

# Wykład 4 – część pierwsza

Diagramy UML 2 – część druga

Na podstawie

**UML 2.0 Tutorial**

[http://sparxsystems.com.au/resources/  
uml2\\_tutorial/](http://sparxsystems.com.au/resources/uml2_tutorial/)

# Dwa rodzaje diagramów UML 2

## **Diagramy UML modelowania strukturalnego**

- *Diagramy pakietów*
- *Diagramy klas*
- **Diagramy obiektów**
- **Diagramy mieszane**
- **Diagramy komponentów**
- **Diagramy wdrożenia**

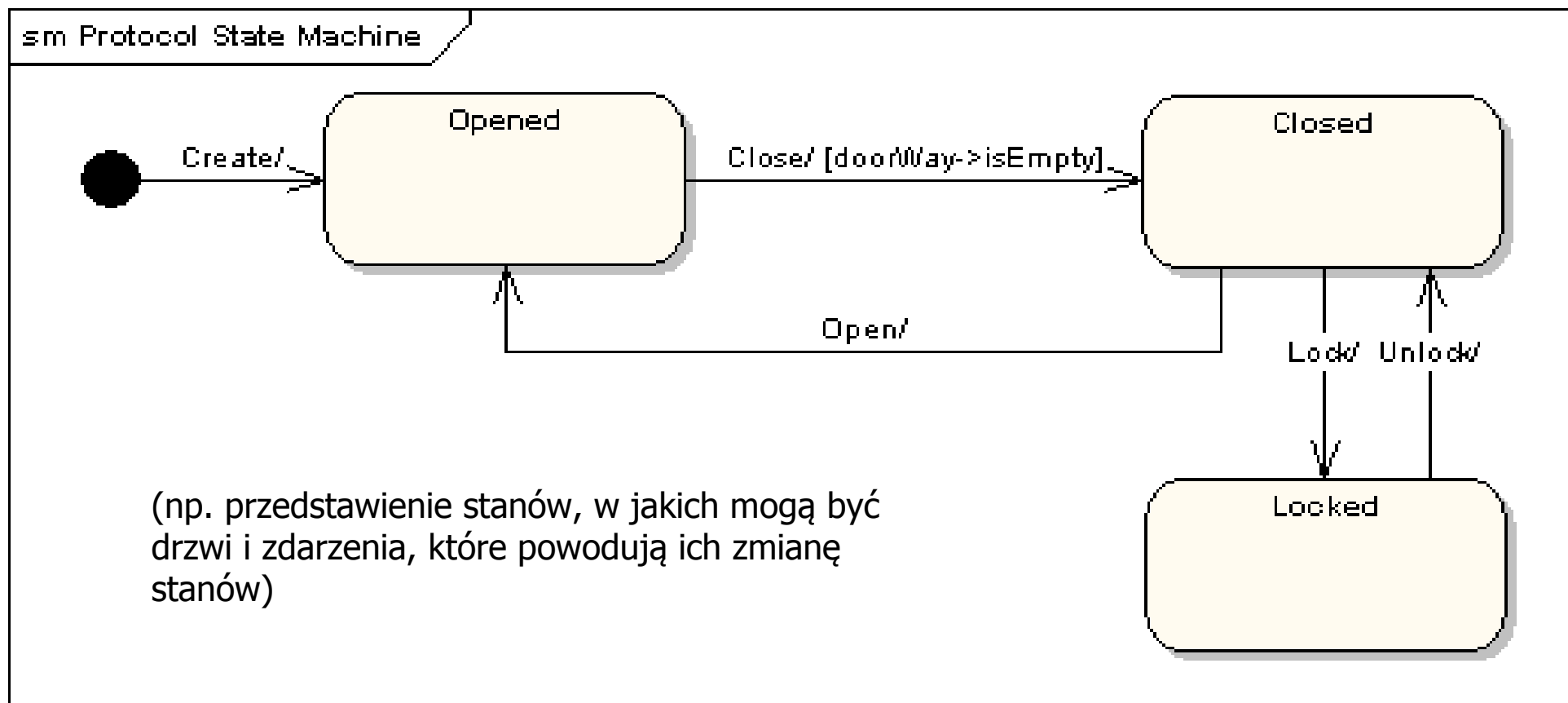
## **Diagramy UML modelowania zachowania**

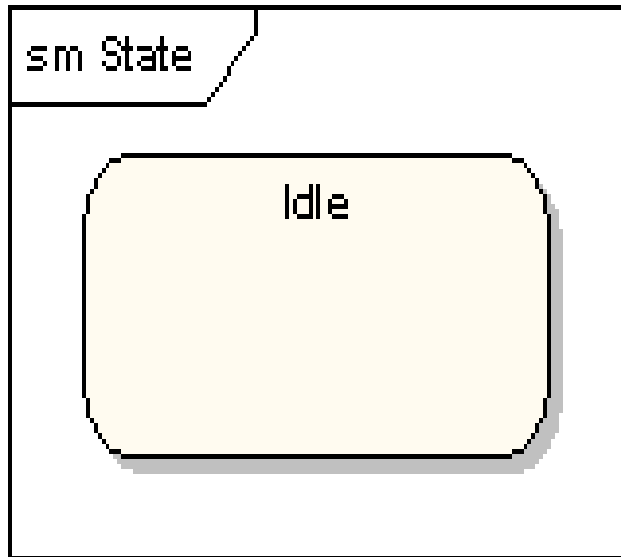
- *Diagramy przypadków użycia*
- *Diagramy aktywności*
- *Diagramy stanów*
- **Diagramy komunikacji**
- *Diagramy sekwencji*
- **Diagramy czasu**
- **Diagramy interakcji**

## Diagramy stanów

**Diagram stanu** opisuje zmiany stanu obiektu, podsystemu lub systemu pod wpływem działania operacji - jest szczególnie przydatny, gdy zachowanie obiektu jest złożone. Przedstawia on **maszynę stanów** jako przepływ sterowania między stanami.

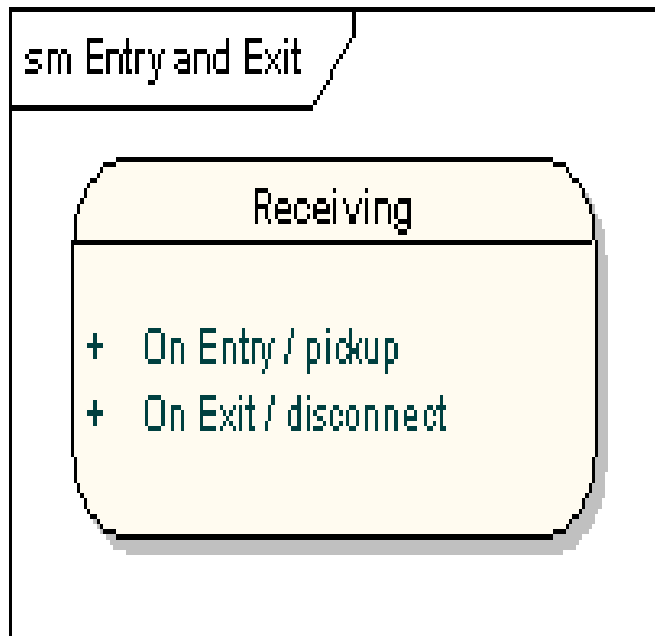
**Diagram stanów** jest grafem złożonym z wierzchołków i krawędzi: wierzchołkami są stany (prostokąty o zaokrąglonych rogach), krawędziami są przejścia (strzałki).





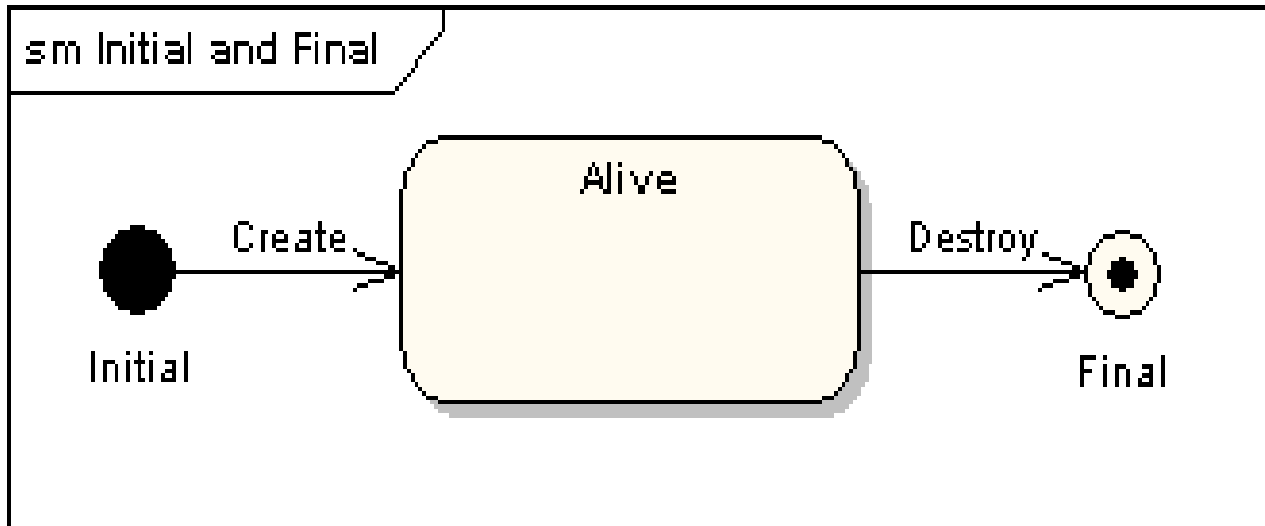
**Stan** jest okolicznością lub sytuacją, w jakiej znajduje się obiekt

- jest rezultatem poprzedniej aktywności
- spełnia jakiś warunek
- jest określony przez wartości własnych atrybutów i powiązań do innych zadań
- wykonuje pewne czynności
- czeka na jakieś zdarzenie



• **Nazwa** - unikatowy ciąg znaków brak nazwy dla stanu anonimowego

• **Akcje wejściowe** (entry) i **wyjściowe** (exit) - akcje wykonywane odpowiednio przy wejściu do stanu i przy wyjściu)

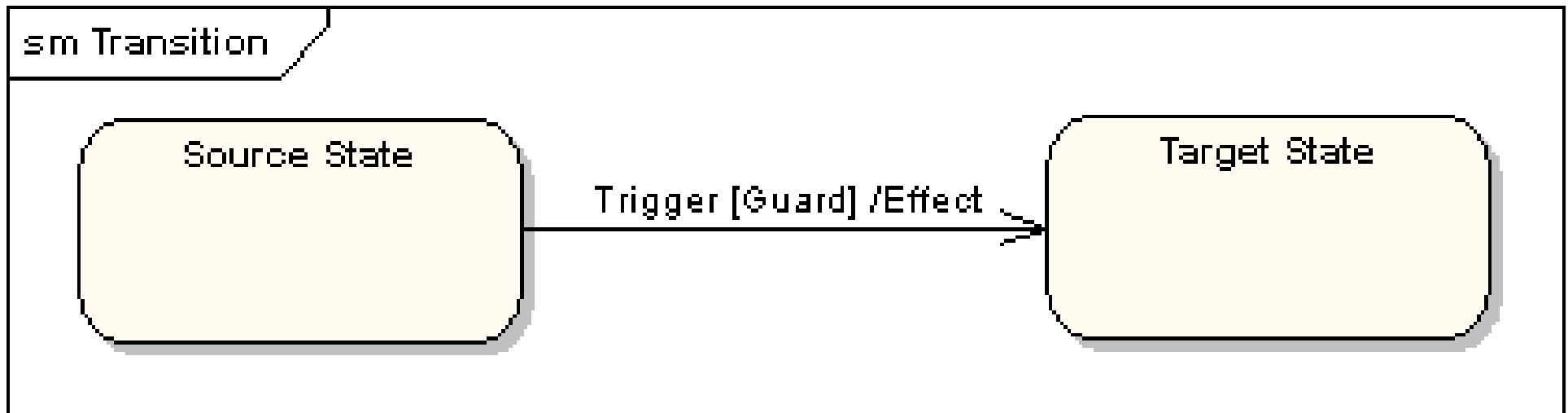


## Stan początkowy

(Initial) – może być tylko jeden stan początkowy

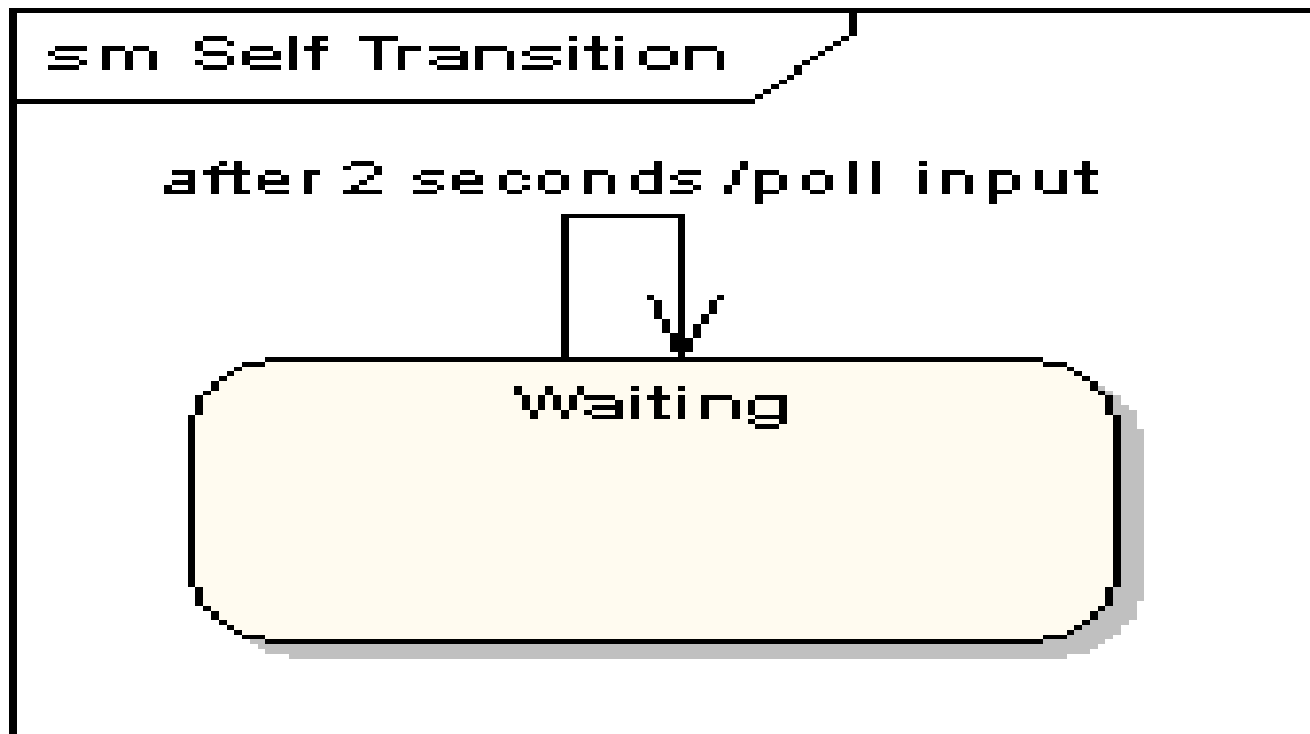
## Stan końcowy

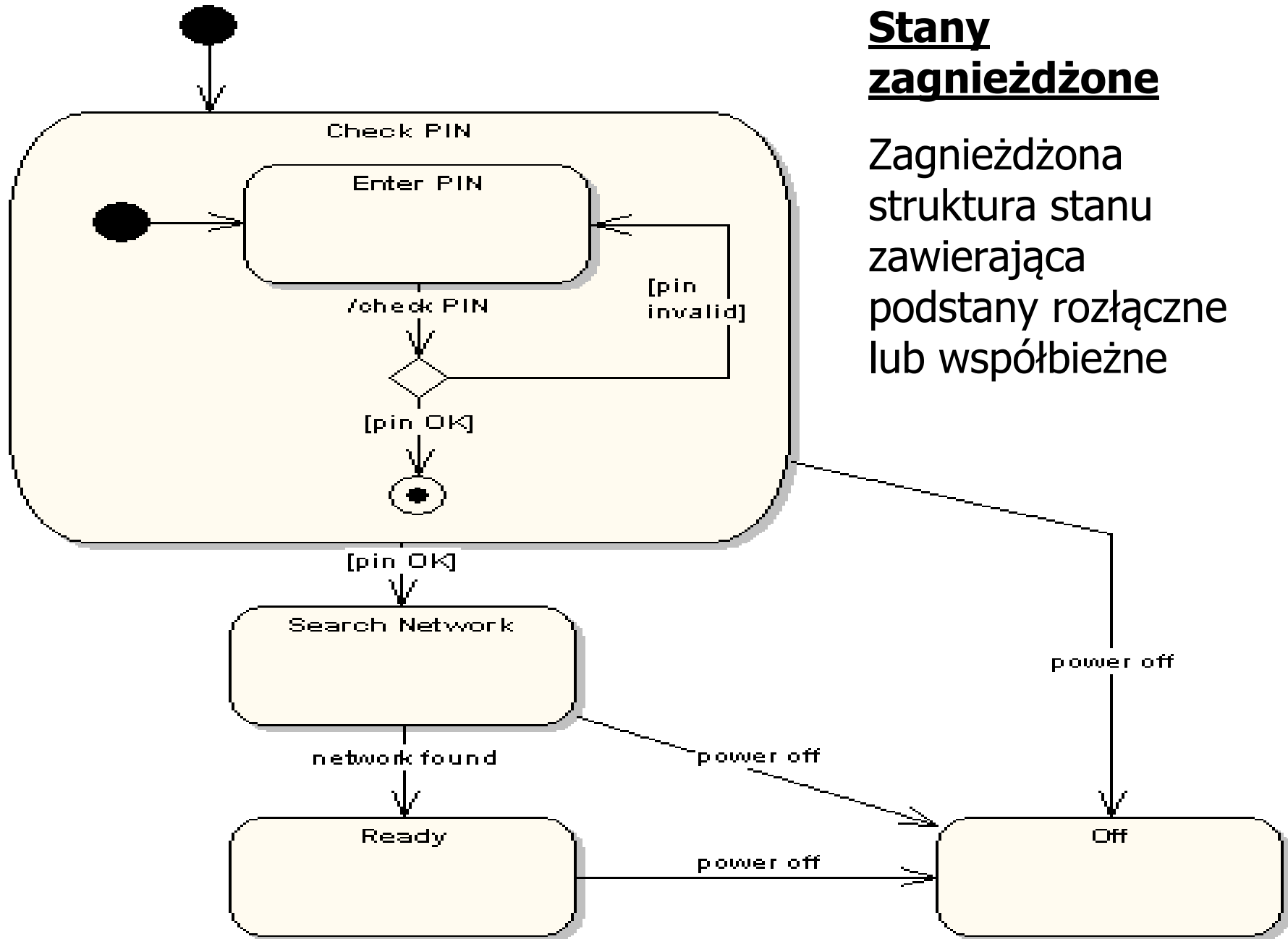
(Final) – może być kilka stanów końcowych



**Przejsie** (transition) jest związkiem między dwoma stanami, który wskazuje, że np. obiekt znajdujący się w pierwszym stanie wykona pewne **akcje (Effect)** i przejdzie do drugiego stanu, ilekroć zaistnieje określone **zdarzenie (Trigger)** i będą spełnione określone **warunki (Guard)**.

**Przejście własne** jest związkiem między tym samym stanem, który wskazuje, że np. obiekt znajdujący się w pewnym stanie wykona pewne **akcje** i powróci do tego samego stanu, ilekroć zaistnieje określone zdarzenie i będą spełnione określone warunki.



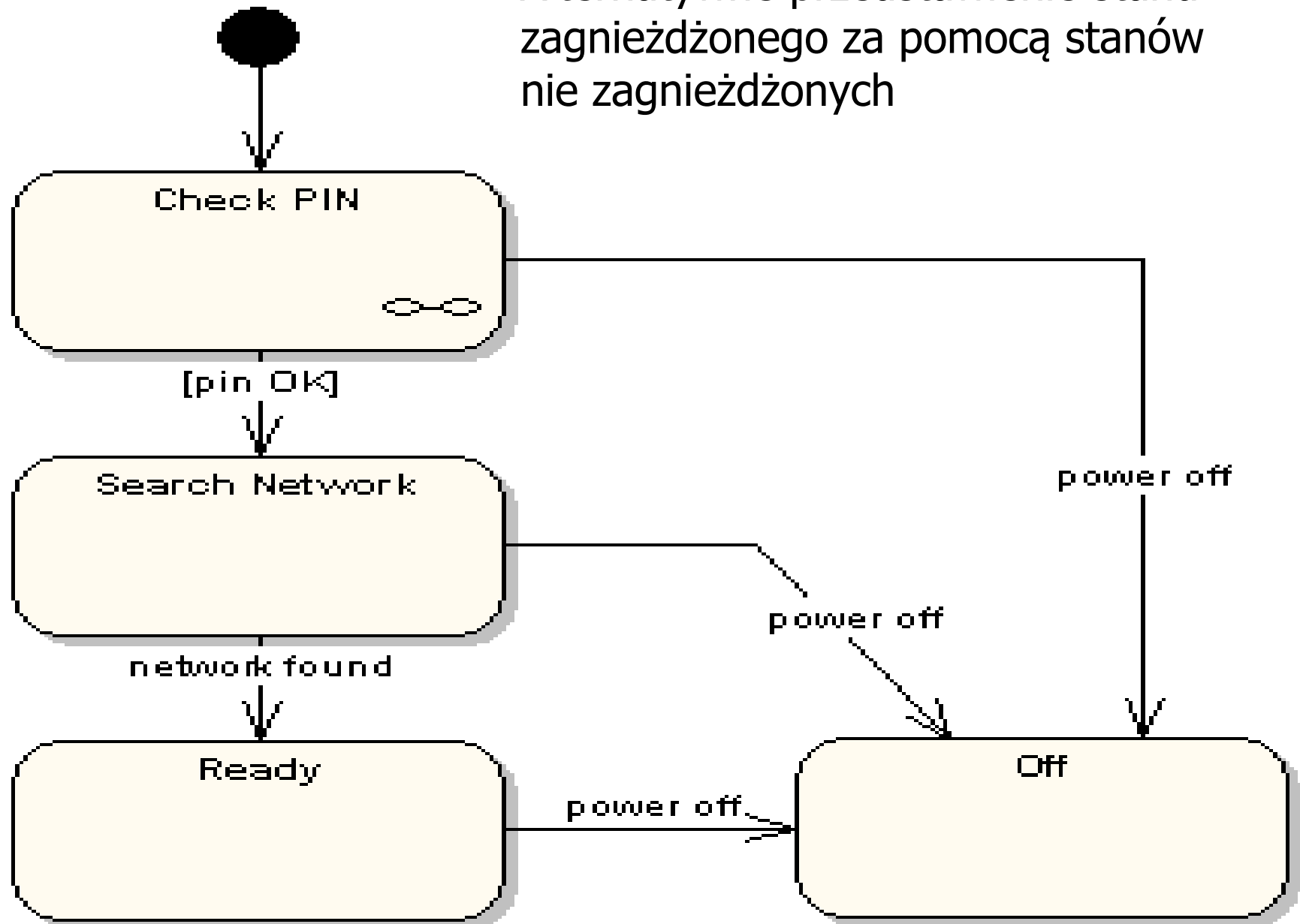


## Stany zagnieżdżone

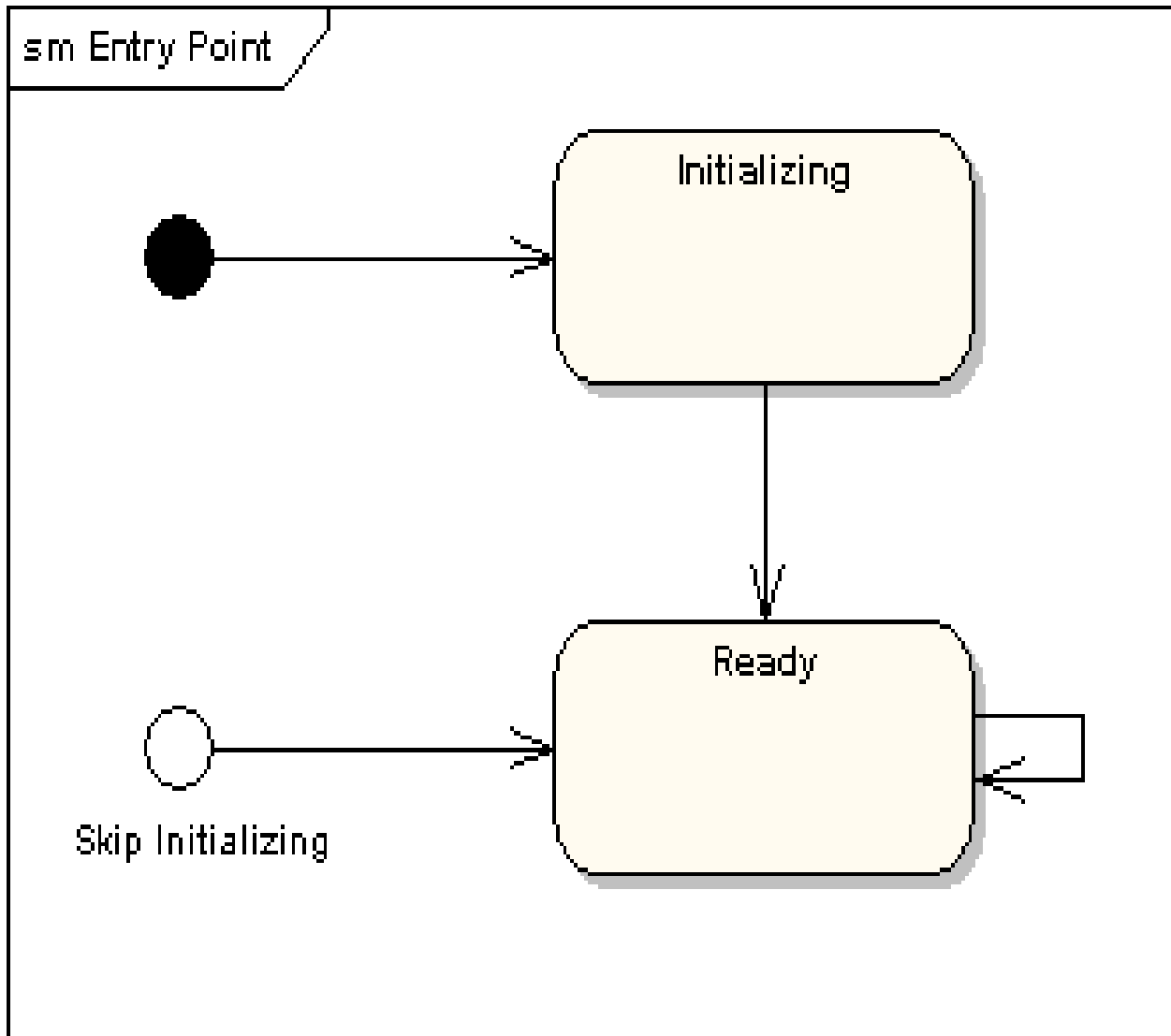
Zagnieżdżona struktura stanu zawierająca podstany rozłączne lub współbieżne

sm Composite

Alternatywne przedstawienie stanu zagnieżdżonego za pomocą stanów nie zagnieżdżonych



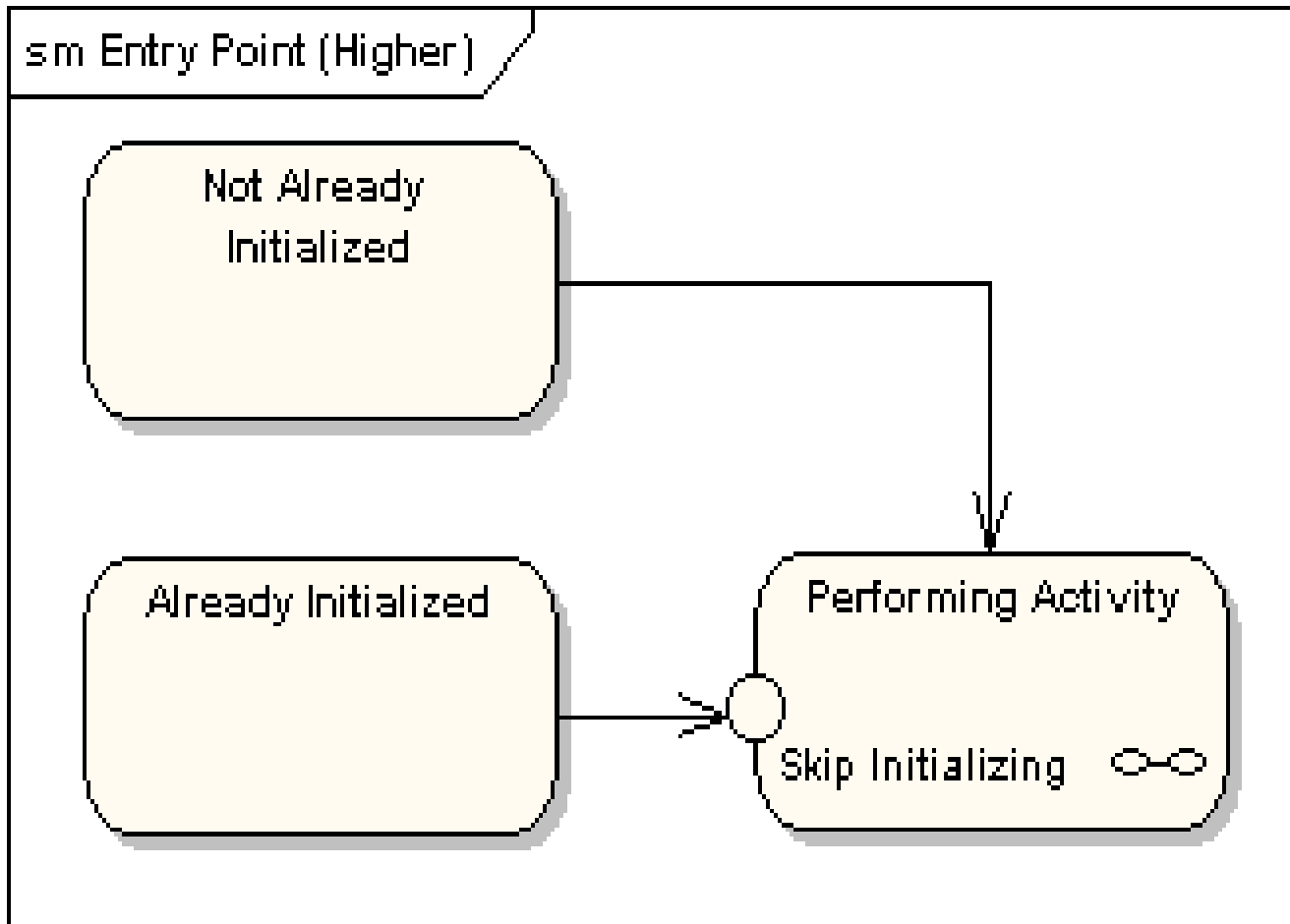




## **Entry point** –

wskazanie różnych stanów początkowych (Initial) obiektu:

- rozpoczęcie stanu z inicjalizacją (normalne)
- bez inicjalizacji (wyjątkowe)

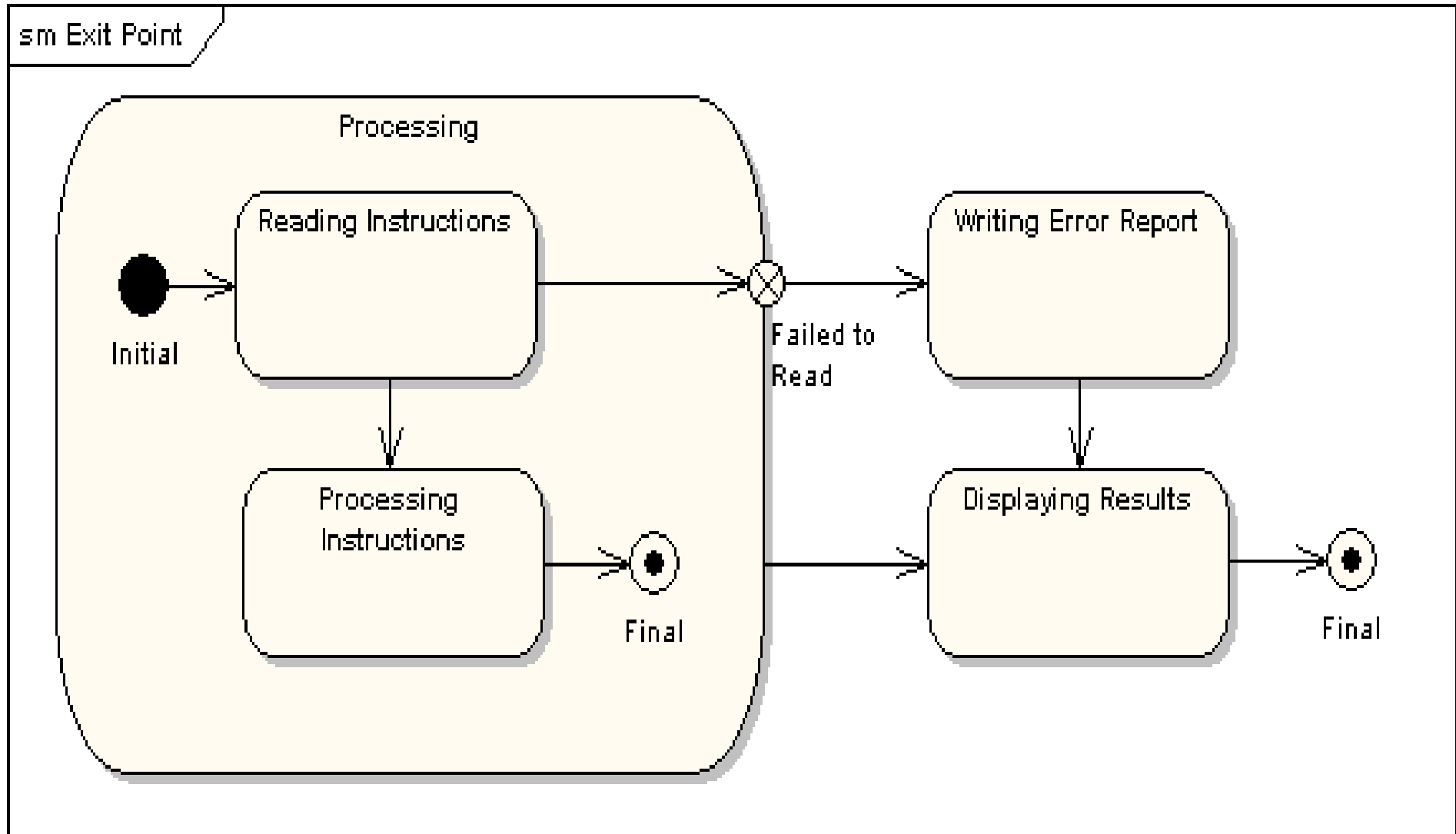


## **Entry point** –

szczegółowy model startu (z poziomu wyższego na poziom niższy), czyli wskazanie różnych stanów początkowych obiektu:

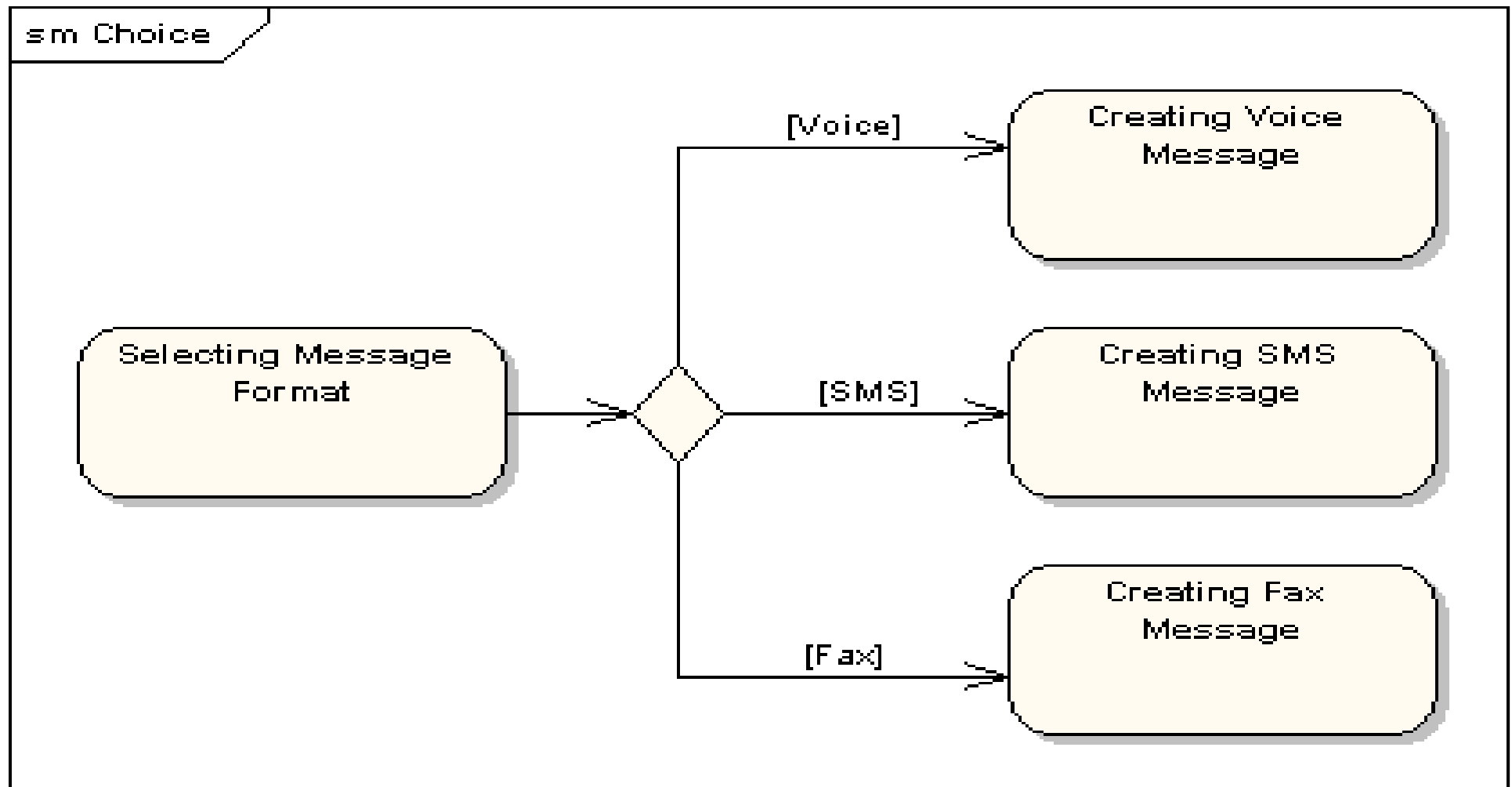
- rozpoczęcie stanu z inicjalizacją (normalne)
- bez inicjalizacji (wyjątkowe)

# Punkt wyjścia – modelowanie osiągnięcia alternatywnych stanów końcowych (Final) przez obiekt

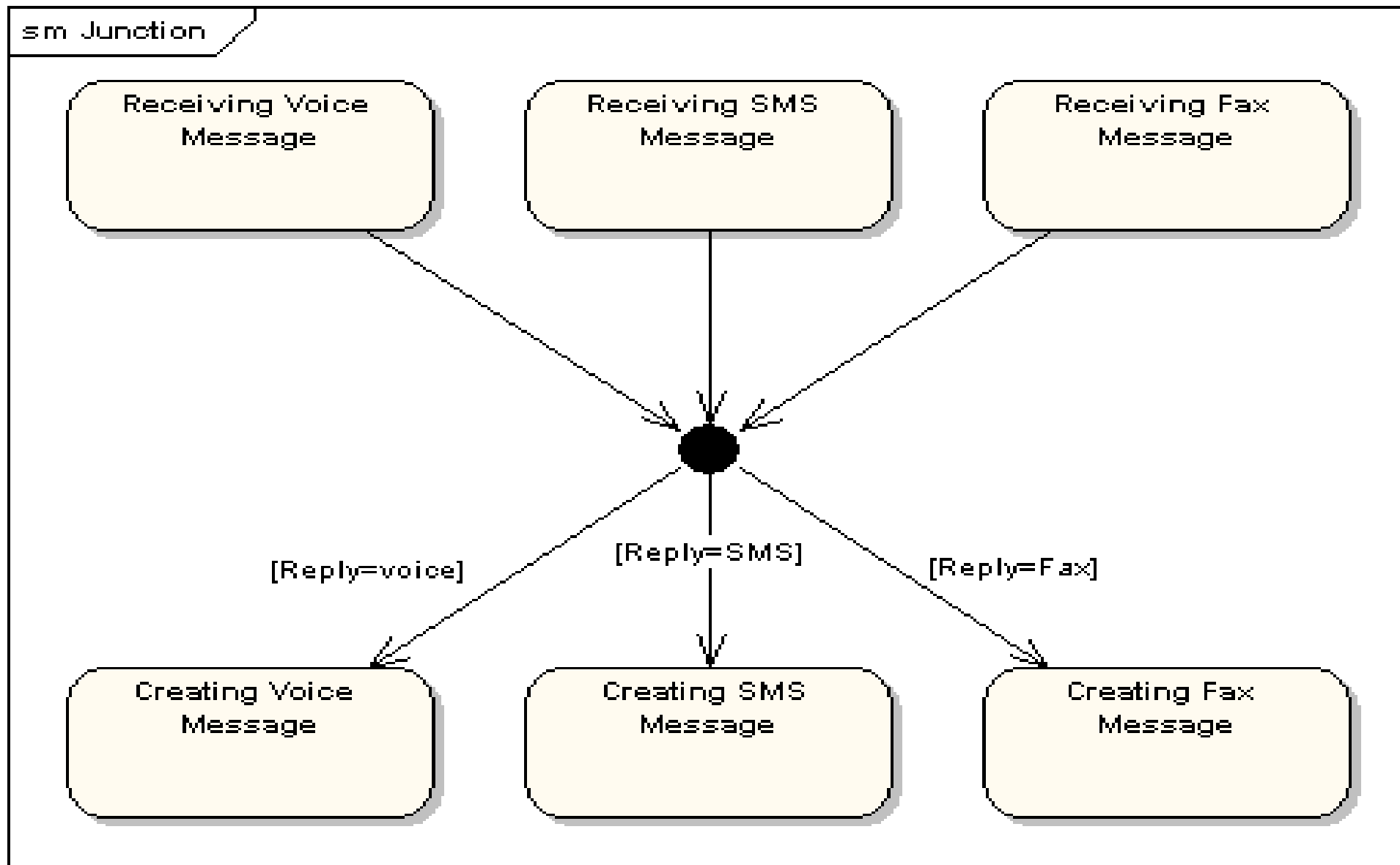


## Pseudo stan wyboru:

- Jeden stan początkowy
- W wyniku zdarzenia następuje przejście ze stanu początkowego i na podstawie formatu wygenerowanej wiadomości możliwość wyboru jednego ze stanów wyjściowych

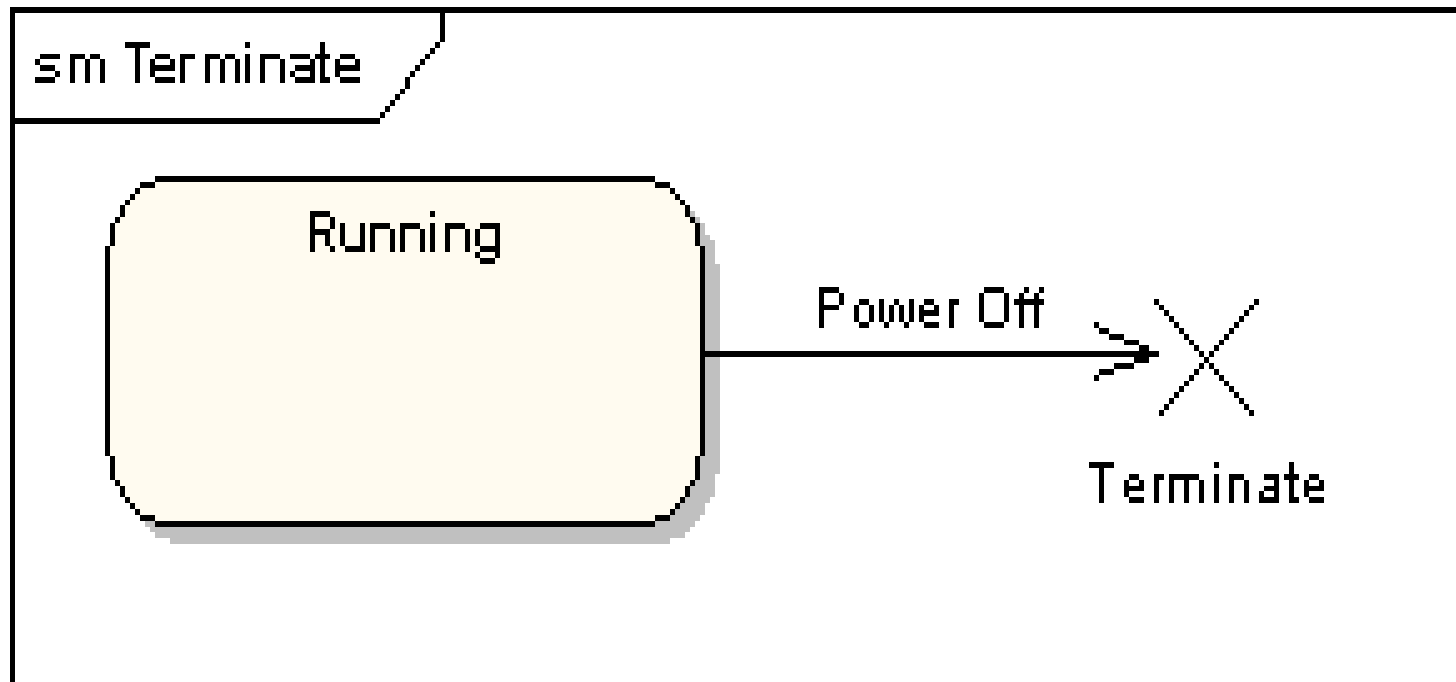


**Pseudo stan typu połączenie** – możliwości wyboru stanów końcowych po zdarzeniach zachodzących po stanach początkowych

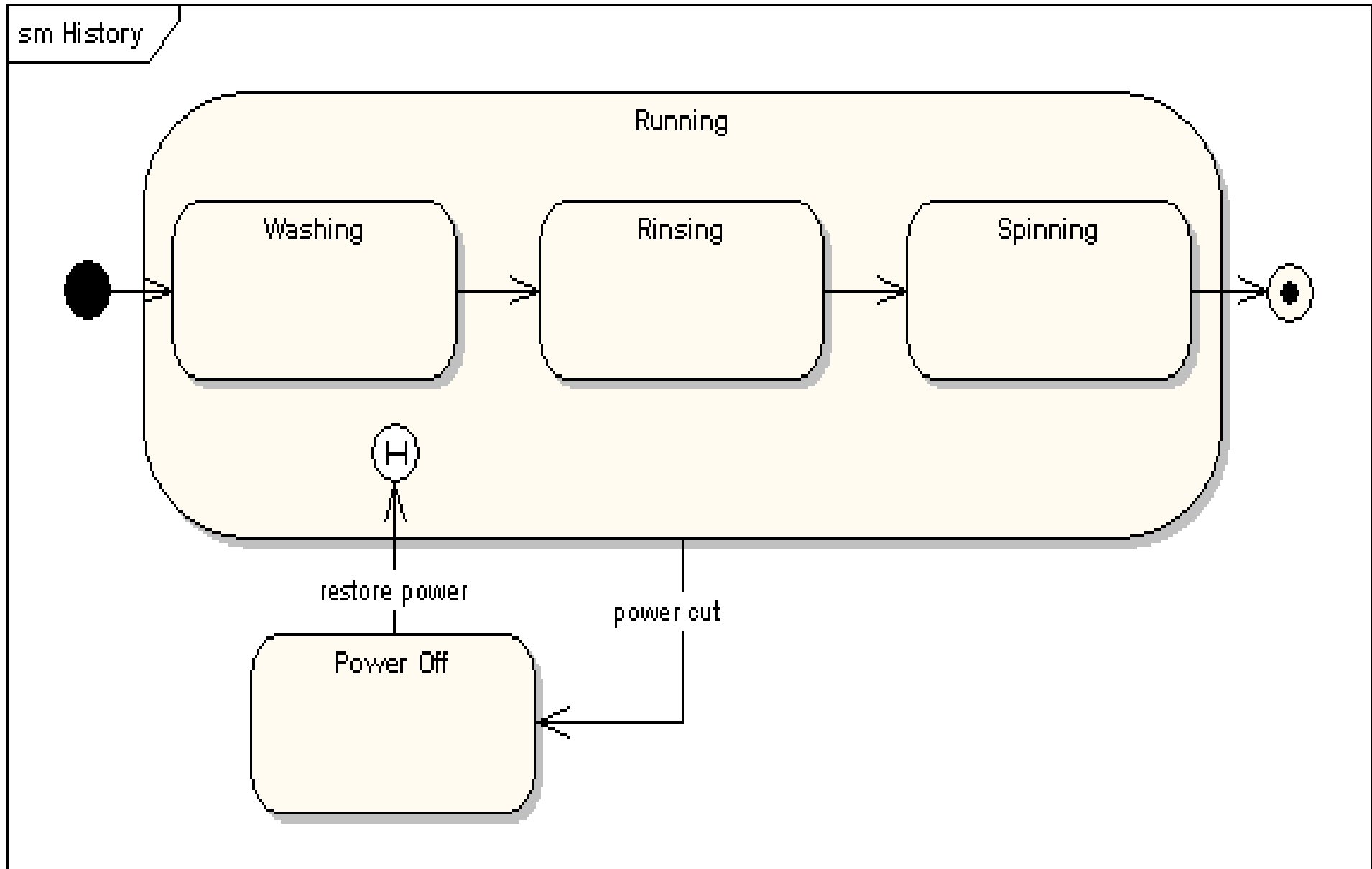


## Pseudo stan typu zakończenie

oznacza zakończenie linii życia obiektu

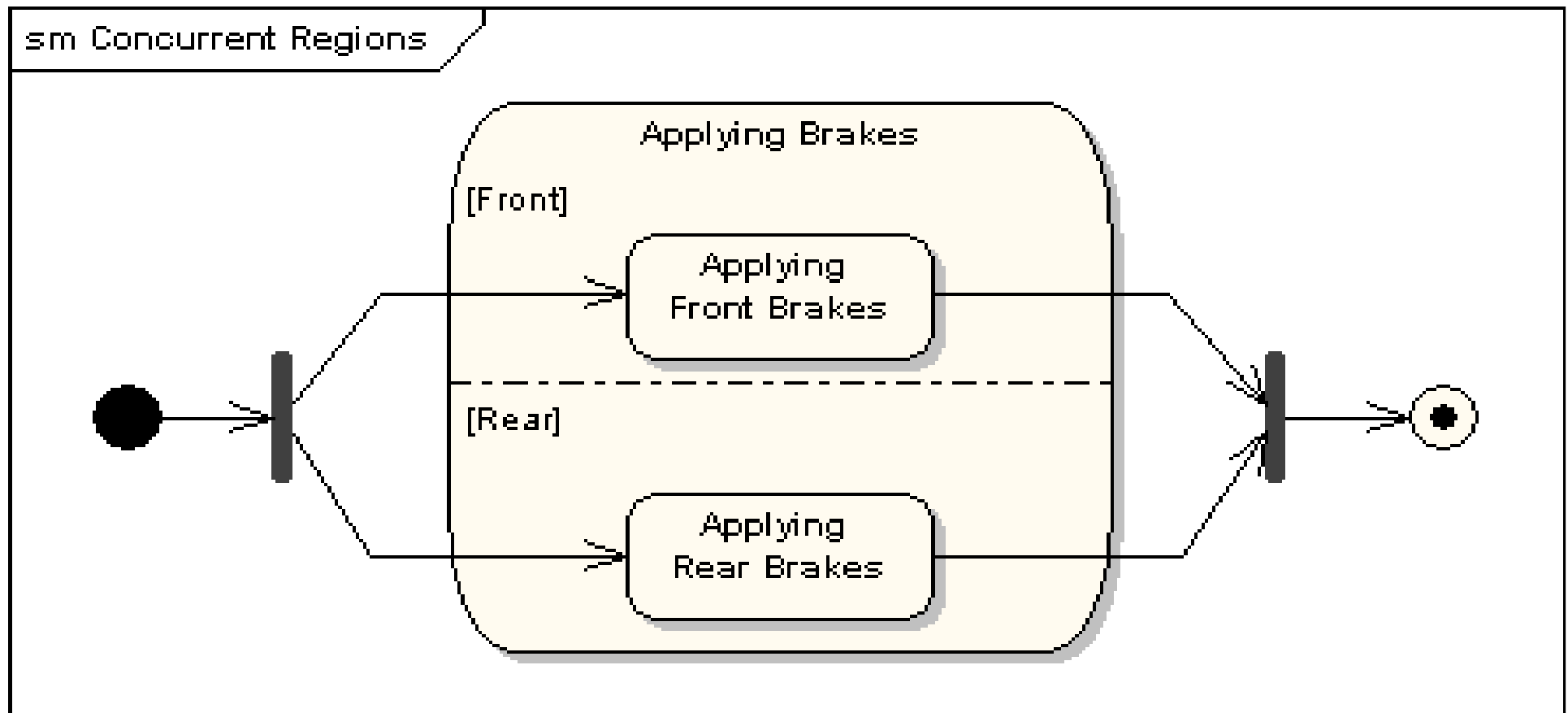


**Stany historyczne** – przedstawiają stany wcześniejsze (historyczne) przed przerwaniem działania maszyny stanowej (np. w chwili załączenia zasilania maszyna stanowa zmywarki pamięta stan, w którym ma wznowić działanie)



## Równoległe podstawy

Stan może być podzielony między równoległe podstawy wykonywane jednocześnie. (np. sterowanie przednimi (front) i tylnymi (rear) hamulcami odbywa się równoległe i musi być zsynchronizowane – wyrażone za pomocą symbolu rozdzielenia na pseudo stany oraz symbolu połączenia pseudo stanów. Równoległe podstawy są używane do modelowania synchronizacji wątków

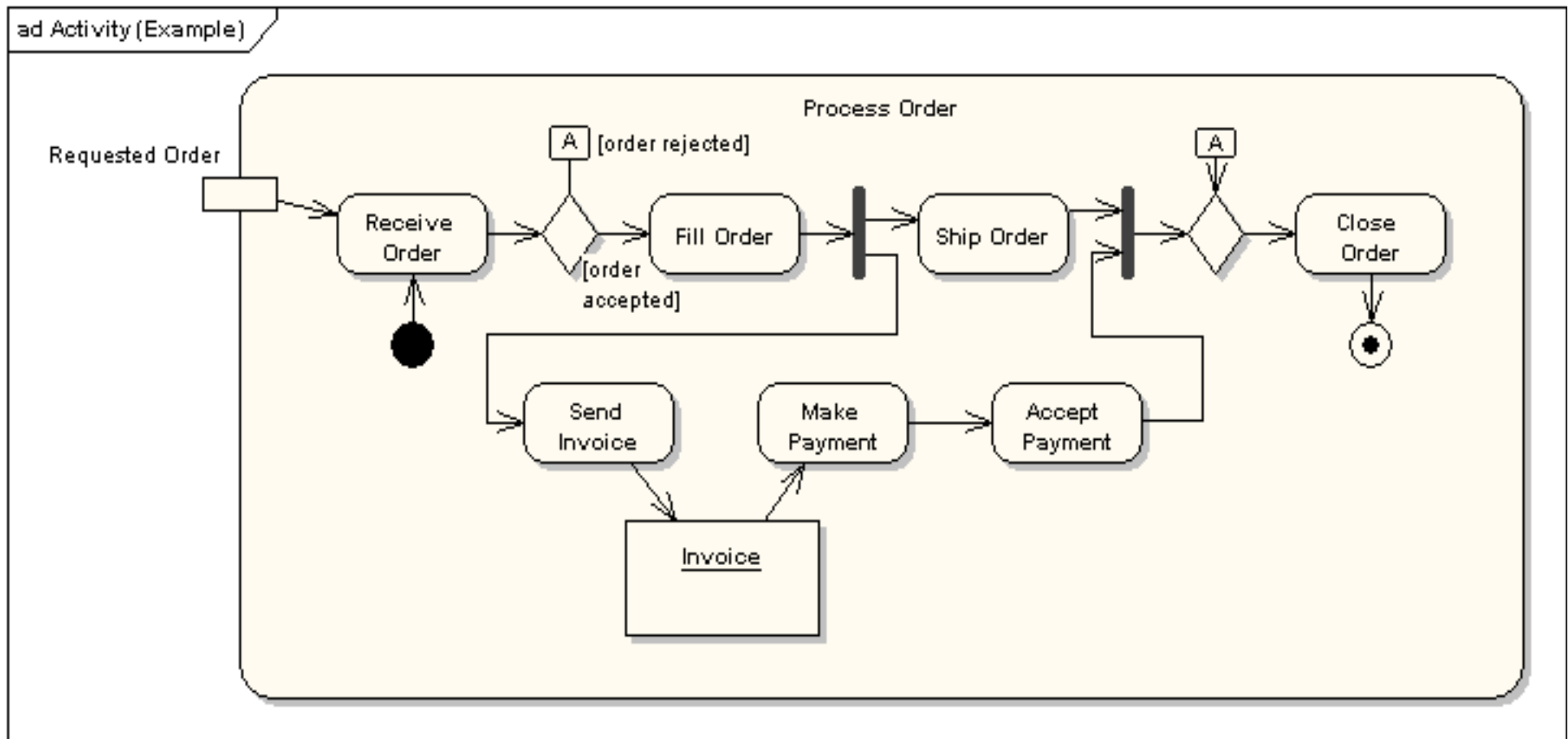


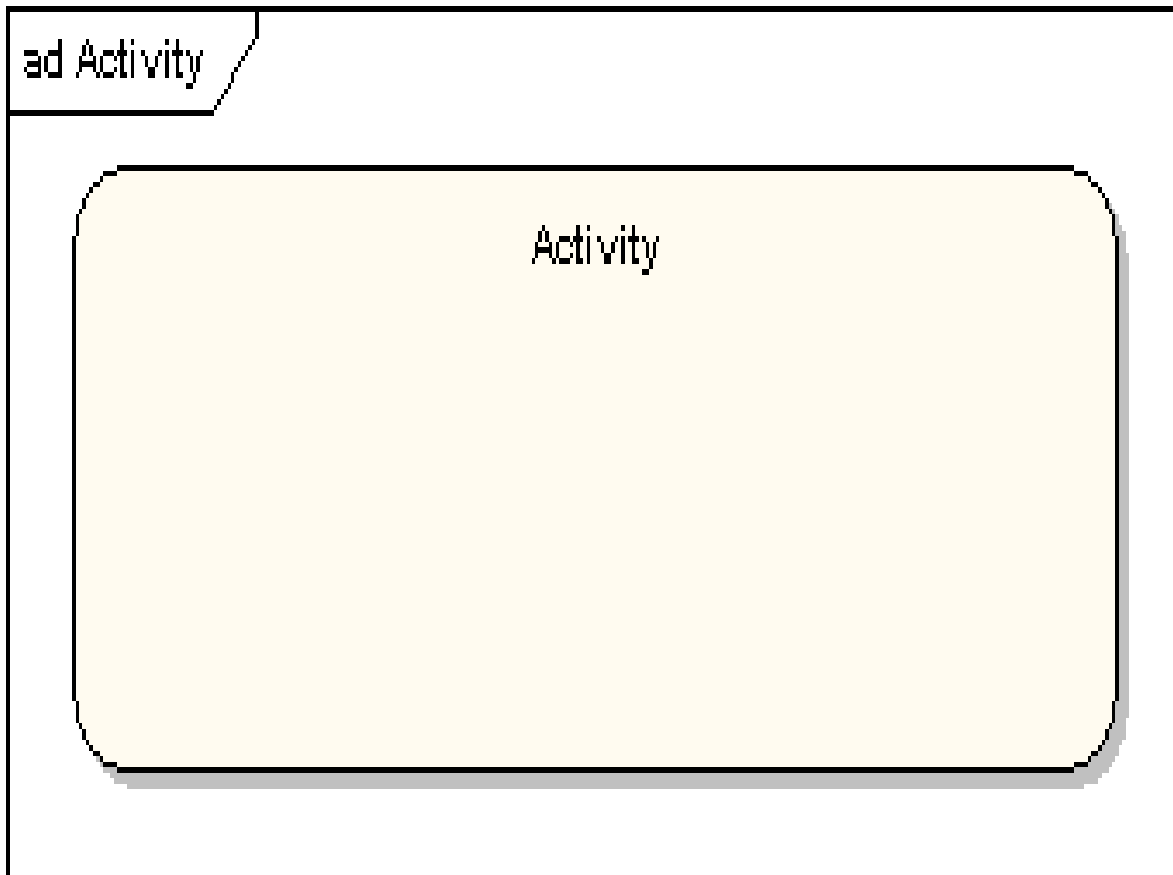


## Diagramy aktywności

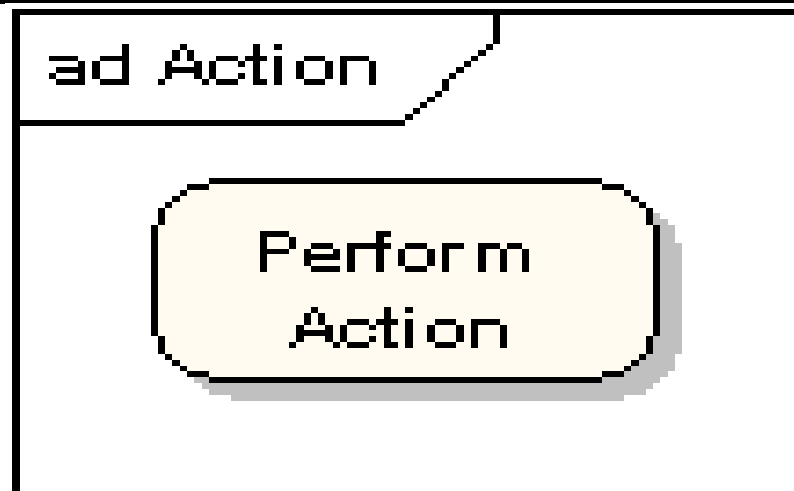
Diagram aktywności jest odmianą diagramu stanu i opisuje interakcje między obiektami.

- **jak** pobierane są operacje,
- **co** operacje wykonują (zmiana stanu obiektu),
- **kiedy** operacje są wykonywane (sekwencje akcji)
- **gdzie** są wykonywane.

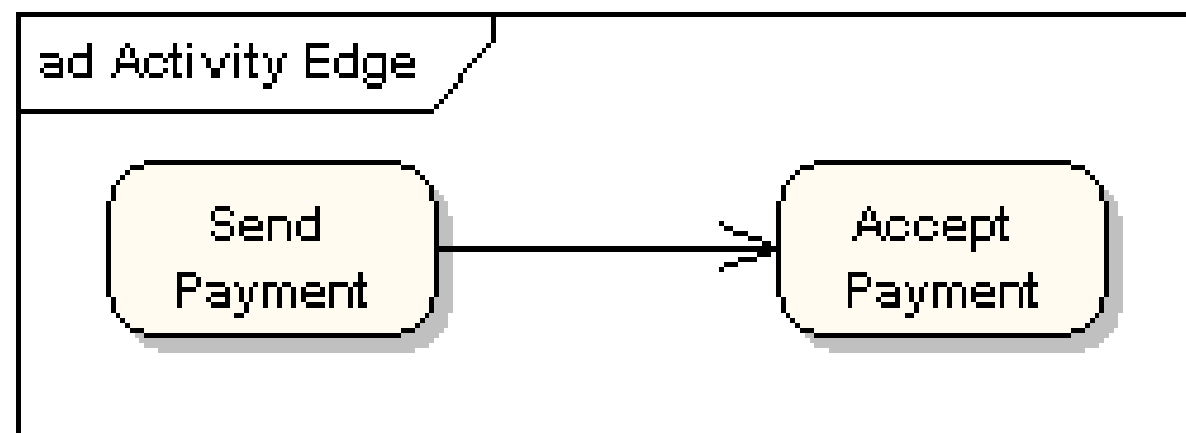
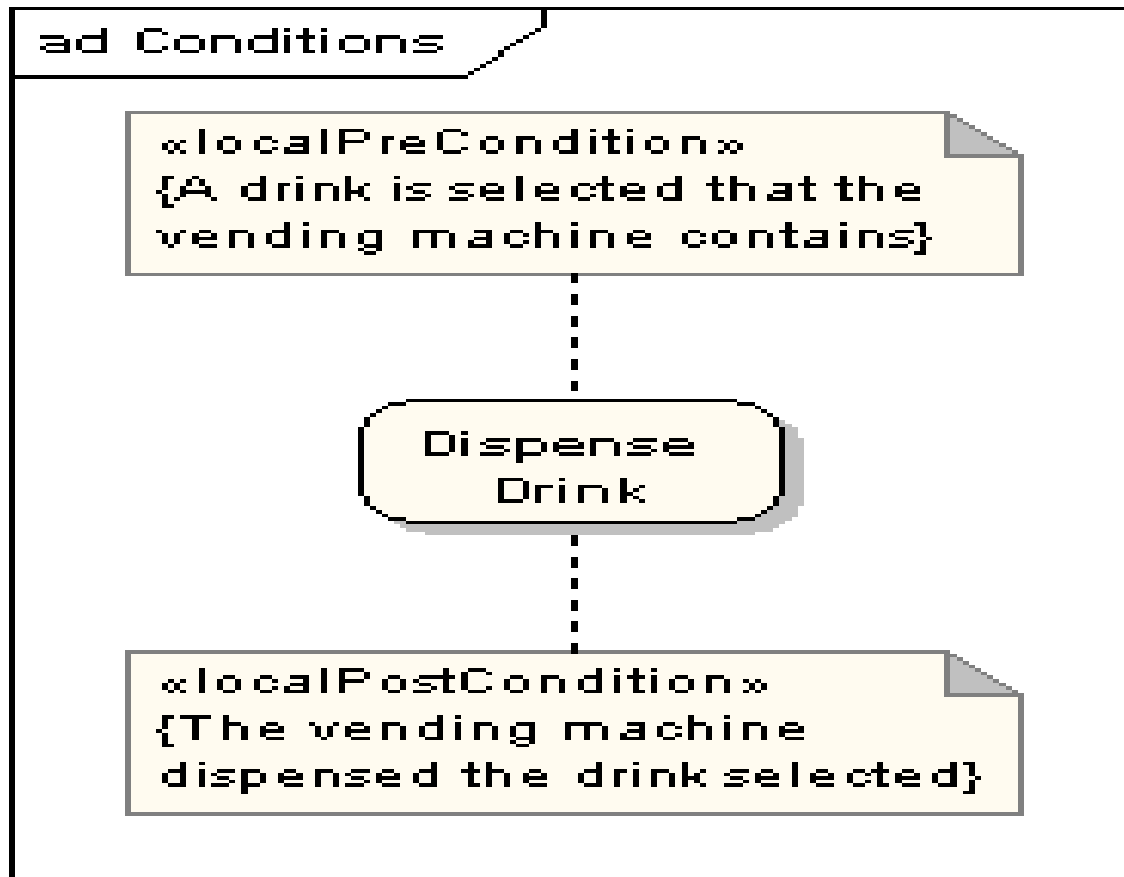




**Aktywność**- zawiera specyfikację sparametryzowanych zachowań (akcje, przepływ sterowania)



**Akcja** – elementarny krok aktywności



## Ograniczenia akcji:

warunki przed akcją i po akcji

Np.

Warunek przed: Wybór napoju w automacie, jeśli istnieje

Stan: akcja wydania napoju (Dispense drink)

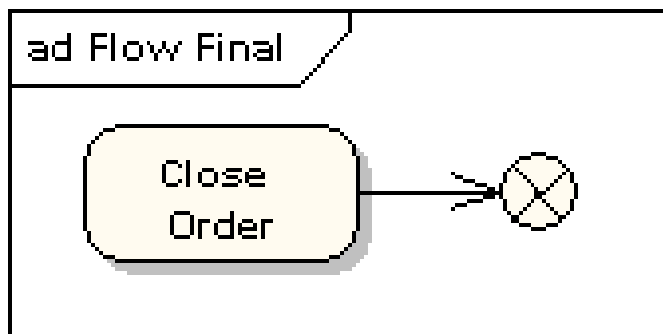
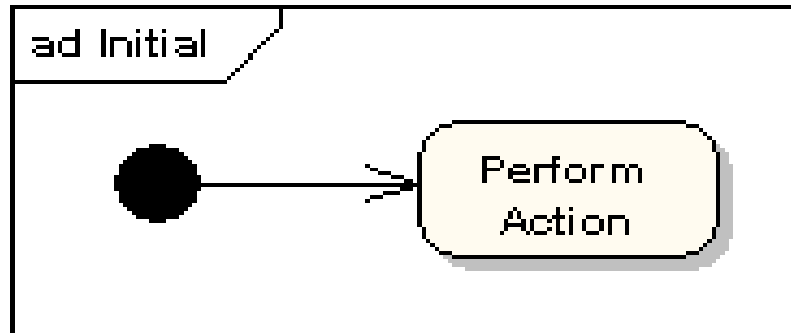
Warunek po: Maszyna wydała wybrany napój

## Przeływ sterowania:

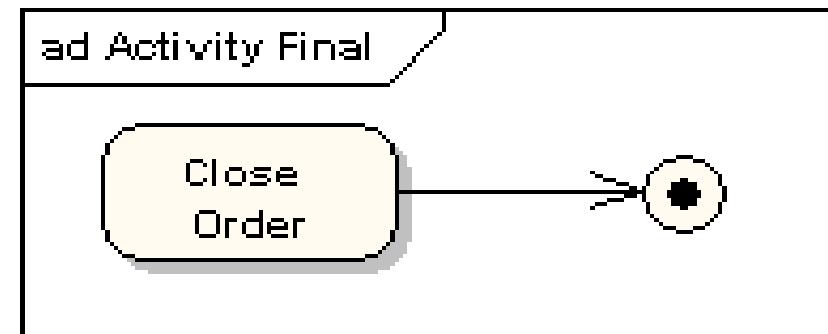
Przejście z jednej akcji do drugiej

np. wysłanie rachunku –Send Payment) i akceptacja rachunku (Accept Payment)

## Wierzchołek początkowy (Initial) i końcowy (Final)

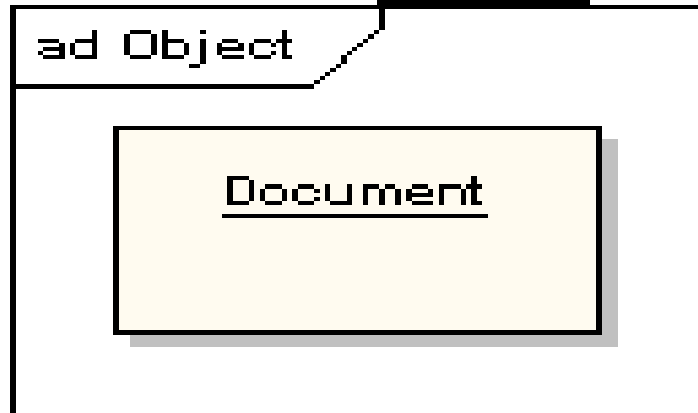


**Wierzchołek końca sterowania** koniec pojedynczego przebiegu sterowania

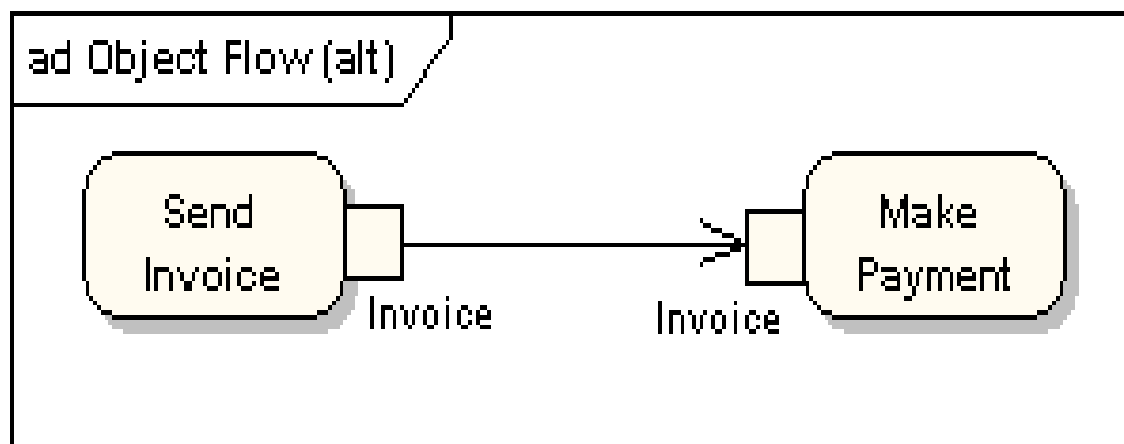
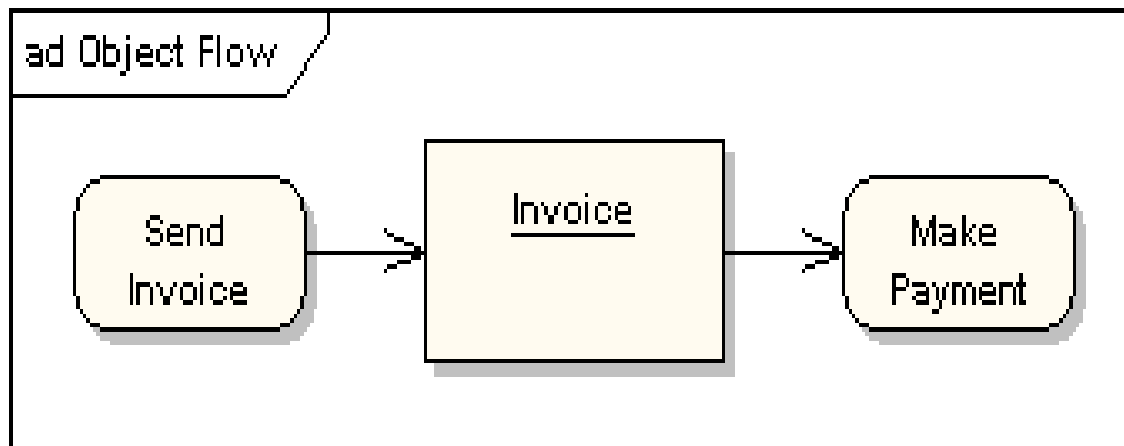


**Wierzchołek końca wielu przebiegów sterowania wewnątrz aktywności**

## Obiekt



## Magazyn danych



## Przepływ obiektów

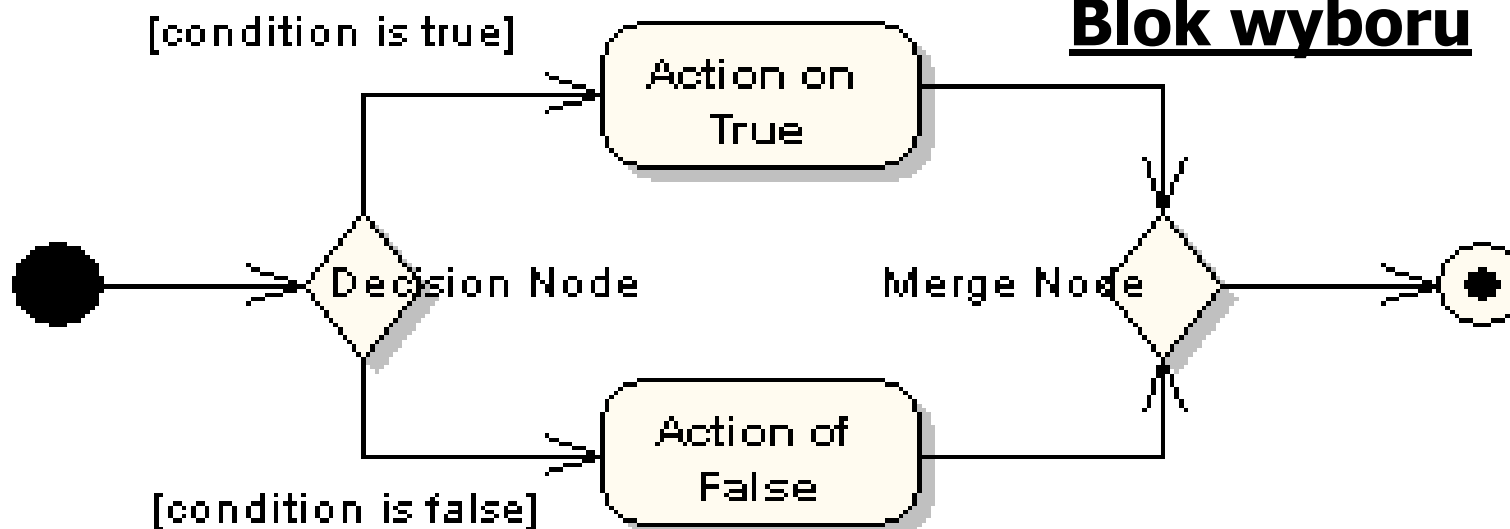
Wysłanie (Send Invoice) obiektu faktura (Invoice) w celu dokonania zapłaty (Make Payment)

## Przepływ obiektów

Równoważny diagram

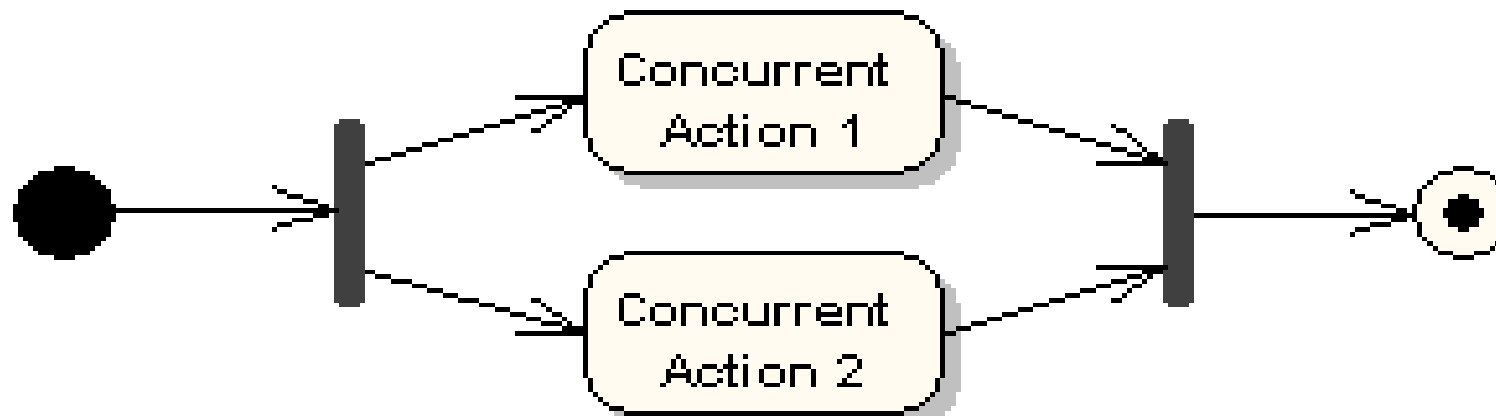
ad Decision or Merge

## Blok wyboru

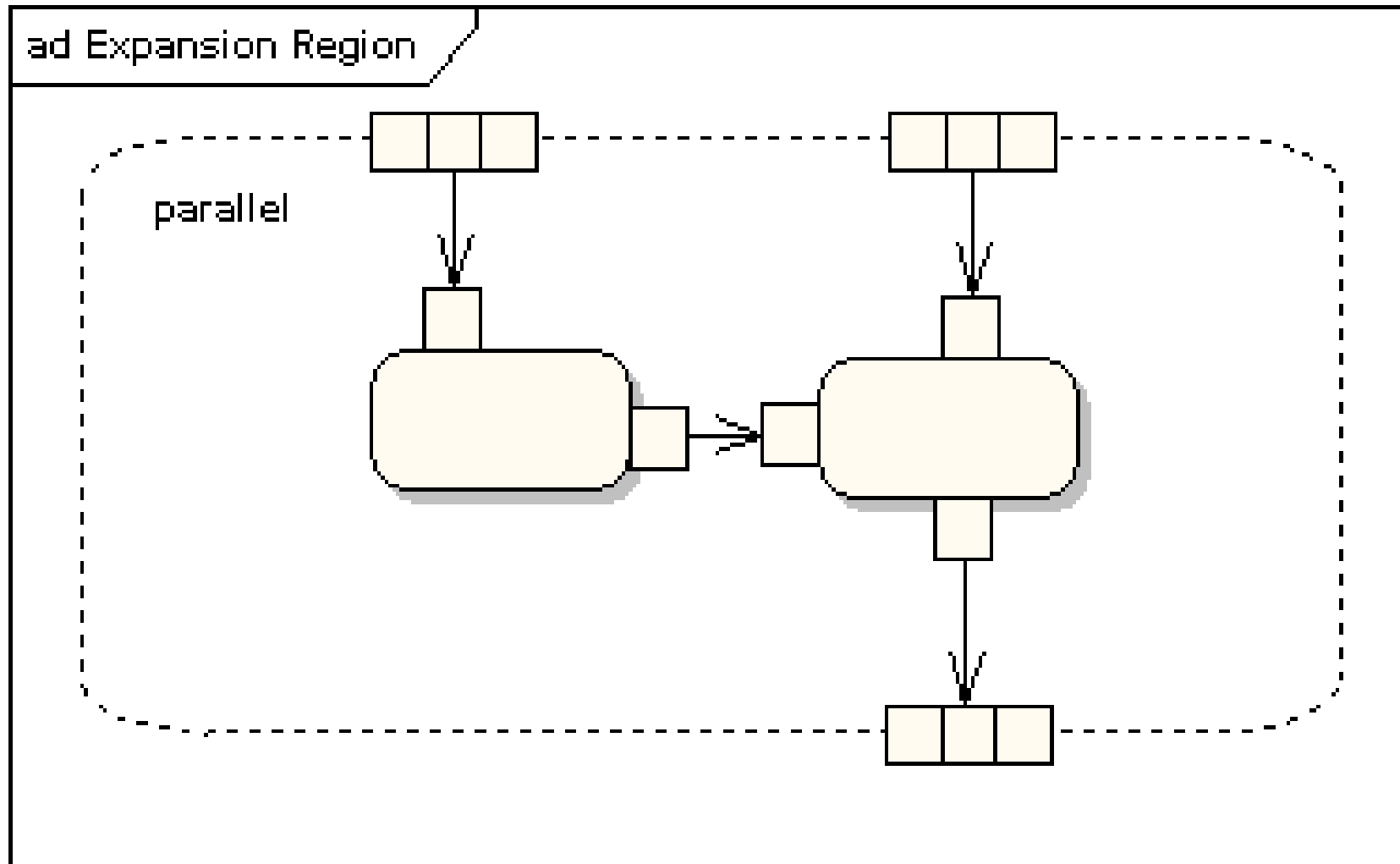


ad Fork and Join

## Bloki rozdzielania i łączenia

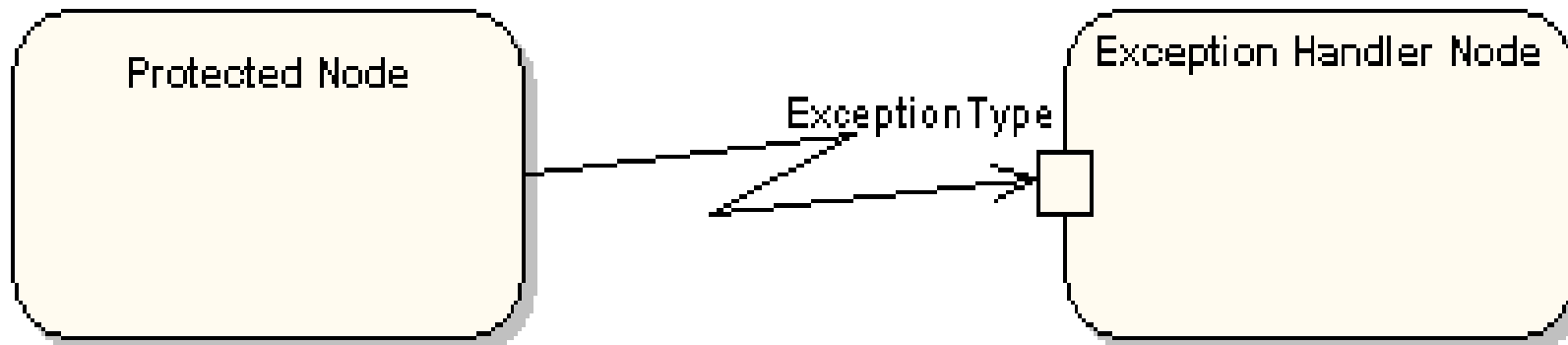


**Powtarzanie aktywności:** iteracyjnie (iterative), równoległe (parallel) lub w postaci strumienia (stream)



ad Exception Handler

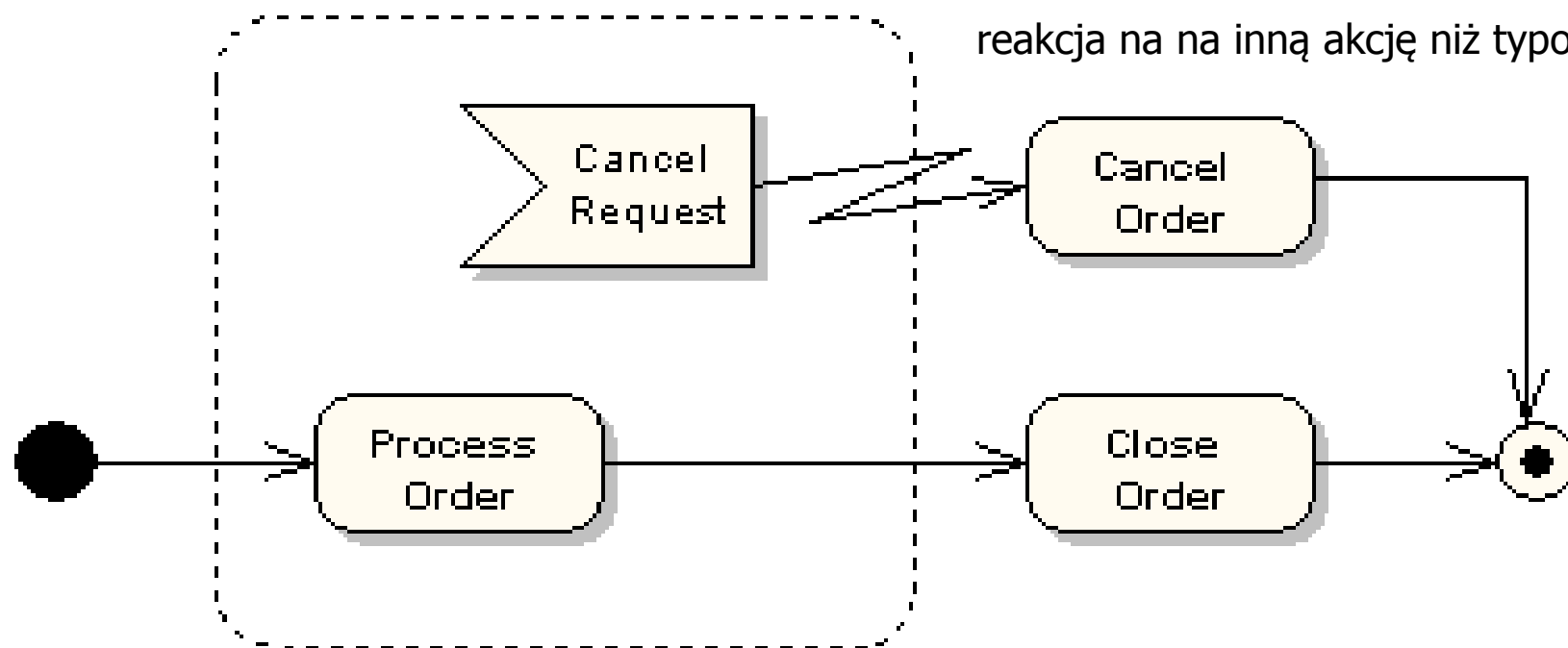
## Obsługa wyjątków reakcja na błąd



ad Interruptible Activity Region

## Obsługa przerwania

reakcja na na inną akcję niż typowa





**Partycje** – np. podział aktywności wykonywanych na obiekcie Faktura (Invoice) przez dwa różne obiekty: Wydział Finansowy (Accounting Department) i klienta (Customer).

