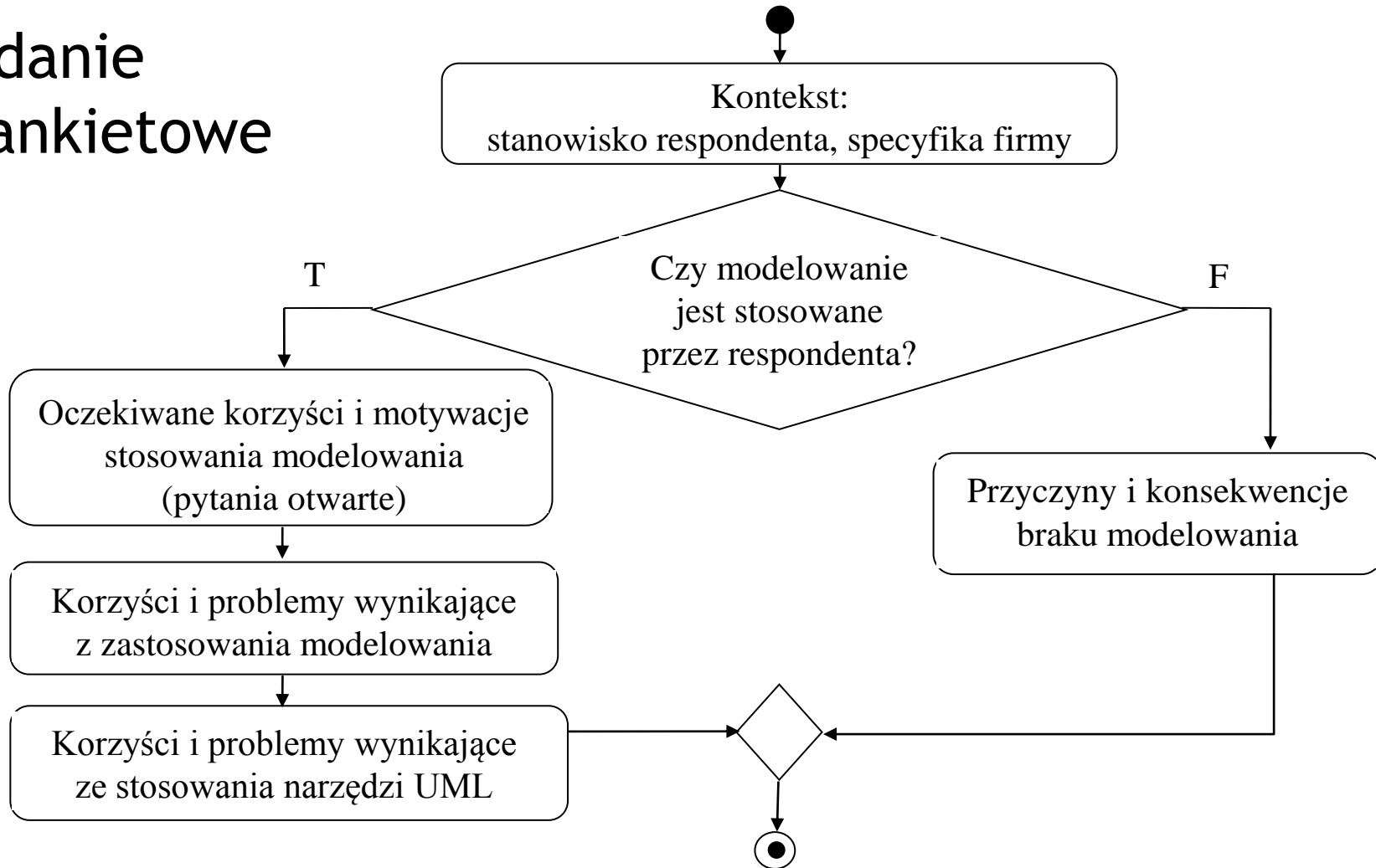
Czynniki powodujące rezygnację z modelowania w praktyce:

- Ograniczenia czasowe nałożone na termin wykonania
- Opracowywanie modeli nie powoduje bezpośredniego przyrostu kodu programu



Wprowadzenie

Badanie ankietowe





Wprowadzenie

Badanie ankietowe

Adres systemu ankietowania www.webankieta.pl

- Ankieta miała charakter anonimowy.
- Prośba o wypełnienie wysłana pod 285 adresów email należących do 170 przedsiębiorstw informatycznych działających na terenie Polski.
- Wzięło udział 66 respondentów (współczynnik odpowiedzi 23%) w tym 38 wypełniło kwestionariusz w pełni, a 28 - częściowo.



Wprowadzenie

Poziom stosowania modelowania rośnie wraz ze:

- wzrostem złożoności systemu,
- wzrostem liczebności zespołu projektowego,
- wzrostem sformalizowania metodyki wytwarzania oprogramowania.

Poziom stosowania modelowania:

- Metodyka RUP (Rational Unified Process) opracowana przez twórców języka UML- 92% respondentów
- Podejście tradycyjne - 65%
- Podejście zwinne - 51%



Wprowadzenie

Wg ankiety

- Wśród języków modelowania najczęściej stosowany jest UML lub jego podzbiór (87%)
- W pozostałych przypadkach (13%) zadeklarowano stosowanie innych metod (BPMN, SysML, DSM)



Wprowadzenie

Przykłady szeroko stosowanych w praktyce języków modelowania przepływu prac i procesów biznesowych:
BPEL - Business Process Execution Language,
BPMN - Business Process Modeling Notation,
Diagramy czynności (aktywności) języka UML.

Według porównania 14 komercyjnych produktów modelowania przepływu prac i procesów biznesowych, BPMN i diagramy czynności języka UML są jednymi z trzech najlepszych pod względem mocy wyrażania przepływu sterowania.

Źródło: N. Russel, A.H.M. ter Hofstede, W.M.P. van der Aalst, N. Mulyar, Workflow control-flow patterns, A revised view.



Wprowadzenie

DSM - Domain Specific Modelling

DSL - Domain Specific Language



Wprowadzenie

Obszary zastosowania modelowania:

- Modelowanie systemu (86%)
- Prezentacja wymagań (83%)
- Modelowanie procesów biznesowych (66%)
- Modelowanie architektury (40%)

Najczęściej modelowanie stosowane jest przez analityków



Wprowadzenie

Oczekiwane korzyści i motywacje stosowania modelowania (66 respondentów, 38 wypełniło kwestionariusz w pełni):

1. Jednoznaczny i spójny opis systemu (10 respondentów)
2. Komunikacja w ramach zespołu (9)
3. Modelowania statyczne systemu (8)
4. Modelowanie biznesowe systemu (5)
5. Modelowanie zachowania systemu (zamiast opisu słownego) (5)
6. Gromadzenie wiedzy (repozytorium modeli systemu) (5)
7. Zarządzanie wymaganiami / zmianą (5)
8. Redukcja złożoności problemu/systemu (5)
9. Ułatwienie przepływu wiedzy (5)
10. Standaryzacja dokumentacji/formalizacja opisu systemu (5)
11. Komunikacja z klientem (5)
12. Redukcja liczby błędów (szczególnie we wstępnych fazach) (3)
13. Poprawa pielęgnowalności (3)
14. Generacja kodu źródłowego (2)
15. Szybsze wprowadzenie nowych osób do projektu (2)
16. Wymaganie ze strony klienta (model jako produkt) (1)



Wprowadzenie

Korzyści **wynikające** z zastosowania **modelowania**

	nie	raczej nie	raczej tak	tak	nie wiem	Liczba odpowiedzi
redukcja złożoności problemu/systemu	0 (0,0 %)	7 (24,14 %)	13 (44,83 %)	8 (27,59 %)	1 (3,45 %)	29
poprawa czytelności dokumentacji	0 (0,0 %)	1 (3,45 %)	5 (17,25 %)	22 (75,87 %)	1 (3,45 %)	29
poprawa komunikacji w zespole	0 (0,0 %)	2 (6,9 %)	7 (24,14 %)	20 (68,97 %)	0 (0,0 %)	29
poprawa komunikacji z klientem	3 (10,35 %)	3 (10,35 %)	9 (31,04 %)	13 (44,83 %)	1 (3,45 %)	29
ułatwione przewidywanie zachowania i właściwości systemu	1 (3,45 %)	4 (13,8 %)	8 (27,59 %)	16 (55,18 %)	0 (0,0 %)	29
redukcja liczby błędów	0 (0,0 %)	6 (20,69 %)	10 (34,49 %)	8 (27,59 %)	5 (17,25 %)	29
wzrost jakości oprogramowania	1 (3,45 %)	1 (3,45 %)	9 (31,04 %)	16 (55,18 %)	2 (6,9 %)	29
wzrost wykorzystania wzorców (projektowych, implementacyjnych)	3 (10,35 %)	7 (24,14 %)	10 (34,49 %)	5 (17,25 %)	4 (13,8 %)	29
koncentracja na aspektach dziedziny problemu, a nie implementacji	0 (0,0 %)	4 (13,8 %)	10 (34,49 %)	13 (44,83 %)	2 (6,9 %)	29

Liczba respondentów którzy odpowiedzieli na to pytanie: 29



Wprowadzenie

Problemy wynikające z zastosowania modelowania

Nie stwierdzono:

- Występowania problemów ze zrozumieniem wykorzystywanej notacji przez zespół wytwórczy
- Niejasnej formy prezentacji modelu
- Braku spójności modeli
- Nieodpowiedniego poziomu szczegółowości modeli

**Występowanie problemów ze zrozumieniem notacji przez klienta -
45% ankietowanych odpowiedziało „trudno powiedzieć”**

Korzyści **wynikające** ze stosowania **narzędzi UML**

	nie	raczej nie	raczej tak	tak	nie wiem	Liczba odpowiedzi
wzrost produktywności	0 (0,0 %)	5 (21,74 %)	12 (52,18 %)	4 (17,4 %)	2 (8,7 %)	23
wzrost jakości oprogramowania	0 (0,0 %)	1 (4,35 %)	12 (52,18 %)	8 (34,79 %)	2 (8,7 %)	23
poprawa jakości dokumentacji	0 (0,0 %)	1 (4,35 %)	7 (30,44 %)	14 (60,87 %)	1 (4,35 %)	23
poprawa komunikacji w zespole	0 (0,0 %)	2 (8,7 %)	8 (34,79 %)	13 (56,53 %)	0 (0,0 %)	23
poprawa komunikacji z klientem	2 (8,7 %)	3 (13,05 %)	6 (26,09 %)	7 (30,44 %)	5 (21,74 %)	23
spadek kosztu realizacji projektu	2 (8,7 %)	7 (30,44 %)	7 (30,44 %)	0 (0,0 %)	7 (30,44 %)	23
skrócenie czasu realizacji projektu	0 (0,0 %)	4 (17,4 %)	13 (56,53 %)	1 (4,35 %)	5 (21,74 %)	23
gromadzenie wiedzy projektowej	0 (0,0 %)	1 (4,35 %)	5 (21,74 %)	16 (69,57 %)	1 (4,35 %)	23
poprawa dostępności informacji	0 (0,0 %)	4 (17,4 %)	8 (34,79 %)	11 (47,83 %)	0 (0,0 %)	23
zwiększone ponowne użycie (reusability)	1 (4,35 %)	4 (17,4 %)	13 (56,53 %)	2 (8,7 %)	3 (13,05 %)	23
wygodny przepływ informacji (modeli, dokumentów, danych) pomiędzy narzędziami	3 (13,05 %)	8 (34,79 %)	2 (8,7 %)	8 (34,79 %)	2 (8,7 %)	23
poprawiona pielęgnowalność (czas poprawy błędu, czas dodania nowej funkcjonalności)	1 (4,35 %)	3 (13,05 %)	7 (30,44 %)	8 (34,79 %)	4 (17,4 %)	23
koncentracja na aspektach dziedziny problemu, a nie na implementacji	0 (0,0 %)	4 (17,4 %)	10 (43,48 %)	7 (30,44 %)	2 (8,7 %)	23
automatyzacja pewnych czynności (wytworzenia dokumentacji, kodu, wykonywanie testów systemu)	1 (4,35 %)	0 (0,0 %)	9 (39,14 %)	13 (56,53 %)	0 (0,0 %)	23
poprawa koordynacji działań zespołu	2 (8,7 %)	6 (26,09 %)	7 (30,44 %)	7 (30,44 %)	1 (4,35 %)	23
specjalizacja pracowników	5 (21,74 %)	6 (26,09 %)	6 (26,09 %)	3 (13,05 %)	3 (13,05 %)	23
poprawa standaryzacji elementów procesu wytwórczego	1 (4,35 %)	2 (8,7 %)	10 (43,48 %)	8 (34,79 %)	2 (8,7 %)	23
poprawa kontroli realizacji projektu	1 (4,35 %)	8 (34,79 %)	8 (34,79 %)	5 (21,74 %)	1 (4,35 %)	23
rozwój pracowników (umiejętności analityczne, modelowanie)	1 (4,35 %)	5 (21,74 %)	9 (39,14 %)	7 (30,44 %)	1 (4,35 %)	23
wzrost satysfakcji z wykonywanej pracy	2 (8,7 %)	5 (21,74 %)	5 (21,74 %)	6 (26,09 %)	5 (21,74 %)	23



Wprowadzenie

Przyczyny braku modelowania

20 z 66 respondentów stwierdziło, że **nie stosuje modelowania**. 14 respondentów (70%) w pełni odpowiedziało na pytania tej ścieżki.

Przedstawiciele tej grupy realizują krótkoterminowe, niskobudżetowe projekty, w których uczestniczy mały zespół projektowy.

Najczęściej metody modelowania nie są stosowane ze względu na:

- ograniczenia projektowe (krótkie terminy, brak dostępności zasobów)
- brak zdefiniowanego procesu wytwórczego z zastosowaniem metod modelowania.

W grupie 14 respondentów, 10 z nich (71%) zadeklarowało, że stosuje metodyki zwinne.



Wprowadzenie

Konsekwencje braku modelowania

	nie	tak	trudno powiedzieć	Liczba odpowiedzi
problemy ze zrozumieniem systemu	4 (28,58 %)	5 (35,72 %)	5 (35,72 %)	14
niska jakość dokumentacji lub jej brak	4 (28,58 %)	8 (57,15 %)	2 (14,29 %)	14
problemy z komunikacją w grupie projektowej	10 (71,43 %)	1 (7,15 %)	3 (21,43 %)	14
problemy z komunikacją z klientem	3 (21,43 %)	4 (28,58 %)	7 (50,0 %)	14

Liczba respondentów którzy odpowiedzieli na to pytanie: 14

Fakt, że 71% respondentów zaprzeczyło istnieniu problemów z komunikacją w grupie, można wytłumaczyć stosowaniem innych sposobów komunikacji w zespole.

Koniec powołań na źródło:

A. Bobkowska, M. Gala, Konsekwencje zastosowania modelowania w projektach informatycznych - badanie z udziałem praktyków, ...



Wprowadzenie

Przyczyny tworzenia narzędzi programistycznych wspierających rysowanie i analizę diagramów języka UML:

- Tylko dla prostych diagramów, rysowanie i analiza nie są uciążliwe,
- Złożoność analizy zgodności między diagramami,
- Możliwości algorytmizacji analizy diagramów.



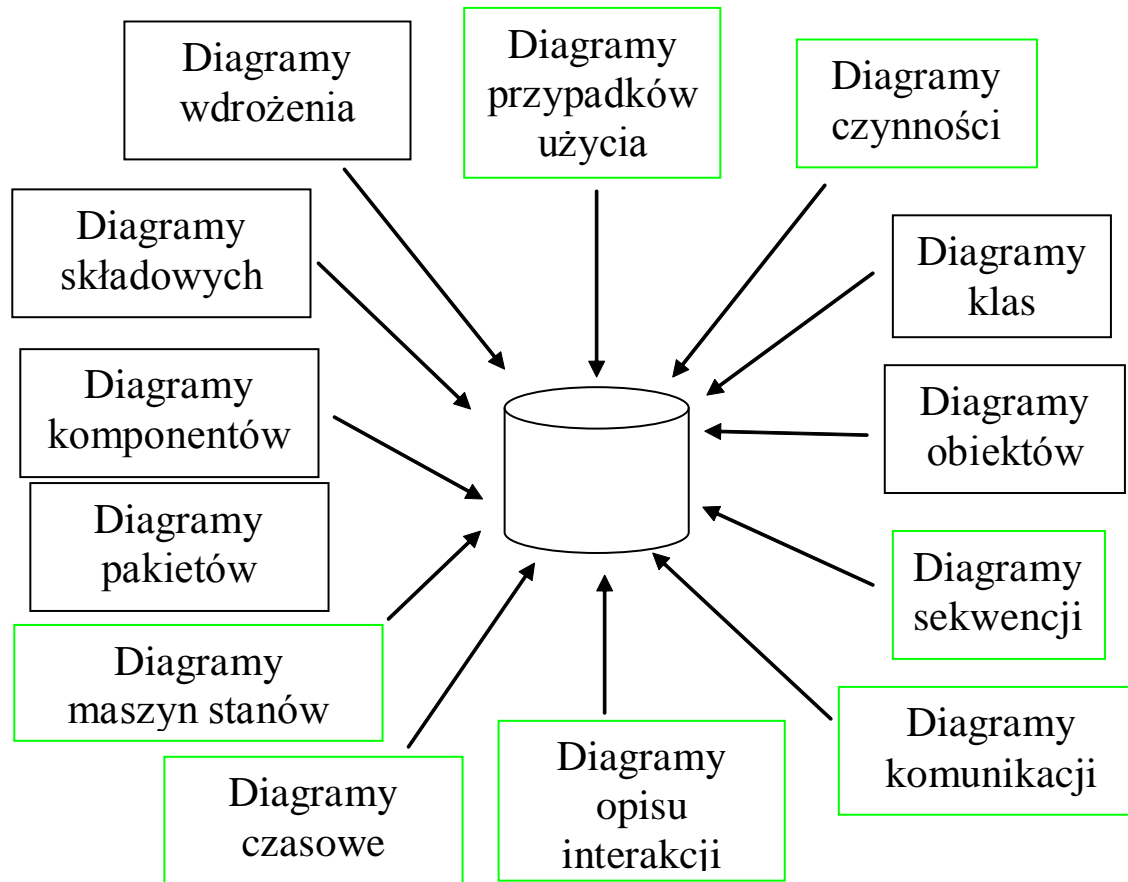
Wprowadzenie

Wymagania stawiane nowoczesnym narzędziom CASE (Computer Aided Software Engineering):

- Rysowanie diagramów ze sprawdzaniem ich poprawności,
- Pełnienie roli repozytorium,
- Umożliwienie nawigacji po komponentach systemu przy pracy z różnymi kategoriami modeli,
- Umożliwienie jednoczesnej pracy wielu użytkowników na tym samym modelu,
- Generowanie kodu,
- Inżynieria odwrotna,
- Integrowanie z innymi narzędziami, np. w celu edycji i kompilacji kodu uzupełniającego szkielet kodu, testowania,
- Praca ze wszystkimi poziomami abstrakcji systemu, od wysokiego poziomu do poziomu kodu,
- Wymiana fragmentów modelu z innymi narzędziami.



Wprowadzenie



Repozytorium: — struktura, — dynamika



Wprowadzenie

Zadania, dla których narzędzie CASE wykorzystuje repozytorium:

- Sprawdzanie zgodności diagramu i zgodności między diagramami,
- Wskazywanie uchybień modeli,
- Sporządzanie dokumentów z projektu,
- Stosowanie elementów projektu w innym projekcie.