

7. Projektowanie systemów bazodanowych

7.1. Etapy tworzenia oprogramowania bazodanowego

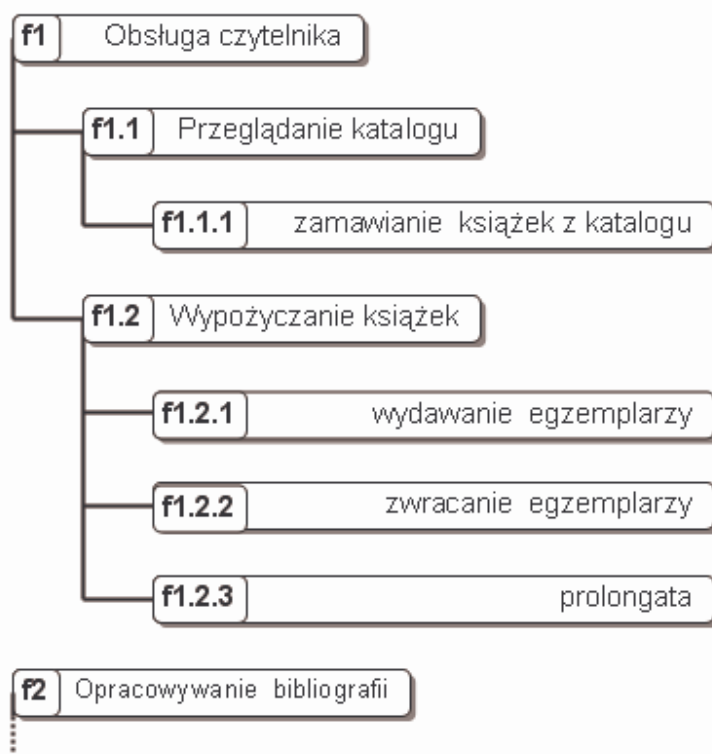
- określenie wymagań
- projektowanie
- implementacja
- testowanie
- konserwacja

7.2. Etapy projektu systemu bazodanowego

- określić środowisko systemu (diagram kontekstowy, UML)
- określić cel budowy systemu,
- określić wymagania stawiane przed systemem (diagramy przypadków użycia),
- określić funkcjonalność systemu – hierarchia (FHD),
- określić strukturę danych i zależności między nimi (ERD, diagram obiektowy),
- opisać przepływy danych, procesy i stany występujące w systemie (DFD, STD)

7.3. Diagramy Hierarchii Funkcji (FHD - Function Hierarchy Diagram)

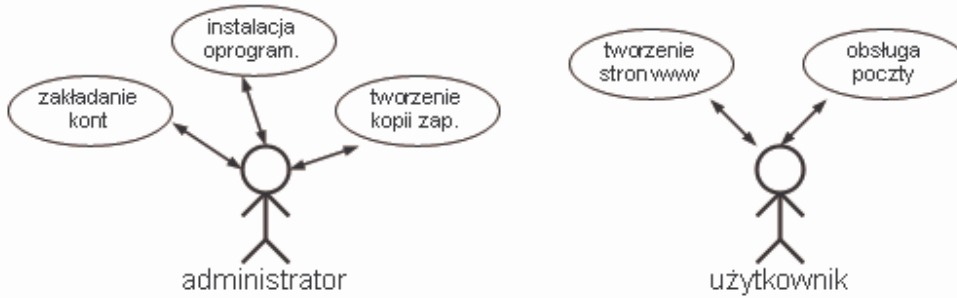
Diagram hierarchii funkcji określa czynności i operacje wykonywane przez system.



Opis funkcji:

nr / nazwa funkcji	1.1.1 / zamawianie książek z katalogu
opis, cel	realizacja zamawiania książek w wypożyczalni
dane wejściowe	nr biblioteczny, hasło, tytuł, autor, rocznik
dane wyjściowe	sygnatura biblioteczna lub 0, jeśli brak egzemplarzy
warunek wykonania	udane logowanie do systemu

7.4. Diagramy przypadków użycia



7.5. Diagramy związków encji ERD

Notacja [1]

Zbiory encji – prostokąty
 Atrybuty – owale
 Związki - romby

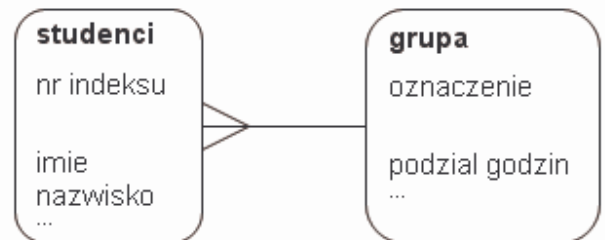
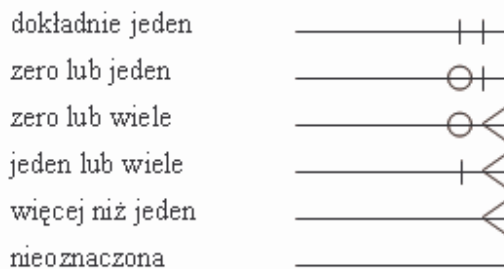


Notacja [17]

Zbiór encji



Związki encji – krotność



7.6 Diagramy przepływów danych (DFD – Data Flow Diagrams)

DFD służy do obrazowania przepływu i przetwarzania danych w systemie.

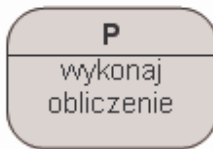
Podstawowe elementy DFD:



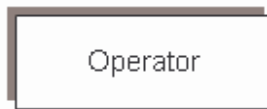
- Magazyn danych (DS – Data Store)



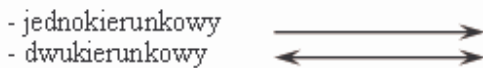
- Proces (operacja wykonywana przez system)



- System zewnętrzny



- Przepływ danych



Możliwe przepływy danych:

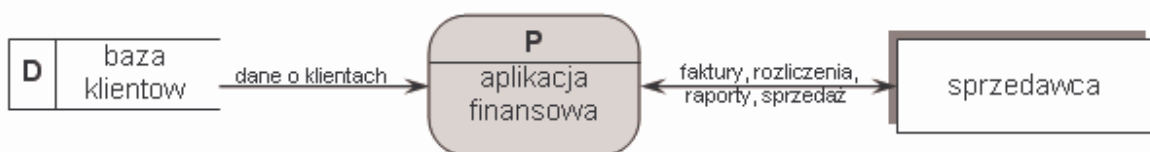
- proces – proces - wymiana danych między procesami,
- proces – zbiornik – odczyt/zapis/modyfikacja danych w magazynie danych,
- proces – system zewnętrzny – wymiana danych pomiędzy systemem a systemem zewnętrznym,

Przepływ między magazynami oraz między magazynem a systemem zewnętrznym – niedopuszczalny.

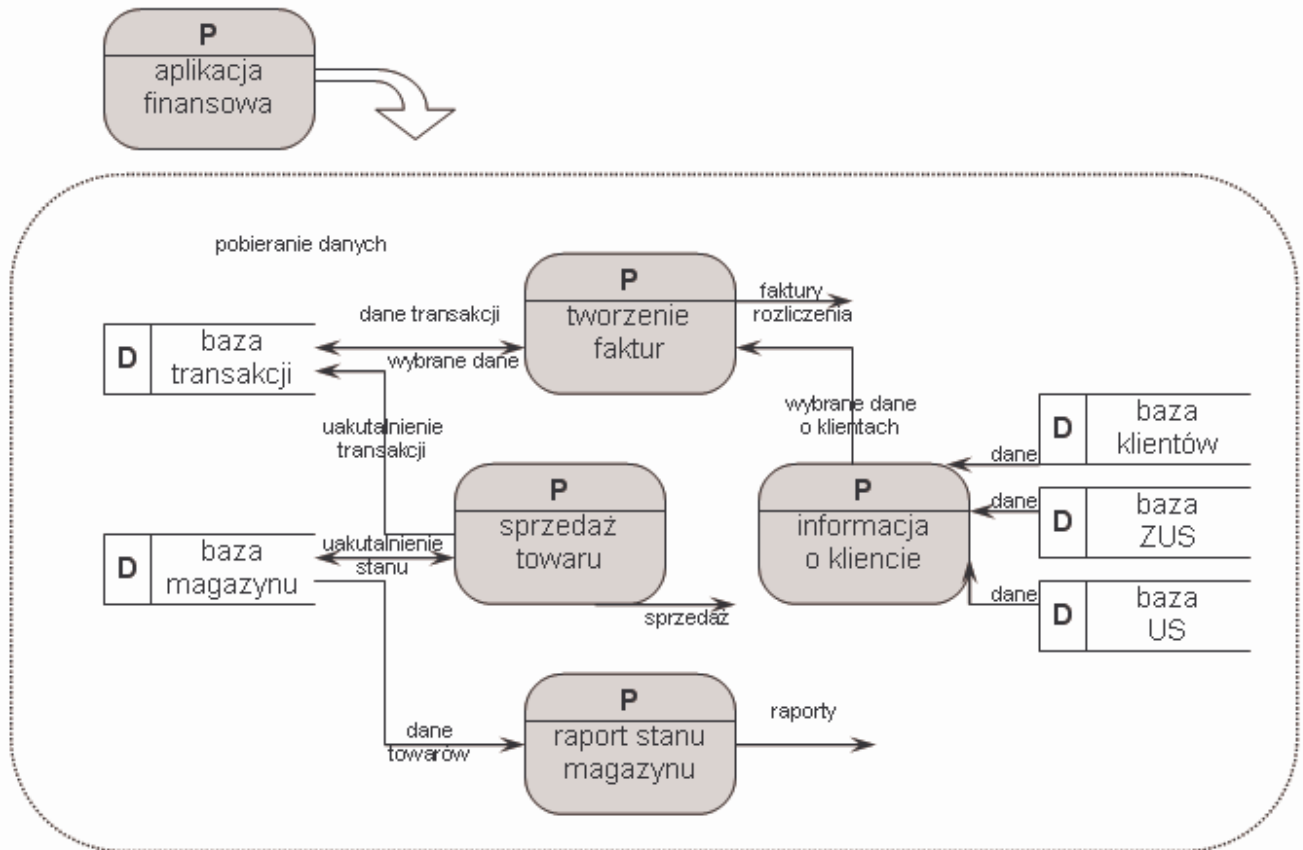
Proces ‘duch’ – pobiera dane z nieokreślonego źródła

Proces ‘studnia’ – kumuluje dane nie dając wyniku

Przykładowy diagram przepływu danych:



Celem zachowania czytelności i przejrzystości diagramy przepływu danych należy organizować hierarchicznie.



Hierarchie diagramów organizuje się następująco:

- diagram kontekstowy – przedstawia powiązania systemu ze środowiskiem (system jako pojedynczy proces), służy do określenia zakresu i granic systemu,
- diagram systemowy – opisuje główne funkcje systemu,
- diagramy niższych poziomów (szczegółowe) – stanowią rozwinięcie diagramów systemowych – aż do poziomu najniższego.

Diagramy kontekstowe i systemowe – projektowanie i dokumentacja.

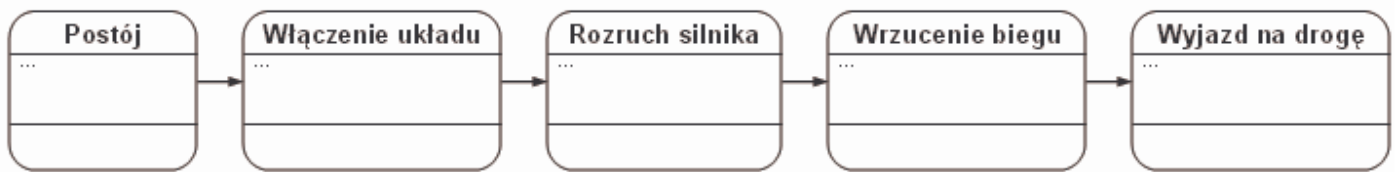
Diagramy niższych poziomów – projektowanie i implementacja.

7.7. Diagramy stanów systemu (STD – State Transitions Diagrams)

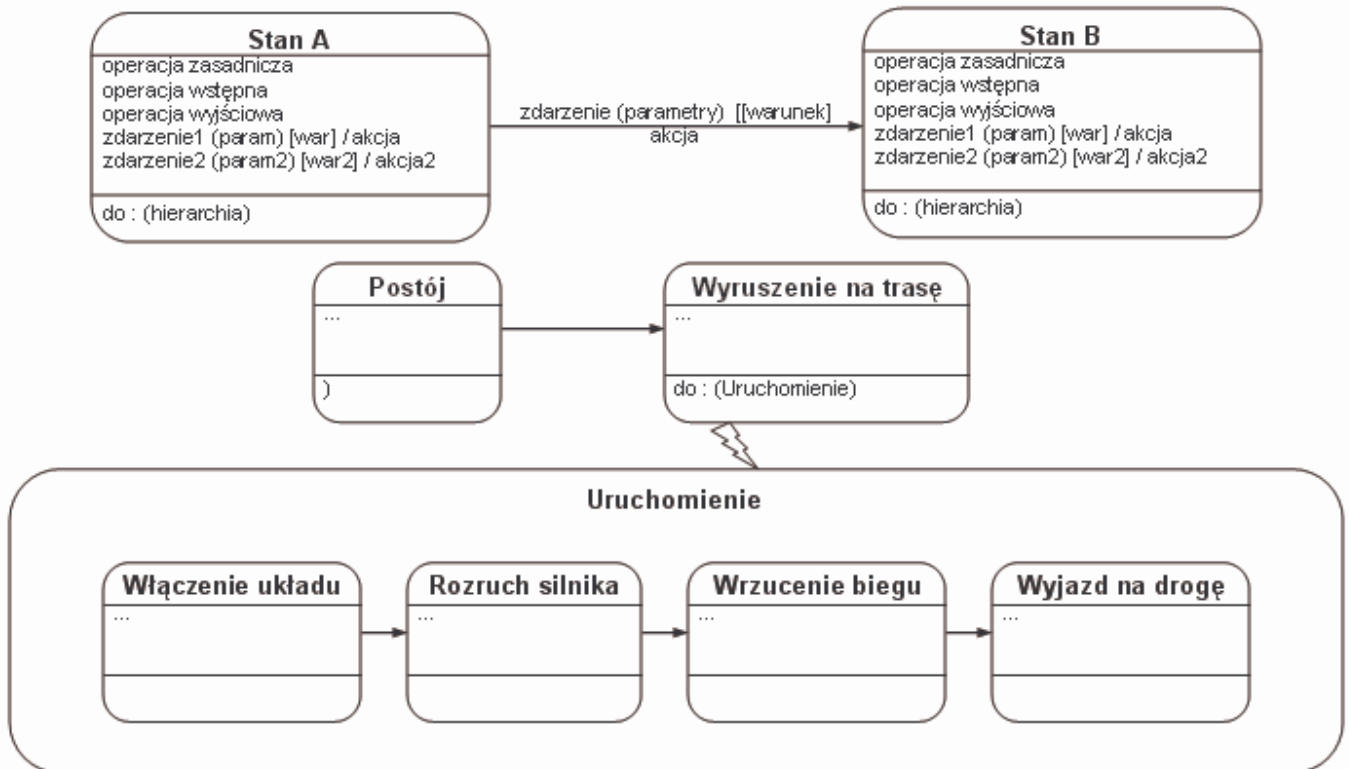
Diagramy stanów systemu służą do określania dynamiki systemu.

Podstawowe elementy STD

- zdarzenie – zjawisko związane z pewnym punktem czasowym, zdarzenie może być wewnętrzne lub zewnętrzne,
- stan – sytuacja, w jakiej znajduje się system w okresie czasu pomiędzy dwoma zdarzeniami,
- przejście – zmiana stanu systemu, zachodząca pod wpływem zdarzenia natychmiastowo lub po spełnieniu wymaganego warunku,
- operacja – czynność wykonywana przez system, trwająca pewien okres czasu, może zostać przerwana przez zdarzenie,
- akcja – wykonana operacja w związku z wystąpieniem zdarzenia.



Diagramy STD przewidują tworzenie hierarchii (generalizacja, specjalizacja)



7.8. Pakiety do wspomaganiania projektowania baz danych (CASE)

- Oracle:
 - Developer / 2000,
 - Designer / 2000,
- IBM
 - Rational Software Architect, Rational Data Architect
- Microsoft:
 - Visio,
- inne
 - DBdesigner

7.9. Przykłady

Diagram DFD (Gane-Sarson), MS Visio

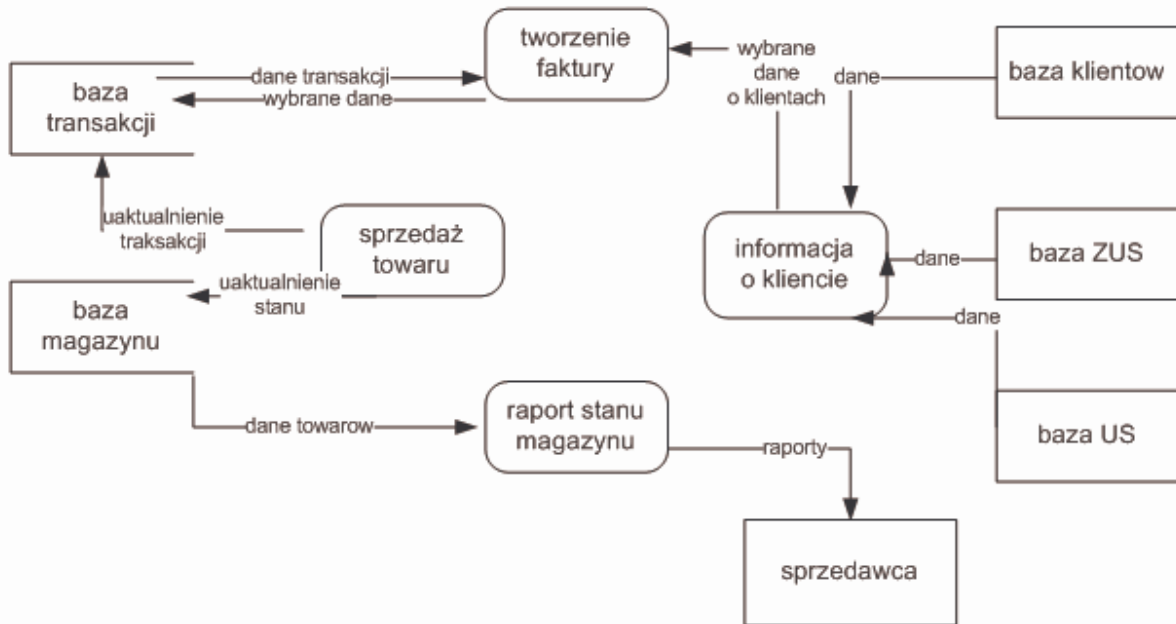
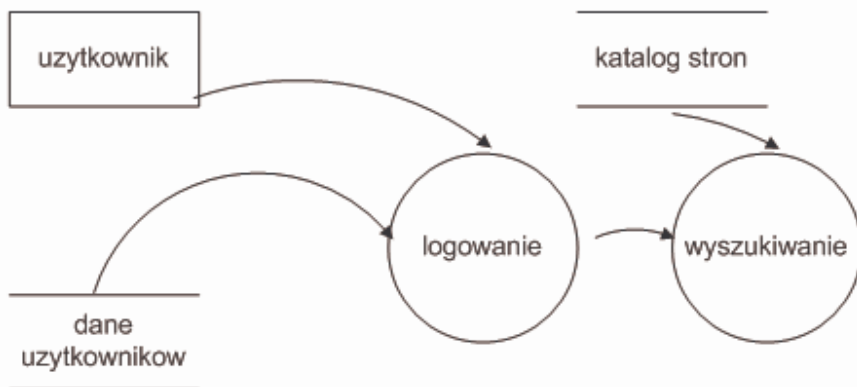


Diagram DFD (Yourdon – De Marco), MS Visio



Schemat relacji, MS Visio

