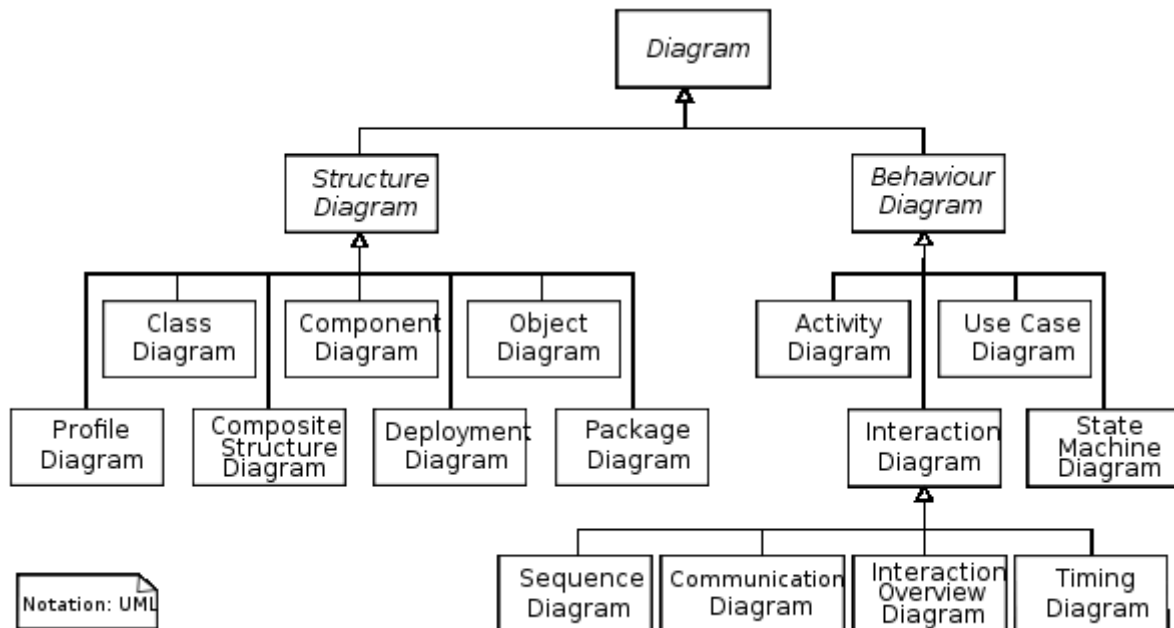


Projektowanie obiektowe oprogramowania

Wykład 1 - UML

Wiktor Zychła 2014

1 Wprowadzenie

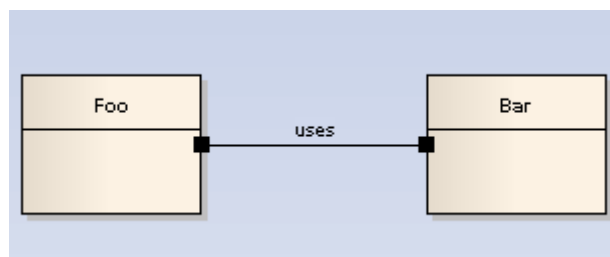


- Diagramy struktury i diagramy zachowań (dynamiki)

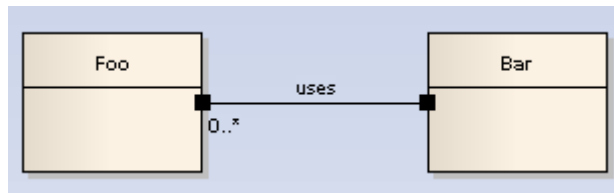
2 Diagramy klas

2.1 Klasy i asocjacje

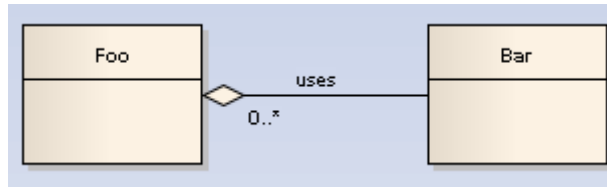
- Zależności (strzałka przerywana) – brak informacji o rodzaju zależności, może być:
 - Tworzy
 - Wykorzystuje (zmienna lokalna)
 - Wykorzystuje (parametr metody)
 - Nadklasa lub interfejs
- Nazwy asocjacji



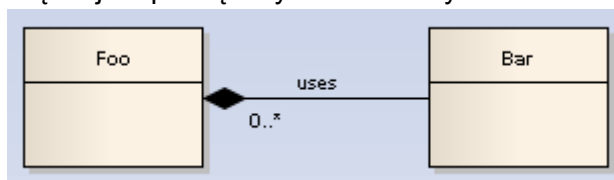
- Liczebność : 1, 1..*, 0..1, *, 0..*, n, 1..n, 0..n, n..m, n..*



- Agregacja vs kompozycja
 - Agregacja – luźniejsza

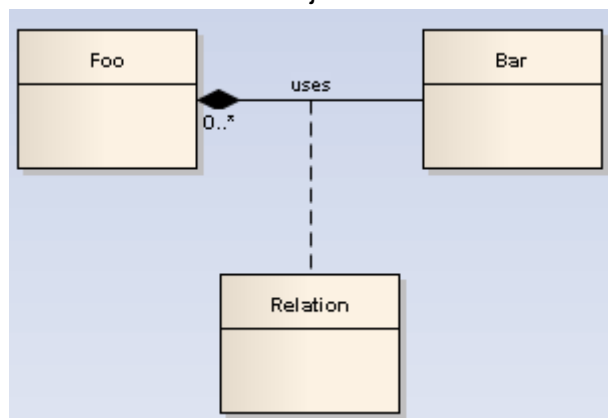


- Kompozycja - ściślejsza
 - Instancja reprezentująca część może należeć tylko do jednej instancji złożonej
 - Czas życia części jest powiązany z czasem życia całości



2.2 Składowe

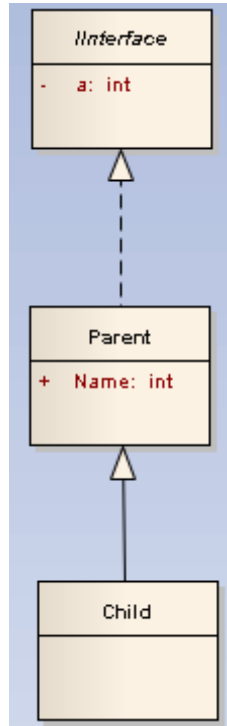
- Składowa prywatna, publiczna, chroniona, stała, statyczna, kolekcja, atrybut pochodny
- Metoda prywatna, publiczna, chroniona, internal, abstrakcyjna, statyczna, konstruktor, parametry
- Atrybut wpisany vs asocjacja – kiedy używać? Atrybut: typ prosty, asocjacja do typu złożonego
- Klasa asocjacyjna – do modelowania relacji wiele-wiele



2.3 Dziedziczenie

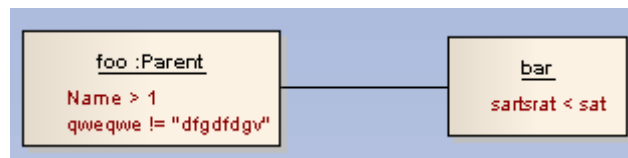
- Realizacja – implementacja interfejsu

- Generalizacja, specjalizacja – dziedziczenie (tylko w zależności od kierunku)



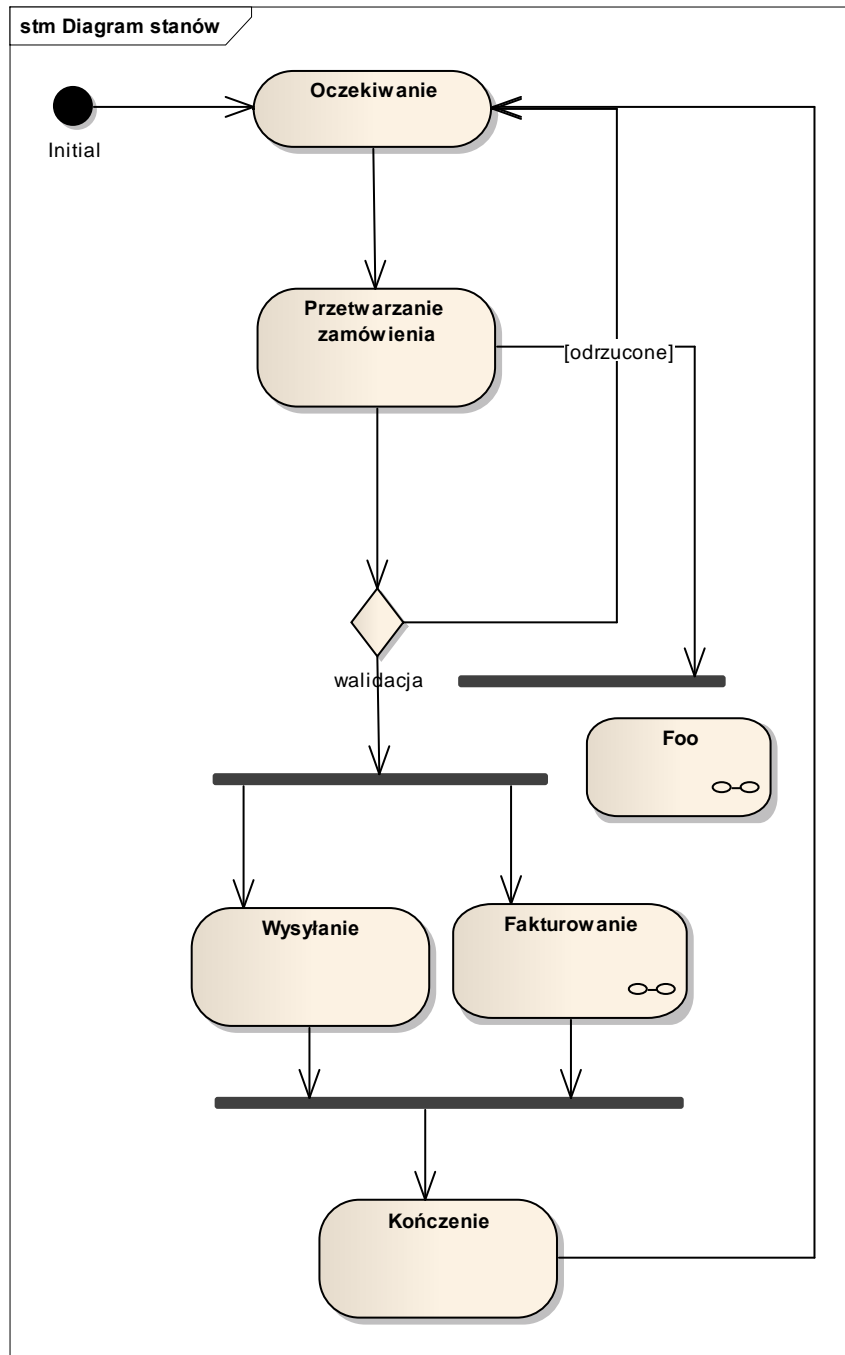
3 Diagramy obiektów

- Migawka systemu
- EA: Advanced / Instance Classifier
- EA: Advanced / Set Run State



4 Diagramy stanów

- Stany i przejścia (akcje) – stany to bloczki, a akcje to strzałki
- Stany – nazwane rzeczownikowo/przymiotnikowo (oczekiwanie/przetwarzanie, oczekujący/aktywny/przydzielony)
- Akcje – nie nazywają się
- Przykładowy schemat
 - Stany – oczekiwanie, przetwarzanie
 - Wariant – nazwany
 - Zrównoleganie – wysyłanie, fakturowanie
 - Stan kompozytowy

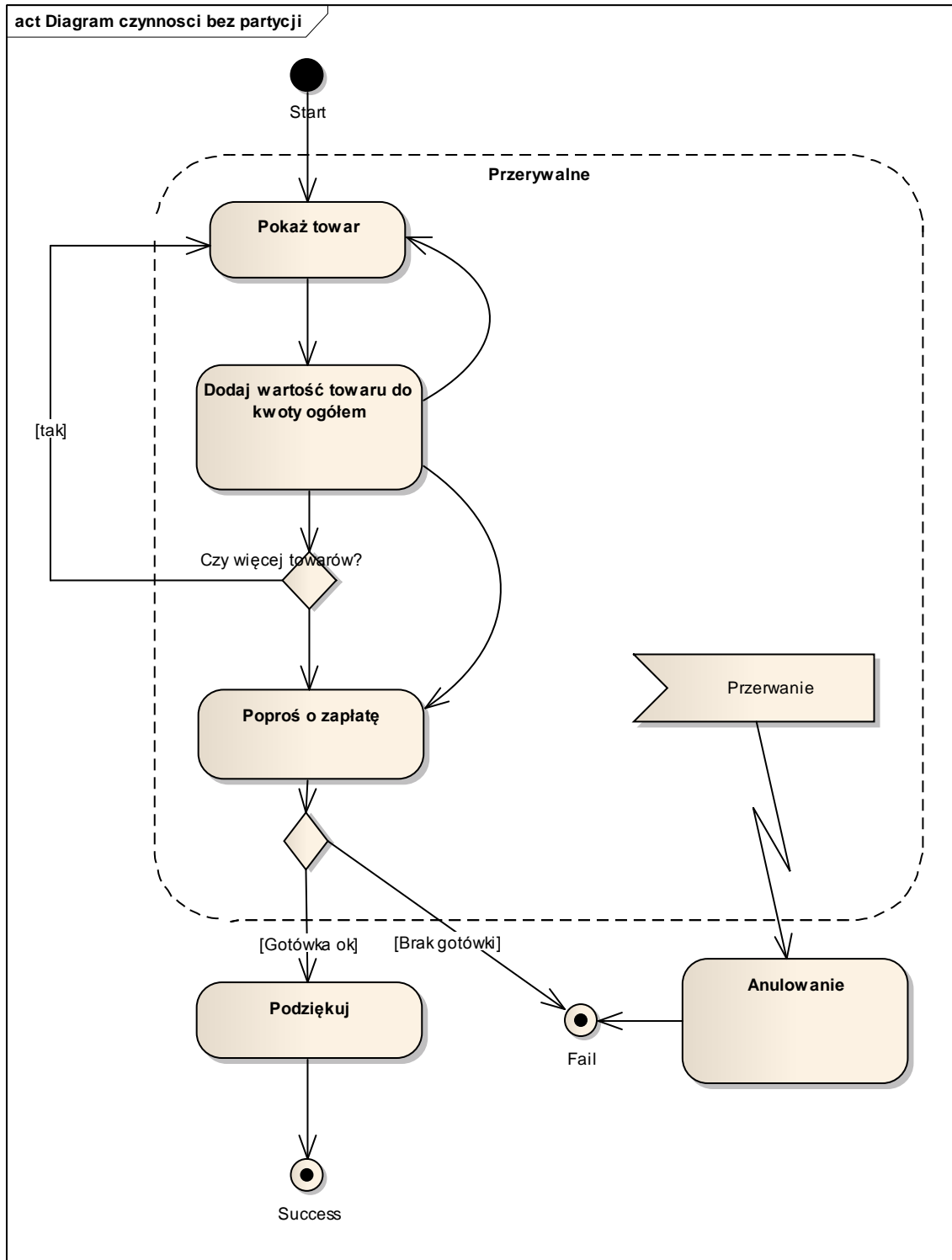


5 Diagramy czynności

- Czynności vs akcje
 - Czynności – długotrwałe, podzielne, ogólne
 - Akcje – krótkotrwałe, niepodzielne, szczegółowe – nazwane czasownikowo (wprowadź/wyberz/zatwierdź/wydrukuj/aktualizuj/weryfikuj)
- Różnica w stosunku do diagramu stanów jeśli chodzi o semantykę bloków vs strzałek – tam bloczek = stan, strzałka = akcja; tu bloczek = akcja, strzałka – wyznacza następstwo akcji
 - Sygnały (zdarzenia) – wyślij, odbierz
 - Warian – „if”

- Zdarzenia – send/receive
- Regiony – na przykład „przerywalny”, pojawia się zdarzenie „przerwij”, anulowanie
- Partycje – podział na aktorów

Diagramy stanów i czynności wykorzystują niemalże ten sam formalizm do reprezentowania różnych kategorii diagramów.



6 Diagramy sekwencji

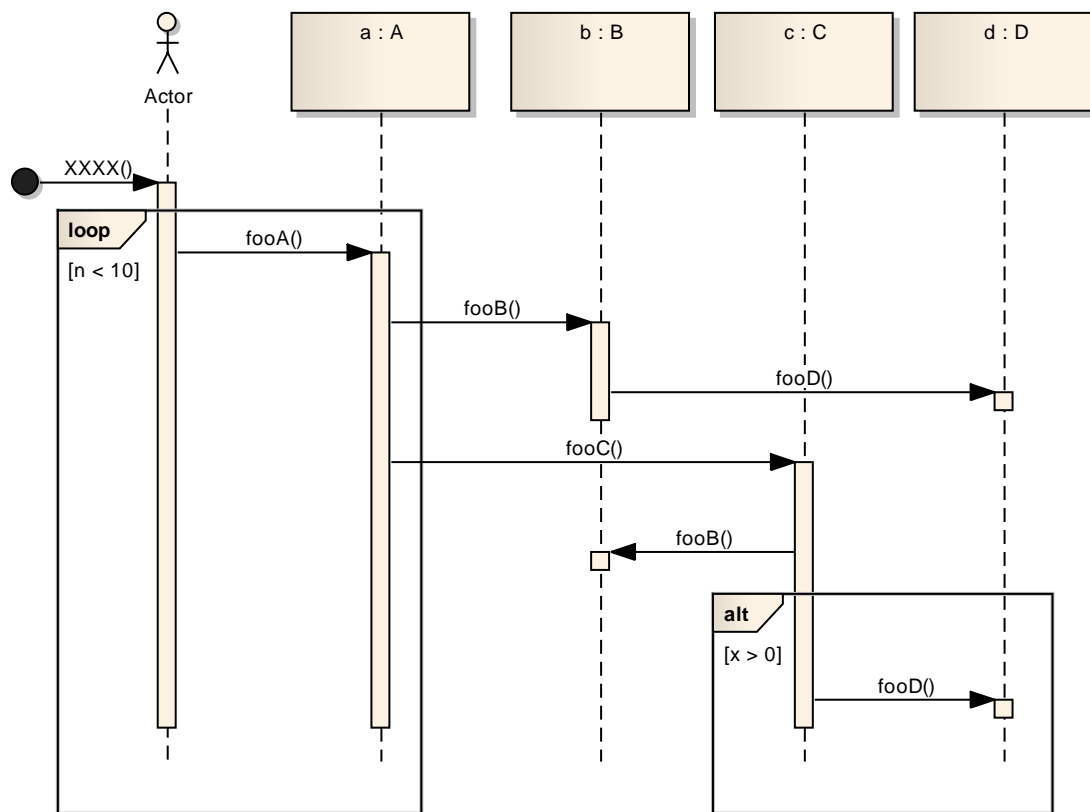
- Linie życia, paski aktywacji/ośrodki sterowania (execution specification)
- Typy obiektów
 - Boundary – widok
 - Control – kontroler
 - Entity – model
- Związek między diagramem sekwencji a diagramem klas – ustalanie typu obiektu
- Komunikat – wartość zwrotna
wartość = komunikat(p1:P1, p2:P2, ...) : typ
- lub przerywana strzałka zwrotna (EA – niekoniecznie)
- Singleton – jedynka w rogu, metoda statyczna – stereotyp „class”, „metaclass”
- Komunikat odnaleziony – „od nikogo”
- Create/destroy
- Ramki, można zagnieżdżać
 - Loop – pętla
 - Alt – if-then-else
 - Opt – if
 - Neg – czynność nieprawidłowa, wyjątek
 - Par - współbieżność
 - Ref – odwołanie do innej, nazwanej ramki
 - Sd – nazwana ramka

Przykładowy pseudokod:

```
public class Actor {
    public void XXXX() {
        while ( n < 10 ) {
            a.fooA();
        }
    }
}
public class A {
    public void fooA() {
        b.fooB();
        c.fooC();
    }
}
public class B {
    public void fooB() {
        d.fooD();
    }
}
public class C {
    public void fooC() {
        b.fooB();
        if ( x > 0 )
            d.fooD();
    }
}
```

i jego diagram

sd Diagram sekwencji



7 Literatura

Wrycza, Marcinkowski, Wyrzykowski - Język UML 2.0 w modelowaniu systemów informatycznych

