Inżynieria oprogramowania

Wykład 2:

Wprowadzenie do Unified Modeling Language
Diagramy przypadków użycia, diagramy czynności

Marek Krętowski
pokój 206
e-mail: m.kretowski(at)pb.edu.pl
http://aragorn.pb.bialystok.pl/~mkret

Modelowanie

Podstawowe zasady modelowania:

- Podjęcie decyzji, jakie modele tworzyć, ma wielki wpływ na to, w jaki sposób "zaatakujemy" problem i jaki kształt przyjmie rozwiązanie
- Każdy model może być opracowany na różnych poziomach szczegółowości
- Najlepsze modele odpowiadają rzeczywistości
- Żaden pojedynczy model nie jest wystarczający. Niewielka liczba niemal niezależnych modeli to najlepsze rozwiązanie w przypadku każdego niebanalnego systemu

Model jest uproszczeniem rzeczywistości, który opracowujemy po to, aby (lepiej) zrozumieć budowany system

Unified Modeling Language

The UML is a language for visualizing, specifying, constructing and documenting the artifacts of software-intensive systems

UML jest językiem do:

- obrazowania
- specyfikowania
- tworzenia
- dokumentowania

wytworów powstałych podczas budowania systemu informatycznego

UML **nie jest** metodyką analizy i projektowania
UML **nie definiuje** procesu rozwoju oprogramowania

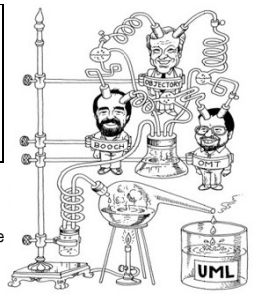
Główni/pierwsi autorzy UML

Połączone siły trzech metodologów:


Grady Booch


Ivar Jacobson


James Rumbaugh



Wykorzystano doświadczenia zdobyte przy tworzeniu nast. metodyk:

- OMT (Rumbaugh *et al.*) - sprawdza się w modelowane dziedziny przedmiotowej
- OOSE (Jacobson) - modelowanie użytkowników (przypadki użycia) i cykl życiowy systemu
- OOAD (Booch) - dobrze podchodzi do kwestii projektowania, konstrukcji i związków ze środowiskiem implementacji
- Fusion (Coleman *et al.*)

Trochę historii

02.2009	Wersja 2.2
2004	Wersja 2.0
2001	Wersja 1.4
1999	Wersja 1.3 opublikowana przez OMG Revision Task Force
11.1997	Wersja 1.1 ostatecznie przyjęta przez OMG
01.1997	UML wersja 1.0 przesłany do Object Management Group jako propozycja standardu języka modelowania obiektowego
	Utworzenie konsorcjum (m.in. DEC, HP, IBM, Microsoft, Oracle, ...)
06.1996	Wersja 0.9
	Do projektu dołącza Jacobson (uwzględnienie OOSE)
10.1995	Wersja 0.8 jeszcze jako Unified Method
10.1994	Początek prac - Booch i Rumbaugh w firmie Rational

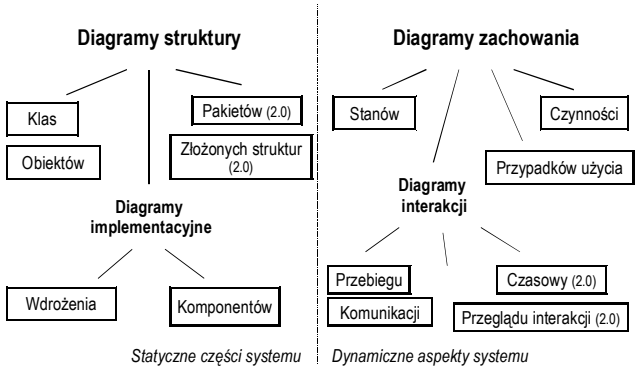
Zakres UML

- Skupiono się na standardzie języka do modelowania, a nie na standardzie procesów tworzenia oprogramowania
  - różne organizacje i problemy wymagają różnych procesów, które mogą być opisane przy pomocy tego samego języka
- Wysiłek autorów jest skoncentrowany na stworzeniu wspólnego metamodelu (unifikacji semantyki) i wspólnej notacji (odbioru tej semantyki przez ludzi)
- Przy czym promowany jest proces, który jest:
  - ukierunkowany na przypadki użycia systemu
  - skoncentrowany na architekturze
  - iteracyjny i przyrostowy

Bloki konstrukcyjne:

- Elementy (ang. *things*):
  - strukturalne (np. przypadki użycia, klasy, interfejsy, komponenty, węzły, ...)
  - czynnościowe (np. interakcja i maszyna stanowa)
  - grupujące (pakiety)
  - komentujące (notatki)
- Związki (zależności, powiązania, uogólnienia, realizacje, ...)
- Diagramy:
  - struktury
  - zachowania

Rodzaje diagramów



Diagramy struktury

- **Klas** - najczęściej spotykany diagram w modelach obiektowych
- **Obiektów** - wyobraża zrzut pewnych egzemplarzy elementów występujących na d. klas
- **Komponentów** - obrazuje zbiór komponentów i związki pomiędzy nimi; ściśle związany z d. klas
- **Wdrożenia** (ang. *deployment*) - obrazuje zbiór węzłów i związki między nimi; wiąże się z d. komponentów, gdyż zwykle każdy węzeł posiada co najmniej jeden komponent
- **Pakietów** – służy do prezentowania grup bytów (w postaci pakietów) i relacji pomiędzy tymi grupami
- **Struktury - złożonych struktur** (ang. *composite structure*) - przedstawienie wewnętrznej struktury klasyfikatora (np. klasy, komponentu, ...) z uwzględnieniem punktów interakcji z innymi częściami systemu

Diagramy zachowania

- **Przypadków użycia** - umożliwia uporządkowanie zachowania systemu
- **Przebiegu - sekwencji** (ang. *sequence*) - kładzie nacisk na kolejność wysyłania komunikatów w czasie
- **Komunikacji - współpracy** (ang. *collaboration*) - kładzie nacisk na strukturalną organizację obiektów, które wysyłają i odbierają komunikaty

  Diagramy przebiegu i współpracy są w zasadzie izomorficzne (można "swobodnie" przekształcać jeden w drugi)
- **Stanów** - zmiany stanów systemu spowodowane zdarzeniami
- **Czynności - aktywności** (ang. *activity*) - obrazuje strumień kolejno wykonywanych czynności; przepływ sterowania od czynności do czynności
- **Przeglądu interakcji – widoku interakcji** (ang. *interaction overview*) – połączanie d. czynności i przebiegu; prezentowanie zależności i przepływu wiadomości pomiędzy interakcjami
- **Czasowy – przebiegów czasowych** (ang. *timing*) – funkcjonowanie obiektów w kontekście ograniczeń/wymagań czasowych

<< Stereotypy >>

- Stereotypy stanowią jeden z mechanizmów rozszerzalności UML. Dają możliwość definiowania nowych elementów, co ułatwia przystosowanie języka do modelowania specyficznego procesu, do specyficznych preferencji użytkownika czy też pozwala na uszczegóławianie semantyki modelu
- Stereotypy są wyrażeniami językowymi (nazwami) umożliwiającymi metaklasyfikację elementów modelu; są wspólnymi nazwanymi własnościami
- Element modelu może mieć co najwyżej jeden stereotyp
- Istnieje lista stereotypów dla każdego rodzaju elementów, są stereotypy predefiniowane (np. stereotypy klas: aktor, zdarzenie, wyjątek, ...), ale użytkownicy mogą też definiować własne
- Stereotypy mogą mieć implikacje semantyczne (ograniczenia)

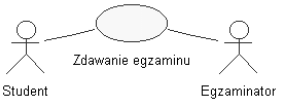
Diagramy przypadków użycia

Przykład scenariusza: student zdaje egzamin

Scenariusz to ciąg kroków opisujących interakcję (pomiędzy użytkownikiem a systemem)

Przykładowy scenariusz:

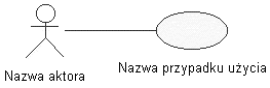
- Student (S) zgłasza się na egzamin
- Daje egzaminatorowi (E) kartę i indeks
- E sprawdza tożsamość S i czy figuruje na liście egzaminacyjnej
  - Jeśli brak to koniec egzaminu
- E zadaje pytanie
- S odpowiada
- E ocenia odpowiedz i notuje ocenę
  - jeśli ostatnie pytanie to dalej
  - wpp powtarzane są 3 ostatnie punkty
- E wystawia ocenę końcową
- E wpisuje ocenę do indeksu, karty i protokołu



- Wartość mierzalna: wpisana ocena
- Potencjalne klasy w systemie: aktorzy (student i egzaminator), indeks, karta, protokół, ocena (?)

Diagramy przypadków użycia

- Diagramy przypadków użycia zawierają
  - przypadki użycia
  - aktorów
  - zależności, uogólnienia i powiązania



- Dwa podstawowe cele:
  - modelowanie otoczenia systemu (wyznaczenie granicy systemu i wskazanie aktorów, którzy wchodzą w interakcję z systemem)
  - modelowanie wymagań stawianych systemowi (określenie, z punktu widzenia otoczenia, co system powinien robić niezależnie od tego jak ma to zrobić)

Aktorzy

- Aktor reprezentuje spójny zbiór ról odgrywanych przez użytkowników przypadków użycia w czasie interakcji
- Aktorami mogą być ludzie, urządzenia i inne systemy informatyczne
- Jedna osoba może wchodzić w interakcję z systemem z pozycji wielu aktorów; np. być zarówno sprzedawcą, jak i klientem. I odwrotnie, jeden aktor może odpowiadać wielu konkretnym osobom, np. aktor "strażnik budynku"
- Aktor jest tu pierwotną przyczyną napędzającą przypadki użycia. Jest on sprawcą zdarzeń powodujących uruchomienie przypadku użycia
- Aktorzy aktywni (inicjują przypadki użycia) i pasywni

lub

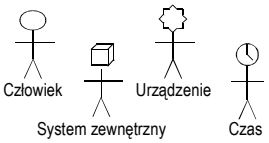
<<actor>>
Nazwa aktora

np.

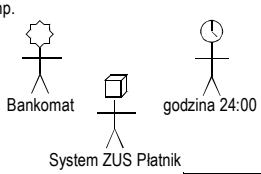
Aktor to jest **rola**, nie jest związany z żadną konkretną osobą

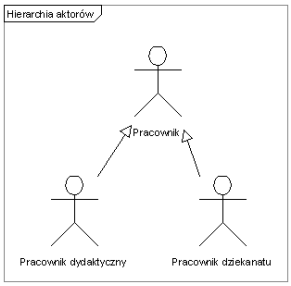
Aktorzy (2)

Rodzaje aktorów - przykładowe stereotypy



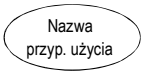
np.



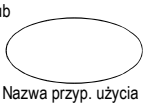


Przypadki użycia

- Przypadek użycia to opis zbioru ciągów akcji (i ich wariantów) wykonywanych przez system w celu dostarczenia określonemu aktorowi godnego uwagi wyniku
  - zbiór scenariuszy powiązanych ze sobą wspólnym celem
- Przypadek użycia opisuje oczekiwane zachowanie budowanego systemu (podsystemu, klasy lub kooperacji), ale nie określa sposobu implementacji tego zachowania
- Z punktu widzenia aktora przypadek użycia opisuje działanie mające dla niego wartość, np. obliczenie wyniku, utworzenie nowego obiektu lub zmianę stanu obiektu
- Średnio skomplikowany system ma około kilkudziesięciu przypadków użycia
- Czasami wyróżnia się 2 rodzaje przypadków użycia:
  - biznesowe - opisują jak przedsiębiorstwo (organizacja) reaguje na klienta lub zdarzenia
  - systemowe - dotyczą interakcji z oprogramowaniem



lub



np.

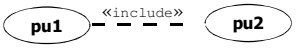
Przypadki użycia - dokument opisu

- Dla każdego przypadku użycia tworzymy dokument opisujący przebieg zdarzeń opisany z punktu widzenia aktora
- UML nie precyzuje standardu opisu przypadków użycia
- Typowy opis zawiera:
  - jak i kiedy przypadek użycia się rozpoczyna i kończy
  - kto uczestniczy (aktorzy)
  - kiedy dochodzi do interakcji z aktorami
  - jakie obiekty są przekazywane
  - **typowy i alternatywne przebiegi (ciągi) zdarzeń**
  - przebieg zdarzeń w sytuacjach szczególnych (wyjątkowych)
  - **parametry czasowe**: częstotliwość wykonania, przewidywane spiętrzenia oraz czasy realizacji (typowy, maksymalny)
  - opis wartości uzyskiwanych przez aktorów po zakończeniu działania p. użycia
- Ciąg zdarzeń można zapisać na wiele sposobów:
  - tekst strukturalny (nieformalny lub formalny)
  - pseudokod - diagram czynności

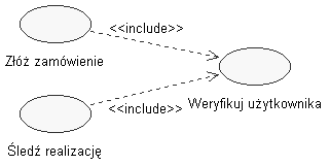
Związki pomiędzy przypadkami - zawieranie

- Zawieranie wykorzystujemy, gdy kilka przypadków użycia ma wspólną sekwencję podobnych kroków, której nie warto ciągle kopiować z jednego przypadku do innych
- Pozwala na uniknięcie wielokrotnego opisywania tego samego ciągu zdarzeń
- Wspólne zachowanie jest definiowane w odrębnym przypadku użycia, który jest następnie włączany przez bazowe przypadki użycia
- Związek zawierania - stereotyp `<<include>>`
- Grot strzałki wskazuje na przypadek włączany!

p1 jest przypadkiem bazowym i zawsze występuje jako pierwsze w kolejności działania

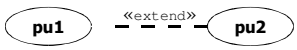


Przebieg podstawowy (sekwencyjny): pu1 **ZAWSZE** włącza (używa) pu2

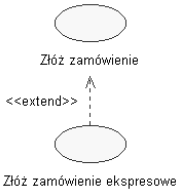


Związki pomiędzy przypadkami - rozszerzenie

- Służy do modelowania fragmentów przypadków użycia postrzeganych przez użytkownika opcjonalnie
- Oddzielenie działań opcjonalnych od wymaganych
- Stereotyp `<<extend>>`
- Uwaga! Należy zwrócić uwagę na grot strzałki reprezentującej zależność - przy rozszerzaniu wskazywany jest przypadek rozszerzany (podstawowy)

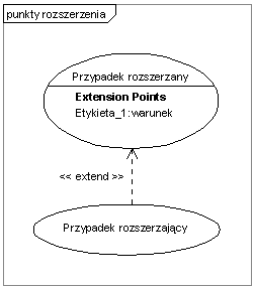


Przebieg opcjonalny (alternatywny):
pu1 jest **CZASAMI** rozszerzane o pu2



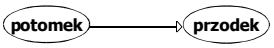
Punkty rozszerzenia (ang. extension points)

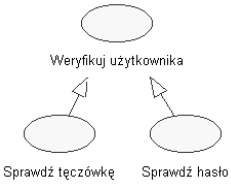
- Przypadek bazowy może być rozszerzony tylko w ściśle określonych miejscach (tzw. miejsca rozszerzenia; rozpoznawane przez etykiety; miejsc tych może być kilka)
- Jeżeli zachodzą warunki związane z rozszerzeniem to w miejscu rozszerzenia wykonywane są czynności opisane w rozszerzającym przypadku użycia po czym kontynuowane jest przetwarzanie bazowe



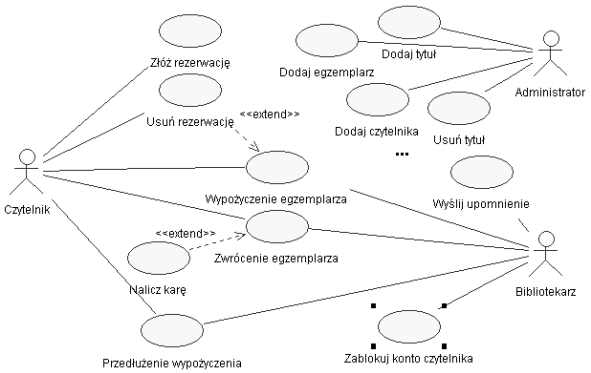
Związki pomiędzy przypadkami - uogólnienie

- Związek analogiczny do uogólnienia między klasami
- Przypadek-potomek dziedziczy całe zachowanie i znaczenie po przypadku-przodku
- Potomek może dodać nowe elementy do odziedziczonego zachowania lub zmienić całkowicie zachowanie
- Potomek zawsze może zastąpić swego przodka
- Przedstawiane jako ciągła linia zakończona zamkniętym, niewypełnionym grotem





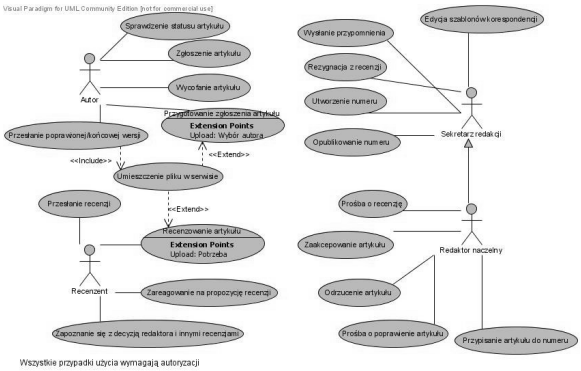
Przykład diagramu przypadków użycia: część systemu bibliotecznego



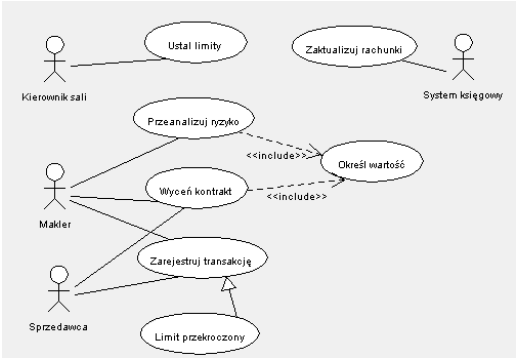
Przykład: opis przypadku użycia Wypożyczenie egzemplarza

- Aktorzy: Czytelnik - inicjuje przypadek, Bibliotekarz
- Typowy przebieg:
  - Czytelnik podaje kartę biblioteczną, która jest skanowana przy użyciu czytnika kodów kreskowych w celu identyfikacji
  - Następnie Bibliotekarz rozkodowuje książki, które wybrał Czytelnik
  - Czytelnik wychodzi z książkami
- Alternatywny przebieg 1:
  - System stwierdza, że Czytelnik jest dłużnikiem i Bibliotekarz zatrzymuje kartę do czasu zwrotu książek i opłacenia kary,
- Alternatywny przebieg 2:
  - Maksymalna liczba książek na koncie - odmowa wypożyczenia
- Alternatywny przebieg 3:
  - Po identyfikacji czytelnika okazuje się, że oczekują na niego zarezerwowane książki
  - książki są przekazywane Czytelnikowi i usuwana jest rezerwacja
- Wartość uzyskiwana:
  - wypożyczone książki
- Parametry czasowe:
  - częstotliwość: z punktu widzenia Czytelnika co 2-3 tyg; Bibliotekarza kilkanaście na godzinę
  - czas trwania: typowo kilkadziesiąt sekund; maksymalnie 3 minuty
  - spiętrzenia: przed długimi weekendami, świętami

Przykład diagramu przypadków użycia: serwis redakcyjny czasopisma naukowego



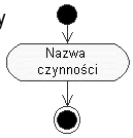
Przykład diagramu przypadków użycia: część systemu maklerskiego



Diagramy czynności

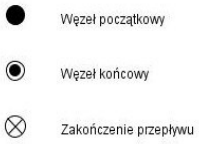
Diagramy czynności

- Służący do modelowania dynamicznych aspektów systemu
- Może być traktowany jako schemat blokowy reprezentujący przepływ sterowania od czynności do czynności
- Zwykle przedstawia sekwencyjnie (rzadziej współbieżne) kroki procesu obliczeniowego lub innego przetwarzania
- Można na nim zobrazować również zmiany zachodzące w obiekcie, gdy przechodzi on z jednego stanu do drugiego w różnych fazach przepływu sterowania
- Służy do modelowania przepływu czynności i modelowania operacji
- Diagramy tego rodzaju mogą być wykorzystywane w systemach inżynierii do przodu (generowanie kodu na podstawie diagramu) i inżynierii wstecz (odtwarzanie diagramu na podstawie kodu programu)



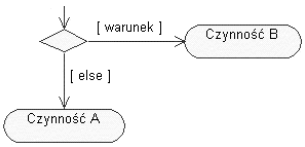
Przejścia

- Gdy czynność kończy się, sterowanie jest natychmiast przekazywane do następnej czynności (przejścia automatyczne, zakończeniowe) zakładając, że odpowiednie akcje wyjściowe i wejściowe są wykonywane (o ile istnieją)
- Przejścia oznaczane są poprzez zwykłą strzałkę
- Przepływ sterowania może trwać bez końca (w przypadku czynności nieskończonych) lub do chwili osiągnięcia węzła końcowego
- Wybrany przepływ może zostać również zakończony bez zakończenia całej czynności
- Przejścia mogą zawierać warunki dozoru
  - podawane w nawiasach [ ] dowolne wyrażenie logiczne



Rozgałęzienia

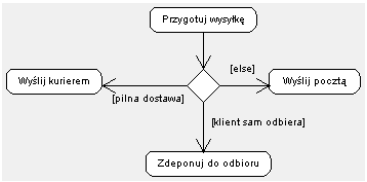
- Rozgałęzienia (symbolem jest romb) opisują ścieżki alternatywne; do wyboru jednej z nich dochodzi na podstawie wyliczonych wartości wyrażeń logicznych
- Na każdym przejściu wyjściowym powinien być umieszczony warunek dozoru (wyliczany raz, w momencie wejścia do rozgałęzienia)
- Warunki nie mogą się nakładać i muszą uwzględniać wszystkie możliwości



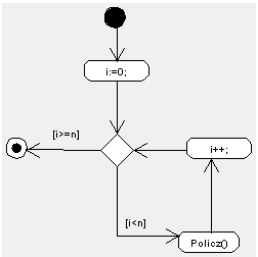
- Słowo kluczowe **`else`** służy do oznaczenia jednego przejścia wyjściowego, reprezentującego ścieżkę wybieraną, gdy wszystkie inne warunki dozoru nie są spełnione

W UML nie jest określona forma wyrażenia dozoru. Może to być np. tekst strukturalny lub konkretny język programowania

Rozgałęzienia - przykłady



Słowo kluczowe `else`



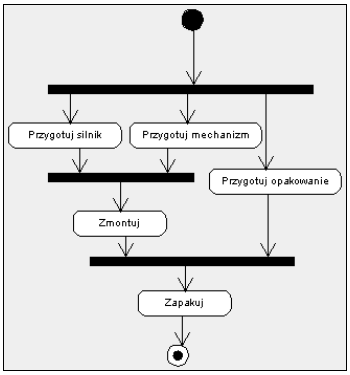
Iteracja

Rozwidlanie i scalanie ścieżek

- Służą do modelowania współbieżnych (równoległych) przepływów sterowania
- Obrazuje się je za pomocą pasków synchronizacyjnych; mają one postać poziomych lub pionowych grubych kresek
- Rozwidlenie reprezentuje podział przepływu sterowania na dwa lub więcej (niezależnych) przepływów współbieżnych
- W punkcie scalenia dochodzi do synchronizacji współbieżnych przepływów sterowania (oczekiwanie aż ostatni przepływ osiągnie punkt i dopiero wtedy scalony przepływ podąża dalej)

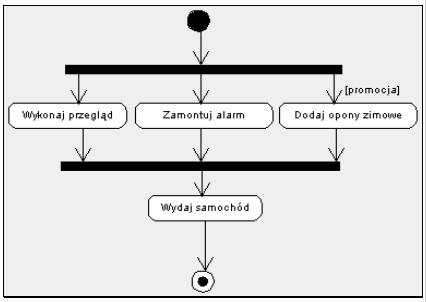
Rozwidlenia i scalenia powinny się równoważyć (liczba przepływów opuszczających rozwidlenie = liczba przepływów wchodzących do odpowiadającego mu scalenia)

Rozwidlenia i scalenia - przykład



Rozwidlenia i scalenia - wątek warunkowy

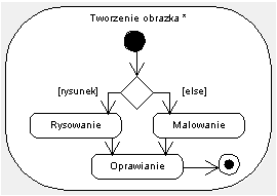
- Do przepływu wychodzącego z rozwidlenia można dodać warunek (tzw. wątek warunkowy)
- W trakcie wykonania, jeśli warunek jest fałszywy, zakłada się, że z punktu widzenia scalenia wątek ten jest już zakończony



Współbieżność dynamiczna

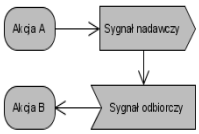
- Umożliwia zilustrowanie iteracji bez konieczności tworzenia pętli
- Znacznik iteracji "*" oznacza, że czynność jest wykonywana wielokrotnie

- W sytuacji, gdy chcemy powtórzyć grupę kilku czynności, wówczas można to oznaczyć wprowadzając czynność złożoną (analogicznie do stanów złożonych na diagramie stanów) składającą się z podczynności

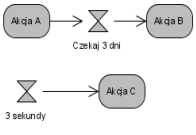


Sygnały i zdarzenia czasowe (2.0)

- Sygnały reprezentują interakcje z zewnętrznymi uczestnikami:
  - sygnały są komunikatami asynchronicznymi
  - węzeł sygnału odbieranego może spowodować uruchomienie akcji przedstawionej na diagramie czynności.
  - węzeł sygnału nadawanego wysyła go do zewnętrznych uczestników.



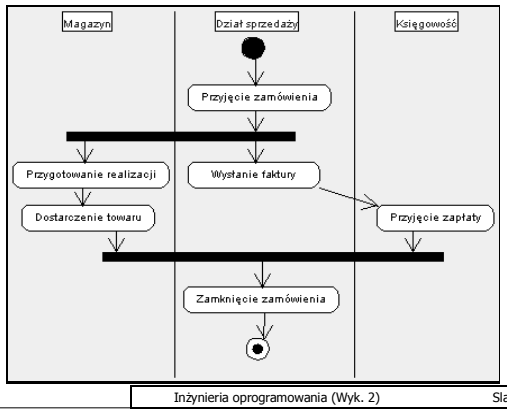
- Zdarzenie czasowe – pozwala zamodelować okres oczekiwania
  - krawędź wchodząca do zdarzenia czasowego oznacza, że jest ono aktywowane tylko raz.
  - zdarzenie czasowe bez wchodzących przepływów jest cykliczne, co oznacza, że jest aktywowane w odstępach czasu podanych obok symbolu klepsydry



Tory (1.5), partycje (2.0)

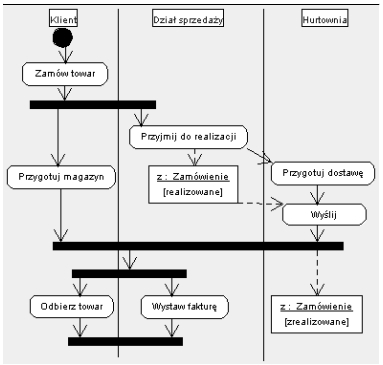
- Służą do podzielenia czynności na grupy, z których każda reprezentuje jednostkę (np. przedsiębiorstwa lub organizacji) odpowiedzialną za przydzielone czynności; każda grupa nosi nazwę toru (ang. swimlanes) (analogia do pływalni czy bieżni)
- Tory wskazują na umiejscowienie czynności i przydatne są zwłaszcza w modelowaniu procesów zachodzących w przedsiębiorstwach
- Każdy tor ma unikatową w ramach diagramu nazwę; sam tor nie ma żadnego szczególnego znaczenia, oprócz tego, że może odpowiadać pewnemu bytowi ze świata rzeczywistego i reprezentuje określone na wysokim poziomie abstrakcji zobowiązanie realizacji czynności
- Tor może być zaimplementowany przez co najmniej jedną klasę
- Na diagramie czynności każda czynność należy do dokładnie jednego toru, ale przejścia mogą przecinać granice torów

Tory - przykład



Przepływ obiektów

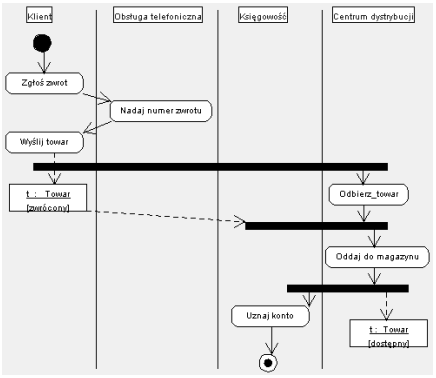
- Obiekty związane z czynnościami lub przepływami można umieścić na diagramie i połączyć je związkiem zależności
- Można zobrazować również np. zmiany stanów obiektów (w nawiasach [ ] pod nazwą obiektu) lub zmiany wartości atrybutów (w dodatkowej sekcji poniżej nazwy)



Modelowanie przepływu czynności

- Koncentrujemy się na czynnościach i badamy je z punktu widzenia aktorów współpracujących z systemem
- Przepływy czynności znajdują się często na pograniczu systemów informatycznych i służą do obrazowania, specyfikowania, tworzenia i dokumentowania procesów zachodzących w przedsiębiorstwie, związanych z modelowanym systemem
- Na jednym diagramie nie da się przedstawić wszystkich istotnych przepływów czynności w danej organizacji (*co chcemy przedstawić?*)
- Przypisz obiektom, które zobowiązane są do realizacji fragmentów złożonego przepływu, tory
- W tym zastosowaniu ważną rolę odgrywają przepływy obiektów (jeżeli w przepływie czynności biorą udział istotne obiekty, należy umieścić je na diagramie, pokazując zmieniające się atrybuty i stany tych obiektów)

Przykład - Klient zwraca towar



Modelowanie operacji

- Spełniają rolę schematów blokowych, na których przedstawia się szczegóły przeprowadzanych obliczeń
- Istotne są tu rozgałęzienia, rozwidlenia i scalenia
- Otoczenie diagramu obejmuje parametry operacji i jej obiekty lokalne
- Podstawową zaletą jest to, że wszystkie byty występujące na diagramie są znaczeniowo powiązane z pełnym modelem

Stosowanie diagramu czynności do modelowania operacji ma sens jedynie w przypadku skomplikowanych algorytmów, które trudniej jest zrozumieć na podstawie kodu

Przykład - algorytm operacji *Punkt przecięcie(p:Prosta)*