Model-Driven Engineering

Radosław Klimek

2015-22

http://home.agh.edu.pl/rklimek

Diagram stanów (maszyny stanowe)

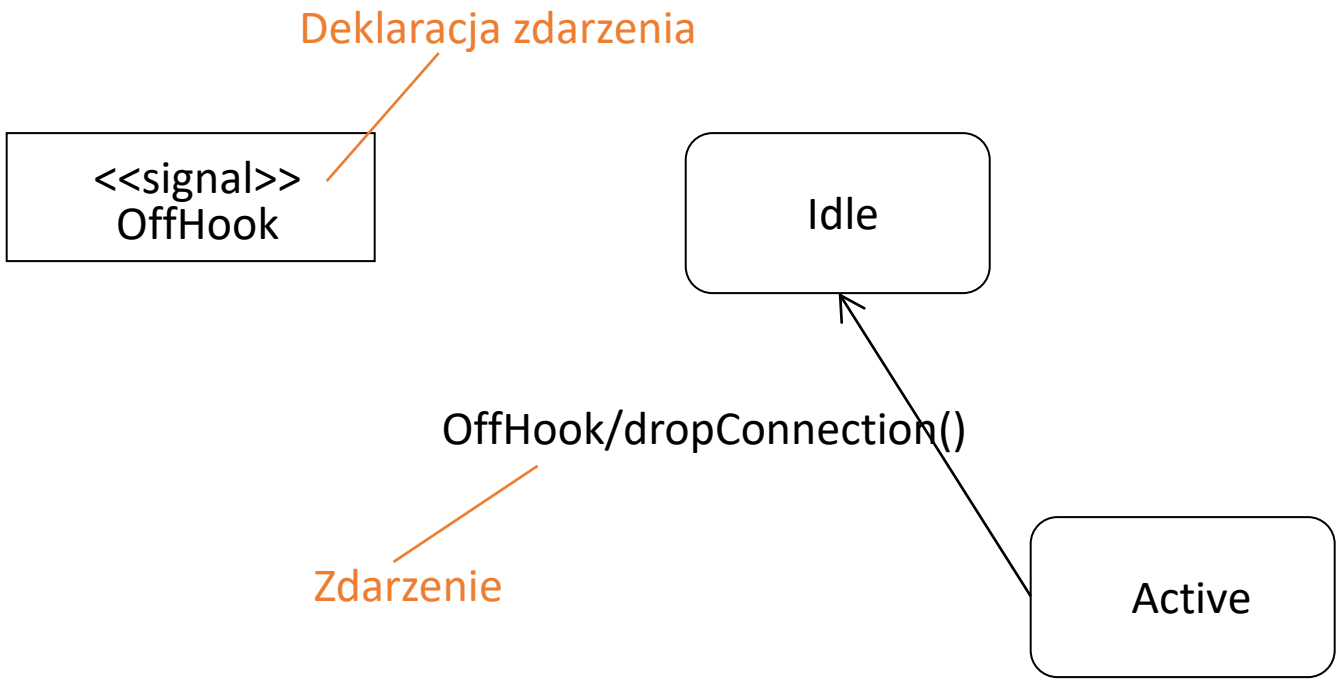
Zdarzenie

Zdarzenie jest specyfikacja zjawiska, które zachodzi w czasie i przestrzeni. W kontekście maszyny stanowej zdarzenie jest wystąpieniem bodźca, które może uruchomić przejście między stanami.

Zdarzenie ma (być może pustą) listę parametrów specyfikujących informację przekazywaną między elementem generującym zdarzenie a jego odbiorcą. Moment czasowy wystąpienia może być parametrem zdarzenia.

Wystąpienie (instancja) zdarzenia charakteryzuje się aktualnymi wartościami poszczególnych parametrów. Wartości te są dostępne dla akcji przypisanej do przejścia wyzwalanego przez to zdarzenie.

Zdarzenie – sygnał



Rodzaje zdarzeń

1. **Sygnały** (*Signals*).
2. **Wywołania** (*call events*).
3. **Upływ czasu** (*time events*).
4. **Zmiana stanu** (*change events*).

Ponadto mam następujący podział:

1. **zdarzenia zewnętrzne** – zachodzą między systemem a aktorami,
2. **zdarzenia wewnętrzne** – między obiektami wewnątrz systemu.

Sygnał

Sygnał jest rodzajem zdarzenia i reprezentuje nazwany obiekt, asynchronicznie wysyłany przez jeden obiekt i odbierany przez drugi.

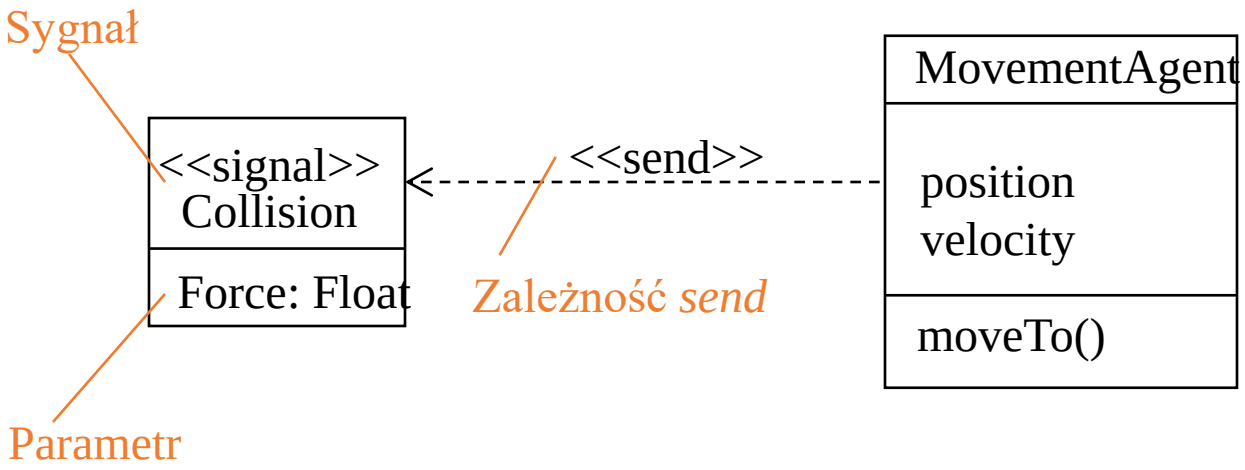
Podobnie jak klasy – sygnały mogą mieć egzemplarze, mogą występować w uogólnieniach, co umożliwia modelowanie hierarchii zdarzeń, np. *AwariaSieci* (zdarzenie ogólne) i *AwariaSerweraPrzedsiębiorstwa* - rodzaj zdarzenia *AwariaSieci.*

Sygnał – realizacja

Sygnał może być wysłany w wyniku:

- realizacji przejścia w maszynie stanowej,
- przekazania komunikatu w interakcji,
- wykonania operacji.

Sygnały i wyjątki są modelowane w postaci stereotypowanych klas.
Zależność stereotypowana <<send>> jest stosowana w celu specyfikowania wysłania sygnału przez operację.



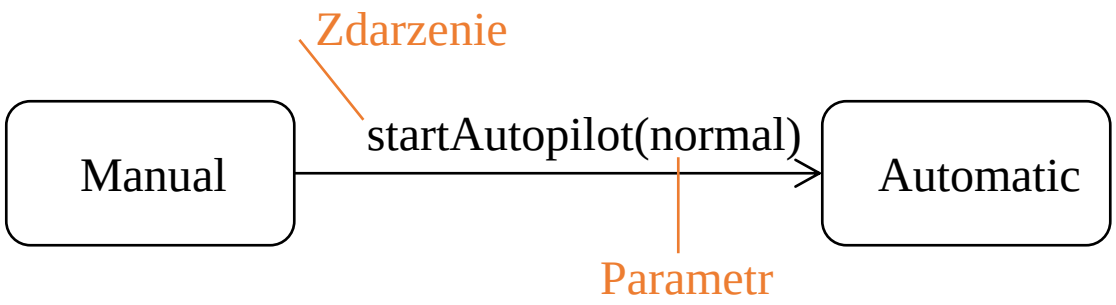
Wywołanie

Wywołanie jest zazwyczaj zdarzeniem synchronicznym.

Jeśli obiekt wywołuje operację innego obiektu, z którym jest skojarzona maszyna stanowa, sterowanie jest przekazywane od nadawcy do odbiorcy, dochodzi do uruchomienia przejścia, operacja zostaje zakończona, odbiorca przechodzi do nowego stanu, a sterowanie wraca do nadawcy.

Zdarzenia wywołania i sygnałowe nie są rozróżniane na diagramie stanów. Sygnał jest jednak zazwyczaj „obsługiwany” przez maszynę stanową, a wywołanie przez metodę.

Wywołanie – przykład



Upływ czasu i zmiana stanu

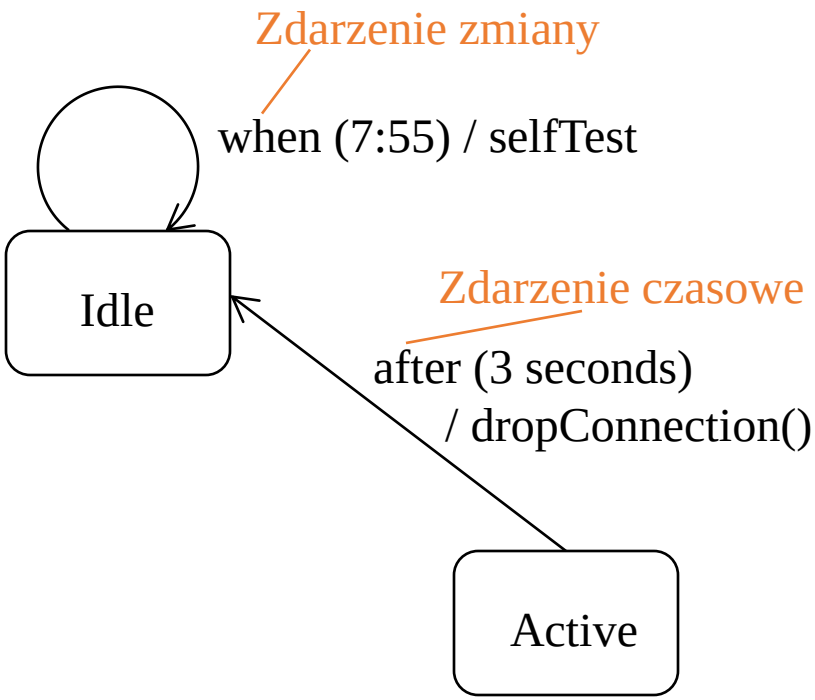
Zdarzenie czasowe reprezentuje upływ czasu. Jest specyfikowane przez słowo kluczowe *after* i wyrażenia, którego wartością jest przedział czasowy.

Domyślnie przyjmuje się, że momentem początkowym przy obliczaniu wyrażenia jest czas (moment) wejścia do aktualnego stanu.

Zdarzenie wywołania reprezentuje zmianę stanu lub spełnienie pewnego warunku.

Modeluje się je przez słowo kluczowe *when* i wyrażenie logiczne, które można użyć do zapisania czasu bezwzględnego (np. *when time* = 7:55), lub nieustannego sprawdzania warunku (np. *when altitude* < 500).

Upływ czasu i zmiana stanu – przykład

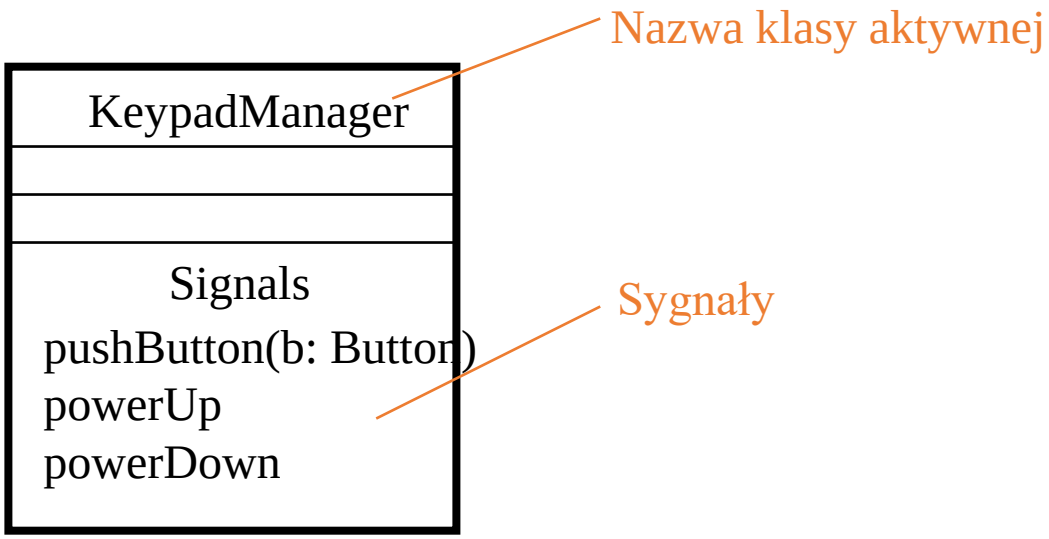


Wysyłanie i odbieranie zdarzeń

• Po wysłaniu sygnału nadawca nadawca (aktor, egzemplarz klasy) kontynuuje swoje czynności (gdy komunikacja asynchroniczna).
• Obiekt wywołujący operację czeka na odpowiedź odbiorcy, zakończenie operacji (gdy komunikacja synchroniczna).
• Egzemplarz klasy może odebrać wywołanie lub sygnał. Zawsze, jeśli zdarzenie wywołania jest synchroniczne, wówczas dochodzi do spotkania nadawcy i odbiorcy, które trwa do zakończenia operacji.
• W przeciwnym przypadku, po wysłaniu sygnału nie dochodzi do spotkania - nadawca wysyła sygnał i nie czeka na odpowiedź.
• W obu przypadkach zdarzenie może być zignorowane (jeśli nie określono odpowiedzi), może uruchomić przejście w maszynie stanowej (jeśli istnieje), lub może spowodować klasyczne wywołanie metody.

Klasa aktywna a sygnały

Zdarzenia wywołania, które obiekt musi odebrać, są modelowane w postaci operacji w klasie. Nazwane sygnały, które obiekt może odebrać są specyfikowane w dodatkowej sekcji symbolu klasy.



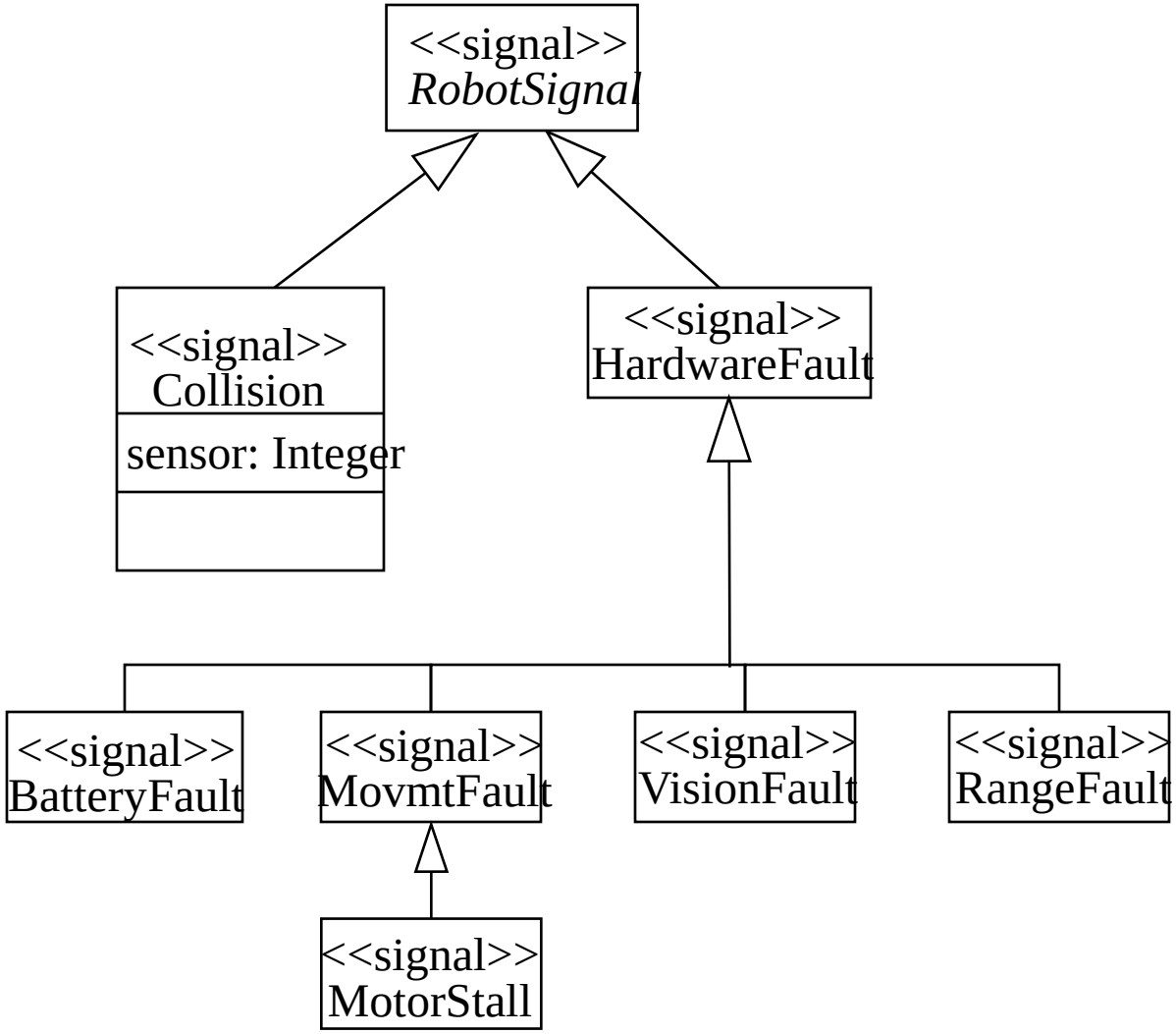
Modelowanie rodziny sygnałów

Należy pamiętać, że w większości modelowanych systemów sterowanych zdarzeniami sygnały tworzą hierarchię.

W modelowaniu hierarchii sygnałów można określić zdarzenia polimorficzne.

Przykładowo przejście (polimorficzne) może być uruchamiane przez sygnał *AwariaOsprzętu* lub jego uszczegółowienie: *WyładowanieAkumulatorów*, *AwariaNarządówRuchu*, *ZatarcieSilnika*.

Modelowanie hierarchii sygnałów – przykład



Wytyczne do modelowania sygnałów

1. Rozważ wszystkie rodzaje sygnałów, na które może reagować wybrany zbiór obiektów aktywnych.

2. Wyszukaj podobne rodzaje sygnałów i utwórz hierarchię uogólnień-uszczegółowień.

3. W maszynach stanowych obiektów aktywnych znajdź miejsca, gdzie można skorzystać z polimorfizmu. Jeśli je znajdziesz, skoryguj hierarchię, wprowadzając pośrednie sygnały abstrakcyjne.

Modelowanie wyjątków

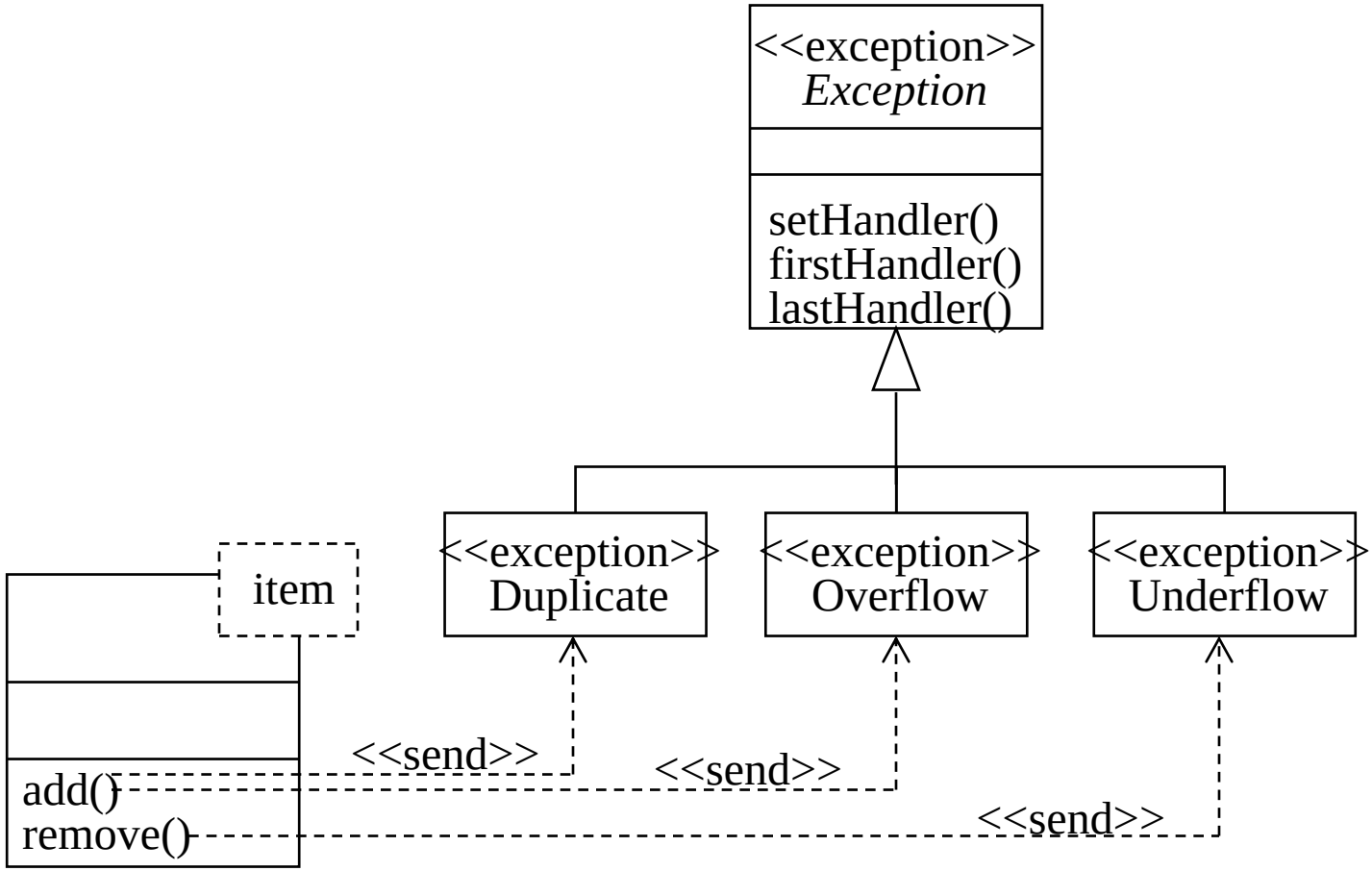
W UML wyjątki są rodzajem sygnałów i modeluje się je z zastosowaniem stereotypowanych klas. Wyjątki mogą być skojarzone ze specyfikacjami operacji.

Modelowanie wyjątków jest pewnym przeciwieństwem w stosunku do sygnałów.

Modeluje się bowiem rodzinę sygnałów odbieranych przez obiekt aktywny.

Natomiast w kontekście wyjątków modeluje się wyjątki zgłaszane przez operacje obiektu aktywnego.

Modelowanie wyjątków – przykład



Wytyczne do modelowanie wyjątków

1. Dla każdej klasy i interfejsu, a także ich operacji podaj sytuacje wyjątkowe.

2. Opracuj hierarchię wyjątków - od ogólnych do bardziej szczegółowych.

3. Określ wyjątki zgłaszane przez poszczególne operacje. Możesz uczynić to jawnie (*send* od operacji do jej wyjątków), możesz je zapisać w specyfikacji.

Diagram stanów – definicja

Diagram stanów (maszyna stanowa) określa ciąg stanów przyjmowanych przez obiekt i odpowiedzi w reakcji na zdarzenia zachodzące w trakcie życia obiektu.

Diagram stanów jest to **graf**, który reprezentuje **maszynę stanów**.

Podstawowymi elementami diagramu są stany obiektu połączone strzałkami przejść. Obiekt, reagując na nadchodzące zdarzenia jeżeli spełnione są określone warunki, zmienia swój stan i położenie na diagramie stanu.

Tak więc diagram składa się z:
Stan obiektu (wierzchołki grafu) to okoliczność lub sytuacja, w jakiej znajduje się obiekt w trakcie swego życia.
Zdarzenie (krawędzie grafu) to specyfikacja zjawiska, które zachodzi w czasie i przestrzeni.

W przypadku maszyny stanowej zdarzeniem jest wystąpieniem bodźca wyzwalającego przejście.

Diagram stanów – uwagi

Jest to diagram używany przy analizie i projektowaniu oprogramowania. Opisuje zachowanie systemu podsystemu lub jego elementu – obiektu a w szczególności zmiany tego zachowania.

Pokazuje wszystkie możliwe stany systemu (obiektu) oraz przejścia, które powodują zmianę tego stanu.

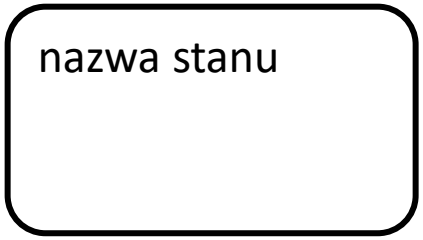
Można z niego odczytać, jakie sekwencje sygnałów wejściowych powodują przejście systemu (obiektu) w dany stan, a także jakie akcje są podejmowane w odpowiedzi na pojawienie się określonych stanów wejściowych. Pokazuje on cykl życia obiektu.

Diagram stanów jest szczególnie przydatny, gdy zachowanie obiektu jest złożone. Przedstawia on maszynę stanów jako przepływ sterowania między stanami.

Stan

Stan obiektu – w podstawowym znaczeniu – dotyczy pewnego fragmentu historii życia obiektu i opisuje pewien aspekt tej historii. Poniżej przedstawiamy trzy najbardziej popularne definicje stanu obiektu:

Stan obiektu jest określony przez aktualne powiązania i wartości atrybutów obiektu;
Stan to okres czasu, w którym obiekt oczekuje na zdarzenie;
Stan to okres czasu, w którym obiekt przetwarza pewne dane.



symbol stanu

Stan – uwagi

Stan określa jak obiekt reaguje na zdarzenia, informacja zależna od poprzednich czynności wykonanych przez obiekt określony wartością atrybutów obiektu i jego powiązań z innymi obiektami

Stan reprezentuje określone warunki, w jakich znajduje się obiekt lub status, jaki może on posiadać.

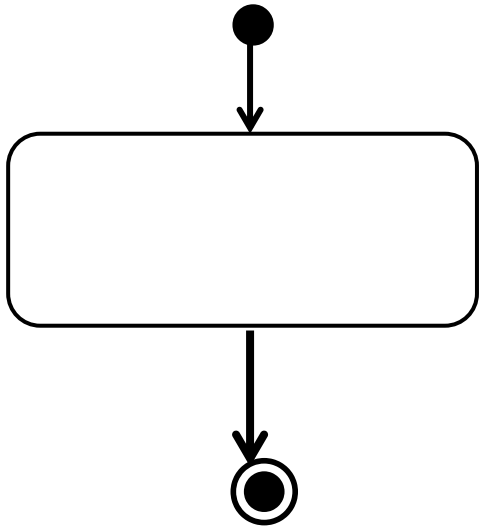
Stany wykluczają się nawzajem – obiekt może być w określonym czasie tylko w jednym stanie.

Każdy obiekt może mieć dowolną ilość stanów.

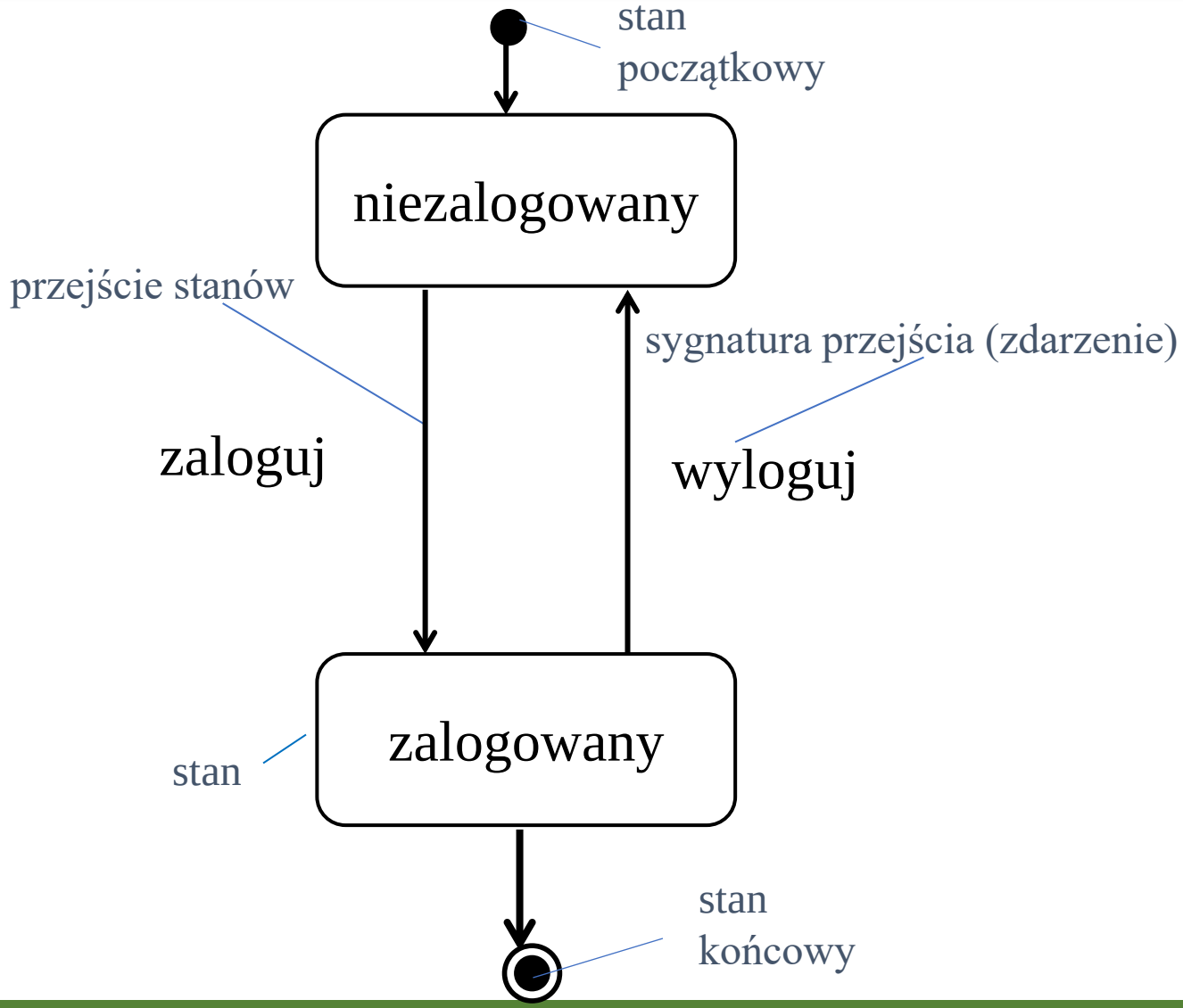
Rodzaje stanów

Każdy obiekt powinien posiadać:

- stan początkowy
- stany pośrednie
- stan końcowy



Elementarny przykład



Koncepcja pseudostanu

Jedynym zadaniem **pseudostanu** jest wskazanie na inny stan, przy czym rodzaj pseudostanu określa informację, która jest w ten sposób przedstawiona.

Najbardziej popularnym pseudostanem jest stan początkowy.

Pseudostan - przykłady

- Stan początkowy
- Stan końcowy
- Decyzja – węzeł wyboru
- Wznowienie płytkie
- Wznowieni głębokie
- Rozwidlenie
- Scalenie
- Węzeł
- Odnośnik

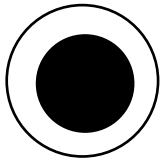
Stan początkowy

- stan początkowy jest pseudostanem.
- umożliwia jedynie zainicjowanie maszyny stanowej, czyli przejście ze stanu początkowego do stanu pierwszego.
- nie posiada wszystkich atrybutów właściwych stanowi.
- zapewnienia połączenia pomiędzy stanami, bez wymogu akcji.
- nie może zawierać zdarzenia uruchamiającego.
- może być tylko jeden stan początkowy

symbol stanu początkowego

Stan końcowy

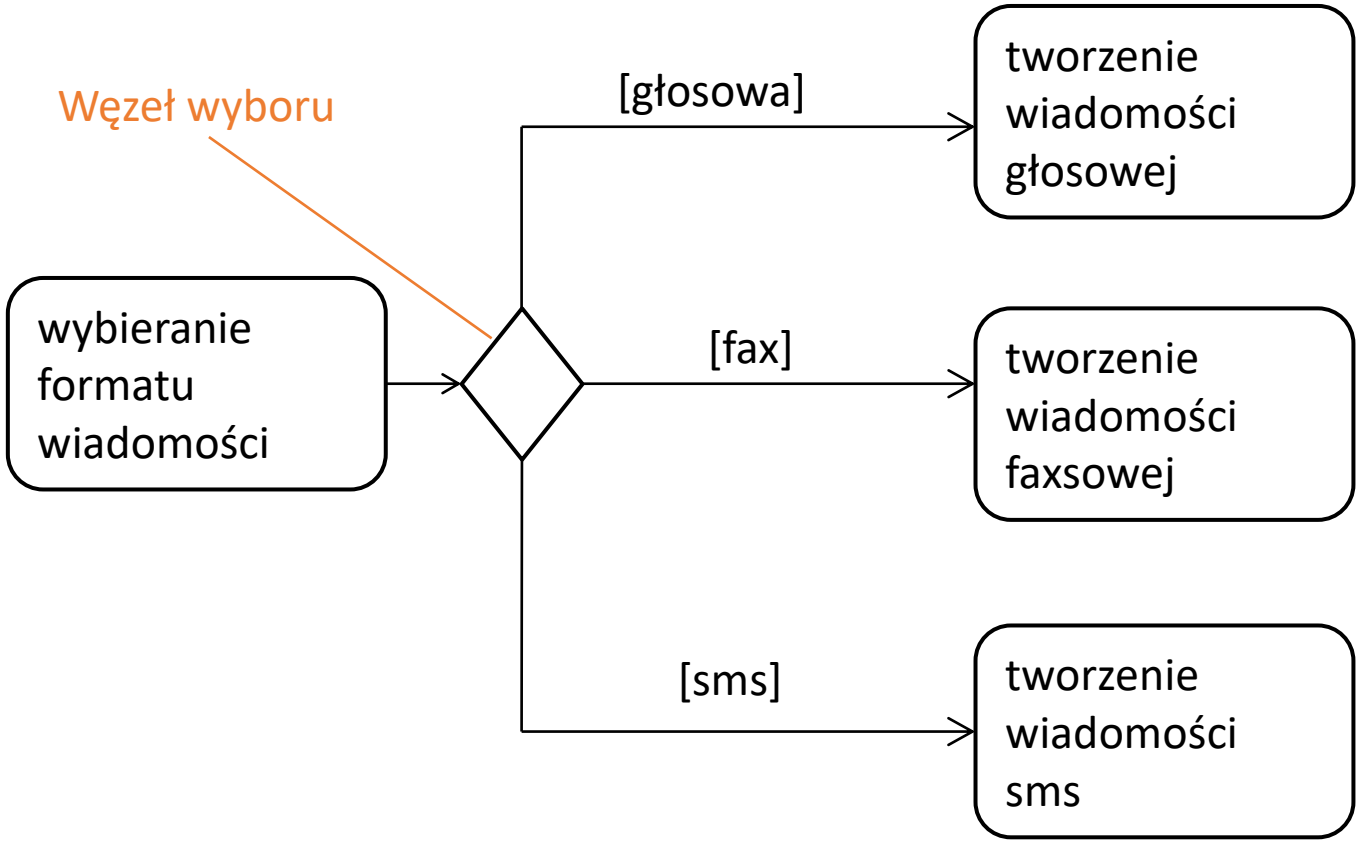
• stan końcowy jest pseudostanem.
• stan końcowy jest to stan, jaki jest przyjmowany w sytuacji zakończenia działania maszyny stanowej.
• stan specjalny – nie ma z niego przejścia do następnego stanu.
• może być kilka stanów końcowych



symbol stanu końcowego

Węzeł wyboru

Węzeł wyboru – to element, który umożliwia dokonanie wyboru spośród kilku możliwości.

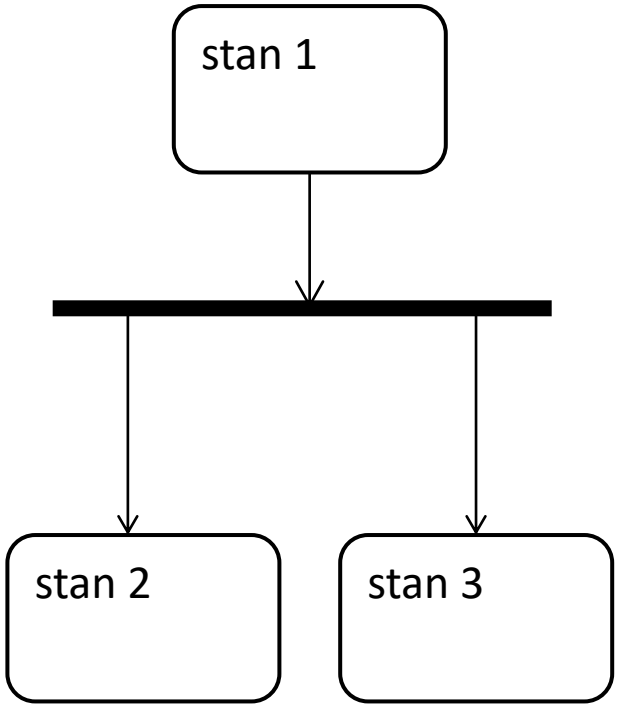


Węzeł wyboru – uwagi

• Umieszczenie węzła wyboru na diagramie oznacza, że nie ma jednej ścieżki wykonywania poszczególnych czynności (istnieją ścieżki alternatywne).
• Węzeł wyboru może mieć jedno wejście dla przepływu sterującego i minimalnie dwa wyjścia przepływów.
• Na wyjściach z węzła wyboru znajdują się wykluczające się warunki dozoru. Istotne jest, by warunki dozoru uwzględniały wszystkie możliwości, tak by nie dopuścić do zatrzymania przepływu na węźle.
• Język UML dopuszcza, by jedno wyjście z węzła było opisane warunkiem [else], który oznacza ścieżkę, jaka powinna być wybrana w przypadku niespełnienia warunku dozoru wszystkich pozostałych wyjść.

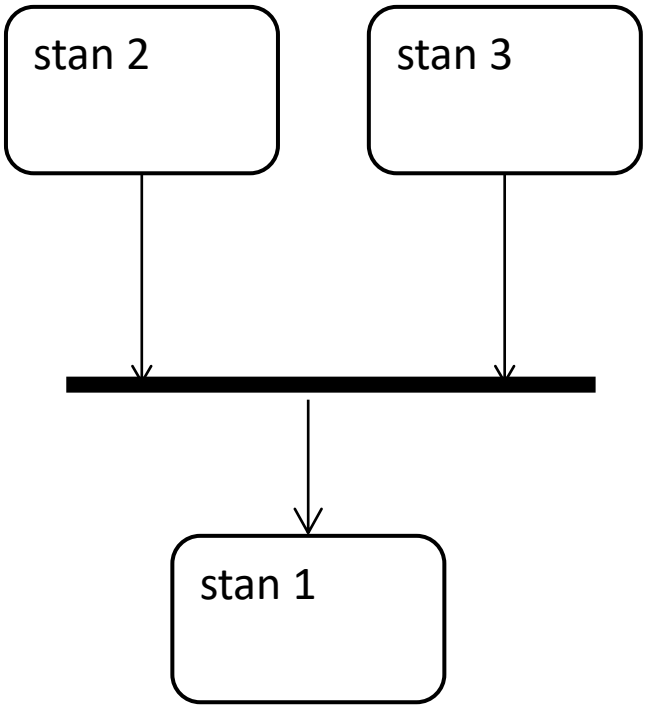
Rozwidlenie

Rozwidlenie - pozwala na rozdzielenie wejściowego przejścia na dwa lub więcej przejść wyjściowych do różnych obszarów współbieżnych, stanów lub maszyn stanowych.



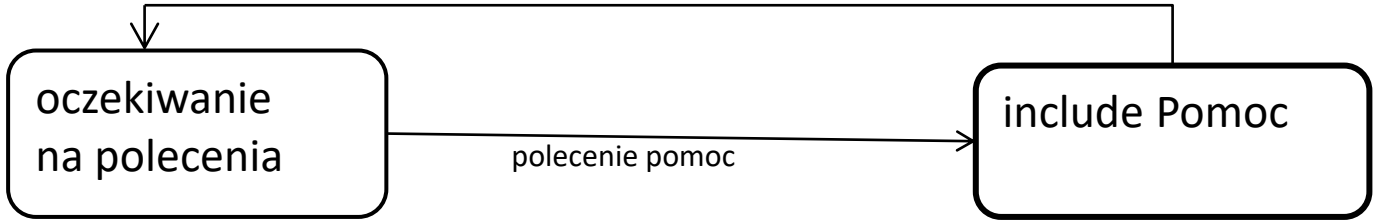
Scalenie

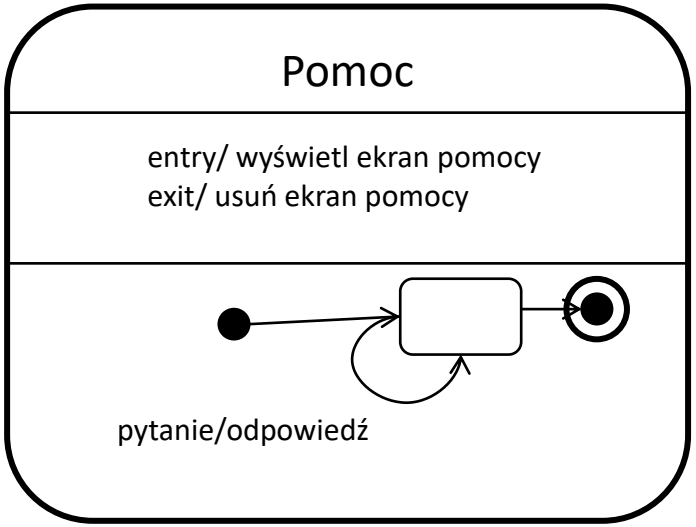
Scalenie - integruje kilka przejść wychodzących z różnych źródłowych obszarów współbieżnych, stanów lub maszyn stanowych.



Pseudostan odnośnikowy

Pseudostan odnośnikowy – odwołanie do innego stanu

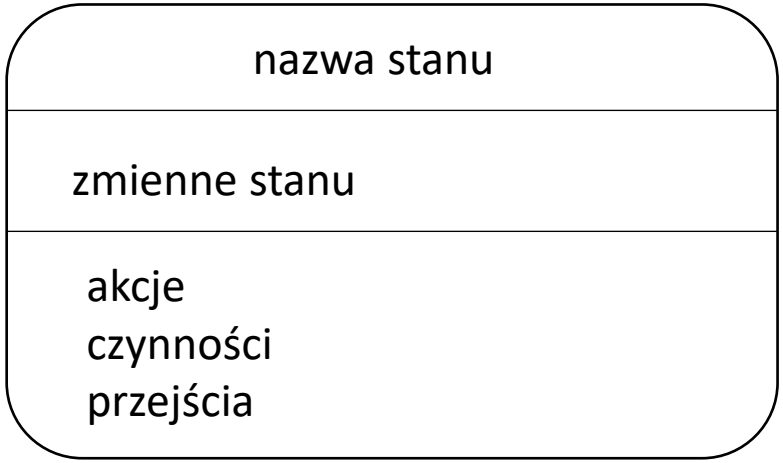




Oznaczenie stanu (w UML)

Opcjonalnie symbol stanu można podzielić na poziome przedziały (opcjonalnie), które opisują:

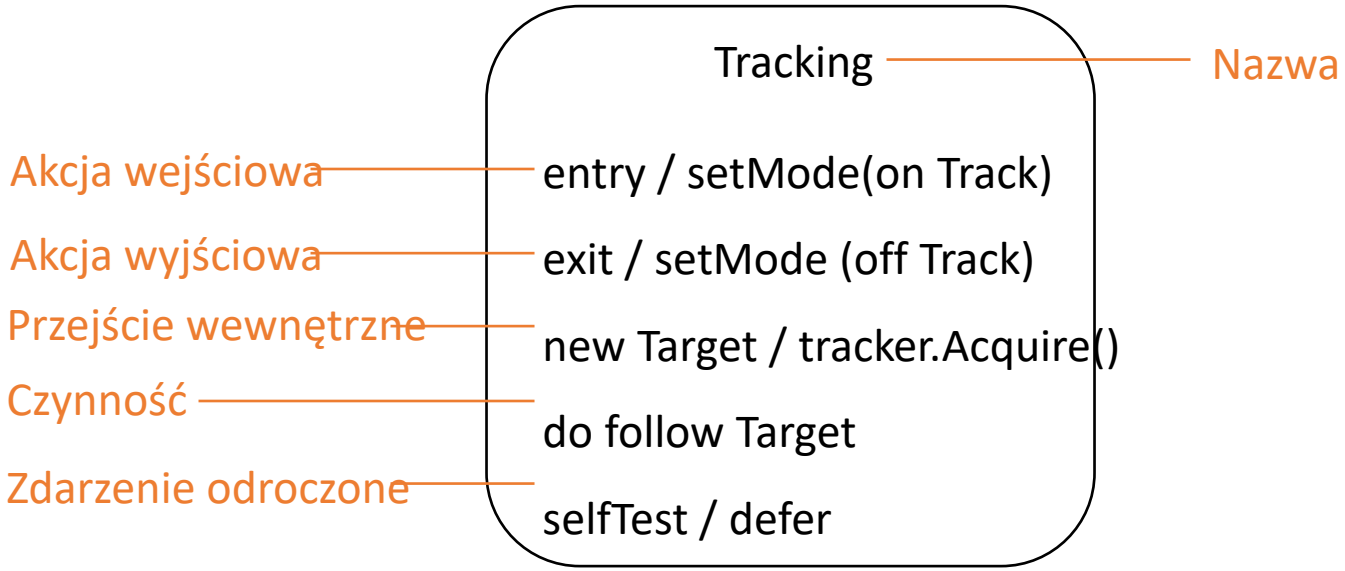
- nazwę stanu
- zmienne stanu
- akcje
- czynności
- przejścia



Składniki stanu

1. **Nazwa** – ciąg znaków identyfikujący stan. Stan anonimowy – bez nazwy.
2. **Akcje wejściowe** i **wyjściowe** – akcje wykonywane przy wejściu do i wyjście z stanu.
3. **Przejścia wewnętrzne** – przejścia realizowane bez zmiany stanu.
4. **Zmienne stanu** – np. liczniki i zegary
5. **Podstany** – zagnieżdżona struktura obejmująca podstany sekwencyjne lub współbieżne. (w przypadku stanów łożonych)
6. **Zdarzenia odroczone** – lista zdarzeń, które nie są w tym stanie obsługiwane, lecz są odraczane i umieszczane w kolejce, o czym są obsługiwane w innym stanie.

Składniki stanu - przykład



Akcje wewnętrzne

entry – akcja wejściowa, akcja wykonywana w momencie gdy obiekt przyjmuje dany stan; akcja ta jest wykonywana jeden raz i niepodzielnie

do – akcja wykonywana w sposób ciągły, nieprzerwanie w czasie, gdy obiekt znajduje się w danym stanie

exit – akcja wyjściowa, akcja wykonywana w momencie wyjścia obiektu ze stanu, akcja taka jest wykonywana tylko raz.

event – akcja wykonywana w momencie nadejścia zdarzenia określonego typu.

include – wywołanie zagnieżdżonej maszyny stanu.

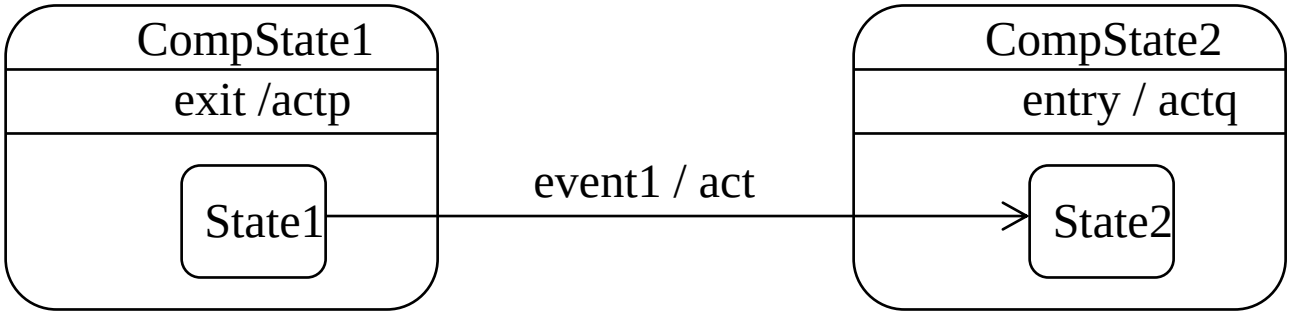
Uwaga: wykonanie każdej z tych akcji może również generować zdarzenie.

Akcje wejściowe i wyjściowe

• Często należy wykonać tę samą akcję ilekroć osiągany jest dany stan, niezależnie od przejścia, które do tego stanu prowadzi.
• Mając do dyspozycji płaskie maszyny należałoby przy każdym takim przejściu umieścić odpowiednie akcje. Może to prowadzić do błędów, np. w przypadku dopisywania dodatkowego przejścia.
•Analogiczne rozumowanie uzasadniające można przeprowadzić dla akcji wyjściowych.

Akcje wejściowe i **wyjściowe** nie mogą mieć argumentów ani warunków dozoru. Jedynie akcja wejściowa najwyższego poziomu w maszynie stanowej klasy ma parametry reprezentujące argumenty przekazane maszynie w chwili utworzenia obiektu.

Akcje wejściowe i wyjściowe – przykład



Kolejność realizacji: *event*1; *actp*; *act*; *actq*

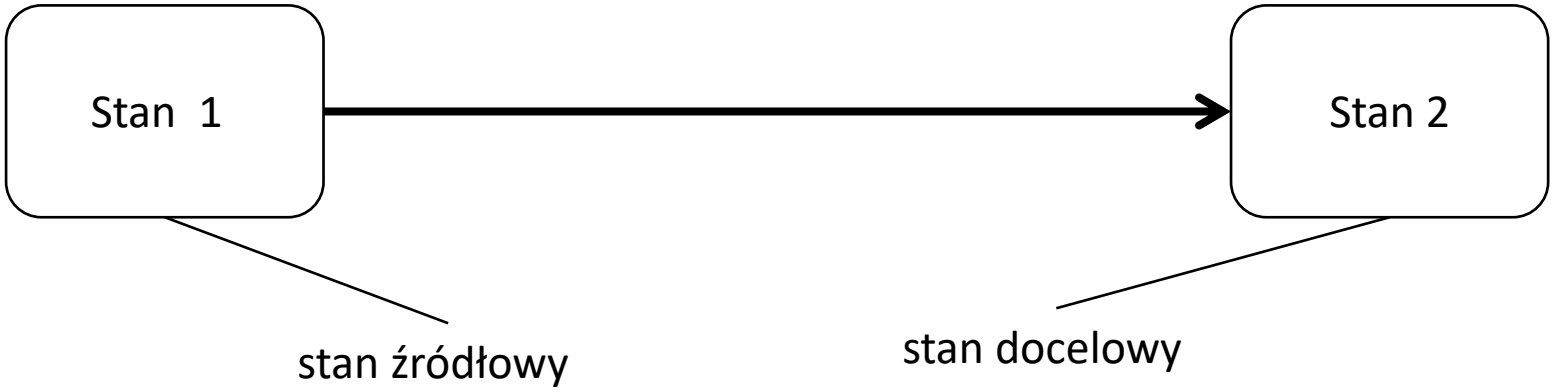
Przejście między stanami

Przejście to związek pomiędzy dwoma stanami, wskazujący, że obiekt znajdujący się w pierwszym (źródłowym) stanie wykona pewną akcję i przejdzie do drugiego stanu (docelowego) ilekroć znajdzie określone zdarzenie i będą spełnione określone warunki.

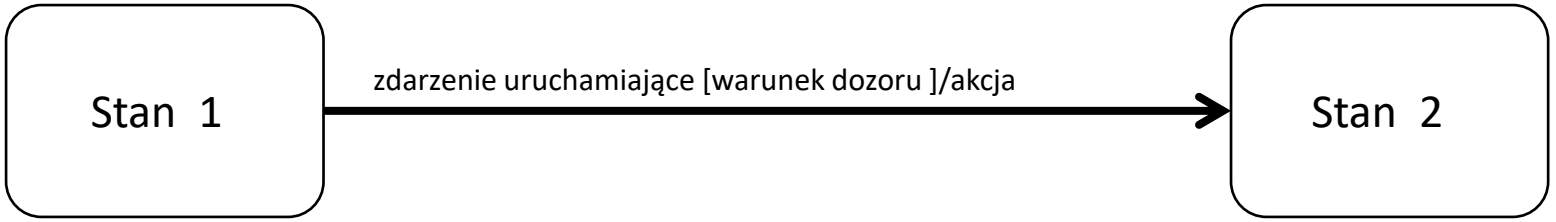
Uwagi:

- przejście reprezentuje zachowanie, które powoduje zmianę stanu obiektu (z określonego stanu do jednego ze stanów możliwych dla niego do osiągnięcia).
- przejścia mogą być wyzwalane przez zakończone aktywności, zdarzenia, ale tylko te, na które odpowiada obiekt będąc w danym stanie.
- przejście jest niepodzielne, tzn. nie można go przerwać, i trwa minimalny okres czasu.

Przejście między stanami – przykład



Przejście może posiadać dodatkowe parametry.



Składniki przejścia

1. **Stan źródłowy** - stan opuszczany w wyniku przejścia.

2. **Zdarzenie uruchamiające** - jest to zdarzenie, które oznacza uruchomienie przejścia pod warunkiem spełnienia warunku dozoru

3. **Warunek dozoru** - jeśli warunek dozoru nie jest spełniony, to mimo pojawienia się zdarzenia uruchamiającego przejście nie będzie wykonane, a zdarzenie zostanie zignorowane. Wartość dozoru jest obliczana po zajściu zdarzenia.

4. **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.

5. **Stan docelowy** - stan obiektu po zakończeniu przejścia.

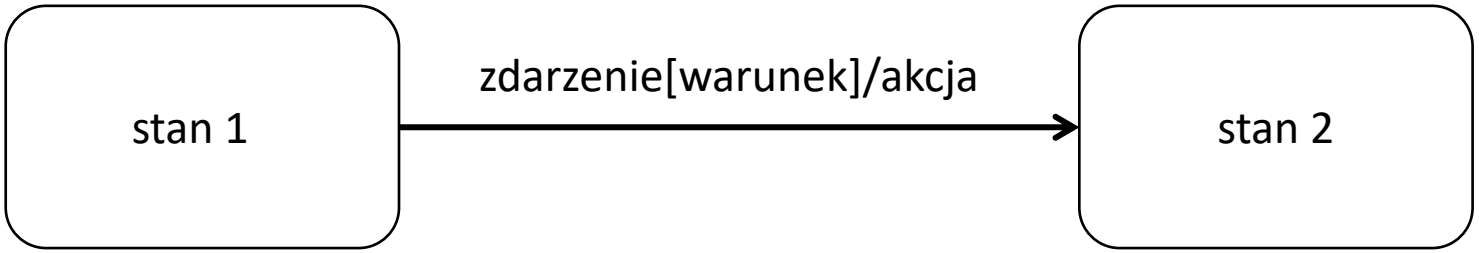
Rodzaje przejść

Mamy kilka rodzajów przejść:

- przejścia zewnętrzne,
- przejścia wewnętrzne,
- samoprzejścia,
- przejścia automatyczne.

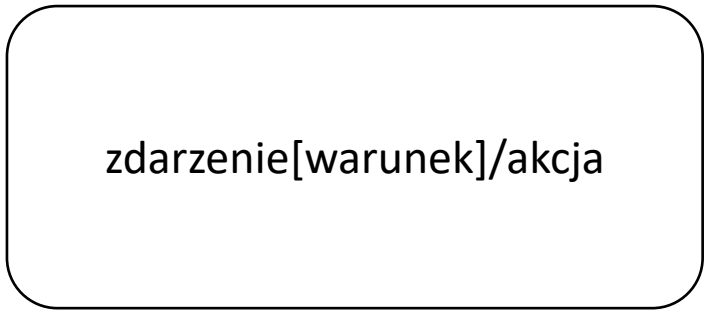
Przejście zewnętrzne

Przejście zewnętrzne – zdarzenie powoduje zmianę stanu obiektu ze stanu stan1 do stanu stan2, o ile jest spełniony warunek oraz przed przejściem obiektu do stanu stan2 wykonywana jest akcja.



Przejście wewnętrzne

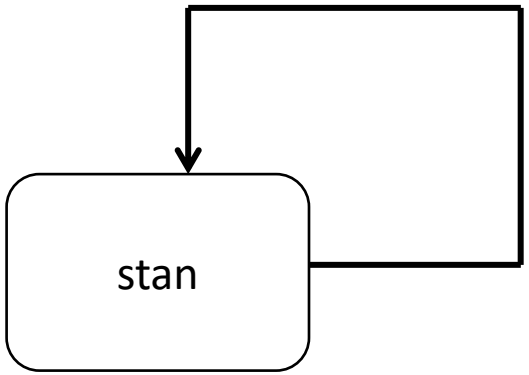
Przejście wewnętrzne - w reakcji na zaistnienie pewnego zdarzenia wykonywana jest akcja, o ile jest spełniony warunek, w przeciwieństwie do przejścia zewnętrznego przejście wewnętrzne nie powoduje zmiany stanu.



Przejście wewnętrzne może mieć zdarzenia z parametrami i warunki dozoru. Takie przejście może być traktowane jako przerwanie.

Samoprzejście

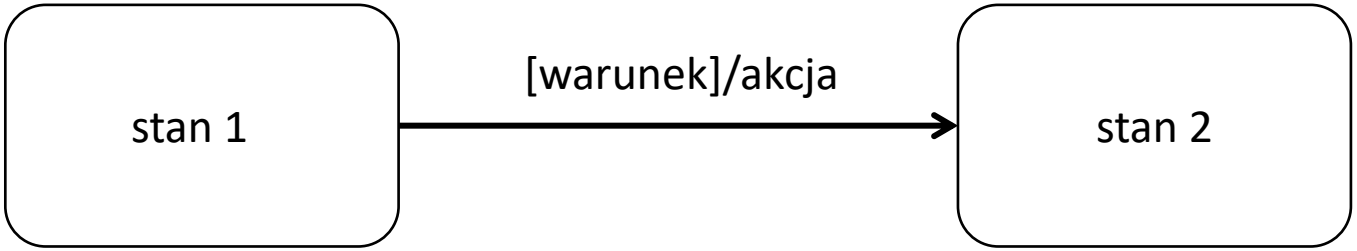
Samoprzejście – jest bardzo podobne do przejścia wewnętrznego z tą różnicą, że obiekt opuszcza stan, ale powraca do niego ponownie po obsłużeniu zdarzenia, co wiąże się z wykonaniem wszystkich akcji wyspecyfikowanych po słowach kluczowych exit i entry.



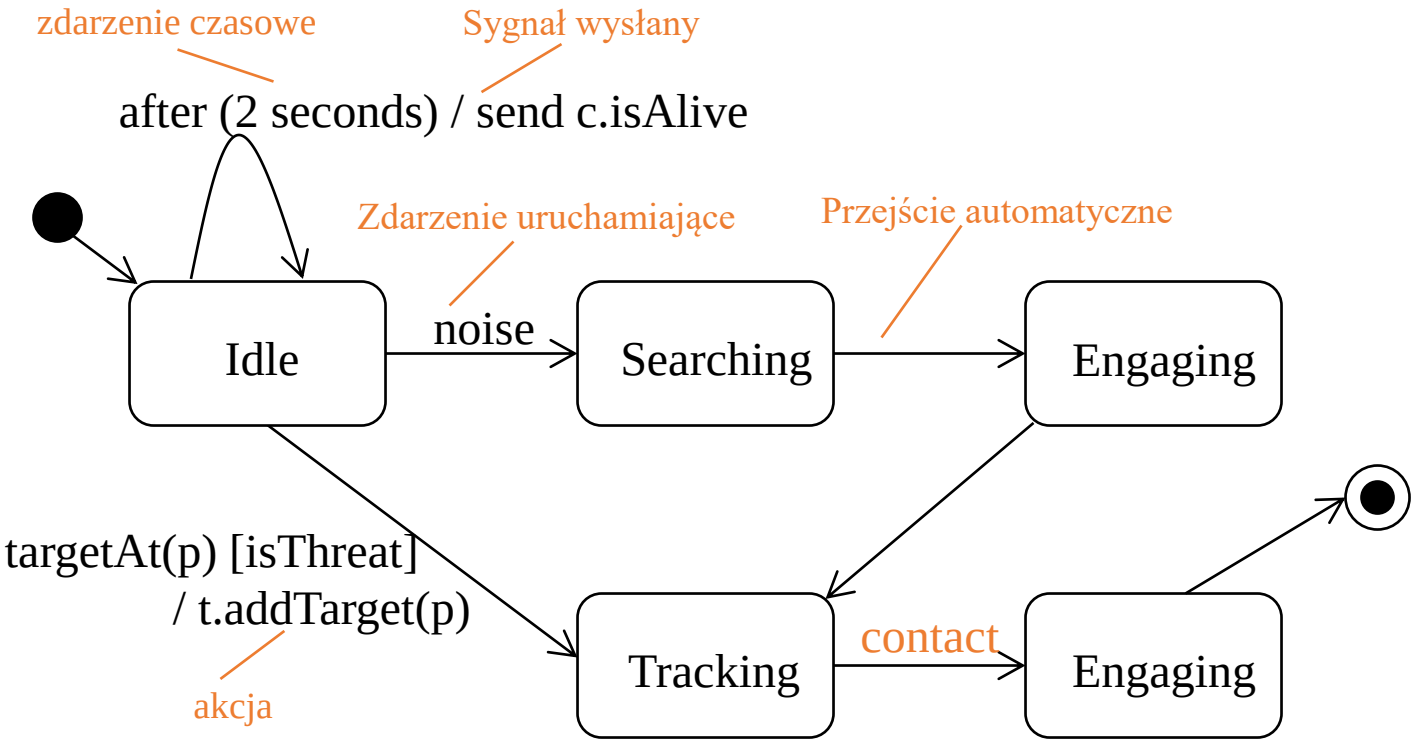
zdarzenie[warunek]/akcja

Przejście automatyczne

Przejście automatyczne – przejście ze stanu stan1 do stanu stan2 następuje wówczas, gdy wykonane zostały wszystkie operacje zdefiniowane w stan1 (specyfikowane po słowach kluczowych: entry, do i exit) oraz spełniony jest warunek. Przed wejściem do stanu stan2 wykonywana jest akcja.



Maszyna stanowa – przykład

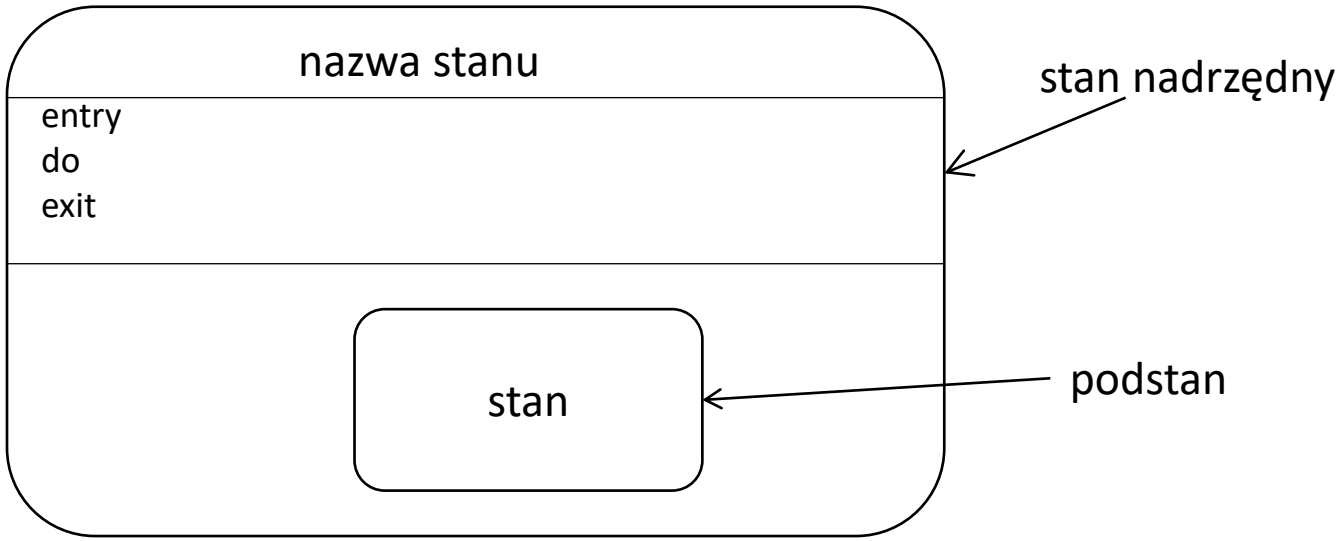


Czynności w stanie

• W danym stanie obiekt może być bezczynny lub wykonywać pewne czynności.
• **Czynność** taka, związana z frazą ***do***, może polegać na uruchomieniu innej maszyny stanowej, może to być również ciąg akcji.
• Między tymi akcjami może zaistnieć zdarzenie, które spowoduje opuszczenie stanu.
• W trakcie przebywania w stanie mogą pojawić się przejścia, które nie „nie pasują” do żadnego z przejść prowadzących od tego stanu. W takich przypadkach zdarzenia te będą ignorowane (tracone), chyba że użyjemy mechanizmu zapamiętania listy zdarzeń, które pojawiły się.
• Te zapamiętane zdarzenia są odroczone (*defer*) do czasu przejścia do stanu, w którym nie są odroczone.

Złożone diagramy stanów

W pojedynczym stanie można umieścić jeden lub więcej stanów, co znaczy, że gdy element znajduje się w danym stanie, to elementy znajdujące się wewnątrz niego mają swoje stany.



Podstany

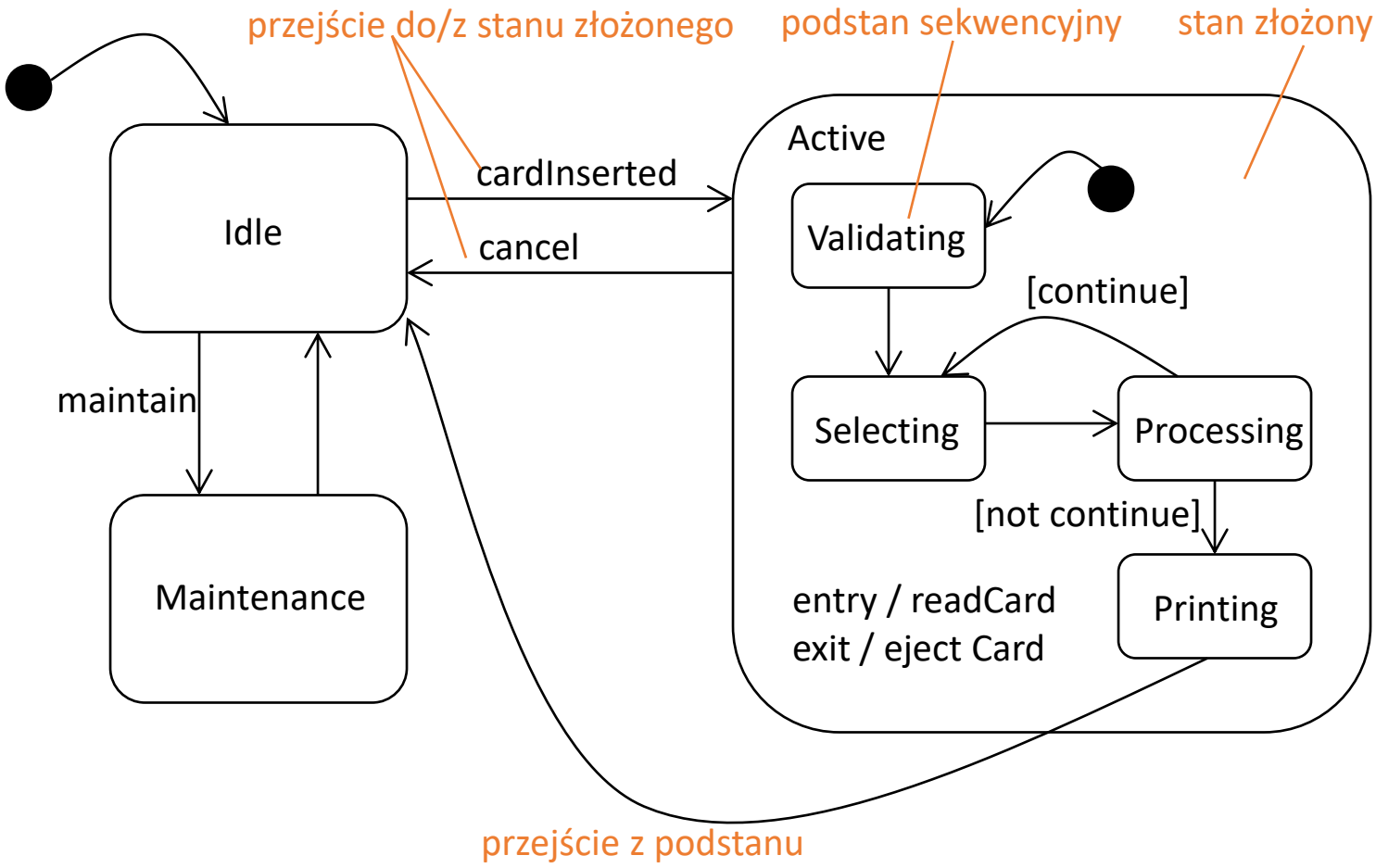
• Zwykły stan nie ma struktury, UML dostarcza możliwości modelowania **stanów złożonych** (zagnieżdżonych).
• W takim przypadku, **stan nadrzędny** (superstan) jest dekomponowany na **podstany**, opisywane przez diagramy stanów.
• Podstany mogą być współbieżne (ortogonalne) lub sekwencyjne (rozłączne).
• Poziom zagnieżdżenia nie jest ograniczany przez standard języka.

Podstany sekwencyjne

Podstany sekwencyjne – zagnieżdżona maszyna stanowa może mieć najwyżej jeden stan początkowy i jeden końcowy.

- Stan prosty jest specyfikowany tylko przez zbiór operacji (akcji i/lub aktywności) oraz przejść wewnętrznych, w związku z czym nie posiada substruktury.
- W przeciwieństwie do stanu prostego, stan złożony sekwencyjny może być zdekomponowany na podstany, z których tylko jeden może być aktywny w danym momencie.
- Dekompozycja ta może być traktowana jako rodzaj specjalizacji, w której każdy z podstanów dziedziczy przejścia nadstanu.

Podstany sekwencyjne – przykład



Stany wznowienia, problem wznowienia

• Po wejściu do stanu złożonego działanie maszyny stanowej rozpoczyna się domyślnie od jej stanu początkowego (jeśli przejście nie trafia bezpośrednio do innego podstanu).
• W modelowaniu stanów złożonych przydatne jest również pamiętanie ostatniego podstanu przy opuszczaniu stanu złożonego. Jest to przydatne np. w przypadkach, gdy wyjście ma charakter tymczasowy i po powrocie należy kontynuować wcześniejszą czynność.
• Można to zrealizować przez przypisanie akcji wyjściowej zmieniającej wartość zmiennej lokalnej przy opuszczaniu podstanu lokalnego. Następnie należy przypisać dozory uwzględniające wartość tej zmiennej kierujące przejście do właściwych podstanów przy wejściu do stanu złożonego.

Prostszy sposób polega na zastosowaniu **stanów wznowienia**, które służą do zapamiętania ostatniego podstanu przed opuszczeniem stanu złożonego.

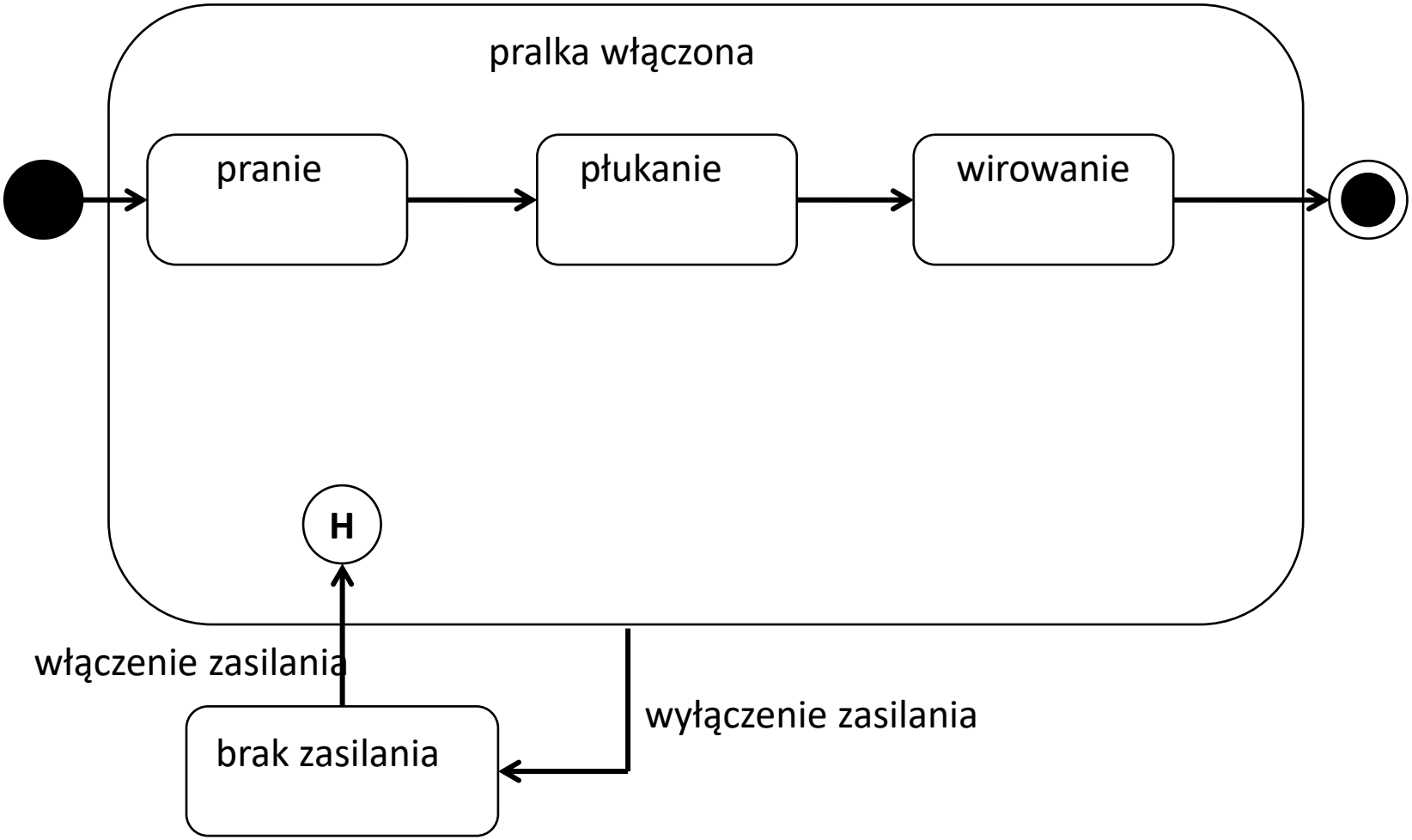
Stan wznowienia (historyczny)

Stan wznowienia inaczej **historyczny** jest pseudostanem, służy do zapamiętania podstanu sekwencyjnego przed opuszczeniem maszyny stanowej.

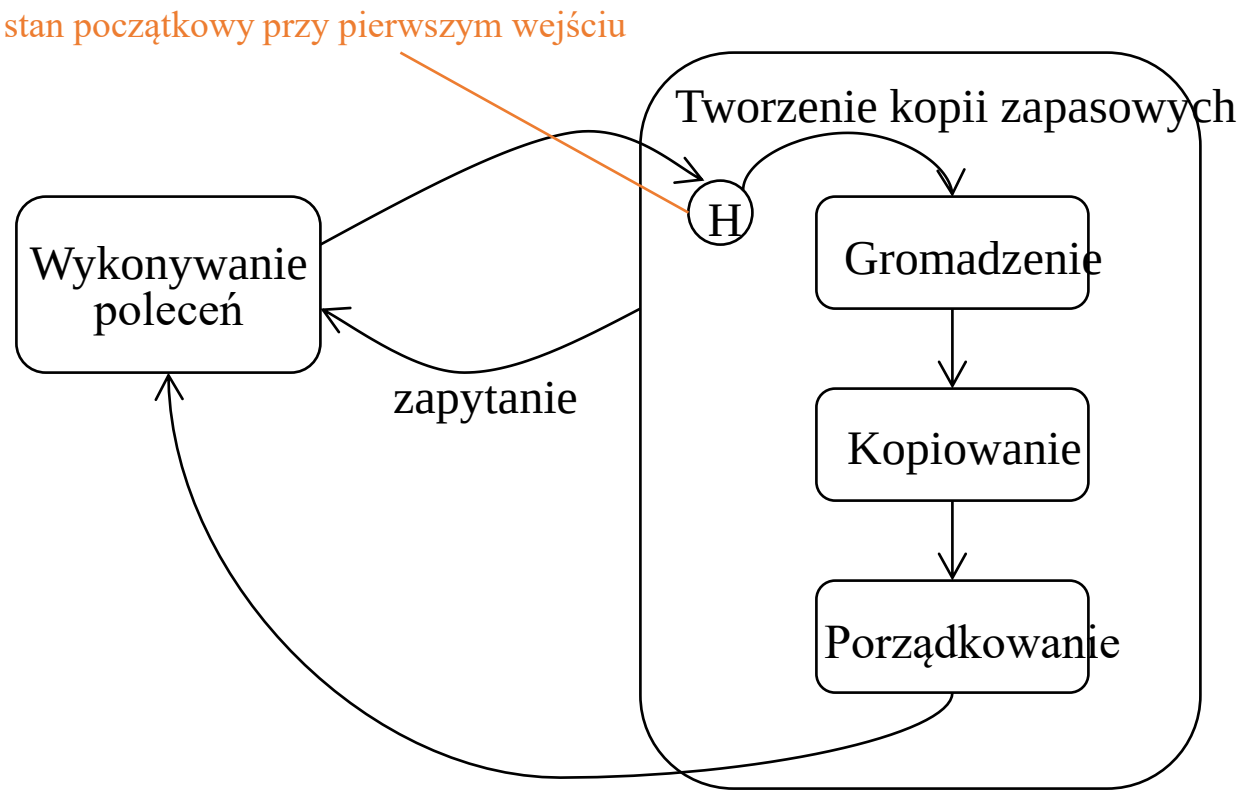
- W przypadku przerwania, w stanie wznowienia zapamiętywany jest podstan, w którym obiekt znajdował się w chwili przerwania.
- Wznowienie następuje w podstanie, w którym przerwano działanie maszyny stanowej.
- Gdy zagnieżdżona maszyna stanowa osiągnie swój wcześniej założony stan końcowy, to zawartość stanu wznowienia jest anulowana.

symbol stanu wznowienia

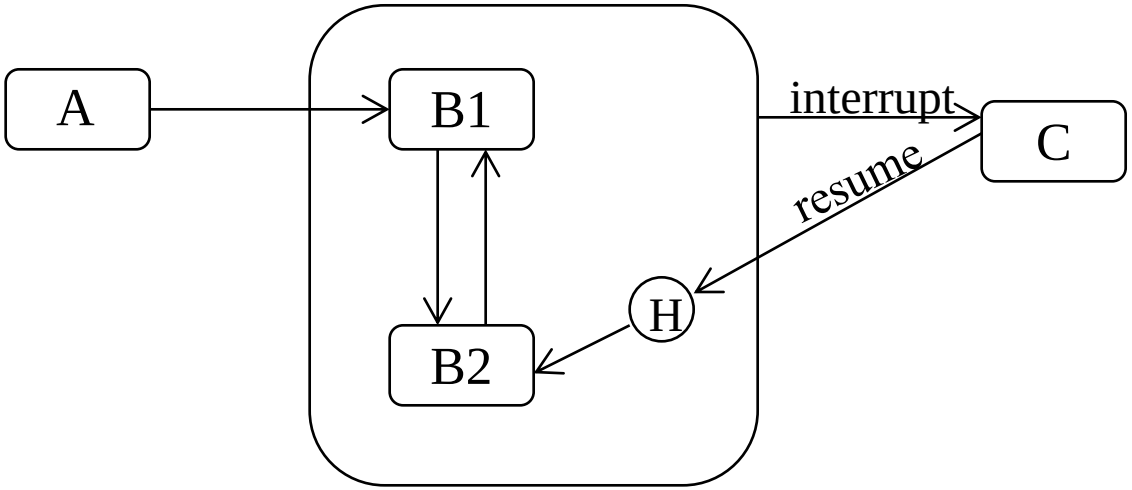
Stan wznowienia – przykład



Stan wznowienia – inny przykład



Stan wznowienia – elementarny przykład

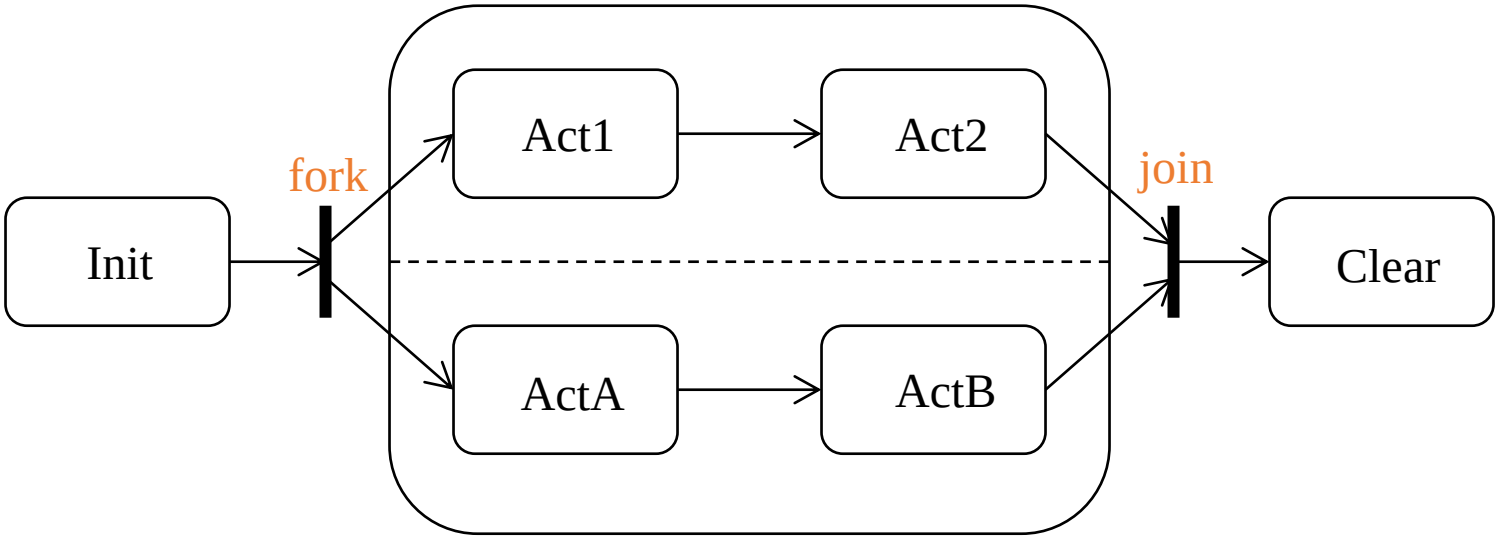


Rodzaje wznowień

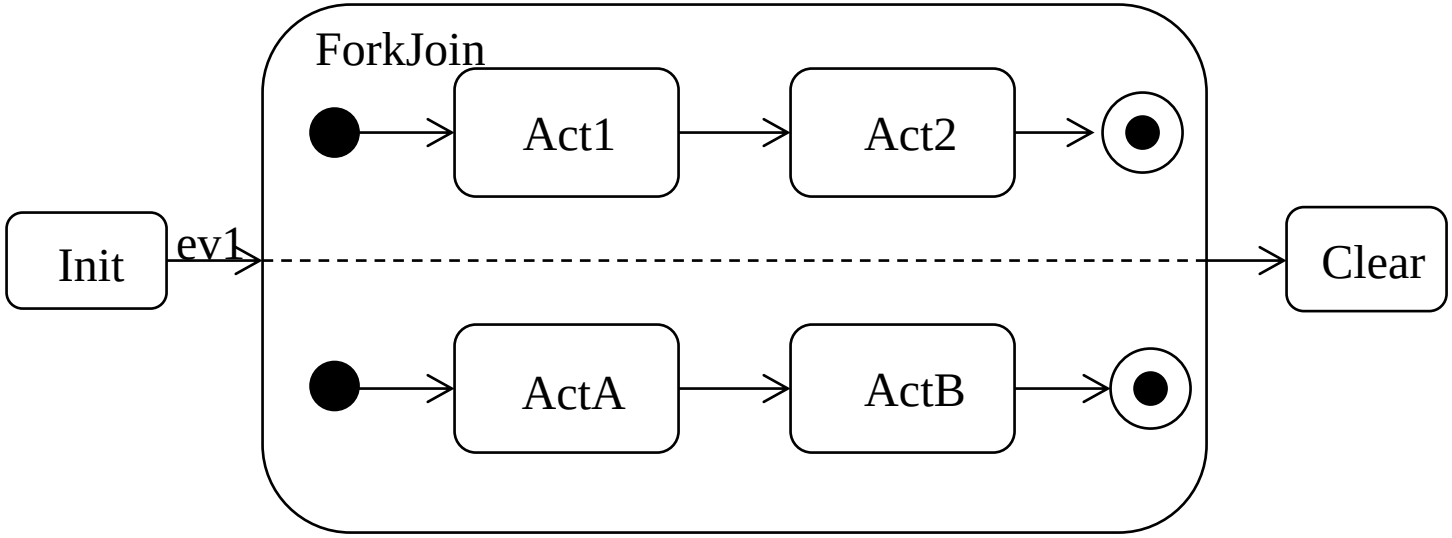
Płytkie wznowienie (*shallow history*) oznacza powrót do ostatniego aktywnego stanu najbliżej zagnieżdżonej maszyny stanowej (najbardziej zewnętrzny podstan).

Głębokie wznowienie (*deep history*) oznacza zapamiętanie zagnieżdżonych podstanów na wszystkich poziomach.

Podstany współbieżne

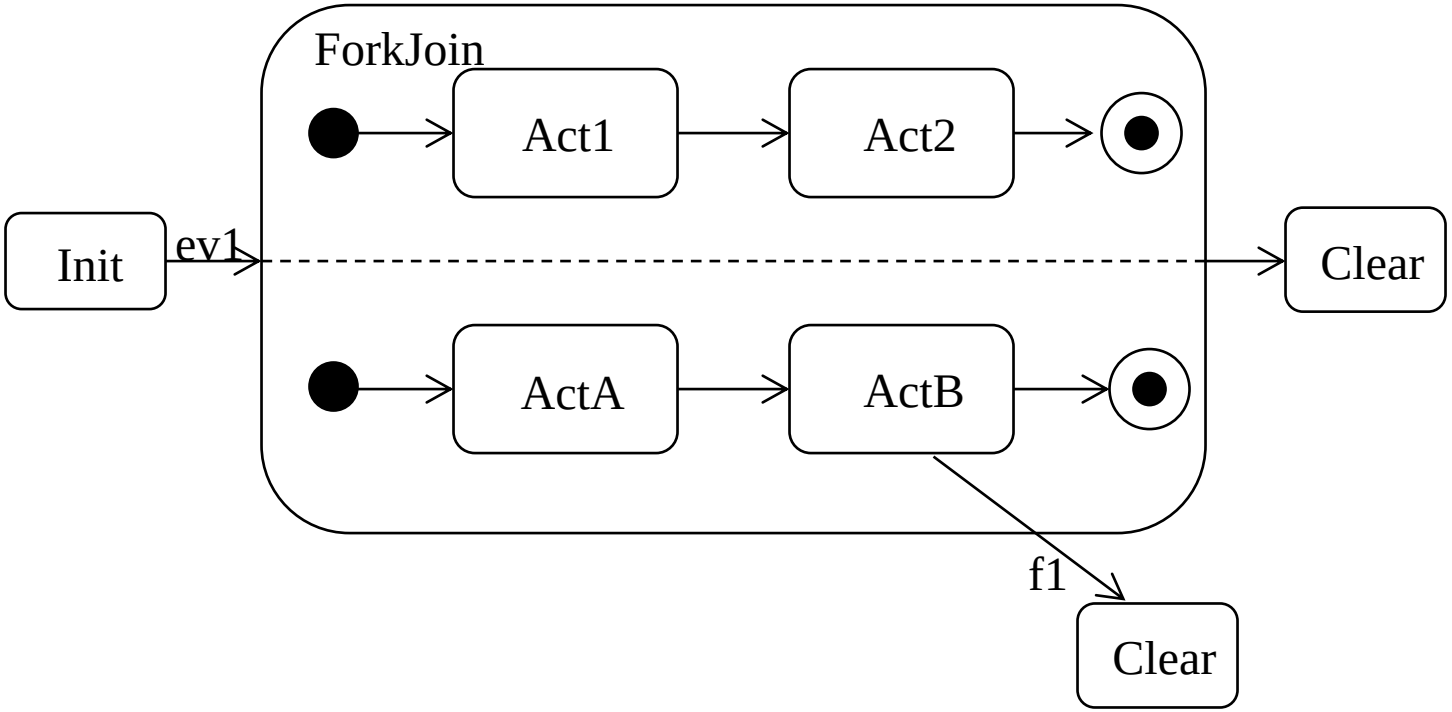


Podstany współbieżne – inna notacja



Wcześniejsze osiągnięcie stanu końcowego przez jeden z postanów współbieżnych powoduje zatrzymanie i oczekiwanie aż drugi z nich osiągnie podstan końcowy i dopiero wówczas może nastąpić opuszczenie obu postanów (obszarów współbieżnych).

Podstany współbieżne, wyjście boczne



Jeśli wystąpi zdarzenie *f*1 w stanie *ActB* wówczas następuje przejście do stanu *Clear*. Takie przejście jest natychmiastową realizacją *join*, tzn. zostaje zawieszony stan *Act*2 i *ActB*.

Zastosowanie

Modelowanie historii życia obiektów (przypadków użycia, systemu) jest najczęściej stosowanym zastosowaniem maszyn stanowych.

Modelowanie rozpoczyna się od określenia:

- zdarzeń, na które obiekt winien reagować,
- odpowiedzi obiektu na te zdarzenia,
-wpływu historii na bieżące zachowanie obiektu.

Modelowanie maszyn stanowych może być ukierunkowane zasadniczo na dwa sposoby:

1. na ukazanie stanów (*state-oriented*) – ukazuje złożoność samej maszyny;
2. na ukazanie przejść (*transition-oriented*) – ukazuje przede wszystkim przejścia i zdarzenia związane z obsługą sygnałów.

Wytyczne dotyczące modelowania

1. Określ otoczenie maszyny stanowej w zależności od kontekstu (klasa, przypadek użycia, system).
2. Ustal stan początkowy i końcowy i ewentualne warunki początkowy i końcowy.
3. Określ zdarzenia, na które obiekt powinien reagować.
4. Zapisz stany najwyższego poziomu, poczynając od początkowego do końcowego. Połącz stany przejściami z uwzględnieniem zdarzeń wyzwalających. Dodaj akcje do przejść.
5. Zidentyfikuj akcje wejściowe i wyjściowe.
6. Jeśli konieczne, rozwiń stany w podstany.
7. Sprawdź czy zdarzenia występujące w maszynie odpowiadają zdarzeniom oczekiwanym przez interfejsy obiektu. Zbadaj również czy wszystkie zdarzenia oczekiwane przez interfejsy są obsługiwane przez maszynę stanową. Zbadaj miejsca, gdzie można pominąć zdarzenia.
8. Sprawdź czy wszystkie akcje wymienione w maszynie stanowej mogą być realizowane przez związki, metody i operacje modelowanego obiektu.
9. Analizuj zachowanie utworzonej maszyny stanowej. Porównaj je ze zbiorem pożądanych zdarzeń i odpowiedzi. Zbadaj osiągalność stanów itp.
10. Po uporządkowaniu porównaj ponownie zachowanie maszyny stanowej z pożądanym.