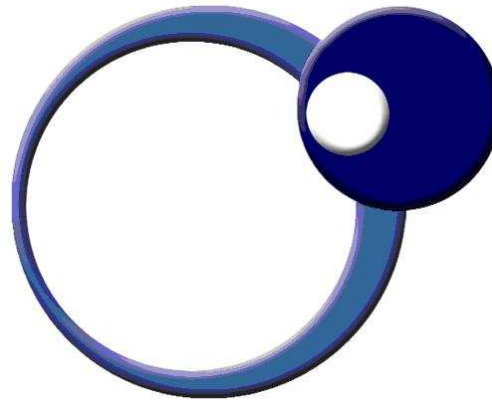
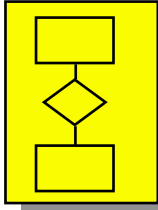


Wprowadzenie do analizy systemów

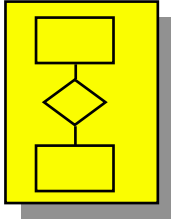


Grupa ANTARES

INFORMATION TECHNOLOGY CONSULTANTS



•
W P R O W A D Z E N I E
D O
A N A L I Z Y
S T R U K T U R A L N E J



Analiza strukturalna

Wykaz literatury

- Ed Yourdon -

Modern Structured Analysis

- Lary Shiller -

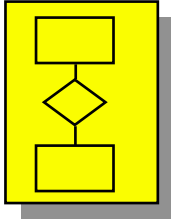
Software Excellence

- Martin E.Modell -

A Professional's Guide to System Analysis

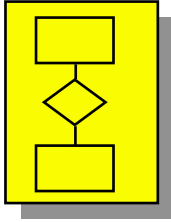
- Ed Yourdon -

Managing the System Life Cycle



Wprowadzenie do analizy strukturalnej

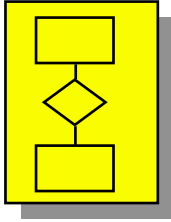
- 2.1. Metody strukturalne
- 2.2. Cele systemu
- 2.3. Narzędzia używane do analizy
- 2.4. Lista zdarzeń
- 2.5. Diagramy przepływu danych
- 2.6. Diagramy encji i relacji
- 2.7. Słownik
- 2.8. Reguły spójności i poprawności



Metody strukturalne

Metody strukturalne

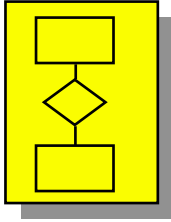
- Sposób na opis, budowę i rozwój systemu.
- Stosuje się je do średnich, dużych i bardzo dużych systemów.



Metody strukturalne

Historia

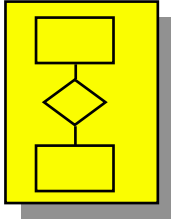
- I okres 1970 - 1980
– tradycyjna analiza strukturalna
- II okres 1981 - 1990
– nowoczesna analiza strukturalna i analiza SA/RT
- III okres po 1991



Metody strukturalne

Historia

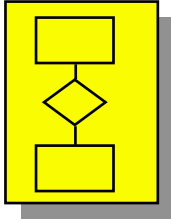
- I okres 1970 - 1980
 - MEYERS, YOURDON, GANE, SARSON, DE MARCO, JACKSON, CONSTANTINE, PAGE-JONES to nazwiska czołowych twórców analizy strukturalnej zwanej tradycyjną.



Metody strukturalne

Historia

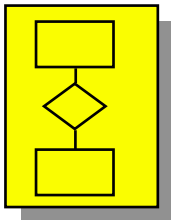
- II okres 1981 - 1990
 - MC MENAMIN, PALMER, HATLEY, PIRBHAI, WARD-MELLOR, YOURDON
to nazwiska czołowych twórców analizy strukturalnej zwanej nowoczesną analizą strukturalną oraz analizą SA/RT.



Metody strukturalne

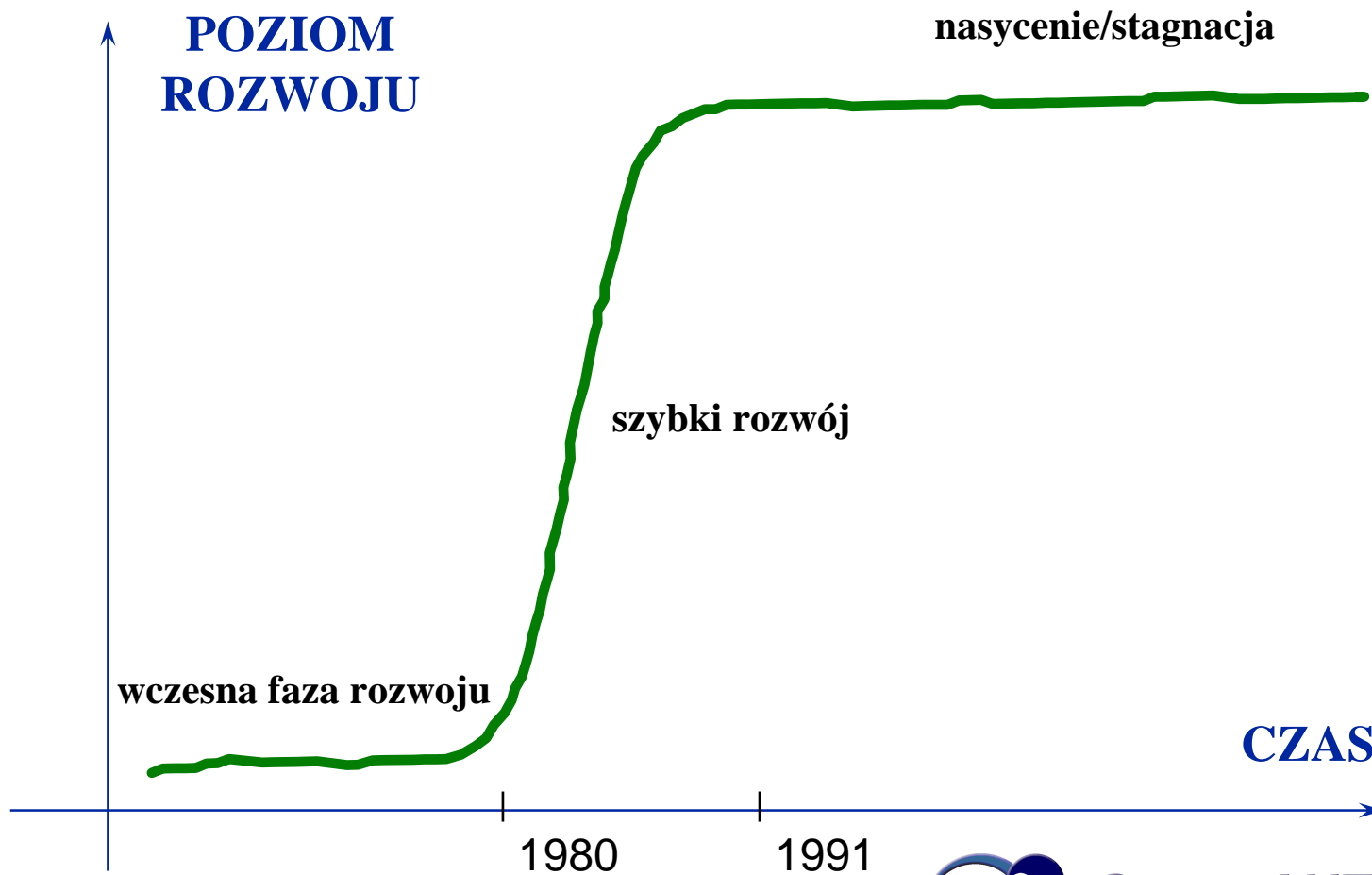
Historia

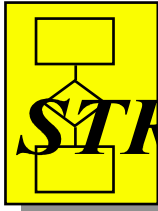
- III okres - po 1990 r.
 - Odchodzi się od metod strukturalnych.
Rozwijają się metody obiektowe.



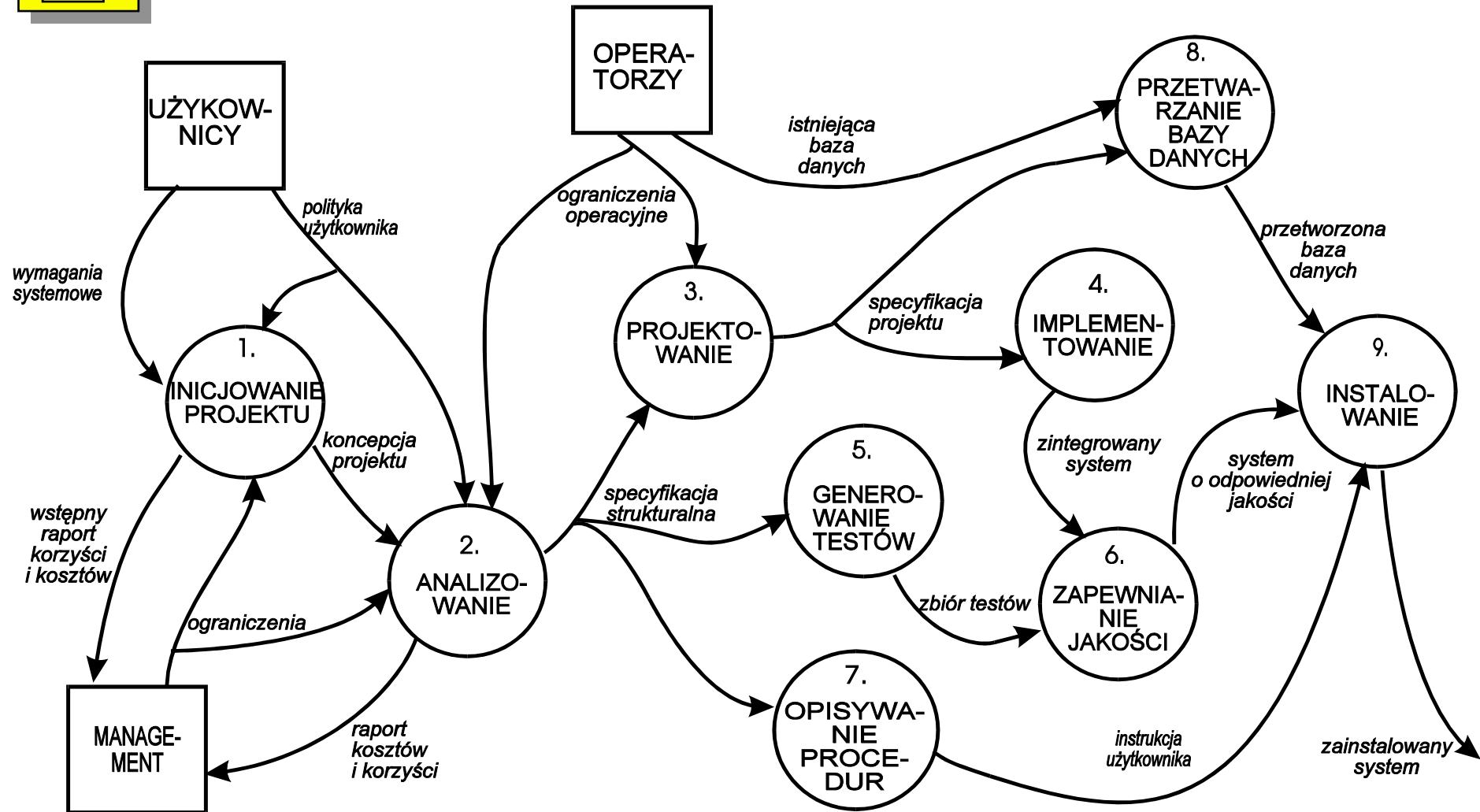
Analiza strukturalna

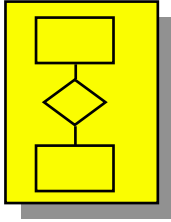
Ewolucja





STRUKTURALNY CYKL ŻYCIA PROJEKTU

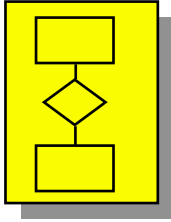




Metody strukturalne

Analiza

to studium problemu przed podjęciem działania. [DeMarco, 1978]

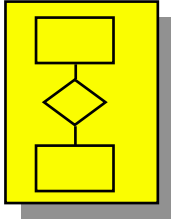


Metody strukturalne

Analiza strukturalna

Opis świata z punktu widzenia przepływu danych.

Proces, podczas którego budujemy funkcjonalną wizję przyszłego systemu informatycznego.



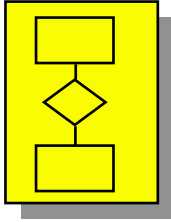
Metody strukturalne

Analiza strukturalna

- Sposób tworzenia

CO

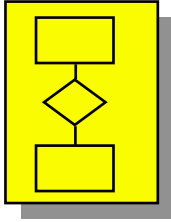
~~jak~~



Analiza strukturalna

Analiza dziedziny

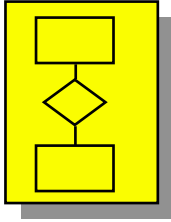
- obserwacja istniejącego stanu
- analiza dokumentów użytkownika (np. globalnej koncepcji)
- analiza badań przeprowadzonych wcześniej w podobnych zagadnieniach
- rozmowy z ekspertami z dziedziny
- wykorzystanie własnej wiedzy



Metody strukturalne

Analiza strukturalna

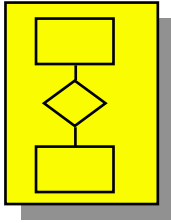
- Cel
 - przetworzenie koncepcji projektu i wymagań użytkownika w specyfikację funkcjonalną



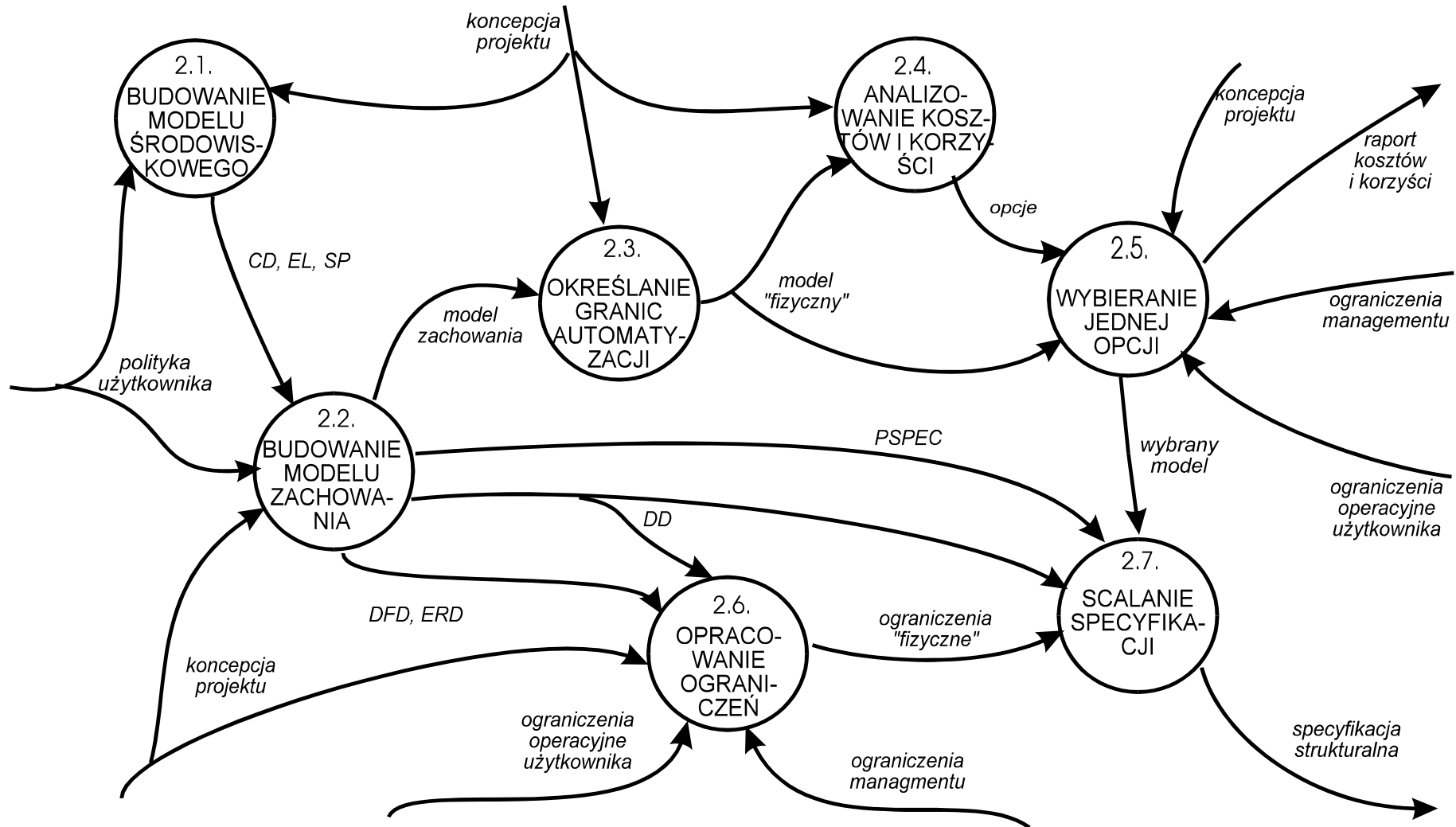
Metody strukturalne

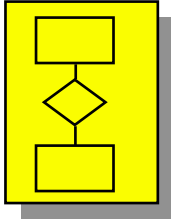
Specyfikacja funkcjonalna

1. Diagramy przepływu danych
2. Diagramy encji i relacji
3. Diagramy stanów i przejść
4. Specyfikacja procesów
5. Słownik
6. Opis ograniczeń fizycznych



ANALIZA STRUKTURALNA

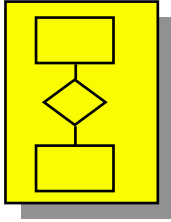




Metody strukturalne

Różne podejścia

- od *DFD*
- od *ERD*
- *DFD* i *ERD* równocześnie



Metody strukturalne

Różne podejścia

- od góry (top-down)

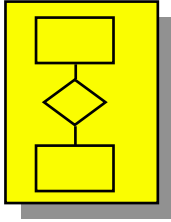


- od dołu (bottom-up)



- od zdarzeń

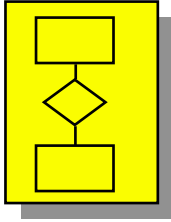




Metody strukturalne

Zalety

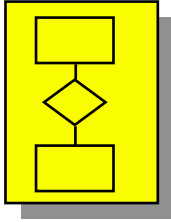
- “zmniejszenie” złożoności
- mały koszt wypróbowywania zmian
- łatwość modyfikacji
- spojrzenie na system z różnych stron
- postać graficzna



Metody strukturalne

Zyski

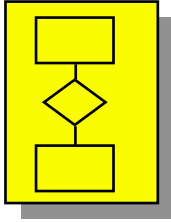
- w kosztach i czasie podczas produkcji systemu informatycznego (10% - 15%)
- w kosztach i czasie podczas pielęgnacji systemu informatycznego (200% - 1.000%)



Metody strukturalne

Wady

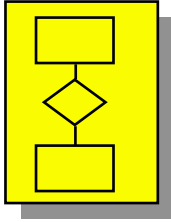
- trudny podział na podprocesy
- duży rozmiar słownika
- słaby nacisk na struktury danych
- nie są zbyt pomocne w systemach aktualizujących i odtwarzających dane
- nie jest sformalizowane przejście od analizy do projektowania



Analiza strukturalna

Główne trudności

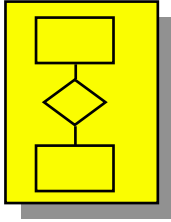
- zrozumienie problemu
- komunikacja międzyludzka
- sprostanie ustawicznym zmianom
- ponowne wykorzystanie wyników analizy



Narzędzia

Wykaz narzędzi

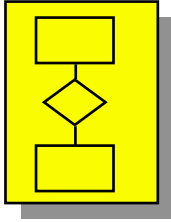
1. Lista zdarzeń (Event List)
2. Diagramy przepływu danych (Data Flow Diagrams)
3. Diagramy encji i relacji (Entity Relationship Diagrams)
4. Słownik (Data Dictionary)
5. Diagramy stanów i przejść (State Transition Diagrams)
6. Specyfikacja procesów (Process Specification)



Lista zdarzeń

Lista zdarzeń

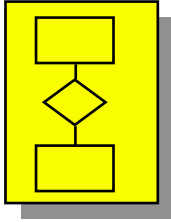
- Definicja
 - ponumerowany zestaw zdarzeń czyli sytuacji w środowisku systemu, które pobudzają system do działania i na które system musi odpowiedzieć



Lista zdarzeń

Lista zdarzeń

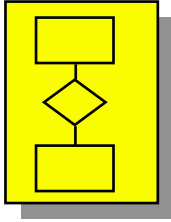
- Cel
 - pomaga zdefiniować zachowanie systemu
- Sposób tworzenia
 - burza mózgów = f (opis istniejącego systemu, wymagania użytkownika, cele przyszłego systemu)
 - “oczyszczenie” produktu burzy mózgów



Lista zdarzeń

Rodzaje zdarzeń

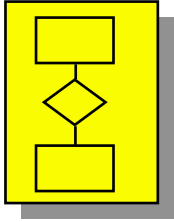
- przepływowe
 - są związane z nadejściem danych
- czasowe
 - występują regularnie, są zależne od czasu
- sterujące
 - nie są związane ani z czasem, ani z nadejściem danych; wymagają natychmiastowej reakcji



Lista zdarzeń

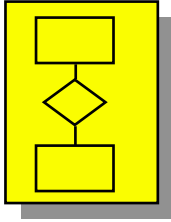
Typowe błędy

1. Zamiast zdarzeń opisujemy reakcje na zdarzenia.
2. Wypisujemy 'zdarzenia', na które system nie musi odpowiadać.
3. Z opisu nie wiadomo, o jakie 'zdarzenie' chodzi.

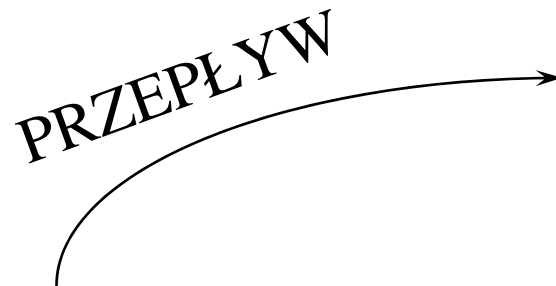
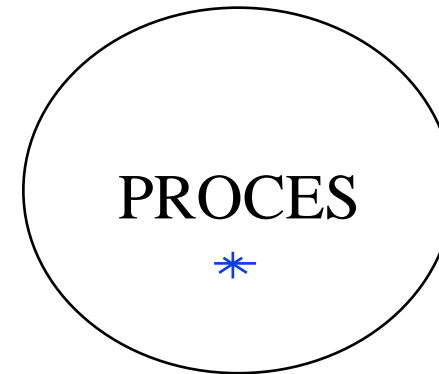
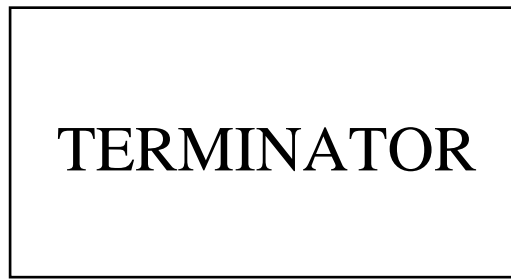


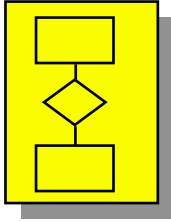
DFD
DFD

- ➔ 1. Syntaktyka
- 2. Diagram kontekstowy
- 3. Dekompozycja i rekompozycja
- 4. Hierarchia DFD
- 5. Specyfikacja procesu
- 6. Realizacja podejścia „od góry”
- 7. Realizacja podejścia „od listy zdarzeń”

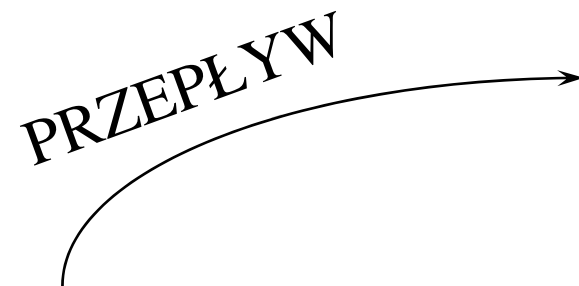
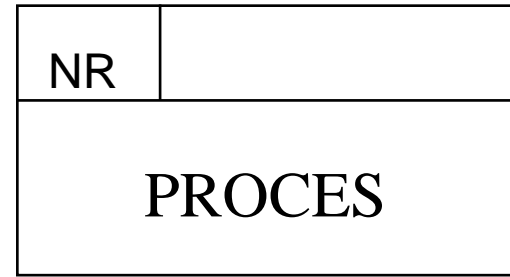


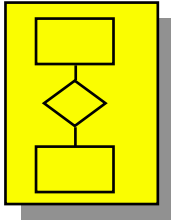
DFD-syntaktyka
Symbole
(Notacja Yourdona)



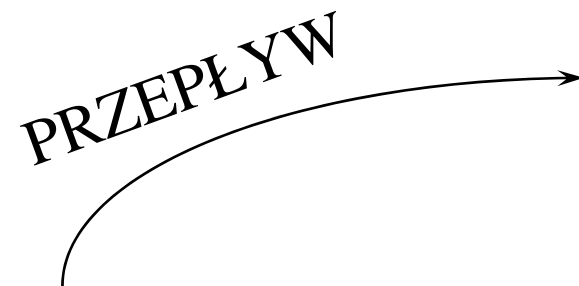
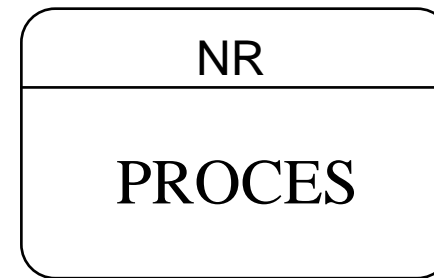


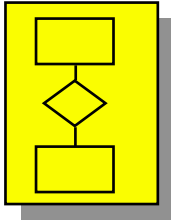
DFD-syntaktyka
Symbole
(Notacja Oracle Methods)



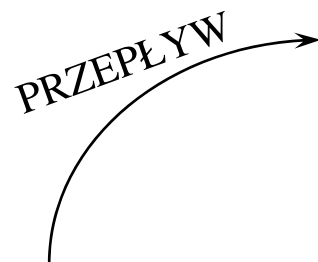
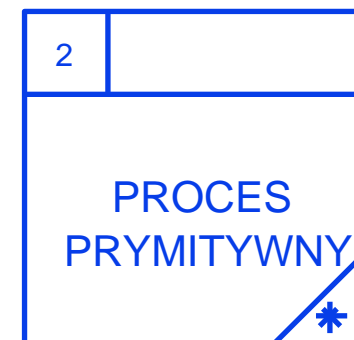
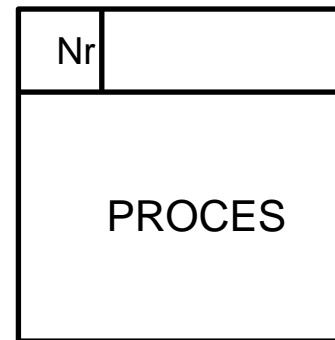


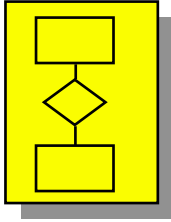
DFD-syntaktyka
Symbole
(Notacja Gana, Sarsona)





DFD-syntaktyka
Symbole
(Notacja SSADM)





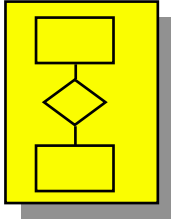
DFD-syntaktyka

Terminator

(External, Terminator)

- Definicja
 - istniejąca rzecz, byt, jednostka, wyodrębniona całość, z którą komunikuje się system i która znajduje się poza jego kontrolą





DFD-syntaktyka

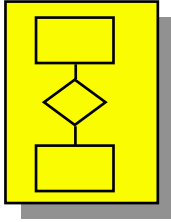
Terminator

- Etykieta
 - rzeczownik lub wyrażenie

Przykłady

SAMOLOT,
MISTRZ, BANK,
GRACZ,
BEZROBOTNY



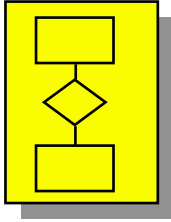


DFD-syntaktyka

Terminator

- Cel
 - zdefiniowanie wszystkich ról w kontaktach systemu ze światem

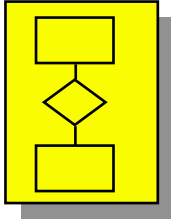
lista terminatorów $\langle \Rightarrow \rangle$ wszystkie role w kontaktach ze światem



DFD-syntaktyka

Terminator

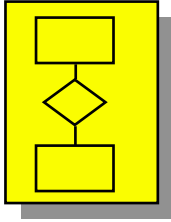
- Zasady
 - terminator jest na zewnątrz modelowanego systemu
 - terminator obrazuje rolę jaką odgrywa
 - rola terminatora musi być dobrze zdefiniowana
 - role terminatorów powinny być niezależne



DFD-syntaktyka

Terminator

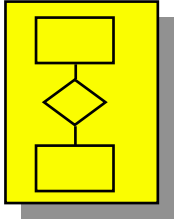
- Zasady c.d.
 - nie mamy wpływu na zachowanie terminatora oraz innych obiektów z nim stowarzyszonych
 - nie pokazujemy przepływów między terminatorami
 - nie numerujemy terminatorów



DFD-syntaktyka

Terminator

- Sposób tworzenia
 - wypisz terminatory z *Zestawu celów*
 - sprawdź role, jakie grają
 - skoncentruj się na rzeczywistych rolach
 - popraw terminatory grające kilka ról
 - opisz krótko rolę każdego terminatora



DFD-syntaktyka

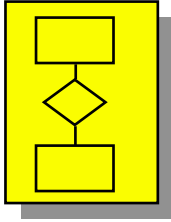
Proces

(Process)

- Definicja

– część systemu przetwarzająca dane





DFD-syntaktyka

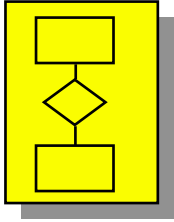
Proces

- **Etykieta**
 - numer oraz rzeczownik odczasownikowy lub wyrażenie albo czasownik

Przykłady

REJESTROWANIE BEZ-
ROBOTNEGO , GRANIE,
ŚLEDZENIE OBIEKTU
LATAJĄCEGO

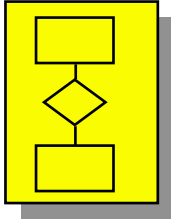




DFD-syntaktyka

Proces

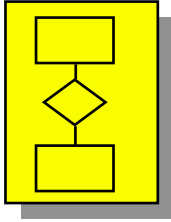
- Cel
 - zdefiniowanie aktywności systemu



DFD-syntaktyka

Proces

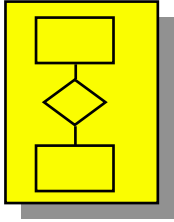
- Zasady
 - proces przetwarza dane
 - proces nie niesie informacji dotyczących szczegółowych operacji
 - proces można dekomponować
 - proces atomowy (prymitywny) opisuje się przy pomocy specyfikacji procesu



DFD-syntaktyka

Proces

- Sposób tworzenia
 - nie jest trywialny; będzie omawiany we wszystkich pozostałych punktach części DFD (czyli w punktach 3, 4, 5, 6 i 7)



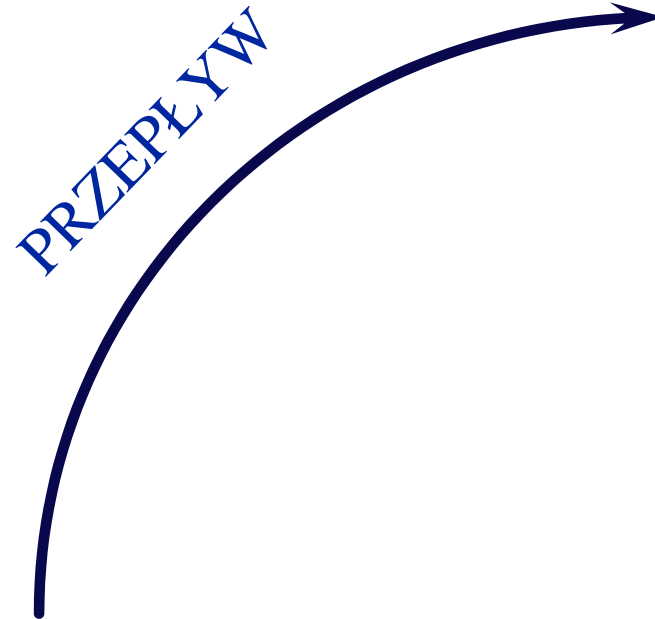
DFD-syntaktyka

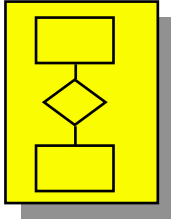
Przepływ

(Flow)

- **Definicja**

– ścieżka dla danych
(droga transportu
danych)



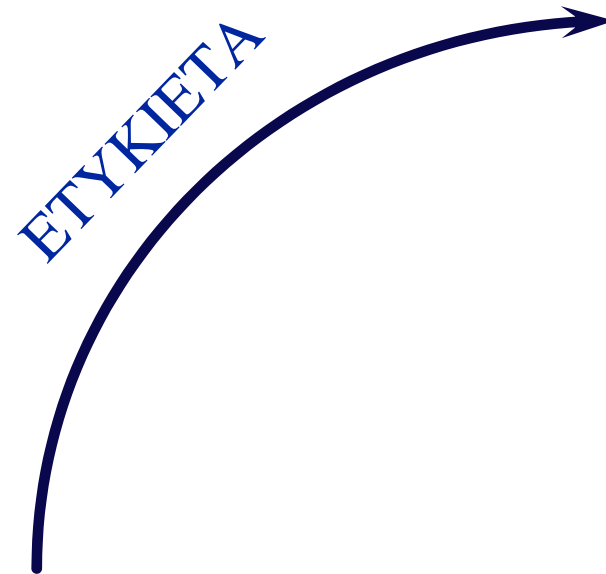


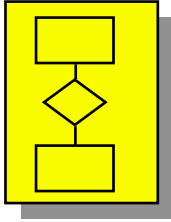
DFD-syntaktyka

Przeptyw

- Etykieta
– rzeczownik lub fraza

Przykłady
PISMO, ODPOWIEDŹ,
DANE PERSONALNE,
NOWA GRA





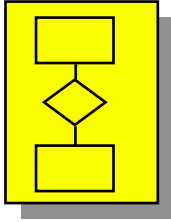
DFD-syntaktyka

Rodzaje przepływów

- wejściowe — karta bezrobotnego



- wyjściowe — dane personalne



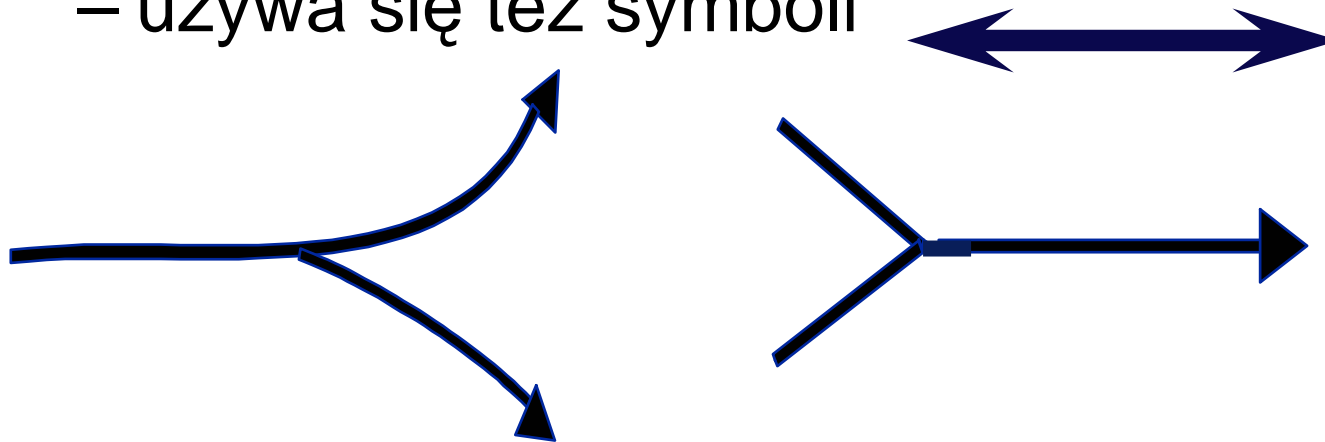
DFD-syntaktyka

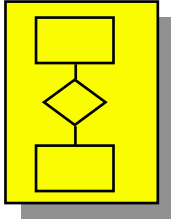
Przepływ

- Uwagi dodatkowe

-  oznacza koniec lub początek przepływu (ujście lub źródło danych)

- używa się też symboli

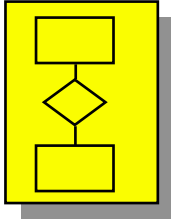




DFD-syntaktyka

Przepływ

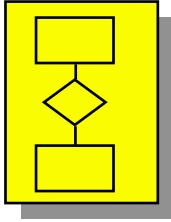
- Cel
 - zdefiniowanie ruchu danych z jednej części systemu do drugiej części albo do terminatora i na odwrót



DFD-syntaktyka

Przepływ

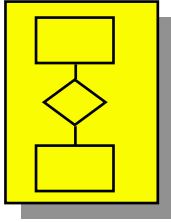
- Zasady
 - przepływ łączy dwa obiekty
 - przepływami nie można łączyć dwóch magazynów ani dwóch terminatorów
 - nazwa przepływu powinna być związana z istotą danych a nie z ich formą
 - przepływ reprezentuje dane w ruchu



DFD-syntaktyka

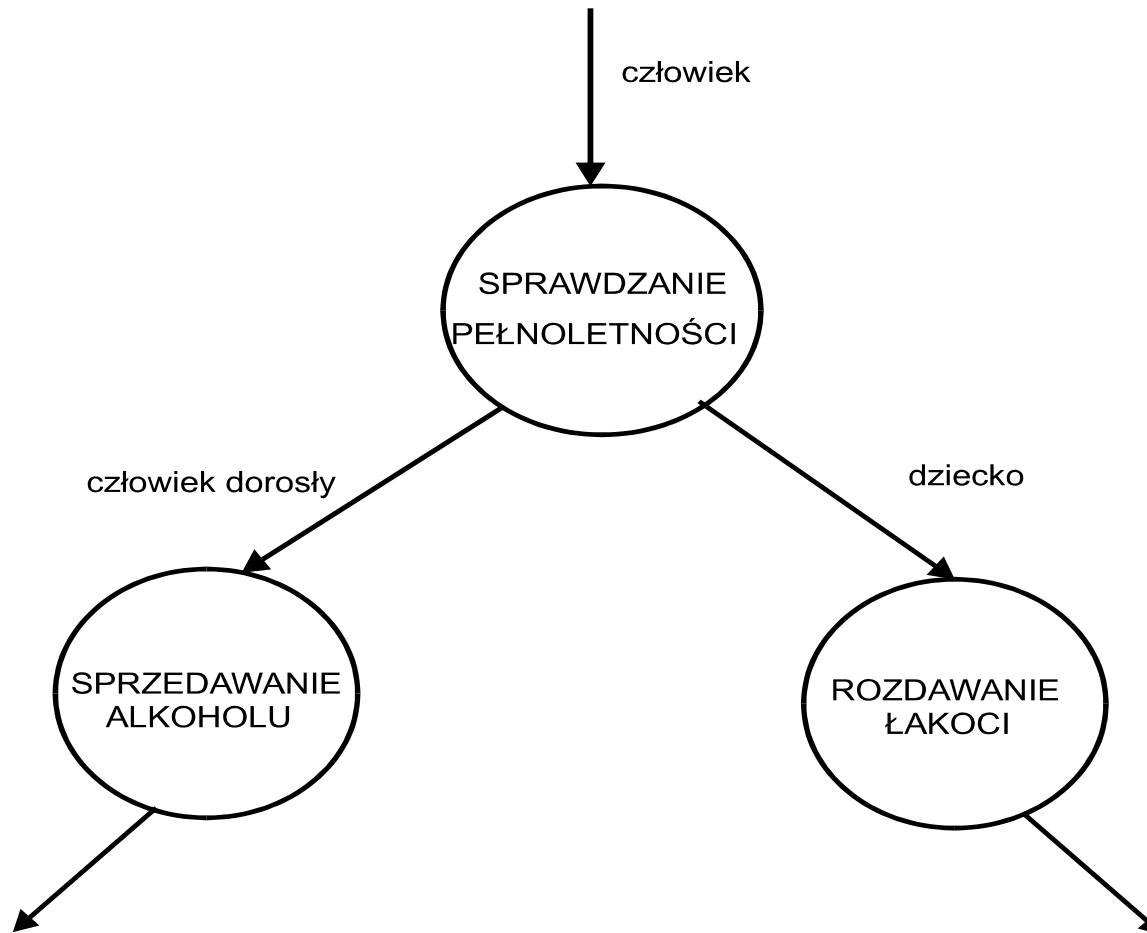
Przepływ

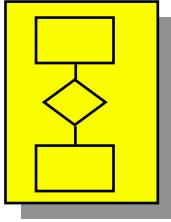
- Zasady c.d.
 - te same fizycznie ale różne logicznie przepływy danych nazywa się innymi nazwami
 - jeśli przepływ niesie jeden rodzaj danych, to najlepiej nadać mu taką nazwę, jaką mają te dane



DFD-syntaktyka

Przykład

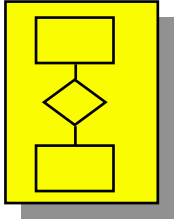




DFD-syntaktyka

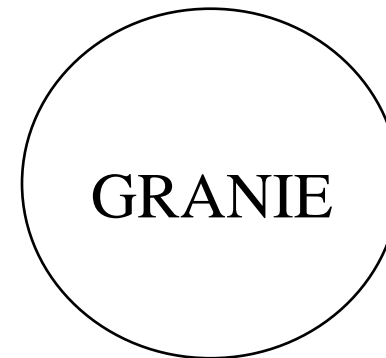
Przepływ

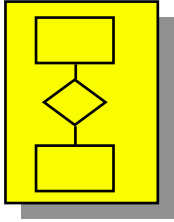
- Sposób tworzenia
 - wybierz pary obiektów typu źródło - ujście
 - narysuj odpowiednią strzałkę
 - określ, jakie dane są przekazywane
 - nazwij ją



DFD-syntaktyka

Ćwiczenie

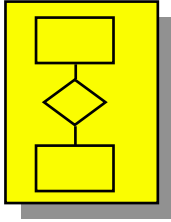




DFD-syntaktyka
Magazyn
(Data Store, Store)

- Definicja
 - miejsce przechowywania danych znajdujących się w spoczynku

MAGAZYN



DFD-syntaktyka

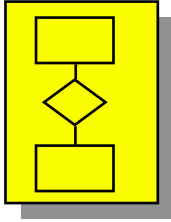
Magazyn

- Etykieta
 - rzeczownik w liczbie mnogiej lub fraza

ETYKIETA

Przykłady

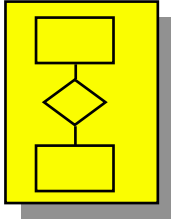
PRACODAWCY,
ROZEGRANE
GRY, PISMA



DFD-syntaktyka

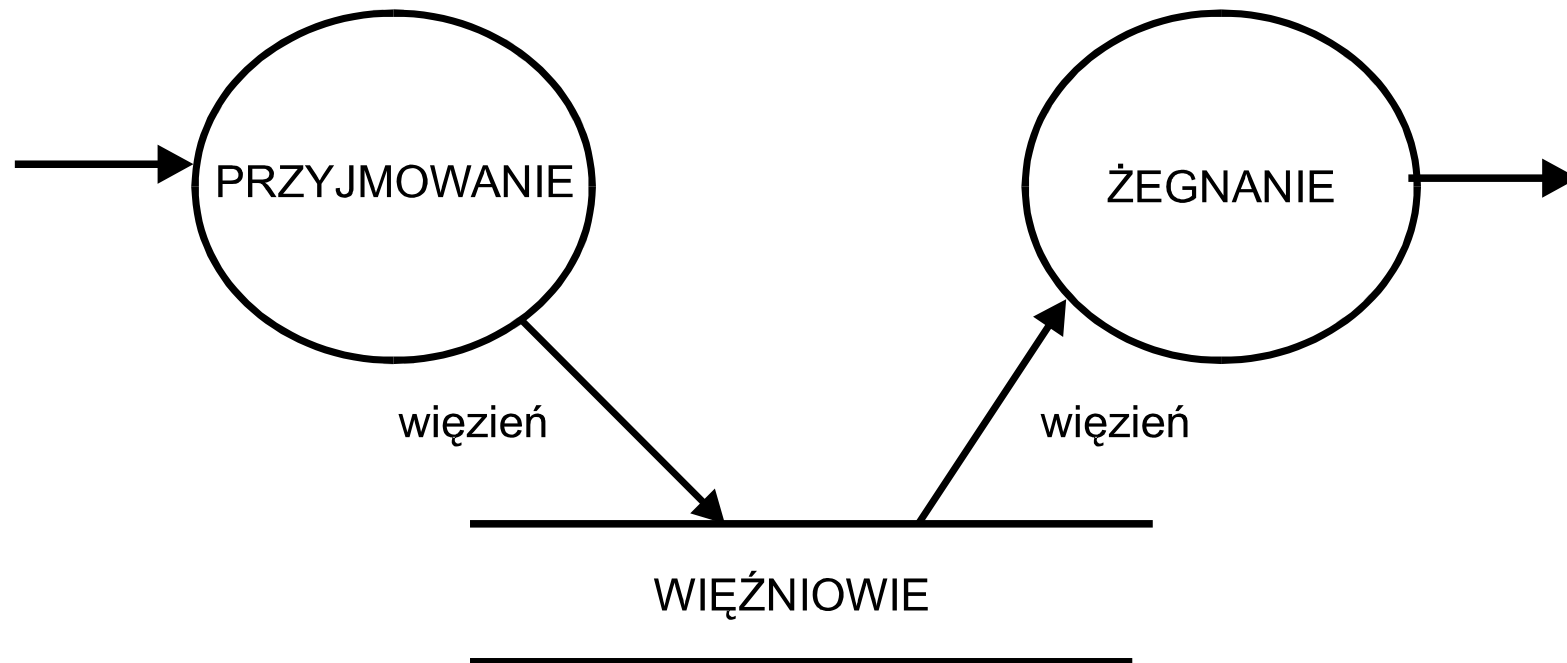
Magazyn

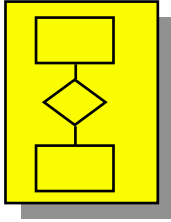
- Cel
 - zaspokojenie potrzeby przechowywania danych w sytuacji gdy:
 - dane są używane nie od razu po wytworzeniu
 - dana jest kilkakrotnie odczytywana
 - dane są używane w innej kolejności niż były wytworzone



DFD-syntaktyka

Przykład

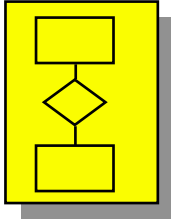




DFD-syntaktyka

Magazyn

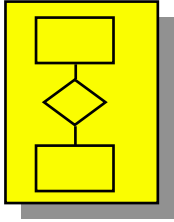
- Zasady
 - magazyn jest pasywny, dane nie wędrują po magazynie
 - magazyn przechowuje oryginały, a oddaje kopie i tylko w przypadku kasowania oryginały z niego znikają



DFD-syntaktyka

Magazyn

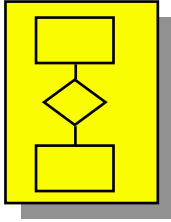
- Zasady c.d.
 - magazyn nie zmienia się, gdy napływają do niego dane
 - wpływ zwykle rozumie się jako zapisywanie danych
 - wypływ zwykle rozumie się jako czytanie lub dostęp do danych



DFD-syntaktyka

Magazyn

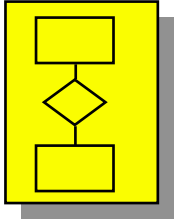
- Zasady c.d.
 - wpływ może nieść tylko te informacje, na które jest miejsce w magazynie
 - wypływ może zawierać tylko te informacje, które są w magazynie
 - suma zawartości wpływów musi być równa sumie zawartości wypływów



DFD-syntaktyka

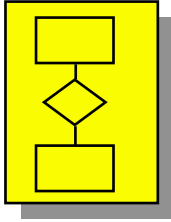
Magazyn

- Sposób tworzenia
 - zbadaj potrzeby istnienia magazynów
 - określ ich zawartość
 - narysuj je i nazwij
 - połącz przepływami z procesami



DFD
DFD

1. Syntaktyka
- ➔ 2. Diagram kontekstowy
3. Dekompozycja i rekompozycja
4. Hierarchia DFD
5. Specyfikacja procesu
6. Realizacja podejścia „od góry”
7. Realizacja podejścia „od listy zdarzeń”

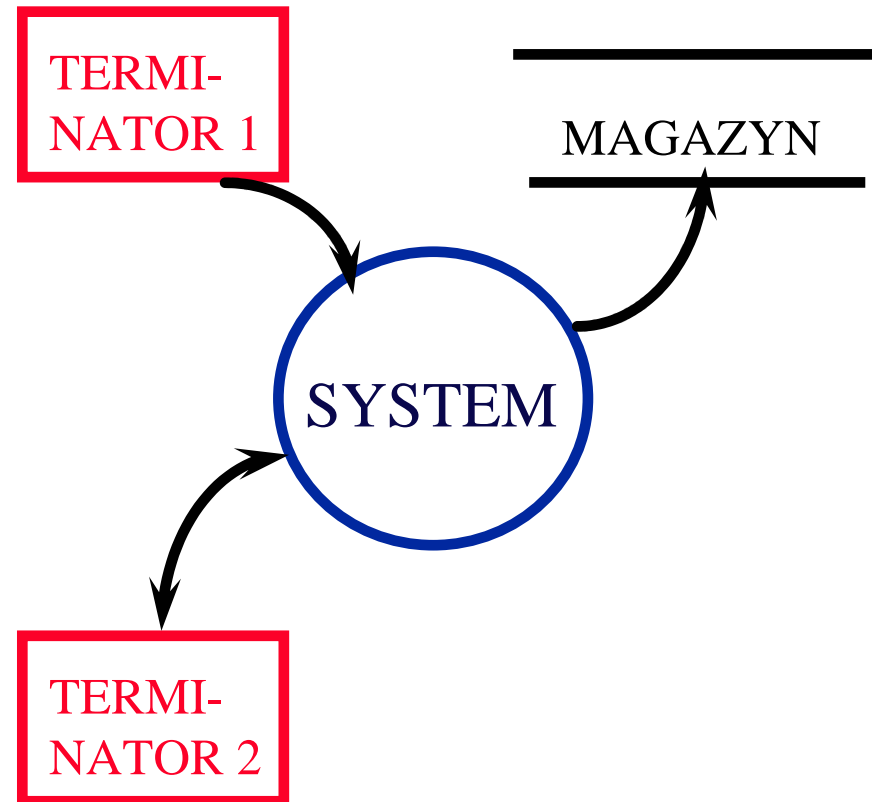


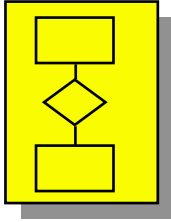
DFD

Diagram kontekstowy

(Context Diagram)

- Definicja
 - specjalny rodzaj diagramu przepływu danych, na którym jest tylko jeden proces reprezentujący system oraz terminatory i magazyny

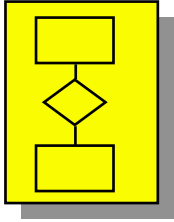




DFD

Diagram kontekstowy *pokazuje*

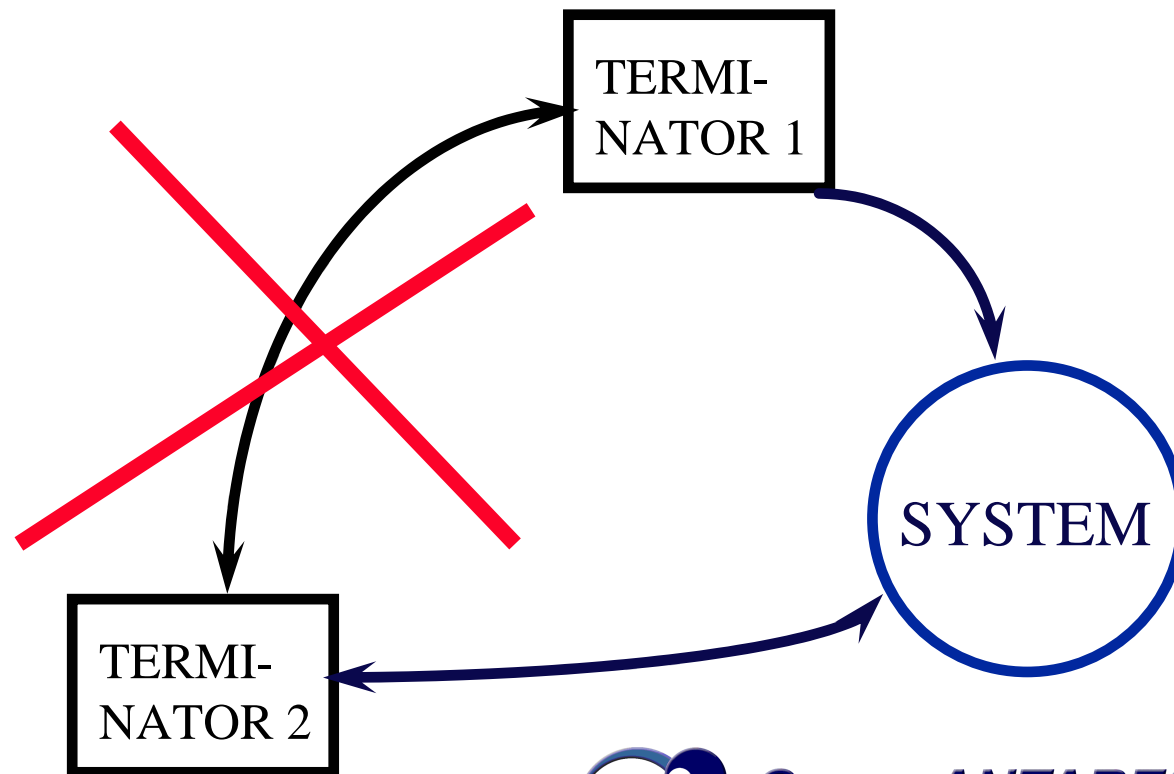
- ludzi, organizacje, systemy, z którymi komunikuje się system (są to terminatory)
- dane, które system otrzymuje z zewnątrz i które musi przetworzyć
- dane wyprodukowane przez system i wysyłane na zewnątrz
- magazyny, które system współdzieli z terminatorami

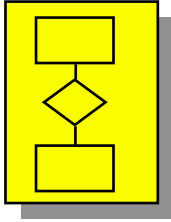


DFD

Diagram kontekstowy *nie pokazuje*

- związków między terminatorami!





DFD

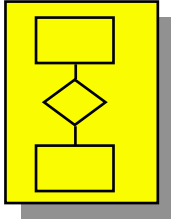
Diagram kontekstowy

- **Etykieta**
 - dowolna, raczej rzeczownik lub zwrot

Przykłady

GRY, BIURO POŚREDNICTWA PRACY

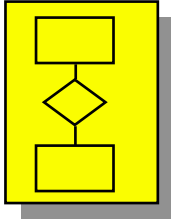
- **Cel**
 - zdefiniowanie granicy między systemem a środowiskiem



DFD

Diagram kontekstowy

- Zasady
 - terminator może komunikować się z systemem bezpośrednio lub przez magazyn
 - przepływy odpowiadają zdarzeniom na liście zdarzeń
 - z/do terminatora może wypływać/wpływać kilka przepływów, dlatego też dla większej czytelności diagramu, dopuszcza się powielanie terminatorów



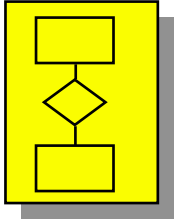
DFD

Diagram kontekstowy

- Sposób tworzenia

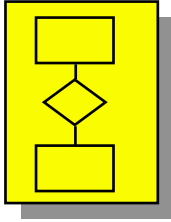
w oparciu o zestaw celów i listę zdarzeń:

- zakaśl granice systemu i nazwij go
- wyodrębnij i nazwij terminatory
- wyodrębnij i nazwij przepływy danych
- wystartowuj z budową słownika



DFD
DFD

1. Syntaktyka
2. Diagram kontekstowy
- ➔ 3. Dekompozycja i rekompozycja
4. Hierarchia DFD
5. Specyfikacja procesu
6. Realizacja podejścia „od góry”
7. Realizacja podejścia „od listy zdarzeń”



DFD

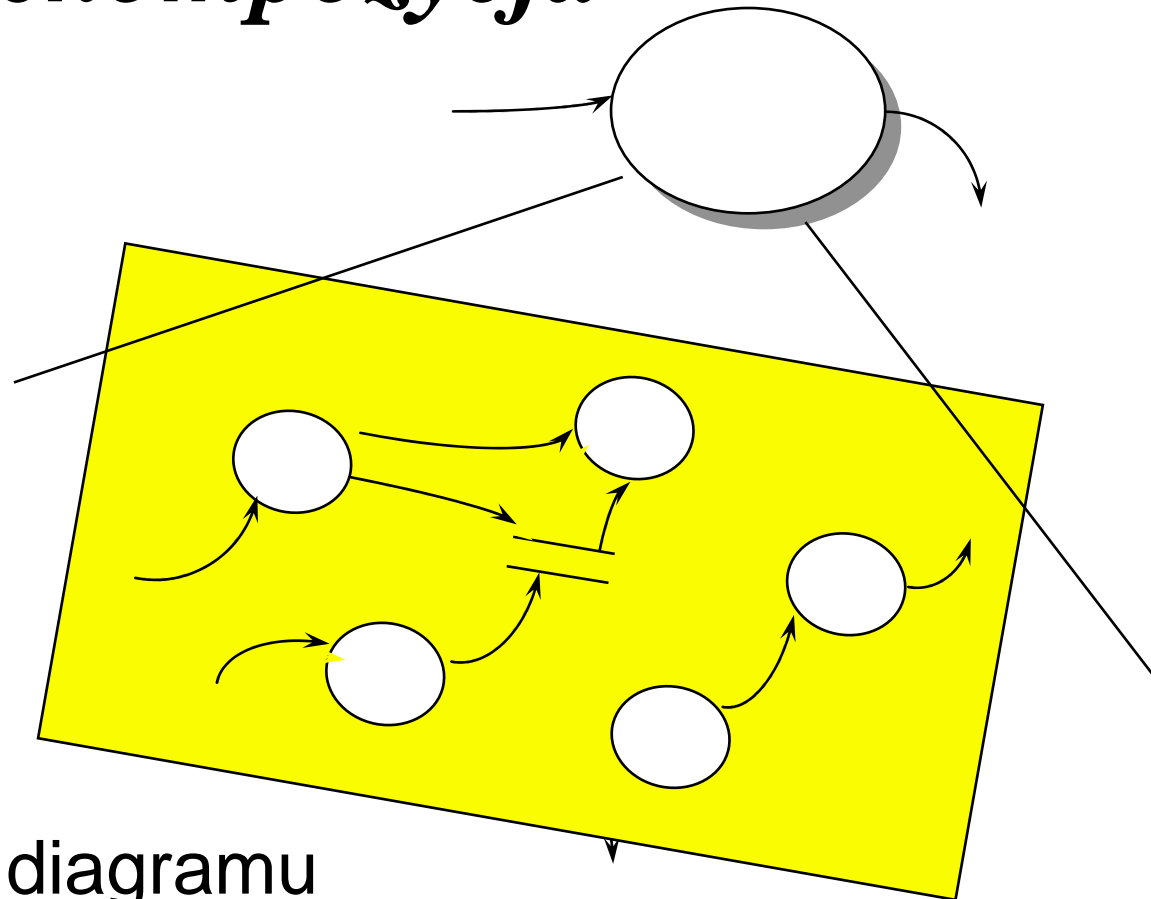
Dekompozycja

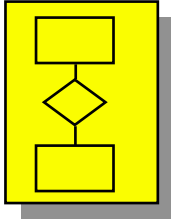
- Definicja

– rozbitcie procesu na podprocesy

- Cel

– zbudowanie diagramu niższego poziomu

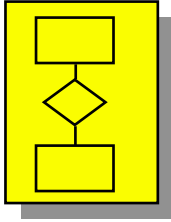




DFD

Dekompozycja

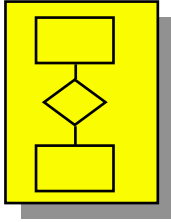
- Zasady / zalecenia
 - dekompozycje prowadź tak długo, aż otrzymasz procesy prymitywne
 - bierz pod uwagę przynajmniej dwa niższe poziomy



DFD

Dekompozycja

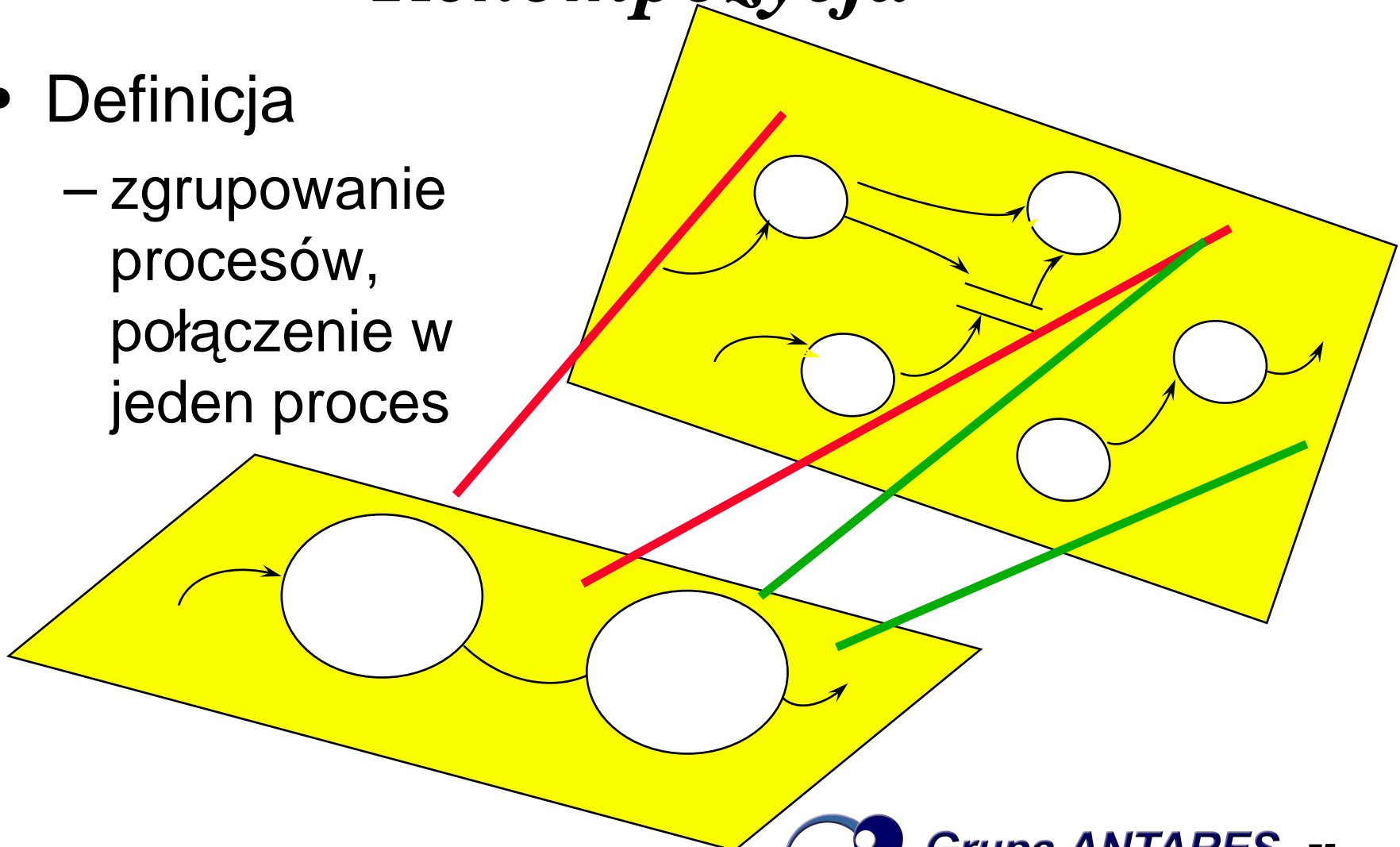
- Sposób tworzenia
 - określ podprocesy dekomponowanego procesu
 - prześledź drogę danych
 - dorysuj potrzebne magazyny
 - uzgodnij (zbalansuj) przepływy między poziomami



DFD

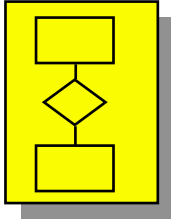
Rekompozycja

- Definicja
 - zgrupowanie procesów, połączenie w jeden proces



Grupa ANTARES

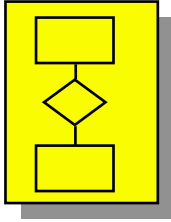
INFORMATION TECHNOLOGY CONSULTANTS



DFD

Rekompozycja

- Cel
 - zbudowanie lub poprawienie diagramu wyższego poziomu



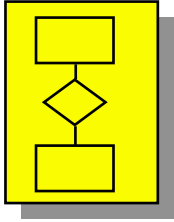
DFD

Rekompozycja

- Zalecenia

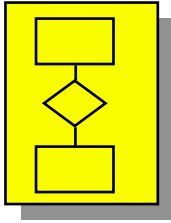
procesy grupuj mając na względzie:

- pokrewne reakcje
- korzystanie z tych samych danych (można "ukryć" magazyny, z których nie korzystają inne procesy)



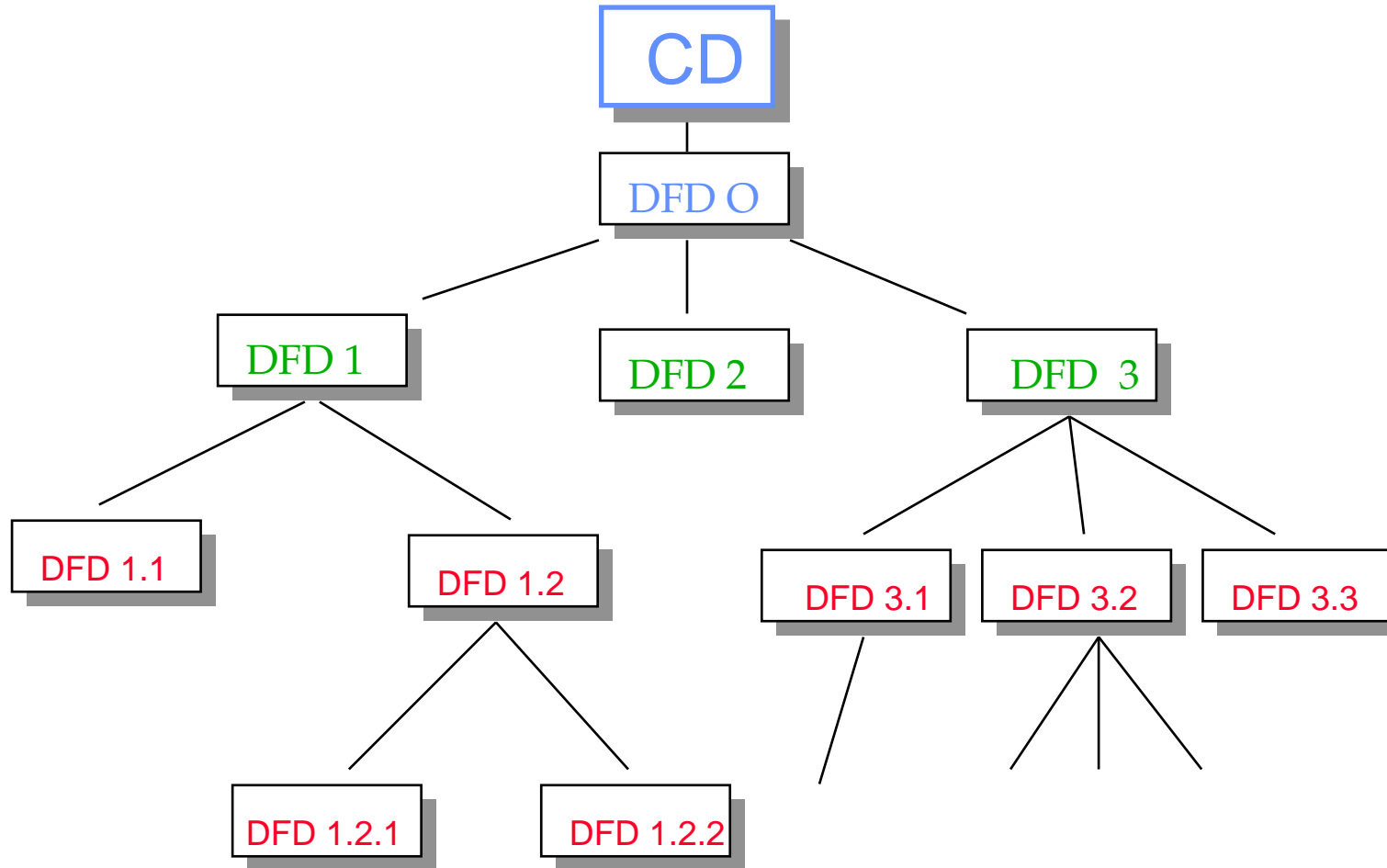
DFD
DFD

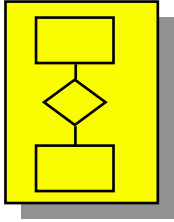
1. Syntaktyka
2. Diagram kontekstowy
3. Dekompozycja i rekompozycja
- ➔ 4. Hierarchia DFD
5. Specyfikacja procesu
6. Realizacja podejścia „od góry”
7. Realizacja podejścia „od listy zdarzeń”



DFD

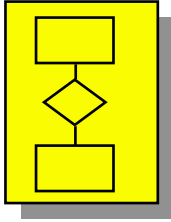
Hierarchia DFD





DFD
DFD

1. Syntaktyka
2. Diagram kontekstowy
3. Dekompozycja i rekompozycja
4. Hierarchia DFD
- ➔ 5. Specyfikacja procesu
6. Realizacja podejścia „od góry”
7. Realizacja podejścia „od listy zdarzeń”

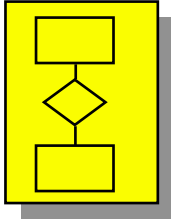


DFD

Specyfikacja procesu

(Process Specification)

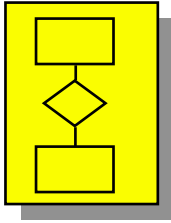
- Definicja
 - możliwie precyzyjny i jednoznaczny opis funkcji wykonywanych przez proces;
powinien mieć formę czytelną dla analityka i dla użytkownika



DFD

Specyfikacja procesu

- Może być:
 - opisem w strukturalnym języku angielskim lub polskim
 - opisem w języku narracyjnym
 - opisem warunków przed/po
 - drzewem decyzyjnym
 - tablicą decyzyjną
 - wzorem matematycznym

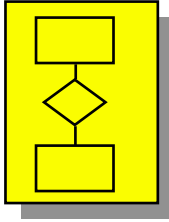


DFD - Pspec

Przykład

Tablica decyzyjna

	1	2	3
Wiek > 15 lat	T	N	N
Waga < 100 kg	N	N	T
Płeć	M	K	K
Aspiryna	5	4	1
Witamina C	10	6	3



DFD - Pspec

Przykład

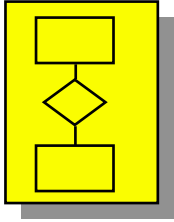
Specyfikacja procesu 3.2.2.1: Przygotowanie raportu

Proces jest związany z trzema przepływami:
przepływy wejściowe - *Dane do raportu, Prośba o raport,*
przepływy wyjściowe - *Raport.*

Opis

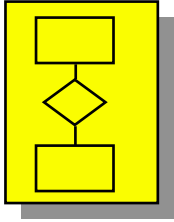
Jeśli zleceniodawca poprosi o raport to:

- pobierz dane do raportu,
- napisz raport,
- wyślij go adresatowi raportu.



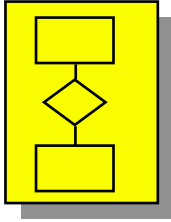
DFD - Pspec

Przykład



DFD
DFD

1. Syntaktyka
2. Diagram kontekstowy
3. Dekompozycja i rekompozycja
4. Hierarchia DFD
5. Specyfikacja procesu
- ➔ 6. Realizacja podejścia „od góry”
7. Realizacja podejścia „od listy zdarzeń”

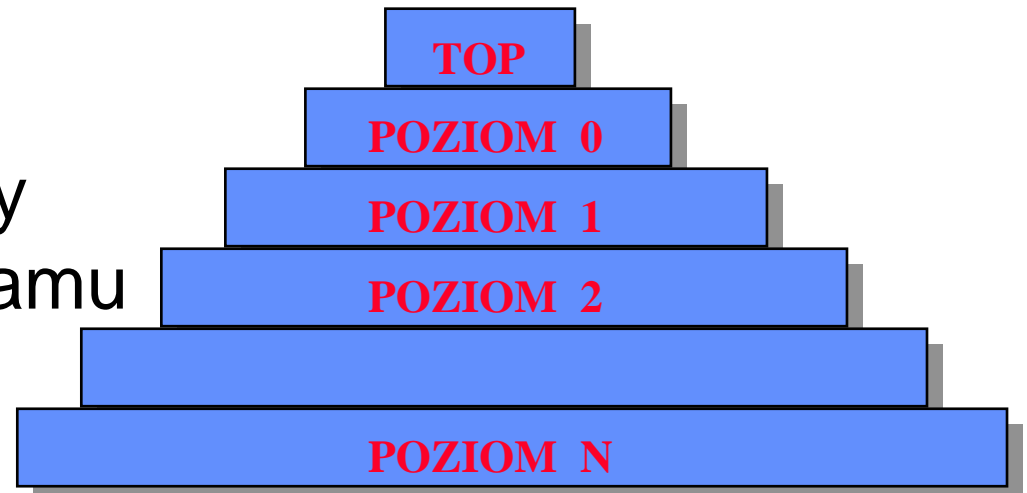


DFD - Podejście od góry

Podejście od góry

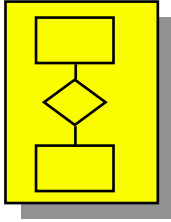
- Definicja

- rozpoczęcie budowy diagramów od diagramu kontekstowego i zakończenie na procesach atomowych



Grupa ANTARES

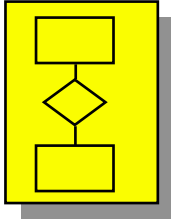
INFORMATION TECHNOLOGY CONSULTANTS



DFD

Podójście od góry

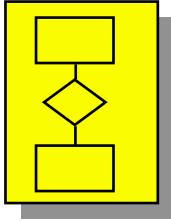
- Zasady
 - unikaj czarnych dziur i źródeł
 - balansuj (uzgadniaj) każdy *DFD* z diagramami z nim stowarzyszonymi (\Leftrightarrow balansuj przepływy między poziomami)
 - balansuj każdy *DFD* z *ERD*
 - uaktualniaj słownik



DFD

Podajście od góry

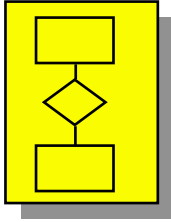
- Zasady /zalecenia
 - numeruj procesy
 - nadawaj wiele mówiące nazwy dla procesów, przepływów i magazynów
 - unikaj nadmiernej zawiłości diagramów
 - dbaj o czytelność diagramu



DFD

Podejście od góry

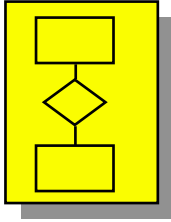
- Zalecenia
 - największy nakład pracy połów na *DFD 0*
 - prowadź równoległe dekompozycje
 - przewiduj przynajmniej o jeden poziom głębiej
 - utrzymaj równowagę poziomów w całym modelu



DFD

Podejście od góry

- Sposób tworzenia
 - narysuj *CD*
 - narysuj *DFD 0*
 - zdekomponuj każdy z procesów diagramu *DFD 0*
 - zdekomponuj każdy z procesów *DFD X*
 - opisz procesy atomowe

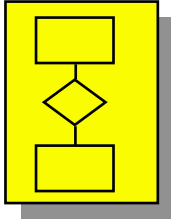


DFD - Podejście od góry

Pierwsze cięcie DFD 0

Budujemy w oparciu o:

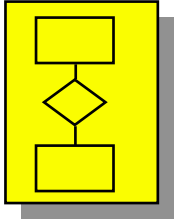
- cele systemu
- diagram kontekstowy
- listę zdarzeń



DFD - Podejście od góry

Pierwsze cięcie DFD 0

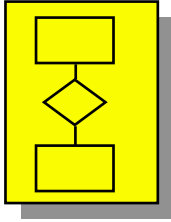
- Sposób tworzenia
 - wyodrębnij główne procesy zachodzące w systemie (np. reakcje na zdarzenia)
 - wyodrębnij potrzebne magazyny
 - połącz procesy ze sobą i z magazynami
 - zbalansuj *DFD 0* z diagramem kontekstowym
 - zbalansuj *DFD 0* z *ERD*
 - uzupełnij słownik



DFD - Podejście od góry

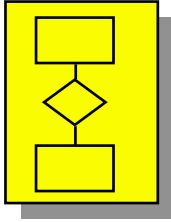
Pozostałe DFD

- Sposób tworzenia
 - podobnie jak *DFD 0*
 - i w oparciu o zasady i zalecenia opisane na slajdach dotyczących podejścia od góry



DFD
DFD

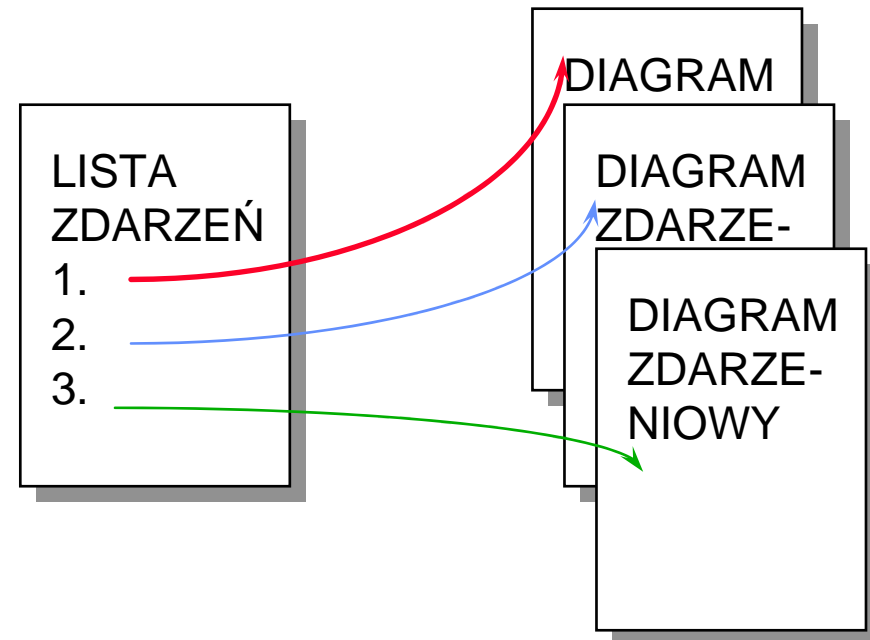
1. Syntaktyka
2. Diagram kontekstowy
3. Dekompozycja i rekompozycja
4. Hierarchia DFD
5. Specyfikacja procesu
6. Realizacja podejścia „od góry”
- ➔ 7. Realizacja podejścia „od listy zdarzeń”



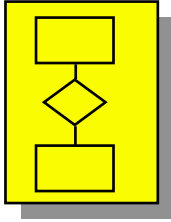
DFD

Podejście od listy zdarzeń

- Definicja
 - rozpoczęcie budowy modelu od listy zdarzeń i diagramów zdarzeniowych



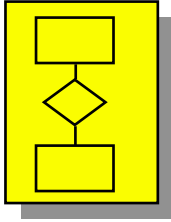
- Cel
 - zbudowanie modelu procesów



DFD

Podejście od listy zdarzeń

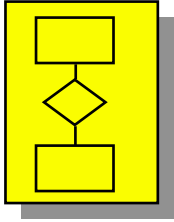
- Zasady / zalecenia
 - każdemu zdarzeniu odpowiada jeden diagram zdarzeniowy
 - synchronizuj każdy diagram zdarzeniowy z *ERD*
 - uaktualnij słownik
 - przestrzegaj zasad i zaleceń jak w metodzie od góry



DFD

Podejście od listy zdarzeń

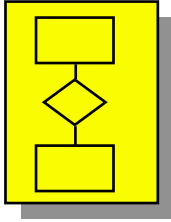
- Sposób tworzenia
 - każde zdarzenie z listy zdarzeń rozrysuj na jednym diagramie zdarzeniowym
 - “połącz” diagramy zdarzeniowe w jeden diagram
 - przeprowadź rekompozycję
 - zrób dekompozycję



Analiza strukturalna

ERD

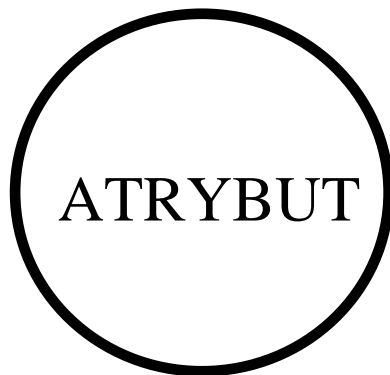
- ➔ 1. Syntaktyka
- 2. Budowanie diagramu

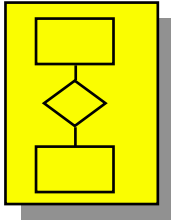


ERD-syntyka

Symbole

(Notacja Chena)

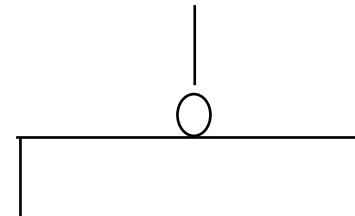
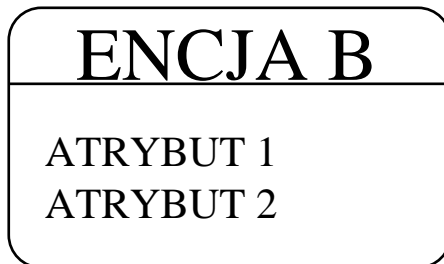
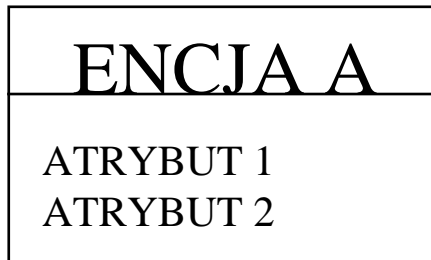


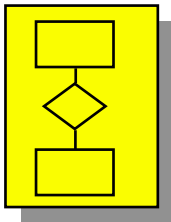


ERD-syntaktyka

Symbole

(Notacja relacyjna)

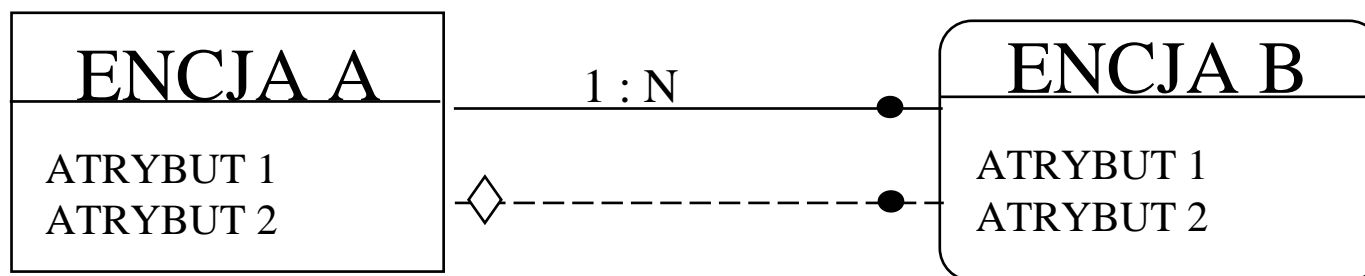
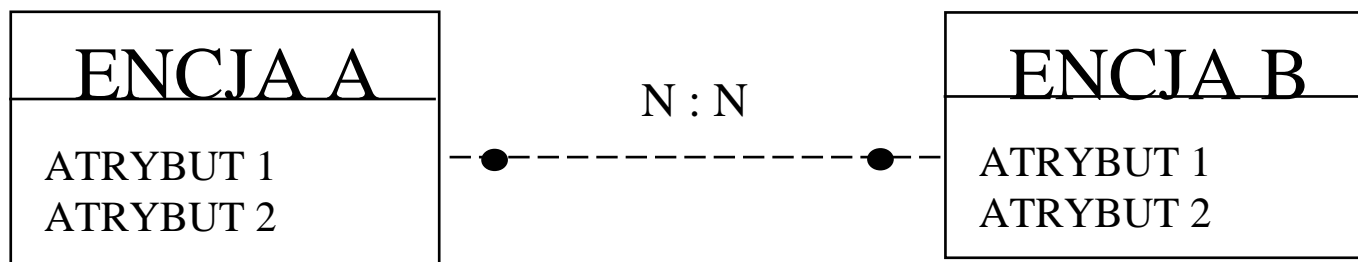


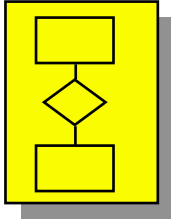


ERD-syntaktyka

Symbole

(Notacja relacyjna)





ERD-syntaktyka

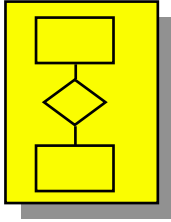
Encja

(Entity)

- Definicja

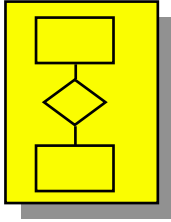
- klasa abstrakcji wystąpień encji
- zbiorowisko ludzi lub rzeczy połączonych podobieństwem





Wystąpienie encji

- Definicja
 - byt (jednostka, wyodrębniona całość, rzecz realnie istniejąca), o którym informacja musi być pamiętana w systemie
- Każde wystąpienie encji jest:
 - rozróżnialne
 - opisane przynajmniej przez jedną cechę



ERD-syntaktyka

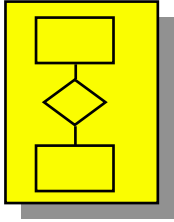
Encja

- Etykieta
– rzeczownik lub
wyrażenie



Przykłady

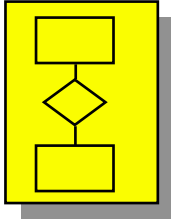
BEZROBOTNY, GRACZ,
RODZAJ GRY



ERD-syntaktyka

Encja

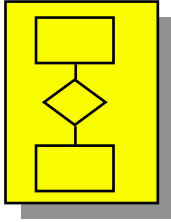
- Cel
 - umożliwienie zapamiętania zbioru informacji potrzebnych systemowi do przetwarzania



ERD-syntaktyka

Encja

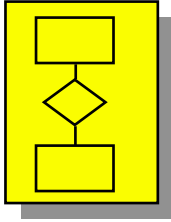
- Zasady/zalecenia
 - nazwa encji powinna oddawać jej rolę a nie fizyczną postać
 - role różnych encji powinny być niezależne



ERD-syntaktyka

Przykład

Encja	Wystąpienie encji
BEZROBOTNY	Jan Kowalski, Bożena Nowak
RODZAJ GRY	szachy, warcaby, go



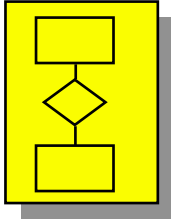
ERD-syntaktyka

Atrybut

(Attribute)

- Definicja
 - klasa abstrakcji wystąpień własności encji; cecha encji, wyróżniająca tę encję od innych





ERD-syntaktyka

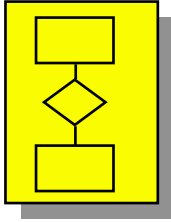
Atrybut

- Etykieta
– rzeczownik

Przykłady

IMIĘ , NAZWISKO,
ADRES





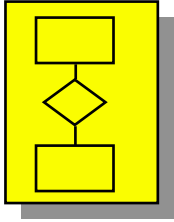
ERD-syntaktyka

Klucz *(Key)*

- Definicja

– zestaw atrybutów,
po których wartości
rozpoznaje się
encję; atrybuty
kluczowe zwykle
oznacza się
gwiazdką



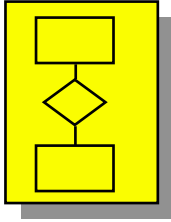


ERD-syntaktyka

Atrybut

- Cel

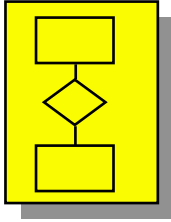
– opisanie cech encji



ERD-syntaktyka

Atrybut

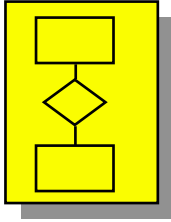
- Zasady/zalecenia
 - każdy atrybut jest połączony z jedną encją albo relacją
 - każda encja ma przynajmniej jeden atrybut (zwykle więcej) i nie powinna posiadać tylko jednego atrybutu (chyba, że jest to silnie uzasadnione)



ERD-syntaktyka

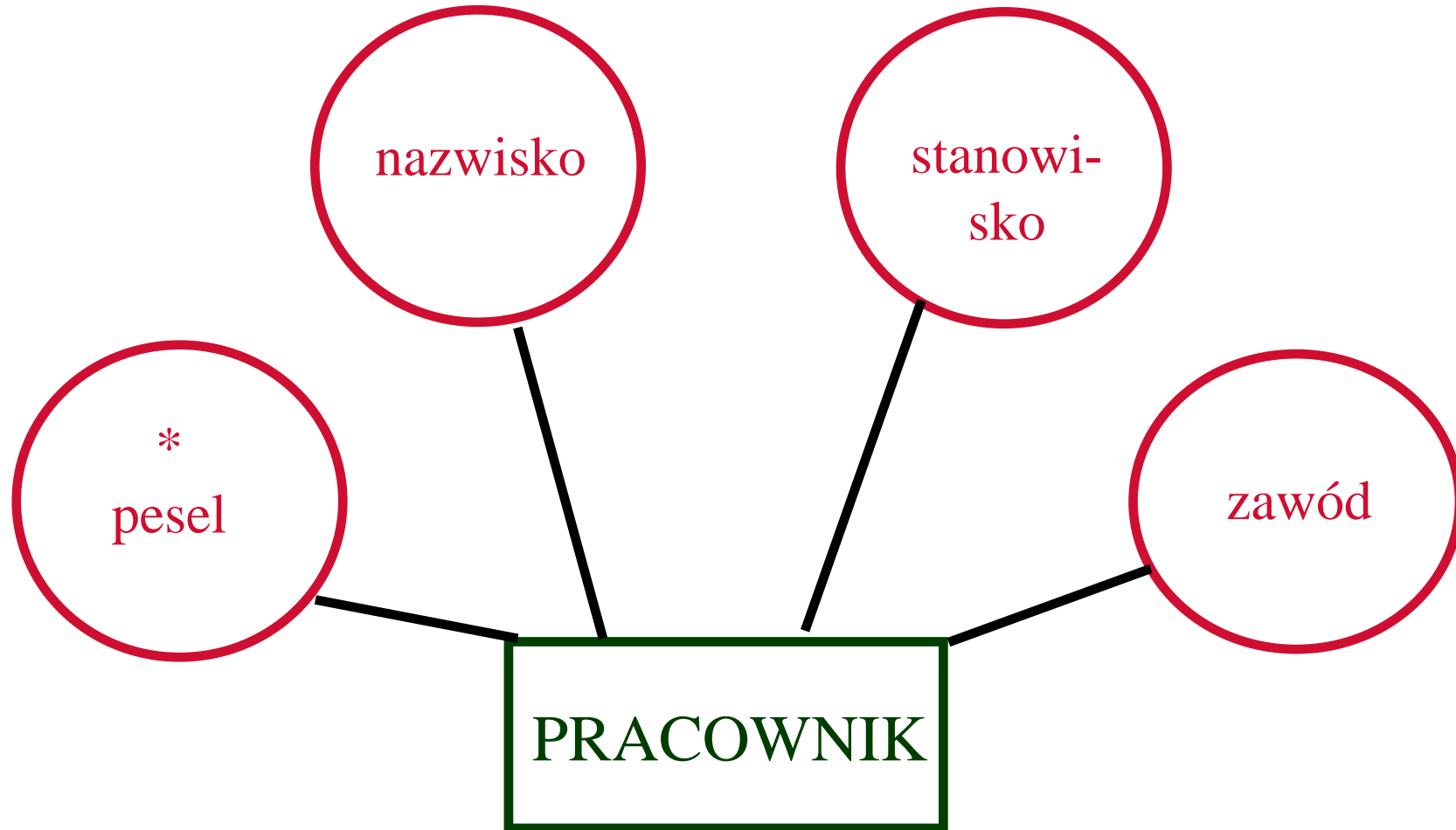
Atrybut

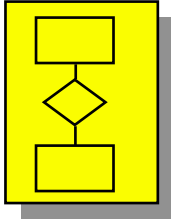
- Zasady/zalecenia
 - atrybut reprezentuje pojęcie atomowe
 - wystąpienie encji musi mieć określone wszystkie wartości atrybutów encji
 - jeśli wystąpienia encji mają te same wartości atrybutów, to jest to jedna encja



ERD-syntaktyka

Przykład





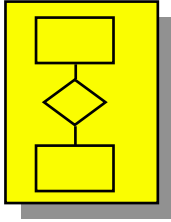
ERD-syntaktyka

Relacja

(Relationship)

- Definicja
 - związek (zależność) między encjami
 - dowolny podzbiór iloczynu kartezyjskiego





ERD-syntaktyka

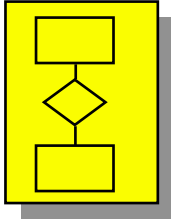
Relacja

- **Etykieta**
 - czasownik w 3. osobie l.p.
lub zwrot

Przykłady

POSIADA, ROZEGRAŁ,
ZATRUDNIA, JEST
RODZAJU

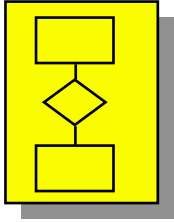




ERD-syntaktyka

Liczność relacji

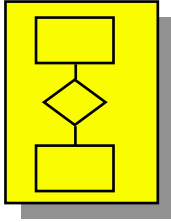
- Definicja
 - maksymalna ilość wystąpień encji w relacji
- Etykieta
 - liczba naturalna albo para liczb naturalnych oddzielonych przecinkiem



ERD-syntaktyka

Relacja

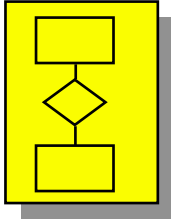
- Cel
 - skojarzenie, powiązanie ze sobą encji



ERD-syntaktyka

Relacja

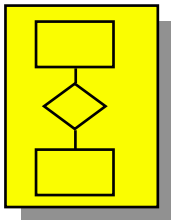
- Zasady
 - relacja musi łączyć encje
 - relacja może łączyć dowolną ilość encji (ilość encji w relacji nazywamy wymiarem)
 - między dwoma encjami może zachodzić więcej niż jedna relacja
 - encja może być w relacji ze sobą



ERD-syntaktyka

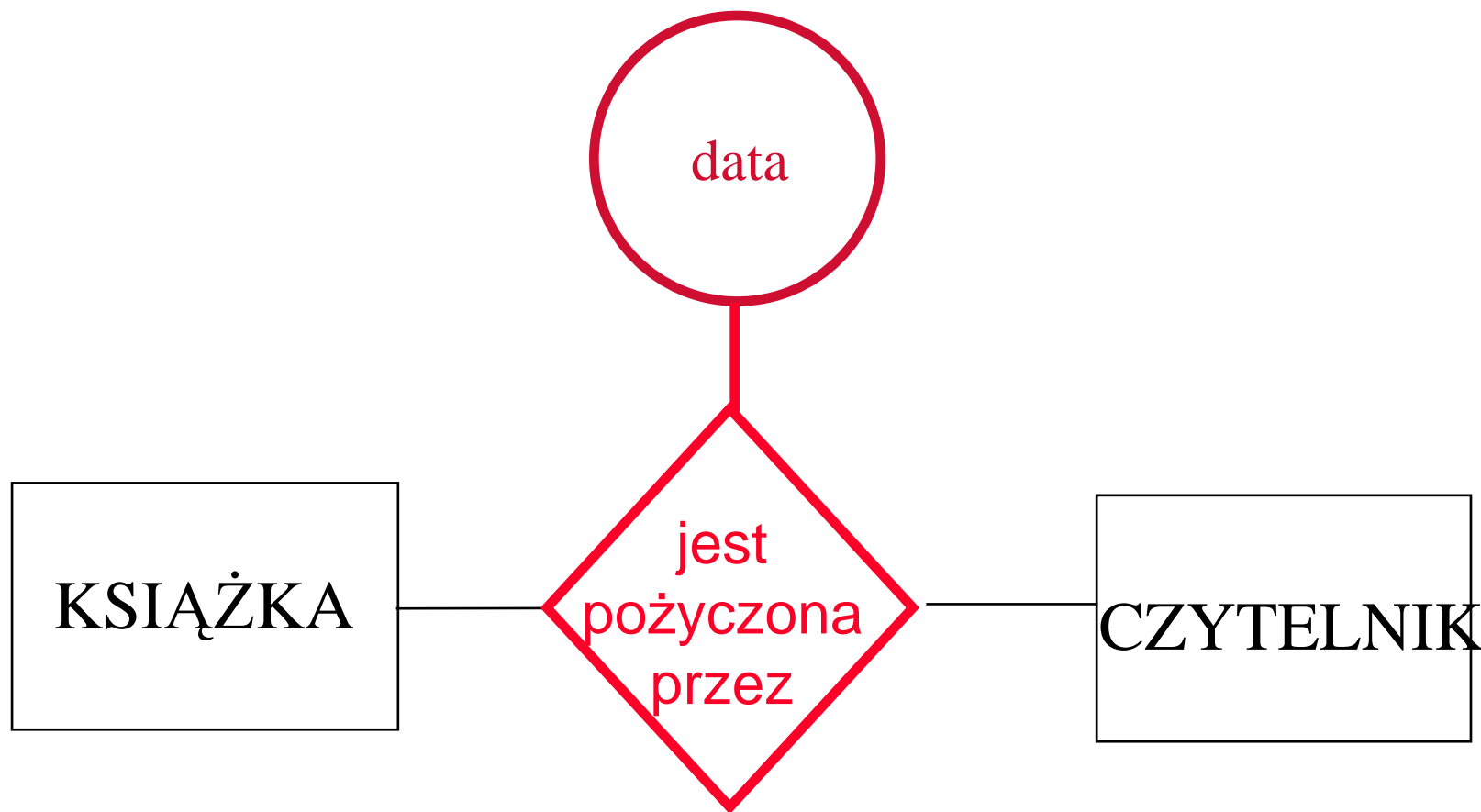
Relacja

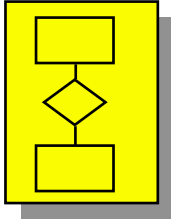
- Zasady
 - relacja powinna mieć określoną liczność
 - liczność ustala się z perspektywy encji
 - relacja może mieć atrybuty



ERD-syntaktyka

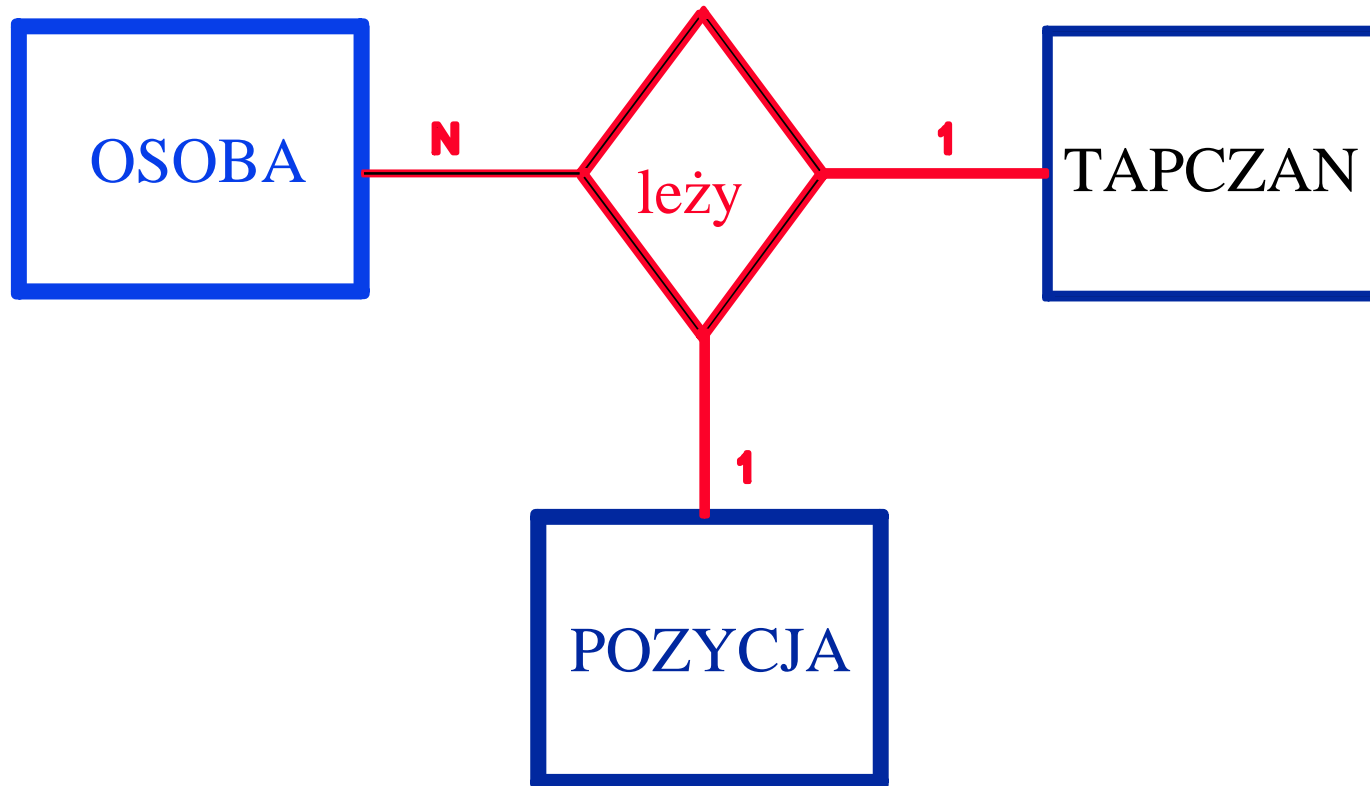
Przykład

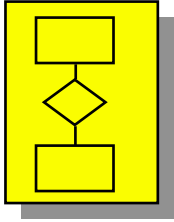




ERD-syntaktyka

Przykład

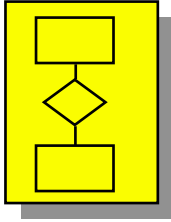




Analiza strukturalna

ERD

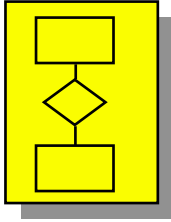
1. Syntaktyka
- ➔ 2. Budowanie diagramu



ERD

Pierwsze cięcie

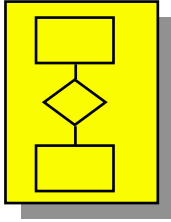
- Sposób tworzenia
 - w oparciu przede wszystkim o listę zdarzeń (encje i relacje wywodzą się od zdarzeń), zestaw celów i diagram kontekstowy



ERD

Pierwsze cięcie

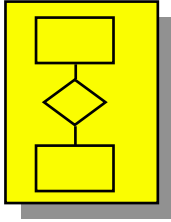
- Sposób tworzenia
 - wyodrębnić i nazwać encje
 - określić między którymi encjami istnieją zależności
 - umieścić nazwy i definicje w słowniku
 - zbalansuj *ERD* z *DFD*



ERD

Drugie cięcie

- Sposób tworzenia
 - wyodrębnić i nazwij atrybuty encji
 - wyodrębnić i nazwij zależności (relacje) między encjami
 - wyodrębnić i nazwij atrybuty relacji
 - umieść nazwy i definicje w słowniku
 - zbalansuj *ERD* z *DFD*

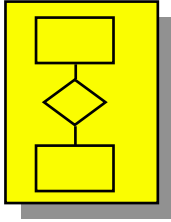


ERD

ERD

- Zasady

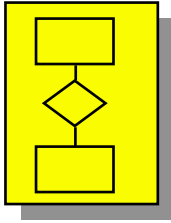
- zawartość magazynu na *DFD* musi być częścią *ERD*
- wszystkie atrybuty muszą być przynajmniej raz użyte w *DFD*
- każda relacja musi być przynajmniej raz wykorzystana w jakimś procesie



ERD

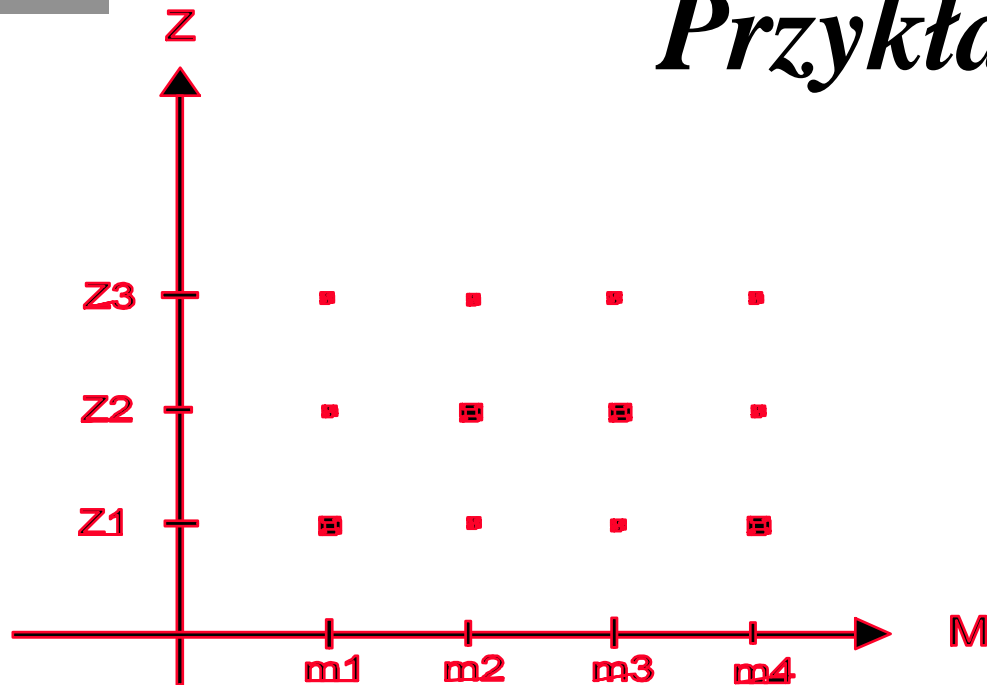
ERD

- Zasady/zalecenia
 - należy unikać encji posiadających tylko jeden atrybut-identyfikator (takie obiekty na ogół nie wnoszą żadnych nowych informacji dla systemu)



ERD

Przykład



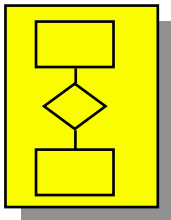
$Z = \{z1, z2, z3\}$

$M = \{m1, m2, m3, m4\}$

$M \times Z = \{(m, z) : m \in M, z \in Z\}$

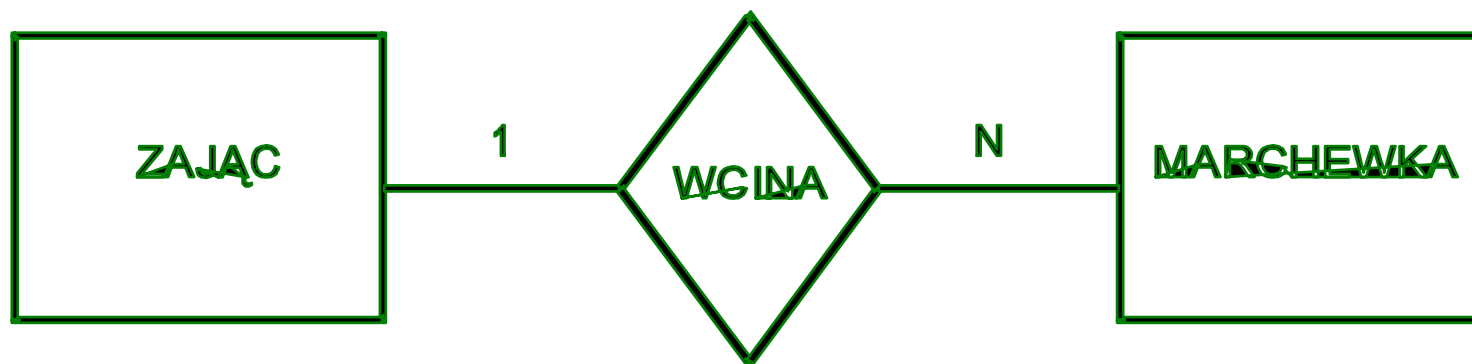


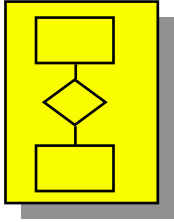
Grupa ANTARES
INFORMATION TECHNOLOGY CONSULTANTS



ERD

Przykład c.d.

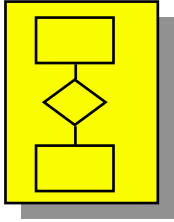




ERD

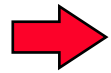
Ćwiczenie 5

Określ encje w systemie _____.



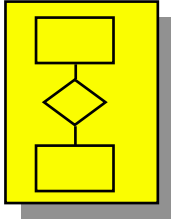
Dzień 2

2.6. Diagramy encji i relacji



2.7. Słownik

2.8. Reguły spójności i poprawności



Słownik

Słownik

(Data Dictionary)

- Definicja

– uporządkowana w sposób leksykalny lista wszystkich obiektów występujących we wszystkich modelach systemu (lub tylko elementów danych) wraz z precyzyjnymi definicjami

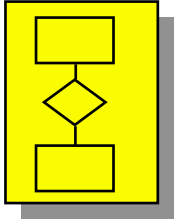


Grupa ANTARES

INFORMATION TECHNOLOGY CONSULTANTS

13

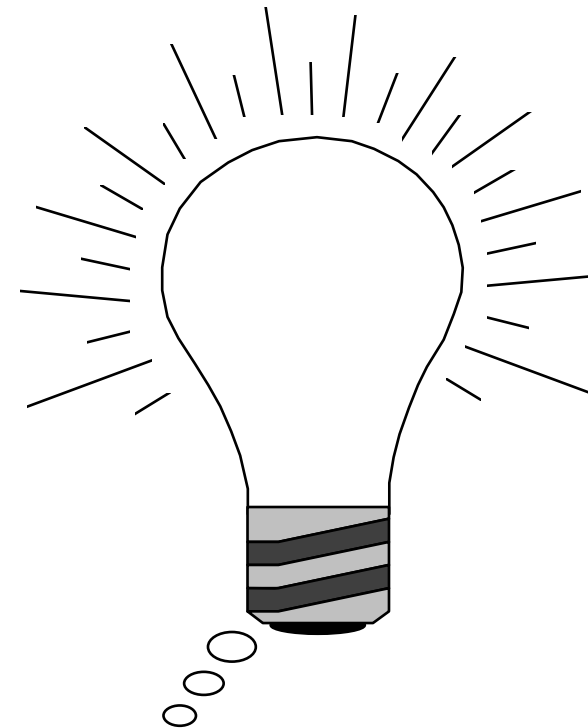
4

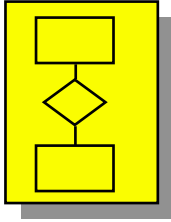


Słownik

Słownik

- Cel
 - zapewnienie integracji całego systemu (wszystkich modeli)

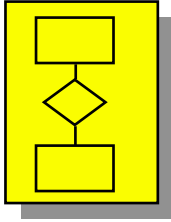




Słownik

Są tu opisy:

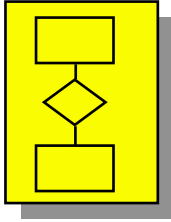
- atrybutów (dopuszczalnych wartości, typów i jednostek elementów danych)
- encji
- relacji
- terminatorów
- magazynów
- złożonych elementów znajdujących się w magazynach



Słownik

Są tu opisy (c.d.):

- elementarnych przepływów
- złożonych przepływów (przez rozbitcie ich na elementarne)
- procesów
- *PSPEC*-i
- informacje o wszystkich obiektach

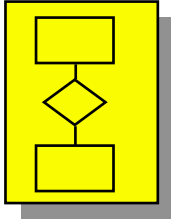


Słownik

Notacja BNF

(Backus-Naur Formalizm)

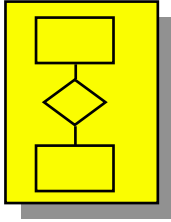
=	równoważność
+	sekwencja (występowanie wszystkich składników po kolei)
{...}	iteracja (powtórzenie)
[.../...]	selekcja (wybór)
(...)	opcja (wystąpienie albo niewystąpienie)
"..."	literał



Słownik

Przykłady

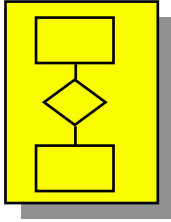
- sytuacja
 - = *opis sytuacji w rozgrywanej grze*
 - identyfikator_sytuacji
 - + numer_sytuacji
 - + opis_sytuacji
- SYTUACJE
 - = {sytuacja}



Słownik

Przykłady

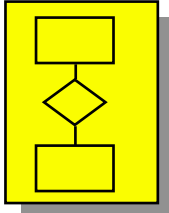
- sytuacja_początkowa
= numer_sytuacji
+ opis_sytuacji
+ identyfikator_sytuacji
- zwycięzca
= ["system" | "gracz" | "rezygnacja gracza" |
"gra trwa"]



Słownik

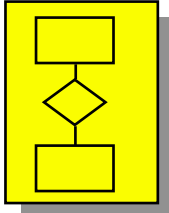
Przykłady

- stopień_trudności
= *określa poziom gry systemu*
["łatwiutki" | "łatwy" | "niezbyt łatwy" |
"średni" | "nieco trudny" | "trudny" |
"niezwykle trudny"]
- strategia
= *sztuka prowadzenia gry*
opis_strategii
+ stopień_trudności



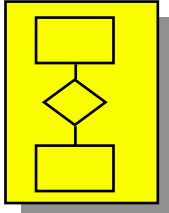
Reguły spójności i poprawności

- *DFD* a *DD*:
 - każdy przepływ i magazyn na *DFD* musi być zdefiniowany w *DD*
 - każdy element danych i magazyn zdefiniowany w słowniku musi być na *DFD*



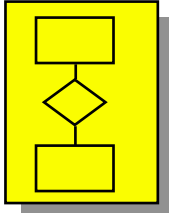
Reguły spójności i poprawności

- *DFD* a *PSPEC*:
 - każdy proces jest zdekomponowany albo ma *PSPEC*
 - każda *PSPEC* musi mieć odpowiadający jej proces prymitywny (bez dekompozycji) na *DFD*
 - wpływy i wypływy na *DFD* i *PSPEC* muszą sobie odpowiadać



Reguły spójności i poprawności

- *PSPEC* a *DFD* i *DD*:
 - każde odwołanie do danych w *PSPEC* jest składnikiem (zdefiniowanym w *DD*) przepływu lub magazynu przyłączonego do procesu prymitywnego albo jest zmienną lokalną (zdefiniowaną w *PSPEC*)
- *DD* a *DFD* i *PSPEC* i *ERD*:
 - każda pozycja w słowniku musi mieć odniesienie w *PSPEC*, *DFD* lub *ERD*



Reguły spójności i poprawności

- *ERD* a *DFD* i *PSPEC*:
 - każdy magazyn na *DFD* musi być zgodny na poziomie atrybutów (lub encji) z *ERD*
 - nazwy encji na *ERD* i odpowiadających im magazynów na *DFD* powinny być podobne
 - dla każdego encji (relacji) na *ERD* powinny istnieć instrukcje CREATE I DELETE (lub im odpowiadające) w *PSPEC*
 - muszą być komendy w *PSPEC*, które czytają i nadają wartości atrybutom dla każdej encji