

IO - inżynieria oprogramowania

dr inż. M. Żabińska,

e-mail: zabinska@agh.edu.pl

<http://home.agh.edu.pl/~zabinska/>

Metody porządkowania wymagań funkcjonalnych

- Liczba wymagań funkcjonalnych może być bardzo duża; konieczne jest pewnego rodzaju uporządkowanie tych wymagań, które ułatwi pracę nad nimi (złożoność !)
- Dwie metody umożliwiające zapanowanie nad dużą liczbą wymagań:
 - **hierarchiczny** zapis wymagań,
 - diagramy **przypadków użycia** (*Use Cases*).

Metody porządkowania wymagań funkcjonalnych

- diagramy **przypadków użycia** (*Use Cases*) →
- **hierarchiczny** zapis wymagań:

Funkcja nadrzędna f1

funkcja f1.1
funkcja f1.2
funkcja f1.3

funkcja f1.3.1
funkcja f1.3.2
funkcja f1.3.3...

Funkcja f1

Funkcja f1.1

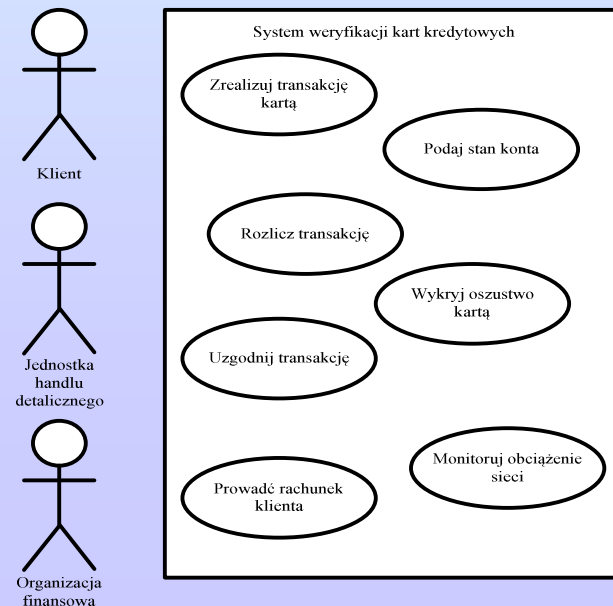
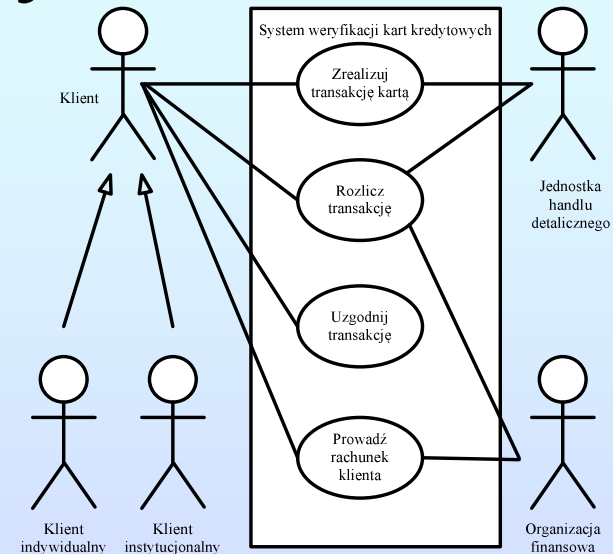
Funkcja f1.2

Funkcja f1.3

Funkcja f1.3.1

Funkcja f1.3.2

Funkcja f1.3.3



Metody analizy

- Dwa podejścia do tworzenia SI (dwie grupy metod analizy)
 - strukturalne (structured methods/analysis)
 - obiektowe (OOA&D)
- Podejście: koncepcja, zasady, notacje
- Zasada dekompozycji i modularyzacji
- Dekompozycja: głównie w warstwie funkcjonalnej systemu
- Powiązanie z zasadą modularyzacji

Strukturalne metody analizy

- Rozwijane od dawna
- Opierają się na wyróżnianiu w analizowanym systemie dwóch rodzajów składowych:
 - Pasywnych – fakt przechowywania w systemie pewnych danych
 - Aktywnych – fakt wykonywania w systemie pewnych operacji
- Budowa dwóch modeli: danych i funkcji (przez różne zespoły); integracja modeli (trudna)

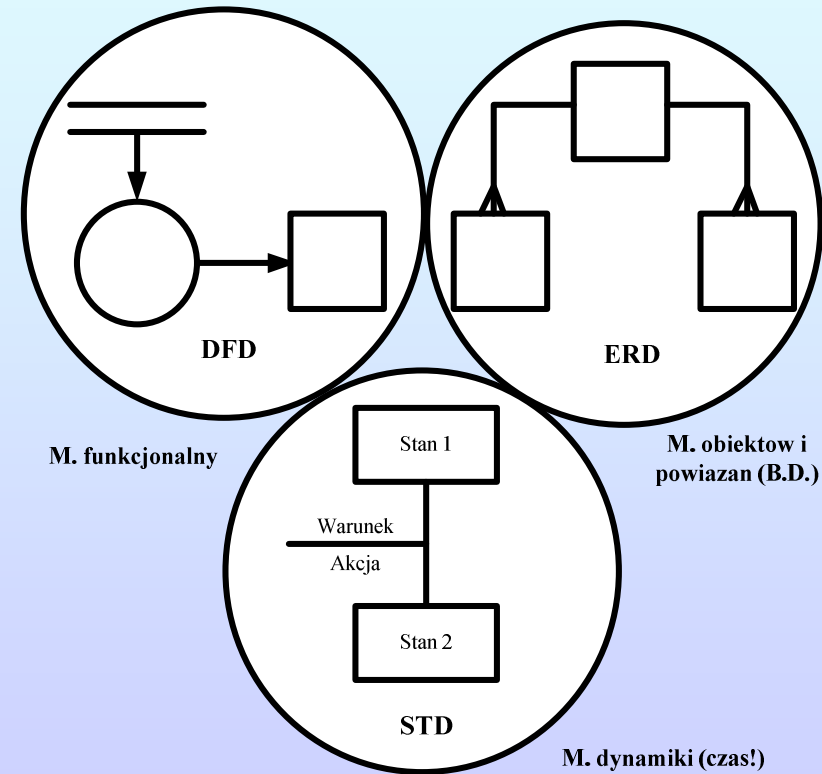
Strukturalne podejście do analizy ⁽¹⁾

Structured Analysis & Design – SA&D lub SA/SD:

- Wg modelu cyklu klasycznego,
- Tworzenie logicznego (podstawowego) modelu systemu, tj. „co system powinien robić, by spełnić postawione wymagania”,
- Problemy i funkcje – iteracyjnie dekomponowane na mniejsze części (reorganizacja i uszczegółowienie modelu zgodnie z wymaganiami), *top-down*

Strukturalne podejście do analizy (2)

- Opis:
 - funkcji systemu,
 - struktur danych,
 - uwzględniając zależności czasowe.
- Koncepcja:
hierarchiczna
dekompozycja
logiczna
(funkcjonalna)



Trzy aspekty (modele) podejścia strukturalnego (SA&D)

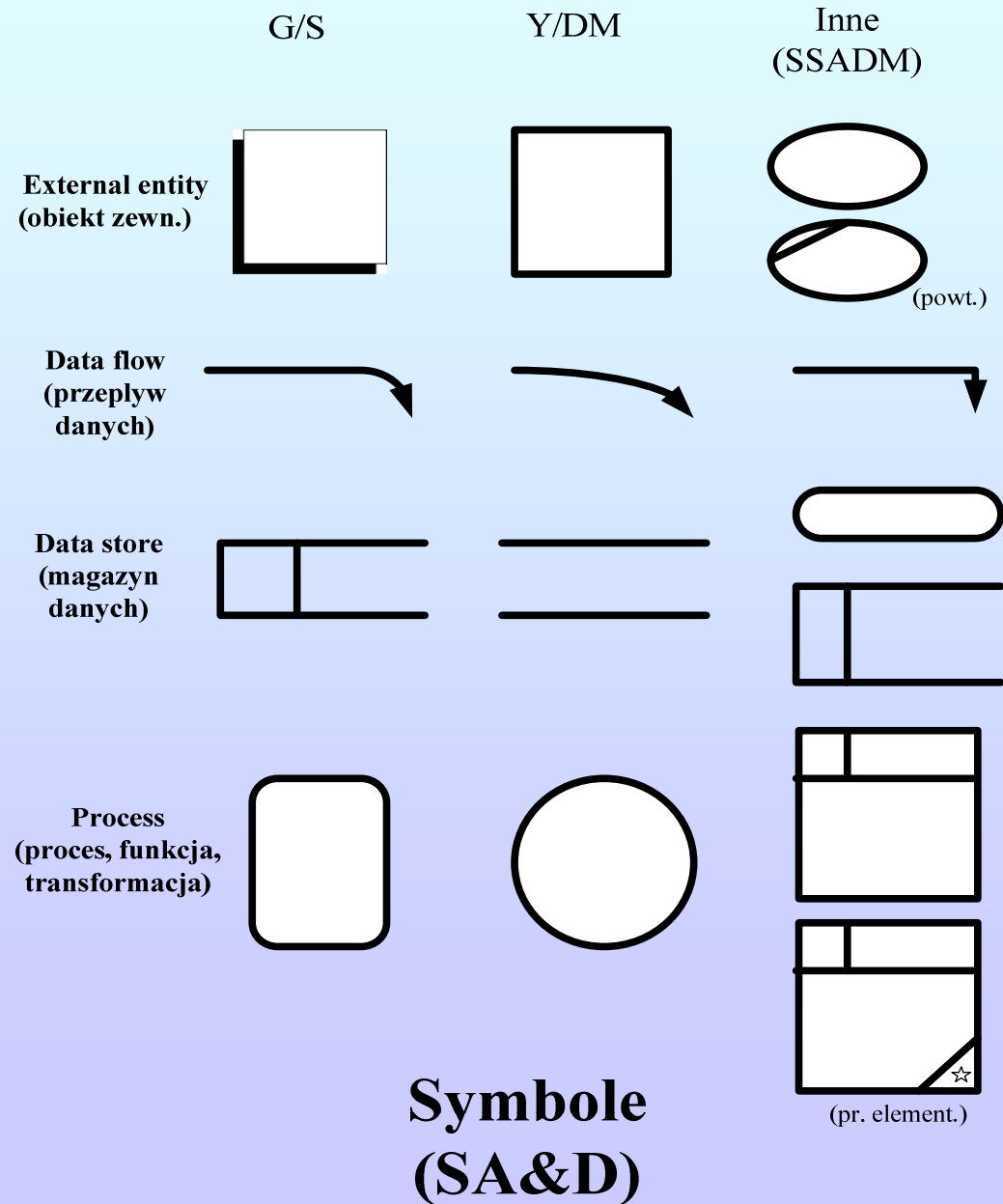
Strukturalne podejście do analizy ⁽³⁾

- Hierarchiczna dekompozycja logiczna (funkcjonalna) – z wykorzystaniem diagramów przepływu danych (DFD – Data Flow Diagrams) i sterowania;
- Uzupełnienie o logiczną reprezentację danych (ERD – Entity Relationship Diagrams)
- Oraz model zależności czasowych (STD – State Transition Diagrams; starsze: ELH – Entity Life History).

Strukturalne podejście do analizy ⁽⁴⁾

- Użycie odpowiednich notacji modelujących (rys.)
- Obowiązują ustalone zasady tworzenia diagramów i dokumentowania
- Koncepcje/notacje:
 - SA/SD (Yourdon, DeMarco; Gane i Sarson),
 - SA/SD/RT (Ward i Mellor) – systemy czasu rzeczywistego,
 - SSADM (System Structured Analysis & Design Model)

Analiza strukturalna - notacje



Charakterystyka modelowania w metodyce strukturalnej

Opis systemu: trzy (aspekty) – trzy modele (rys.)

- Model (aspekt) funkcjonalny – transformacje danych wewnątrz systemu
(Data Flow Diagram **DFD** graf: węzły – *procesy*, łuki – *przepływy danych*)
- Model (aspekt) danych – statyczna struktura systemu
(Entity Relationship Diagram **ERD** graf: węzły – *obiekty* abstrakcja świata rzeczywistego, łuki – *relacje/związki* pomiędzy obiektami)
- Model (aspekt) dynamiki – zmienność w czasie
(State Transition Diagram **STD** graf węzły – *stany*, łuki – *przejścia* pomiędzy stanami, wywoływane przez zdarzenia),
lub Entity Life History **ELH** (obiekty z ERD, *zdarzenia* – *operacje BD*, **SSADM**)

Aspekt funkcjonalny

- **Pierwszy model:** określenie interfejsów pomiędzy systemem a środowiskiem – model środowiskowy – otoczenie systemu (definicja granic pomiędzy systemem a środowiskiem):
 - co jest na zewnątrz systemu a co wewnątrz
 - co dostarczane przez środowisko do systemu
 - co produkowane przez system i dostarczane środowisku

Model środowiskowy

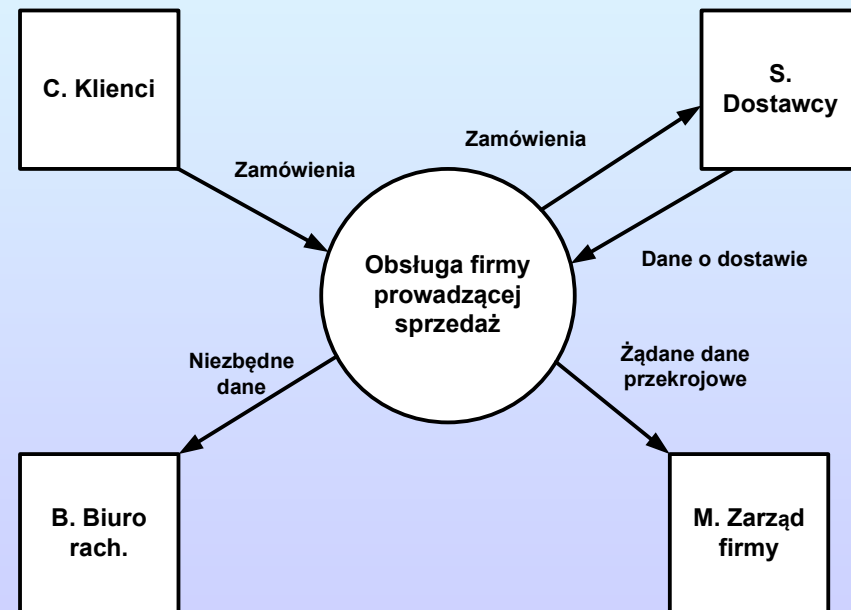
- Składowe modelu środowiskowego:
- definicja celu (krótkie, zwarte, tekstowe określenie celu systemu);
- określenie zdarzeń ("bodźce" z zewnątrz – przepływy danych; zdarzenia temporalne – wewnętrzny zegar systemu; zdarzenia sterujące – przepływy sterujące – binarne),
- diagram kontekstowy (szczególny model: cały system = jeden proces: granice, we/wy – osoby, organizacje, systemy; dane otrzymywane i produkowane/przesyłane na zewnątrz)

Diagram kontekstowy ⁽¹⁾

- Diagram kontekstowy (rys.):
 - określa obszar systemu (**granica** system-otoczenie)
 - obejmuje osoby/organizacje/systemy – komunikacja (**obiekty zewnętrzne**)
 - określa dane z zewnątrz – do przetwarzania (**przepływy**)
 - określa dane wytwarzane przez system przesyłane do otoczenia (**przepływy**)

Diagram kontekstowy (2)

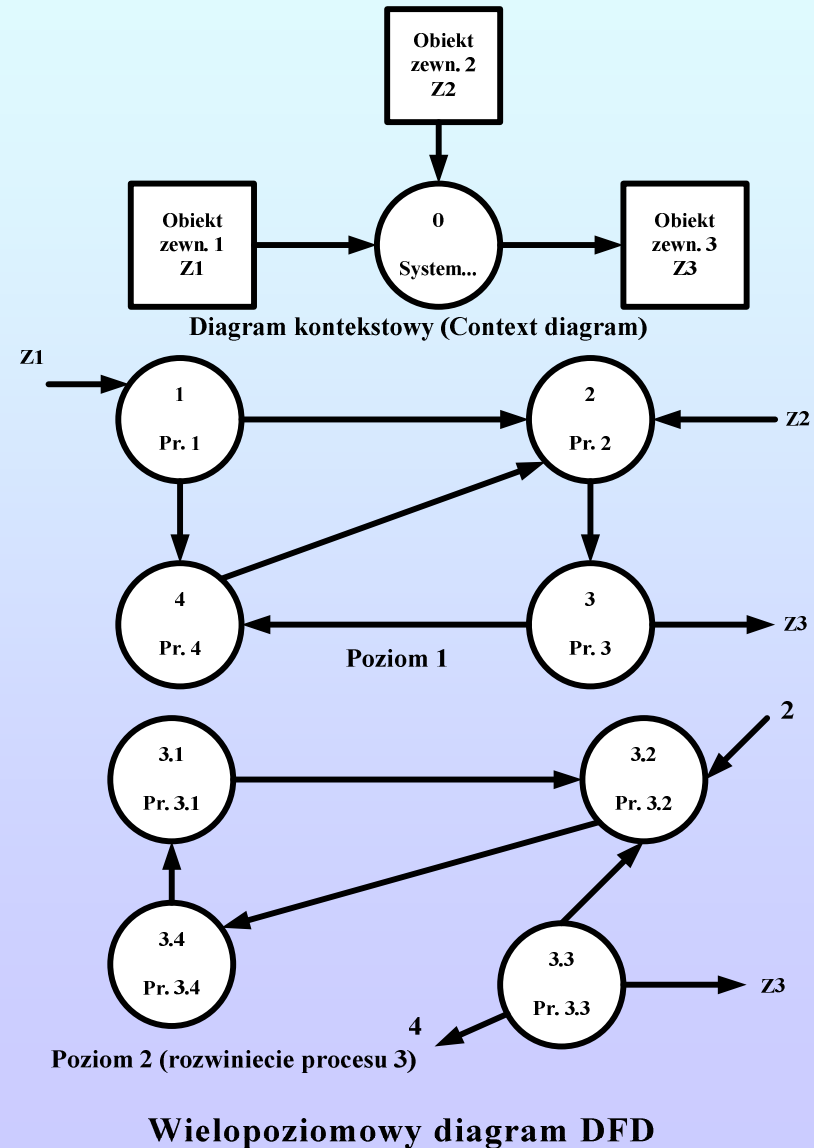
- określa: obszar systemu (**granica** system-otoczenie)
- obejmuje byty – komunikacja (**obiekty zewnętrzne**)
- określa dane do:
 - przetwarzania oraz
 - wytwarzane przez system → **przepływy**)



Model zachowania

- Model zachowania:
- rozwinięcie *Diagramu Kontekstowego* – *Context Diagram*
- **oraz** model zależności czasowych);
- Procedura:
 - Tworzenie DK (CD) i rozwijanie top-down, uszczegółowienie w dół, kolejne poziomy abstrakcji, do diagramów niższych rzędów

Diagram kontekstowy i rozwinięcie



Model zachowania - uwagi

- Rozwijanie w dół – prosty system: 2-3 poziomy, średni: 3-5 poziomów, duży: do 8 poziomów,
- Magiczne "7" – liczba obiektów do analizy
- Nie wszystkie procesy (części systemu) mają równy stopień złożoności – nie muszą być rozwijane do tej samej liczby poziomów (kryterium: procesy elementarne)
- Bilansowanie diagramów (przepływy wchodzące i wychodzące z procesu na pewnym poziomie muszą odpowiadać przepływowi we/wy na niższym poziomie – rozwinięciu tego procesu); bilansowanie magazynów danych

Modelowanie strukturalne

- Tworzone:
 - Opis celu systemu, zdarzeń zewnętrznych, Diagram Kontekstowy (Context Diagram – DFD0) = **model środowiskowy**
 - Diagramy (DFD) kolejnych poziomów (rys.),
 - Diagramy dynamiki (STD) z opisem
 - Diagramy obiektów i powiązań/związków encji (ERD) z opisem (uzasadnienie wyboru obiektów, struktury)
 - **Opisy uzupełniające:**
 - *specyfikacje* przetwarzania (jak formularz opisu WF uzupełnione o pseudokod),
 - *słownik danych* – jednoznaczna i niesprzeczna definicja obiektów (koncepcja Yourdona).

Modelowanie strukturalne - wyniki

- Wyniki fazy modelowania:
- Model środowiska (**SA&D – Model środowiskowy**),
- **Model zachowania** (rozwińnięcie DK – Context Diagram oraz **model** zależności czasowych);
- **Model obiektów** i powiązań
- Słownik danych (definicje obiektów)
- Specyfikacje funkcji (opis przetwarzania – poziom najbardziej szczegółowy – DFDn)

DFD – kroki (1)

Kolejne fazy modelowania systemu (pierwsze cztery kroki wg *T. DeMarco*):

- Model fizyczny systemu istniejącego (*current physical DFD*),
- Model logiczny systemu istniejącego,
- Model logiczny systemu wymaganego (*required logical DFD*),
- Model fizyczny systemu wymaganego.

DFD – kroki (2)

Tworzenie DFD :

- **identyfikacja** kluczowych **dokumentów** wykorzystywanych w systemie oraz wytwarzanych przez system,
- **opis** fizycznego **przepływu** ze wskazaniem źródła i odbiorcy danych (DFD, na którym są tylko obiekty zewnętrzne wystawiające i odbierające dokumenty narysowane w postaci przepływów),
- **uzgodnienie** z użytkownikiem **granic** tworzonego systemu i zaznaczenie ich na diagramie przepływu,

DFD – kroki (3)

Tworzenie DFD, c.d. :

- **identyfikacja procesów** pobierających/ wytwarzających dokumenty: dla każdego przepływu przecinającego granice systemu - zidentyfikowanie i narysowanie procesu pobierającego ten dokument (dla wpływających) lub wytwarzającego dokument (dla opuszczających system) oraz niezbędnych magazynów danych),
- **uzupełnienie** o dalsze przepływy i procesy jeśli konieczne,
- **weryfikacja** DFD poprawności - bilanse (procesów i magazynów danych),
- **weryfikacja** z użytkownikiem.

DFD – weryfikacja ⁽¹⁾

Weryfikacja:

- nazwy i identyfikatory,
- poprawność użycia symboli,
- wykorzystanie danych przez proces (czy dane wpływające do procesu są rzeczywiście wykorzystywane przez proces),
- bilans poziomy procesów (porównywanie zawartości przepływów danych i składnic danych; UWAGA: istnieją dane wytwarzane i/lub pochłaniane przez proces !),

DFD – weryfikacja (2)

- bilans poziomy magazynów danych: suma zawartości przepływów danych wpływających = sumie zawartości przepływów wypływających z MD,
- złożoność diagramu (czytelność !):
 - rozwijanie pionowe (bilansowanie pionowe - suma przepływów wpływających/wypływających na diagramie-rodzicu równa sumie odpowiednich przepływów na diagramie-potomku),
 - rozwijanie poziome (bilanse).

DFD – weryfikacja (3)

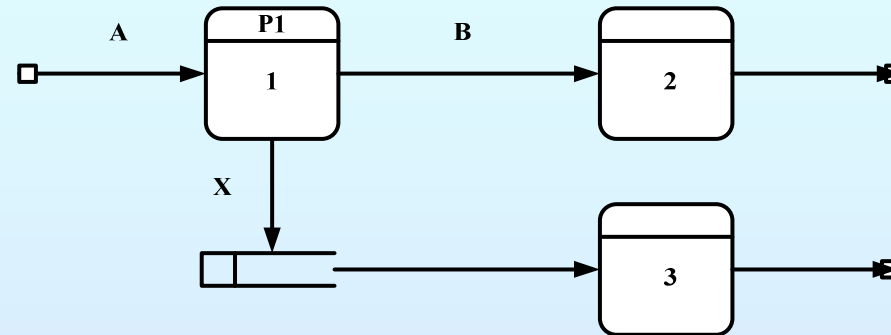


Diagram rodzic - bilans pionowy

$$A=R+S$$
$$X+B=X+B1+B2$$

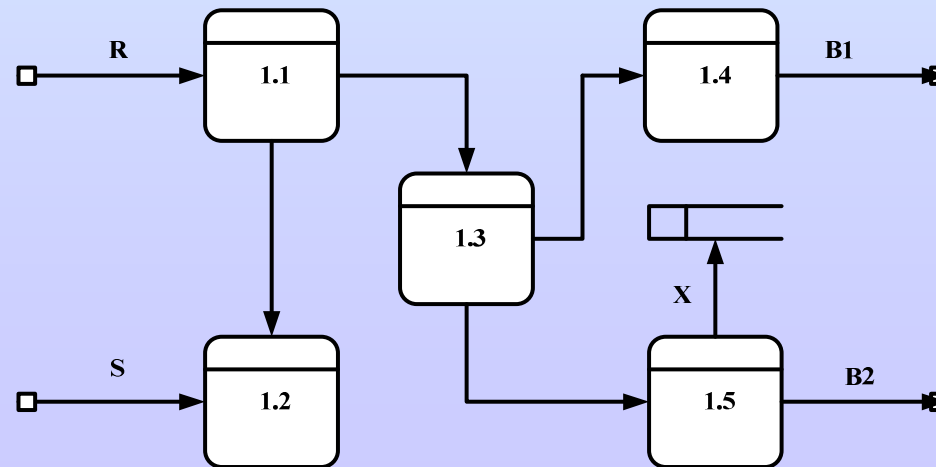
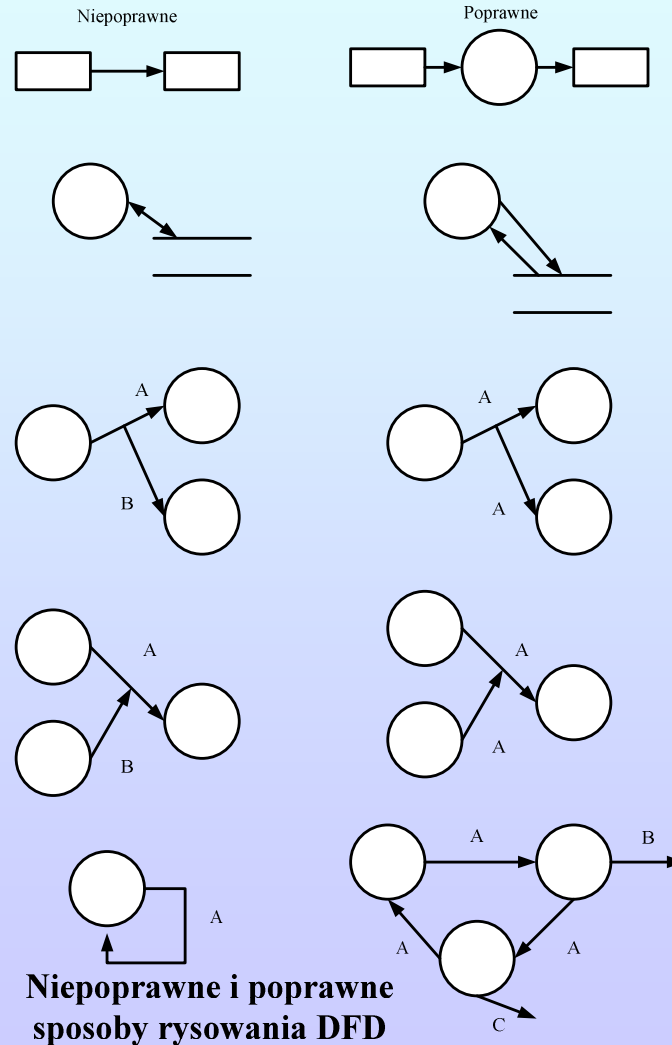


Diagram potomek - bilans pionowy

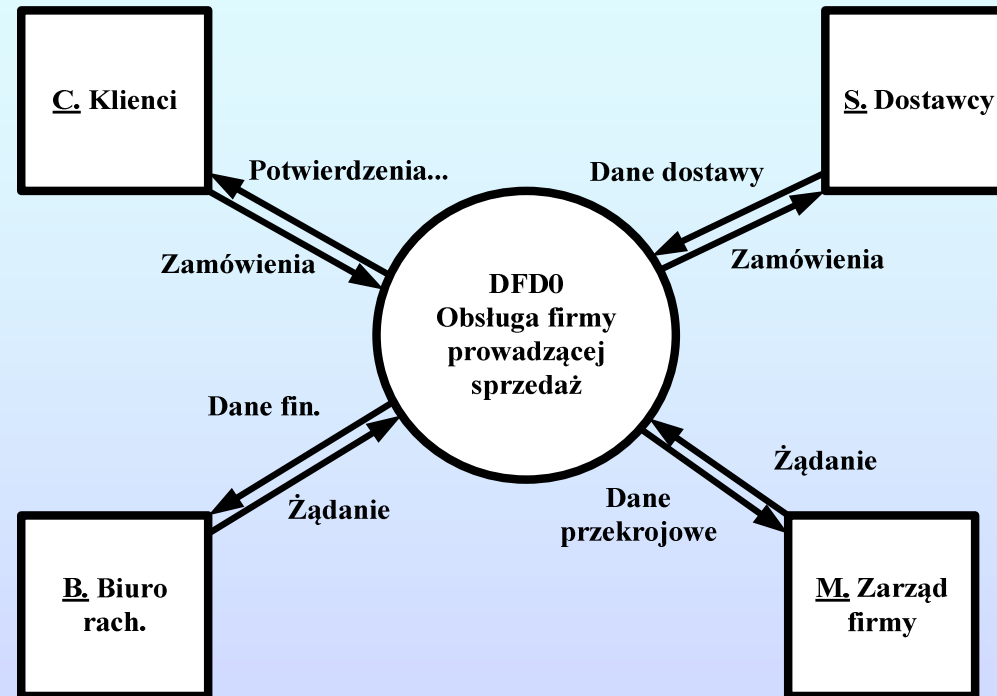
Przykład – typowe błędy



Przykład – model środowiskowy (1)

- Model środowiskowy *dla Małej Firmy Handlowej*:
 - ❖ ***Cel systemu:***
 - Usprawnienie pracy w zakresie przyjmowania zamówień od Klienta, tworzenia zamówień do Dostawcy, tworzenia raportów dla Zarządu Firmy, tworzenia zestawień finansowych
 - ❖ ***Zdarzenia (Lista zdarzeń):***
 - przybycie zamówienia od Klienta,
 - dostawa wraz z danymi,
 - otrzymanie żądania od Zarządu Firmy i/lub Biura Rachunkowego.
 - Uwaga:**
 - zamówienia (Klient, Dostawca) – różne;
 - żądania (od Zarządu Firmy, Biura Rachunkowego) – różne.
- Potem stworzymy:**
 - ❖ Diagram Kontekstowy

Przykład – DK (CD) (1)



F5 - Diagram kontekstowy dla małej firmy prowadzącej sprzedaż

Przykład – model środowiskowy

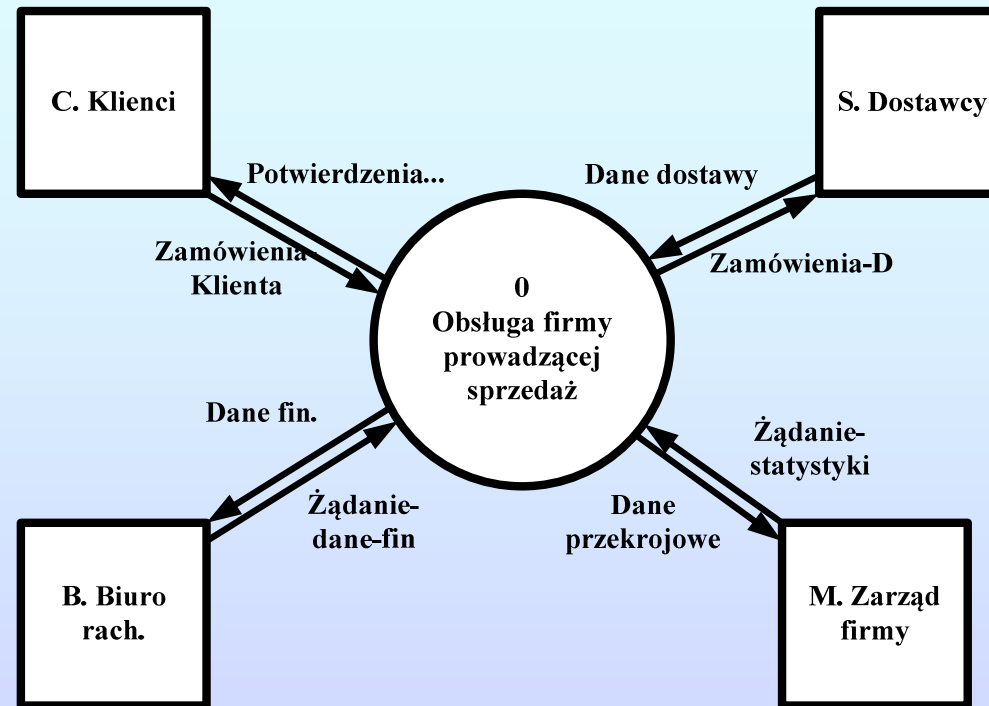
Uwaga:

- zamówienia (Klient, Dostawca) – różne
- żądania (od Zarządu Firmy, Biura Rachunkowego) – różne

Potem stworzymy jeszcze:

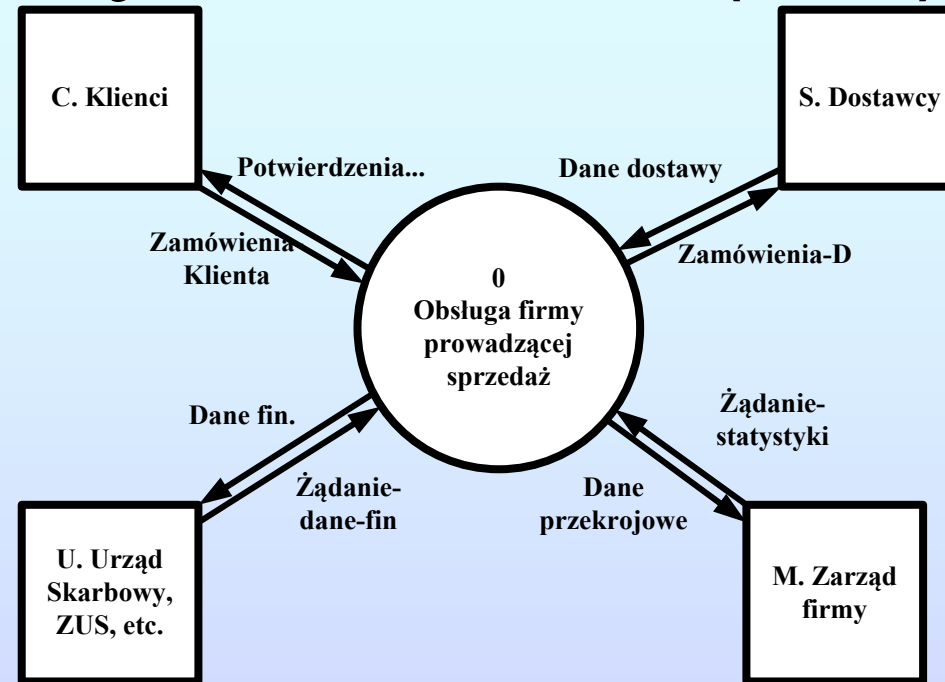
- **Model zachowania (rozwińcie Diagramu Kontekstowego – Context Diagram) aż do DFn oraz**
- **Model zależności czasowych (dynamiki systemu)**
- **Model obiektów i powiązań (dla BD)**
- **Słownik danych (definicje obiektów)**
- **Specyfikacje funkcji (opis przetwarzania – dla poziomu najbardziej szczegółowego – DFDn)**

Przykład – DK (CD) (2)



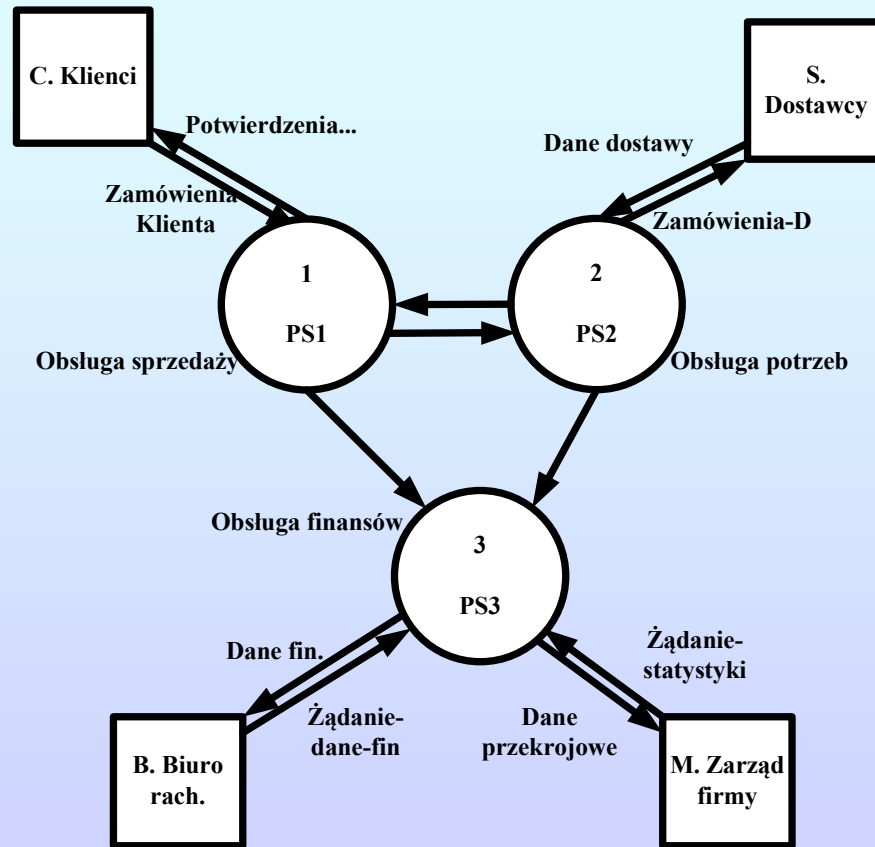
F5 a - Diagram kontekstowy dla małej firmy prowadzącej sprzedaż

Przykład – DK (CD) (3)



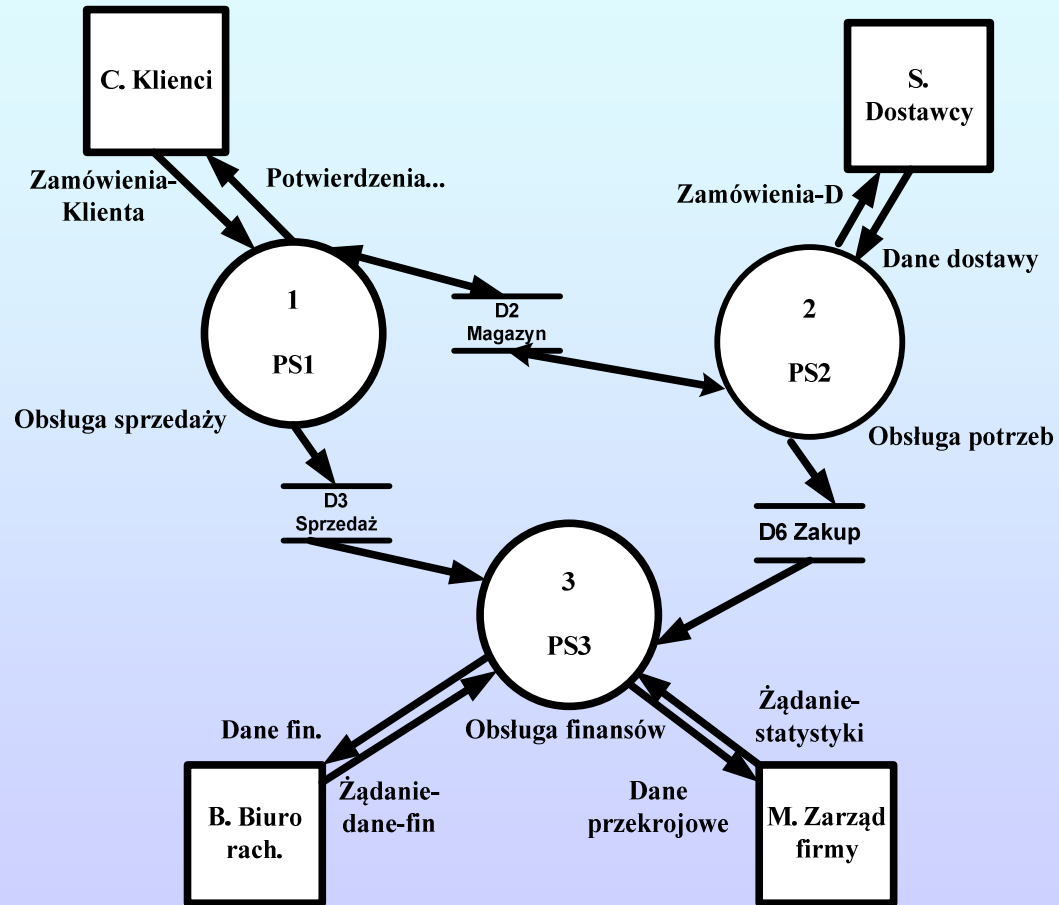
F5 b - Diagram kontekstowy dla małej firmy prowadzącej sprzedaż – zmiana zakresu systemu

Przykład – DFD1 (1)



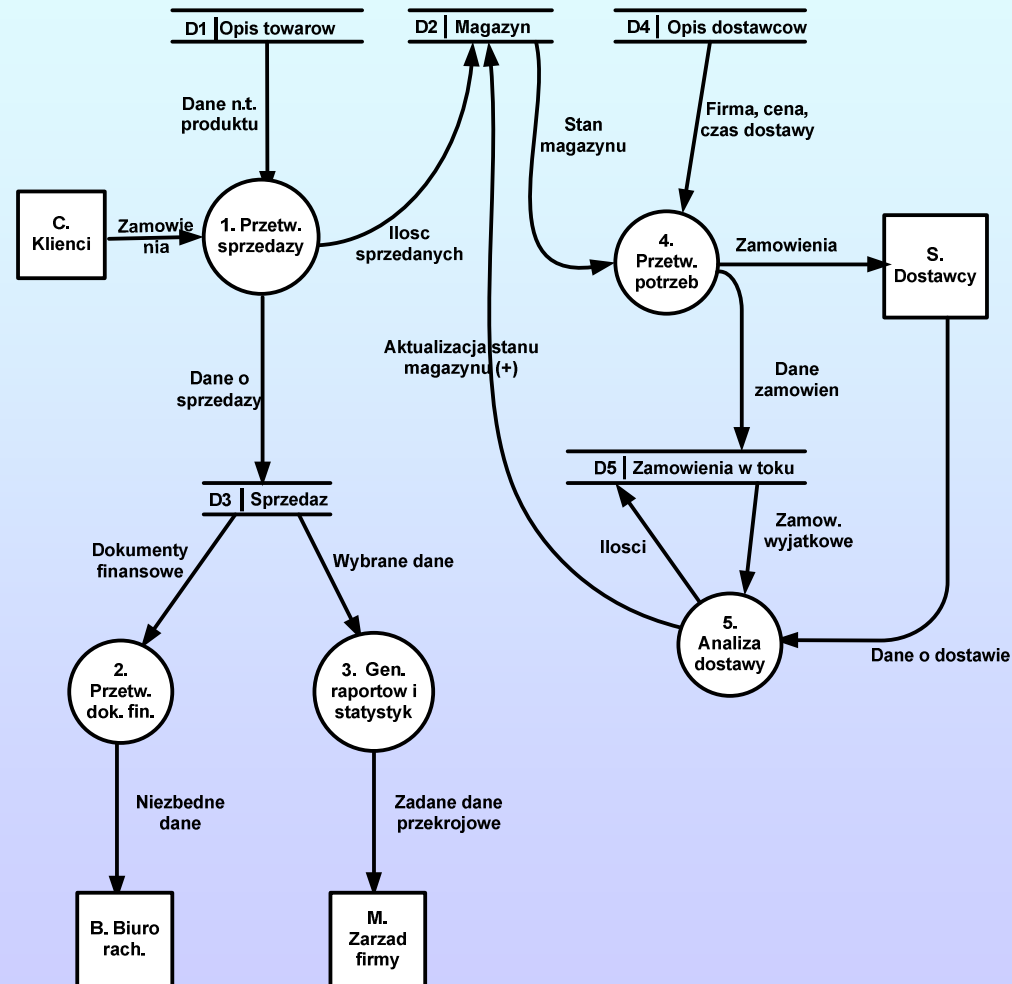
F5 - Diagram poziomu 1 – podsystemy – dla małej firmy prowadzącej sprzedaż (lub rozwinięty DK)

Przykład – DFD1 (2)



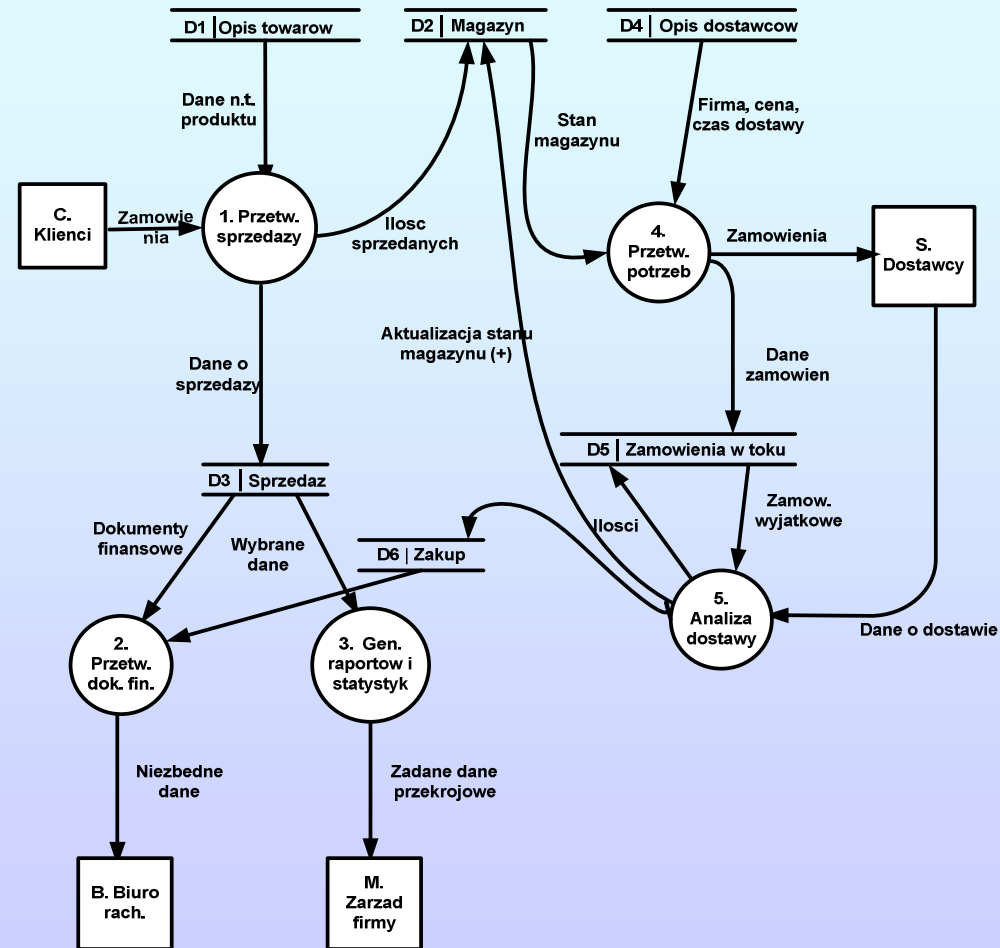
F5 - Diagram poziomu 1 (podsystemy) dla małej firmy prowadzącej sprzedaż

Przykład – DFDn



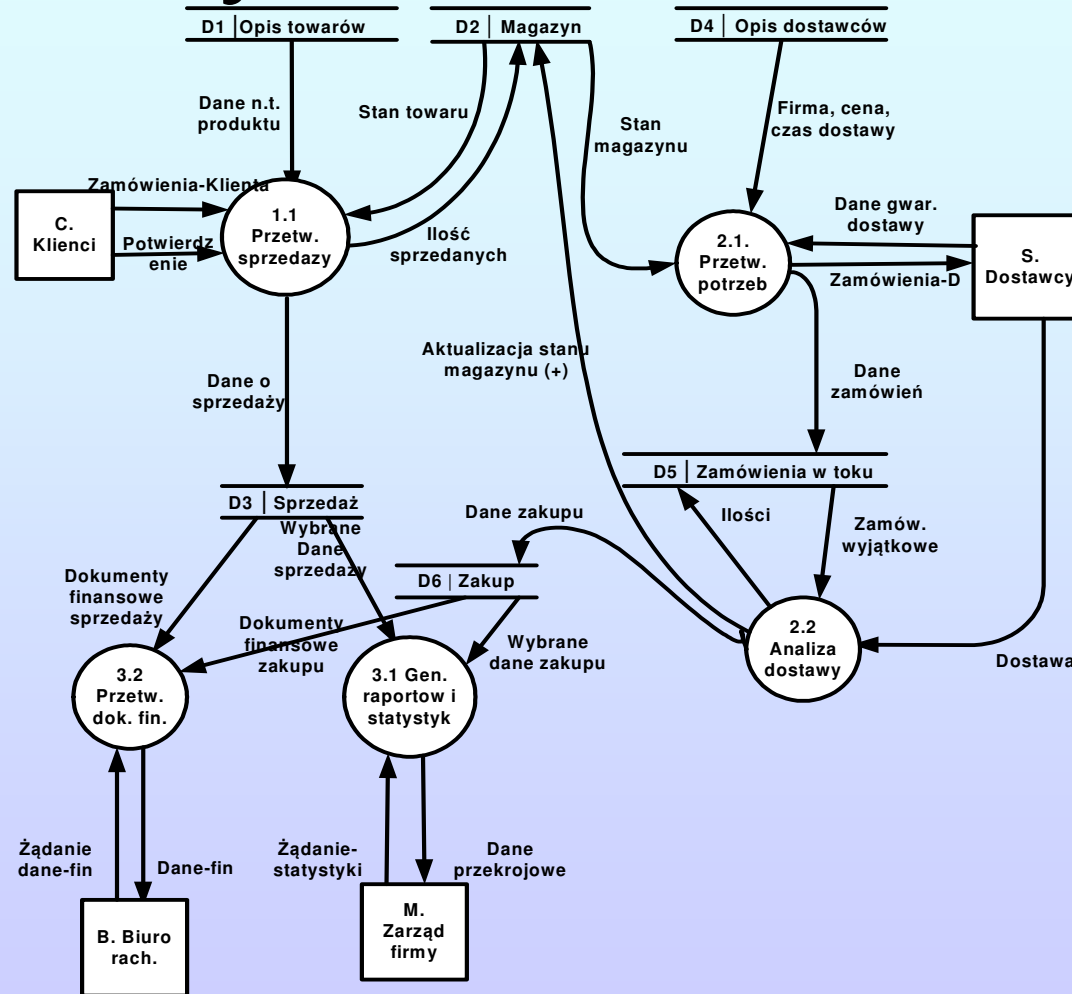
F3 - Diagram DFD dla małej firmy prowadzącej sprzedaż towarów

Przykład – DFDn c.d.



F3 - Diagram DFD dla małej firmy prowadzącej sprzedaż towarów

Przykład – DFDn c.d.



F3 - Diagram DFD dla malej firmy prowadzącej sprzedaż towarów

Model obiektów i powiązań

Entity Relationship Model (ERM) – ER Diagrams (ERD)

Cel

- pokazanie dla pewnej dziedziny:
- obiektów w systemie
- powiązań pomiędzy nimi

DFD przedstawia procesy i dane (w formie magazynów/składnic danych - Data Stores), natomiast ERD koncentruje się wyłącznie na danych (zawartości Data Stores).

ERD (1)

Elementy diagramu ERD:

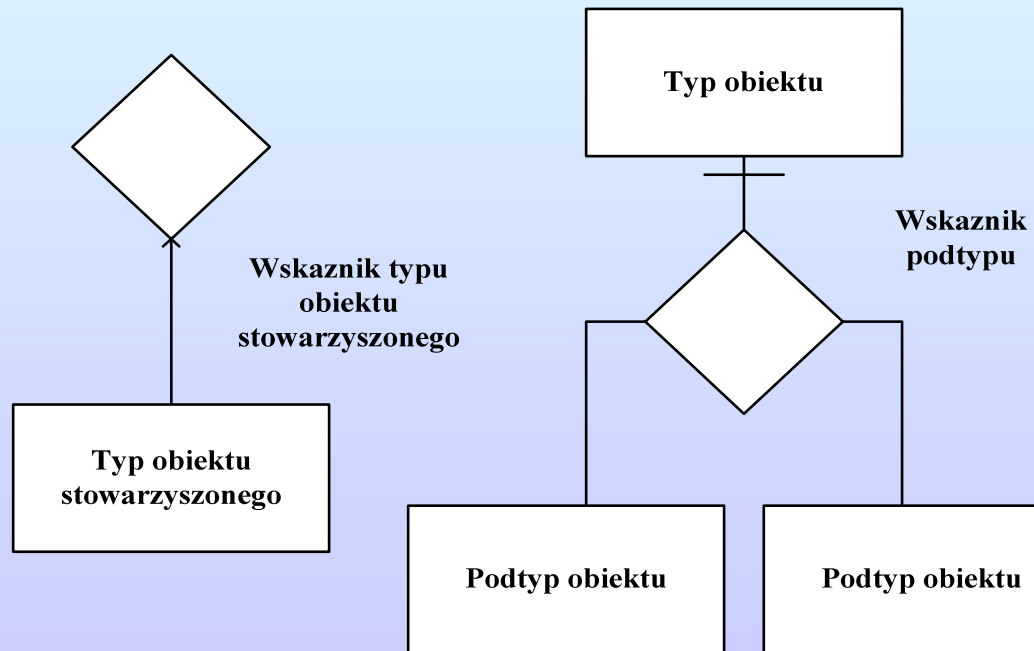
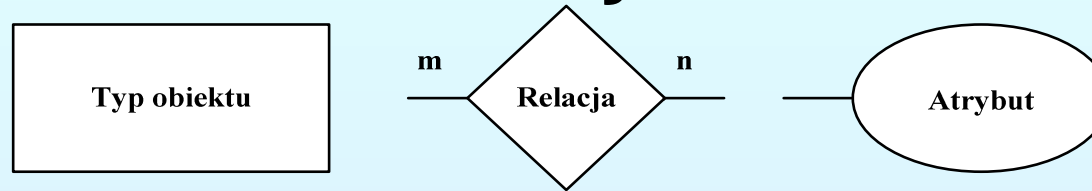
- generalnie obiekty na ERD odpowiadają magazynom (składnicom) danych
- **Data Stores** na DFD
- podstawowe komponenty ERD:
 - typy obiektów
 - relacje (związki)

ERD (2)

Główne fazy tworzenia modelu „informacyjnego”:

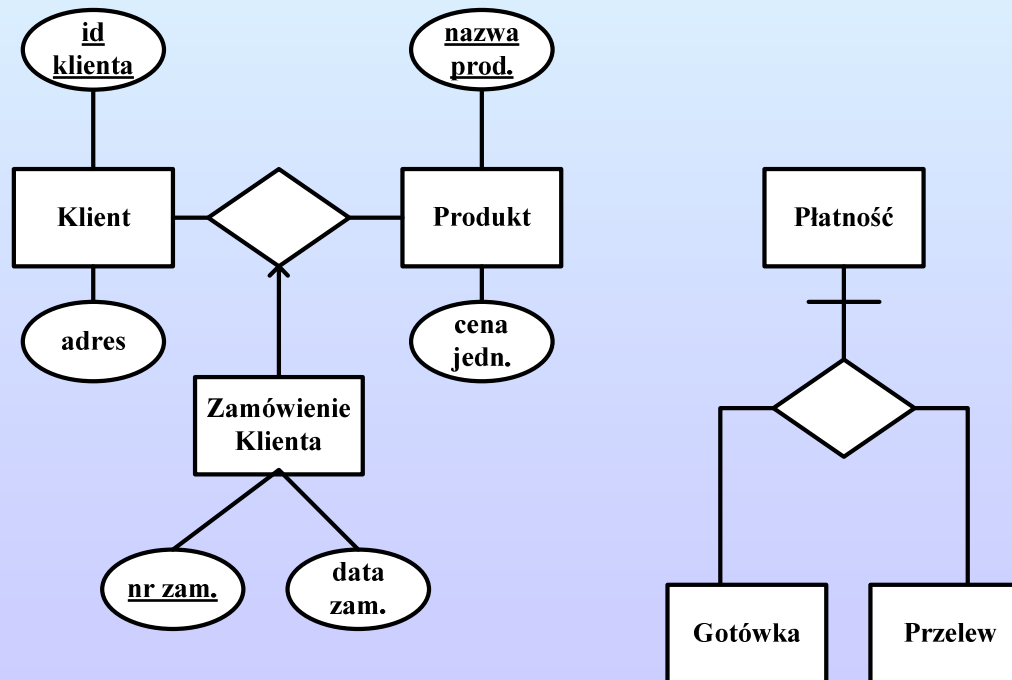
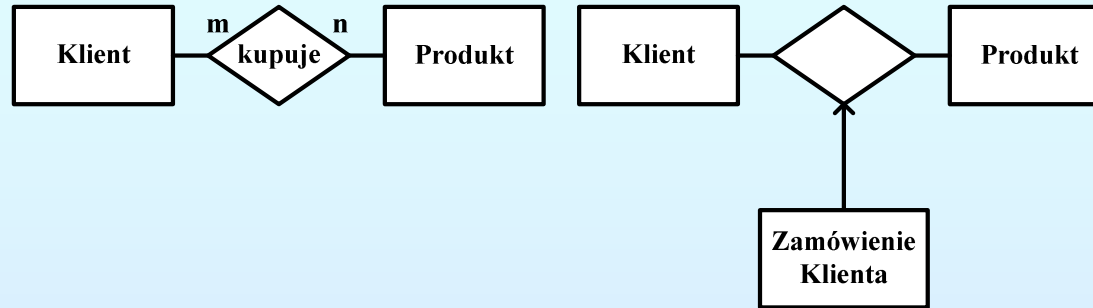
- identyfikacja obiektów (d.: danych (jednostkowych)) – *Data Stores na DFD*,
- określenie związków (relacji) pomiędzy obiektami (d.: danymi) - *tablica krzyżowa obiektów*:
 - fakt istnienia związku (i jego charakter),
 - typy powiązań - relacje ilościowe (cardinality),
 - rodzaje związków,
 - rola obiektów z związku (np. nadrzędny, podrzędny),
- przypisanie atrybutów obiektom (*uzupełnienie* na ERD - np. wg notacji Chena lub innej – *opis*),
- opis – *Data Dictionary (DD)*.

ERD – notacja Chena (1)



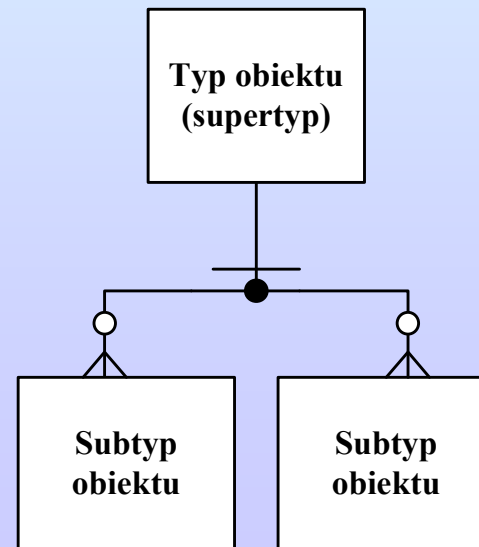
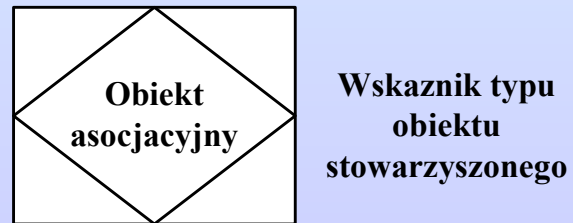
ERD - notacja Chena

ERD – notacja Chena (2)



ERD - notacja Chena - przykłady

ERD – notacja Martina



ERD (3)

Związki (powiązania, relacje) pomiędzy obiektami:

- charakter związku - opis słowny - nazwa związku (powiązania)
- typy powiązań (wg liczebności związku - cardinality):
 - jeden-do-jeden (1:1), jeden-do-wielu (1:N), wiele-do-wielu (M:N).
- rodzaje związków:
 - opcjonalne, obligatoryjne, rekurencyjne.

ERD – proces tworzenia ⁽¹⁾

Identyfikujemy:

1. obiekty – istnienie, nazwy (Data Stores);
2. fakt istnienia relacji (związku), charakter związku (nazwa);
3. relacje ilościowe (typy powiązań), liczebność (cardinality);
4. rodzaje związków (opcjonalne, obligatoryjne, rekurencyjne);
5. role obiektów w związku (hierarchie)

Tablica krzyżowa obiektów w systemie

	Klient	Wyrób	Region	Magazyn	Zamówienie
• Klient			X		X
• Wyrób				X	X
• Region	X			X	
• Magazyn		X	X		
• Zamówienie	X	X			

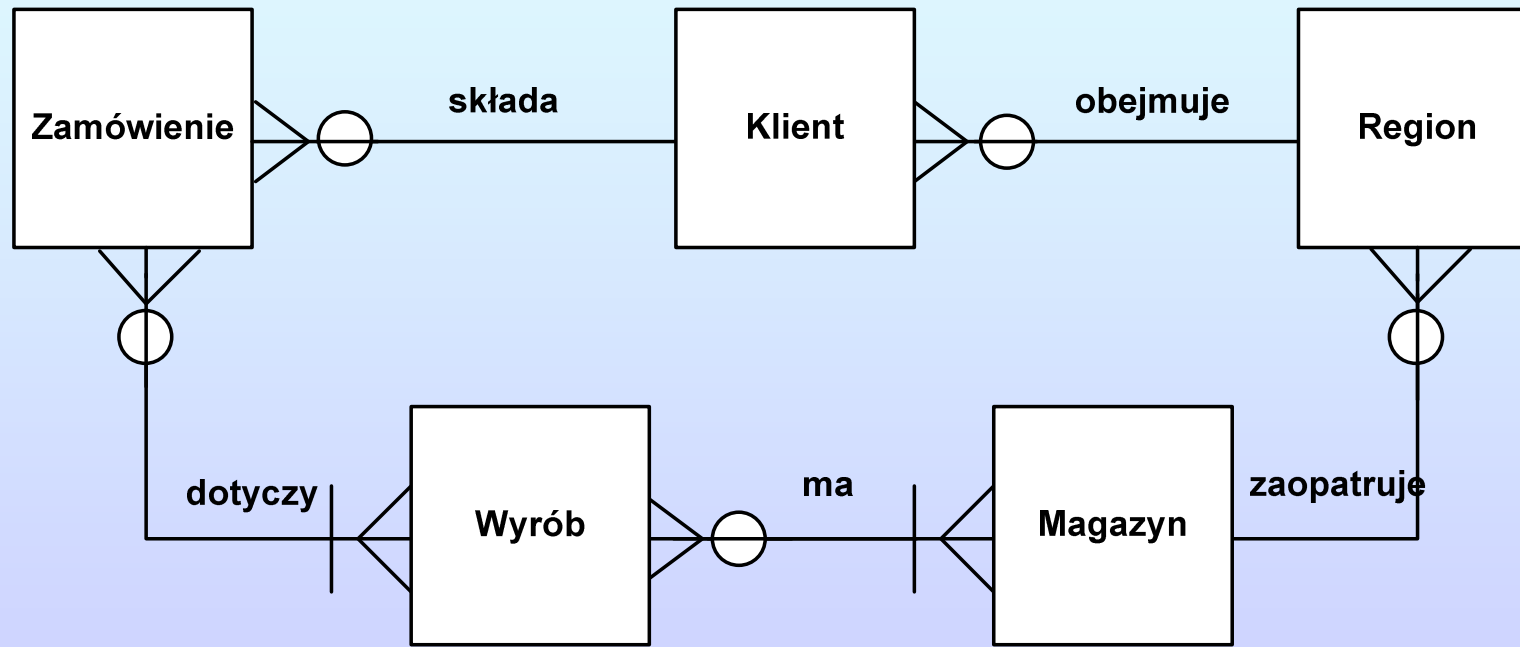
- X- istnienie bezpośredniego, istotnego powiązania,
- np. Klient *składa* Zamówienie (1:N), opcja *składa* (Ad 2);
1:N (Ad 3);
opcja (Ad 4).

ERD – proces tworzenia (2)

Modelowanie "informacji" - kroki:

- **Identyfikacja** (wydzielenie) zbioru **obiektów** (grup danych) w systemie z ich atrybutami *kluczowymi*,
- **Identyfikacja** bezpośrednich **zależności** pomiędzy obiektami oraz ich *typu i rodzaju*: tablica krzyżowa (*punkty 1, 2, 3, 4*),
- **Utworzenie** pojęciowego **modelu danych** – ERD (np.przekształcenie tablicy krzyżowej oraz identyfikacja pozostałych atrybutów obiektów: (O-A-Z). Opis formalny: *Słownik Danych*,

ERD – przykład (1)



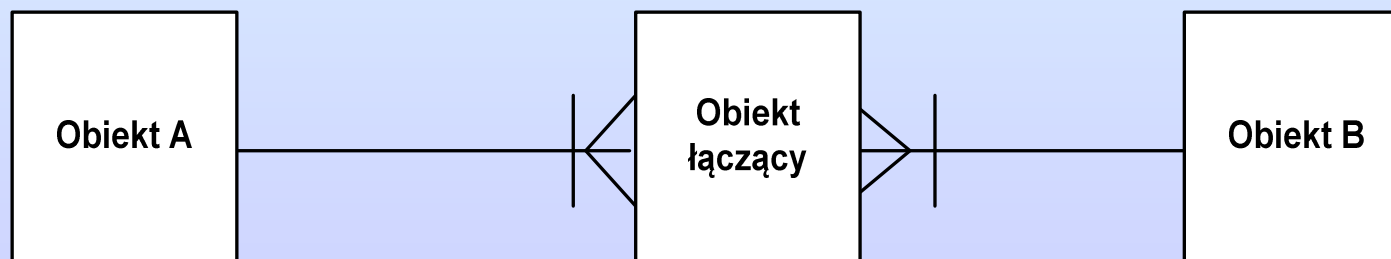
Wstępny model danych dla firmy X

ERD – proces tworzenia (3)

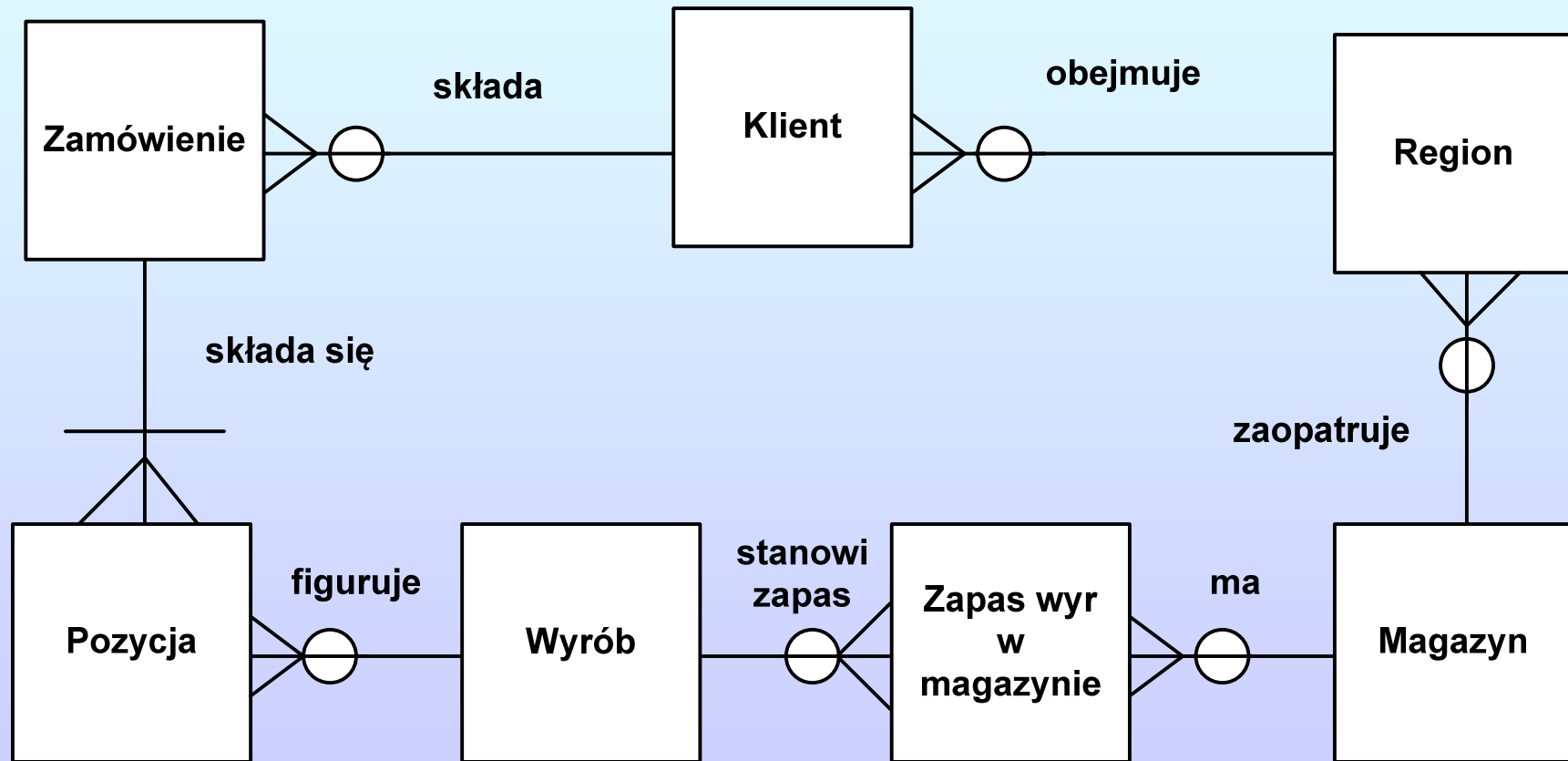
Modelowanie "informacji" - kroki:

- **Przekształcenie** powiązań typu **wiele do wielu** (tj. każde powiązanie typu M:N przekształcamy na dwa - typu 1:N); identyfikacja dodatkowych atrybutów nowo powstałych obiektów (na potrzeby relacyjnej bazy danych),
- **Weryfikacja** uzyskanego modelu danych: przez **porównanie z wymaganiami** odnośnie systemu (dostęp: do obiektu, hierarchii, rodzaje zapytań),
- **Weryfikacja DFD względem ERD**: każdy obiekt z ERD powinien znaleźć się w pewnym magazynie danych z DFD (metoda: np. tworzenie tablicy krzyżowej obiekt/magazyn danych).

ERD – przykład (2)



ERD – przykład (3)



Docelowy model danych dla firmy X

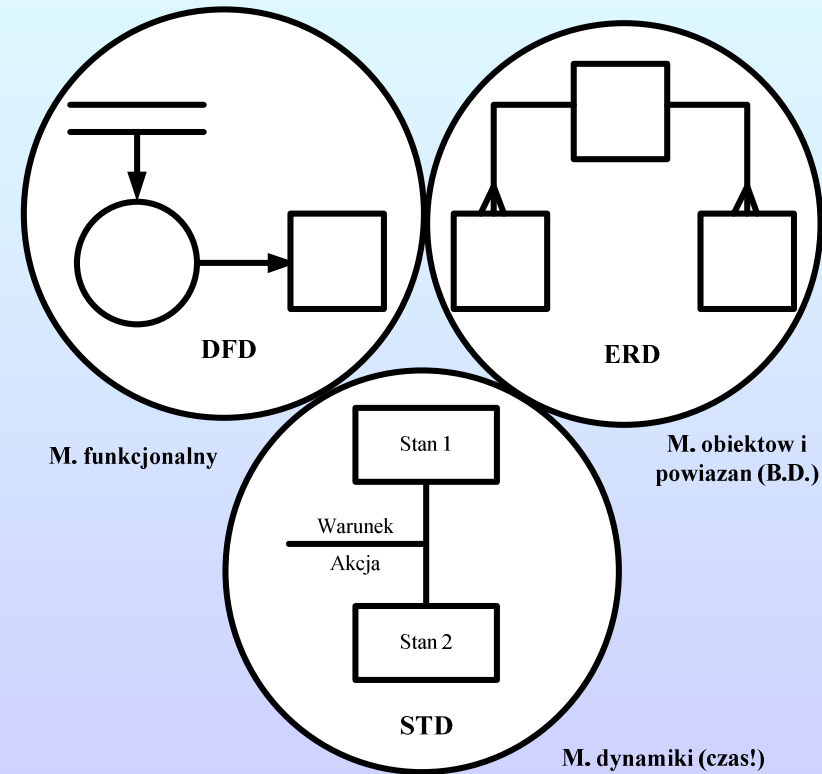
ERD– dalsze kroki

Modelowanie "informacji" – dalsze kroki:

- **Opis: *Słownik Danych - DD*** (atrybuty, obiekty, powiązania - w pewnej konwencji, np. Yourdona, narzędzia CASE)
- **Rola:** jak specyfikacje procesów dla DFD
- Ewentualne **generowanie opisu** (np. w **SQL**, narzędzia)

Trzy aspekty

- Opis:
 - funkcji systemu,
 - struktur danych,
 - uwzględniając zależności czasowe.
- Koncepcja:
 - hierarchiczna
 - dekompozycja
 - logiczna
 - (funkcjonalna)



Trzy aspekty (modele) podejścia strukturalnego (SA&D)

Modelowanie dynamicznych aspektów systemu

Cel:

- przedstawienie zmian stanu obiektów w czasie
- trzecia płaszczyzna widzenia systemu oparta na zdarzeniach (z DFD), które oddziałują na obiekty (z ERD).

DFD przedstawia procesy i dane

*ERD koncentruje się wyłącznie na danych
(zawartości Data Stores) - podejście statyczne*

Potrzeba pokazania dynamiki systemu

Modelowanie dynamicznych aspektów systemu

- ***Stan [Y] - zbiór okoliczności lub cech charakteryzujących obiekt w danej chwili***
- Do opisywania RT (głównie)
- Obejmuje trzy typy obiektów: stan, przejście, interface (warunek/akcja)
- Dobrze uzupełnienie DFD (pokazuje następstwa czasowe procesów z DFD): warunki = zdarzenia - przepływy danych wejściowych (powodują zmianę stanu); akcje - dane wyjściowe z procesu
- Analiza Strukturalna [Y] oraz OOA+OOD [C+Y] i in., np. UML

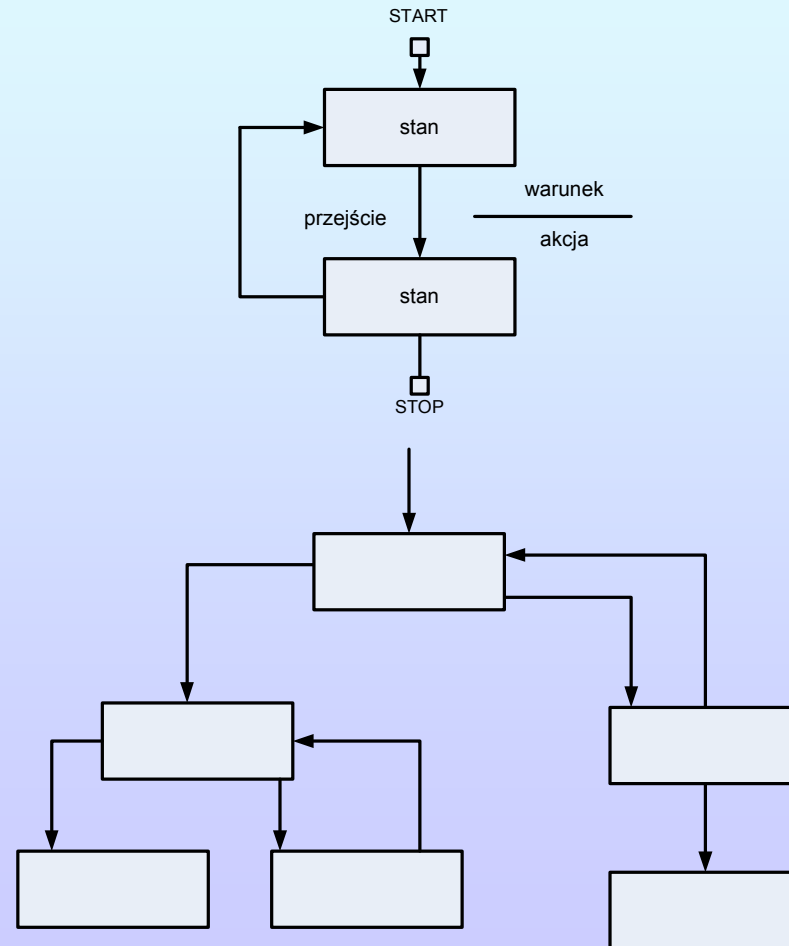
Modelowanie dynamicznych aspektów systemu

Symbole używane na diagramach STD.

Pokazywanie warunków i akcji:

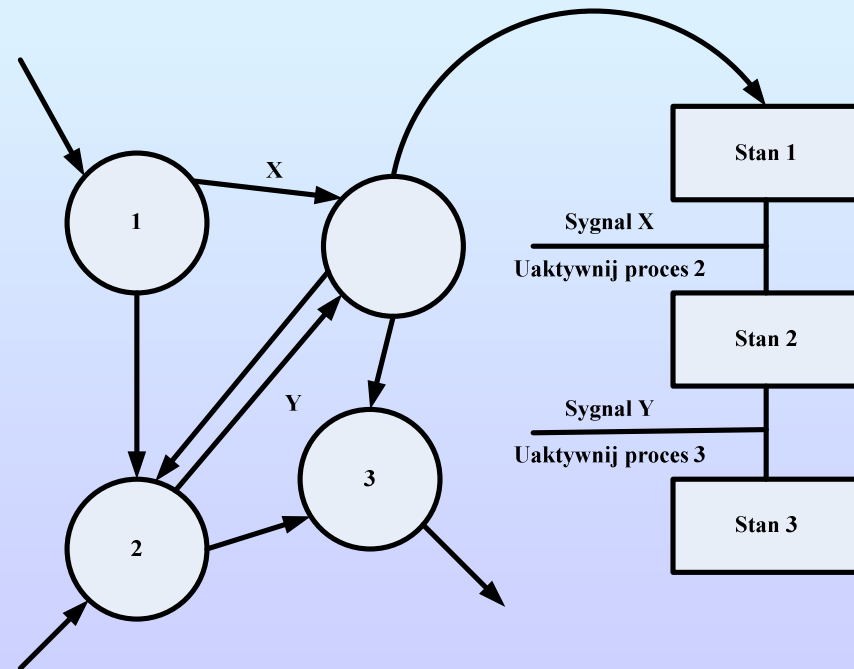
stan, przejście, warunek, akcja, interfejs.

[Yourdon „Analiza strukturalna”]



Modelowanie dynamicznych aspektów systemu

- **Związek między DFD a STD**



Modelowanie dynamicznych aspektów systemu

Budowanie STD

- Identyfikacja wszystkich możliwych stanów obiektu/systemu - obraz graficzny (prostokąty), potem badanie sensownych połączeń (zmian stanów),
- Określenie stanu początkowego dla obiektu, potem metodyczne przechodzenie do następnych stanów.

Modelowanie dynamicznych aspektów systemu

Sprawdzanie diagramów STD

- Czy zdefiniowano wszystkie stany,
- Czy wszystkie stany są osiągalne (każdy stan dostępny ze stanu początkowego),
- Czy istnieją wyjścia ze wszystkich stanów (stan końcowy dostępny dla każdego stanu),
- Czy warunek \rightarrow przejście ze stanu do tylko jednego innego stanu,
- Czy w każdym stanie poprawna odpowiedź,
- Czy uwzględniono sytuacje nieokreślone.

Modelowanie dynamicznych aspektów systemu

Diagram przejść/zmian stanów STD (Diagram Stanów)

- Technika opisu zachowania obiektu, różne notacje;

Opis:

- wszystkich możliwych stanów, do których może przejść dany obiekt
- jak zmienia się stan obiektu pod wpływem zdarzeń do niego docierających

Modelowanie dynamicznych aspektów systemu

Stany

- Stan jest chwilą w życiu obiektu
- Stan jest odpowiedzią obiektu na zdarzenia
- Stan reprezentuje przedział czasowy między dwoma zdarzeniami oddziałującymi na obiekt
- Zdarzenia to punkty w czasie, stany reprezentują okresy czasu

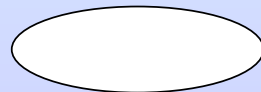
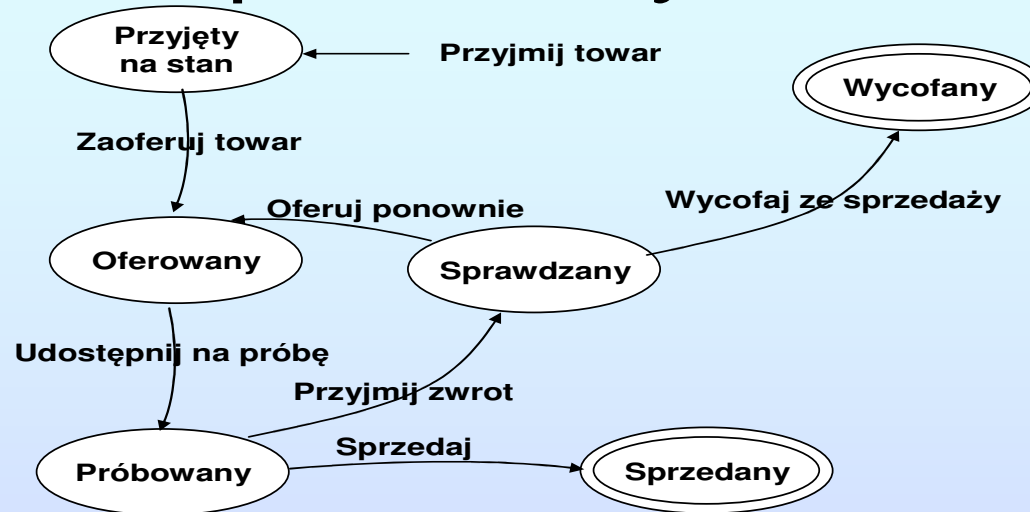
Modelowanie dynamicznych aspektów systemu

- Na diagramie przejść stanowych, klasa obiektów w systemie reprezentowana jako automat skończony, czyli mechanizm, który może się znajdować w jednej chwili w jednym ze skończonej liczby ustalonych stanów.
 - **Rys. Pewna notacja (Przykład „Towar”)**

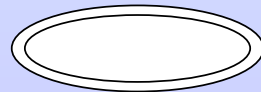
Techniki obiektowe

- Diagramy stanów rysowane dla pojedynczych klas, aby pokazać cały cykl życia pojedynczego obiektu
- Wiele postaci diagramów stanu, każda z nieco inną semantyką
- Postać z UML – oparta na mapach stanów Davida Harela.
- *(„Rzecz o istocie informatyki”)*
- **Rys. Notacja UML („Zamówienie”)**

Modelowanie dynamicznych aspektów systemu



Stan



Stan końcowy



Stan początkowy



Zdarzenie

Diagram stanu dla „Towar”

Modelowanie dynamicznych aspektów systemu

Syntaktyka etykiety przejścia – trzy części; każda opcjonalna:

- *zdarzenie [dozór]/ akcja*
- *dozór*= *warunek logiczny* (przejście gdy zwracana „prawda”, ze stanu można wybrać tylko jedno przejście, warunki dla zdarzeń–wykluczanie)
- *akcje* – związane z przejściami (procesy „szybkie”)
- *czynności* – związane ze stanami („dłuższe”), mogą być przerwane przez zdarzenie; etykieta: *do/ czynność*
- oba procesy implementowane przez metody klasy

Modelowanie dynamicznych aspektów systemu

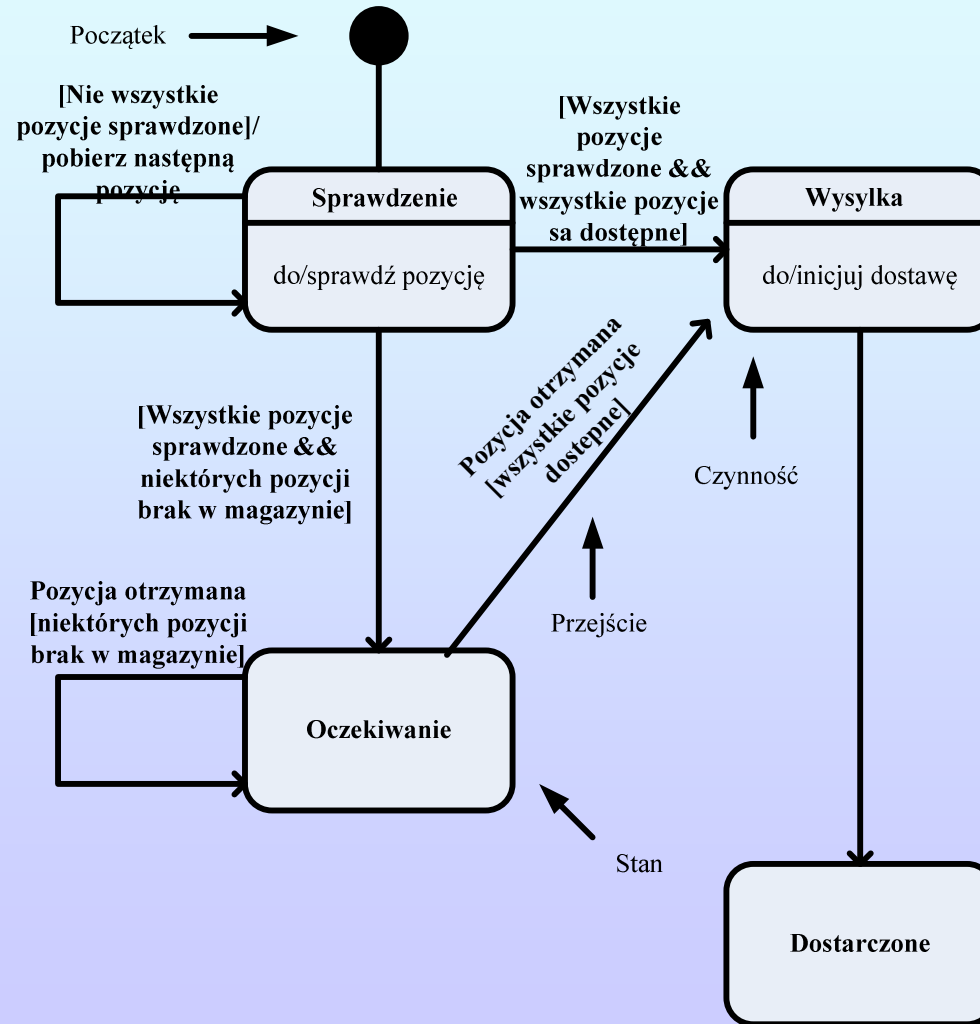


Diagram stanów

Modelowanie dynamicznych aspektów systemu

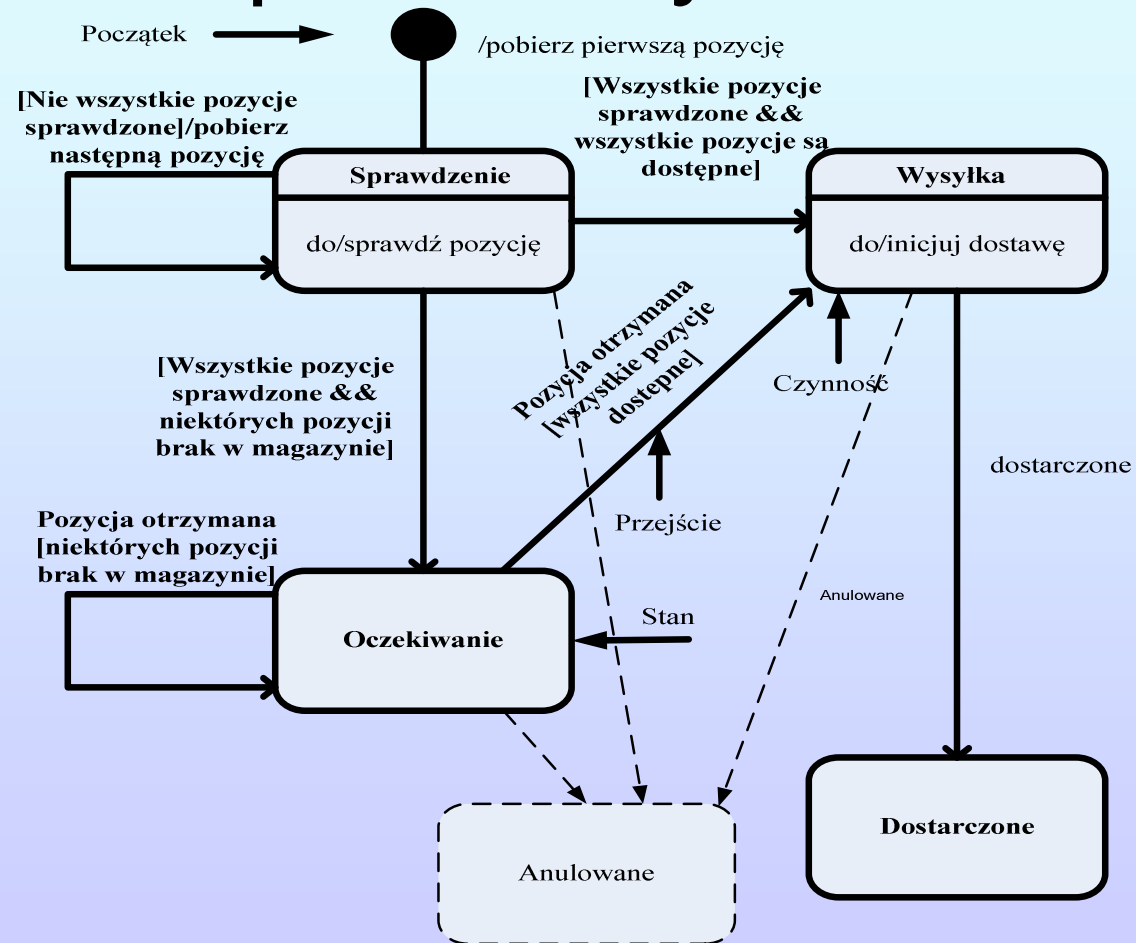


Diagram stanów dla „zamówienie”
(II wariant: stan złożony)

Modelowanie dynamicznych aspektów systemu

- Jeżeli z przejściem nie jest stowarzyszone zdarzenie => przejście zaraz po zakończeniu czynności związanych ze stanem
- Np. stan „Sprawdzanie”; *Jeśli... należy przejść do stanu...*(trzy warunki)
- „Sprawdzanie” pobranie pozycji i powrót
- „Oczekiwanie” bez czynności, czeka na zdarzenie „Pozycja otrzymana”
- „Wysyłka” – przejście po zdarzeniu do „Dostarczone”
- Anulowanie zamówienia przed dostawą: przejścia z trzech stanów: „Sprawdzanie”, „Oczekiwanie”, „Wysyłka”, lub stan złożony
- Zdarzenia: zewnętrzne (nazwane), wewnętrzne (czas, np. after ..., warunek when ...), specjalne (entry, exit, każde we/wy do/ze stanu)

Modelowanie dynamicznych aspektów systemu

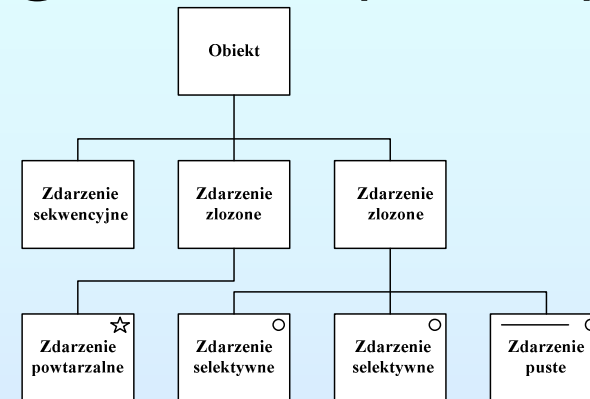
Uwagi:

- Przydają się do opisywania zachowania obiektu obejmującego kilka przypadków użycia systemu
- Nie nadają się do opisu zachowań obejmujących współdziałanie wielu obiektów
- Nie należy rysować dla każdej klasy systemu, tylko dla klas, które mają interesujące zachowanie
- Tworzyć je, gdy pomagają zrozumieć co się dzieje
- Popularny pogląd: zachowanie obiektów sterujących i obiektów interfejsu warto przedstawić na STD

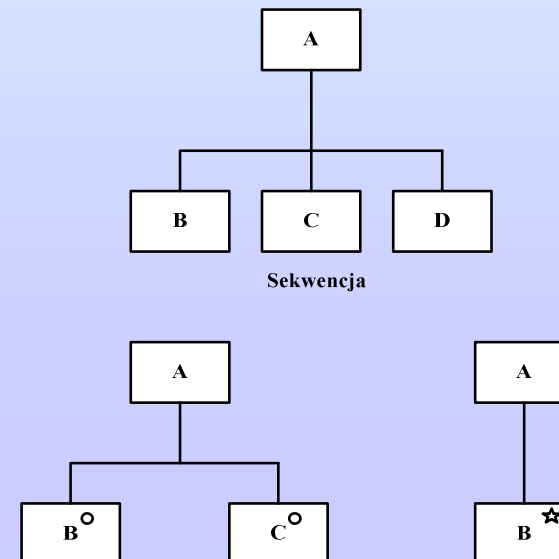
Entity Life History Diagrams (ELH)

Zasady tworzenia diagramów ELH:

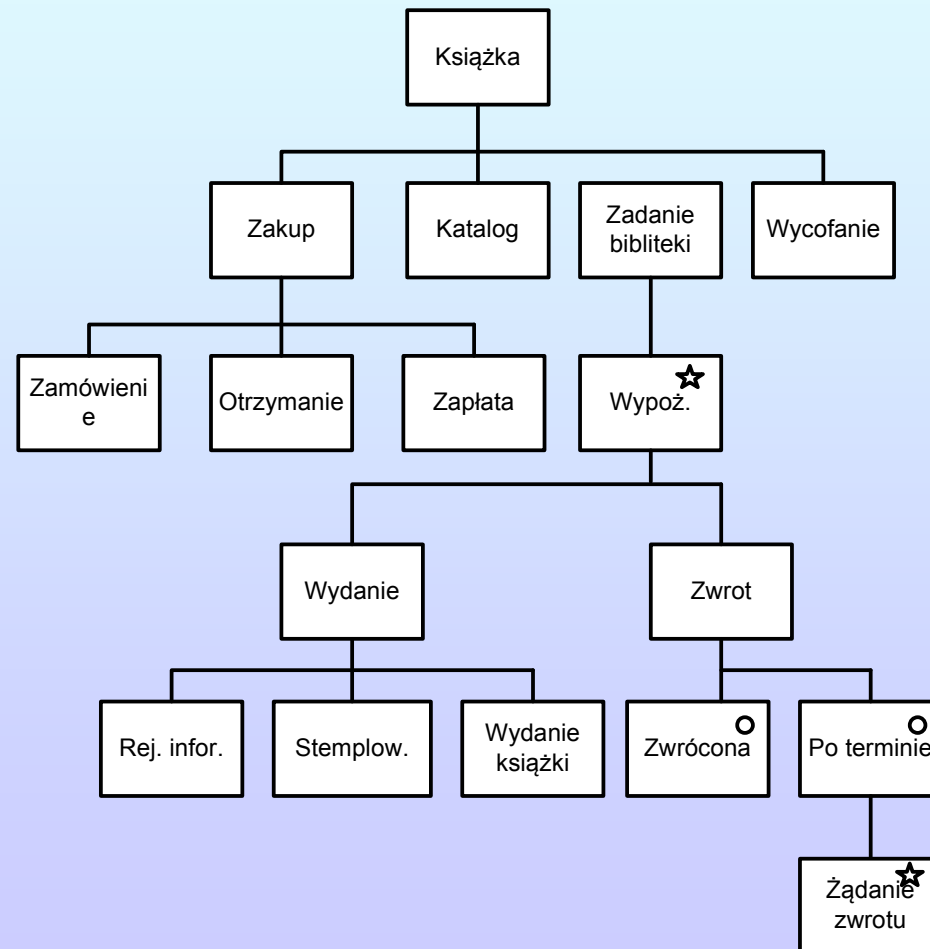
- tworzenie tablicy krzyżowej obiekt/zdarzenia przez:
 - wybranie obiektów z ERD,
 - identyfikację (na podstawie DFD) zdarzeń dotyczących danego obiektu,
 - dodatkowe rozważenie funkcji utrzymania w systemie - CRUD (I, M, D, R)
- rozważenie dla każdego obiektu z ERD:
 - normalnego cyklu życia,
 - zdarzeń specjalnych (wyjątkowych),
 - sytuacji błędnych.



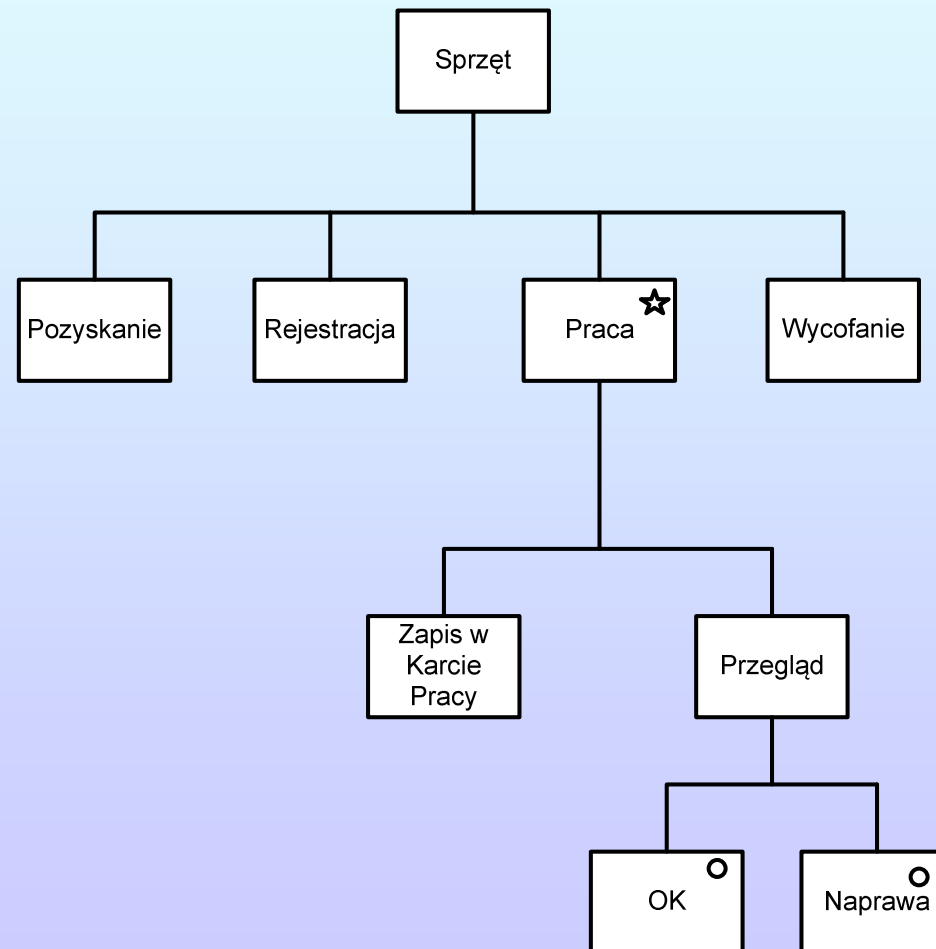
Komponenty i notacja diagramów ELH



Entity Life History Diagrams (ELH)



Entity Life History Diagrams (ELH)



Opis Dziedziny Problemu/ Obszaru Modelowania/Zakresu Odpow. Systemu

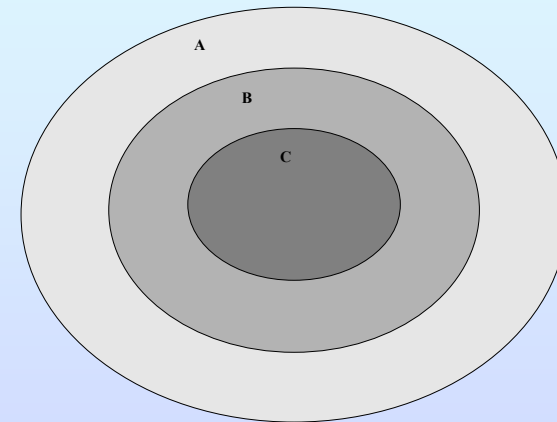
Dziedzina Problemu

por. Obszar

Modelowania

por. Zakres

Odpowiedzialności
Systemu



A Dziedzina problemu (Problem Domain)

B Obszar modelowania (System Model)

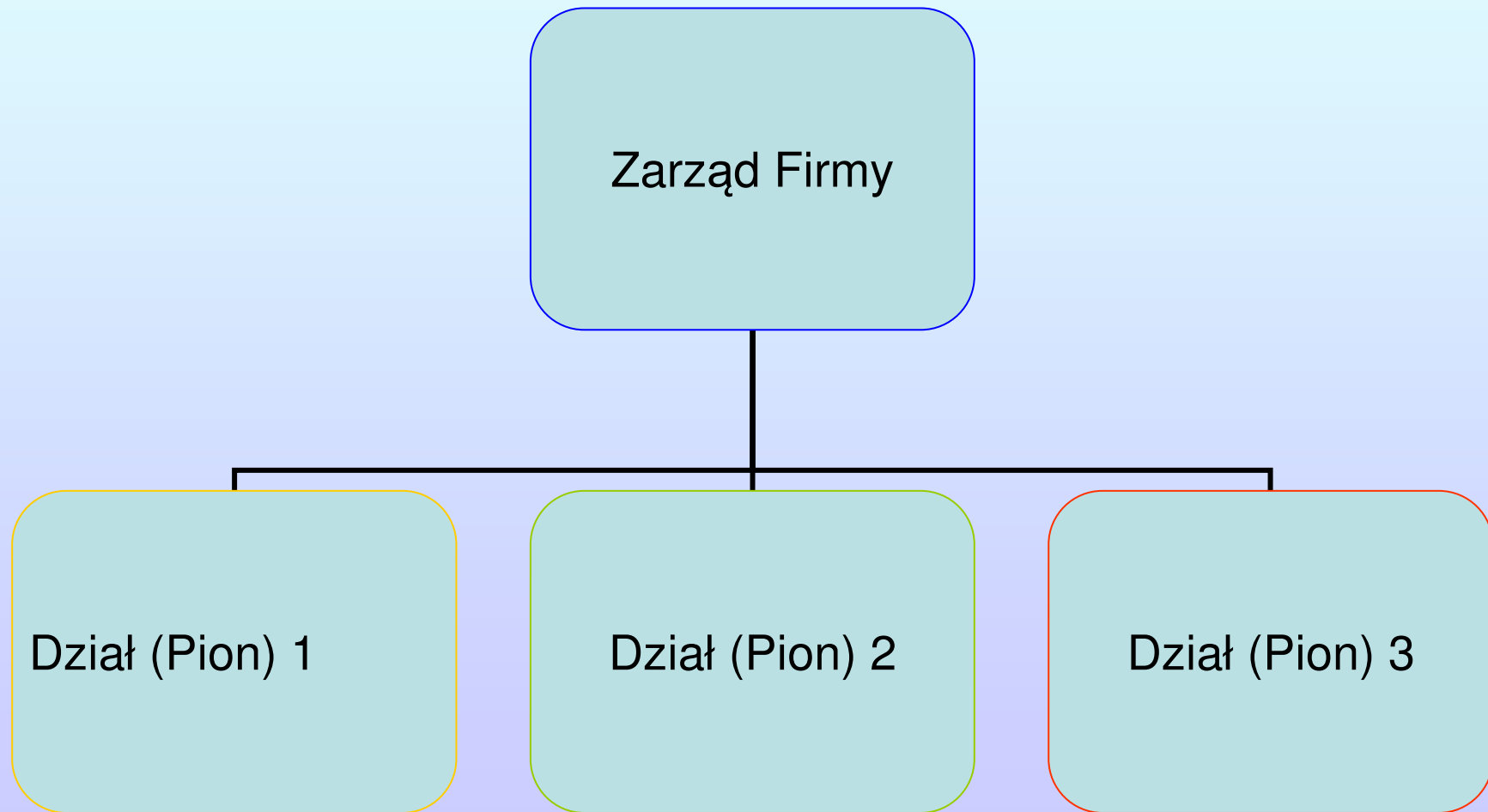
C Zakres odpowiedzialności systemu
(System Responsibilities)

Opis Dziedziny Problemu/ Obszaru Model., Zakresu Odpow.Syst. – uzupełnienie

Struktura organizacyjna

- Schemat/y
- Opis - rozwijany selektywnie w trakcie procesu analizy, nazywane stanowiska pracy, które biorą udział
- Związek struktury organizacyjnej z dziedziną, obszarem modelowania, zakresem odpowiedzialności systemu
- Przydatne do modelowania przedsiębiorstwa/organizacji (business modeling)
- Modelowanie działalności (inżynieria działalności)

Opis Dziedziny Problemu/ Obszaru Modelowania – struktura organizacyjna



Opis Dziedziny Problemu/ Obszaru Modelowania – *Przykład*

Przykład:

- FPH (Firma Produkcyjno-Handlowa)
- Cel (sformułowany wstępnie): System do wspomagania zarządzania FPH
- Dziedzina (całość) – bazuje na schemacie organizacyjnym, dołączyć ten schemat
- Składniki organizacyjne Firmy – Działy: Produkcji, Sprzedaży, Zaopatrzenia, Magazynowy, Księgowości
- Sporządzić opis działania tej organizacji

Opis Dziedziny Problemu/ Obszaru Modelowania – *Przykład*

Przykład, c.d. (FPH)

Sporządzić opis...

- wymaga wywiadu z właścicielami lub kierownictwem organizacji (lub osobami wskazanymi) w celu ustalenia (1) przeznaczenia tworzonego systemu,
- (2) chcemy uzyskać podstawowe zrozumienie sposobu funkcjonowania organizacji (do wykorzystania na dalszych etapach procesu projektowania).

Opis Dziedziny Problemu/ Obszaru Modelowania – *Przykład*

Przykład, c.d. (FPH: produkcja, sprzedaż, zamówienia, magazyn...)

Formułowanie **definicji celu:**

- określa czemu ma służyć projektowany system,
- powinna być krótka, zwięzła i ogólna
- nie opisywać szczegółowych zadań

Np.:

Celem systemu jest obsługa wykorzystywanych danych..., oraz udostępnianie informacji niezbędnych dla codziennego funkcjonowania firmy

Opis Dziedziny Problemu/ Obszaru Modelowania – *Przykład*

Przykład, c.d. (FPH)

- zrozumienie sposobu funkcjonowania organizacji (do wykorzystania na dalszych etapach procesu projektowania) – pytania
- Np.:
 - Jak opisaliby Państwo nowemu klientowi działalność Waszej organizacji ? (produkcja, sprzedaż, zamówienia...)
 - Co jest celem organizacji ?
 - Jaka jest podstawowa funkcja ?
 - Jaki jest powód istnienia organizacji ?
 - Na czym koncentruje się działalność organizacji ?
 - ...

Opis Dziedziny Problemu/ Obszaru Modelowania – *Przykład*

Przykład, c.d. (FPH) Składniki organizacyjne Firmy

Działy: Produkcji, Sprzedaży, Zaopatrzenia, Magazynowy, Księgowości.

- **Obszar modelowania:**

Ta część FPH, która jest bezpośrednio związana z dokumentami sprzedaży i ruchem towarów:

- **Składniki organizacyjne dla Obszaru Modelowania:**

- Sprzedaży (Dział Obsługi klienta)
- Zaopatrzenia (Dział Zamówień),
- Magazynowy (magazyn, magazynier),
- Księgowości (dla OM, ale **usunięte** dalej dla **ZOS**)

Opis Dziedziny Problemu/ Obszaru Modelowania – obszary aktywności

- Obszary aktywności OA (podsystemy)
 - Grupy czynności, które są ze sobą powiązane poprzez charakter działań, obiekty, na które oddziałują, osoby (grupy osób, komórki org.), które je przeprowadzają, itp.
 - Podział wszystkich czynności na obszary aktywności zależy od precyzji kryteriów podziału
 - Otrzymujemy ziarnistość odpowiednio dużą lub małą, w zależności od potrzeb i ścisłe lub słabsze powiązanie czynności w ramach jednego obszaru aktywności

Opis Dziedziny Problemu/ Obszaru Modelowania: OA i PB - *Przykład*

Obszary Aktywności:

- 1. Obsługa klienta,
- 2. Zamawianie towarów,
- 3. Kontrola stanów magazynowych,
- 4. Obsługa magazynu

Procedury biznesowe w OA (opis w logice biznesowej)

- Przeprowadza się dla każdego obszaru aktywności, który bierzemy pod uwagę w modelowaniu...

Opis Dziedziny Problemu/ Obszaru Modelowania – procedury biznesowe

- Procedury biznesowe w OA (opis w logice biznesowej)
 - Przeprowadza się dla każdego obszaru aktywności, który bierzemy pod uwagę w modelowaniu
 - Jeżeli opisujemy procedurę biznesową jednego rodzaju, to musimy sporządzić opis dla wszystkich z danego obszaru (uzasadnienie – d. prawdopodobieństwo gdy opisujemy jedną, to w trakcie procesu modelowania i tak okazuje się, że istnieją inne z tego obszaru aktywności)
 - Podstawą do decyzji o opisaniu procedury jest występowanie w niej akcji, która ma być realizowana/wspomagana przez SI !
 - Por.: Proc. Biz., czynności, akcje...

Opis Dziedziny Problemu/ Obszaru Modelowania – proc. biz. czynności

- Czynności w ramach procedur biznesowych (opis)
 - Opisuje się kolejne akty decyzyjne i sprawcze, które się składają na realizację procedury biznesowej (cały ciąg działań, czynności)
 - Powiązanie z dokumentami i „rejestrami”
 - Terminologia „biznesowa” – Słownik Pojęć
 - Przykład: akty prawne (administracja państwowa, np. Gmina – wydanie pozwolenia na budowę) - to jest opis procedur biznesowych

Opis Dziedziny Problemu/ Obszaru Modelowania proc. biz. czynności, c.d.

- Pojedyncze akty (działania) opisuje się wg schematu:
 - kto (stanowisko pracy a NIE osoby, które to robią)
 - co robią
 - po co
 - w jakich okolicznościach
 - z pomocą czego (ręcznie/wspomaganie/system)
- Algorytm (wszystkie sytuacje),
- Scenariusz (opis dla sytuacji) w ramach procedur biznesowych

Opis Dziedziny Problemu/ Obszaru Modelowania proc. biz. czynności, c.d.

JAK ? Wywiad (User Stories)

Np. pytania:

- Jak Pan(i) opisał(a)by swą pracę?
- Jaką funkcję pełni Pan(i) w organizacji?
- Z jakimi danymi ma Pan(i) do czynienia?
- Jakich raportów Pan(i) używa?
- Jakich informacji potrzebuje Pan(i) w swej pracy?
-
- => **chcemy** przechowywać, generować, wyświetlać,

Opis Dziedziny Problemu/ Obszaru Modelowania: OA i PB - *Przykład*

Przykład:

Obszary Aktywności:

- **1. Obsługa klienta,**

Procedury biznesowe w OA1:

1.1 Prowadzenie bazy klientów (*OPIS !*)

1.1.1 Rejestrowanie klienta (*Obejmuje.....*)

1.1.2 Korygowanie danych klienta (*Polega na....*)

1.2 Przyjmowanie zamówień od klienta (rezerwacja towaru)

1.2.1 Sprawdzenie aktualnych stanów magazynowych

1.2.2 Rezerwacja towaru

1.2.3 Zmiana lub likwidacja rezerwacji towaru

1.3 Generowanie dokumentów sprzedaży (1.3.1 Wystawianie faktury; 1.3.2 Zatwierdzanie faktury; 1.3.3 Generowanie dokumentów KP i WZ)



Koniec

Podsumowanie



How the customer explained it



How the Project Leader understood it



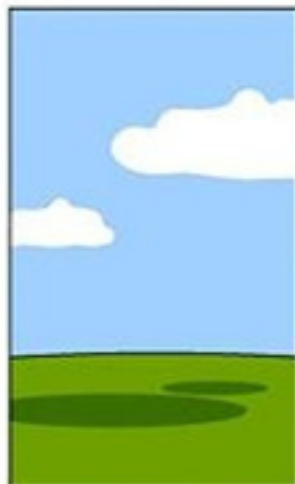
How the Analyst designed it



How the Programmer wrote it



How the Business Consultant described it



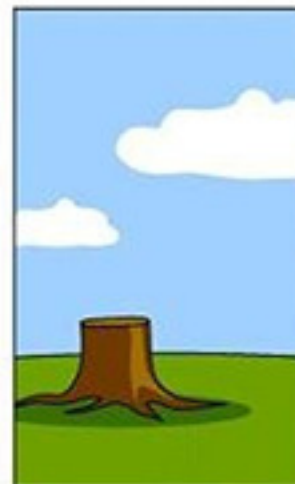
How the project was documented



What operations installed



How the customer was billed



How it was supported



What the customer really needed