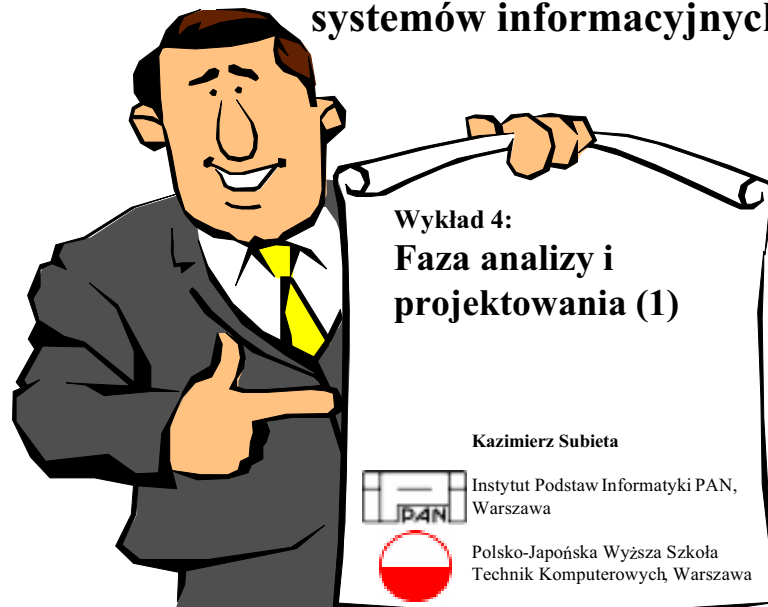
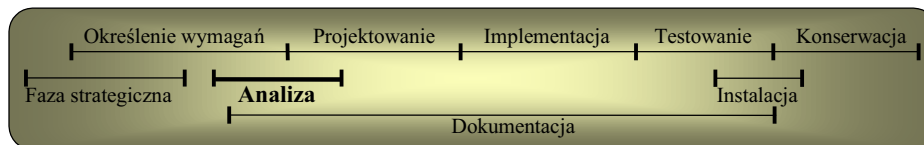


Wytwarzanie, integracja i testowanie systemów informacyjnych



K.Subieta. Wytwarzanie, integracja i testowanie SI, Wykład 4, Folia 1

Faza analizy



- ★ Celem fazy **analizy** jest udzielenie odpowiedzi na pytanie:
Co i przy jakich ograniczeniach system ma robić?
Wynikiem tej fazy jest zbiór wymagań, czyli opis sposobu implementacji.
- ★ W odróżnieniu, celem fazy **projektowania** jest udzielenie odpowiedzi na pytanie:
Jak system ma być zaimplementowany?
Wynikiem jest projekt oprogramowania, czyli opis sposobu implementacji.
- ★ Celem fazy **analizy** jest udzielenie odpowiedzi na pytanie:
Jak system ma działać?
Wynikiem jest logiczny model systemu, opisujący sposób realizacji przez system postawionych wymagań, lecz abstrahujących od szczegółów implementacyjnych.

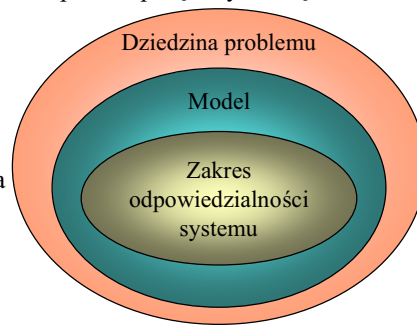
Fazie analizy był poświęcony poprzedni wykład nt OMT.

K.Subieta. Wytwarzanie, integracja i testowanie SI, Wykład 4, Folia 2

Model analityczny

Z reguły wykracza poza zakres odpowiedzialności systemu. Przyczyny:

- ✦ Ujęcie w modelu pewnych elementów dziedziny problemu nie będących częścią systemu czyni model bardziej zrozumiałym. Przykładem jest ujęcie w modelu systemów zewnętrznych, z którymi system ma współpracować.
- ✦ Na etapie modelowania może nie być jasne, które elementy modelu będą realizowane przez oprogramowanie, a które w sposób sprzętowy lub ręcznie.
- ✦ Dostępne środki mogą nie pozwolić na realizację systemu w całości. Celem analizy może być wykrycie tych fragmentów dziedziny problemu, których wspomaganie za pomocą oprogramowania będzie szczególnie przydatne.



K.Subieta. Wytwarzanie, integracja i testowanie SI, Wykład 4, Folia 3

Obiektowe i strukturalne metodyki analizy

Metodyki strukturalne - metody tradycyjne rozwijane od lat 60-tych. Łączy składowe pasywne (opis danych) oraz aktywne (wykonywanie operacji).

Metodyki obiektowe - rozwijane od lat 80-tych, oparte na wyróżnianiu obiektów łącznie z operacjami.

Analiza strukturalna (*structured analysis*) rozpoczyna się od budowy dwóch różnych modeli systemu: **modelu danych** oraz **modelu funkcji**. Te dwa modele są integrowane. Wynikiem jest **model przepływu danych** (*data flow model*). Wadą podejścia są trudności w zintegrowaniu modeli.

Ostatnio bardziej popularne są **metodyki obiektowe**, chociaż nie jest do końca pewne, czy jest to trwała tendencja, czy tylko sprawa pewnej mody. Przykładem metodyki obiektowej jest OMT (obecnie najbardziej popularna). Metodyki posługują się **notacją**, z reguły graficzną, którą dość często utożsamia się z samą metodyką (nieszłusznie). Przykładem notacji jest UML. Można go stosować do dowolnej metodyki.

K.Subieta. Wytwarzanie, integracja i testowanie SI, Wykład 4, Folia 4

Notacje wykorzystywane w analizie i projektowniu

Rodzaje notacji

- Język naturalny
- Notacje graficzne
- Specyfikacje - częściowo ustrukturalizowany zapis tekstowy i numeryczny

Szczególne znaczenie mają notacje graficzne.

Inżynieria oprogramowania wzoruje się tu na innych dziedzinach techniki, takich jak elektronika i mechanika. Zalety notacji graficznych potwierdzają badania psychologiczne.

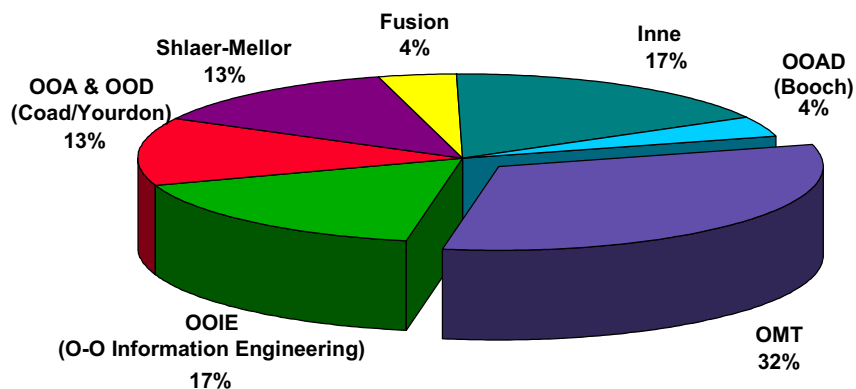
Funkcje notacji

- Narzędzie pracy analityka i projektanta, zapis i analiza pomysłów
- Współpraca z użytkownikiem
- Komunikacja z innymi członkami zespołu
- Podstawa implementacji oprogramowania
- Zapis dokumentacji technicznej

Notacje powinny być przejrzyste, proste, precyzyjne, łatwo zrozumiałe, umożliwiające modelowanie złożonych zależności.

K.Subieta. Wytwarzanie, integracja i testowanie SI, Wykład 4, Folia 5

Jakie metodyki obiektowe są używane?



Źródło: Gartner Group - 1995

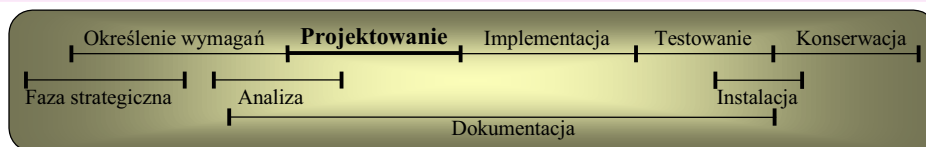
K.Subieta. Wytwarzanie, integracja i testowanie SI, Wykład 4, Folia 6

Różnice pomiędzy metodykami

- ✦ Podejścia proponowane przez różnych autorów różnią się częściowo, nie są jednak ze sobą sprzeczne.
- ✦ Poszczególne metodyki zawierają elementy rzadko wykorzystywane w praktyce (np. model przepływu danych w OMT).
- ✦ Notacje proponowane przez różnych autorów nie są konieczne nierozdzielne z samą metodyką. Np. notacji OMT można użyć dla metodyki Coad/Yourdon.
- ✦ Nie ma metodyk całkowicie uniwersalnych. W praktyce analitycy wybierają kombinację technik, metodyk i notacji, która jest w danym momencie najbardziej przydatna. Wybór może być jednak utrudniony istniejącą rutyną w danej firmie.
- ✦ Narzędzia CASE nie narzucają metodyki; raczej, określają one tylko notację. Twierdzenia, że jakieś narzędzie CASE "jest oparte" na konkretnej metodyce jest najczęściej wyłącznie reklamowym (mocno przesadzonym) hasłem.

K.Subieta. Wytwarzanie, integracja i testowanie SI, Wykład 4, Folia 7

Projektowanie



- ✦ Celem projektowania jest opracowanie szczegółowego opisu implementacji systemu.
- ✦ W odróżnieniu od analizy, w projektowaniu dużą rolę odgrywa środowisko implementacji. Projektanci muszą więc posiadać dobrą znajomość języków, bibliotek, i narzędzi stosowanych w trakcie implementacji.
- ✦ Dążenie do tego, aby struktura projektu zachowała ogólną strukturę modelu stworzonego w poprzednich fazach (analizie). Niewielkie zmiany w dziedzinie problemu powinny implikować niewielkie zmiany w projekcie.
- ✦ Wykorzystanie idei programowania strukturalnego i obiektowego.

K.Subieta. Wytwarzanie, integracja i testowanie SI, Wykład 4, Folia 8

Zadania wykonywane w fazie projektowania

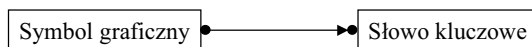
- ✦ Uszczegółowienie wyników analizy. Projekt musi być wystarczająco szczegółowy aby mógł być podstawą implementacji. Stopień szczegółowości zależy od poziomu zaawansowania programistów.
- ✦ Projektowanie składowych systemów nie związanych z dziedziną problemu
- ✦ Optymalizacja systemu
- ✦ Dostosowanie do ograniczeń i możliwości środowiska implementacji
- ✦ Określenie fizycznej struktury systemu.

K.Subieta. Wytwarzanie, integracja i testowanie SI, Wykład 4, Folia 9

Techniki obiektowe w projektowaniu

W projektowaniu często pomocne są specjalne notacje, jako uzupełnienie do notacji stosowanych w analizie.

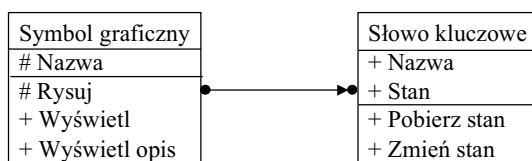
Związek skierowany: oprócz zaznaczenia związku zaznacza się kierunek przesyłania komunikatów. Np. w systemie SIG obiekty klasy "Symbol graficzny" wysyłają komunikaty do obiektów "Słowo kluczowe". Jest to jeden ze scenariuszy; w innym może być inaczej.



Symbole dostępu do pól i metod:

Jest to związane z konwencją C++, gdzie dostęp może być:

- (+) **publiczny** - dla wszystkich funkcji i metod
- (#) **zabezpieczony** (*protected*) - dostęp do metod danej klasy oraz jej specjalizacji
- (-) **prywatny** - dostęp tylko dla funkcji danej klasy



K.Subieta. Wytwarzanie, integracja i testowanie SI, Wykład 4, Folia 10

Dodatkowe elementy notacji

Wiele z nich występuje w innych metodykach, np. w metodyce Booča:

- ◆ **Wzorce klas** (*class templates*)
- ◆ **Metaklasy**, tj. klasy zawierające pola i metody dotyczące klasy jako całości a nie pojedynczych obiektów, np. pola i metody statyczne.
- ◆ **Wolne funkcje** nie będące metodami żadnej z klas.
- ◆ **Sposoby widoczności obiektu** do którego wysyłany jest komunikat. Obiekt ten może być widoczny, gdyż znajduje się w tym samym zakresie, jest przekazany przez parametr lub jest polem klasy, której metoda wysyła komunikat.

K.Subieta. Wytwarzanie, integracja i testowanie SI, Wykład 4, Folia 11

Uszczegółowianie wyników analizy (1)

Uszczegółowianie poprzez podanie reguł odwzorowanie notacji w struktury języka programowania.

Dane z analizy:

Adres
Ulica +
Numer domu +
Numer mieszkania +
Miasto +
Kod

Dane osobowe
Imię +
Nazwisko +
Adres



Realizacja w C/C++:

```
typedef struct {  
    char Ulica[30];  
    char NumerDomu[10];  
    char NumerMieszkania[10];  
    char Miasto[30];  
    char Kod[5];  
} TypAdres;
```

```
typedef struct {  
    char Imię[30];  
    char Nazwisko[30];  
    TAdres Adres;  
} TypDaneOsobowe;
```

K.Subieta. Wytwarzanie, integracja i testowanie SI, Wykład 4, Folia 12

Uszczegółowianie wyników analizy (2)

Uszczegółowianie metod

- ✦ Podanie nagłówek metod oraz ich parametrów.
- ✦ Określenie, które z metod będą realizowane jako funkcje wirtualne (pożno wiązane) a które jako zwyczajne funkcje (wiązane statyczne).
- ✦ Zastąpienie niektórych prostych metod bezpośrednim dostępem do atrybutów. Np. metody PobierzNazwisko, UstawNazwisko, etc.
- ✦ Zastąpienie niektórych atrybutów redundantnych przez odpowiednie metody, np. Wiek = BieżącaData - DataUrodzenia;
KwotaDochodu = KwotaPrzychodu - KwotaKosztów;

K.Subieta. Wytwarzanie, integracja i testowanie SI, Wykład 4, Folia 13

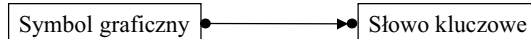
Uszczegółowianie wyników analizy (3)

Określenie sposobów implementacji związków (asocjacji)

Związki można zaimplementować na wiele sposobów, z reguły poprzez wprowadzenie dodatkowych atrybutów (pól). Mogą one być następujące:

- obiekty powiązanej klasy
- wskaźniki (referencje) do obiektów powiązanej klasy
- identyfikatory obiektów powiązanej klasy
- klucze kandydujące obiektów powiązanej klasy

W zależności od przyjętego sposobu oraz od liczności związków (1:1, 1:n, n:1, m:n) możliwe są bardzo różne deklaracje w przyjętym języku programowania.



Tablica obiektów: → TypS³owoKluczowe S³owaKluczowe[100];
Lista wskaźników: → list< TypS³owoKluczowe *> S³owaKluczowe;
Tablica wskaźników: → **char** * WskaźnikiS³ówKluczowych[100];

Dodatkowe reguły dla transformacji schematów obiektowych na relacyjne.

K.Subieta. Wytwarzanie, integracja i testowanie SI, Wykład 4, Folia 14

Techniki/diagramy strukturalne (1)

structure charts/diagrams

Moduł: aktywna składowa programu, tj. procedura lub funkcja (lub ich zestaw).

Drukuj zeznanie

Moduł biblioteczny: gotowa procedura lub funkcja wykorzystywana w systemie.

Moduł biblioteczny

Dane: relacja w bazie danych, plik lub zmienne programu

Zeznanie podatkowe

Wywołanie (call): wywołanie przez pewien moduł innego modułu.

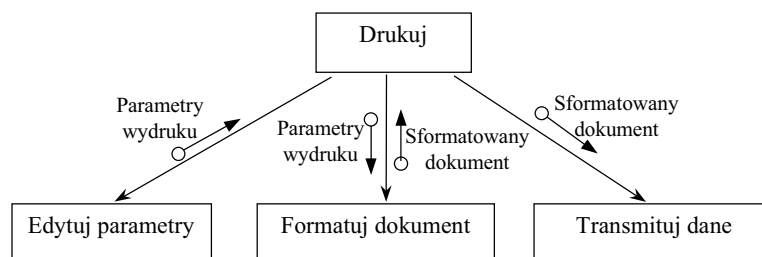
Ewidencja zeznań
podatkowych

Drukuj zeznanie

K.Subieta. Wytwarzanie, integracja i testowanie SI, Wykład 4, Folia 15

Techniki/diagramy strukturalne (2)

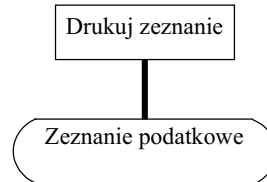
Flagi przepływu danych: z wywołaniem modułu może być związany przepływ danych z modułu wywołującego do wywoływanego i odwrotnie. Pierwszy rodzaj odpowiada parametrom wejściowym, drugi wynikowi i parametrom wyjściowym.



K.Subieta. Wytwarzanie, integracja i testowanie SI, Wykład 4, Folia 16

Techniki/diagramy strukturalne (3)

Korzystanie z danych:



Diagramy strukturalne formatuje się z reguły tak, aby moduły wyższego poziomu - moduły wywołujące - znajdowały się **powyżej** modułów niższego poziomu - modułów wywoływanych.

Diagramy strukturalne są uszczegółowieniem **diagramów przepływu danych**.

- ✦ Moduł odpowiadający procesowi wyższego poziomu wywołuje moduł będący źródłem danych, a następnie moduł będący odbiorcą danych.
- ✦ Moduł odpowiadający procesowi wyższego poziomu wywołuje moduł będący źródłem danych, który z kolei wywołuje moduł będący odbiorcą danych.
- ✦ Moduł odpowiadający procesowi wyższego poziomu wywołuje moduł będący odbiorcą danych, który z kolei wywołuje moduł będący źródłem danych.

Techniki/diagramy strukturalne (4)

