Prezentacja wykonana na bazie
standardów określonych przez:

BPMN™

Podstawy
Business Process Modeling Notation

Materiały i narzędzia wykorzystane podczas tworzenia niniejszej prezentacji

- **Object Management Group/Business Process Management Initiative**
  [http://www.bpmn.org/]
- **Documents Associated with Business Process Model and Notation (BPMN) 1.2**
  [http://www.omg.org/spec/BPMN/1.2/PDF]
- **Wzorce projektowe:**
  [http://is.ieis.tue.nl/research/patterns/patterns.htm]
- **BizAgi Process Modeler©**
- **Sybase PowerDesigner®**
- **Microsoft Office PowerPoint 2007®**

Wstęp

- **BPMN™** – Notacja zarządzania procesami biznesowymi (graficzny język wizualizacji, specyfikowania, tworzenia i dokumentowania procesów biznesowych)
- Główne obszary zastosowań BPMN™
  - projektowanie procedur biznesowych,
  - projektowanie systemów przepływu zadań i informacji,
  - modelowanie systemów informacyjnych.

Standardy

- **OMG** (**O**bject **M**anagement **G**roup) – międzynarodowa organizacja, której misją jest promowanie praktyki i teorii obiektowych w tworzeniu oprogramowania.

Podstawowe pojęcia

Proces to zbiór działań wzajemnie powiązanych lub wzajemnie oddziałujących, które przekształcają wejścia w wyjścia.

Proces biznesowy to zaplanowany przebieg działań mający na celu uzyskiwanie określonych rezultatów.

Mapa procesów biznesowych to obraz przedstawiający wszystkie czynności i ich agregacje, czyli procesy, jakie są realizowane wewnątrz organizacji w celu wytworzenia produktu finalnego.

Procesy biznesowe można sklasyfikować w trzy podstawowe rodzaje procesów:

- **Procesy zarządzania** – mające na celu formułowanie strategii i celów oraz kontrolowanie przebiegu procesów operacyjnych oraz pomocniczych.
- **Procesy operacyjne** – mające na celu wykonywanie typowych działań zmierzających do realizacji założonych celów i strategii.
- **Procesy pomocnicze** – mające na celu wsparcie procesów zarządzania i operacyjnych.

Podstawowe pojęcia

- **Procesy zarządzania**
  - Proces zarządzania strategią i jej realizacji
  - Proces zarządzania cyklem życia infrastruktury
  - Proces zarządzania cyklem życia produktu
- **Procesy operacyjne**
  - Proces wsparcia i zapewnia ciągłości produkcji
  - Proces dostarczania produktów i usług
  - Proces zarządzania kontaktami z klientami
  - Proces zarządzania jakością produktów i usług
  - Proces rozliczania dostarczonych produktów i usług
  - Proces zarządzania utrzymaniem klienta i budowanie lojalności
  - Proces monitorowania i raportowania
  - Proces zarządzania kontaktami z dostawcami
- **Procesy pomocnicze**
  - Proces zarządzania strategicznego i planowania przedsiębiorstwa
  - Proces zarządzania finansowego przedsiębiorstwa
  - Proces zarządzania ryzykiem przedsiębiorstwa
  - Proces zarządzania relacjami z właścicielami i otoczeniem zewnętrznym przedsiębiorstwa
  - Proces zarządzania efektywnością
  - Proces zarządzania personelem
  - Proces zarządzania wiedzą i badaniami

Podstawowe pojęcia

Reengeneering procesów biznesowych

Etapy reengineeringu:

- **Identyfikacja procesów biznesowych**, które zachodzą w firmie. Etap ten skupia się na identyfikacji procesów biorąc pod uwagę czas, koszt, znaczenie (priorytet), problemy pojawiające się przy realizacji danego procesu. Należy także zdefiniować początek i koniec procesu.
- **Wybór procesu do reengineeringu.** Firma nie jest w stanie podjąć się przeprojektowania wszystkich, czy też wielu procesów za jednym razem. Dlatego też musi wybrać określony proces.
- **Utworzenie zespołu reengineeringu.** Wynikiem tego etapu jest określenie składu grupy roboczej, której zadaniem będzie przeprowadzenie reengineeringu.
- **Analiza procesu.** Etap ten ma na celu określenie szczegółowego przebiegu procesu oraz określenie potrzeb klientów danego procesu.
- **Rekonstrukcja procesu.** Odbywa się w całkowitym oderwaniu od procesu istniejącego.
- **Wdrożenie rekonstrukcji.** Wdrożenie będzie przeprowadzane całościowo lub za pomocą projektów pilotażowych.

Podstawowe pojęcia

Korzyści z modelowania w języku BPML

- Integracja procesów biznesowych na poziomie zewnętrznym i wewnętrznym.
- Powiązanie procesów wykonywanych przez ludzi z procesami wykonywanymi przez maszyny.
- Wykorzystanie usług sieciowych w przedsiębiorstwach.
- Umieszczenie w procesach systemów typu „back-office” i na odwrót, włączenie procesów do tych systemów.
- Integracja z funkcjonującymi systemami typu „middleware”.
- Wspieranie zmian zachodzących w przedsiębiorstwie.
- Opracowanie platformy modelowania produkcji rozproszonej.
- Zainicjowanie tzw. rynku procesów.
- Wspieranie istniejących w biznesie standardów.

Elementy składowe notacji BPM

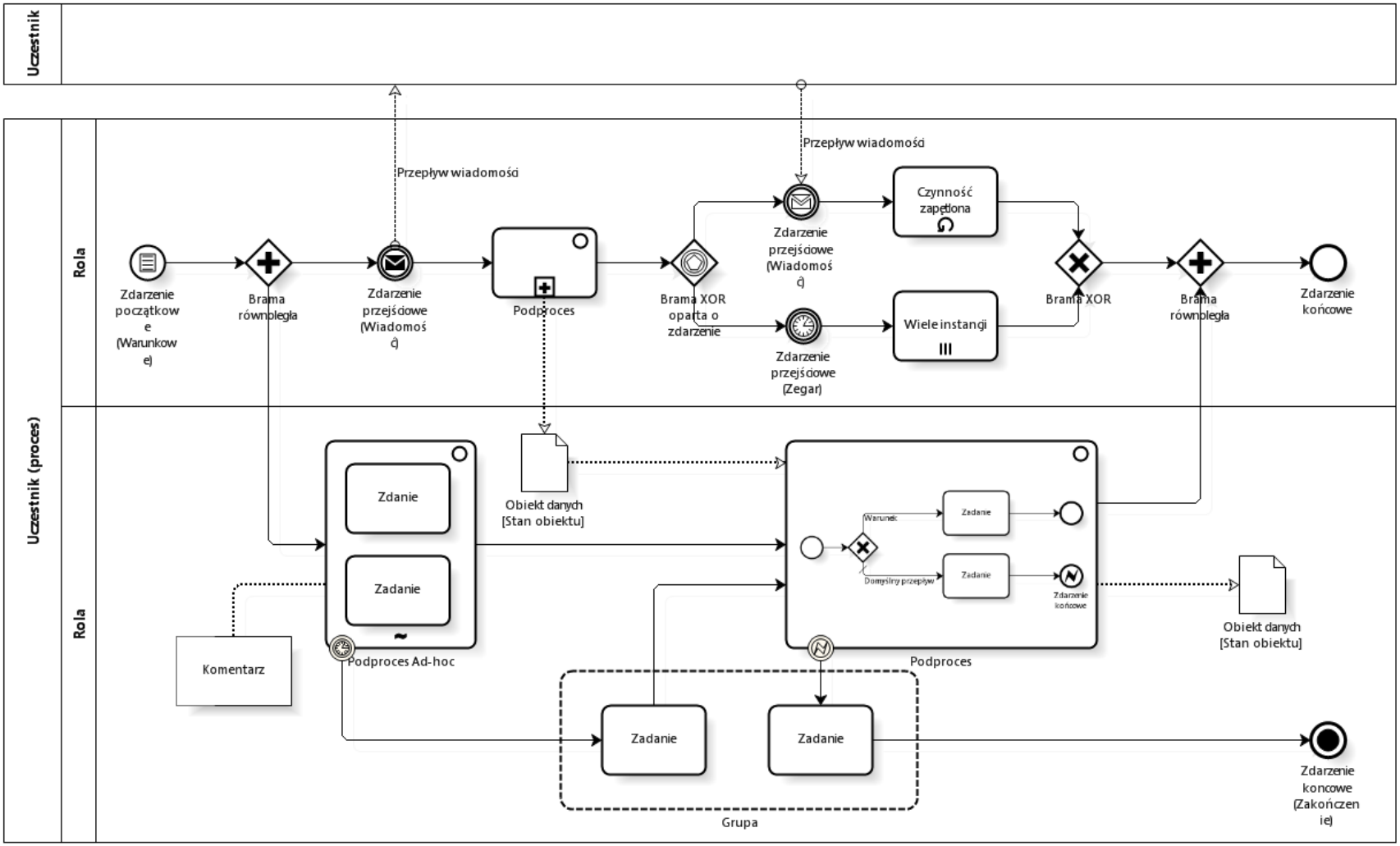
- **Podstawowe kategorie elementów:**

- **obiekty przepływu** (*Flow Objects*)
  - zdarzenia (*Events*)
  - aktywności (*Activities*)
  - bramy (*Gateways*)
- **obiekty powiązań** (*Connecting Objects*)
  - przepływy sekwencji (*Sequence Flows*)
  - przepływy wiadomości (*Message Flows*)
  - powiązania (*Associations*)
- **tory** (*Swimlanes*)
  - obszar wspólny* (*Pools*)
  - tor** (*Lanes*)
- **artefakty** (*Artifacts*)
  - obiekty danych (*Data Objects*)
  - grupy (*Groups*)
  - adnotacje (*Annotations*)

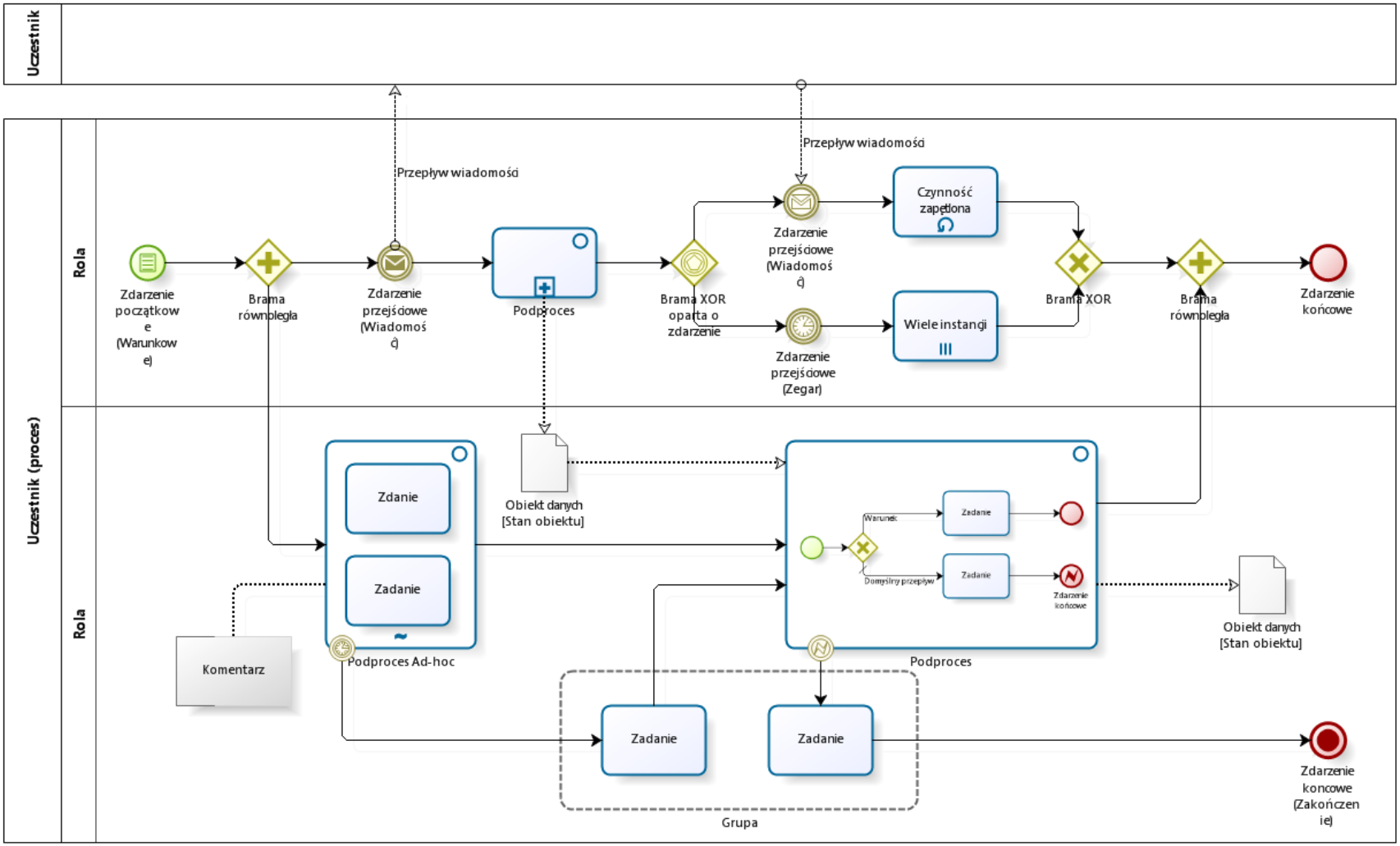
* Termin „*Pool*” oznacza w kontekście BPMN „**obszar wspólny**”. Czasami w literaturze w języku polskim jest określane jako „**uczestnik**”.

** Termin „*Lane*” oznacza w kontekście BPMN „**tor**” lub „**pas drogi**”. Czasami w literaturze w języku polskim jest określane jako „**rola**”.

Elementy składowe notacji BPM



Elementy składowe notacji BPM



Obiekty przepływu ZDARZENIA

Idea tokenów

- Ścieżka tokenów (*Token*) musi być **możliwa do prześledzenia** przez sieć przepływu sekwencji (*Sequence Flow*) tzn. nie może być żadnych niejawnych przepływów;
- Tokeny (*Tokens*) mogą być kierowane przez zdarzenia przejściowe (*Intermediate Events*) obsługujące wyjątki, **które działają jak koniec czynności**.
- Token (*Token*) nie może przechodzić przez przepływ wiadomości (*Message Flow*) ponieważ tylko wiadomości mogą być przekazywane poprzez niego.

Obiekty przepływu
ZDARZENIA

Zdarzenie pojawia się w trakcie przebiegu procesu. Ma ono wpływ na przebieg procesu. Zazwyczaj stanowi wyzwalacz lub efekt jakiegoś działania.

Rodzaje zdarzeń

- **początkowe** (startowe) – *Start Event*,
- **przejściowe** (pośrednie) – *Intermediate Event*,
- **końcowe** – *End Event*.

Obiekty przepływu
ZDARZENIA

Typy zdarzeń

- bez określonego rodzaju,
- wiadomość,
- zegar,
- błąd,
- kasowanie,
- kompensacja,
- warunkowe,
- wiązanie,
- sygnał,
- zakończenie,
- złożone.

Obiekty przepływu
ZDARZENIA

Charakter zdarzeń

- **przechwytujące** – proces odbiera zdarzenie,
- **wysyłające** – proces wysyła zdarzenie.

Obiekty przepływu
ZDARZENIA

Zestawienie rodzajów, typów i charakteru zdarzeń:

Rodzaje zdarzeń	Początkowe	Przejściowe		Końcowe
	Przechwytujące		Wyrzucające	
Wiadomość				
Zegar				
Błąd				
Kasowanie				
Kompensacja				
Warunkowe				
Powiązanie				
Sygnał				
Kończące				
Złożone				

Obiekty przepływu
ZDARZENIA

Początkowe (startowe) – *Start Event*

Reprezentacja graficzna: 

Zdarzenie początkowe (*Start Event*):

- wskazuje gdzie dany proces **zaczyna się** i w związku z tym żadna przepływ sekwencji **nie może wiązać się z nim**;
- generuje token, który musi być skonsumowany przez zdarzenie końcowe.
- jest **opcjonalne**, tzn. proces może mieć go ale nie jest on wymagany. W przypadku procesów złożonych i/lub gdy warunki rozpoczęcia nie są oczywiste zaleca się jednak, używanie zdarzenia początkowego.
- jest **niezależne** na każdym poziomie zagnieżdżenia procesów.

Obiekty przepływu
ZDARZENIA

Początkowe (startowe) – *Start Event*

- Jeżeli zdarzenie początkowe (*Start Event*) nie zostało jawnie użyte, to domyślne zdarzenie początkowe procesu nie może mieć wyzwalacza (*Trigger*).
- Jeżeli zostało użyte zdarzenie końcowe (*End Event*), to musi być **przynajmniej jedno** zdarzenie początkowe (*Start Even*).
- Jeżeli zdarzenie początkowe (*Start Event*) zostało użyte, to nie może być innych elementów przepływu, które nie posiadają przychodzących przepływów sekwencji (*Sequence Flow*) – wszystkie pozostałe obiekty przepływu muszą być celem przynajmniej jednego przepływu sekwencji (*Sequence Flow*).
  - Wyjątkiem są czynności, które zostały zdefiniowane jako czynności kompensujące (*Compensation*). Czynności kompensujące nie mogą mieć żadnych przepływów sekwencji, nawet jeżeli zostało użyte zdarzenie początkowe.
  - Drugim wyjątkiem jest zdarzenie przejściowe (*Intermediate Event*), które może pozostawać bez przychodzącego przepływu sekwencji, gdy jest przymocowane do granicy czynności.

Obiekty przepływu ZDARZENIA

Początkowe (startowe) – *Start Event*

- Jeżeli zdarzenie początkowe (*Start Event*) nie zostało użyte, to wszystkie obiekty przepływu (*Flow Objects*), które nie posiadają przychodzącego przepływu sekwencji (*Sequence Flow*) należy wytworzyć w momencie wytworzenia procesu (*Process*). Istnieje założenie, że istnieje tylko jedno domyślne zdarzenie początkowe (*Start Event*) oznaczające, że wszystkie początkowe obiekty przepływu (*Flow Objects*) rozpoczynają się w tym samym czasie.
  - Wyjątkiem są czynności, które są zdefiniowane jako czynności kompensujące (Compensation). Czynności kompensujące (*Compensation Activities*) nie są uważane jako część normalnego przepływu (*Normal Flow*) i nie mogą być wytworzone w momencie wytworzenia procesu.
- Może być **wiele zdarzeń początkowych** (*Start Events*) dla danego poziomu procesu (*Process*).

Obiekty przepływu ZDARZENIA

Początkowe (startowe) – *Start Event*

- Każde zdarzenie początkowe (*Start Event*) jest **niezależnym** zdarzeniem. Gdy zdarzenie początkowe (*Start Event*) jest wywoływane, to należy wygenerować instancję Procesu (*Process Instance*).
- Jeżeli proces (*Process*) jest użyty jako podproces (*Sub-Process*) i istnieje wiele nieoznaczonych zdarzeń początkowych (*None Start Events*), to przepływ jest **przeniesiony** z procesu rodzica do podprocesu, w którym tylko jedno zdarzenie początkowe będzie wywołane (*Triggered*). W celu określenia odpowiedniego zdarzenia początkowego atrybut (*TargetRef*) przepływu sekwencji (*Sequence Flow*) przychodzącej do podprocesu może być rozszerzony.

Obiekty przepływu
ZDARZENIA

Początkowe (startowe) – *Start Event*

Typy zdarzeń początkowych:

Wyzwalacz	Opis	Symbol
Nieokreślonego typu (*None*)	Nieokreślony typ zdarzenia. Używa się również w podprocesach, które rozpoczynają się, gdy przepływ jest wywoływany przez proces nadrzędny.	
Wiadomość (*Message*)	Wiadomość przychodzi od nadawcy i powoduje rozpoczęcie procesu.	
Zegar (*Timer*)	Określenie czasu i/lub daty lub cyklu czasowego (np. każdy poniedziałek o godz. 9:00), które powoduje rozpoczęcie procesu.	
Warunkowy (*Conditional*)	Ten typ zdarzenia jest wywoływany, gdy warunek (*Condition*) staje się prawdą. Atrybut *ConditionExpression* dla tego zdarzenia musi przyjąć wartość fałsz, a następnie prawda, aby zdarzenie mogło zostać wywołane ponownie.	
Sygnał (*Signal*)	Gdy przychodzi sygnał wyemitowany przez inny proces, to następuje zainicjowanie rozpoczęcie procesu. Sygnał (*Signal*) nie jest wiadomością (*Message*), która posiada określony cel. Wiele procesów może posiadać zdarzenia początkowe wywoływane przez ten sam wyemitowany sygnał.	
Złożony (*Multiple*)	Oznacza, że istnieje wiele sposobów rozpoczęcia procesu. Wystarczy jeden, aby tego dokonać.	

Obiekty przepływu
ZDARZENIA

Początkowe (startowe) – *Start Event*

Zasady połączeń przepływu sekwencji (*Sequence Flow*):

- Zdarzenie początkowe (*Start Event*) nie może być celem przepływu sekwencji (*Sequence Flow*) tzn. nie może posiadać sekwencji przychodzącej.
  - Wyjątkiem od tej reguły jest sytuacja, gdy zdarzenie początkowe (*Start Event*) jest użyte w rozwiniętym podprocesie (*Expanded Sub-Process*) i jest przymocowane do granicy podprocesu. W tym przypadku, przepływ sekwencji (*Sequence Flow*) z procesu wyższego poziomu może być połączona ze zdarzeniem początkowym (*Start Event*) zamiast połączenia z granicą danego podprocesu.
- Zdarzenie początkowe (*Start Event*) musi być **źródłem** przepływu sekwencji (*Sequence Flow*).
- Wiele przepływów sekwencji (*Sequence Flow*) może zostać zainicjowanych przez pojedyncze zdarzenie początkowe (*Start Event*). Dla każdej przepływu sekwencji (*Sequence Flow*), która posiada źródło w zdarzeniu początkowym (*Start Event*) zostanie utworzona oddzielna, **równoległa ścieżka**.
- Jeżeli zdarzenie początkowe (*Start Event*) nie zostało użyte, to wszystkie obiekty przepływu (*Flow Objects*), które nie posiadają przychodzącej przepływu sekwencji (*Sequence Flow*), rozpoczynają się w oddzielnych, **równoległych ścieżkach**.
- Każda ścieżka posiada **oddzielny**, unikalny token (*Token*), który przesuwa się zgodnie z sekwencją przepływu (*Sequence Flow*).

Obiekty przepływu ZDARZENIA

Początkowe (startowe) – *Start Event*

Zasady połączeń przepływu wiadomości (*Message Flow*):

- Wszystkie przepływy wiadomości (*Message Flow*) muszą łączyć oddzielne obszary wspólne (*Pools*). Mogą one być połączone z granicą obszaru wspólnego (*Pool*) lub z obiektem przepływu (*Flow Object*) w granicach tego obszaru, jednak nie mogą łączyć dwóch obiektów wewnątrz tego samego obszaru wspólnego (*Pool*).
- Zdarzenie początkowe (*Start Event*) może być celem przepływu wiadomości (*Message Flow*). Może ono posiadać zero lub więcej przychodzących przepływów wiadomości (*Message Flow*). Każdy z przepływów wiadomości (*Message Flow*) przychodzący do zdarzenia początkowego (*Start Event*) reprezentuje mechanizm inicjowania procesu. Wystarczy **jeden** z wyzwalaczy (*Triggers*), aby rozpocząć nowy proces.
- Zdarzenie początkowe (*Start Event*) nie może być źródłem przepływu wiadomości (*Message Flow*) tzn. nie może posiadać wychodzącego przepływu wiadomości (*Message Flow*).

Obiekty przepływu
ZDARZENIA

Końcowe – *End Event*

Reprezentacja graficzna: 

Zdarzenie końcowe (*End Event*):

- określa miejsce gdzie proces się kończy. W sensie przepływu sekwencji (*Sequence Flow*), zdarzenie końcowe (*End Event*) kończy przepływ procesu i z tego powodu nie posiada żadnego wychodzącego przepływu sekwencji (*Sequence Flow*);
- konsumuje token (*Token*), który został wygenerowany przez zdarzenie początkowe (*Start Event*) znajdujące się na tym samym poziomie zagnieżdżenia procesów. Jeżeli równoległy przepływ sekwencji (*Sequence Flow*) kończy się zdarzeniem końcowym (*End Event*), to tokeny (*Tokens*) będą konsumowane zgodnie z kolejnością ich przybywania. Wszystkie tokeny (*Tokens*), które zostały wygenerowane musza być skonsumowane przez zdarzenie końcowe (*End Event*) nim proces zostanie zakończony.

Obiekty przepływu
ZDARZENIA

Końcowe – *End Event*

Uwaga w związku z innymi okolicznościami:

- Jeżeli proces jest podprocesem, to może zostać zatrzymany przed normalnym zakończeniem przez **przerwania zdarzeń przejściowych** (*Intermediate Events*). W tej sytuacji tokeny (*Tokens*) zostaną **skonsumowane** przez **zdarzenie przejściowe** (*Intermediate Event*) przymocowane do granic podprocesu.

Obiekty przepływu
ZDARZENIA

Końcowe – *End Event*

- Może istnieć wiele zdarzeń końcowych (*End Event*) na danym poziomie procesu.
- Zdarzeni końcowe (*End Event*) jest **opcjonalne**.
- Jeżeli zdarzenie końcowe (*End Event*) nie zostało użyte, to domyślne zdarzeni końcowe (*End Event*) procesu nie może mieć wyniku (*Result*).
- Jeżeli użyto zdarzenia początkowego (*Start Event*), to musi istnieć przynajmniej jedno zdarzeni końcowe (*End Event*).
- Jeżeli użyto zdarzenia końcowego (*End Event*), to nie może być innych elementów przepływu, które nie posiadają żadnego wychodzącego przepływu sekwencji (*Sequence Flow*) tzn. wszystkie inne obiekty przepływu (*Flow Objects*) muszą być źródłem przynajmniej jednego przepływu sekwencji (*Sequence Flow*).
  - Wyjątek stanowią czynności zdefiniowane jako kompensacje (*Compensation*). Czynności kompensujące (*Compensation Activities*) nie mogą posiadać żadnego wychodzącego przepływu sekwencji (*Sequence Flow*), nawet jeżeli istnieje zdarzenie końcowe (*End Event*) na danym poziomie procesu.

Obiekty przepływu
ZDARZENIA

Końcowe – *End Event*

- Jeżeli zdarzenie końcowe (*End Event*) nie zostało użyte, to wszystkie obiekty przepływu (*Flow Objects*), które nie posiadają żadnego wychodzącego przepływu sekwencji (*Sequence Flow*) oznaczają koniec ścieżki w procesie, jednak proces nie może się zakończyć dopóki **wszystkie ścieżki równoległe** nie zostaną **zakończone**.
  - Wyjątek stanowią czynności zdefiniowane jako kompensacje (*Compensation*). Czynności kompensacyjne (*Compensation Activities*) nie są uważane jako część normalnego przepływu (*Normal Flow*) i nie oznaczają końca procesu.
- BPD (*Business Process Diagram*) może posiadać więcej niż jeden poziom procesów i na każdym z nich użycie zdarzeń początkowych (*Start Events*) i zdarzeń końcowych (*End Events*) jest **niezależne**.

Obiekty przepływu
ZDARZENIA

Końcowe – *End Event*

Typy zdarzeń końcowych: (1 / 2)

Wyzwalacz	Opis	Symbol
Nieokreślonego typu (*None*)	Nieokreślony typ zdarzenia. Używa się również w podprocesach, które gdy kończą się, przepływ powraca do procesu nadrzędnego.	
Wiadomość (*Message*)	Wiadomość wychodzi jako wynik procesu.	
Błąd (*Error*)	Oznacza, że nazwany błąd powinien zostać wygenerowany. Błąd zostanie przechwycony przez zdarzenie przejściowe błędu (*Error Intermediate Event*), które znajduje się na granicy najbliższej (w hierarchii) otaczającej czynności nadrzędnej. Zachowanie procesu jest nieokreślone jeżeli żadna z czynności w procesie nie posiada w/w zdarzenia przejściowego błędu (*Error Intermediate Event*).	
Kasowanie (*Cancel*)	Jest używany z podprocesem transakcji (*Transaction Sub-Process*). Wskazuje, że transakcja powinna zostać skasowana i wywołuje zdarzenie pośrednie kasowania (*Cancel Intermediate Event*) przymocowane do granicy podprocesu (*Sub-Process*).	

Obiekty przepływu
ZDARZENIA

Końcowe – *End Event*

Typy zdarzeń końcowych: (2/2)

Wyzwalacz	Opis	Symbol
Kompensacja (*Compensation*)	Wskazuje, że konieczna jest kompensacja. Jeżeli czynność podlegająca kompensacji jest zidentyfikowana, to jest ona kompensowana. W przeciwnym wypadku wszystkie zakończone czynności wewnątrz procesu, rozpoczynając od najwyższego poziomu oraz łącznie ze wszystkimi podprocesami, podlegają kompensacji przebiegającej w odwrotnym kierunku. Aby kompensacja była możliwa czynność musi posiadać zdarzenie pośrednie kompensacji (*Compensation Intermediate Event*) przymocowane do jej granic.	
Sygnał (*Signal*)	Gdy zostanie wywołane to zdarzenie, to nadany jest sygnał końca. Sygnał może być wysłany do jakiegokolwiek procesu, który jest w stanie odbierać ten sygnał na wskroś poziomów procesów lub obszarów wspólnych (*Pools*). Sygnał nie jest wiadomością, która w odróżnieniu od niego posiada określonego nadawcę i odbiorcę.	
Zakończenie (*Terminate*)	Oznacza, że wszystkie czynności w procesie powinny zostać natychmiast zakończone. Włączając wszystkie wystąpienia i złożone wystąpienia czynności. Proces jest zakończony bez kompensacji lub przechwycenia zdarzenia.	
Złożony (*Multiple*)	Oznacza, że istnieje wiele konsekwencji zakończenia procesu i wszystkie one występują. Atrybuty zdarzenia końcowego (*End Event*) definiuje które typy wyników zastosować.	

Obiekty przepływu
ZDARZENIA

Końcowe – *End Event*

Zasady połączeń przepływu sekwencji (*Sequence Flow*):

- Zdarzenie końcowe (*End Event*) musi być celem przepływu sekwencji (*Sequence Flow*) tzn. musi posiadać sekwencję przychodzącą. Może istnieć wiele przychodzących przepływów sekwencji (*Sequence Flow*) do danego zdarzenia końcowego (*End Event*).
- Przepływ może przychodzić zarówno od alternatywnych jaki równoległych ścieżek. Dla wygody modelowania, każda ścieżka może łączyć się z oddzielnym zdarzeniem końcowym (*End Event*).
- Zdarzenie końcowe jest używane jako ujście dla wszystkich tokenów (*Tokens*), które przybywają do niego. Proces jest w stanie **przebiegu** aż do momentu aż **wszystkie tokeny** (*Tokens*) nie zostaną skonsumowane.
- Zdarzenie końcowe (*End Event*) nie może być źródłem przepływu sekwencji (*Sequence Flow*).
  - Wyjątkiem od tej reguły jest sytuacja, gdy zdarzenie końcowe (*End Event*) jest użyte w rozwiniętym podprocesie (*Expanded Sub-Process*) i jest przymocowane do granicy tego podprocesu. W tym przypadku, przepływ sekwencji (*Sequence Flow*) z procesu wyższego poziomu może być połączona ze zdarzeniem końcowym (*End Event*) zamiast połączenia z granicą danego podprocesu.

Obiekty przepływu
ZDARZENIA

Końcowe – *End Event*

Zasady połączeń przepływu wiadomości (*Message Flow*):

- Wszystkie przepływy wiadomości (*Message Flow*) muszą łączyć oddzielne obszary wspólne (*Pools*). Mogą one być połączone z granicą obszaru wspólnego (*Pool*) lub z obiektem przepływu (*Flow Object*) w granicach tego obszaru, jednak nie mogą łączyć dwóch obiektów wewnątrz tego samego obszaru wspólnego (*Pool*).
- Zdarzenie końcowe (*End Event*) nie może być celem przepływu wiadomości (*Message Flow*) tzn. nie może ono posiadać przychodzących przepływów wiadomości (*Message Flow*).

Obiekty przepływu
ZDARZENIA

Przejściowe – *Intermediate Event*

Reprezentacja graficzna: 

Zdarzenie przejściowe (*Intermediate Event*):

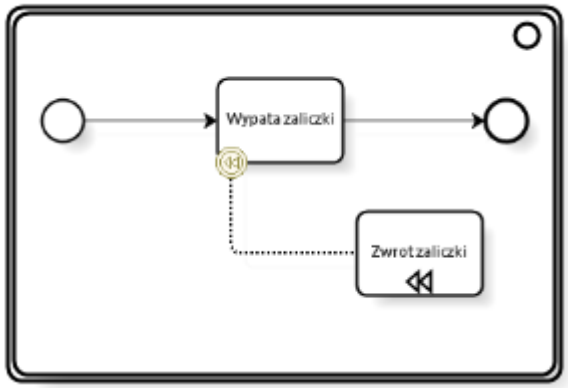
- określa miejsce w którym występuje zdarzenie pomiędzy początkiem, a końcem procesu. Wpływa na proces ale bezpośrednio go nie zaczyna, ani nie kończy.
- wskazuje gdzie wiadomości są oczekiwane lub wysyłane w procesie;
- wskazuje miejsce, w którym spodziewane są opóźnienia,
- przerywa normalny przepływ (*Normal Flow*) poprzez obsłużenie przerwania,
- wskazuje dodatkową pracę wymaganą dla kompensacji.

Obiekty przepływu
ZDARZENIA

Przejściowe – *Intermediate Event*

Obsługa wyjątków i kompensacji:

Obsługa **wyjątków** i **kompensacji** jest reprezentowana graficznie poprzez umieszczenie zdarzenia przejściowego **przymocowanego do granicy zadania** (*Task*) lub podprocesu (*Sub-Process*).



Obiekty przepływu ZDARZENIA

Przejściowe – *Intermediate Event*

Wywoływanie zdarzeń pośrednich:

Zdarzenie pośrednie (*Intermediate Event*) umieszczone wewnątrz normalnego przepływu może być użyte w jednym z dwóch celów. Może ono:

- reagować na przechwycenie wywołania zdarzenia (*Event Trigger*) lub
- wysłać wywołanie zdarzenia (*Event Trigger*).

Gdy token (*Token*) dociera do zdarzenia przejściowego (*Intermediate Event*) umieszczonego w normalnym przepływie procesu, to:

- dla zdarzeń o charakterze **przechwytującym**: **pozostaje w zdarzeniu** aż do momentu pojawieniu się w nim wywołania, np. wiadomości, następnie token przemieszcza się do wychodzącego przepływu sekwencji (*Sequence Flow*);
- dla zdarzeń o charakterze **wysyłającym**: wywołanie pojawia się natychmiast, np. wysyłana jest wiadomość, a następnie token (*Token*) **przemieszcza się** do wychodzącego przepływu sekwencji (*Sequence Flow*).

Obiekty przepływu
ZDARZENIA

Przejściowe – *Intermediate Event*

Typy zdarzeń przejściowych: (1/4)

Wyzwalacz	Opis	Symbol
Nieokreślonego typu (*None*)	Nieokreślony typ zdarzenia. Można go używać tylko w głównym przepływie procesu. Jest używany w celu zamodelowania zdarzeń wskazujących jakieś zmiany w procesie.	
Wiadomość (*Message*)	Zdarzenie przechwytujące: proces jest zatrzymany do momentu przybycia wiadomości. Po przybyciu jest kontynuowany. Zdarzenie wysyłające: Wiadomość jest wysyłana i proces jest natychmiast kontynuowany. Jeżeli zdarzenie jest użyte w celu obsłużenia wyjątku, to normalny przepływ (*Normal Flow*) jest zmieniany w przepływ wyjątku (*Exception Flow*).	 
Zegar (*Timer*)	Określa czas lub cykliczność. Jeżeli został użyty wewnątrz głównego przepływu, to pełni rolę opóźnienia. Jeżeli został użyty w celu obsłużenia wyjątku, to normalny przepływ (*Normal Flow*) jest zmieniany w przepływ wyjątku (*Exception Flow*).	
Błąd (*Error*)	Ten typ wyjątku może być tylko przypięty do granicy aktywności. Przechwytuje nazwany błąd lub jakikolwiek błąd jeżeli nazwa błędu nie została określona.	
Kasowanie (*Cancel*)	Ten typ wyjątku jest używany dla podprocesu będącego transakcją (*Transaction Sub-Process*). Musi on być przymocowany do granicy podprocesu. Jest wywoływany, gdy podproces transakcyjny osiągnie końcowe zdarzenie kasowania (*Cancel End Event*) lub gdy wiadomość „*Cancel*" protokołu transakcyjnego (*Transaction Protocol*) zostanie otrzymana w czasie przebiegu transakcji.	

Obiekty przepływu
ZDARZENIA

Przejściowe – *Intermediate Event*

Typy zdarzeń przejściowych: (2/4)

Wyzwalacz	Opis	Symbol
Kompensacja (*Compensation*)	Jest używany zarówno przy aktywacji jak i wykonywaniu kompensacji. W normalnym przepływie (*Normal Flow*) oznacza, że konieczna jest kompensacja, więc jest zdarzeniem o charakterze wysyłającym. Jeżeli zdarzenie dotyczy aktywności, to tylko ta aktywność i nic więcej nie zostanie skompensowane. W przeciwnym przypadku polecenie kompensacji jest emitowane do wszystkich aktywności, które zostały zakończone wewnątrz danego procesu, włączając proces najwyższego poziomu i wszystkie zawierające się w nim podprocesy. Wszystkie podlegające kompensacji aktywności zostaną skompensowane w odwrotnej kolejności w stosunku do ich zakończenia. Aby możliwe było skompensowanie danej aktywności konieczne jest przejściowe zdarzenie kompensacji (*Compensation Intermediate Event*) przymocowane do jej granicy. Gdy zdarzenie kompensacji jest przymocowane do granicy aktywności, to zdarzenie będzie wywołane przez kompensację o charakterze wysyłającym, w którym określono tą aktywność lub przez kompensację emitującą. W przypadku, gdy użyta jest kompensacja o charakterze przechwytującym, jeżeli zdarzenie jest wywołane, to aktywność kompensująca (*Compensation Activity*) powiązana (*Associated*) ze zdarzeniem zostanie wykonana.	
Warunkowy (*Conditional*)	Ten typ zdarzenia jest wywołany, gdy warunek (*Condition*) przyjmie wartość prawda.	

Obiekty przepływu
ZDARZENIA

Przejściowe – *Intermediate Event*

Typy zdarzeń przejściowych: (3/4)

Wyzwalacz	Opis	Symbol
Wiązanie (*Link*)	Jest to mechanizm wiązania dwóch sekcji procesu. Zdarzenie wiązania może być użyte w celu realizacji zapętlenia lub uniknięcia długich linii przepływu sekwencji (*Sequence Flow*). Wiązanie może być używane tylko na pojedynczym poziomie procesu tzn. nie może łączyć np. procesu nadrzędnego i podprocesu. Pary wiążących zdarzeń przejściowych (*Intermediate Event*) mogą być używane jako łączniki międzystronicowe. Wiązanie pełni też rolę specyficznego polecenia „Go To" wewnątrz danego poziomu procesu. Może istnieć wiele wysyłających zdarzeń wiązania ale może być tylko jedno o charakterze przechwytującym.	
Sygnał (*Signal*)	Ten typ zdarzenia jest używany w celu wysyłania lub odbierania sygnałów. Sygnał służy komunikacji zarówno wewnątrz danego poziomu procesu jak i pomiędzy różnymi jego poziomami. Używany jest również do komunikacji między obszarami wspólnymi (*Pools*) oraz BPD. Sygnał nie ma sprecyzowanego odbiorcy. Gdy zdarzenie jest przymocowane do granicy aktywności, to wtedy może tylko przechwytywać sygnały w odróżnieniu od zdarzeń w normalnym przepływie (*Normal Flow*), które mogą być zarówno źródłem jak i odbiorcą sygnałów. Zdarzenie sygnałowe (*Signal Event*) różni się od zdarzenia błędu (*Error Event*) tym, że sygnał określa się bardziej ogólne jako niedotyczące błędu przerwanie aktywności oraz ma szerszy zasięg niż zdarzenia błędu (*Error Events*).	

Obiekty przepływu
ZDARZENIA

Przejściowe – *Intermediate Event*

Typy zdarzeń przejściowych: (4/4)

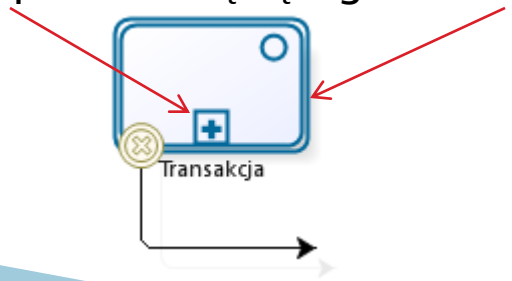
Wyzwalacz	Opis	Symbol
Złożony (*Multiple*)	Oznacza, że wiele wyzwalaczy jest przypisanych do tego zdarzenia. Gdy zdarzenie jest przymocowane do granicy aktywności, to wtedy może tylko przechwytywać wyzwalacze w odróżnieniu od zdarzeń w normalnym przepływie (*Normal Flow*), które mogą zarówno być źródłem wyzwalaczy jak i ich odbiorcą. W przypadku zdarzenia o charakterze przechwytującym wystarczy tylko jeden wyzwalacz w celu jego aktywowania. Dla zdarzenia o charakterze wysyłającym wszystkie przypisane wyzwalacze zostają wysłane, gdy przepływ dotrze do nie niego.	

Obiekty przepływu
ZDARZENIA

Przejściowe – *Intermediate Event*

Zdarzenia przymocowane do granic aktywności

- Może być więcej niż jedno zdarzenie przymocowane do granicy danej aktywności.
- Do granic aktywności mogą być przymocowane zdarzenia następujących typów:
  - Wiadomość (*Message*),
  - Zegar (*Timer*),
  - Błąd (*Error*),
  - Kasowanie (*Cancel*),
  - Kompensacja (*Compensation*),
  - Warunkowe (*Conditional*),
  - Sygnał (*Signal*),
  - Złożone (*Multiple*).
- Zdarzenie przejściowe kasowania (*Cancel Intermediate Event*) wolno przymocować tylko do podprocesu będącego transakcją.



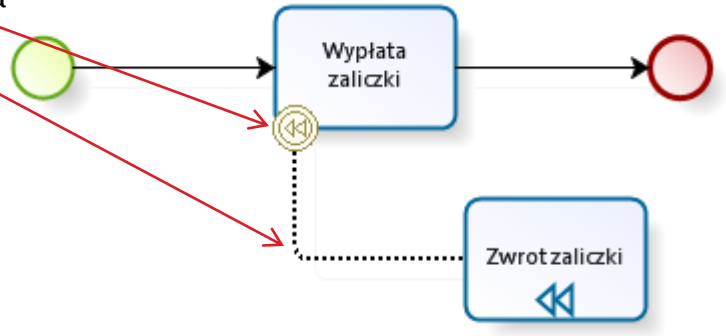
Obiekty przepływu ZDARZENIA

Przejściowe – *Intermediate Event*

Zasady połączeń przepływu sekwencji (*Sequence Flow*):

- Jeżeli zdarzenie przejściowe jest przymocowane do granicy aktywności to:
  - nie może być celem przepływu sekwencji (*Sequence Flow*)
  - musi być źródłem przepływu sekwencji (*Sequence Flow*)

- Wyjątek: Zdarzenie przejściowe (*Intermediate Event*) z wyzwalaczem kompensacyjnym (*Compensation Trigger*) nie może mieć wychodzącego przepływu sekwencji (*Sequence Flow*), może posiadać wychodzące powiąza



Obiekty przepływu ZDARZENIA

Przejściowe – *Intermediate Event*

Zasady połączeń przepływu sekwencji (*Sequence Flow*):

- Następujące zdarzenia przejściowe (*Intemediate Events*) mogą być użyte w normalnym przepływie (*Normal Flow*):
  - Nieokreślonego typu (*None*),
  - Wiadomość (*Message*),
  - Zegar (*Timer*),
  - Kompensacja (*Compensation*),
  - Warunkowe (*Conditional*),
  - Wiązanie (*Link*),
  - Sygnał (*Signal*).
- Następujących zdarzeń przejściowych (*Intemediate Events*) nie wolno użyć w normalnym przepływie (*Normal Flow*):
  - Kasowanie (*Cancel*),
  - Błąd (*Error*),
  - Złożone (*Multiple).*

Obiekty przepływu
ZDARZENIA

Przejściowe – *Intermediate Event*

Zasady połączeń przepływu sekwencji (*Sequence Flow*):

- Jeżeli zdarzenie przejściowe (*Intemediate Events*) jest użyte w normalnym przepływie (*Normal Flow*) to:
  - zdarzenia typu:
    - Nieokreślonego (*None*),
    - Kompensacji (*Compensation*)

    muszą mieć jeden i tylko jeden przepływ przychodzący
  - zdarzenia typu:
    - Wiadomość (*Message*),
    - Zegar (*Timer*),
    - Warunkowe (*Conditional*),
    - Wiązanie (*Link*),
    - Sygnał (*Signal*)

    mogą mieć maksymalnie jeden przepływ przychodzący.

Obiekty przepływu ZDARZENIA

Przejściowe – *Intermediate Event*

Zasady połączeń przepływu sekwencji (*Sequence Flow*):

- Zdarzenia przejściowe (*Intermediate Events*): Wiadomość (*Message*), Zegar (*Timer*), Warunkowy (*Conditional*), Wiązanie (*Link*), Sygnał (*Signal*) są zawsze przygotowane do jednorazowego przyjęcia wyzwalacza zdarzenia jeżeli proces je zawierający jest aktywny. Nie mają one charakteru opcjonalnego i oczekują na wywołanie w czasie wykonywania procesu.
- Zdarzenia przejściowe (*Intermediate Events*) muszą być **źródłem** dokładnie jednego przepływu sekwencji (*Sequence Flow*).
  - Wyjątek: zdarzenie przejściowe wiązania o charakterze wysyłającym (*Source Link Intermediate Event*) nie musi posiadać wychodzącego przepływu sekwencji (*Sequence Flow*)
- Zdarzenie przejściowe typu wiązanie (*Link Intermediate Event*) nie może być **jednocześnie źródłem i celem** przepływu sekwencji (*Sequence Flow*).
- Jeżeli zdarzenie przejściowe jest typu źródłowe wiązanie (*Source Link Intermediate Event*), to musi istnieć zdarzenie przejściowe typu cel wiązania (*Target Link Intermediate Event*) posiadające tą samą nazwę.
- Może istnieć **wiele źródłowych** zdarzeń przejściowych typu wiązanie dla pojedynczego celu zdarzenia przejściowego typu wiązanie.
- Dla danego zdarzenia przejściowego typu źródłowe wiązanie (*Source Link Intermediate Event*) nie można określić więcej niż jednego zdarzenia przejściowego typu cel wiązania (*Target Link Intermediate Event*).

Obiekty przepływu
ZDARZENIA

Przejściowe – *Intermediate Event*

Zasady połączeń przepływu wiadomości (*Message Flow*):

- Wszystkie przepływy wiadomości (*Message Flow*) muszą łączyć oddzielne obszary wspólne (*Pools*). Mogą one łączyć zarówno granice obszaru jak i obiekty przepływu (*Flow Objects*) wewnątrz obszaru wspólnego. Nie mogą łączyć obiektów wewnątrz tego samego obszaru wspólnego (*Pool*).
- Zdarzenie przejściowe typu wiadomość może być **źródłem** jednego przepływu wiadomości (*Message Flow*).
- Zdarzenie przejściowe typu wiadomość może być **celem** jednego przepływu wiadomości (*Message Flow*).
- Zdarzenie przejściowe typu wiadomość nie może być **jednocześnie źródłem i celem** przepływu wiadomości (*Message Flow*).

Obiekty przepływu AKTYWNOŚCI

Aktywność – *Activity*

Aktywność reprezentuje pracę jaka jest wykonywana wewnątrz procesu biznesowego. Aktywność może być atomowa (elementarna) lub nieatomowa (złożona). Występują następujące rodzaje aktywności będące częścią BPD:

- Proces (*Process*),
- Podproces (*Sub-Process*),
- Zadanie (*Task*).

Proces nie posiada odpowiadającego mu symbolu graficznego, gdyż jest zbiorem obiektów graficznych.

Obiekty przepływu
AKTYWNOŚCI

Podproces – *Sub-Process*

Podprocesy wbudowane (zagnieżdżone)

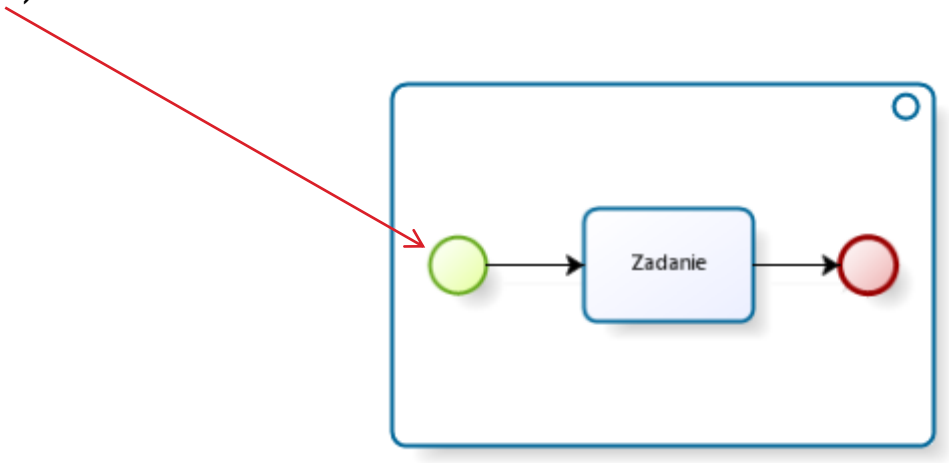
Reprezentacja graficzna: 

- Obiekt podprocesu wbudowanego (*Embedded Sub-Proces*) jest aktywnością, która zawiera inne aktywności (procesy). Proces wewnątrz innego procesu jest zależny od wywołania go przez proces nadrzędny. Podproces posiada dostęp do danych procesu nadrzędnego, więc nie istnieje potrzeba mapowania danych.
- Obiekty wewnątrz podprocesu wbudowanego (*Embedded Sub-Proces*) będąc zależnym od procesu nadrzędnego, nie posiadają wszystkich cech pełnego Diagramu Procesów Biznesowych (*Business Process Diagram*) jak obszary wspólne (*Pools*) i tory (*Lanes*). W związku z tym rozwinięta postać podprocesu wbudowanego może posiadać:
  - obiekty przepływu (*Flow Objects*),
  - obiekty powiązań (*Connecting Objects*),
  - artefakty (*Artifacts*).

Obiekty przepływu AKTYWNOŚCI

Podproces – *Sub-Process*

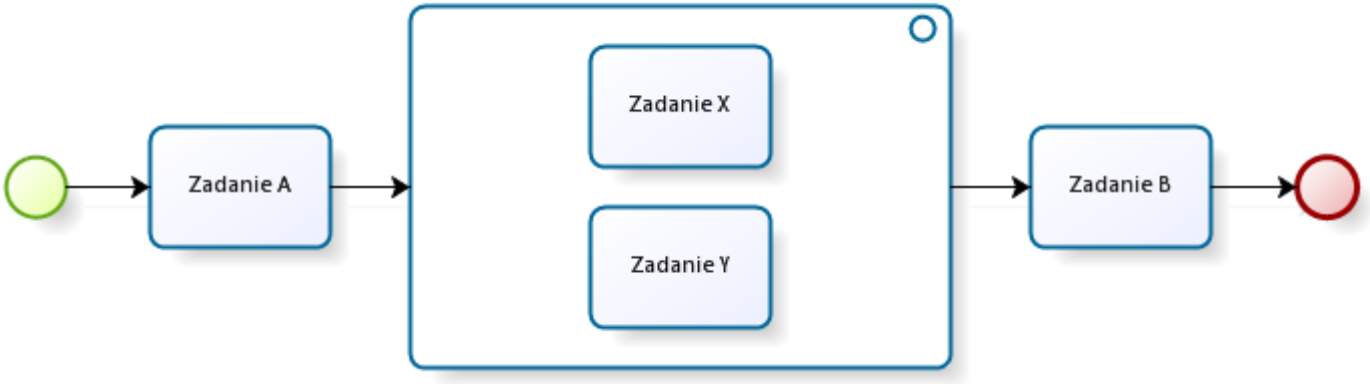
- Wszystkie zdarzenia początkowe (*Start Events*) w podprocesach wbudowanych (*Embedded Sub-Proces*) muszą być **typu nieokreślonego** (*None*).



Obiekty przepływu
AKTYWNOŚCI

Podproces – *Sub-Process*

Przykład rozwiniętego podprocesu zawierającego zadania równoległe.

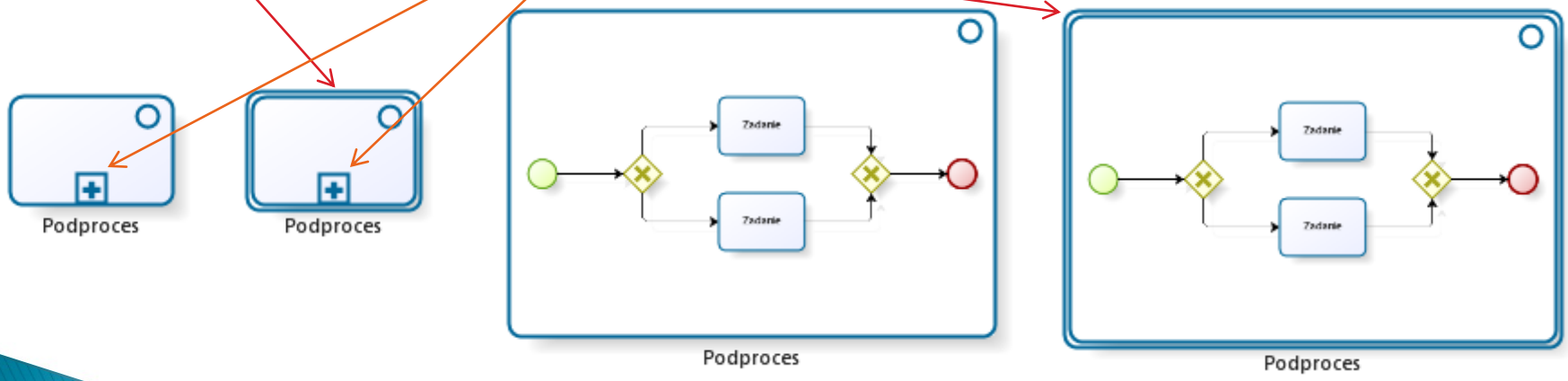


Obiekty przepływu
AKTYWNOŚCI

Podproces – *Sub-Process*

Podproces jest złożoną aktywnością posiadającą informacje szczegółowe dotyczące przepływu innych aktywności. Podproces jest obiektem graficznym w danym przepływie procesu (*Process Flow*), który może być „otwarty” w celu pokazania innego procesu (*Process*).

Podproces może mieć postać zwiniętą lub rozwiniętą, ponadto może być on transakcją.



Obiekty przepływu AKTYWNOŚCI

Podproces – *Sub-Process*

Występują następujące typy znaczników podprocesów:

- Pętla (*Loop*),

- Wieloinstancyjny (*Multiple Instance*),

- Doraźny (*Ad-Hoc*),

- Kompensacyjny (Compensation).

Obiekty przepływu AKTYWNOŚCI

Podproces – *Sub-Process*

Znacznik pętli (*Loop Marker*) może być używany w kombinacji z innymi, z wyjątkiem znacznika wieloinstancyjności (*Multiple Instance Marker*).

- Znacznik wieloinstancyjności (*Multiple Instance Marker*) może być używany w kombinacji z innymi, z wyjątkiem znacznika pętli (*Loop Marker*).

- Znacznik doraźny (*Ad-Hoc Marker*) może być używany w kombinacji ze wszystkimi innymi znacznikami.

Obiekty przepływu AKTYWNOŚCI

Podproces – *Sub-Process*

- Znacznik kompensacyjny (*Compensation Marker*) może być używany w kombinacji ze wszystkimi innymi znacznikami.

Obiekty przepływu AKTYWNOŚCI

Podproces – *Sub-Process*

Podproces ponownego użycia – (*Reusable Sub-Process*):

Reprezentacja graficzna:

- Podproces ponownego użycia jest aktywnością wewnątrz procesu, który wywołuje inny proces istniejący na BPD.
- Wywoływany proces nie jest zależny od procesu nadrzędnego (*parent Process*) w stosunku do wywołującego go podprocesu ponownego użycia (*Reusable Sub-Process*) z punktu widzenia danych o zakresie globalnym.
- Obiekt podprocesu ponownego użycia (*Reusable Sub-Process*) może przesyłać dane z/do wywoływanego procesu.

Obiekty przepływu
AKTYWNOŚCI

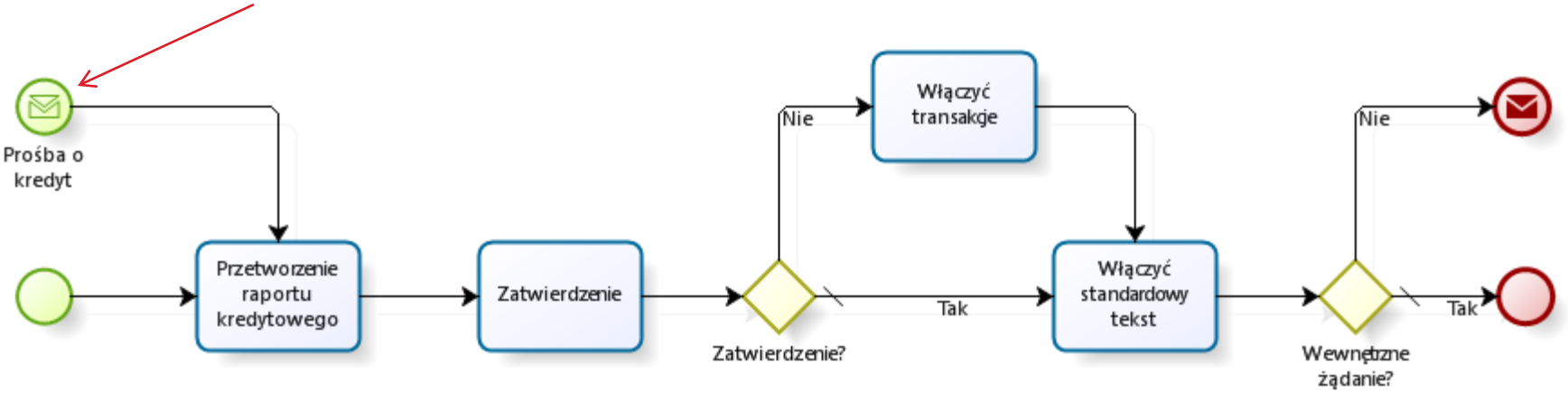
Podproces – *Sub-Process*

- Wywoływany proces istnieje w oddzielnym diagramie, który może mieć wiele obszarów wspólnych.
- Wywoływany proces może być wizualizowany za pomocą dowolnego widoku (uwzględniając widok rozszerzony), przy czym ukaże on cały diagram, w którym znajduje się wywoływany proces.
- Mapowanie danych nastąpi jedynie do wywoływanego procesu, a nie do żadnego innego procesu, który mógłby się znajdować w wywoływanym diagramie.
- Wywoływany proces musi być utworzony jako podproces (*Sub-Process*) poprzez zdarzenie początkowe nieokreślonego typu (*None Start Events*).
- Proces ponownego użycia (*Reusable Sub-Process*) może być również utworzony jako podproces przez inny niezależny obiekt podprocesu (w tym samym lub innym diagramie).

Obiekty przepływu AKTYWNOŚCI

Podproces – *Sub-Process*

- Podproces ponownego użycia (*Reusable Sub-Process*) może być również utworzony jako podproces **najwyższego poziomu** przez oddzielne zdarzenie początkowe posiadające **wyzwalacz różny od wyzwalacza nieokreślonego typu**.



Obiekty przepływu AKTYWNOŚCI

Podproces – *Sub-Process*

Podproces odwołujący się – (*Reference Sub-Process*):

Reprezentacja graficzna:

Wielokrotnie może wystąpić sytuacja, w której modelujący chciałby odwołać się do innego zdefiniowanego podprocesu (*Sub-Process*).

- Jeżeli dwa podprocesy posiadają dokładnie takie same właściwości oraz zachowania, to poprzez odwołanie jednego do drugiego, atrybuty definiujące zachowania muszą zostać utworzone tylko raz i będą utrzymywane tylko w jednym miejscu.

Obiekty przepływu
AKTYWNOŚCI

Podproces – *Sub-Process*

Podproces funkcjonujący jako transakcja.

Reprezentacja graficzna:

Podproces, zarówno w postaci zwiniętej jak i rozwiniętej, może być zdefiniowany jako transakcja (*Transaction*) charakteryzująca się specjalnym zachowaniem kontrolowanym przez protokół transakcyjny (np. BTP lub WS-Transaction).

Obiekty przepływu AKTYWNOŚCI

Podproces – *Sub-Process*

Istnieją trzy podstawowe możliwości zakończenia transakcji:

- Prawidłowe – Jest przedstawione jako normalny przepływ sekwencji (*Sequence Flow*) i powoduje opuszczenie podprocesu (*Sub-Process*)
- Nieprawidłowe (Kasowanie) – Gdy transakcja jest kasowana, to aktywności wewnątrz Transakcji zostają poddane akcjom kasowania, które mogą zawierać procedurę odwracania procesu i kompensacji wybranych aktywności. Należy zauważyć, że inne mechanizmy przerwania podprocesu nie będą skutkowały kompensacją (*Compensation*) np. Błąd (*Error*), Zegar (*Timer*) i jakiekolwiek inne specyficzne dla aktywności nietransakcyjnej. Zdarzenie przejściowe kasowania (*Cancel Intermediate Event*) przymocowane do granic aktywności kieruje przepływ bezpośrednio po odwróceniu transakcji oraz zakończeniu wszystkich kompensacji. Zdarzenie przejściowe kasowania (*Cancel Intermediate Event*) może być użyte tylko, gdy jest przymocowane do granic aktywności transakcyjnej. Nie może zostać użyte w jakimkolwiek normalnym przepływie (*Normal Flow*) lub przymocowane do aktywności nietransakcyjnej (*non-Transaction*).

Obiekty przepływu AKTYWNOŚCI

Podproces – *Sub-Process*

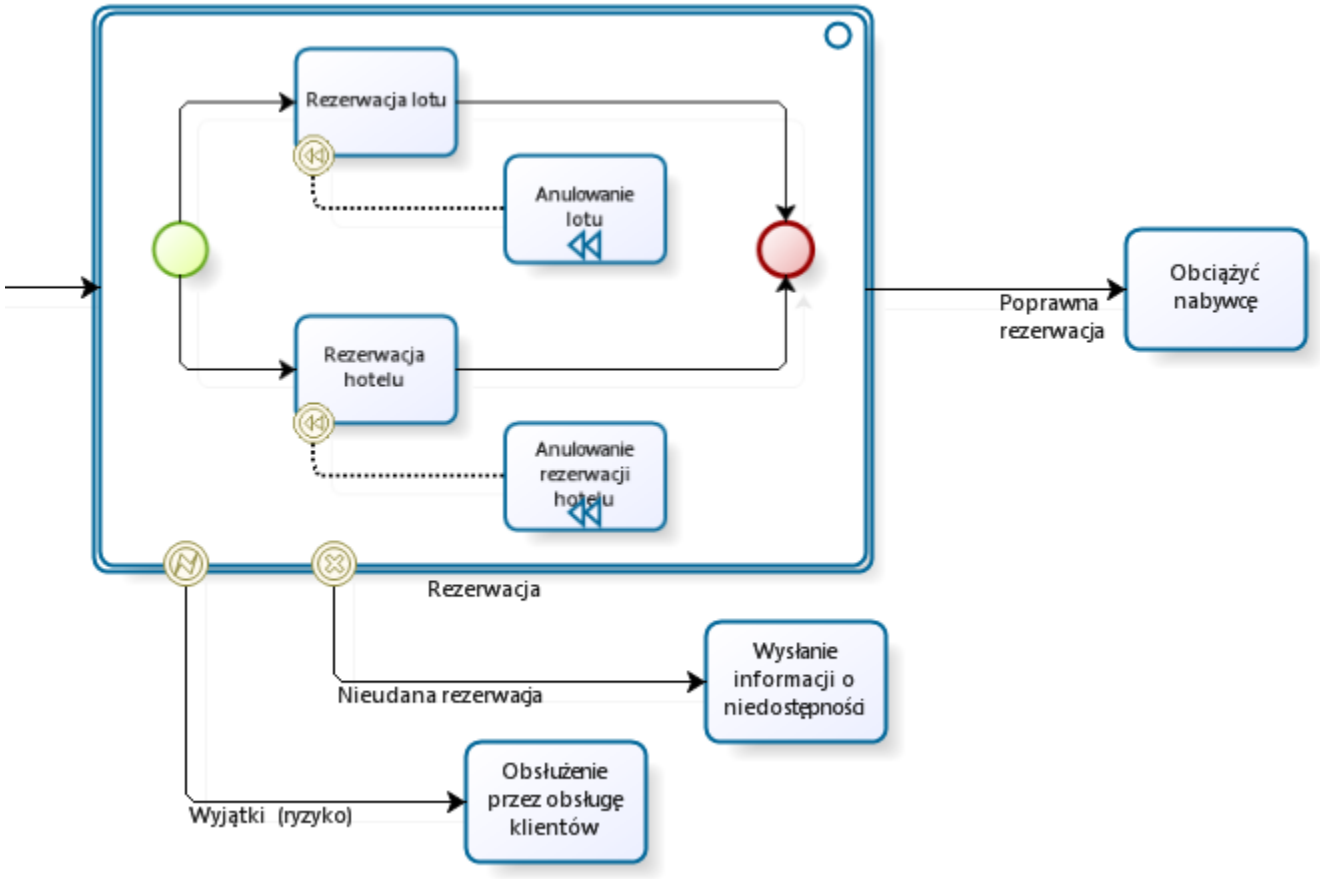
Istnieją dwa mechanizmy które mogą być sygnałem do skasowania transakcji (*Transaction*):

- Osiągnięto **zdarzenie końcowe kasowania** (*Cancel End Event*) wewnątrz podprocesu transakcyjnego (*Transaction Sub-Process*). Zdarzenie końcowe kasowania (*Cancel End Event*) może być użyte tylko wewnątrz podprocesu transakcyjnego (*Transaction Sub-Process*) .
- Otrzymano poprzez protokół transakcyjny (*Transaction Protocol*) wspierający wykonanie podprocesu (*Sub-Process*) **wiadomość kasowania** (*Cancel Message*).

- Ryzykowne – Oznacza, że coś potoczyło się całkowicie źle i prawidłowe zakończenie lub skasowanie nie jest możliwe. Używa się błędu (*Error*) do pokazania niebezpieczeństwa. Kiedy niebezpieczeństwo się pojawia, aktywność jest przerywana (bez kompensacji), a przepływ jest kontynuowany od zdarzenia przejściowego błędu (*Error Intermediate Event*).

Obiekty przepływu
AKTYWNOŚCI

Podproces – *Sub-Process*



Obiekty przepływu AKTYWNOŚCI

Podproces – *Sub-Process*

Zachowanie na końcu prawidłowo wykonanej podprocesu transakcyjnego (*Transaction Sub-Process*) jest nieco odmienne od zwykłego podprocesu (*Sub-Process*).

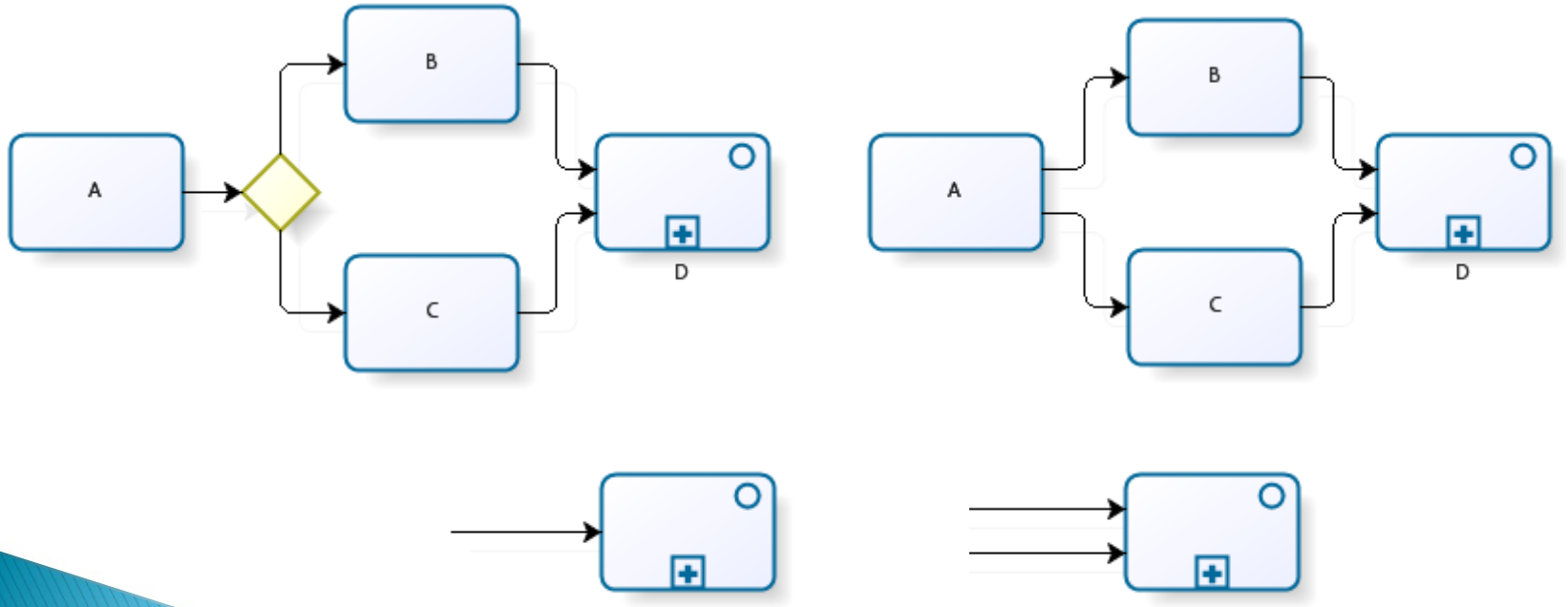
- Gdy każda ścieżka podprocesu transakcyjnego (*Transaction Sub-Process*) osiągnie zdarzenie różne od zdarzenia końcowego kasowania (*non-Cancel End Event*), to przepływ nie przeniesie się natychmiast do procesu nadrzędnego, jak się dzieje w przypadku zwykłego podprocesu (*Sub-Process*).
- Na wstępie, protokół transakcyjny musi zweryfikować czy wszyscy uczestnicy poprawnie zakończyli transakcję. Zazwyczaj tak się dzieje, jednak możliwe jest, że jakiś z uczestników może zakończyć transakcję z problemem będącym przyczyną kasowania (*Cancel*) lub ryzyka (*Hazard*). W takim przypadku, przepływ zostanie przeniesiony do odpowiedniego zdarzenia przejściowego (*Intermediate Event*), nawet jeżeli pozornie zakończył się prawidłowo.

Obiekty przepływu
AKTYWNOŚCI

Podproces – *Sub-Process*

Zasady połączeń przepływu sekwencji (*Sequence Flow*):

- Podproces (*Sub-Process*) może być celem przepływu sekwencji (Sequence *Flow*). Może posiadać wiele sekwencji przychodzących. Przychodzące przepływy mogą pochodzić od alternatywnych i/lub równoległych ścieżek.



Obiekty przepływu AKTYWNOŚCI

Podproces – *Sub-Process*

Zasady połączeń przepływu sekwencji (*Sequence Flow*):

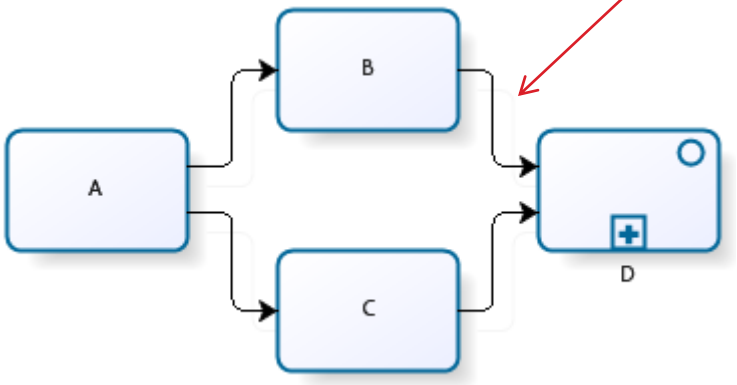
- Atrybut *TargetRef* należący do przychodzącego przepływu sekwencji (*Incoming Sequence Flow*) może zostać rozszerzony zarówno o obiekt podprocesu na poziomie nadrzędnym, jak i o zdarzenie początkowe (*Start Event*) znajdujące się w podprocesie (*Sub-Process*). Zapewnia to bezpośrednie połączenie nadrzędnego przepływu sekwencji (*Sequence Flow*) z podrzędnym zdarzeniem startowym (*Start Event*), w sytuacji gdy istnieje więcej niż jedno zdarzenie początkowe (*Start Event*) w podprocesie (*Sub-Process*).
- Jeżeli szczegóły podprocesu (*Sub-Process*) (np. zdarzenie początkowe) są niewidoczne lub niedostępne dla modelującego, to nie jest określone które zdarzenie początkowe (*Start Event*), jeżeli jest ich wiele, zostanie wywołane. Pewne jest, że tylko jedno zdarzenie początkowe (*Start Event*) zostanie wywołane.

Obiekty przepływu AKTYWNOŚCI

Podproces – *Sub-Process*

Zasady połączeń przepływu sekwencji (*Sequence Flow*):

- Jeżeli podproces (*Sub-Process*) posiada wiele przychodzących przepływów sekwencji (Sequence *Flow*), to oznacza to **przepływ niekontrolowany**. Oznacza to, że gdy token (*Token*) przybywa **z jednej ze ścieżek**, to podproces (*Sub-Process*) zostaje utworzony.
- **Nie czeka się** na przybycie tokenów (*Tokens*) z innych ścieżek.
- Dla każdego przybywającego tokenu (*Token*), z tej samej lub innej ścieżki, zostaje utworzony **oddzielny** podproces (*Sub-Process*).

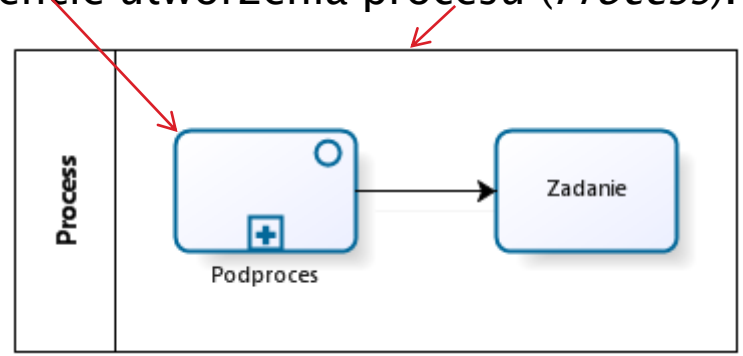


Obiekty przepływu
AKTYWNOŚCI

Podproces – *Sub-Process*

Zasady połączeń przepływu sekwencji (*Sequence Flow*):

- Jeżeli podproces (*Sub-Process*) nie posiada przychodzącego przepływu sekwencji (*Sequence Flow*) i nie ma w procesie (*Process*) zdarzenia początkowego (*Start Event*), to podproces (*Sub-Process*) musi zostać utworzony w momencie utworzenia procesu (*Process*).



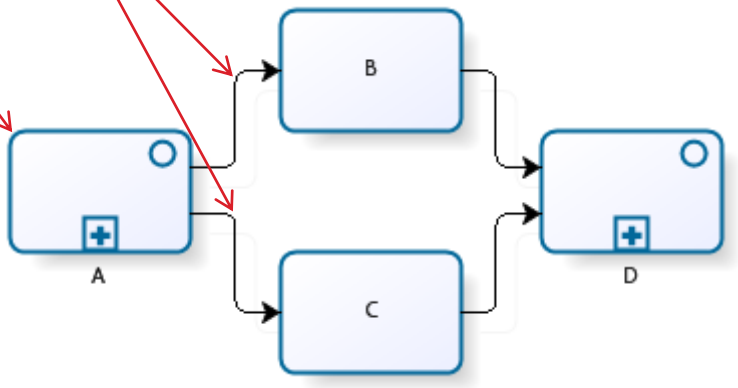
  - Wyjątek stanowią podprocesy (*Sub-Process*), które zostały zdefiniowane jako aktywności kompensujące. Podprocesy kompensujące (*Compensation Sub-Processes*) nie są uznawane za część normalnego przepływu (*Normal Flow*) i nie mogą być utworzone podczas tworzenia procesu (*Process*)

Obiekty przepływu
AKTYWNOŚCI

Podproces – *Sub-Process*

Zasady połączeń przepływu sekwencji (*Sequence Flow*):

- Podproces (*Sub-Process*) może być źródłem przepływu sekwencji (*Sequence Flow*). Może posiadać wiele sekwencji wychodzących, co oznacza, że oddzielne, **równoległe** ścieżki zostaną utworzone dla każdego przepływu (*Flow*).



Obiekty przepływu
AKTYWNOŚCI

Podproces – *Sub-Process*

Zasady połączeń przepływu sekwencji (*Sequence Flow*):

- Jeżeli podproces (*Sub-Process*) nie posiada wychodzącego przepływu sekwencji (*Sequence Flow*) i nie istnieje zdarzenie końcowe dla procesu (*Process*), to podproces (*Sub-Process*) oznacza koniec jednej lub większej liczby ścieżek w procesie (*Process*). Gdy podproces (*Sub-Process*) się kończy i nie ma innych aktywnych równoległych ścieżek, to proces (*Process*) musi się zakończyć.
  - Wyjątek stanowią podprocesy (*Sub-Process*), które zostały zdefiniowane jako aktywności kompensujące. Podprocesy kompensujące (*Compensation Sub-Processes*) nie są uznawane za część normalnego przepływu (*Normal Flow*) i nie mogą oznaczać końca procesu (*Process*).

Obiekty przepływu AKTYWNOŚCI

Podproces – *Sub-Process*

Zasady połączeń sekwencji wiadomości (*Message Flow*):

- Wszystkie przepływy wiadomości (*Message Flow*) muszą łączyć oddzielne obszary wspólne (*Pools*). Mogą one być połączone z granicą obszaru wspólnego (*Pool*) lub z obiektem przepływu (*Flow Object*) w granicach tego obszaru, jednak nie mogą łączyć dwóch obiektów wewnątrz tego samego obszaru wspólnego (*Pool*).
- Podproces (*Sub-Process*) może być **źródłem** zerowej lub większej liczby wychodzących przepływów wiadomości (*Message Flow*).
- Podproces (*Sub-Process*) może być **celem** zerowej lub większej liczby przychodzących przepływów wiadomości (*Message Flow*).

Obiekty przepływu AKTYWNOŚCI

Zadanie – *Task*

Reprezentacja graficzna:

- Zadnie jest **atomową** (niepodzielną) aktywnością zawartą w procesie (*Process*). Zadnie jest używane, gdy praca w procesie (*Process*) nie jest rozbijana na drobniejsze poziomy. Użytkownik końcowy i/lub aplikacja zwykle wykonuje zadanie kiedy zostaje ono uruchomione.
- Istnieją trzy kategorie znaczników zadania:
  - pętli (*Loop*),

  - wieloinstancyjny (*Mulitple Instance*),

  - kompensacyjny (*Compensation*).

Obiekty przepływu AKTYWNOŚCI

Zadanie – *Task*

- Jako uzupełnienie istniejących kategorii, istnieją różne typy zadań (*Task*) w celu rozróżnienia wrodzonych zachowań, które zadania (*Tasks*) mogą reprezentować. BPMN nie specyfikuje graficznych znaczników dla tych typów. Modelujący lub narzędzia do modelowania mogą tworzyć własne znaczniki w celu wskazania czytelnikowi typu zadania (*Task*):
  - *Service* 

  - *Receive* 

  - *Send* 

  - *User* 

  - *Script* 

  - *Reference* 

  - *Manual* 

Obiekty przepływu AKTYWNOŚCI

Zadanie – *Task*

Zadanie odbierające (*Receive Task*)

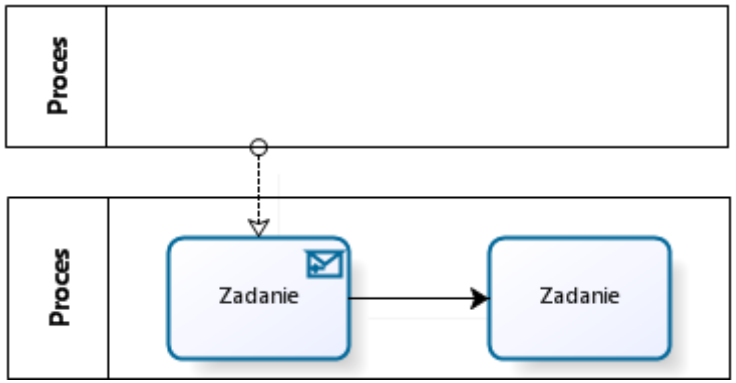
- Zadanie odbierające (*Receive Task*) jest prostym zadaniem przeznaczonym do oczekiwania na wiadomość od uczestnika zewnętrznego w stosunku do procesu biznesowego (*Business Process*).
- W momencie przyjścia wiadomości zadanie jest zostaje zakończone.

Obiekty przepływu
AKTYWNOŚCI

Zadanie – *Task*

Zadanie odbierające (*Receive Task*)

- Zadanie odbierające (*Receive Task*) jest często używane w celu rozpoczęcia procesu (*Process*) w sensie, że proces (*Process*) jest inicjowany przez przychodzącą wiadomość.
- Aby zadanie (*Task*) inicjowało proces (*Process*) musi być spełniony **jeden** z poniższych warunków:
  - Proces (*Process*) nie posiada zdarzenia początkowego (*Start Event*) oraz zadanie odbierające (*Receive Task*) nie posiada przychodzącego przepływu sekwencji (*Sequence Flow*).

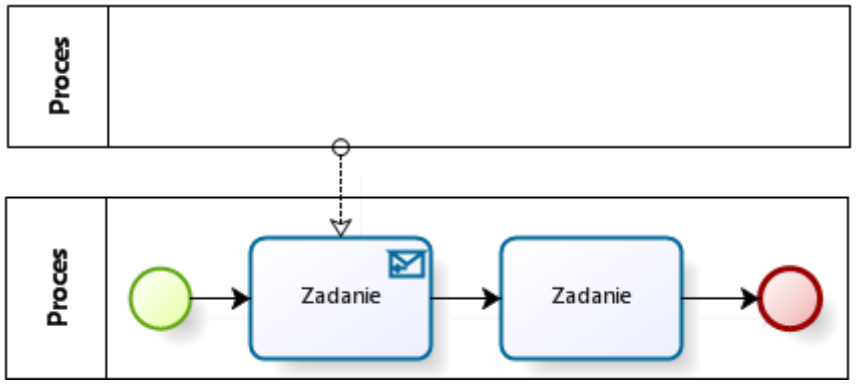


Obiekty przepływu
AKTYWNOŚCI

Zadanie – *Task*

Zadanie odbierające (*Receive Task*)

- Przychodzący przepływ sekwencji (*Sequence Flow*) zadana odbierającego (*Receive Task*) posiada zwoje źródło w zdarzeniu początkowym (*Start Event*). Dla zadana odbierającego (*Receive Task*) nie jest dozwolona żadna inna przychodzący przepływ sekwencji (*Sequence Flow*).



Obiekty przepływu AKTYWNOŚCI

Zadanie – *Task*

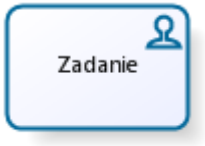
Zadanie wysyłające (*Send Task*)

- Zadanie wysyłające (*Send Task*) jest prostym zadaniem przeznaczonym do wysyłania wiadomości do uczestnika zewnętrznego w stosunku do procesu biznesowego (*Business Process*).
- W momencie wysłania wiadomości zadanie jest zostaje zakończone.

Obiekty przepływu AKTYWNOŚCI

Zadanie – *Task*

Zadanie użytkownika (*User Task*)



- Zadanie użytkownika jest typowym zadaniem przepływu pracy, gdzie człowiek wykonuje zadanie przy wsparciu oprogramowania i które jest wynikiem zaplanowania przez, pewnego rodzaju, zarządzającego listą zadań.

Obiekty przepływu
AKTYWNOŚCI

Zadanie – *Task*

Zadanie skryptu (*Script Task*)

- Zadanie skryptu jest wykonywane przez silnik procesów biznesowych. Modelujący lub implementujący definiuje skrypt w języku, który silnik potrafi implementować.
- Gdy zadanie (*Task*) jest gotowe do rozpoczęcia, silnik wykonuje skrypt.
- W momencie zakończenia działania skryptu, zadanie również zostaje zakończone.

Obiekty przepływu AKTYWNOŚCI

Zadanie – *Task*

Zadanie ręczne (*Manual Task*)

- Zadanie ręczne jest zadaniem, którego wykonanie jest realizowane bez żadnego wsparcia silnika wykonawczego procesów biznesowych lub jakiejkolwiek aplikacji. Przykładem może być technik telekomunikacyjny montujący telefon u klienta.

Obiekty przepływu AKTYWNOŚCI

Zadanie – *Task*

Zadanie odwołujące się (*Reference Task*)

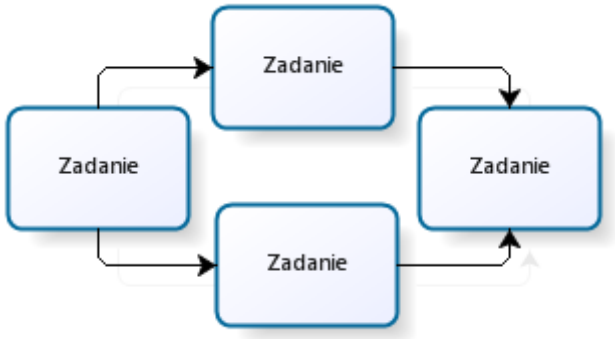
- Wielokrotnie modelujący może chcieć odwołać się do innego zdefiniowanego zadania (*Task*). Jeżeli dwa lub więcej zadań (*Tasks*) współdzieli to samo zachowanie, to poprzez odwołanie do innego, atrybuty definiujące zachowanie muszą być utworzone i wspierane tylko w jednym miejscu.

Obiekty przepływu AKTYWNOŚCI

Zadanie – *Task*

Zasady połączeń przepływu sekwencji (*Sequence Flow*):

- Zadanie może być celem wielu przepływów sekwencji. Przepływy przychodzące mogą pochodzić zarówno od ścieżek alternatywnych jak i równoległych.
- Jeżeli zadanie (*Task*) posiada wiele przychodzących przepływów sekwencji (*Sequence Flow*), to oznacza to **przepływ niekontrolowany**. Oznacza to, że gdy token (*Token*) przybywa **z jednej ze ścieżek**, to zadanie (*Task*) zostaje utworzone.
- **Nie czeka się** na przybycie tokenów (*Tokens*) z innych ścieżek.
- Dla każdego przybywającego tokenu (*Token*), z tej samej lub innej ścieżki, zostaje utworzony **oddzielne** zadanie (*Task*).

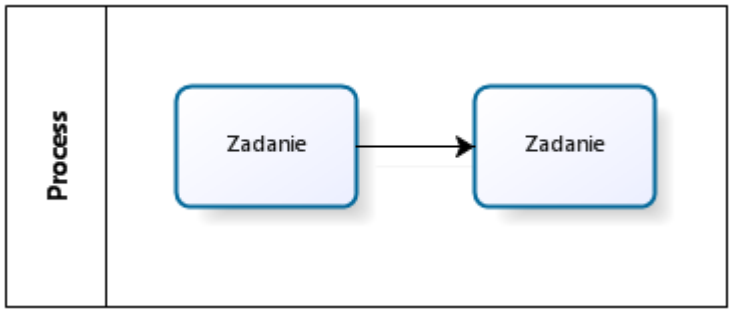


Obiekty przepływu AKTYWNOŚCI

Zadanie – *Task*

Zasady połączeń przepływu sekwencji (*Sequence Flow*):

- Jeżeli zadanie (*Task*) nie posiada przychodzącego przepływu sekwencji (*Sequence Flow*) i nie ma w procesie (*Process*) zdarzenia początkowego (*Start Event*), to zadanie (*Task*) musi zostać utworzone w momencie utworzenia procesu (*Process*).



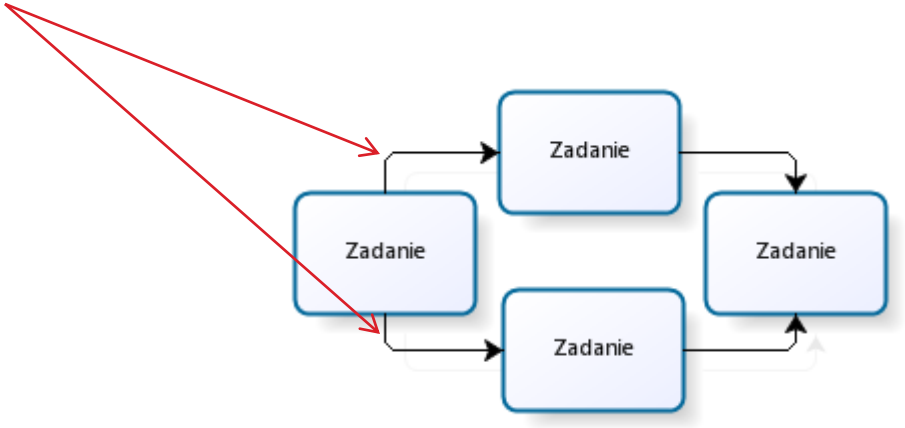
  - Wyjątek stanowią zadania (*Tasks*), które zostały zdefiniowane jako aktywności kompensujące. Zadania kompensujące (*Compensation Tasks*) nie są uznawane za część normalnego przepływu (*Normal Flow*) i nie mogą być utworzone podczas tworzenia procesu (*Process*)

Obiekty przepływu
AKTYWNOŚCI

Zadanie – *Task*

Zasady połączeń przepływu sekwencji (*Sequence Flow*):

- Zadanie (*Task*) może być źródłem przepływu sekwencji (*Sequence Flow*). Może posiadać wiele sekwencji wychodzących, co oznacza, że oddzielne, **równoległe** ścieżki zostaną utworzone dla każdego przepływu (*Flow*).



Obiekty przepływu AKTYWNOŚCI

Zadanie – *Task*

Zasady połączeń przepływu sekwencji (*Sequence Flow*):

- Jeżeli zadanie (*Task*) nie posiada wychodzącego przepływu sekwencji (*Sequence Flow*) i nie istnieje zdarzenie końcowe dla procesu (*Process*), to zadanie (*Task*) oznacza koniec jednej lub większej liczby ścieżek w procesie (*Process*). Gdy Zadanie (*Task*) się kończy i nie ma innych aktywnych równoległych ścieżek, to proces (*Process*) musi się zakończyć.
  - Wyjątek stanowią zadania (*Tasks*), które zostały zdefiniowane jako aktywności kompensujące. Zadania kompensujące (*Compensation Tasks*) nie są uznawane za część normalnego przepływu (*Normal Flow*) i nie mogą oznaczać końca procesu (*Process*).

Obiekty przepływu AKTYWNOŚCI

Zadanie – *Task*

Zasady połączeń sekwencji wiadomości (*Message Flow*):

- Wszystkie przepływy wiadomości (*Message Flow*) muszą łączyć oddzielne obszary wspólne (*Pools*). Mogą one być połączone z granicą obszaru wspólnego (*Pool*) lub z obiektem przepływu (*Flow Object*) w granicach tego obszaru, jednak nie mogą łączyć dwóch obiektów wewnątrz tego samego obszaru wspólnego (*Pool*).
- Zadanie (*Task*) może być **źródłem** zerowej lub większej liczby wychodzących przepływów wiadomości (*Message Flow*). Jeżeli istnieje wiele wychodzących przepływów wiadomości (*Message Flow*), to pojedyncza wiadomość zostanie przypisana do wszystkich przepływów wiadomości (*Message Flow*). Wiadomość ta zostanie wysłana do wszystkich wychodzących przepływów wiadomości (*Message Flow*)
- Zadanie (*Task*) może być **celem** zerowej lub większej liczby przychodzących przepływów wiadomości (*Message Flow*). Tylko jedna wiadomość może zostać otrzymana z pojedynczego przepływu wiadomości (*Message Flow*) dla danego egzemplarza zadania.

Obiekty przepływu
BRAMY

Bramy – *Gateways*

Bramy (*Gateways*) są elementami modelującymi, które mają zastosowanie w kontrolowaniu zachowania przepływu sekwencji (*Sequence Flow*) w momencie jego schodzenia i rozchodzenia w trakcie procesu (*Process*).

- Brama jest zbędna jeżeli przepływ nie musi być kontrolowany.
- Termin "Brama" wskazuje, że istnieje mechanizm bramkowania, który pozwala albo uniemożliwia przejście przez bramę (*Gateway*) – czyli, gdy Tokeny docierają do bramy (*Gateway*), mogą być scalone razem na wejściu lub/i rozdzielone na wyjściu, gdy mechanizmy bramy (*Gateway*) są wzbudzone. W celu zobrazowania, brama (*Gateway*) jest tak naprawdę zbiorem „bramek" (*Gates*). Pomimo, że bramki (*Gates*) nie posiadają reprezentacji graficznej, to są one używanie przez przepływ sekwencji (*Sequence Flow*) w celu połączenia z bramą *(Gateway).*
- Są różne rodzaje bram (*Gateway*). Zachowanie każdego z rodzajów bram (*Gateway*) określa jak wiele bramek (*Gates*) jest dostępnych dla kontynuacji przepływu. Każdemu wyjściowemu przepływowi sekwencji (*Sequence Flow*) odpowiada jedna bramka (*Gate*).

Obiekty przepływu
BRAMY

Bramy – *Gateways*

- Bramy mogą określać wszystkie typy zachowań procesów biznesowych w przepływie sekwencji, tj.: decyzje/rozgałęzienia (rozłączne, łączne i złożone), złączanie, rozwidlanie i łączenie. W ten sposób romb, który był używany tradycyjnie dla rozłącznych decyzji, został rozszerzony w BPMN, tak aby odzwierciedlać dowolny typ kontroli nad przepływem sekwencji. Każdy z typów bramy posiada odpowiedni identyfikator graficzny, zgodnie z poniższym zestawieniem:
- Rozłączny
- – oparty o dane 
- – oparty o zdarzenie 
- Łączny 
- Złożony 
- Równoległy 

Obiekty przepływu
BRAMY

Bramy – *Gateways*

- Bramy mogą kontrolować przepływ zarówno rozchodzącego się, jak i złączającego się przepływu sekwencji. Oznacza to, że konkretna brama (*Gateway*) może mieć wiele bramek (*Gates*) wejściowych, jak i wyjściowych w tym samym czasie (ale może być tylko jeden przepływ sekwencji dla bramki). Typ bramy definiuje ten sam typ zachowań dla łączących i rozchodzących się przepływów sekwencji (*Sequence Flow*).

Obiekty przepływu
BRAMY

Bramy – *Gateways*

Łączenie bram (*Gateways*) z przepływem sekwencji (*Sequence Flow*):

- Brama (*Gateway*) może być celem dla zera lub dowolnej liczby przychodzących przepływów sekwencji (*Sequence Flow*). Przychodzący przepływ może pochodzić z alternatywnej lub równoległej ścieżki.
  - Jeżeli Brama nie posiada przychodzących przepływów sekwencji i nie ma zdarzenia startowego dla procesu, to rozdzielenie przepływu (zależnie od typu bramy) nastąpi w momencie inicjacji procesu.
- Brama (*Gateway*) może być źródłem zera lub dowolnej liczby przepływów sekwencji.
- Brama (*Gateway*) może mieć wiele przychodzących i wychodzących przepływów sekwencji

Obiekty przepływu
BRAMY

Bramy – *Gateways*

Łączenie bram (*Gateways*) z przepływem wiadomości (*Message Flow*):

- Brama (*Gateway*) nie może być celem ani źródłem przepływu wiadomości (*Message Flow*).

Obiekty przepływu
BRAMY

Bramy rozłączne – *Exclusive Gateways*

- Rozłączne bramy (decyzyjne) są miejscem w procesie biznesowym, w którym przepływ sekwencji może obrać dwie lub więcej alternatywnych ścieżek. Jest to po prostu rozwidlenie (*Fork*) dróg dla procesu. Dla danego działania (wywołania) procesu tylko jedna ze ścieżek może zostać wybrana. Decyzja nie jest aktywnością z punktu widzenia procesu biznesowego, ale jest typem bramy, który kontroluje przepływ sekwencji pomiędzy aktywnościami. Każda z bramek (*Gates*) decyzyjnych jest powiązana z wyrażeniem warunkowym umieszczonym w atrybucie wychodzącego przepływu sekwencji (*Sequence Flow*). Kiedy następuje wybór bramki (*Gate*) w trakcie realizacji procesu, następuje równoczesny wybór danego przepływu sekwencji (*Sequence Flow*). Token przybywający do bramki decyzyjnej zostanie skierowany do odpowiedniej ścieżki, zgodnej z decyzją na bramie.

Obiekty przepływu
BRAMY

Bramy rozłączne oparte na danych – *Data Based Exclusive Gateways*

- Rozłączna brama oparta na danych jest najczęściej używanym typem bramy (*Gateway*). Zestaw bramek (*Gates*) jest oparty na wyrażeniach logicznych zawartych w atrybucie Wyrażenie Warunkowe (*ConditionExpression*) wychodzącego z bramy (*Gateway*) przepływu sekwencji (*Sequence Flow*). Wyrażenia te używają wartości danych procesu do określenia właściwej ścieżki, stąd też ich nazwa.

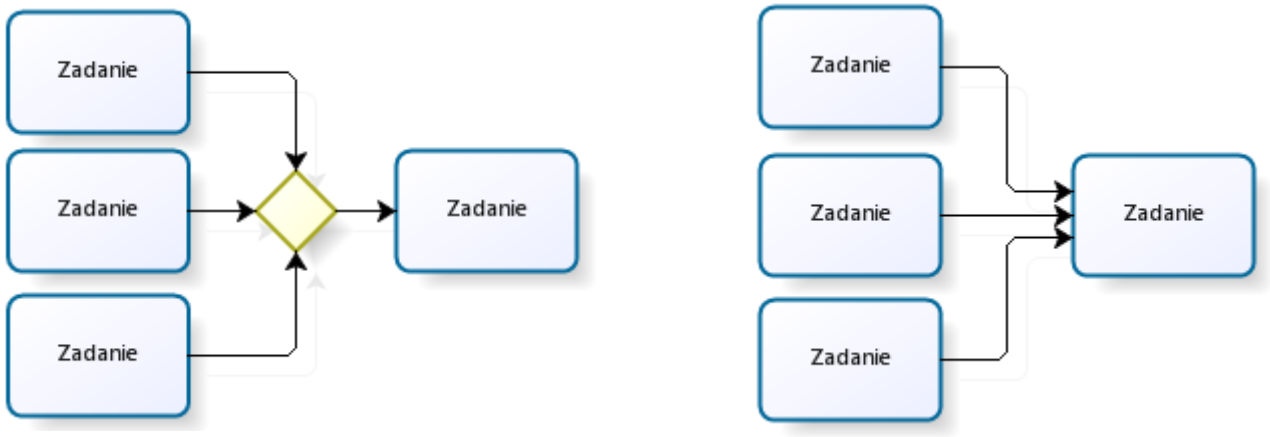
Obiekty przepływu
BRAMY

Bramy rozłączne oparte na danych – *Data Based Exclusive Gateways*

Obiekty przepływu
BRAMY

Bramy rozłączne oparte na danych – *Data Based Exclusive Gateways*

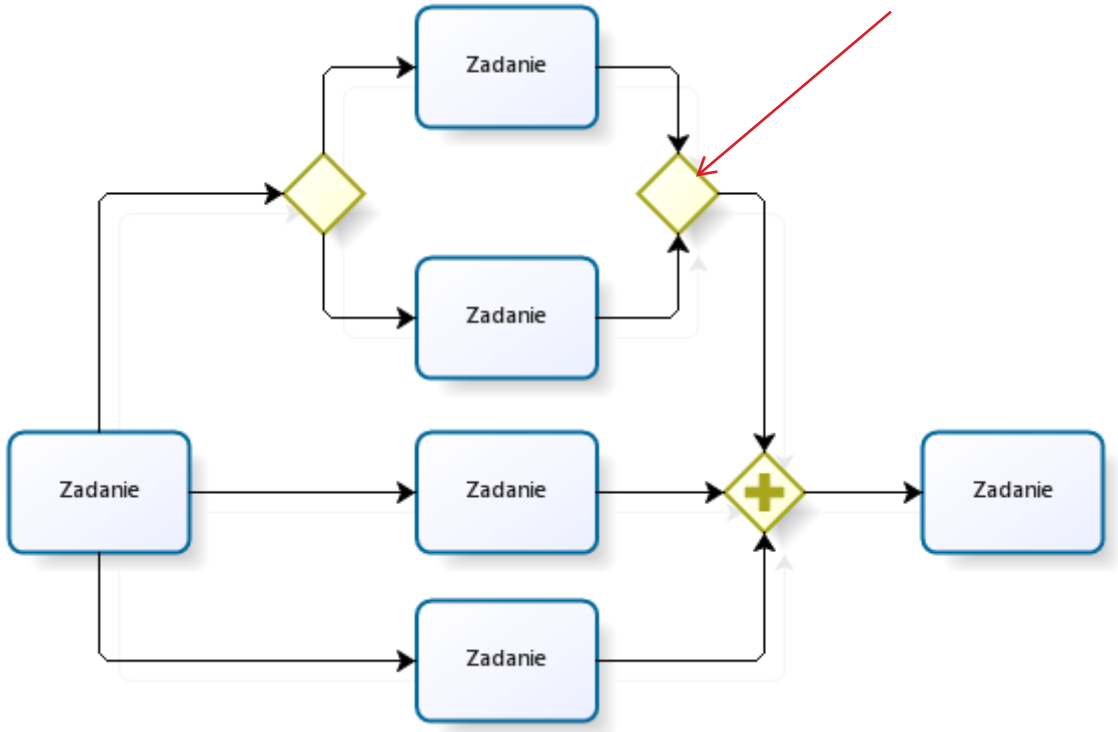
Różnica pomiędzy kontrolowanym połączeniem przepływów sekwencji, a połączeniem niekontrolowanym



Obiekty przepływu
BRAMY

Bramy rozłączne oparte na danych – *Data Based Exclusive Gateways*

Przykład bramy rozłącznej użytej do złączenia dwóch przepływów sekwencji przed bramą równoległą:



Obiekty przepływu
BRAMY

Bramy rozłączne oparte na danych – *Data Based Exclusive Gateways*

Zasady połączeń przepływu sekwencji (*Sequence Flow*):

- Jeżeli jest wiele przychodzących przepływów sekwencji (*Sequence Flow*), to wszystkie z nich zostaną użyte w celu kontynuowania przepływu procesu.
- Proces będzie kontynuowany kiedy token (*Token*) przybędzie z któregokolwiek z przepływów sekwencji.
- Tokeny z innych przepływów sekwencji z danego zestawu mogą przybywać w innym czasie, a przepływ będzie kontynuowany również w trakcie ich przybycia, bez potrzeby synchronizacji sygnałów przybywających z innych przepływów sekwencji.

Obiekty przepływu BRAMY

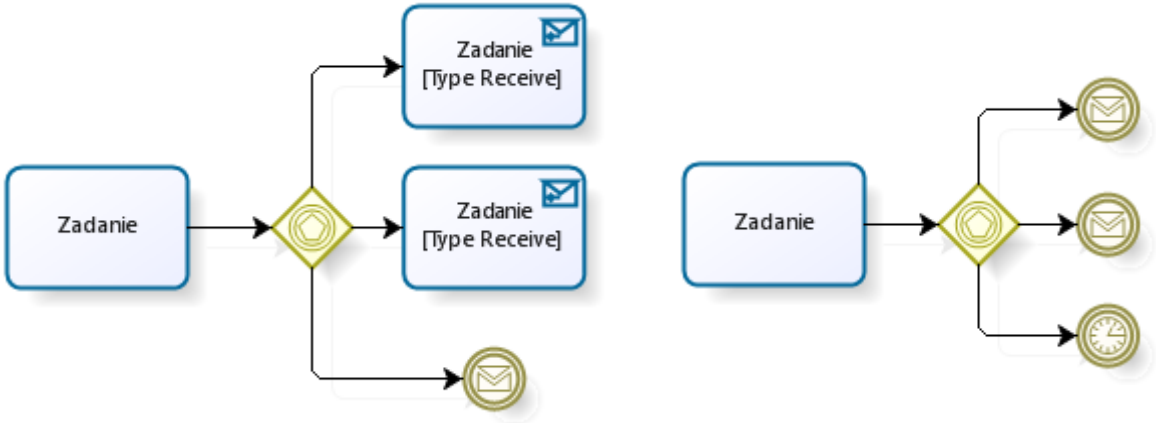
Bramy rozłączne oparte na danych – *Data Based Exclusive Gateways* Zasady połączeń przepływu sekwencji (*Sequence Flow*):

- Jeżeli jest wiele wychodzących przepływów sekwencji, to tylko jedna bramka (*Gate*) (lub bramka domyślna) może zostać wybrana w trakcie realizacji procesu.
- Bramka (*Gate*) musi zostać wybrana na podstawie wyniku wyrażenie warunkującego zdefiniowanego w przepływie sekwencji powiązanym z bramką.
  - Warunki powiązane z bramkami (*Gates*) powinny być rozwijane w kolejności w jakiej występują na liście bramek dla danej bramy.
  - Jeżeli warunek zostaje określony jako „PRAWDA”, to ta brama (*Gates*) zostanie wybrana, a warunki pozostałych bramek nie zostaną rozwijane.
  - Jeżeli żadna z bramek (*Gates*) nie zostanie określona jako „PRAWDA” to zostanie wybrana bramka (*Gates*) domyślna.
- W przypadku nie możności określenia prawdziwości żadnej z bramek i braku bramki (*Gates*) domyślnej występuje błąd modelu.

Obiekty przepływu
BRAMY

Bramy rozłączne oparte na zdarzeniach – *Data Based Exclusive Gateways*

- Rozłączne bramy oparte na zdarzeniach od strony wejścia zachowują się tak samo jak bramy rozłączne oparte działające w oparciu o dane. Na wyjściu natomiast założeniem jest, że decyzje te odzwierciedlają rozgałęzienia przepływu powstające na skutek zdarzeń mających miejsce w trakcie realizacji procesu. Określone zdarzenie, zazwyczaj jest to zdarzenie wiadomości określa, która ze ścieżek powinna zostać wybrana.



Obiekty przepływu
BRAMY

Bramy rozłączne bazujące na zdarzeniach – *Event Based Exclusive Gateways*

Zasady połączeń przepływu sekwencji (*Sequence Flow*):

- Jeżeli jest wiele przychodzących przepływów sekwencji, wszystkie z nich zostaną użyte do kontynuowania procesu, tzn. tak jakby nie było bramy (*Gateway*).
- Proces będzie kontynuowany kiedy token (*Token*) przybędzie z któregokolwiek z przepływów sekwencji.
- Tokeny z innych przepływów sekwencji z danego zestawu mogą przybywać w innym czasie, a przepływ będzie kontynuowany również w trakcie ich przybycia, bez potrzeby synchronizacji sygnałów przybywających z innych przepływów sekwencji.

Obiekty przepływu BRAMY

Bramy rozłączne bazujące na zdarzeniach – *Event Based Exclusive Gateways*

Zasady połączeń przepływu sekwencji (*Sequence Flow*):

- Tylko jedna bramka wychodząca może zostać wybrana w trakcie wykonywania procesu.
- Bramka zostaje wybrana na podstawie celu przepływu sekwencji bramy
- Jeżeli cel jest zainicjowany, to wtedy bramka (*Gates*) musi zostać wybrana, a pozostałe bramki (*Gates*) nie są rozważane.
- Atrybut warunku (*Condition*) wychodzącego przepływu sekwencji musi pozostać nieokreślony (*None*).
- Celem wychodzącego przepływu sekwencji bramy (*Gateway*) musi być:
  - Zadanie z określonym typem jako zadanie odbierające.
  - Zdarzenie przejściowe (*Intermediate Event*) z atrybutem wyzwalacza ustawionym jako wiadomość (*Message*), zegar (*Timer*) lub sygnał (*Signal*).
- Jeżeli celem jednej z bramek (*Gates*) jest zadanie, to zdarzenie przejściowe wiadomości nie może być użyte jako cel innej bramki. Oznacza to, że wiadomości dla danej bramy muszą być odbierane tylko przez zdarzenia odbierające lub tylko przez zdarzenia wiadomości, ale nie przez oba typy jednocześnie.

Obiekty przepływu
BRAMY

Bramy łączne – *Inclusive Gateways*

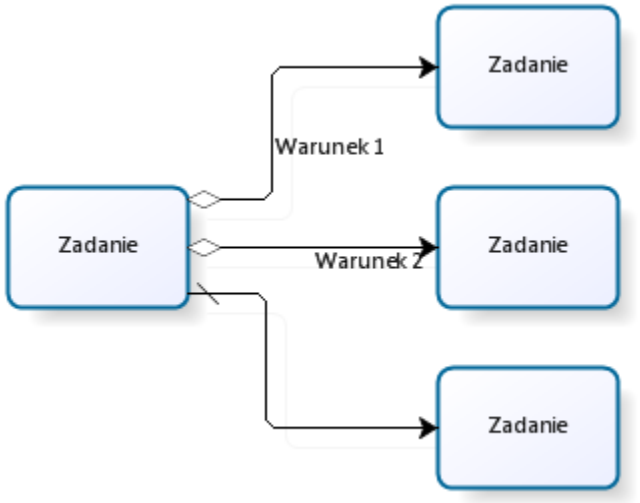
Brama łączna reprezentuje typ decyzji, która stanowi punkt rozgałęzienia przepływu sekwencji, w którym wybór bazuje na wyrażeniach warunkowych zawartych w wychodzącym przepływie sekwencji. Jednakże w tym typie bramy określenie jednego z warunków jako prawdziwego nie wyklucza rozważania pozostałych warunków. Do wszystkich rozgałęzień w których warunek zostanie określony jako prawdziwy zostanie skierowany token. W pewnym sensie można taką bramę uznać jako grupę niezależnych decyzji binarnych.

- Skoro każda ze ścieżek jest niezależna, to wszystkie kombinacje wyboru ścieżek mogą być zamodelowane, tj. od wyboru wszystkich, aż do żadnej. Jednakże zaleca się takie modelowanie, aby co najmniej jedna ścieżka została osiągnięta.

Obiekty przepływu
BRAMY

Bramy łączne – *Inclusive Gateways*

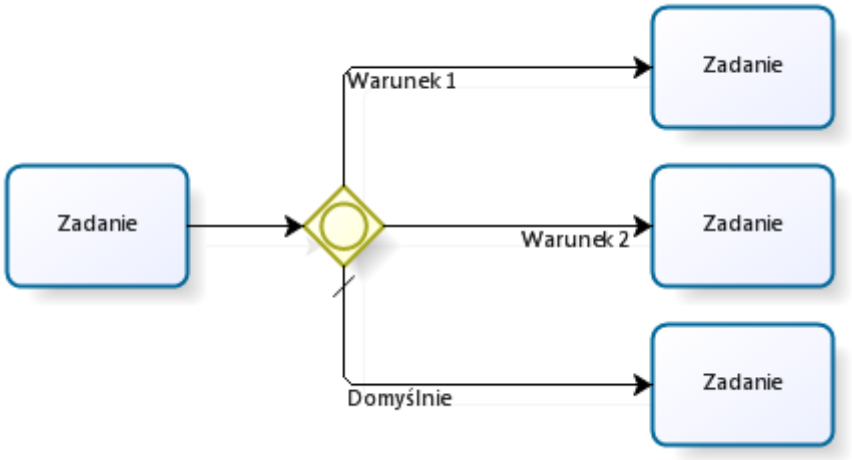
Przykład bramy łącznej używającej warunkowej sekwencji przepływu.



Obiekty przepływu
BRAMY

Bramy łączne – *Inclusive Gateways*

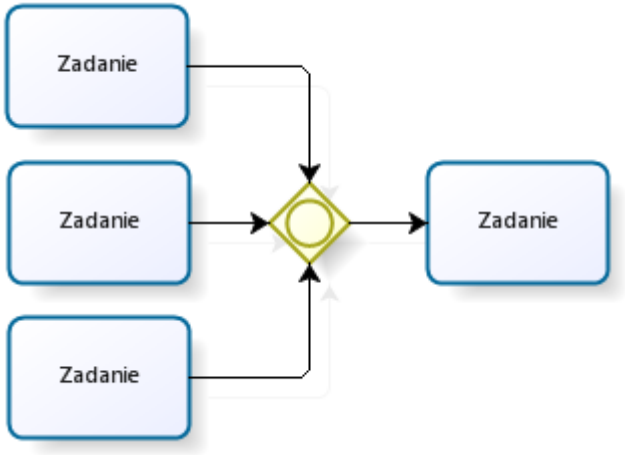
Przykład decyzji łącznej używającej bramy łącznej.



Obiekty przepływu
BRAMY

Bramy łączne – *Inclusive Gateways*

Przykład bramy łącznej łączącej sekwencje przepływu.



Obiekty przepływu
BRAMY

Bramy łączne – *Inclusive Gateways*

Łączenie bram łącznych z przepływem sekwencji (*Sequence Flow*):

- Jeżeli jest wiele przychodzących przepływów sekwencji, to jedna lub więcej z nich zostanie użyta do kontynuowania procesu.
- Proces będzie kontynuowany, gdy token dotrze ze wszystkich przychodzących przepływów sekwencji oczekujących na token w oparciu o strukturę przepływu procesu.
- Niektóre z przychodzących przepływów sekwencji nie będą miały tokenów, a wzór według którego przepływ sekwencji będzie miał tokeny może się zmieniać dla każdego z wywołań procesu.

Obiekty przepływu
BRAMY

Bramy łączne – *Inclusive Gateways*

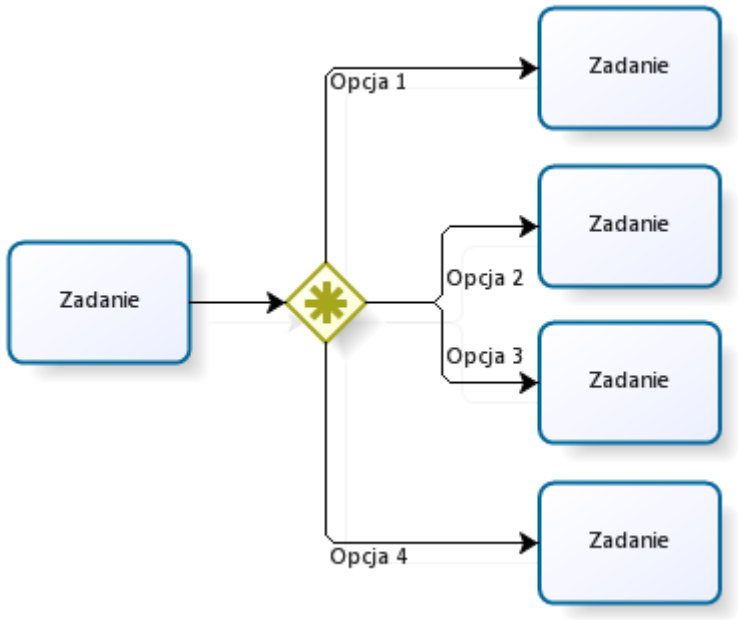
Łączenie bram łącznych z przepływem sekwencji (*Sequence Flow*):

- Jedna lub więcej bramek (*Gates*) wyjściowych może być wybranych w trakcie realizacji procesu.
- Bramki musza być wybierane na podstawie wyrażenia warunkującego, które jest zdefiniowane dla przepływu sekwencji związanego z bramkami (*Gates*):
  - Warunki związane ze wszystkimi bramkami (*Gate*) musza być sprawdzone.
  - Jeżeli warunek zostanie określony jako prawdziwy, wtedy ta bramka (*Gate*) zostaje wybrana, niezależnie od pozostałych.
  - Jeżeli żaden z warunków nie zostanie określony jako prawdziwy należy wybrać bramkę domyślną.

Obiekty przepływu
BRAMY

Bramy złożone – *Compex Gateways*

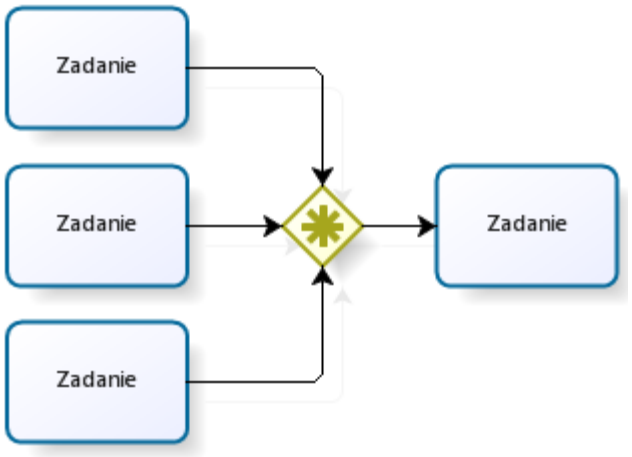
Bramy złożone są używane w sytuacjach, w których trudno byłoby zastosować inny typ bramy. Bramy złożone mogą być również do łączenia bram prostych w jeden większy mechanizm. Jeżeli brama złożona jest używana jako decyzja, to wyrażenie określa który z wychodzących przepływów sekwencji należy wybrać. Wyrażenie to może odwoływać się do danych procesu, jak i do przychodzących przepływów sekwencji.



Obiekty przepływu
BRAMY

Bramy złożone – *Compex Gateways*

- W przypadku użycia tej bramy jako złączenia przepływów sekwencji, wyrażenie określać będzie który z przychodzących przepływów sekwencji będzie wymagany aby proces mógł być kontynuowany



Obiekty przepływu
BRAMY

Bramy złożone – *Complex Gateways*

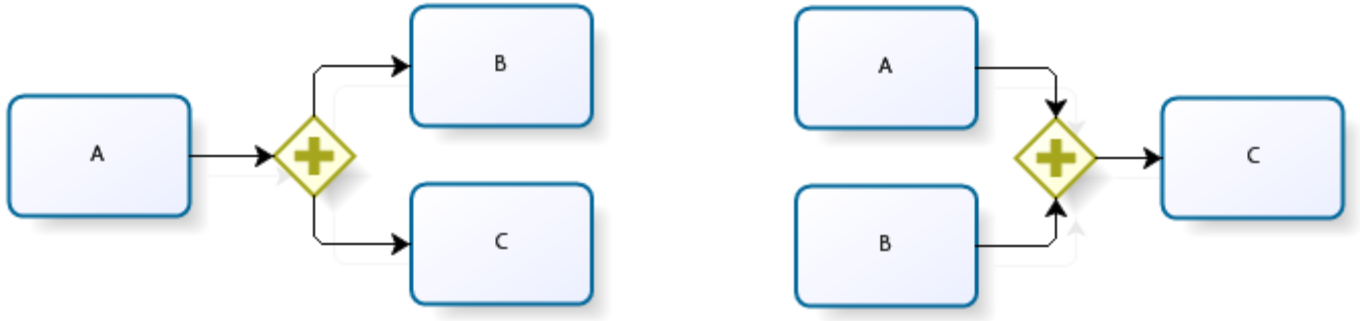
Powiązania z przepływem sekwencji:

- Jeżeli jest wiele przychodzących przepływów sekwencji, to jedna lub więcej z nich zostanie użyta do kontynuowania procesu. Dokładna kombinacja przychodzących przepływów sekwencji zostanie określona przez wyrażenie warunkowe przychodzące bramy (*Gateway*).
- Proces będzie kontynuowany gdy odpowiednia liczba tokenów dotrze do odpowiednich przepływów sekwencji.
- Niektóre z przychodzących przepływów sekwencji nie otrzymają tokenów, a wzór według którego tokeny są wymaga jest różny dla każdego z wywołań procesu.
- Tokeny z innych przepływów sekwencji z danego zbioru mogą przybyć, ale nie mogą być użyte do kontynuowania przepływu procesu
- Jedna lub więcej bramek (*Gates*) wyjściowych może być wybranych w trakcie realizacji procesu
- Bramki (*Gates*) muszą być wybierane na podstawie wyrażenia warunkującego wychodzącego bramy (*Gateway*).

Obiekty przepływu
BRAMY

Bramy równoległe – *Parallel Gateways*

Bramy równoległe dostarczają mechanizmu umożliwiające synchronizowanie równoległych przepływów jak i mechanizmu umożliwiającego tworzenie przepływów równoległych. Bramy te nie są wymagane do stworzenia przepływu równoległego, ale mogą być użyte w celu dokładniejszego określenie zachowań w złożonych sytuacjach, gdzie ciągi bram są używane i przepływ równoległy jest wymagany.



Obiekty przepływu
BRAMY

Bramy równoległe – *Parallel Gateways*

Powiązania z przepływem sekwencji:

- Jeżeli jest wiele przychodzących przepływów sekwencji, to wszystkie z nich zostaną użyte do kontynuowania procesu, a przepływ zostanie zsynchronizowany, tj.
  - Proces będzie kontynuowany, gdy token dotrze ze wszystkich przychodzących przepływów sekwencji (proces będzie czekał na przybycie tokenów do wszystkich przychodzących przepływów).

- Wszystkie bramki (*Gates*) wychodzące muszą zostać użyte przez proces.