

1. Zasady identyfikacji przypadków użycia

<http://zofia.kruczkiewicz.staff.iar.pwr.wroc.pl/wyklady/analizasi/Wykladasi3.pdf>

(1) Identyfikacja aktorów i przypadków użycia – przypadek prostego systemu

1. Należy wyznaczyć granice systemu w ramach środowiska
2. Należy zdefiniować wymagania systemu: należy podać użytkowników - aktorów systemu, zależności między aktorami typu dziedziczenie (**generalization**) lub powiązanie (**association**), oczekiwane funkcje systemu (przypadki użycia), powiązania między aktorami i przypadkami użycia oraz zależności między przypadkami użycia
3. Należy opisać każdy przypadek użycia np. wg szablonu podanego poniżej:

Przypadek użycia zawiera:

- **Nazwę i opis**
- **Wymagania** funkcjonalne spełniane dla użytkownika
- **Ograniczenia** – warunki przed- po- przypadku użycia oraz niezmienną się na skutek wykonania przypadku użycia
- **Scenariusze** – sekwencja zdarzeń między systemem i zewnętrznymi użytkownikami (opis tekstowy)
- **Diagramy scenariuszy** – diagramy sekwencji
- **Dodatkowe informacje** – np. identyfikacja karty płatniczej przed dokonaniem wyciągu z konta

(2) Wytyczne przy modelowaniu granic systemu

- Należy zidentyfikować aktorów działających wokół systemu. Oznacza to wyznaczenie grup użytkowników korzystających w określonym celu z projektowanego systemu (zarządzanie, pielęgnacja, usługi) - (analiza wspólności i zmienności)
- Należy uporządkować aktorów wg zależności typu powiązanie np. Klient korzysta z usług Sprzedawcy lub dziedziczenia: Komercyjny Klient dziedziczy przypadki użycia od Klienta (analiza wspólności i zmienności)
- Należy powiązać aktorów z przypadkami użycia za pomocą powiązań nadając im zidentyfikowane znaczenie za pomocą stereotypu wg podanych definicji
- Należy określić otoczenie systemu, czyli zidentyfikować aktorów
- Dla każdego aktora należy podać działania, jakie każdy aktor oczekuje od systemu
- Działania należy zapisać jako przypadki użycia
- Należy wyłączyć powtarzające się ciągi działań i zastąpić je jednym nowym połączonym relacją <<**include**>> lub <<**extends**>> lub <<**use**>> lub zwykłym powiązaniem typu **association** bez stereotypu (analiza wspólności i zmienności)
- Należy uwzględnić tych aktorów, przypadki użycia oraz zidentyfikowane powiązania między nimi
- Można dodać do każdego aktora i przypadku użycia notatkę opisującą wymagania niefunkcjonalne (np. sprzęt, język, korzystanie z Internetu)
- Należy opisać główny i nadzwyczajne ciągi zdarzeń każdego przypadku użycia podając: **czynności i dane używane podczas działania przypadku użycia**
- Należy zdefiniować testy systemu w odniesieniu do wybranego aktora i powiązanego za nim jednego lub grupy przypadków użycia podając stan początkowy i końcowy oznaczający powodzenie testu przypadku użycia (np. *Wstawianie nowej książki jest możliwe tylko wtedy, gdy istnieje już jej tytuł w katalogu oraz posiada unikatowy numer. Po wstawieniu tej książki nie może być dwóch książek o tym samym numerze*)

Przykłady wymagań i opisu przypadków użycia (slajdy 11-20)

http://zofia.kruczkiewicz.staff.iar.pwr.wroc.pl/wyklady/analizasi/Wykladasi5_1.pdf

Przykłady opisu biznesowego, wymagań i opisu przypadków użycia (slajdy 3 - 13)

http://zofia.kruczkiewicz.staff.iar.pwr.wroc.pl/wyklady/analizasi/Laboratorium2_3_4.pdf

2. Identyfikacja encji w diagramie związków encji na podstawie opisów przypadków użycia (na podstawie Lech Banachowski, BazyDanych, Tworzenie aplikacji)

Dane przechowywane w bazie danych służą do odpowiedzi na polecenia (pytania, wstawianie, usuwanie i modyfikacja) wynikające z przypadków użycia.

W tym celu należy zidentyfikować dane występujące w przypadkach użycia – zarówno te, które należy dostarczyć oraz te, które należy uzyskać.

Należy zaprojektować pośredni model między wymaganiami przedstawionymi za pomocą przypadków użycia a docelowym schematem bazy danych. Ten pośredni model powinien w sposób jednoznaczny wyrażać wymagania użytkowników, umożliwiające klientowi sprawdzenie, czy analityk systemu dobrze zrozumiał ich intencje i specyfikę działania firmy. Ten pośredni model jest prostszy od schematu bazy danych, ponieważ abstrahuje od szczegółów implementacyjnych, które muszą być później opracowane przez projektanta bazy danych, aby baza danych mogła powstać i spełniać pozostawione przed nią zadania.

Tym modelem pośrednim jest diagram związków encji ERD.

Definicje związane z diagramem związków encji:

- Encja (obiekt) – coś, co istnieje, co jest odróżniane od innych, o czym informację trzeba znać lub przechowywać. Encje o tych samych własnościach tworzą zbiory encji. Na diagramie encje prezentuje się za pomocą prostokąta z podaną nazwą – oznacza to **typ encji**, gdyż encja jest traktowana jako konkretny **egzemplarz encji** np., *Tytuł_ksiazki*
- Atrybut jest to własność encji danego typu, reprezentowana pewną wartością np. łańcuch znaków, liczba całkowita, liczba rzeczywista itp. **Atrybut powinien opisywać encję**, przy której się go umieszcza. Należy odróżnić atrybuty, które:
 - atrybuty opisujące dane encję (są jej atrybutami „rodzonymi”),
 - atrybuty reprezentujące związki danej encji z innymi encjami.W pierwszej fazie projektowania identyfikuje się tylko atrybuty opisujące daną encję. Dla każdego egzemplarza encji każdy jej atrybut opisujący powinien przyjmować pojedynczą, atomową wartość. W jednej chwili encja powinna mieć tylko jedną wartość dla każdego atrybutu.
- Związek jest to uporządkowana lista encji, poszczególne encje mogą występować wielokrotnie. Każdy związek określa pewną relację między zbiorami egzemplarzy encji wchodzącymi w skład związku. Relację nazywamy instancją związku. Związek jest opisywany przy użyciu zdania zawierającego nazwy encji, których dotyczy i określającego, na czym ten związek polega.

Przykłady związków dwu- oraz trzy-argumentowych:

Np. Tytuł jest na Książce

Książka ma Tytuł

Rachunek zawiera informację o Zakupie (pozycja rachunku),

Zakup zawiera dane o zakupionym Produkcie

Wypożyczenie należy do Klienta

Klient posiada Wypożyczenie

Wypożyczenie zawiera dane Książki

Dane Książki po wypożyczeniu są umieszczone na Wypożyczeniu

Pracownik pracuje w Projekcie w Roli

- Typy związków
 - **związek wieloznaczny**, czyli wiele do wiele
np. Na podstawie związku między Pracownikiem i Projektem można powiedzieć, że wielu Pracownikach pracuje w jednym Projekcie i jeden Pracownik pracuje w wielu Projektach
 - **związek jednoznaczny**, czyli wiele do jeden, jeden do wiele
Przykładem związku jeden do wiele: Tytuł jest na wielu Książkach
Przykładem związku wiele do jeden: Wiele Książek ma jeden Tytuł
 - **związek jedno-jednoznaczny**, czyli jeden do jeden
Każdy Produkt może być kupiony, ale każdy Zakup dotyczy tylko jednego Produktu

- Każdy fakt przechowywany w bazie danych powinien być wyrażony w niej tylko na jeden sposób
 - (1) Informacja o Tytule powinna być zapisana tylko w jednym miejscu, a nie przy każdej książce, która ten Tytuł.
 - (2) Jeśli zapisujemy informacje o Wypożyczeniu Książki, to nie ma potrzeby zapisywania informacji o Tytule, gdyż można te informacje wyprowadzić z Książki
 - (3) Jeśli istnieją dwie odrębne encje w systemie sprzedaży: Pracownik i Klient, i jeśli Pracownik jest jednocześnie Klientem firmy, wówczas jego dane będą zapisane w dwóch różnych miejscach
- Każda dana w modelu powinna być istotna z punktu widzenia np. firmy
- Każda dana ważna w firmie, powinna być uwzględniona w modelu

Przykład

1) Identyfikacja encji oraz ich atrybutów opisujących encje i związki

http://zofia.kruczkiewicz.staff.iar.pwr.wroc.pl/wyklady/analizasi/Wykladasi5_1.pdf

Przykłady opisu biznesowego, wymagań i opisu przypadków użycia (slajdy 3 - 13)

Nazwa przypadków użycia –PU (podstawowe operacje na danych)	Atrybuty „rodzone”	Atrybuty reprezentujące związki	Atrybut identyfikujący encję	Nazwa Encji
Szukanie produktu	Nazwa Cena Podatek Promocja		Id_produkту	Produkt
Wstawianie nowego produktu	Nazwa Cena Podatek Promocja		Id_produkту	Produkt
Szukanie rachunku	Numer_rachunku		Numer_rachunku	Rachunek
Wstawianie nowego rachunku	Numer_rachunku		Numer_rachunku	Rachunek
Wstawianie nowego zakupu	Numer_rachunku Ilosc_produkту Nazwa Cena Podatek Promocja	Numer_rachunku Id_produkту	Numer_rachunku	Rachunek
			Id_zakupu	Zakup
			Id_produkту	Produkt
Obliczanie wartosci rachunku	<i>Wartosc_rachunku</i> (atrybut wyliczeniowy= Cena_brutto, Ilosc_produkту) <i>Cena_brutto</i> (atrybut wyliczeniowy= Cena, Podatek, Promocja) Podatek	Numer_rachunku	Numer_rachunku	Rachunek
			Id_zakupu	Zakup
			Id_produkту	Produkt

Tabela 1. Identyfikacja encji oraz atrybutów opisujących encje oraz związki między encjami

2) Identyfikacja liczności związków

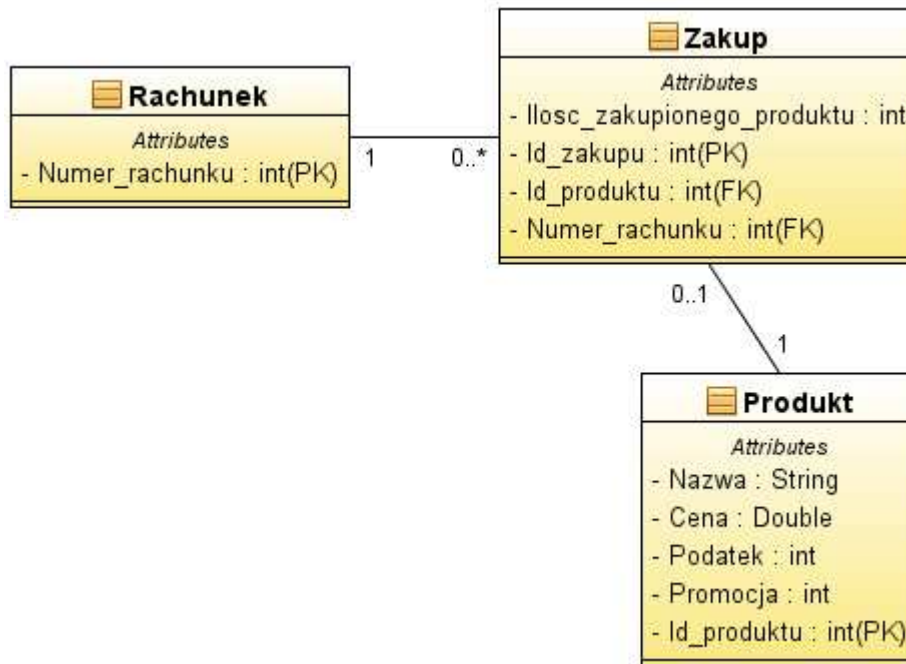
Rachunek zawiera informację o wielu Zakupach (pozycja rachunku) – jeden do wiele, Rachunek może istnieć bez Zakupow

Zakup należy do Rachunku – wiele do jeden,

Zakup nie może istnieć bez Rachunku – wiele..0 do jeden

Zakup zawiera dane o zakupionym Produkcie - jeden do jeden, w tym samym Rachunku może wystąpić tylko jeden Zakup z danym typem Produktu, można zwiększyć Ilosc produktu,

3) Uzyskany diagram reprezentujący informacje o encjach, atrybutach encji oraz związkach między encjami



Rys.1. Diagram reprezentujący encje, atrybuty encji oraz związki między encjami