

Strukturalne metodyki projektowania systemów informatycznych

Kalendarium

1976 — Chen P. (Entity Relationship Model – ERD)

1978 — DeMarco T.

1979 — Yourdon E., Constantine L.

1983 — Jackson M. 1989 — Yourdon (synteza rozwiązań)

1991 — Rumbaugh J. (Object Modeling Technique)

1992 — Jacobson J.

1994 — Booch G.

1997 — UML został uznany przez OMG (Object Management Group) za standard notacji dla obiektowej metodyki projektowania.

Metodyki strukturalne

Trzy podstawowe rodzaje modeli opisujących system:

- Model danych – opisuje obiekty występujące w systemie i relacje między nimi.
- Model dynamiki - opisuje oddziaływanie pomiędzy obiektami.
- Model funkcjonalny - opisujący transformacje danych w systemie.

Model danych – opisuje statyczną strukturę systemu, grupując dane w kolekcje zwane obiektami (encje). Graficznym odpowiednikiem jest diagram ERD (ang. *Entity Relationship Diagram*), którego węzły reprezentują obiekty natomiast łuki odzwierciedlają relacje pomiędzy obiektami.

Model dynamiki – opisuje zmieniające się w czasie aspekty systemu, używany jest do specyfikacji sterowania w systemie. Graficznym odpowiednikiem jest diagram stanów, którego węzły są stanami, a łuki są przejściami pomiędzy stanami wywoływanymi przez zdarzenia

Model funkcjonalny – opisuje transformacje danych wewnątrz systemu. Odpowiednikiem graficznym jest graf, którego węzły są zwykle procesami, a łuki – przepływami danych. Graf ten nazywany jest diagramem przepływu danych – DFD (ang. *Data Flow Diagram*).

Modelowanie procesów – diagramy przepływu danych DFD

DFD (ang. *Data Flow Diagram*) – diagramy przepływu danych pozwalają na modelowanie procesów w systemie informatycznym lub organizacji. Podstawowe elementy diagramów DFD to:

- **proces** (ang. *process*),
- **przepływ** (ang. *flow*),
- **magazyn** inaczej **skład/składnica danych** (ang. *datastore*),
- **terminator** (ang. *terminator*) inaczej **jednostka zewnętrzna** (ang. *external entity*).

Każdy z powyższych elementów ma odpowiedni symbol graficzny jednoznacznie wyróżniający go na diagramie.

Niestety, różne metodyki używają różnej symboliki – zwykle jednak koncepcja i semantyka diagramów jest jednakowa.

Procesy

Proces reprezentuje często *system informatyczny*, jego część lub wybrany *podprogram*. Czasem jednak reprezentuje czynność wykonywaną bez wsparcia komputera.

Proces zamiennie nazywany bywa *funkcją* lub *transformacją* — dokonuje bowiem pewnych operacji na danych, *transformując* otrzymane dane wejściowe i przekazując je na wyjścia. Dane wejściowe i wyjściowe procesu to *przepływy danych*.

Każdy proces ma swoją *nazwę*, jest to pojedyncze słowo, fraza lub proste zdanie. W języku polskim jest to fraza czasownikowa lub zdanie taką frazę zawierające.

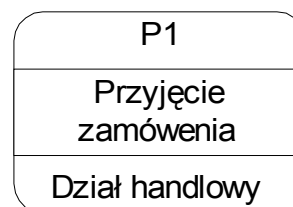
Każdemu procesowi przydziela się zwykle *unikatowy numer* i najczęściej to on właśnie jednoznacznie identyfikuje dany proces.

Notacja

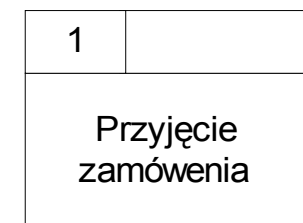
Metodyka Yourdona



Metodyka Gene-Sarsona



Metodyka SSADM



Przepływy

Przepływy służą do modelowania *wymiany danych między procesami*.

Przepływy reprezentują *dane w ruchu* – jest to porcja danych albo pakiet informacji przekazywany z jednej części systemu do drugiej.

Notacja

Metodyka Yourdona



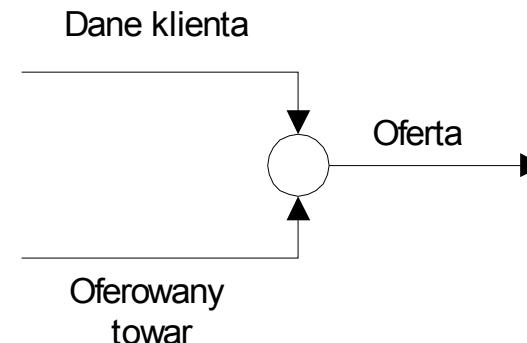
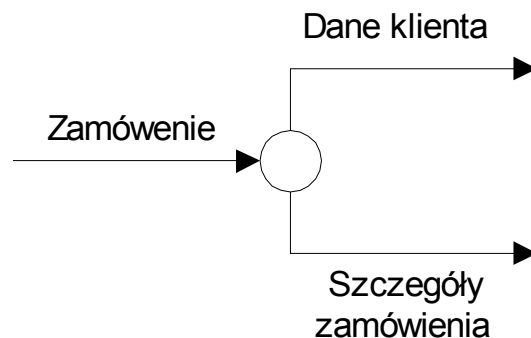
Metodyka Gene-Sarsona



Metodyka SSADM



Rozdzielanie/łączenie przepływów



Składnica danych (magazyny)

Składnice danych służą do modelowania trwałych danych — danych przechowywanych na nośnikach pamięci zewnętrznej, zwykle plikach lub bazach danych systemu informatycznego.

Składnice danych służą zwykle do reprezentowania istniejących lub przewidywanych baz danych systemu.

Jeżeli do składnicy wchodzi jeden złożony przepływ danych to jego budowa jest zwykle tożsama ze strukturą informacji przechowywanych bazie danych.

Jeżeli do składnicy wchodzi kilka przepływów, to złożenie tych przepływów reprezentują zwykle strukturę bazy danych.

Notacja

Metodyka Yourdona



Metodyka Gene-Sarsona



Metodyka SSADM



Obiekty zewnętrzne (terminatory)

Obiekty zewnętrzne reprezentują *źródła* i/lub miejsca *przeznaczenia danych*, które są zewnętrzne w stosunku do systemu.

Obiekty zewnętrzne dostarczają informacji (danych), które są potem przetwarzane w systemie oraz/lub odbierają informacje (dane) wyprodukowane przez system.

Uwaga — obiekty zewnętrzne są *poza systemem* i najczęściej analitycy i projektanci nie mają wpływu na ich funkcjonowanie.

Notacja

Metodyka Yourdona



Metodyka Gene-Sarsona



Metodyka SSADM



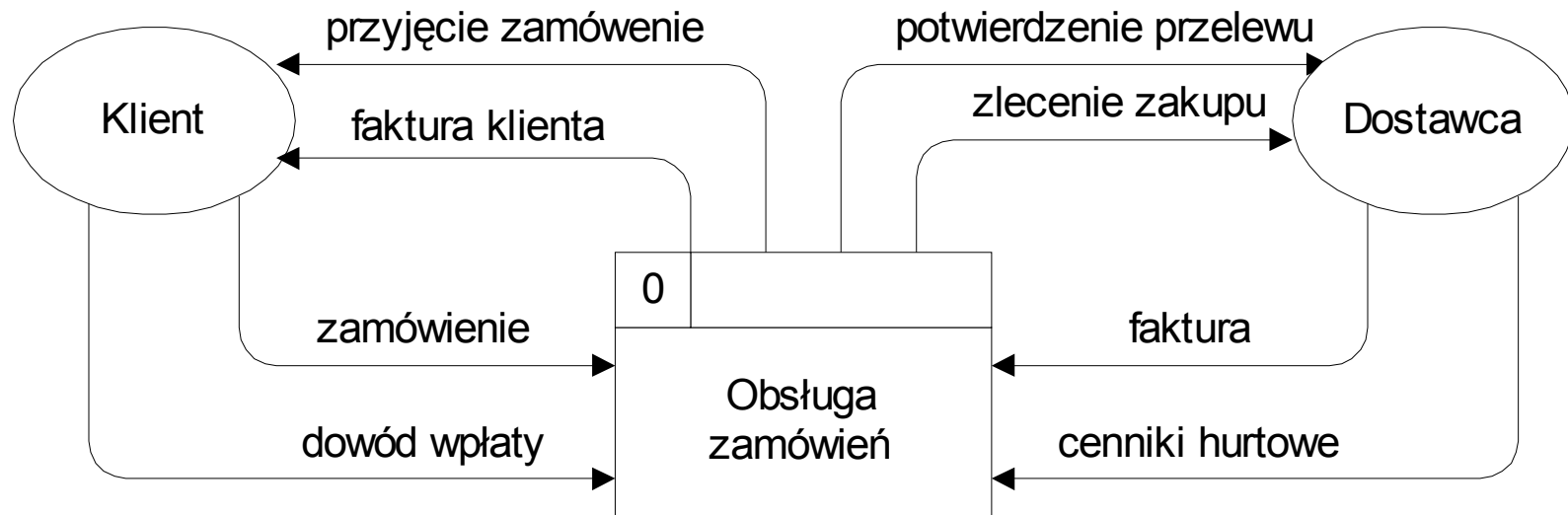
Diagramy przepływu danych DFD w akcji

Diagram kontekstowy

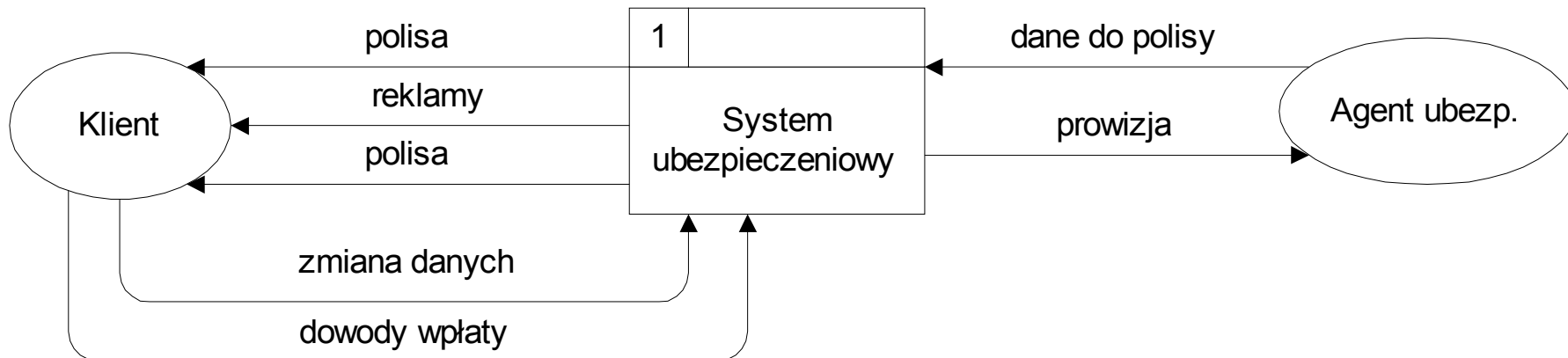
Diagram kontekstowy obrazuje modelowany system oraz obiekty zewnętrzne, które są z nim interakcji.

Diagram kontekstowy zwany jest też *diagramem poziomym* o lub *diagramem ogólnym systemu*.

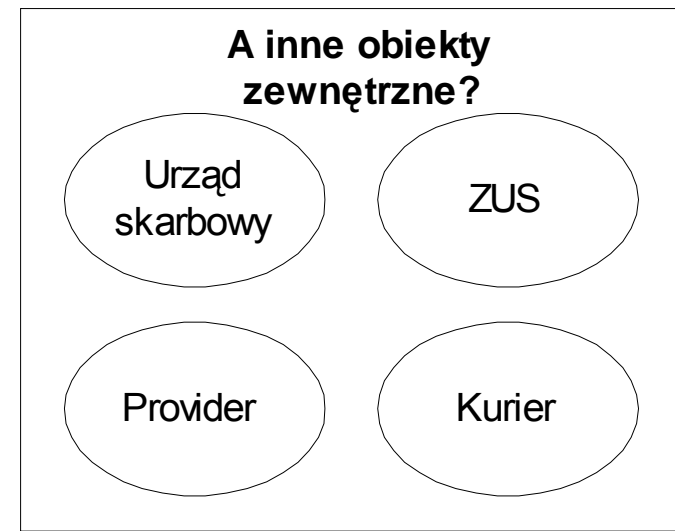
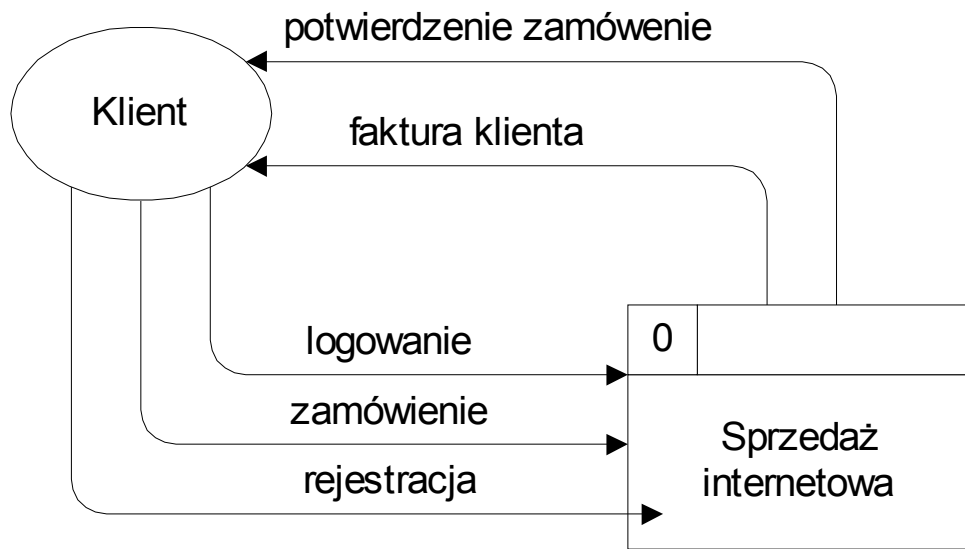
Przykład diagramu kontekstowego – przykład 1



Przykład diagramu kontekstowego – przykład 2

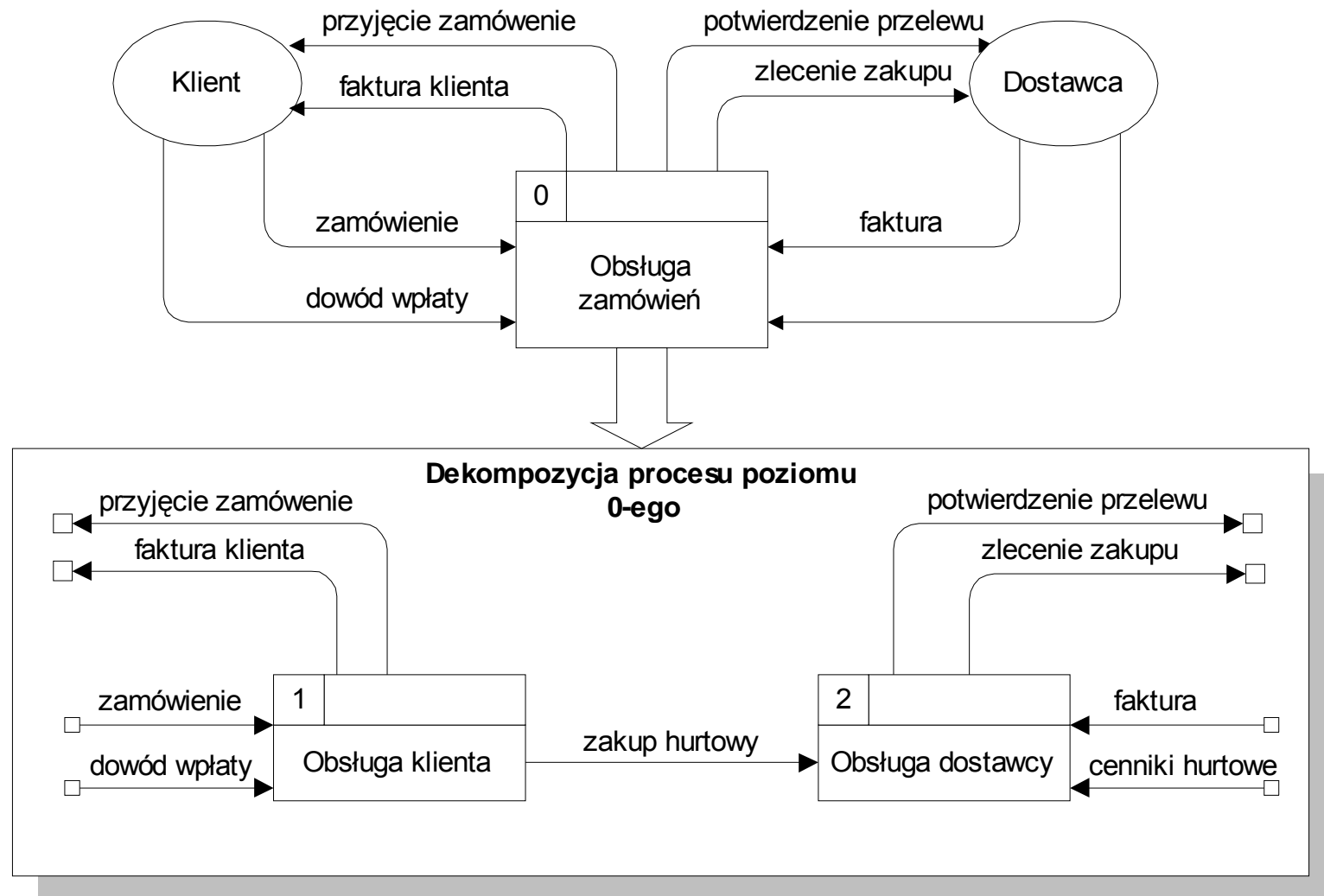


Przykład diagramu kontekstowego – przykład 3

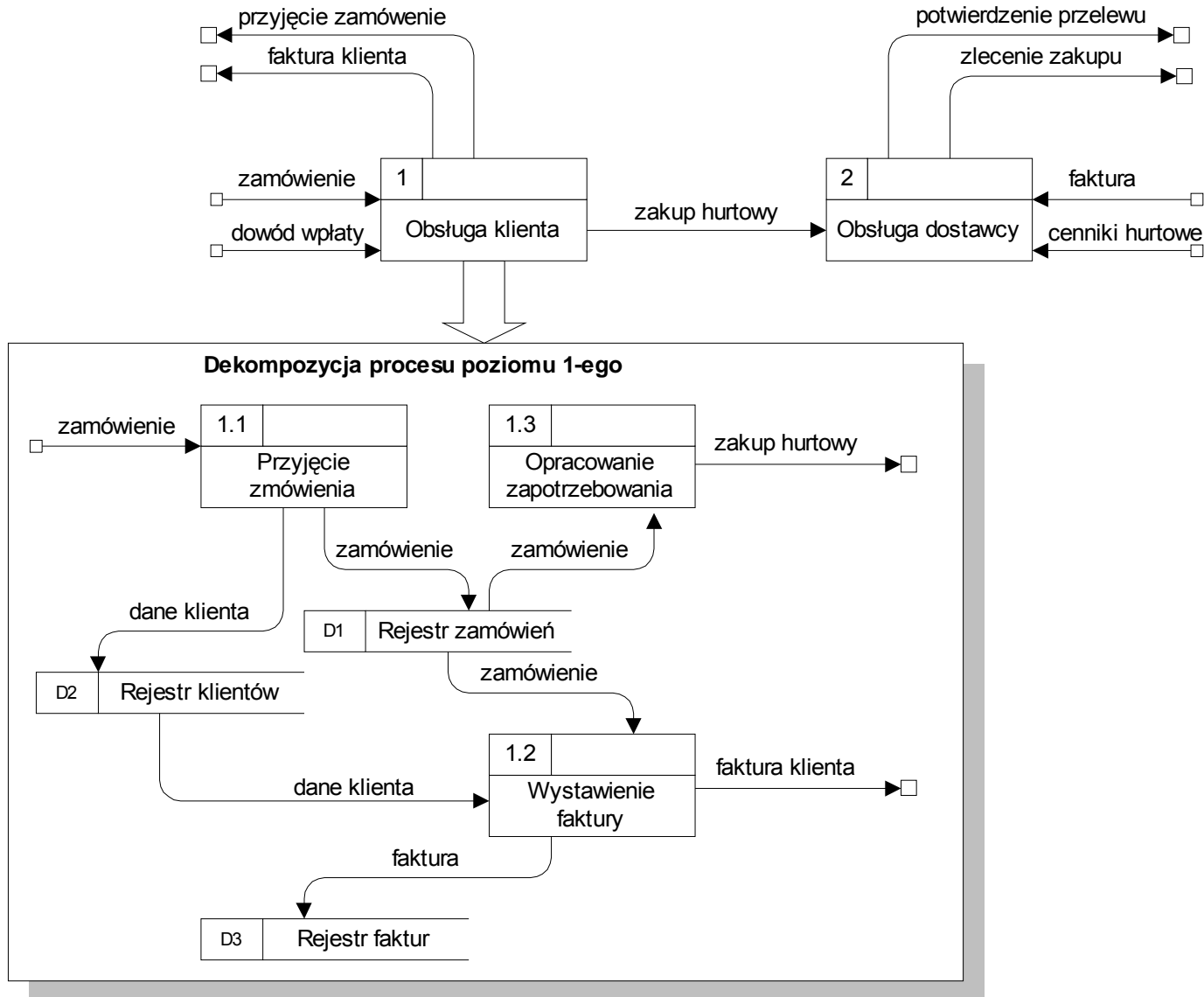


Diagramy przepływu podlegają hierarchicznej dekompozycji

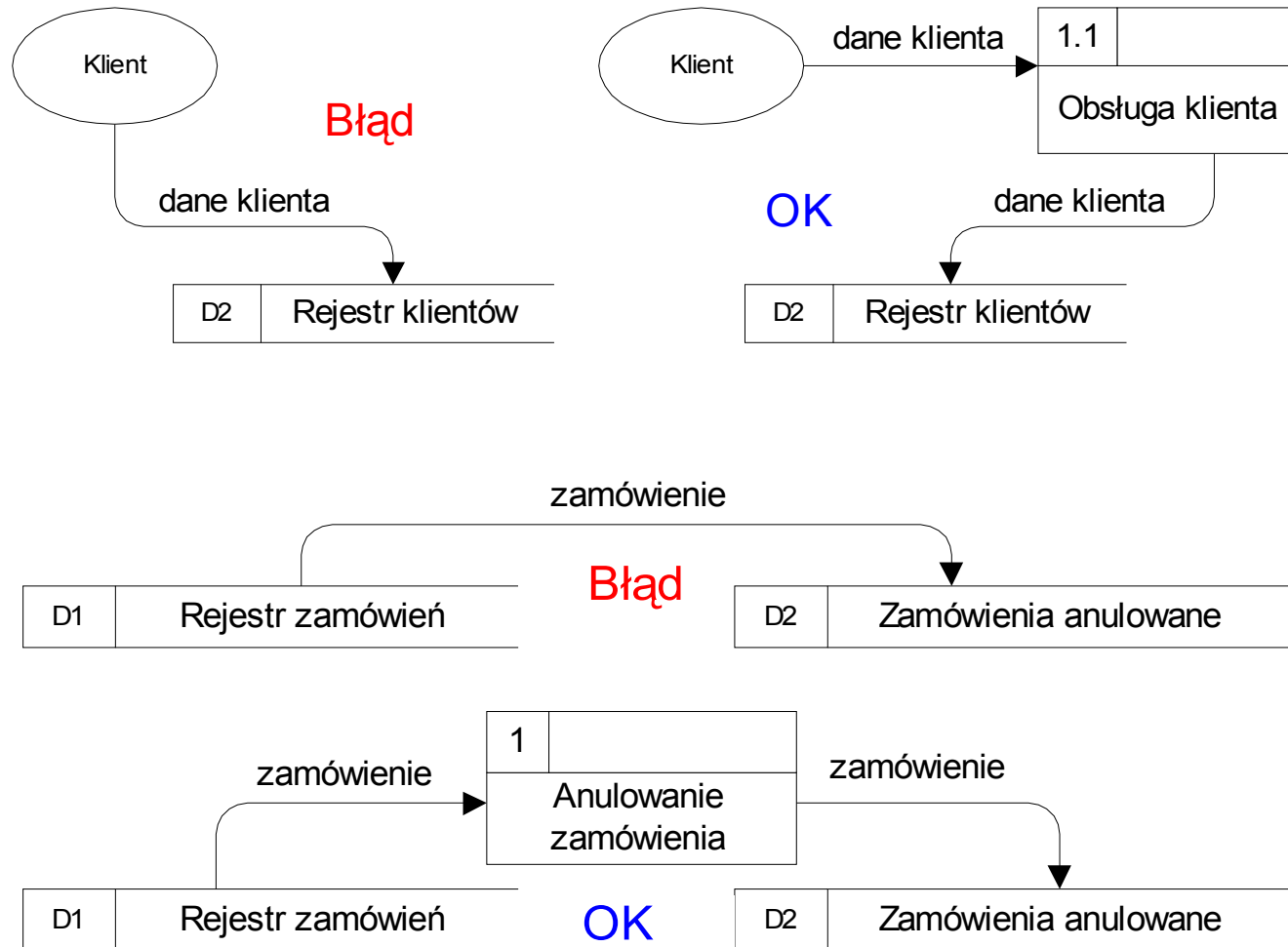
Dekompozycja diagramu kontekstowego



Dekompozycja diagramu poziomu pierwszego : proces 1

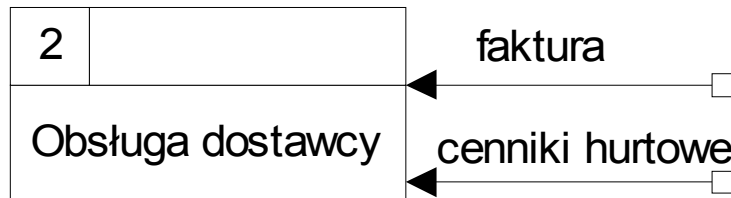


Najczęściej popełniane błędy

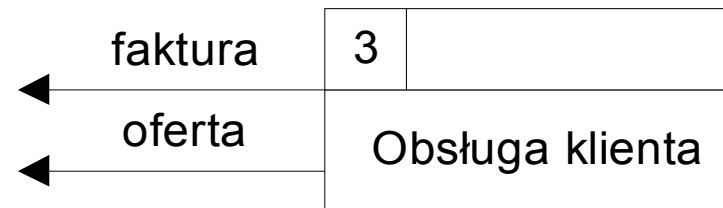


Najczęściej popełniane błędy – cd

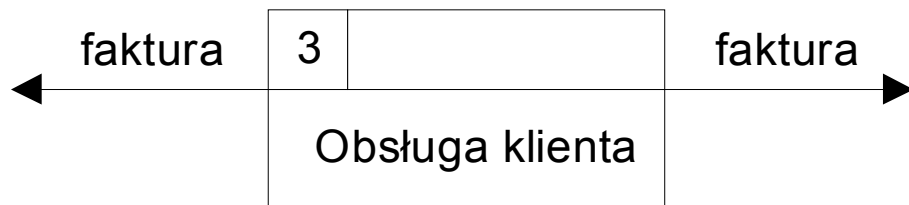
Tylko wejścia? Czarna dziura!



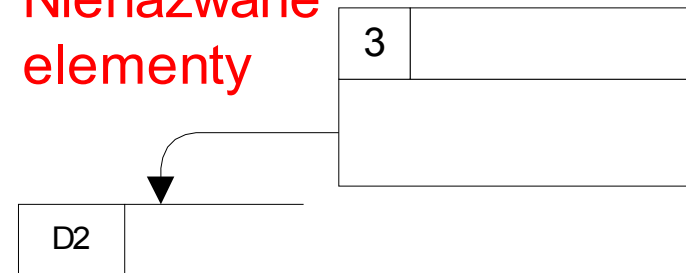
Tylko wyjścia? Czarodziej?!



Powielanie informacji



Nienazwane elementy



Podsumowanie modelowania procesów wykorzystaniem DFD

Diagram kontekstowy zawiera istotne dla systemu charakterystyki, określa:

- Osoby, organizacje lub inne systemy, z którymi komunikuje się nasz system — są to obiekty zewnętrzne.
- Dane, które napływają od otoczenia systemu reprezentowanego przez jednostki zewnętrzne. Te dane podlegają przetwarzaniu przez system.
- Dane, które są wytwarzane przez system i są przekazywane do jego otoczenia.

Przepływy niosą w sobie dane, których budowa stanowi podstawę do określenia schematu baz danych.

Składnice reprezentują trwałe magazyny danych, reprezentowane w systemie zwykle w postaci bazy danych.

Diagramy DFD podlegają hierarchicznej dekompozycji na diagramy potomne, zawierające podprocesy specyfikujące określone działania systemu.