

# Inżynieria oprogramowania

Wykład 4:

## UML: Diagramy interakcji i diagramy stanów

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



Wersja wykładu 1.32

## Dynamiczne aspekty systemu

- **Interakcja** - zachowanie polegające na wymianie komunikatów między obiektami w pewnym (ustalonym) otoczeniu, w pewnym (ściśle określonym) celu
- **Komunikat** - specyfikacja łączności między obiektami, uwzględniająca zlecenia wykonania określonych czynności
- Interakcja jest wykorzystywana do modelowania dynamiki kooperacji reprezentujących zestawy obiektów odgrywających specyficzne role, współdziałających w celu wywołania pewnego zachowania, niemożliwego do zrealizowania w pojedynkę
- Każda interakcja może być modelowana na 2 sposoby, podkreślając:
  - kolejność komunikatów w czasie lub
  - następstwa komunikatów w otoczeniu pewnej organizacji strukturalnej obiektów

Inżynieria oprogramowania (Wyk. 4)

Slajd 2 z 29

## Diagramy interakcji

Diagram interakcji - obrazuje interakcję jako zbiór obiektów i związków między nimi, w tym też komunikaty, które obiekty przekazują między sobą; zawierają na ogół obiekty, wiązania oraz komunikaty; cel: modelowanie przepływu sterowania

Diagram **współpracy, komunikacji, kooperacji** (ang. collaboration) - uwypukla związki strukturalne pomiędzy obiektami (organizacja strukturalna) wysyłającymi i odbierającymi komunikaty; graficznie jest to zestaw wierzchołków i krawędzi

- nacisk na związki strukturalne między egzemplarzami uczestniczącymi w interakcji oraz komunikaty przesyłane między nimi
- wygodniejsze do przedstawiania złożonych iteracji i rozgałęzień
- stosowane przy wielu współbieżnych przepływach sterowania

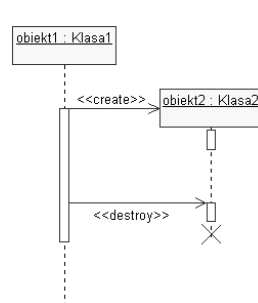
Diagram **przebiegu** (ang. sequence) - uwypukla kolejność komunikatów w czasie; ma postać tabeli, w której obiekty są ułożone wzdłuż osi X a komunikaty wzdłuż osi Y, uporządkowane wg czasu ich wysłania

- nacisk na sposób przekazywania komunikatów w miarę ich pojawiania się
- szczególnie przydatne w kontekście scenariusza przypadków użycia
- wygodniejsze do przedstawiania prostych iteracji i rozgałęzień

Inżynieria oprogramowania (Wyk. 4)

Slajd 3 z 29

## Diagramy przebiegu



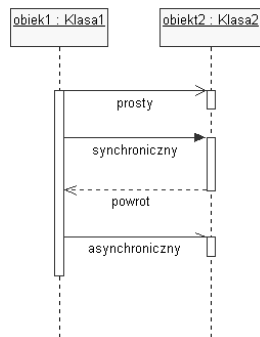
- **Linia życia obiektów** - pionowe przerywane kreski reprezentujące czas istnienia obiektów (zwykle obiekty żyją przez cały czas trwania interakcji, wpp od odebrania komunikatu stereotypowego <<create>> aż do otrzymania <<destroy>>)
- **Ośrodek sterowania** (ang. focus of control) - podłużny, cienki prostokąt reprezentujący okres wykonywania przez obiekt akcji - osobiście lub przez procedury podrzędne; zagnieżdżenia sterowania (np. wywołania rekurencyjne lub wywołania własnych operacji) oznaczane są za pomocą dodatkowego prostokąta ośrodka sterowania umieszczonego na prawo od jego przodka

Inżynieria oprogramowania (Wyk. 4)

Slajd 4 z 29

## Rodzaje komunikatów (do wersji UML 1.5)

- **Prosty** - powoduje jedynie przekazanie sterowania od obiektu do obiektu; na danym etapie modelowania nie są istotne szczegóły tego przekazania
- **Synchroniczny** - obiekt wysyłający komunikat oczekuje na odpowiedź zwrotną i dopiero po jej otrzymaniu przechodzi do dalszych działań; zwykle reprezentuje wywołanie proceduralne
- **Powrót** - oznacza powrót z wywołania procedury; może być pomijany (i najczęściej jest), gdyż jest nieuchronną konsekwencją wywołania
- **Asynchroniczny** - po wysłaniu komunikatu obiekt kontynuuje swoje działania bez oczekiwania na odpowiedź

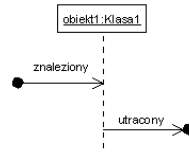


Inżynieria oprogramowania (Wyk. 4)

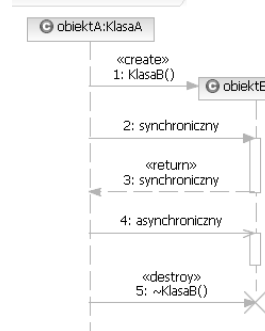
Slajd 5 z 29

## Rodzaje komunikatów (UML 2.0)

- **Znaleziony** - jeżeli nadawca nie jest znany w obrębie danego fragmentu
- **Utracony** - jeżeli odbiorca komunikatu nie jest znany w obrębie danego fragmentu



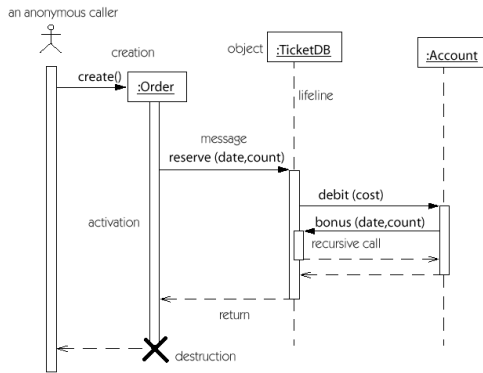
### Rodzaje komunikatów



Inżynieria oprogramowania (Wyk. 4)

Slajd 6 z 29

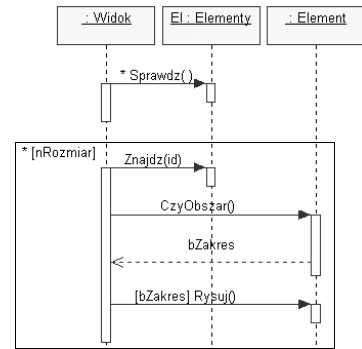
## Diagram przebiegu - przykład (1)



Inżynieria oprogramowania (Wyk. 4) Slajd 7 z 29

## Iteracja (do wersji UML 1.5)

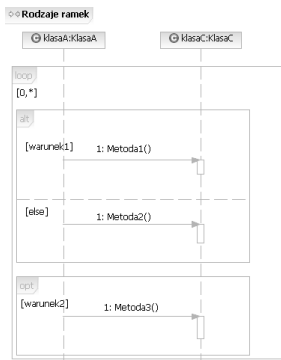
- Stosowana w celu zobrazowania ciągu powtarzających się komunikatów
- Zaznaczana na diagramie wyrażeniem iteracyjnym przed nazwą komunikatu:
  - \* - ("sama gwiazdka" - liczba powtórzeń nie jest określona)
  - \* [t:1..n]
- W sytuacji gdy iteracja dotyczy kilku komunikatów, obejmowane one są w ramkę i wyrażenie iteracyjne podawane jest w lewym górnym rogu



Inżynieria oprogramowania (Wyk. 4) Slajd 8 z 29

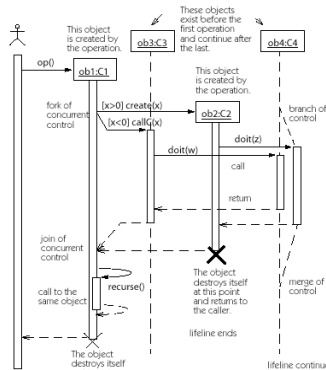
## Ramki (UML 2.0)

- Alt - alternatywa; wykonywany jest ten fragment, przy którym spełniony jest warunek
- Opt - opcjonalny; wykonywany jeśli spełniony jest warunek
- Par - współbieżność; wszystkie fragmenty wykonywane współbieżnie
- Loop - iteracja; warunek określa podstawę iteracji
- Ref - odwołanie; stanowi referencję do innej interakcji; przykrywa linie życia
- Critical (region) - obszar krytyczny; fragment może mieć tylko jeden wątek uruchomiony w danej chwili
- Neg, Break, Assert, Consider...



Inżynieria oprogramowania (Wyk. 4) Slajd 9 z 29

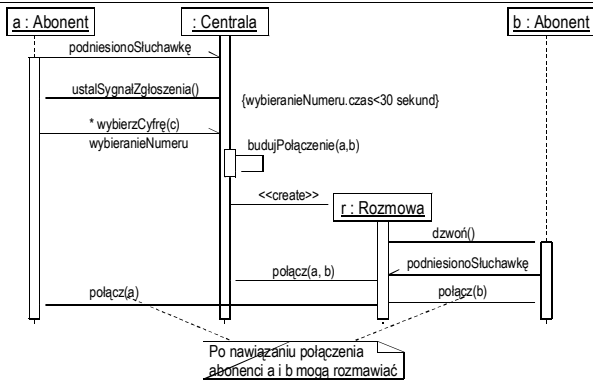
## Rozgałęzienia



- Wykonanie komunikatu uzależnione jest od spełnienia warunku logicznego; na diagramie oznaczone w postaci klauzuli (np. [x>0]) przed numerem komunikatu
- Postać wyrażenia nie jest sprecyzowana, może to być dowolne wyrażenie logiczne, którego wartość jest wyznaczana w momencie wysłania komunikatu
- Nie wszystkie narzędzia umożliwiają wykorzystanie rozgałęzień

Inżynieria oprogramowania (Wyk. 4) Slajd 10 z 29

## Diagram przebiegu - przykład (2)



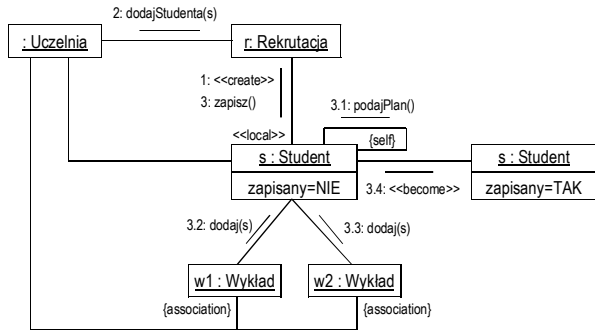
Inżynieria oprogramowania (Wyk. 4) Slajd 11 z 29

## Diagramy współpracy

- Cechy wyróżniające:
  - występowanie ścieżek - dodajemy stereotypy ścieżki (np. local, parameter, global, ...) do drugiego końca wiązania
  - ciągi komunikatów - aby wskazać kolejność komunikatu w czasie, poprzedzany jest on numerem w ciągu; zagnieżdżenia obrazuje się za pomocą notacji Deweya (1, 1.1, 1.2, 2, ...)

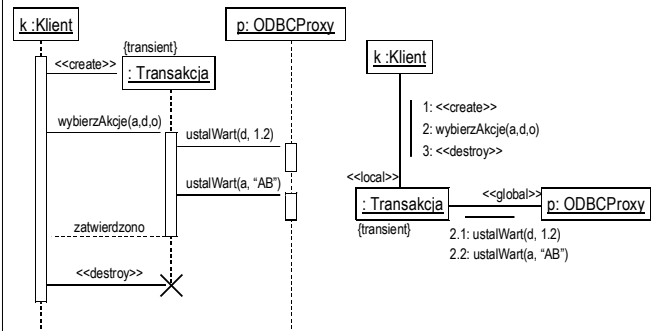
Inżynieria oprogramowania (Wyk. 4) Slajd 12 z 29

## Diagram współpracy - przykład

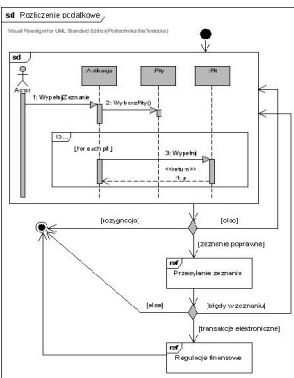


## Równoważność znaczeniowa (przykład)

Diagram przebiegu Diagram współpracy



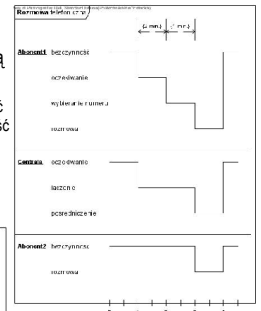
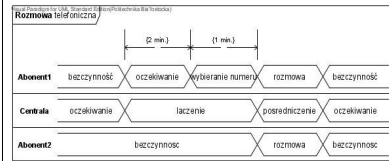
## Diagram przeglądu interakcji (UML 2.0)



- Połączenie notacji diagramów przebiegu z notacją diagramów czynności
  - diagram ma strukturę diagramu czynności, przy czym zamiast elementów czynności występują elementy interaktywne z d. przebiegu (lub całe diagramy przebiegu)
- Zastosowanie: modelowanie systemu na wysokim poziomie abstrakcji; przedstawienie połączeń pomiędzy scenariuszami modelowanego systemu

## Diagram czasowy (UML 2.0)

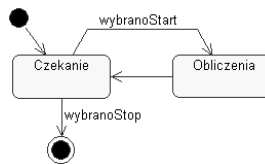
- Modelowanie wielu przebiegów czasowych
- Wykorzystuje dwa rodzaje linii czasu (obie rosną wzdłuż osi X):
  - linia wartości (ang. *value lifeline*) – wskazuje wartość jaką przyjmuje rozpatrywany parametr obiektu (wartość określana **tylko** za pomocą tekstu)
  - linia stanu (ang. *state lifeline*) – wskazuje stan obiektu wyrażony za pomocą wartości dyskretnych; odpowiadają im poziomy (różne amplitudy) linii



## Diagram stanów (zmiany stanów)

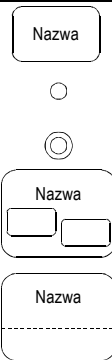
## Maszyny stanowe - podstawowe pojęcia

- **Maszyna stanowa** - określa ciąg stanów przyjmowanych przez obiekt w odpowiedzi na zdarzenia zachodzące w czasie jego życia, a także reakcje na te zdarzenia; bardzo przydatne, gdy bieżące zachowanie obiektu zależy od jego przeszłości
- **Zdarzenie** - specyfikacja zjawiska, które zachodzi w czasie i przestrzeni; jest bodźcem, który może uruchomić przejście pomiędzy stanami
- **Stan** - okoliczność lub sytuacja, w jakiej się obiekt znajduje, kiedy spełnia jakiś warunek, wykonuje jakąś czynność lub czeka na jakieś zdarzenie; zwykle obiekt pozostaje w pewnym stanie przez skończony czas
- **Przejście** - związek między dwoma stanami; wskazuje, że obiekt znajdujący się w pierwszym stanie wykona pewne akcje i przejdzie do drugiego stanu, o ile zajdzie określone zdarzenie i będą spełnione określone warunki
- **Akcja** - wykonywalna niepodzielna procedura obliczeniowa prowadząca do zmiany stanu systemu lub do przekazania wartości



## Rodzaje stanów

- **Prosty** (zwykły) - nie posiada wewnętrznej struktury
- **Początkowy** - pseudostan oznaczający punkt startowy - miejsce rozpoczęcia działania maszyny stanowej lub podstanu
- **Finalny** - pseudostan oznaczający punkt końcowy
- **Złożony sekwencyjny** - złożony z jednego lub więcej podstanów, z których tylko jeden jest aktywny, gdy aktywny jest stan złożony
- **Złożony współbieżny** - podzielony na dwa lub więcej współbieżnych podstanów; wszystkie podstany są jednocześnie aktywne, gdy aktywny jest stan złożony (jako całość)

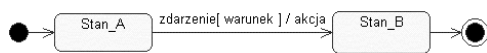


## Składniki stanów

- **Nazwa** - odróżnia poszczególne stany (nie może zawierać dwukropka); może też być stan bez nazwy - anonimowy
- **Akcje wejściowe i wyjściowe** - wykonywane odpowiednio przy wejściu do stanu i przy wyjściu z niego
- **Przejścia wewnętrzne** - realizowane bez zmiany stanów
- **Podstany** - zagnieżdżona struktura stanu, obejmująca podstany rozłączne (aktywne sekwencyjne) lub współbieżne (aktywne równoległe)
- **Zdarzenia odroczone** - lista zdarzeń, które nie są w tym stanie obsługiwane, są odkładane i umieszczane w kolejce, po czym są obsługiwane w innym stanie

Nazwę stanu podaje się w formie rzeczownika lub wyrażenia rzeczownikowego pochodzącego ze słownictwa modelowanego systemu

## Składniki przejść



### zdarzenie\_uruchamiające [warunek\_dozoru] / akcja

- **Stan źródłowy** - gdy obiekt jest w stanie źródłowym i nastąpi zdarzenie uruchamiające, a warunek dozoru (o ile istnieje) jest spełniony, przejście może być uruchomione; do czasu aż przejście nie zostanie uruchomione, obiekt pozostaje w stanie źródłowym
- **Akcja** - wykonywalna niepodzielna procedura obliczeniowa, która może mieć bezpośredni wpływ na obiekt będący właścicielem maszyny stanowej i pośredni wpływ na inne obiekty znajdujące się w jego zasięgu
  - może to być wywołanie operacji (obiektu lub innych dostępnych obiektów), utworzenie lub zniszczenie obiektu oraz wysłanie sygnału do obiektu
  - nie może być przerwana przez zdarzenie - zawsze jest wykonywana w całości
- **Stan docelowy** - stan obiektu po zakończeniu przejścia

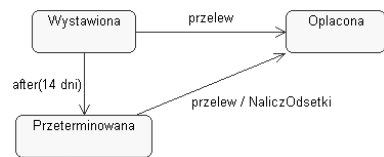
## Zdarzenia uruchamiające

**Zdarzenie uruchamiające** (ang. *event trigger*) - zdarzenie oznaczające, że przejście może nastąpić ( sygnał, wywołane operacji, upływ czasu i zmiana stanu)

- sygnał lub wywołanie może mieć parametry, których wartości są dostępne w ramach przejścia
- przejścia automatyczne (bez zdarzenia uruchamiającego) - przejścia (tzw. *zakończeniowe*) uruchamiane są natychmiast po zakończeniu czynności w stanie źródłowym

• zdarzenia czasowe:

- **after** (okres\_czasu)  
np. after(3 miesiące),  
after(1 godzina)
- **when** (moment\_czasu)  
np. when(31.12),  
when (godzina 22.00)

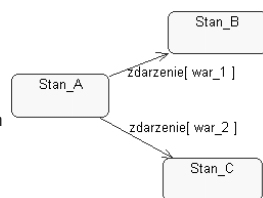


Faktura

## Składniki przejść (2)

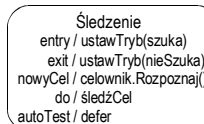
**Warunek dozoru** (ang. *guard condition*) - wyrażenie logiczne, którego wartość jest wyznaczana w chwili otrzymania zdarzenia uruchamiającego i spełnienie go warunkuje dokonanie przejścia

- obrazowany w postaci wyrażenia logicznego w nawiasach kwadratowych tuż za zdarzeniem uruchamiającym
- można określić wiele przejść z tego samego źródła i z tym samym zdarzeniem uruchamiającym o ile warunki dozoru się nie nakładają
- zdarzenie uruchamiające jest ignorowane, gdy żadne przejście przez nie inicjowane nie może dojść do skutku
- warunek dozoru może występować bez zdarzenia uruchamiającego => zmiana wartości wyrażenia skutkuje umożliwieniem przejścia



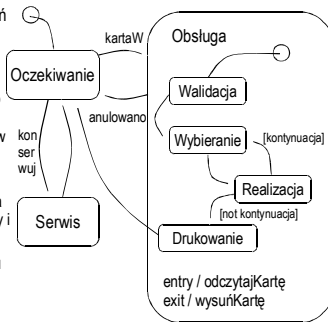
## Złożone elementy stanów i przejść

- **Akcje wejściowe (entry) i wyjściowe (exit)** - wykonywane zawsze, gdy przyjmowany / opuszczany jest stan; w zasadzie nie mogą mieć argumentów ani warunków dozoru
- **Przejścia wewnętrzne** - do zdarzeń obsługiwanych bez wyjścia ze stanu; ilekroć zachodzi zdarzenie to skojarzona z nim akcja jest realizowana bez opuszczania stanu (nie są wykonywane akcje wyjściowe i wejściowe)
- **Czynności** - będąc w jakimś stanie obiekt może realizować pewne zadania aż do chwili zajścia zdarzenia; słowo kluczowe "do" służy do wskazania prac wykonywanych w danym stanie od chwili zakończenia akcji wejściowej (może to być uruchomienie innej maszyny stanowej lub ciąg akcji, rozdzielonych średnikami)
- **Zdarzenia odroczone** - lista zdarzeń, których zajście w stanie jest odroczone do czasu, aż stan, w którym nie są odroczone, się uaktywni - dopiero wtedy te zdarzenia powodują przejścia, jakby właśnie zaszyły; zdarzenia takie oznaczają się specjalnym rodzajem akcji **defer**



## Podstany sekwencyjne

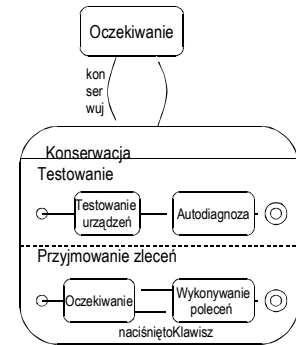
- Podstany sekwencyjne dzielą przestrzeń stanów stanu złożonego na stany rozłączne
- Przejście od źródła znajdującego się na zewnątrz może prowadzić do stanu złożonego (wtedy musi być określony stan początkowy) lub bezpośrednio do jednego z jego podstanów
- Niezależnie od przyjętego rozwiązania akcja wejściowa stanu złożonego jest realizowana
- Zagnieżdżona sekwencyjna maszyna stanowa może mieć co najwyżej jeden stan początkowy i jeden stan końcowy
- Aby zapamiętać aktualny podstan obiektu po opuszczeniu używamy **stanów wznowienia** (płytkie i głębokie)



(H) (H\*)

## Podstany współbieżne

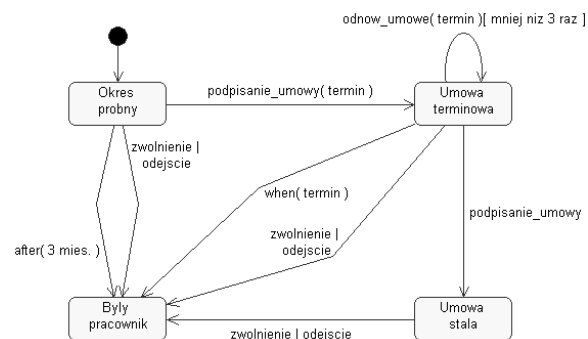
- Umożliwiają tworzenie dwu lub więcej maszyn stanowych, działających równolegle w ramach jednego obiektu
- Czynności podstanów współbieżnych przebiegają równolegle i ostatecznie każda z nich osiąga swój stan końcowy; jeżeli jeden dotrze do swego stanu końca wcześniej to czeka na pozostałe i dopiero wtedy następuje połączenie przepływu sterowania
- Zagnieżdżona współbieżna maszyna stanowa nie ma jako całość ani stanu początkowego ani końcowego; natomiast mogą je mieć składowe podstany sekwencyjne



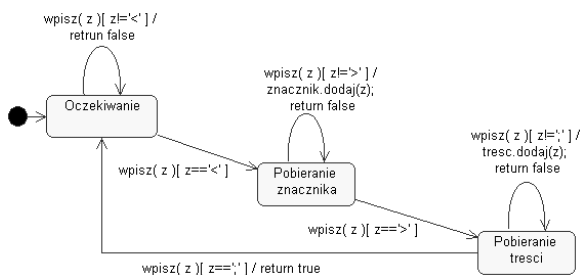
## Diagramy stanów

- Przedstawiają maszyny stanowe z uwypukleniem przepływu sterowania między stanami; pokazują jak zachowanie obiektów zależy od kolejności zachodzących zdarzeń
- Zawierają:
  - stany zwykłe (proste) i złożone
  - przejścia ze zdarzeniami i akcjami
- Najczęściej wykorzystywane do modelowania obiektów reaktywnych (sterowanych zdarzeniami - ang. *event-driven*)
  - zachowanie obiektów reaktywnych jest najlepiej charakteryzowane przez ciąg odpowiedzi na zdarzenia wywołane w jego otoczeniu, przy czym obiekt taki jest zwykle beczny do chwili zajścia zdarzenia
  - reakcja na konkretne zdarzenie najczęściej zależy od wcześniejszych zdarzeń
  - nacisk kładziony jest na stany stabilne, zdarzenia uruchamiające przejścia i akcje wykonywane po każdej zmianie stanu

## Diagram stanów - przykład (ścieżka zatrudnienia pracownika)



## Diagram stanów - przykład (2) (analyzer składniowy języka bezkontekstowego)



Akceptuje komunikaty w postaci: '< napis\_1 > napis\_2';  
gdzie napis\_1 - znacznik a napis\_2 - treść komunikatu