

Dariusz Pietrzyk

**Modelowanie procesów biznesowych
w języku BPMN**

**(podręcznik do szkolenia –
Kraków, 14-16.09.2016)**

Spis treści

Ogólne wiadomości o języku BPMN	5
Standard BPMN	5
Narzędzia wspomagające	6
Kartka i ołówek	6
Programy specjalizowane do używania BPMN	6
Yaoqiang BPMN Editor	6
Podstawowe pojęcia	6
Czego potrzeba w procesach?	7
Typy diagramów BPMN	8
Podstawowe elementy procesu	9
Ramy procesu – zdarzenia (Events) i przepływy (Sequence Flows)	9
Konwencje	9
Reguły łączenia zdarzeń za pomocą przepływów	10
Przyczyny rozpoczęcia procesu – zdarzenia początkowe (Start Events)	10
Rodzaje przyczyn (wyzwalaczy) uruchamiających proces	10
Konwencje	11
Wyniki działania procesu – zdarzenia końcowe (End Events)	12
Rodzaje efektów działania procesu	12
Dodatkowe informacje na diagramie (Artifacts)	14
Przebieg procesu – czynności (Activities)	15
Zadania (Tasks) – czynności niepodzielne	15
Typy/rodzaje zadań ze względu na wykonywane czynności	16
Konwencje dla rodzajów zadań	18
Podprocesy (Sub-Processes) – czynności podzielne	19
Uściślenia przebiegu procesu	21
Zdarzenia pośrednie wpływające na proces (Intermediate Events)	21
Typy zdarzeń pośrednich	21
Zdarzenia w głównym przebiegu procesu	22
Typy zdarzeń pośrednich włączanych do czynności	29
Zdarzenia włączane do czynności	30
Zróżnicowane ścieżki działania w procesie – bramki (Gateways)	39
Połączenia bramek przepływami sterowania (Sequence Flows)	40
Wyjście tylko jedną ścieżką (Exclusive Gateway)	41
Wyjście co najmniej jedną ścieżką (Inclusive Gateway)	42
Wyjście wieloma ścieżkami równocześnie (Parallel Gateway)	43
Złożona (Complex Gateway)	44
Wyjście zależne od zdarzenia (Event-Based Gateway)	45
Reguły łączenia bramek za pomocą Sequence Flows	47
Przetwarzanie grup obiektów – czynności iteracyjne i zwielokrotnione	48
Iteracyjna czynność powtarzalna (Loop Activity)	49
Czynność zwielokrotniona dla obiektów w grupie (Multiple Instances Activity)	50
Dane wykorzystywane w procesie (Data Objects)	51
Pobieranie/wprowadzanie danych (Data Input)	53
Przekazywanie danych wyjściowych (Data Output)	53
Połączenia danych z elementami procesu (Data Associations)	55
Użycie predefiniowanego procesu lub zadania (Call Activity)	56
Nietypowe etapy procesu	58
Nadzwyczajna ingerencja w działanie procesu (Escalation)	58
Uzgłađnianie wpływu czynników zewnętrznych – podprocesy wywoływanie zdarzeniami (Event Sub-Process)	59

Podprocesy zależne od wykonawcy (Ad-Hoc Sub-Process)	62
Błędy i przerwania procesu	64
Obsługa błędów procesu	64
Przerwania procesu	65
Wycofanie kroków zakończonej czynności (Compensation)	65
Uruchamianie kompensacji	66
Konwencje dla kompensacji	66
Związek między obsługą błędów a kompensacją	67
Podprocesy transakcyjne (Transactions)	68
Poprawne zakończenie transakcji	68
Anulowanie transakcji (Cancel Event)	69
Błąd transakcji (Error Event)	69
Interakcje między uczestnikami procesu	71
Schemat współpracy w procesie (Collaboration)	71
Baseny (Pools)	71
Tory (Lanes)	72
Komunikacja między uczestnikami procesu	72
Schemat koordynacji działań uczestników procesu (Choreography)	73
Poglądowy schemat komunikacji w procesie (Conversation)	76
Bibliografia	79

Ogólne wiadomości o języku BPMN

I Standard BPMN

• **BPMN:**

- pierwotnie – Business Process Modeling Notation (Notacja Modelowania Procesów Biznesowych),
- obecnie – Business Process Model and Notation (Modelowanie i Dokumentowanie Procesów Biznesowych).
- Jest to graficzny język, który w obszarze procesów biznesowych wspomaga:
 - projektowanie – planowanie płynnie działających i skutecznych procesów,
 - dokumentowanie – zapis i standaryzację procesów już funkcjonujących,
 - ulepszanie i standaryzację procesów.
- Nie jest językiem programowania – służy komunikacji międzyludzkiej.
- Oferuje zestaw ujednoliconych symboli graficznych ułatwiających zapis przebiegu różnorakich procesów oraz czynników wpływających bezpośrednio na ich działanie i efekty.
- Jest utrzymany w zgodności z koncepcją SOA (ang. Service-Oriented Architecture), czyli architektury zorientowanej na usługi (dla użytkownika systemu).
- Jest standardem, czyli – jeśli jest używany dokładnie zgodnie ze specyfikacją – daje m.in. możliwość przekazania skutecznego opisu poszczególnych procesów osobom, które będą w stanie zrozumieć proces, nawet jeśli nie otrzymają jego dodatkowego opisu tekstowego.
- Standard ten jest wspierany i upowszechniany przez organizację OMG (Object Management Group – zob. też www.omg.org).
- Język powstał w roku 2004 (BPMN 1.0), zaś obecnie (od ok. 2009 r.) obowiązuje wersja 2.0.
- Specyfikacja języka (czyli dokument normatywny dotyczący jego zawartości i używania) znajduje się w specjalizowanym serwisie OMG – zob. www.bpmn.org.
- Model procesu biznesowego w języku BPMN:
 - Stanowi zwykle punkt wyjścia do tworzenia modeli systemów wspomagających realizację procesu (np. w języku UML).
 - Może stanowić podstawę do przejścia na model BPEL – Business Process Execution Language lub dokładniej Business Process Execution Language for Web Services (WS-BPEL), czyli bazujący na XML-u język definiowania procesów biznesowych opartych na usługach sieciowych.

I Narzędzia wspomagające

Kartka i ołówek

- BPMN jest językiem graficznym, zatem można stworzyć prostszy diagram nawet bez użycia komputera. Czasem – w przypadku pojedynczych, prostych diagramów – nawet szybciej będzie uchwycić myśl na kartce niżłączyć komputer i program wspomagający.
- Np.: szkic pomysłu lub schemat wspomagający dyskusję o jakimś rozwiązaniu podczas biznesowego obiadu czy w kawiarni. Nawet brak komputera nie uniemożliwi wtedy zanotowania dobrego pomysłu czy rodzącego się rozwiązania.
- Kartka i ołówek są ciągle lżejsze i bardziej dostępne od komputerów, a poza tym zanim pojawiły się komputery, wiele dobrych pomysłów powstało i zostało zrealizowanych w oparciu właśnie o te najprostsze narzędzia ;--)

Programy specjalizowane do używania BPMN

- Ich zaletą jest przede wszystkim pełniejsza obsługa standardu BPMN oraz możliwość (w ograniczonym zakresie) tworzenia powiązań między diagramami modelu (uniwersalne programy do tworzenia schematów takich możliwości nie mają – oferują tylko „statyczne” zestawy diagramów).
- Jeden z dostępnych wykazów narzędzi wspomagających posługiwanie się językiem BPMN można znaleźć pod adresem: http://en.wikipedia.org/wiki/Comparison_of_Business_Process_Modeling_Notation_tools (przede wszystkim informacje o wspieranych platformach systemowych, wersjach darmowych i płatnych oraz podstawowych funkcjach).

Yaoqiang BPMN Editor

- Wieloplatformowy – Windows/Linux/Mac OS (potrzebne JRE lub JDK).
- Darmowy (licencja GPL) do wersji 2.2 włącznie.
- Intuicyjny i prosty w obsłudze.
- Ma wiele przydatnych funkcji – m.in. proste zapisywanie diagramów w różnych formatach graficznych (np. PNG) oraz kontrolę składni.

I Podstawowe pojęcia

- Biznes – działalność mająca na celu zaspokojenie ludzkich potrzeb i polegająca na wymianie wartości (najczęściej, choć niekoniecznie, poprzez środek wymiany, jakim jest pieniądz).
- Proces biznesowy (metoda biznesowa):
 - Seria powiązanych ze sobą działań lub zadań, które rozwiązują określony problem lub prowadzą do osiągnięcia określonego efektu.
 - Bywa opisywany schematem blokowym.

- Wynika z potrzeb klientów, a jego wynikiem jest zaspokojenie tych potrzeb (określony, mierzalny cel).
- Typy procesów biznesowych:
 - Proces zarządczy – kieruje działaniem systemu (np. proces zarządzania przedsiębiorstwem lub zarządzania strategicznego).
 - Proces operacyjny – stanowi istotę biznesu i jest pierwotnym źródłem wartości dodanej (np. zaopatrzenie, produkcja, marketing, sprzedaż).
 - Proces pomocniczy – wspiera procesy główne (np. księgowość, rekrutacja, wsparcie techniczne).
- Proces biznesowy można podzielić na podprocesy o własnych atrybutach, które dają wkład w rezultat procesu nadziednego.
- Wymagane cechy procesu biznesowego:
 - Definiowalność – musi mieć jasno zdefiniowane granice, wejście i wyjście.
 - Porządek – musi się składać z działań uporządkowanych (usytuowanych w czasie i przestrzeni).
 - Inicjator – proces musi zostać uruchomiony, mieć przyczynę sprawczą (może to być osoba uruchamiająca/inicjująca, ale również polecenie zewnętrzne, inny proces, upływ czasu lub punkt w czasie, sygnał itp.).
 - Klient – musi być odbiorca rezultatów procesu.
 - Zwiększenie wartości – transformacja w trakcie procesu musi dawać odbiorcy dodatkową wartość.
 - Osadzenie – nie może egzystować samodzielnie, musi być wbudowany w strukturę organizacyjną.
 - Wielofunkcyjność – może (ale nie musi) obejmować wiele funkcji.
- Właściciel procesu:
 - osoba odpowiedzialna za jego działanie i ciągłe ulepszanie,
 - podstawowe źródło informacji dla analityka.

I Czego potrzeba w procesach?

- Elementów rutynowych, czyli:
 - przyczyny rozpoczęcia/uruchomienia (Start Events),
 - wykonywania rutynowych działań zmierzających do osiągnięcia celu procesu (Activities – Tasks i Sub-Processes),
 - sterowania przebiegiem (Gateways, Intermediate Events),
 - obsługi/przetwarzania/używania danych (Data Objects),
 - przetwarzania grup obiektów (Loop Activity, Multiple Instances Activity),
 - efektu – wyniku/produkту działania procesu (End Events).

- Elementów nierutynowych:
 - obsługi błędów (Error Event),
 - przerwania przebiegu procesu (Cancel Event, Terminate Event),
 - ścisłej kontroli kompletności realizacji niektórych części procesu (Transaction Sub-Process),
 - wycofania wykonanych wcześniej działań w przypadku anulowania/przerwania procesu (Compensation Events),
 - uwzględniania wpływu czynników zewnętrznych (podprocesy Event Sub-Process),
 - podjęcia nierutynowych działań (Escalation Events, Ad-Hoc Sub-Process).
- Poszerzonej standaryzacji działań powtarzalnych (Global Tasks i Call Activity).
- Interakcji między uczestnikami procesu, czyli przede wszystkim:
 - podziału odpowiedzialności za wykonanie poszczególnych działań (Pools i Lanes),
 - komunikacji między uczestnikami (Message Flows).

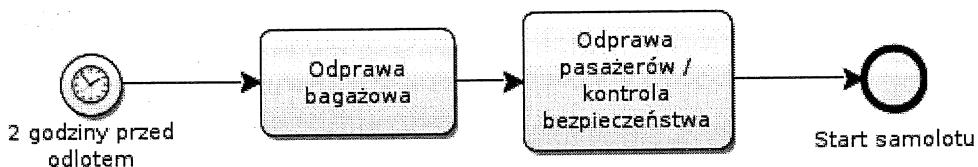
| Typy diagramów BPMN

- Język/standard BPMN daje możliwość obrazowania procesów z kilku różnych punktów widzenia odzwierciedlonych na różnych, niżej wymienionych typach diagramów.
- Process/Orchestration:
 - partitura lub aranżacja (w muzyce jest to rozpis melodii na instrumenty),
 - szczegółowy przebieg procesu,
 - podstawowy i najważniejszy diagram, który ma swoje uzupełnienia w trzech pozostałych typach diagramów.
- Collaboration:
 - schemat współpracy między uczestnikami procesu,
 - oparty na diagramach Process/Orchestration.
- Conversation:
 - konwersacja,
 - poglądowy schemat komunikacji między uczestnikami procesu z punktu widzenia obiektów biznesowych „materializujących” tę komunikację (np. zamówienia, zlecenia, przelewy, harmonogramy itp.).
- Choreography:
 - choreografia (w muzyce jest to układ par tanecznych oraz ich poruszania się w trakcie wykonania utworu),
 - binarny albo szczegółowy schemat wymiany informacji między uczestnikami procesu, pokazujący przede wszystkim synchronizację ich działań.

Podstawowe elementy procesu

Ramy procesu – zdarzenia (Events) i przepływy (Sequence Flows)

- Symbol zdarzenia w diagramie BPMN – odpowiednik zdarzenia w modelowanym procesie biznesowym.
- Symbol przepływu sterowania – pokazuje kolejność występowania zdarzeń oraz wykonywania czynności składowych procesu.
- Są to elementy niezbędne do opisania (utworzenia modelu) podstawowego (najbardziej ogólnego) przebiegu procesu.
- Zdarzenia stanowią szkielet procesu, ponieważ wpływają na sterowanie jego przebiegiem. Np. proces rozpoczyna się od konkretnego zdarzenia uruchamiającego, niektóre zdarzenia pośrednie wywołują zmiany warunków działania procesu, a zatem wymagają podjęcia decyzji co do przebiegu ścieżki kolejnych działań itp.
- Przykładowy model ogólny procesu (zdarzenia, zadania i przepływy):



- Token ('żeton'):
 - Abstrakt ułatwiający wyobrażenie sobie zachowania konkretnego procesu realizowanego w oparciu na tworzonym/analizowanym modelu.
 - Zachowanie elementów procesu może być opisywane jako sposób, w jaki oddziałują one na token „poruszający się” w obrębie procesu.
 - Inaczej mówiąc, token oznacza konkretne „miejsce”, w którym znajdujemy się w konkretnym momencie analizy przebiegu procesu.

Konwencje

Symbol	Element	Opis
	Event – zdarzenie	Coś, co się dzieje („przydarza”) w trakcie trwania procesu. Zwykle ma jakąś przyczynę (trigger) oraz rezultat / wynik / skutek (result). Symbol zdarzenia może zawierać wewnątrz uściślenie przyczyny oraz rezultatu.
○	Start Event – zdarzenie początkowe	Zdarzenie, które rozpoczyna proces. Czasem może zostać pominięte (zob. specyfikacja BPMN, s. 238-9), ale dla zachowania czytelności i przejrzystości diagramu warto je umieszczać.

Symbol	Element	Opis
	Intermediate Event – zdarzenie pośrednie	Zdarzenie zachodzące pomiędzy początkiem a końcem procesu. Wpływa na przebieg procesu, ale nie rozpoczyna ani (bezpośrednio) nie kończy procesu głównego.
	End Event – zdarzenie końcowe	Zdarzenie, które kończy proces (moment zakończenia procesu).
	Sequence flow – przepływ sterowania	Połączenie pokazujące kolejność zachodzenia zdarzeń oraz wykonywania czynności w procesie.

Reguły łączenia zdarzeń za pomocą przepływów

- Zdarzenie początkowe:
 - musi rozpoczynać co najmniej jeden przepływ sterowania,
 - nie może otrzymywać przepływów sterowania.
- Zdarzenie końcowe:
 - musi otrzymywać co najmniej jeden przepływ sterowania,
 - nie może rozpoczynać żadnego przepływu sterowania.
- Dopuszczalne przepływy sterowania – ze zdarzenia:
 - początkowego do pośredniego,
 - początkowego do końcowego,
 - pośredniego do pośredniego,
 - pośredniego do końcowego.

Przyczyny rozpoczęcia procesu – zdarzenia początkowe (Start Events)

Rodzaje przyczyn (wyzwalaczy) uruchamiających proces

- Proces główny może rozpoczynać się bez zdefiniowanej przyczyny, jednak najczęściej można wyraźnie tę przyczynę określić.
- Możliwe kategorie przyczyn:
 - odebranie wiadomości (np. otrzymanie maila, przyjęcie zamówienia itp.),
 - okres czasu (np. dzień przed...), częstotliwość (np. w każdy ostatni dzień miesiąca) lub punkt w czasie (np. 1 kwietnia 2015),
 - spełnienie określonego warunku (np. zwiększenie liczby zgłoszeń reklamacyjnych o 5%),

- odebranie sygnału (np. o konieczności uzupełnienia zapasów),
- wystąpienie jednej z kilku konkretnych przyczyn (np. odebranie wiadomości albo odebranie sygnału),
- równoczesne wystąpienie kilku konkretnych przyczyn (np. sygnał o konieczności uzupełnienia zapasów i wiadomość o stanie budżetu umożliwiającym realizację zamówienia).
- **Uwaga:** sygnał nie jest wiadomością. Wiadomość ma konkretnego nadawcę i adresata, zaś sygnał jest wysyłany na zasadzie rozgłoszenia (broadcast), „wysłania w eter” bez konkretnego adresata, i może być wysłany przez jeden proces, a odebrany przez wiele procesów jednocześnie (i procesy te zarazem rozpoczynać).

Np. sygnał dźwiękowy alarmu samochodowego jest przeznaczony dla osób, znajdujących się w zasięgu słyszalności syreny (i ma zwrócić uwagę takich osób na samochód), zaś osoba zaalarmowana może przekazać wiadomość skierowaną do policji o próbie włamania/kradzieży (z dodatkową konkretną informacją o miejscu zdarzenia).

Konwencje

Symbol	Przyczyna / wyzwalacz	Opis
	None (brak / nieokreślona)	Brak/bez zdefiniowanej przyczyny rozpoczęcia.
	Message (wiadomość)	Otrzymanie pojedynczej wiadomości od uczestnika procesu (Participant). Występuje w sytuacji, gdy użyte jest zarazem oznaczenie przepływu wiadomości (Message Flow) w ramach współpracy z uczestnikiem procesu.
	Timer (data/czas)	Osiągnięty został punkt w czasie lub początek kolejnego cyklu.
	Conditional (warunek)	Został spełniony określony (pojedynczy) warunek. Aby proces został uruchomiony ponownie, warunek musi najpierw przyjąć wartość logiczną „Fałsz”, a następnie ponownie wartość „Prawda”. Wyrażenie określające warunek może (ale nie musi) odnosić się bezpośrednio do procesu, jego atrybutów czy też stanów.
	Signal (sygnał)	Został odebrany pojedynczy sygnał pochodzący np. z innego procesu.

Symbol	Przyczyna / wyzwalacz	Opis
	Multiple (jedna z kilku możliwych)	Oznacza, że nastąpiła jedna (i tylko jedna) z co najmniej dwóch różnych możliwych przyczyn uruchamiających proces.
	Parallel Multiple (równocześnie kilka możliwych)	Oznacza, że jest wiele różnych możliwych przyczyn uruchamiających proces, przy czym do uruchomienia procesu niezbędne jest wystąpienie wszystkich jednocześnie.

Wyniki działania procesu – zdarzenia końcowe (End Events)

Rodzaje efektów działania procesu

- Wiadomość (Message) – wysłana do uczestnika (odbiorcy efektów) procesu lub do innego procesu.
- Błąd lub wyjątek (Error):
 - Typowy, przewidywalny i możliwy (oraz konieczny) do obsłużenia (np. poprawienia lub przynajmniej zasygnalizowania uczestnikom procesu przyczyn jego wystąpienia w celu podjęcia odpowiednich działań naprawczych/awaryjnych).
 - Jeśli wykracza poza te ramy, być może konieczne jest zastosowanie oznaczenia eskalacji lub natychmiastowego przerwania.
- Eskalacja (Escalation) – sytuacja biznesowa, która może wymagać specjalnej/nadzwyczajnej interwencji np. menedżera lub innej osoby kompetentnej do rozwiązyania nietypowego problemu, który pojawił się w trakcie trwania procesu.
- Natychmiastowe przerwanie (Terminate):
 - Następujące w nadzwyczajnych sytuacjach z przyczyn obiektywnych (wykraczających poza możliwości działania i kontroli jednostki organizacyjnej, systemu lub osób realizujących proces).
 - Np. katastrofa lotnicza czy klęska żywiołowa.
- Kompensacja (Compensation):
 - Wycofanie wykonanych działań ze względu na to, iż ich efekty nie są już potrzebne (w procesie nadziednym lub dla obiektów korzystających z wyników działania procesu).
 - Występuje najczęściej w momencie anulowania procesu transakcyjnego bądź też wystąpienia wyjątku/błędu uniemożliwiającego kontynuację procesu.
 - Np. odwołanie przez przedstawiciela firmy uczestnictwa w targach wywoła automatycznie konieczność odwołania dla niego rezerwacji miejsca hotelowego i kateringu.

- Kompensacja może czasem wymagać wycofywania działań w kolejności odwrotnej do ich pierwotnego wykonywania.
- Możliwa jest też sytuacja, gdy kompensacja jednej czynności automatycznie będzie pociągała za sobą konieczność kompensacji innej (lub innych) czynności.
- Sygnał (Signal):
 - Wysłany do uczestników lub do innych procesów.
 - Np. zgłoszenie przestoju (biegu jałowego) wynikającego z braku zamówień do realizacji, które trafia do wszystkich jednostek lub osób zajmujących się sprzedażą.
 - Wiele efektów jednocześnie (Multiple) – np. wysłanie kilku wiadomości do różnych adresatów (powiadomienie kuriera o konieczności dostarczenia zamówienia oraz powiadomienie klienta o przewidywanym terminie odbioru zamówienia).

Symbol	Efekt / wynik	Opis
	None (brak)	Brak/bez zdefiniowanego efektu procesu.
	Message (wiadomość)	Efektem końcowym procesu jest wiadomość wysłana do uczestnika (Participant). Występuje w sytuacji, gdy użyte jest zarazem oznaczenie przepływu wiadomości (Message Flow) w ramach współpracy z uczestnikiem procesu.
	Error (błąd)	Efektem końcowym procesu jest przerwanie wszystkich aktywnych wątków procesu oraz wygenerowanie określonego rodzaju błędu. Błąd powinien zostać obsłużony przez inny, przeznaczony do tego celu proces.
	Escalation (nadzwyczajna interwencja)	Wymagana nadzwyczajna interwencja (np. przejęcie przez człowieka ręcznego sterowania procesem, który rutynowo jest realizowany automatycznie).
	Cancel (anulowanie)	Związane wyłącznie z podprocesami transakcyjnymi (czyli działającymi na zasadzie „wszystko albo nic”). Oznacza, że transakcja ma zostać anulowana i uruchomić zdarzenie pośrednie tego samego rodzaju, znajdujące się w obrębie danego podprocesu. Dodatkowo oznacza też, że protokół transakcyjny (Transaction Protocol) powinien wysłać wiadomość o anulowaniu do wszystkich elementów związanych z (zainteresowanych) daną transakcją.
	Compensation (kompensacja)	Sygnalizuje, że potrzebna jest kompensacja działań wykonanych w trakcie procesu ¹ lub podprocesu.

¹ Specyfikacja BPMN jest w tym miejscu nieco nieprecyzyjna – na s. 278, przy omówieniu oznaczania efektów działania procesów wspomina właściwie tylko o czynnościach i podprocesach, a przecież kompensacji może wymagać też proces jako całość (np. w przypadku wystąpienia błędu przerywającego działanie procesu). Do tego na s. 292 podaje wyraźnie, że zdarzenie typu kompensacja może kończyć

Symbol	Efekt / wynik	Opis
	Signal (sygnal)	Efektem końcowym jest pojedynczy sygnał wysłany do innych procesów. Np. zakończenie procesu weryfikacji stanu zapasów może dać w efekcie sygnał o konieczności rozpoczęcia procesu uzupełnienia magazynu.
	Terminate (natychmiastowe przerwanie)	Oznacza, że wszystkie czynności w procesie muszą być natychmiast przerwane, ale nie nastąpi dla nich kompensacja ani inny rodzaj obsługi zdarzenia. Np. przerwanie procesu realizacji zamówień z powodu klęski żywiołowej (i np. zniszczenia fabryki).
	Multiple (kilka możliwych)	Oznacza, że jest wiele różnych możliwych efektów zakończenia procesu, przy czym pojawiają się wszystkie (np. wysłanie kilku różnych wiadomości do różnych odbiorców).

Dodatkowe informacje na diagramie (Artifacts)

- Artefakty (Artifacts):
 - Elementy, które rozszerzają możliwości języka BPMN.
 - Mogą być stosowane na dowolnym diagramie.
 - Nie mogą rozpoczynać przepływów sterowania i wiadomości ani być dla tych przepływów elementami docelowymi.
- Typy artefaktów standardowych:
 - Notatka (Text Annotation) – dodatkowe informacje dla odbiorcy diagramu (np. tytuł diagramu, opis, uściślenie, komentarz, uwaga itp.).
 - Grupa (Group) – umożliwia dodatkowe grupowanie elementów diagramu (np. niezależnie od basenów czy torów), ale jej obecność nie ma wpływu na działanie procesu.
 - Asocjacja (Association) – służy do łączenia pozostałych artefaktów z innymi elementami diagramu. Asocjacja może być ukierunkowana jedno- lub obustronnie.

Symbol	Nazwa	Opis
	Text Annotation (notatka)	Dodatkowe informacje dla odbiorcy diagramu (tu: z widoczną asocjacją prowadzącą do opisywanego przez notatkę elementu diagramu).
	Group (grupa)	Środek wizualny do dowolnego grupowania elementów diagramu (np. niezależnie od basenów czy torów).
	Association (asocjacja)	Łącznik artefaktu z innymi elementami diagramu.

dowolny proces lub podproces.

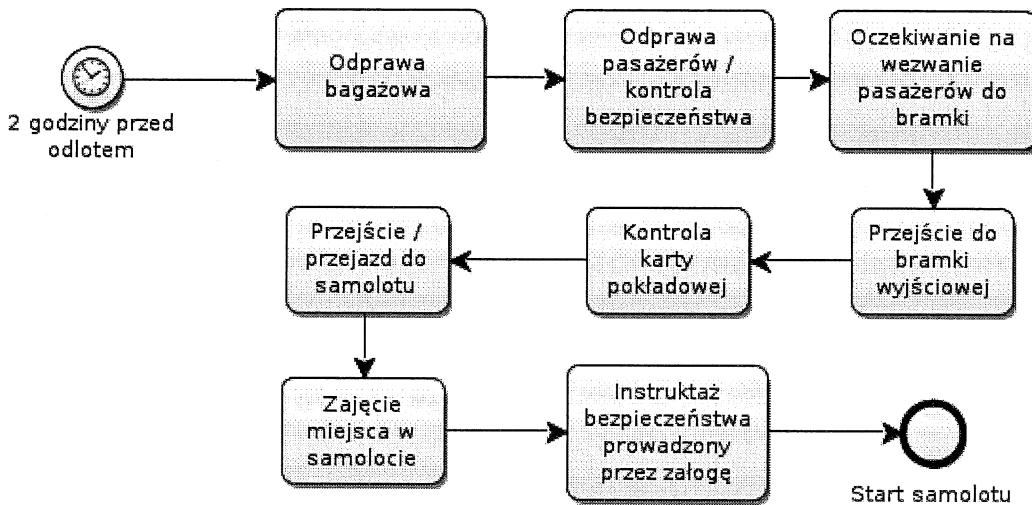
■ Przebieg procesu – czynności (Activities)

- Rutynowy przebieg procesu (przebieg normalny, normal flow) – obejmuje przede wszystkim czynności zaplanowane lub wynikające w konieczny sposób z celu, który ma zostać osiągnięty przez proces.
- Czynność – (konkretna) praca wykonywana w trakcie procesu lub inaczej: punkt/moment procesu, w którym wykonywana jest owa praca.
- Pojedyncza czynność może posiadać wiele przepływów sterowania (Sequence Flows) przychodzących (incoming), jak też wychodzących (outgoing), przy czym:
 - jeśli ma wiele przychodzących, oznacza to, że są to przepływy niekontrolowane – zostanie wykonana za każdym razem, gdy dostrze do niej token z którykolwiek ze ścieżek wejściowych (jeśli potrzebna jest kontrola, przed czynnością musi wystąpić bramka),
 - jeśli ma wiele wychodzących, traktowane są one jako przepływ równoległy/współbieżny (token jest generowany dla każdej ze ścieżek wychodzących).
- Pojedyncza czynność (z wyjątkiem Compensation Activities oraz Event Sub-Processes) może również:
 - nie posiadać przepływów sterowania przychodzących – wówczas jest ona uruchamiana w każdym uruchomieniu (instancji) procesu, którego jest częścią,
 - nie posiadać przepływów sterowania wychodzących (jedynie przychodzące) – wówczas proces po dojściu do tej czynności się kończy (bez oznaczania na diagramie zdarzenia kończącego).
- Pojedyncza czynność może też posiadać wiele przepływów wiadomości (Message Flows) przychodzących (incoming), jak też wychodzących (outgoing).
- Czynności dzielą się na:
 - niepodzielne – zadania (Tasks),
 - podzielne – podprocesy (Sub-Processes).

Zadania (Tasks) – czynności niepodzielne

- Zadanie – elementarna, niepodzielna czynność występująca w procesie.
- Zadanie jest na diagramie używane, gdy trzeba oznaczyć wykonanie konkretnej pracy, której nie da się rozbić na czynności składowe (bardziej szczegółowe działania) lub gdy nie ma potrzeby takiego uszczegółowienia pokazywać.
- Zadanie może być początkiem przepływu sterowania prowadzącego do:
 - podprocesu, bramki, zdarzenia pośredniego i zdarzenia końcowego – zawsze,
 - innego zadania – czasem (np. w podprocesach Ad-Hoc zadanie nie musi rozpoczynać przepływu sterowania),
 - zdarzenia początkowego – nigdy.
- Zadanie może znajdować się na końcu przepływu sterowania prowadzonego ze:

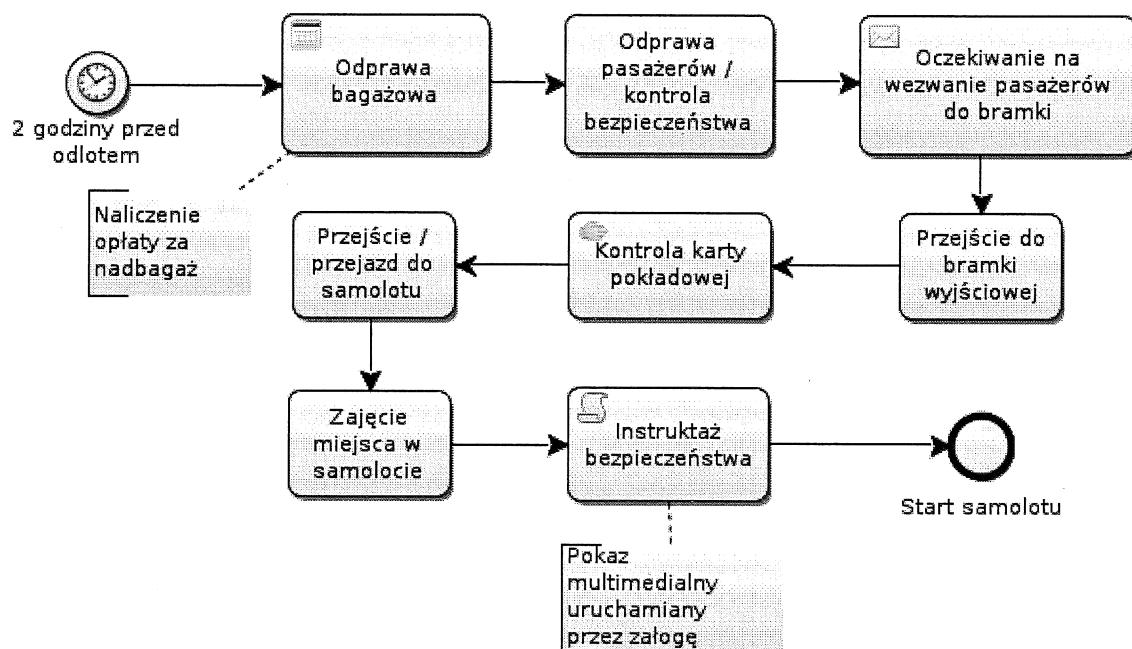
- zdarzenia początkowego, innego zadania, podprocesu, bramki i zdarzenia pośredniego – czasem,
- zdarzenia końcowego – nigdy.
- Przykład bardziej rozbudowanego procesu:



Typy/rodzaje zadań ze względu na wykonywane czynności

- Zadania obejmują wykonywanie czynności kilku określonych typów.
- Usługa (Service Task):
 - Podczas realizacji zadania wykorzystywana jest usługa.
 - Np. skorzystanie ze strony internetowej.
 - Na diagramie współpracy uczestnik, który realizuje usługę, może być wskazany połączeniem w postaci przepływu wiadomości (Message Flows).
- Wysłanie wiadomości (Send Task):
 - Wysłanie wiadomości do uczestnika spoza procesu.
 - Zadanie jest ukończone, gdy wiadomość zostaje wysłana.
 - Np. wysłanie do klienta powiadomienia o planowanym/szacowanym terminie realizacji zamówienia.
 - Na diagramie współpracy uczestnik, który wysyła wiadomość, może być wskazany połączeniem w postaci przepływu wiadomości (Message Flow) dołączonego do adresata wiadomości.
- Odebranie wiadomości (Receive Task):
 - Oczekiwanie na wiadomość pochodząą od uczestnika spoza procesu.
 - W momencie odebrania wiadomości zadanie jest zakończone.
 - Np. oczekiwanie przez sprzedawcę na potwierdzenie dokonania wpłaty przez klienta.

- Na diagramie współpracy uczestnik, od którego wiadomość pochodzi, może być wskazywany poprzez przepływ wiadomości (Message Flow).
- Zadanie odebrania wiadomości często jest wykorzystywane do rozpoczęcia procesu (wówczas nie może mieć żadnych przychodzących przepływów sterowania).
- Wykonywane przez użytkownika (User Task):
 - Typowa sytuacja, w której użytkownik wykonuje zadanie przy użyciu programu komputerowego, zgodnie z planem zadań do realizacji, który również jest zarządzany za pomocą jakiejś aplikacji (programu komputerowego).
 - Zadanie, w którego realizację jest bezpośrednio zaangażowany człowiek.
 - Np. przygotowanie oferty (pliku DOC/ODT/RTF) zgodnie z harmonogramem prac (zapisanym w aplikacji typu PIM – Personal Information Manager).
- Manualne (Manual Task):
 - Zadanie wykonywane bez udziału oprogramowania komputerowego (w tym również bez używania aplikacji zarządzającej momentem rozpoczęcia i zakończenia realizacji zadania).
 - Drugi typ zadania, w którego realizację jest bezpośrednio zaangażowany człowiek.
 - Np. monter wymienia okno u klienta, zaś zlecenie na wykonanie tej pracy otrzymuje w formie papierowego formularza.
- Zastosowanie reguły biznesowej (Business Rule Task):
 - Przewiduje użycie mechanizmu (niekoniecznie informatycznego), który na podstawie reguł działania umożliwia podjęcie decyzji/wyboru odpowiedniego do zaistniałej sytuacji (wprowadzamy dane wejściowe i uzyskujemy wynik na podstawie zapisanej/ustalonej reguły).
 - Zadanie to umożliwia zarazem procesowi przekazanie i odebranie danych od mechanizmu przetwarzającego reguły biznesowe.
 - Na początku zadania jest wywoływana (called) reguła biznesowa, zaś zadanie jest zakończone, gdy reguła zostanie zrealizowana/otrzymamy jej wynik (on completion).
 - Np. zasada stosowania w nagłówku korespondencji adekwatnego zwrotu grzecznościowego (z ustaleniem kryteriów wyboru tego zwrotu), która w efekcie da konkretne zwroty grzecznościowe odpowiednie do osoby adresata – Szanowny Panie/Szanowna Pani, czy też Szanowny Panie Profesorze/Doktorze/Inżynierze.
- Wykonanie skryptu (Script Task):
 - Uruchomienie skryptu wykonywanego przez mechanizm przetwarzający (business process engine).
 - Gdy zadanie jest gotowe do realizacji, skrypt jest uruchamiany, natomiast gdy skrypt zostanie zakończony, zadanie również jest zakończone.
 - Analityk może do zadania dołączyć skrypt.
 - Np. użycie makra wysyłającego maile do członków zespołu.
- Przykład procesu z uściśleniem rodzajów zadań (odprawa z punktu widzenia pasażera):



Konwencje dla rodzajów zadań

Symbol	Rodzaj zadania	Opis
Label	Service (usługa)	Zadanie, które opiera się na realizacji różnego rodzaju usług, np. automatycznie działającej aplikacji.
Label	Send (wysłanie wiadomości)	Wysłanie wiadomości do uczestnika spoza procesu.
Label	Receive (odebranie wiadomości)	Oczekiwanie na wiadomość pochodząą od uczestnika spoza procesu.
Label	Receive instantiates process (odebranie wiadomości uruchamiającej proces)	Wariant zadania odebrania wiadomości, który stanowi zarazem początek procesu (bez zdarzenia początkowego). Np. zamówienie od klienta (rozpoczynające realizację), zlecenie na wykonanie usługi (rozpoczynające jej wykonanie).
Label	User (wykonywane przez użytkownika)	Wykonanie zaplanowanego zadania przy użyciu oprogramowania komputerowego.

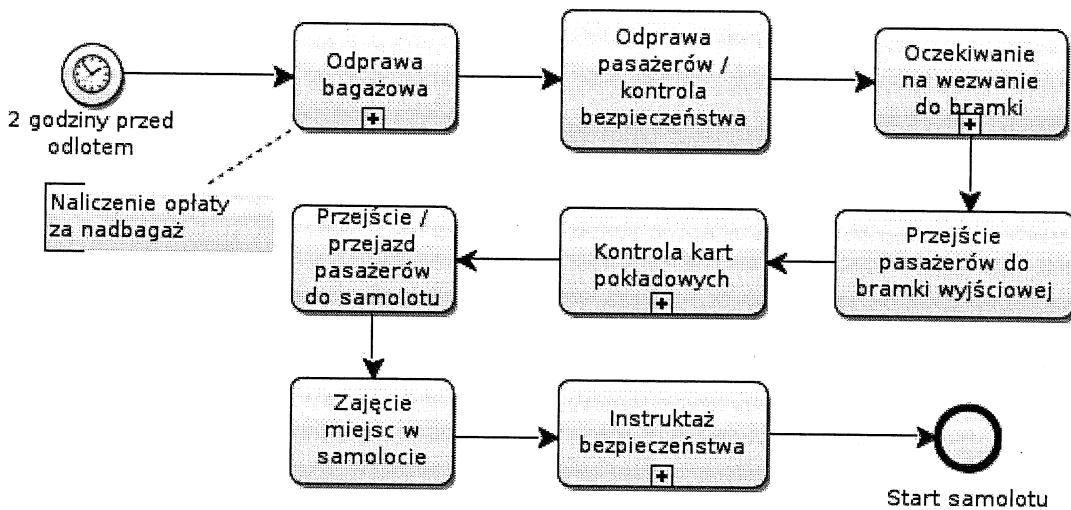
Symbol	Rodzaj zadania	Opis
	Manual (manualne)	Wykonanie zadania bez użycia programu komputerowego (również do bezpośredniego odnotowywania momentu rozpoczęcia i zakończenia realizacji).
	Business Rule (reguła biznesowa)	Zastosowanie reguły/zasady w celu wyboru sposobu działania lub podjęcia decyzji zależnej od sytuacji.
	Script (wykonanie skryptu)	Wykonanie skryptu przez mechanizm przetwarzający.

Podprocesy (Sub-Processes) – czynności podzielne

- Podproces (Sub-Process) – czynność, której wewnętrzne szczegóły zostały zamodelowane za pomocą czynności, bramek, zdarzeń i przepływów sterowania.
- Graficznie podproces może być pokazany w diagramie jako:
 - „zwinięty” (collapsed) – ze szczegółami ukrytymi (pokazuje wówczas tylko lokalizację podprocesu w procesie nadziednym),
 - „rozwinięty” (expanded) – z widocznymi szczegółami.

Symbol	Nazwa	Opis
	Podproces „zwinięty” (collapsed)	Oznaczenie ogólne procesu z ukrytymi szczegółami.
	Podproces „rozwinięty” (expanded)	Oznaczenie ogólne procesu z widocznymi szczegółami. Wewnątrz obramowania będą umieszczane oznaczenia czynności i innych elementów potrzebnych do opisania przebiegu podprocesu.

- Przykład procesu z podprocesami (odprawa z punktu widzenia obsługi):



- Typy podprocesów ze względu na lokalizację w procesie głównym:
 - Osadzony (Embedded Sub-Process) – zwykły podproces stanowiący część integralną normalnego przepływu sterowania w procesie nadziednym.
 - Przywoływany (Call Activity) – wywołanie innego, wcześniej zdefiniowanego procesu (w BPMN 1.2 był określany jako Reusable Sub-Process), zob. też *Użycie predefiniowanego procesu lub zadania (Call Activity)*, s. 56.

Symbol	Nazwa	Opis
	Embedded Sub-Process (osadzony)	Zwykły podproces.
	Call Activity (przywoływanie)	Wywołanie predefiniowanego (pod)procesu (obramowanie grubą linią).

Uściślenia przebiegu procesu

Zdarzenia pośrednie wpływające na proces (Intermediate Events)

- W trakcie trwania procesu występują różne zdarzenia, które mają wpływ na jego przebieg. Są one zwane zdarzeniami pośrednimi (Intermediate Events).
- Odpowiadają one następującym sytuacjom:
 - oczekiwaniu na wiadomość lub wysłaniu wiadomości w czasie trwania procesu,
 - oczekiwaniem/zaplanowanym opóźnieniom/przerwom w działaniu,
 - zakłóceniom normalnego przebiegu procesu (obsługa wyjątków/błędów procesu),
 - zaistnieniu potrzeby wykonania dodatkowych działań w celu kompensacji czynności lub podprocesów.
- Zdarzenia pośrednie mogą być:
 - Częścią normalnego (podstawowego) przepływu sterowania w procesie – oznaczane jako osobne elementy procesu (na równi z czynnościami).
 - Włączone/dołączone do czynności („Intermediate Events attached to an activity boundary”, określane zwykle jako Boundary Intermediate Events) – umieszczane na krawędzi symbolu czynności.

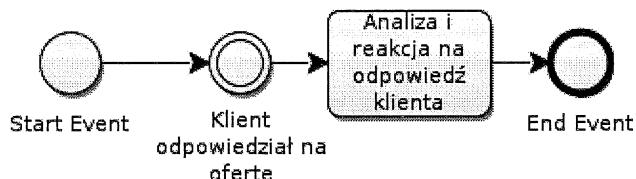
Typy zdarzeń pośrednich

- Zdarzenia pośrednie mogą obrazować jedną z dwóch sytuacji (oznaczają trigger'y/wyzwalacze typu):
 - **catch** ('chwycić') – oczekивание на przyczynę sprawczą oraz jej odebranie/przechwycenie/zaistnienie, co ma spowodować wykonanie kolejnej czynności czy uruchomić podproces (np. odebranie wiadomości),
 - **throw** ('rzuć') – wynik działania czynności, czyli udostępnienie lub wysłanie czegoś, co będzie przyczyną sprawczą dalszych działań w procesie i będzie miało wpływ na dalszy przebieg procesu lub na inne procesy (np. wysłanie wiadomości).
- Jeśli zdarzenie jest typu:
 - **catch** – token oczekuje na zadziałanie przyczyny/wyzwalacza (np. odebranie wiadomości), a po jego zadziałaniu przechodzi przepływem sterowania do kolejnego etapu realizacji procesu,
 - **throw** – wyzwalacz działa natychmiast (np. zostaje wysłana wiadomość), zaś token przechodzi przepływem sterowania do kolejnego etapu.
- Zdarzenia poszczególnych typów są oznaczane:
 - **catch** – konturem niewypełnionym,

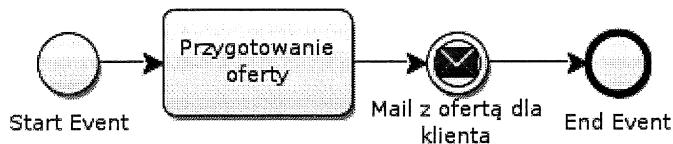
- `throw` – konturem wypełnionym (zwykle zaczernionym wewnątrz, ale zależnie od programu używanego do modelowania może to być też inny kolor).

Zdarzenia w głównym przebiegu procesu

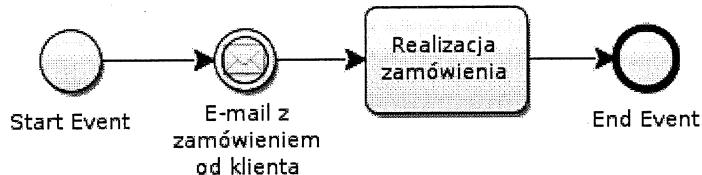
- Nieokreślone (`None`):
 - Występuje tylko na poziomie podstawowego przepływu sterowania w procesie (jako rutynowy element procesu) – nie może być używane jako włączone do czynności (`boundary/attached`).
 - Występuje tylko jako `throw` – przyczyna sprawcza kolejnej czynności w procesie.
 - Może służyć do oznaczania zdarzeń bez uściślenia przyczyny lub pewnych zmian stanów w procesie.
 - Np. klient odpowiedział na ofertę, a w efekcie – zależnie od reakcji (akceptacji, odrzucenia czy też uwag do niektórych punktów/warunków) – będą podejmowane dalsze działania (być może też bliżej nieokreślone, jeśli opisywany proces nie wymaga takiej dokładności).



- Wiadomość (`Message`):
 - Jako `throw` oznacza wysłanie wiadomości, a jako `catch` – oczekiwanie na wiadomość i jej odebranie.
 - Typ `catch` oznacza dodatkowo, że po przybyciu tokenu do takiej wiadomości następuje podjęcie dalszych działań w procesie (oczekiwana wiadomość nadeszła, można kontynuować), zaś jeśli token nie przybywa (a powinien), czyli oczekiwanie na wiadomość jest bezskuteczne, przepływ zmienia się na exception handling (konieczna jest obsługa błędu procesu).
 - Jeśli diagram pokazuje współpracę różnych uczestników procesu (`Collaboration`), uczestnik, od którego wiadomość przychodzi, jest połączony za pomocą `Message Flow`.
 - Np. wysłanie do klienta e-mail-a z ofertą (typ `throw`).

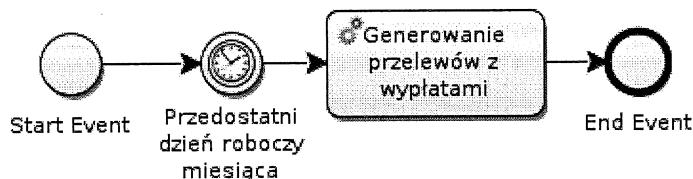


- Np. odebranie e-mail-a z zamówieniem klienta (typ `catch`).



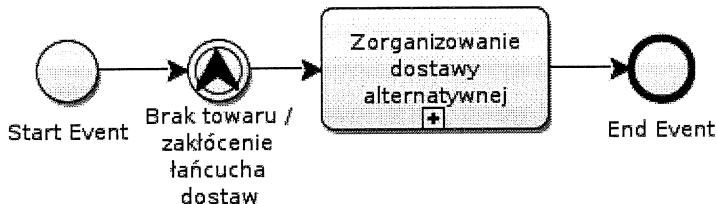
- Czas (Timer):

- Oznacza opóźnienie/oczekiwanie przez określony czas lub też cykliczne wykonywanie czynności, przed którą zdarzenie czasowe zostało umieszczone.
- Np. standardowy czas oczekiwania klienta od momentu zgłoszenia usterki do momentu realizacji to 1 dzień (raczej maksymalnie niż dokładnie, ale będzie to zależeć od kontekstu), bądź też przygotowanie/generacja przelewów z wypłatami w przedostatni dzień roboczy miesiąca (czynność cykliczna).



- Występuje tylko jako catch (na bieg czasu nie mamy wpływu).
- Nadzwyczajna interwencja (Escalation):

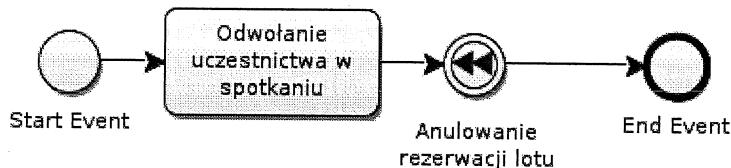
- Na podstawowym poziomie przebiegu procesu powoduje uruchomienie czynności związanej z nadzwyczajną interwencją „obiektu spoza procesu”.
- Np. na półce w hipermarketie zabrakło jakiegoś często kupowanego towaru (zakłócenie działania łańcucha dostaw). Potrzebna jest zatem interwencja osoby odpowiedzialnej za zaopatrzenie i/lub menedżera w celu jak najszybszego udrożnienia kanału dostaw i/lub zorganizowania jak najszybciej dostawy alternatywnej.



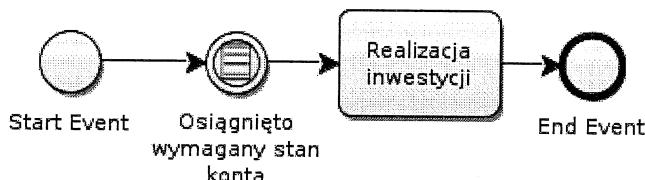
- Na poziomie podstawowym procesu występuje tylko jako throw.

- Kompensacja (Compensation):

- W normalnym przebiegu procesu oznacza, że konieczne jest dokonanie kompensacji (wycofania określonych wykonanych, poprawnie zakończonych działań i/lub ich efektów).
- Np. odwołanie uczestnictwa w wyjazdowym spotkaniu biznesowym powoduje potrzebę anulowania rezerwacji przelotu na miejsce spotkania.



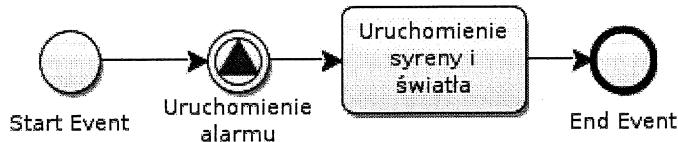
- W normalnym przebiegu występuje tylko jako **throw**.
- Jeśli ze zdarzeniem kompensacji jest powiązana konkretna czynność, która została poprawnie wykonana/zakończona, wówczas zostanie skompensowana. Czynność taka musi być widoczna bezpośrednio ze zdarzenia kompensacji (nie wolno wskazywać do kompensacji czynności znajdujących się na innym poziomie realizacji procesu, czyli np. stanowiących część podprocesu, jeśli zdarzenie występuje na poziomie nadziedznym).
- W przypadku kompensacji konkretnej czynności, czynność taka musi zarazem mieć włączone zdarzenie kompensacji (Boundary Compensation Event) i musi posiadać zadanie kompensacyjne (Compensation Task) lub podproces kompensacyjny (Compensation Event Sub-Process).
- Jeśli zdarzenie kompensacji nie wskazuje na żadną czynność, wszystkie zakończone poprawnie dotychczas czynności, które są zarazem widoczne bezpośrednio ze zdarzenia, zostaną skompensowane. „Widoczne” oznacza tu, że znajdują się na tym samym poziomie realizacji procesu, co wywołujące kompensację zdarzenie.
- Warunkowe (Conditional):
 - Jest uruchamiane, jeśli zostanie spełniony określony warunek (wyrażenie dające możliwe wartości „prawda” lub „fałsz”).
 - Np. osiągnięto stan konta umożliwiający realizację inwestycji.



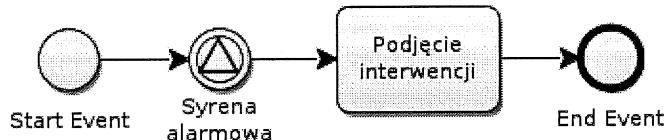
- Na poziomie podstawowym jest tylko typu **catch** (odgrywa rolę „czujnika” osiągnięcia pożądanego stanu).
- Połączenie (Link):
 - Element pomocniczy, który może wystąpić tylko na poziomie podstawowym procesu.
 - **Nie może być użyte** jako włączone do czynności (czyli jako Boundary Intermediate Event).
 - Nie może łączyć np. procesu nadziedznego z podprocesem.
 - Służy do pokazania połączenia między dwoma częściami procesu.
 - Może być użyte do opisania pętli (przeskok z czynności kończącej do czynności rozpoczęjącej pętlę – aby uniknąć prowadzenia długich przepływu sterowania), do

„przeskoków” między częściami procesu (cel – jw.) oraz do łączenia kolejnych drukowanych stron w opisach bardzo rozbudowanych procesów (Off-Page Connectors).

- Dopuszczalne jest wystąpienie wielu linków początkowych (typu throw), zmierzających do „wspólnego celu”, ale w takiej sytuacji musi być tylko jeden ścisłe określony link końcowy (typu catch).
- Jedno konkretne zdarzenie Link nie może jednocześnie otrzymywać i rozpoczynać przepływu sterowania.
- Sygnał (Signal):
 - Oznacza wysłanie lub odebranie sygnału.
 - Jest używany do pokazania komunikacji między poziomami procesu, basenami (Pools – np. współpracującymi w ramach procesu oddziałami czy firmami) oraz procesami (różnymi diagramami).
 - Sygnał ma określonego nadawcę, ale nie ma określonych odbiorców (jak np. syrena alarmowa czy flara sygnalizacyjna).
 - Np. dla typu throw – uruchomienie przez system alarmowy syreny połączonej z migającym światłem ostrzegawczym.

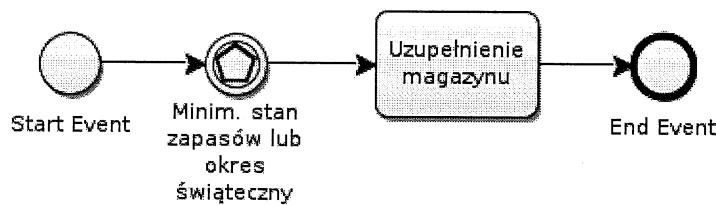


- Np. dla typu catch – dostrzeżenie sygnału alarmowego przez ochronę obiektu (czego efektem będzie podjęcie interwencji).

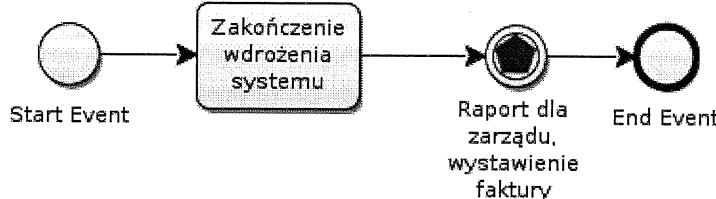


- Na poziomie podstawowym procesu można wysyłać i odbierać sygnały, ale np. czynności mogą sygnały tylko odbierać (jeśli mają takie zdarzenie włączone w swój obręb).
- Zdarzenia wysłania/odebrania sygnału mogą wywoływać przerwanie czynności, jednakże w takiej sytuacji różnią się od zdarzeń oznaczających występowanie błędów, ponieważ oznaczają bardziej ogólne i niezwiązane z błędami przyczyny takiego przerwania (np. zakończenie wykonania innej czynności). Mają też szersze zastosowanie niż oznaczanie błędów.
- Wielokrotne (Multiple):

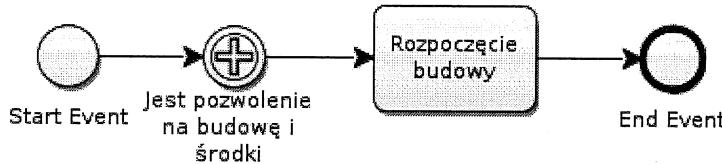
- Ze zdarzeniem związanym jest wiele wyzwalaczy, czyli przyczyn (dla typu `catch`) lub efektów działania (dla typu `throw`).
- Typu `catch` oznacza, że konieczna jest tylko jedna z określonych przyczyn, by zdarzenie nastąpiło.
- Np. system magazynowy zgłosił osiągnięcie minimalnego stanu zapasów towarów albo zbliża się okres świąteczny wraz ze wzmożonym zapotrzebowaniem na określone towary (w efekcie zostanie uruchomione zamówienie/dostarczenie towarów do magazynu).



- Typu `throw` oznacza, że pojawiają się wszystkie efekty czynności, z której zdarzenie zostało wyprowadzone.
- Np. zakończyło się wdrożenie systemu, zatem przygotowany został raport dla zarządu firmy, a księgowość wystawia fakturę.



- Wielokrotne równoległe (Parallel Multiple):
 - Oznacza, że aby zdarzenie zaistniało, muszą wystąpić wszystkie określone jego przyczyny jednocześnie.
 - Np. uzyskano pozwolenie na budowę oraz zgromadzono potrzebne środki i fachowców (zatem można rozpoczęć inwestycję).



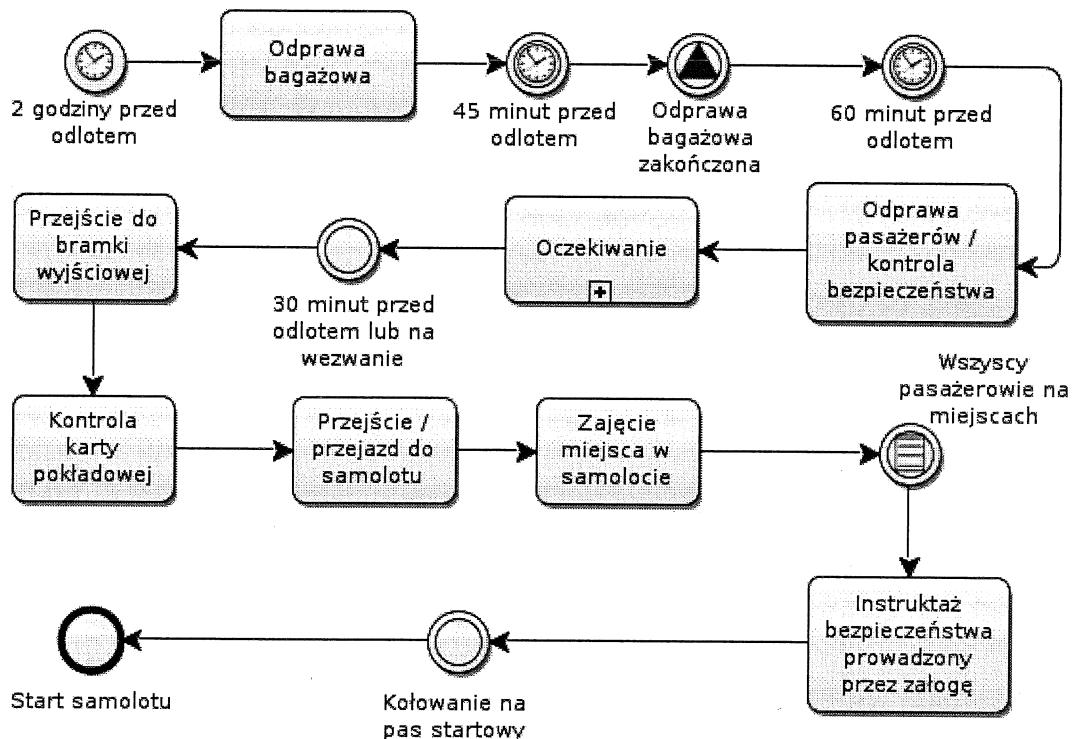
- Na poziomie podstawowym procesu występuje tylko jako `catch`.
- Połączenia zdarzeń – zdarzenie pośrednie znajdujące się w głównym nurcie procesu (z wyjątkiem zdarzenia Link):

- Musi otrzymywać chociaż jeden przepływ wejściowy (jest ścisłe związane z przebiegiem/realizacją procesu).
- Może otrzymywać więcej niż jeden przepływ wejściowy.
- **Uwaga:** w BPMN 1.2 niektóre zdarzenia pośrednie nie mogły otrzymywać przepływów wejściowych w ogóle.
- Musi rozpoczynać choć jeden przepływ wyjściowy (jest przyczyną kolejnych działań w procesie).
- Jeśli rozpoczyna więcej niż jeden przepływ wyjściowy, traktowane są one jako współbieżne/równoległe.

Symbol	Wyzwalacz / przyczyna	Opis
	None (nieokreślona)	Blizej nieokreślone zdarzenie mające wpływ na przebieg procesu (występuje tylko jako throw). Np. klient odpowiedział na ofertę, a w efekcie – zależnie od kategorii tej reakcji (akceptacji, odrzucenia czy też uwagi do niektórych punktów/warunków) – będą podejmowane dalsze działania (być może też bliżej nieokreślone, bo opisywany proces nie wymaga takiej dokładności).
	Message - throw (wiadomość – wysłanie)	Wysłanie wiadomości do innego uczestnika procesu. Np. wysłanie do klienta e-mail-a z ofertą.
	Message - catch (wiadomość – odebranie)	Odebranie wiadomości od innego uczestnika procesu. Np. odebranie e-mail-a z ofertą przez klienta (po czym następuje przeanalizowanie treści oferty).
	Timer (czas)	Osiagnięcie określonego punktu w czasie (po określonym okresie oczekiwania lub też konkretnej daty czy godziny) lub początku kolejnego okresu cyklicznego. Np. godzina 5:00 (kiedy następuje zaplanowany wyjazd autobusu miejskiego na linię).
	Escalation (nadzwyczajna interwencja)	Potrzeba uruchomienia czynności związanej z nadzwyczajną interwencją „obiektu z zewnątrz procesu”. Np. na półce zabrakło towaru i potrzebna jest interwencja osoby odpowiedzialnej za zaopatrzenie i/lub menedżera sklepu.
	Compensation (kompensacja)	Potrzeba kompensacji czynności (wycofania określonych wykonanych już działań). Np. odwołanie uczestnictwa w wyjazdowym spotkaniu biznesowym powoduje potrzebę anulowania rezerwacji przelotu na miejsce spotkania.
	Conditional (warunkowe)	Spełniony został przewidziany warunek. Np. osiągnięto stan konta umożliwiający realizację inwestycji.

Symbol	Wyzwalacz / przyczyna	Opis
	Link - throw (połączenie – początek)	Początek, punkt wyjściowy połączenia (np. na pierwszej stronie większego diagramu).
	Link - catch (połączenie – koniec)	Koniec, punkt docelowy połączenia (np. na drugiej stronie większego diagramu).
	Signal - throw (sygnał – wysłanie)	Wysłanie sygnału, który może być odebrany przez innych uczestników procesu. Np. uruchomienie przez system alarmowy syreny połączonej z migającym światłem ostrzegawczym.
	Signal - catch (sygnał – odebranie)	Odebranie sygnału wysłanego przez innego uczestnika procesu lub inny proces, który to sygnał spowoduje wykonanie określonej czynności. Np. dostrzeżenie sygnału alarmowego przez ochronę obiektu (czego efektem będzie podjęcie interwencji).
	Multiple - throw (wielokrotne – efekt/wynik)	Pojawiły się określone (ale wszystkie) efekty czynności, która poprzedza zdarzenie. Np. zakończyło się wdrożenie systemu, zatem przygotowany został raport dla zarządu firmy, a księgowość wystawia fakturę.
	Multiple - catch (wielokrotne – uruchamiające)	Pojawiła się jedna z określonych przyczyn uruchamiających działanie. Np. system magazynowy zgłosił zbliżanie się do minimalnego stanu zapasów towarów albo zbliża się okres świąteczny wraz ze wzmożonym zapotrzebowaniem na określone towary (w efekcie zostanie uruchomione zamówienie/dostarczenie towarów do magazynu).
	Parallel Multiple (wielokrotne równolegle)	Wystąpiły wszystkie przyczyny konieczne do wykonania czynności. Np. uzyskano pozwolenie na budowę oraz zgromadzono potrzebne środki (można rozpoczęć inwestycję).

- Przykład procesu z uwzględnieniem zdarzeń pośrednich w głównym przebiegu:



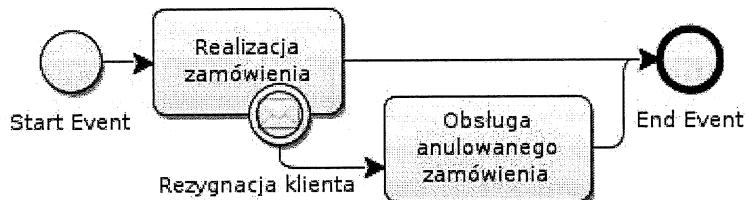
Typy zdarzeń pośrednich włączanych do czynności

- Czyli Intermediate boundary Events.
- Określenie „boundary” = ‘włączane, dołączane’ (Boundary Intermediate Events), można rozumieć też jako ‘w obrębie, w ramach, wewnętrz (czynności)’.
- Są wyłącznie typu catch – token oczekuje na wystąpienie przyczyny, a następnie przechodzi do „wnętrza” czynności (co oznacza, że „przechwycona” przyczyna ma wpływ na realizację tej czynności).
- Wpływ na czynność zawierającą takie zdarzenie może być dwójaki:
 - Interrupting – czynność wymaga przerwania, natomiast w konsekwencji trzeba wykonać jakąś czynność alternatywną albo zakończyć proces.
 - Np. klient kupuje tylko jeden towar, ale przy kasie stwierdza, że nie dokona zakupu towaru (bo cena w systemie jest niezgodna z ceną na półce, a kasjer twierdzi, że nie może naliczyć innej ceny). Wówczas czynność naliczania należności zostaje przerwana, a towar trzeba odnieść z powrotem na półkę (choć niekoniecznie od razu).
 - Non-Interrupting – czynność nie wymaga przerwania, natomiast trzeba wykonać jakąś czynność dodatkową.
 - Np. rozbicie szklanki przez kelnera nie powoduje przerwania obsługi klienta, a jedynie konieczność uprzątnięcia rozbitego szkła i przyniesienia drugiej szklanki.
- Zdarzenie Interrupting:

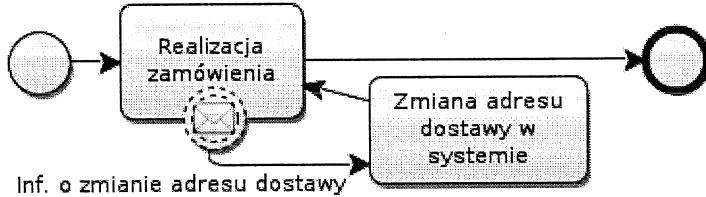
- Jeśli zaistnieje, wówczas czynność, z którą jest ono związane, zostaje natychmiast przerwana.
- Zostaje też w tym momencie wygenerowany token, który uaktywnia następny element procesu (połączony za pomocą nieuwarunkowanego przepływu sterowania określonego jako Exception Flow).
- Zdarzenie Non-Interrupting:
 - Jeśli zaistnieje, wówczas czynność jest kontynuowana (np. przy okazji czynności nadzędnej jest wykonywana czynność dodatkowa).
 - Ponieważ jednocześnie generowany jest również token, który porusza się przepływem sterowania (współbieżnie z czynnością, z której wyszedł), trzeba zachować ostrożność – zwykle powraca on do czynności „głównej”.
 - Np. podczas obsługi klienta składającego w okienku kasowym banku polecenie przelewu (czynność nadzędna), kasjer aktualizuje dane klienta (czynność dodatkowa, która nie przerywa czynności nadzędnej).

Zdarzenia włączane do czynności

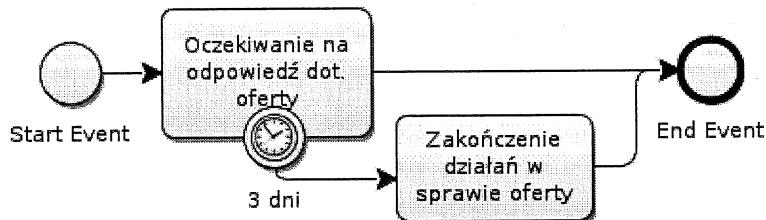
- Do jednej czynności można włączyć więcej niż jedno zdarzenie pośrednie.
- Wiadomość (Message):
 - Typu Interrupting oznacza, że odebrana wiadomość wymaga przerwania czynności i przejścia procesu na ścieżkę obsługi wyjątku przewidzianego dla takiej sytuacji.
 - Np. klient powiadamia firmę o rezygnacji ze złożonego wcześniej zamówienia (zatem realizacja zamówienia zostaje przerwana), natomiast firma rozpakowuje towary przygotowane do wysyłki i ew. obciąża klienta przewidzianymi w regulaminie kosztami manipulacyjnymi.



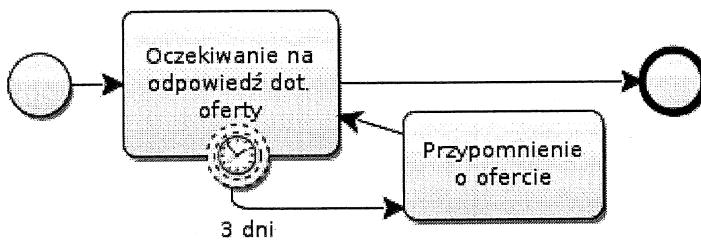
- Typu Non-Interrupting oznacza, że czynność jest kontynuowana, natomiast pojawia się konieczność wykonania dodatkowej czynności (związaną z odebraną wiadomością).
- Np. klient poinformował o zmianie adresu dostarczenia zamówionych towarów (wiadomość). Zatem realizacja zamówienia (czynność, do której trafiła wiadomość na diagramie), trwa nadal. Pojawia się natomiast dodatkowa czynność obejmująca zmianę adresu dostawy w systemie i/lub dokumentach przewozowych przeznaczonych dla kuriera.



- Jeśli diagram pokazuje współpracę uczestników procesu (Collaboration), wówczas zdarzenie jest połączone z nadawcą wiadomości za pomocą Message Flow.
- Czas (Timer):
 - Określona data/czas, okres czasu lub początek okresu cyklicznego.
 - Typu Interrupting oznacza, że w podanym momencie czynność zostanie przerwana i proces przejdzie na ścieżkę obsługi wyjątku przewidzianego dla takiej sytuacji.
 - Np. bezskuteczne oczekiwanie przez 3 dni na odpowiedź dotyczącą oferty oznacza (zgodnie z poczynionymi z klientem ustaleniami) rezygnację klienta, a zatem automatyczne przerwanie/zakończenie działań firmy związanych z tą ofertą.

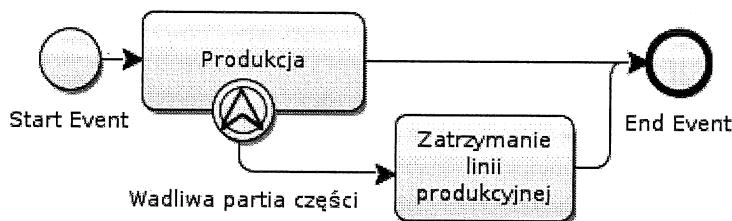


- Typu Non-Interrupting oznacza, że czynność jest kontynuowana, natomiast pojawia się konieczność wykonania dodatkowej czynności (związanej z osiągnięciem określonego momentu w czasie).
- Np. bezskuteczne oczekiwanie przez 3 dni na odpowiedź dotyczącą oferty oznacza konieczność podjęcia odpowiednich kroków (np. sprawdzenia, czy klient zapoznał się z ofertą lub przypomnienia klientowi o ofercie dodatkowym mailem bądź telefonicznie).

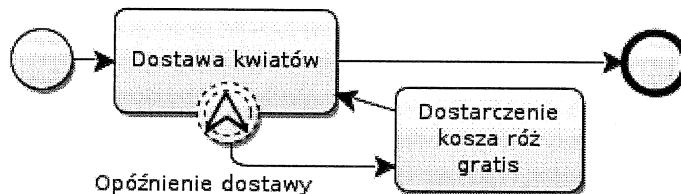


- Nadzwyczajna interwencja (Escalation):
 - Używane do obsługi eskalacji – sytuacji wymagającej podjęcia nadzwyczajnej interwencji.

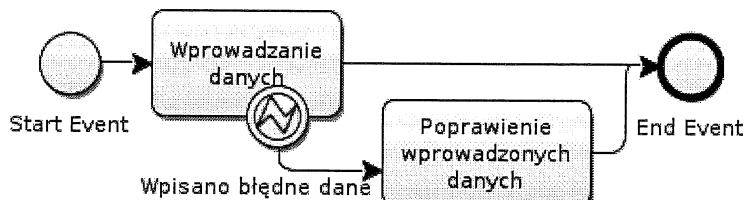
- Zasadniczo nie jest przewidziane do przerywania czynności, do której jest włączone (nadzwyczajna interwencja raczej oznacza wykonanie jakiegoś działania, tyle tylko, że przez osobę lub obiekt, który rutynowo nie bierze udziału w realizacji procesu). Jednak decyzja o wystąpieniu tego zdarzenia jako Non-Interrupting lub Interrupting zależy od twórcy modelu.
- Np. dla typu Interrupting – główny inżynier decyduje o zatrzymaniu linii produkcyjnej (ze względu na wadliwą partię części dostarczonych przez kontrahenta).



- Np. dla typu Non-Interrupting – kierownik kwaciarni wysyłkowej podejmuje decyzję o dostarczeniu klientowi gratis dodatkowego kosza róż (jako rekompensatę za drobne opóźnienie w dostawie zamówionych przez tego klienta bukietów do dekoracji sali bankietowej).



- **Błąd (Error):**
 - Może być tylko włączone do czynności (nie może być używane w normalnym przebiegu procesu).
 - Oznacza obsługę błędu określonego w nazwie zdarzenia lub dowolnego błędu (jeśli błąd nie został określony).
 - Np. błąd przy wprowadzaniu danych do systemu lub błąd montażu maszyny.

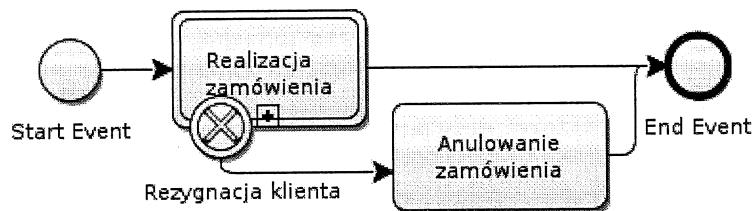


- Występuje tylko jako Interrupting – zawsze przerywa czynność, do której jest włączone, jako że na błędy trzeba jak najszybciej zareagować i je (albo ich efekty)

niezwłocznie poprawić (w innym przypadku cały proces będzie dawał niepoprawne efekty, czyli nie osiągnie zamierzonego celu).

- Anulowanie (Cancel):

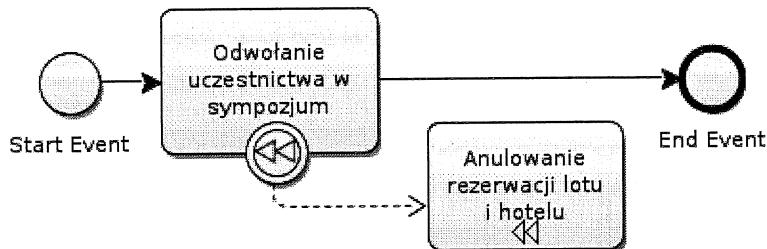
- Jest używane tylko jako włączane do podprocesu transakcyjnego (Transaction Sub-Process). Nie może być użyte w normalnym przebiegu procesu.
- Musi zostać uaktywnione wówczas, gdy w procesie transakcyjnym zostanie osiągnięte zdarzenie kończące typu anuluj (Cancel End Event), ale może też być uaktywnione, gdy w czasie wykonywania transakcji do protokołu kontrolującego jej przebieg (transaction protocol) dotrze wiadomość „anuluj”.
- Np. klient rezygnuje ze złożonego wcześniej zamówienia (co oznacza przede wszystkim, że realizacja zamówienia zostaje przerwana, przygotowane do wysyłki towary wracają na półki magazynowe itd.).



- Zawsze przerywa podproces, do którego jest włączone.

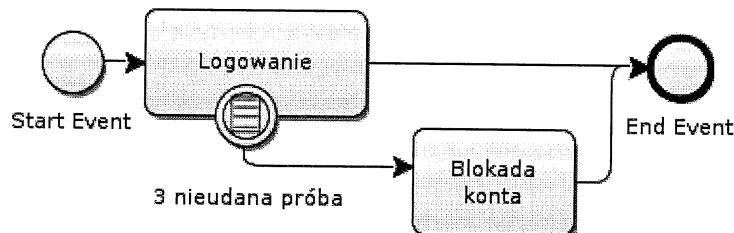
- Kompensacja (Compensation):

- Oznacza konieczność przeprowadzenia kompensacji dla czynności, która zawiera to zdarzenie.
- Czynność wymagająca kompensacji musi zostać zakończona, by kompensacja była możliwa.
- Np. anulowanie rezerwacji biletu lotniczego oraz noclegu w hotelu (jako konsekwencja odwołania uczestnictwa pracownika naukowego w sympozjum).

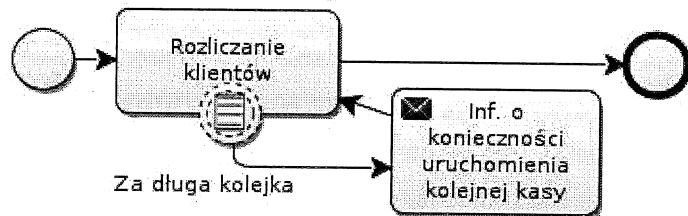


- **Uwaga:** czynność obsługująca zdarzenie kompensacji jest połączona asocjacją (Association), a nie przepływem sterowania.
- Warunkowe (Conditional):
 - Następuje, gdy zostanie spełniony warunek związany z tym zdarzeniem.

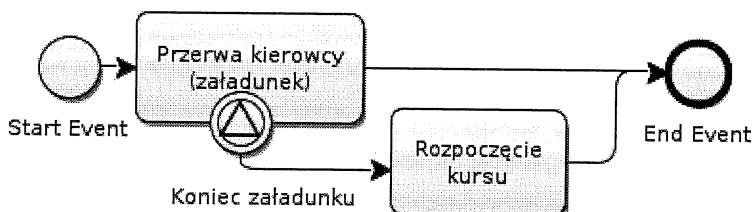
- Typu Interrupting oznacza, że czynność zostaje przerwana, zaś przepływ normalny zmienia się w przepływie obsługi wyjątku.
- Np. trzecie niepoprawne logowanie do systemu powoduje przerwanie procesu uzyskiwania dostępu do systemu (przez zablokowanie sesji użytkownika).



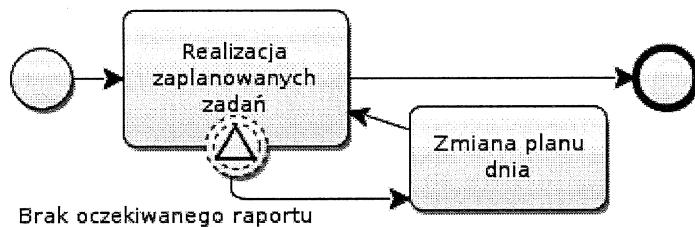
- Typu Non-Interrupting oznacza, że czynność jest kontynuowana, natomiast pojawia się konieczność wykonania dodatkowej czynności (związanej ze spełnieniem podanego warunku).
- Np. kasjer w hipermarketie widzi, że kolejka przy jego kasie szybko się wydłuża, zatem (nie przerywając obsługi klientów) daje znać kierownikowi o konieczności uruchomienia dodatkowej kaszy.



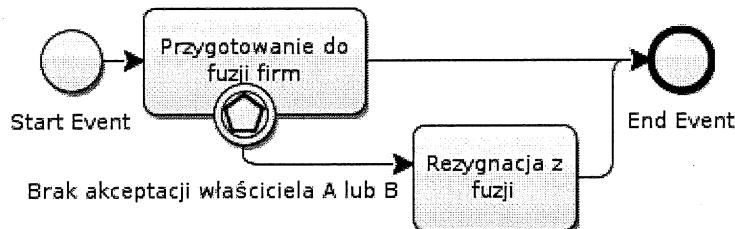
- Sygnał (Signal):
 - Odebranie sygnału.
 - Typu Interrupting oznacza, że czynność ulega przerwaniu, zaś przepływ normalny zmienia się w przepływie obsługi wyjątku.
 - Od zdarzenia Error różni się tym, że przyczyna przerwania może być bardziej ogólna – nie oznacza błędu (tylko np. zakończenie innej czynności).
 - Np. sygnał o zakończeniu załadunku ciężarówki oznacza dla kierowcy koniec przerwy (trwającej przez czas załadunku).



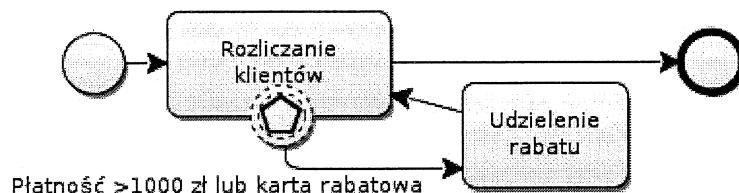
- Typu Non-Interrupting oznacza, że czynność jest kontynuowana, natomiast pojawia się konieczność wykonania dodatkowej czynności (związanej z odebranym sygnałem).
- Np. kierownik realizuje zaplanowane na dany dzień zadania. Jednym z nich miała być analiza raportu. Raport jednak nie zostaje dostarczony. Kierownik zatem nadal realizuje plan dnia, ale dodatkowo modyfikuje go, przenosząc analizę raportu na kolejny dzień.



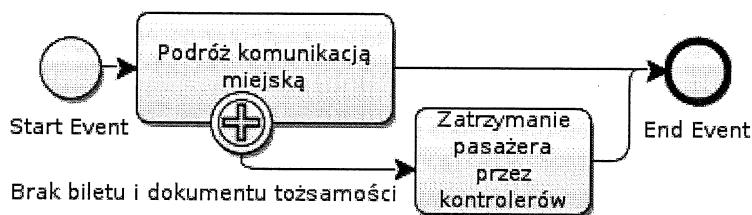
- Wielokrotne (Multiple):
 - Oznacza, że do uruchomienia zdarzenia jest wymagane wystąpienie tylko jednej przyczyny spośród określonych.
 - Typu Interrupting oznacza, że czynność zostaje przerwana, zaś przepływ normalny zmienia się w przepływ obsługi wyjątku.
 - Np. brak akceptacji przez jednego ze współwłaścicieli firmy powoduje przerwanie przygotowania fuzji z inną firmą.



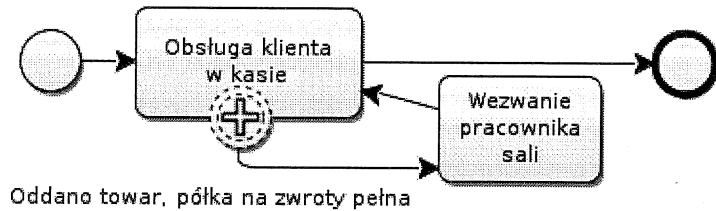
- Typu Non-Interrupting oznacza, że czynność jest kontynuowana, natomiast pojawia się konieczność wykonania dodatkowej czynności (związanej z wystąpieniem jednej z podanych przyczyn).
- Np. klientowi zostaje udzielony rabat, ponieważ dokonał zakupów za 1000 zł (inną przyczyną udzielenia rabatu jest posiadanie karty rabatowej).



- Wielokrotne równoległe (Parallel Multiple):
 - Oznacza, że do uruchomienia zdarzenia są wymagane wszystkie z podanych przyczyn jednocześnie.
 - Typu Interrupting oznacza, że czynność zostaje przerwana, zaś przepływ normalny zmienia się w przepływ obsługi wyjątku.
 - Np. brak ważnego biletu i zarazem brak dokumentu tożsamości powoduje przerwanie przez kontrolerów dalszej podróży pasażera tramwaju.



- Typu Non-Interrupting oznacza, że wówczas czynność jest kontynuowana, natomiast pojawia się konieczność wykonania dodatkowej czynności (związanej z wystąpieniem wszystkich podanych przyczyn jednocześnie).
 - Np. klient przy kasie rezygnuje z zakupu jednego z towarów, zaś półka przeznaczona przy kasie na odkładane towary zostaje zapełniona. Wówczas kasjer (w trakcie obsługiwania tego klienta) wzywa osobę odpowiedzialną za odłożenie towarów na półki.

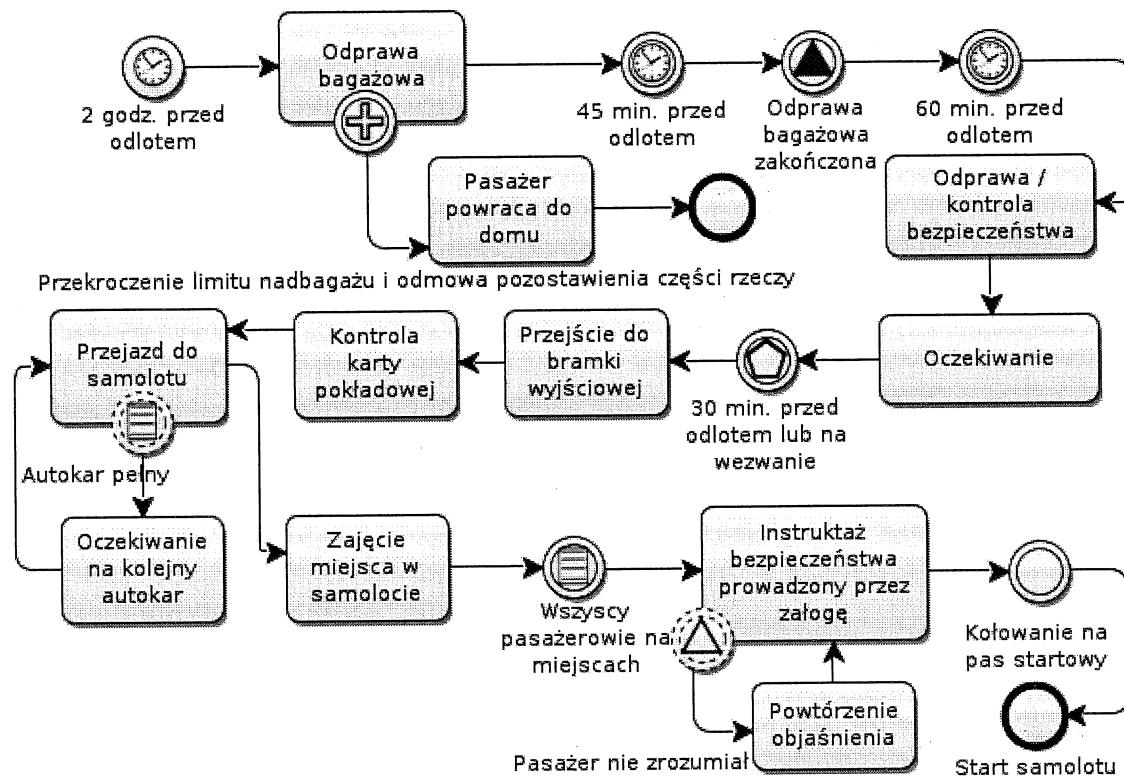


- Połączenia zdarzeń – zdarzenie pośrednie włączone do czynności:
 - Nie może otrzymywać przepływów wejściowych – jego „przyczyny” są ściśle związane z realizacją czynności, której jest częścią (przyczyny takiego zdarzenia tkwią w czynności).
 - Musi rozpoczynać chociaż jeden przepływ wyjściowy (może rozpoczynać więcej – w takiej sytuacji są one traktowane jako współbieżne/równoległe). Jest bowiem mniej lub bardziej bezpośrednią przyczyną kolejnych działań w procesie.
 - Wyjątek stanowi zdarzenie Compensation, które nie może rozpoczynać przepływów wyjściowych (natomiast może rozpoczynać połączenie Association).

Symbol	Wyzwalać / przyczyna	Opis
	Message – Interrupting (wiadomość – przerywające)	Odebrana wiadomość wymaga przerwania czynności i przejścia procesu na ścieżkę obsługi wyjątku przewidzianego dla takiej sytuacji.
	Message – Non-Interrupting (wiadomość – nieprerywające)	Czynność jest kontynuowana, ale trzeba oprócz niej wykonać czynność dodatkową.
	Timer – Interrupting (czas – przerywające)	W podanym momencie czynność zostanie przerwana.
	Timer – Non-Interrupting (czas – nieprerywające)	Czynność jest kontynuowana, ale trzeba oprócz niej wykonać czynność dodatkową.
	Escalation – Non-Interrupting (nadzwyczajna interwencja – nieprerywające)	Obsługa eskalacji – podjęcie nadzwyczajnej interwencji.
	Escalation – Interrupting (nadzwyczajna interwencja – przerywające)	Obsługa eskalacji – nadzwyczajna interwencja przerywa wykonanie czynności podlegającej eskalacji.
	Error (błąd)	Obsługa błędu określonego w nazwie zdarzenia lub dowolnego błędu (jeśli błąd nie został określony).
	Cancel (anulowanie)	Tylko w podprocesach transakcyjnych (Transaction Sub-Processes). Anulowanie procesu transakcyjnego i wykonanie działań przewidzianych w protokole transakcyjnym (Transaction Protocol).
	Compensation (kompensacja)	Konieczność przeprowadzenia kompensacji dla czynności, która zawiera to zdarzenie.
	Conditional – Interrupting (warunkowe – przerywające)	Spełnienie warunku związanego z tym zdarzeniem przerywa czynność.

Symbol	Wyzwalacz / przyczyna	Opis
	Conditional - Non-Interrupting (warunkowe – nieprzerwywające)	Czynność jest kontynuowana, ale trzeba oprócz niej wykonać czynność dodatkową (związaną z określonym warunkiem).
	Signal - Interrupting (sygnał – przerwywające)	Odebranie sygnału przerwywa czynność.
	Signal - Non-Interrupting (sygnał – nieprzerwywające)	Czynność jest kontynuowana, ale trzeba oprócz niej wykonać czynność dodatkową (związaną z odebranym sygnałem).
	Multiple - Interrupting (wielokrotne – przerwywające)	Po wystąpieniu jednej z określonych przyczyn czynność zostaje przerwana.
	Multiple - Non-Interrupting (wielokrotne – nieprzerwywające)	Czynność jest kontynuowana, ale trzeba oprócz niej wykonać czynność dodatkową (związaną z wystąpieniem jednej z określonych przyczyn).
	Parallel Multiple - Interrupting (wielokrotne równolegle – przerwywające)	Wystąpienie wszystkich podanych przyczyn przerwywa wykonywanie czynności.
	Parallel Multiple - Non-Interrupting (wielokrotne równolegle – nieprzerwywające)	Czynność jest kontynuowana, ale trzeba oprócz niej wykonać czynność dodatkową (związaną z wystąpieniem wszystkich podanych przyczyn).

- Przykład procesu z uwzględnieniem Boundary Intermediate Events:



Zróżnicowane ścieżki działania w procesie – bramki (Gateways)

- Proces może przebiegać różnymi ścieżkami, które w określonych sytuacjach rozdzielają się i łączą.
- Ścieżki te mogą być:
 - alternatywne – token może zostać przepuszczony jednym z przepływów wyjściowych,
 - równoległe (przepływ równoczesny/współbieżny) – token zostaje podzielony na kilka przebiegających równocześnie osobnymi przepływaniami wyjściowymi.
- Obrazowanie tych sytuacji umożliwiają bramki (Gateways).
- Rodzaje bramek:
 - wykluczająca – Exclusive Gateway,
 - włączająca – Inclusive Gateway,
 - równoległa – Parallel Gateway,
 - złożona – Complex Gateway,
 - zależna od zdarzenia – Event-Based Gateway,
 - zależna od zdarzenia wykluczająca (rozpoczynająca proces) – Exclusive Event-Based Gateway,
 - zależna od zdarzenia równoległa (rozpoczynająca proces) – Parallel Event-Based Gateway).

Symbol	Nazwa	Opis
 Label	Exclusive Gateway (wykluczająca ze znacznikiem)	Wyjście jednym i tylko jednym przepływem. Symbol może też łączyć ścieżki sterowania rozgałęzione wcześniej bramką wykluczającą. Warunki oparte na danych dostępnych bezpośrednio wykonawcy czynności.
 Label	Exclusive Gateway (wykluczająca bez znacznika)	Wyjście jednym i tylko jednym przepływem (druga możliwość oznaczania). Symbol może też łączyć ścieżki sterowania rozgałęzione wcześniej bramką wykluczającą. Warunki oparte na danych dostępnych bezpośrednio wykonawcy czynności.
 Label	Inclusive Gateway (niewykluczająca/włączająca)	Wyjście jednym lub więcej przepływem. Symbol może też łączyć ścieżki sterowania rozgałęzione wcześniej bramką niewykluczającą lub łączyć ścieżki sterowania alternatywne ze współbieżnymi.
 Label	Parallel Gateway (równoległa)	Początek lub koniec przepływu współbieżnego.
 Label	Complex Gateway (złożona)	Główne synchronizacja (łączenie) przepływów, gdy potrzebne jest sprawdzanie różnych zestawów warunków składowych dla przepływów wejściowych. Może też łączyć przepływy rozpoczynające się bramką równoległą (gdy niezbędne jest wykonanie kilku różnych czynności, ale przejście do czynności kolejnych będzie uzależnione od efektów końcowych tylko niektórych z nich).
 Label	Event-Based Gateway (zależna od zdarzenia)	Wyjście przepływem uzależnionym od zdarzenia (np. otrzymania wiadomości). Uwaga: pięciokąt jest obwiedziony podwójnym okręgiem.
 Label	Exclusive Event-Based Gateway (zależna od zdarzenia wykluczająca)	Rozpoczynająca proces, gdy wystąpi jedno z kilku zdarzeń.
 Label	Parallel Event-Based Gateway (zależna od zdarzenia równoległa)	Rozpoczynająca proces, gdy wystąpi jednocześnie kilka zdarzeń.

Połączenia bramek przepływami sterowania (Sequence Flows)

- Użycie bramek wymaga ich połączenia przepływami z czynnościami lub innymi bramkami.

- Używane przepływy:
 - Zwykły (Sequence Flow) – przepływ wejściowy do bramki (np. łączy bramkę z poprzedzającą ją czynnością).
 - Warunkowy (Conditional Sequence Flow) – przepływ wyjściowy z bramki związany z warunkiem, którego spełnienie uruchomi ten przepływ (zostanie wykonana czynność wskazana przepływem).
 - Domyślny (Default Sequence Flow) – przepływ wyjściowy z bramki uruchamiany jako domyślny. Nie ma on własnego warunku, zostaje natomiast uruchomiony wtedy, gdy nie znajdują się żadne pozostałe przepływy wyjściowe z tej bramki (np. przyjęcie domyślnego sposobu płatności w sytuacji, gdy klient nie wybrał innego).

Uwaga

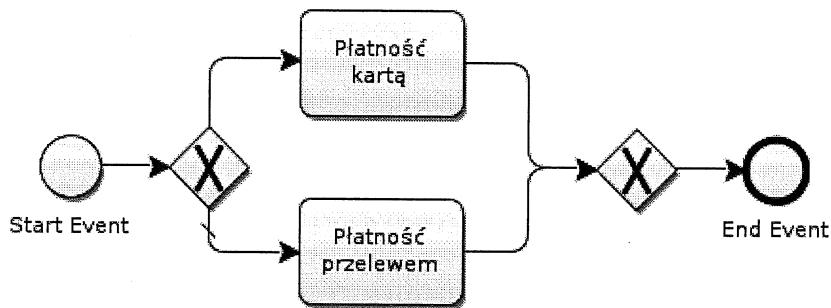
- Początkiem warunkowego przepływu sterowania (Conditional Sequence Flow) może być również czynność (Activity), ale w takiej sytuacji z tej czynności muszą wychodzić co najmniej dwa przepływy sterowania (wszystkie uwarunkowane).

Symbol	Nazwa	Opis
	Sequence Flow (przepływ sterowania)	Przepływ wejściowy do bramki.
	Conditional Sequence Flow (warunkowy przepływ sterowania)	Przepływ wyjściowy związany ze spełnieniem warunku. Jeśli wychodzi z bramki, romb na początku przepływu nie może być umieszczany. Nie może wychodzić z bramek typu Parallel oraz Event.
	Default Sequence Flow (domyślny przepływ sterowania)	Przepływ wyjściowy uruchamiany domyślnie. Może wychodzić z bramek typu Exclusive, Inclusive i Complex.

- **Uwaga:** aby warunek pojawił się na diagramie, trzeba jeszcze przejść do edycji etykiety przepływu (na przepływie klawisz [F2], np. wpisać i skasować spację, klawisz [Enter]).

Wyjście tylko jedną ścieżką (Exclusive Gateway)

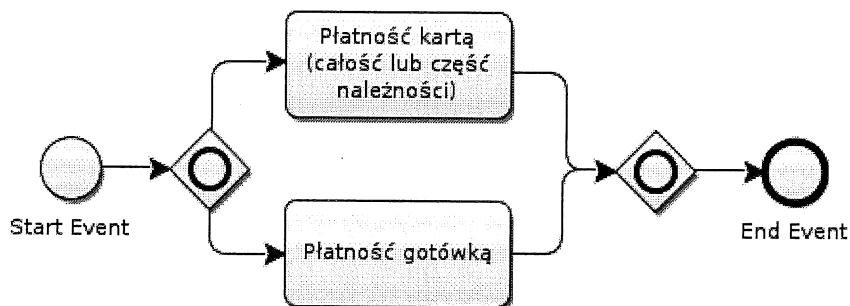
- Bramka wykluczająca (Exclusive Gateway) – jest to odpowiednik logicznego operatora xOr = „eXclusive Or” („albo, albo”), który oznacza, że jeden i tylko jeden warunek składowy może być prawdziwy, aby całość wyrażenia była prawdą.
- Jest to decyzja – warunki na wszystkich przepływach wyjściowych stanowią zestaw wykluczających się możliwości, co oznacza, że po otrzymaniu przepływu wejściowego, dopuszczalne jest wyjście z bramki jednym i tylko jednym przepływem (uzależnionym od spełnienia warunku związanego z tym przepływem).
- Np. w sklepie internetowym można uregulować płatność kartą płatniczą albo przelewem.



- Jeśli bramka wykluczająca łączy przepływy, każdy token przychodzący jest automatycznie kierowany na wyjście (uruchamia bramkę).
- Warunki są oparte na danych wykorzystywanych/dostępnych w ramach procesu (dla wykonawców czynności).
- Jeśli dane, na których opiera się decyzja, nie są dostępne bezpośrednio (czyli decyzję podejmuje inny obiekt niż wykonawca czynności), trzeba użyć bramki zależnej od zdarzenia (np. otrzymania wiadomości).
- Można ją oznaczać:
 - rombem ze znacznikiem X w środku,
 - pustym rombem.
- Zalecane jest utrzymanie jednolitości na każdym diagramie (czyli konsekwentnie wszystkie bramki tego rodzaju powinny być oznaczane takim samym symbolem).
- Można w niej stosować domyślny przepływ wyjściowy:
 - nie posiada on własnego warunku,
 - jest uaktywniany tylko wówczas, gdy żaden z warunków związanych z pozostałymi przepływami wyjściowymi nie daje wartości „prawda”.

Wyjście co najmniej jedną ścieżką (Inclusive Gateway)

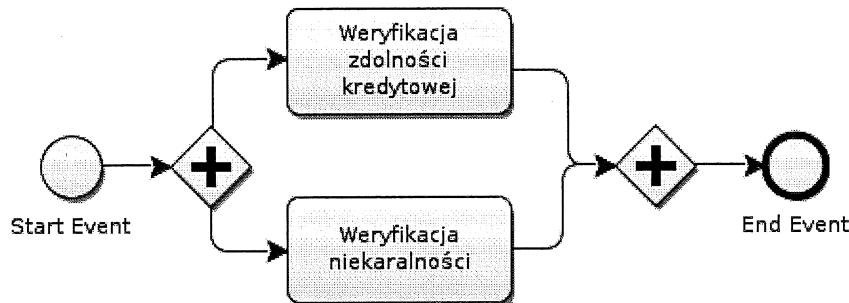
- Bramka niewykluczająca/włączająca (Inclusive Gateway) – jest to odpowiednik logicznego operatora Or = 'lub', który oznacza, że jeden lub więcej warunków składowych wyrażenia może być prawdziwy, aby całość była prawdą.
- Np. klient w kasie może zapłacić gotówką, ale jeśli nie posiada pełnej potrzebnej kwoty, może różnicę pokryć kartą płatniczą.



- Bramka ta może zatem tworzyć przepływy:
 - alternatywny – jeśli tylko jeden z przepływów wyjściowych posiada warunek prawdziwy w momencie trwania procesu,
 - wspólnie – jeśli przepływy wyjściowe z warunkami prawdziwymi jest więcej.
- Inaczej mówiąc – warunek prawdziwy dla jednego z przepływów wyjściowych nie wyklucza uruchomienia równocześnie innego przepływu, który również ma warunek prawdziwy (czyli odmiennie niż w przypadku bramki typu Exclusive).
- Bramka ta służy zarazem do:
 - łączenia przepływów rozgałęzionych wcześniej bramką typu Inclusive,
 - łączenia i synchronizacji ścieżek alternatywnych (wychodzących z bramek Inclusive i Exclusive) ze ścieżkami równoległymi (wychodzącymi np. z bramki typu Parallel).
- Może ona synchronizować przepływ tokenów docierających do niej później w stosunku do poprzednich.
- Warunki są oparte na danych wykorzystywanych/dostępnych w ramach procesu (dla wykonawcy czynności).
- Jeśli dane, na których opiera się decyzja, nie są dostępne bezpośrednio (czyli decyzję podejmuje inny obiekt niż wykonawca czynności), trzeba użyć bramki zależnej od zdarzenia (np. otrzymania wiadomości).
- Przy użyciu tej bramki w modelu trzeba projektować ją tak, by było z niej co najmniej jedno „skuteczne” wyjście adekwatne do modelowanej rzeczywistości. W przeciwnym razie proces będzie w niektórych sytuacjach ulegał „zawieszeniu” (nie będzie skuteczny, czyli nie będzie działał poprawnie).
- Może tu być stosowany przepływ domyślny – działający analogicznie, jak w przypadku bramki Exclusive:
 - nie ma on własnego warunku,
 - zostanie uruchomiony, jeśli wszystkie pozostałe przepływy będą mieć w danej sytuacji warunki dające fałsz.
- Jeśli nie będzie przepływu domyślnego, a żaden z pozostałych przepływów wyjściowych nie może zostać uruchomiony, wówczas mamy do czynienia z błędem procesu (runtime exception).

Wyjście wieloma ścieżkami równocześnie (Parallel Gateway)

- Bramka równoległa (Parallel Gateway) – wyjście wieloma ścieżkami jednocześnie.
- Np. ocena wiarygodności klienta starającego się o kredyt obejmuje toczące się równolegle procedury weryfikacji zdolności kredytowej i niekaralności.

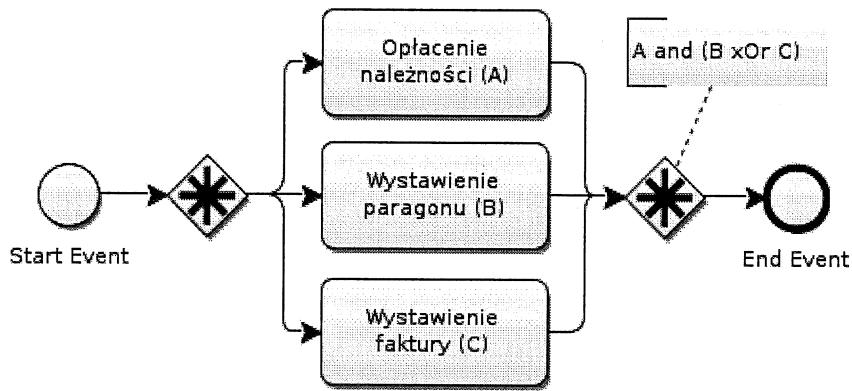


- Jeśli przepływy współbieżne:

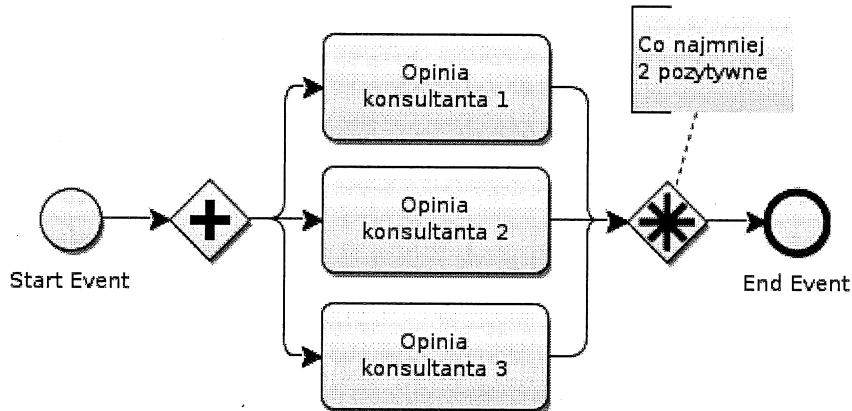
- rozpoczyna – wówczas przepływ wejściowy dzieli się na dwa lub więcej przepływów wyjściowych współbieżnych (z bramki wyrusza jeden token dla każdego przepływu wyjściowego),
- kończy – wówczas przepływy współbieżne wejściowe łączą się w bramce w jeden przepływ wyjściowy, zaś uruchomienie przepływu wyjściowego następuje po dotarciu do bramki wszystkich przepływów wejściowych.
- Bramka równoległa kończąca przepływ współbieżny jest odpowiednikiem operatora logicznego „And” („wszystkie warunki składowe muszą być prawdziwe, aby całość wyrażenia była prawdą”).

Złożona (Complex Gateway)

- Bramka złożona (Complex Gateway) – służy do synchronizacji (łączenia) przepływów w procesie, gdy potrzebne jest sprawdzanie różnych zestawów warunków składowych (różnych wariantów/konfiguracji tych warunków) dla przepływów wejściowych (czyli gdy nie wszystkie przepływy wejściowe muszą zostać zrealizowane, by bramka została uaktywniona).
- Może też rozpoczynać zestaw przepływów, z których nie wszystkie muszą być realizowane.
- W takiej sytuacji działa analogicznie do bramki typu Inclusive, ale jest używana, gdy bramka typu Inclusive nie może być użyta (bo nie daje ona możliwości określania zestawów przepływów wyjściowych – każdy przepływ wyjściowy w bramce Inclusive jest traktowany niezależnie od pozostałych).
- Przykład 1:
 - W kasie hipermarketu, aby klient mógł odebrać zakupiony towar (czynność wyjściowa), musi najpierw zapłacić (czynność A), a zarazem kasa może mu wydać tylko jeden z dwóch możliwych dowodów zakupu – paragon (czynność B) albo fakturę (czynność C).
 - Możliwe są zatem dwa zestawy warunków: A i B (zapłata i otrzymanie paragonu) albo A i C (zapłata i otrzymanie faktury).



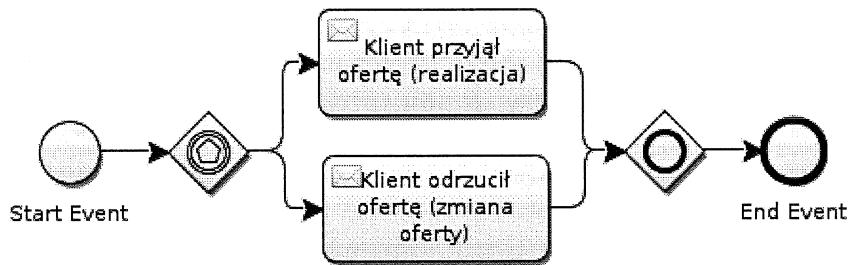
- Przykład 2 (łączenie przepływu współbieżnego):
 - Jednostka opiniująca realizację projektu przekazuje projekt do oceny 3 konsultantom.
 - Aby projekt został zatwierdzony do realizacji, wymagana jest pozytywna opinia przynajmniej 2 z nich.



- Uruchomienie przepływu wyjściowego w bramce typu Complex:
 - Jest uzależnione od warunku uruchamiającego (activationCondition) związanego z przepływami wejściowymi.
 - Warunek uruchamiający określa, które tokeny (czyli z których przepływów wejściowych pochodzące) mają uruchomić przepływ wyjściowy (lub przepływy wyjściowe).

Wyjście zależne od zdarzenia (Event-Based Gateway)

- Bramka zależna od zdarzenia (Event-Based Gateway) – służy do rozgałęzienia przepływu sterowania w momencie wystąpienia określonego zdarzenia.
- Np. klient informuje firmę o akceptacji warunków oferty, co rozpoczyna standardową ścieżkę realizacji oferty. Jeśli natomiast klient warunki odrzuci, rozpoczyna się inny ciąg czynności (np. ponowne przygotowanie oferty z nieco niższą ceną lub nieco szerszym zakresem oferowanych usług).

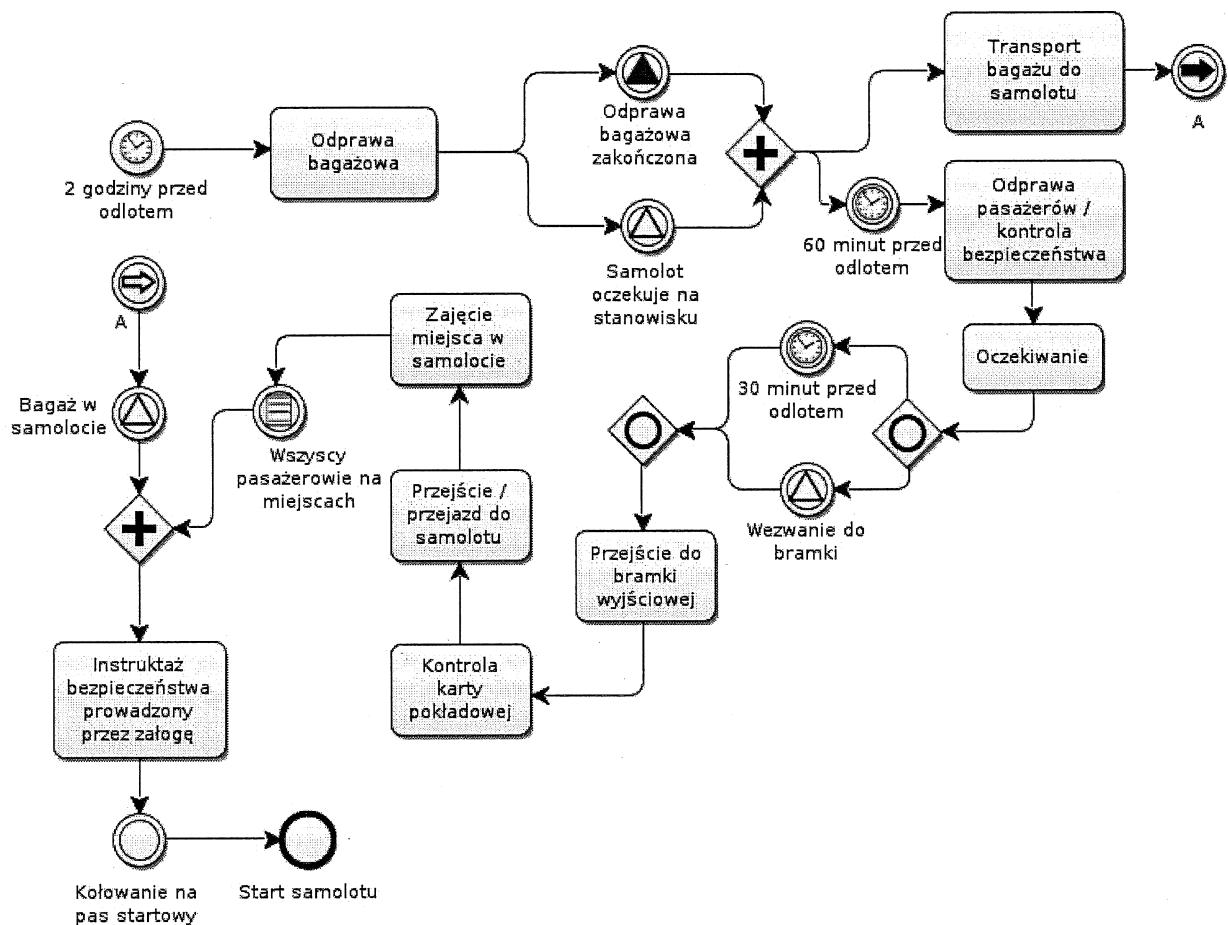


- Jeśli warunek oparty jest na danych dostępnych bezpośrednio uczestnikowi procesu, stosuje się raczej bramki Exclusive lub Inclusive.
- Zdarzeniem uruchamiającym bramkę Event-Based jest zwykle odebranie wiadomości (choć może być też inne), gdyż decyzja wpływająca na uruchomienie bramki jest zwykle podejmowana przez innego uczestnika procesu, ale na podstawie danych niewidocznych bezpośrednio w procesie.
- W przypadku bramki Event-Based:
 - muszą wystąpić co najmniej dwa przepływy wyjściowe,
 - żaden z przepływów wyjściowych nie może posiadać warunków.
- Ograniczenia użycia:
 - przepływy wyjściowe z bramki Event-Based mogą trafić tylko do zadań typu Receive Tasks lub do niektórych zdarzeń pośrednich Intermediate Events,
 - jeśli użyte są zdarzenia Message Intermediate Event, nie mogą być użyte zadania Receive Task i na odwrót,
 - Receive Tasks nie mogą mieć dołączonych zdarzeń pośrednich (Attached Intermediate Events),
 - elementy docelowe nie mogą otrzymywać żadnych innych przepływów sterowania z wyjątkiem tych, które wychodzą z poprzedzającej je bramki,
 - dopuszczalne są tylko zdarzenia typu: Message, Signal, Timer, Conditional i Multiple,
 - niedopuszczalne są natomiast zdarzenia typu Error, Cancel, Compensation i Link.
- Przepływy wyjściowe z bramki uruchamiane są na zasadzie wykluczenia:
 - jeśli zostanie uruchomiony jeden, pozostałe przestają funkcjonować (w danej instancji procesu),
 - czyli obowiązuje zasada „kto (który) pierwszy, ten lepszy”.
- W pewnych sytuacjach dopuszczalne jest użycie jednego z dwóch uszczegółowionych wariantów bramki Event-Based, które służą do oznaczania momentu utworzenia odrębnej instancji procesu (instantiate process):
 - Exclusive Event-Based Gateway – rozpoczyna proces jednym i tylko jednym spośród określonych zdarzeń,

- Parallel Event-Based Gateway – rozpoczyna proces, gdy jednocześnie wystąpi dwa lub więcej spośród określonych zdarzeń (i/lub synchronizuje dwa lub więcej zdarzeń, które rozpoczynają proces).
- Warianty bramki, które uruchamiają proces, nie otrzymują żadnych wejściowych przepływów sterowania (te, które otrzymują przepływy sterowania, oznaczają zwykłe bramki używane wewnątrz procesu).

Reguły łączenia bramek za pomocą Sequence Flows

- Wejściowe przepływy sterowania do bramek mogą pochodzić:
 - ze zdarzenia początkowego, rozpoczynającego proces (Start Event),
 - z zadania (Task),
 - z podprocesu (Sub-Process),
 - z innej bramki (Gateway),
 - ze zdarzenia pośredniego (Intermediate Event).
- Wyjściowe przepływy sterowania z bramek mogą wychodzić do:
 - niektórych zadań (Task),
 - podprocesu (Sub-Process),
 - innej bramki (Gateway),
 - zdarzenia pośredniego (Intermediate Event),
 - zdarzenia końcowego, kończącego proces (End Event).
- Przykład procesu z uwzględnieniem bramek:



Przetwarzanie grup obiektów – czynności iteracyjne i z wielokrotnione

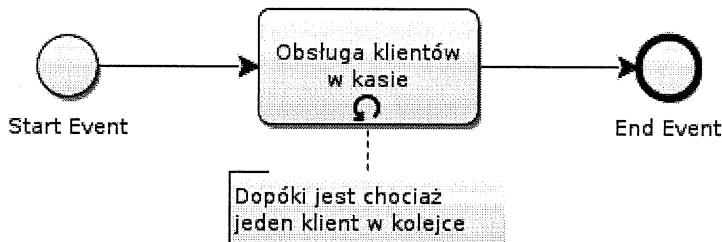
- W procesach pojawiają się często wymagające obsługi kolekcje obiektów – np. zbiory dokumentów do analizy/rozpatrzenia/zaopiniowania, kolejki obsługiwanych klientów, zestandardyzowane czynności wykonywane przez wiele zespołów roboczych w ramach jednej firmy itp.
- Przetwarzanie kolekcji obiektów może zachodzić:
 - w pętli, czyli iteracyjnej sekwencji działań (dopóki warunek jest prawdziwy) – Loop Activity,
 - pojedynczo dla każdego obiektu w grupie – Sequential Multiple Instances Activity,
 - równocześnie/współbieżnie dla obiektów w grupie – Parallel Multiple Instances Activity.
- Jeśli potrzebne jest pokazanie szczegółów wykonywanej czynności, używane jest oznaczenie podprocesu (Sub-Process), jeśli nie – oznaczenie zadania (Task).

Symbol	Nazwa	Opis
	Loop Task (zadanie powtarzalne /iteracyjne)	Wykonywane, dopóki warunek logiczny jest prawdziwy. Nie można łączyć z „multi-instance”.
	Sequential Multi-Instance Task (zadanie zwielokrotnione sekwencyjnie)	Wykonywane m.in. dla kolekcji (zbiorów) obiektów przetwarzanych w sekwencji (jeden po drugim).
	Parallel Multi-Instance Task (zadanie zwielokrotnione współbieżnie)	Wykonywane m.in. dla kolekcji (zbiorów) obiektów przetwarzanych współbieżnie (jednocześnie).
	Loop Sub-Process (pętla)	Podproces iteracyjny. Nie można łączyć z „multi-instance”.
	Sequential Multi-Instance Sub-Process (zwielokrotniony sekwencyjnie)	Podproces wykonywany w zwielokrotnieniu sekwencyjnie dla każdego obiektu w grupie.
	Parallel Multi-Instance Sub-Process (zwielokrotniony współbieżnie)	Podproces wykonywany w zwielokrotnieniu współbieżnie dla każdego obiektu w grupie.

Iteracyjna czynność powtarzalna (Loop Activity)

- Jest to czynność (Task lub Sub-Process), która jest wykonywana, dopóki warunek logiczny jest prawdziwy (odpowiednik przede wszystkim pętli z warunkiem typu „while” z różnych języków programowania).
- Warunek:
 - jest wyliczany przy każdej iteracji,
 - może być sprawdzany na początku lub końcu każdej iteracji (cyklu),
 - może być zastąpiony niezbędną liczbą powtórzeń (cykli).

- Np. obsługa klientów w kasie, dopóki jest chociaż jeden klient w kolejce.
- Np. gromadzenie na koncie środków do momentu osiągnięcia kwoty potrzebnej do realizacji zakupu.



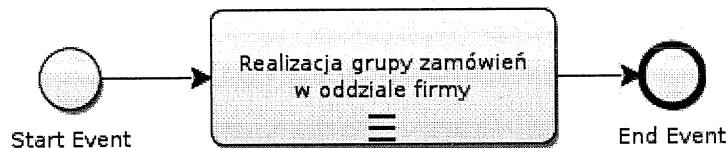
Wskazówki

- Pętle można też w wyjątkowych sytuacjach pokazać z użyciem:
 - bramek (Gateways) – poprowadzenie dodatkowego przepływu sterowania od czynności późniejszej kończącej powtarzaną sekwencję działań do bramki znajdującej się na wcześniejszym etapie procesu (bardzo przejrzyste w przypadku prostej sekwencji działań, obejmującej np. 1-2 czynności, dla których dodatkowo da się warunek kończący działanie pętli wyrazić przy przepływach wyjściowych z bramki rozpoczynającej taką pętlę),
 - łączy (Link Events) – w sytuacji podobnej, jak przy użyciu bramek, tylko zamiast przepływu sterowania użyta jest para Throw Link Event (przy czynności kończącej pętlę) i Catch Link Event (przy czynności rozpoczynającej pętlę).

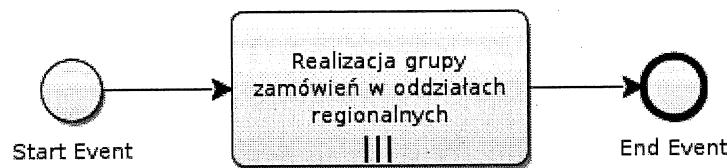
Czynność zwielokrotniona dla obiektów w grupie (Multiple Instances Activity)

- Służy do pokazania wykonania czynności wiele razy (tworzenia potrzebnej liczby instancji czynności), czyli do obsługi kolekcji obiektów lub danych, w których każdy obiekt z osobna (lub każda porcja danych) wymaga podjęcia określonych działań.
- Krotkość wykonania (czyli ilość instancji) czynności może:
 - być ustalona z góry,
 - opierać się na ilości przetwarzanych w procesie danych (w takiej sytuacji może zostać zarazem pokazane wejście/pobranie danych do procesu za pomocą elementu Data Input i połączenia Data Association).
- Czynności zwielokrotnione mogą być wykonywane:
 - sekwencyjnie (Sequential Multiple Instances Activity) – jedno wykonanie czynności dla jednego obiektu (lub porcji danych) z kolekcji,
 - równocześnie/współbieżnie (Parallel Multiple Instances Activity) – równoczesne wykonywanie czynności dla wszystkich obiektów (lub porcji danych) z kolekcji.
- Np. grupa zamówień realizowanych:

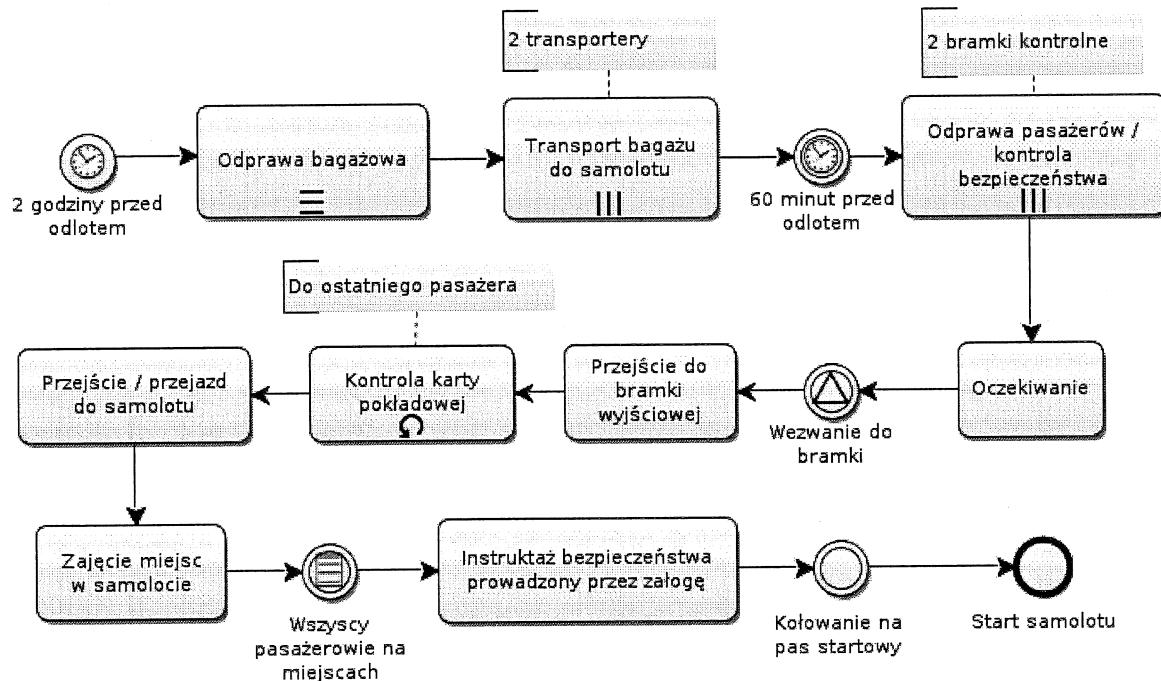
- sekwencyjnie – przez jeden oddział firmy,



- współbieżnie – przez kilku pracowników oddziału (bądź przez kilka oddziałów).



- Przykład procesu z uwzględnieniem czynności iteracyjnych i zwielokrotnionych:



Dane wykorzystywane w procesie (Data Objects)

- W procesach bardzo ważną rolę odgrywają rozmaite dane (Data Objects) tworzone, modyfikowane i używane, zatem zachodzi również potrzeba ich oznaczania w modelu.
- Obejmują one m.in. dane:
 - przetwarzane w trakcie procesu – np. zamówienie realizowane dla klienta,
 - informacje potrzebne do wykonania poszczególnych czynności – np. reguły przetwarzania, zasady działania, regulaminy pracy, procedury opisujące sposób wykonania pracy, parametry/właściwości obiektów podlegających przetwarzaniu lub wytworzeniu itp.,

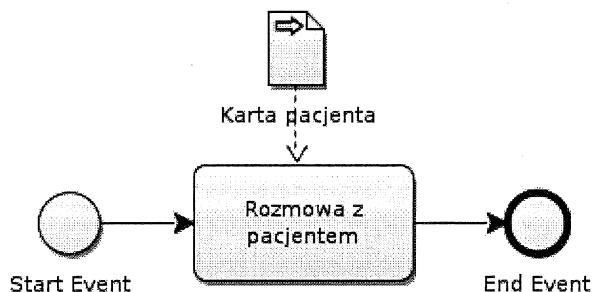
- wynikowe – które powstają w trakcie wykonywania czynności lub jako efekt końcowy całego procesu (i przekazywane dalej do innych procesów), np. dokumenty (elektroniczne czy fizyczne), rekordy w tabelach bazodanowych itp.,
- pobierane ze źródeł zewnętrznych – systemów, baz danych, archiwów itp. (np. rekordy pobrane z bazy, służące do przygotowania analizy/raportu itp.).
- Dane mogą trafiać do procesu:
 - pojedynczymi porcjami – np. pojedyncze zamówienia,
 - grupami/kolekcjami – np. karty pacjentów, których ma przyjąć lekarz podczas dyżuru,
 - za pośrednictwem mechanizmów pobierania i aktualizowania (Data Stores) – np. systemu księgowego czy magazynowego.
- Dane dzielą się na:
 - wejściowe (Data Input) – wprowadzane/pobierane do procesu/czynności,
 - wyjściowe (Data Output) – wytworzzone/wyprowadzane z procesu/czynności.
- Dane są łączone z innymi elementami diagramu za pomocą Data Associations, zaś połączenia te mogą prowadzić:
 - od/do elementów przepływu w procesie (np. czynności),
 - do przepływów sterowania – alternatywne skrótowe oznaczenie, gdy dane przepływiąją np. między czynnościami połączonymi przepływem sterowania, ale nie jest zarazem potrzebne podkreślanie elementu wyjściowego i wejściowego danych.
- Data Objects muszą być umieszczane w obrębie procesu lub podprocesu, którego dotyczą.
- Możliwe jest stosowanie wskaźników (Data Object Reference), które umożliwiają wielokrotne wykorzystanie raz zdefiniowanego obiektu danych.
- Dane mogą posiadać właściwości (Properties), ale właściwości te nie są pokazywane bezpośrednio na diagramie.

Symbol	Nazwa	Opis
	Data Object (dane)	Ogólna reprezentacja danych w procesie.
	Data Object / collection (kolekcja danych)	Użycie kolekcji/zbioru danych. Np. karty pacjentów oczekujących na wizytę u lekarza.
	Data Store (magazyn danych, repozytorium)	Skorzystanie z repozytorium („sięgniecie do magazynu”). Np. znalezienie i przyniesienie do gabinetu lekarskiego kart pacjentów z kartoteki przychodni (tu: kartoteka = repozytorium/„magazyn” kart pacjentów).

Symbol	Nazwa	Opis
	Data Input (dane wejściowe)	Dane wchodzące do procesu/czynności. Np. karta pacjenta, z której korzysta przyjmujący go lekarz.
	Data Output (dane wyjściowe)	Dane wychodzące z procesu/czynności. Np. recepta wypisana przez lekarza pacjentowi.
	Data Association (połączenie danych)	Połączenie danych z innymi elementami diagramu. Może łączyć np. czynność z symbolem danych lub symbol danych np. z przepływem sterowania między czynnościami (wówczas nie wymaga użycia strzałki).

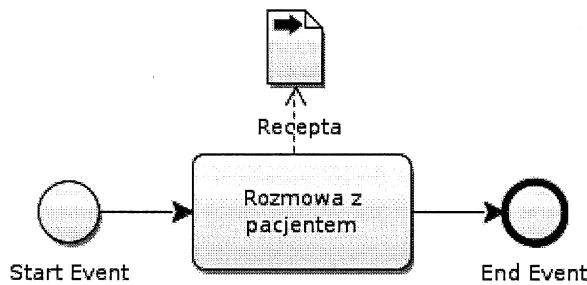
Pobieranie/wprowadzanie danych (Data Input)

- Dane wejściowe (Data Input) – to określenie konkretnego rodzaju danych pojawiających się na wejściu (wchodzących i wykorzystywanych w procesie/czynności).
- Dane wejściowe mogą być kolekcją – wówczas na symbolu danych pojawia się dodatkowe oznaczenie kolekcji (trzy pionowe kreseczki).
- Np. karta pacjenta, z której korzysta przyjmujący go lekarz.



Przekazywanie danych wyjściowych (Data Output)

- Dane wyjściowe (Data Output) – to określenie konkretnego rodzaju danych, które mogą zostać wytworzone jako efekt działania (procesu/czynności).
- Dane wyjściowe na głównym poziomie procesu oznaczają efekt działania całego tego procesu.
- Dane wyjściowe mogą być kolekcją – wówczas na symbolu danych pojawia się dodatkowe oznaczenie kolekcji (trzy pionowe kreseczki).
- Np. recepta wypisana przez lekarza pacjentowi.

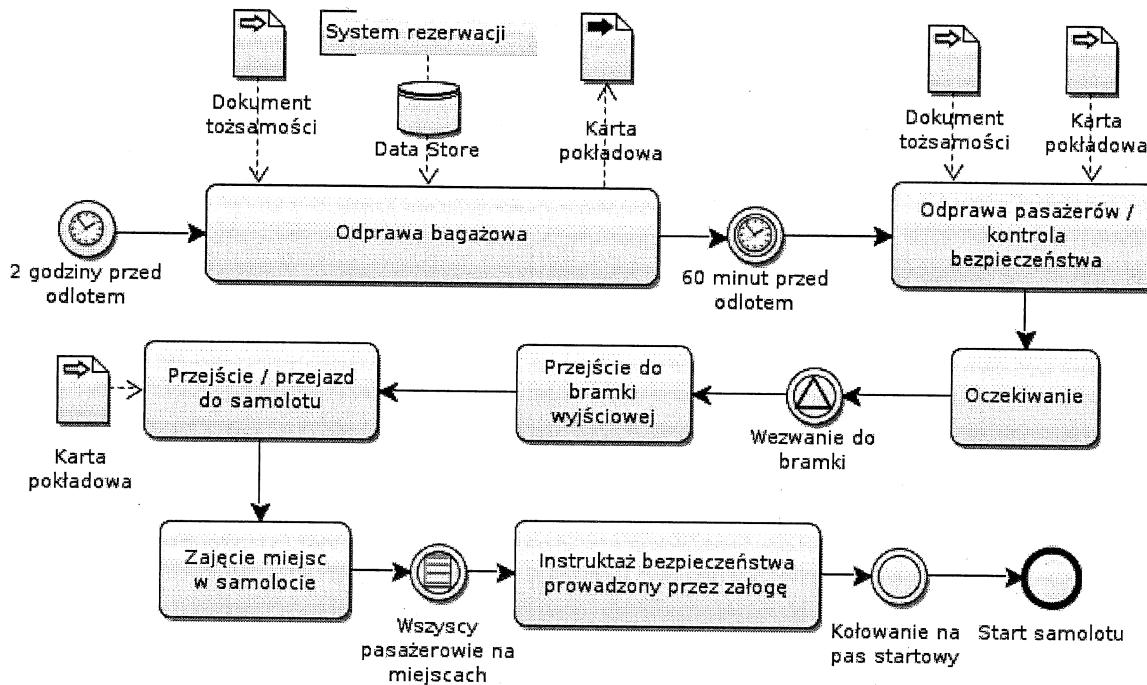


- **Service Task (usługa):**
 - Jeśli jest powiązane z działaniem (Operation), musi posiadać dane wejściowe wiadomości (Message Data Input) oraz wejście inputSet.
 - Jeśli operacja, z którą zwiążane jest zadanie, wysyła wiadomość, wówczas zadanie to ma również jedno wyjście danych (Data Output) związane zową wiadomością oraz wyjście outputSet.
- **Send Task (wysłanie wiadomości):**
 - Ma co najwyżej jedno wejście danych (Data Input) oraz wejście inputSet – jeśli jest powiązane z wiadomością (dane wejściowe dla wysyłanej wiadomości).
 - Jeśli posiada dane wejściowe, muszą one być adekwatne do wysyłanej wiadomości związanego z zadaniem – w momencie wykonania zadania automatycznie przechodzą one do wysyłanej wiadomości.
 - Jeśli danych wejściowych nie ma, wysłana w trakcie realizacji zadania wiadomość nie będzie ich zawierać (czyli nie będzie przenosić danych pochodzących z procesu).
- **Receive Task (odebranie wiadomości):**
 - Jeśli jest powiązane z wiadomością, musi mieć przynajmniej jeden zestaw danych wyjściowych Data Output (tzn. danych, które z odebranej wiadomości trafią do procesu) wraz z wyjściem outputSet. W przeciwnym razie dane zawarte w wiadomości (payload within the Message) nie wyjdą z zadania, czyli nie trafią do procesu.
 - W momencie wykonania zadania dane przechodzą automatycznie na wyjście danych.
 - **User Task** (zadanie wykonywane przez użytkownika) – ma dostęp do danych wejściowych, wyjściowych i innych elementów danych będących w zasięgu zadania.
 - **Script Task** (wykonanie skryptu) – ma dostęp do danych wejściowych, wyjściowych i innych elementów danych będących w zasięgu zadania.
 - **Call Activity** (proces predefiniowany) – wejścia i wyjścia danych odwołują się do adekwatnych odpowiedników w obrębie procesów, z których wywołanie pochodzi, bez jawnego powiązania danych (Data Associations).
 - **Events (zdarzenia)** – jeśli zdarzenie ma związek z elementem wymagającym danych (takim jak wiadomość, nadzwyczajna interwencja/eskalacja, błąd lub sygnał), wówczas podlega następującym ograniczeniom:

- Jeśli zdarzenie jest powiązane z wieloma definicjami zdarzeń (multiple EventDefinitions), musi posiadać jedno wejście (jeśli jest to zdarzenie typu `throw`) lub jedno wyjście danych (jeśli jest to zdarzenie typu `catch`) dla każdej z definicji.
- Kolejność definicji zdarzeń i kolejność wejść/wyjść danych ma wpływ na przyporządkowanie wejść/wyjść danych do poszczególnych zdarzeń.
- Dla każdej pary „definicja zdarzenia – wejście/wyjście danych”, która posiada wejście/wyjście danych, musi być podany odpowiednik porcji danych (`ItemDefinition`) dla tego, co jest określone przez wiadomość, eskalację, błąd lub sygnał odnoszący się do powiązanej definicji zdarzenia.
- W przypadku zdarzenia typu `throw`, jeśli nie ma wejścia danych, wówczas wiadomość, eskalacja, błąd lub sygnał nie zostanie przekazana razem z danymi.
- W przypadku zdarzenia typu `catch`, jeśli nie ma wyjścia danych, wówczas zawartość wiadomości, eskalacji, błędu lub sygnału nie wyjdzie ze zdarzenia i nie trafi do procesu.

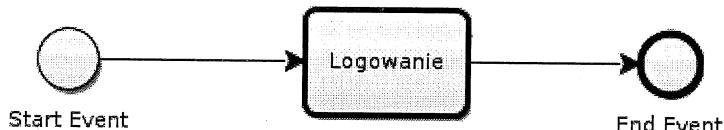
Połączenia danych z elementami procesu (Data Associations)

- Połączenia danych (Data Associations) oznaczają przekazanie danych do/z czynności, procesów oraz predefiniowanych zadań (GlobalTasks).
- Przepływ danych nie ma bezpośredniego wpływu na sam proces – przez przepływ danych nie przechodzą tokeny.
- Przepływ danych może oznaczać samo ich skopiowanie (np. pobranie danych klienta) lub też skopiowanie i przekształcenie (np. z notatek powstaje tekst artykułu).
- Jeśli nie ma zdefiniowanego sposobu przekształcenia danych, wówczas może zostać określone tylko jedno źródło danych, zaś połączenie danych będzie oznaczać wyłącznie skopiowanie ich ze źródła do elementu docelowego.
- Przykład procesu z uwzględnieniem danych wejściowych i wyjściowych:



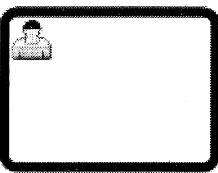
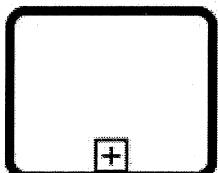
Użycie predefiniowanego procesu lub zadania (Call Activity)

- Czynność przywołania (Call Activity lub Reusable Sub-Process w BPMN 1.2) – umożliwia włączenie do procesu zadań i procesów globalnych (predefiniowanych dla całego modelu).
- Np. włączenie do procesu obsługi klienta w systemie logowania do tego systemu (proces zestandardyzowany i powtarzający się w kontekście różnych procesów nadrzędnych).



- Pozwala upraszczać modele szczególnie w przypadku opisu procesów pracy standaryzowanej (np. w Toyota Production System), ale też wszędzie tam, gdzie występują zestandardyzowane działania biznesowe.
- Czynność przywołania korzysta z takich samych danych i odwołań do danych, jak proces lub zadanie przywoływanie (dane wejściowe, dane wyjściowe, kolekcje danych itp.), gdyż jest to de facto przekazanie sterowania do takiego procesu/zadania.
- Zadania globalne (Global Tasks) – zadania zdefiniowane wspólnie dla całego modelu (do wykorzystywania w różnych momentach trwania procesu).
- BPMN przyjmuje jako globalne tylko następujące zadania:
 - wykonywane przez użytkownika (Global User Task),
 - manualne (Global Manual Task),

- zastosowanie reguły biznesowej (Global Business Rule Task),
 - wykonanie skryptu (Global Script Task).
- Czynność przywołania:
- procesu globalnego – może być prezentowana ze szczegółami ukrytymi (Collapsed) lub widocznymi (Expanded), czyli tak samo, jak podproces, ale z krawędzią rysowaną pogrubioną linią,
 - zadania globalnego – jest oznaczana tak, jak odpowiedni do potrzeb typ przywoływanego zadania, ale również z krawędzią rysowaną pogrubioną linią.

Symbol	Nazwa	Opis
	Call Activity calling a Global Task (przywołanie predefiniowanego zadania)	Tu: przywołanie zadania wykonywanego przez użytkownika.
	Call Activity calling a Process (przywołanie predefiniowanego procesu)	Tu: z ukrytymi szczegółami.

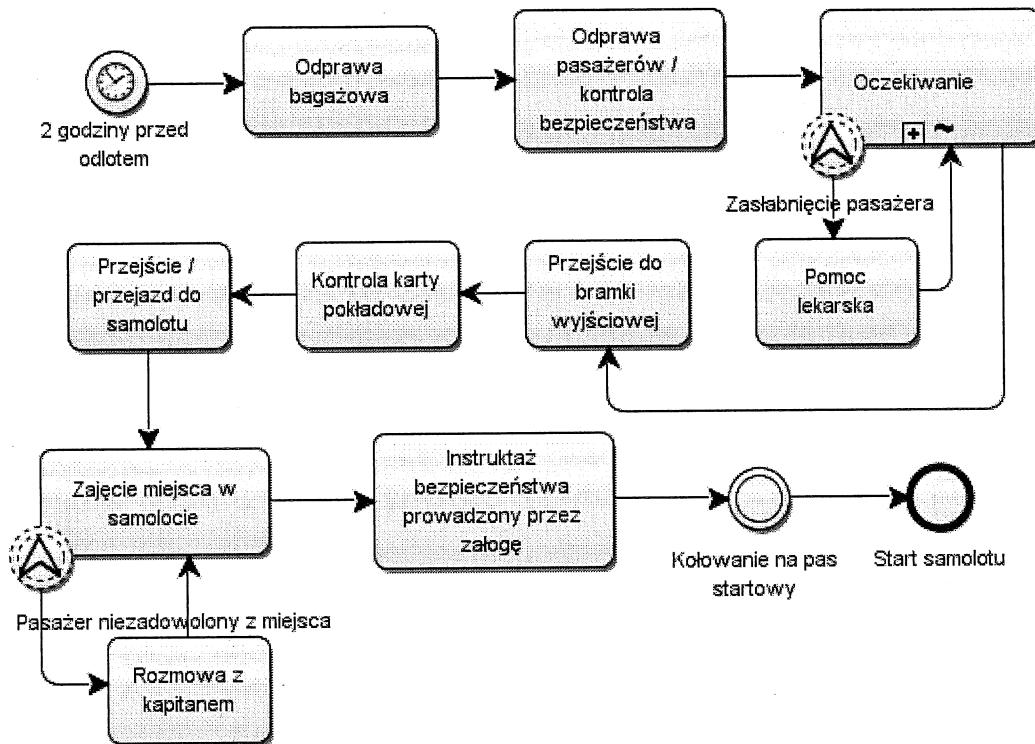
Nietypowe etapy procesu

- Poza rutynowym przebiegiem procesu (normal flow) mogą się w nim pojawić różne niezaplanowane elementy, które mają wpływ na przebieg.
- Trzeba przewidywać i uwzględniać w modelu te elementy, aby zapewnić procesom jak najbardziej płynny, niezakłócony i efektywny przebieg w różnych potencjalnie możliwych warunkach.
- Np.:
 - ingerencja zewnętrzna (specjalista wezwany w celu rozwiązania nietypowego problemu),
 - nieoczekiwany wpływ czynników zewnętrznych (warunków pogodowych na transport i terminowość dostaw),
 - konieczne działanie doraźne podjęte przez uczestnika procesu (zatrzymanie pojazdu ze względu na zwalone na drogę drzewo).

| Nadzwyczajna ingerencja w działanie procesu (Escalation)

- Escalation – oznaczenie sytuacji, w której może być potrzebna nadzwyczajna (zewnętrzna) interwencja i ingerencja w przebieg procesu.
- Nadzwyczajna ingerencja może być oznaczana za pomocą:
 - zdarzenia końcowego (End Event) typu throw – jeśli jest efektem działania procesu lub podprocesu,
 - zdarzenia pośredniego (Intermediate Event) typu throw w głównym przebiegu procesu – jeśli jest potrzebna w trakcie działania procesu,
 - zdarzenia pośredniego (typu catch) dołączonego do czynności (Boundary Intermediate Event) – jeśli jest obsługiwana konkretną czynnością,
 - podprocesu uruchamianego zdarzeniem (Event Sub-Process), dla którego zdarzeniem początkowym będzie eskalacja – jeśli taka sama obsługa eskalacji może być konieczna w różnych momentach trwania procesu.

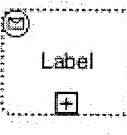
- Przykład procesu z uwzględnieniem eskalacji (nadzwyczajnych interwencji):



Uwzględnianie wpływu czynników zewnętrznych – podprocesy wywoływane zdarzeniami (Event Sub-Process)

- Event Sub-Process – specyficzny rodzaj czynności, która może stanowić część procesu głównego lub część innego podprocesu.
- Nie jest natomiast częścią normalnego przepływu w procesie, czyli:
 - nie kończy ani nie rozpoczyna żadnego przepływu sterowania,
 - nie musi, choć może zadziałać (raz lub więcej) w trakcie trwania procesu głównego.
- Zatem podproces:
 - wywołyany zdarzeniem – jest jedynie możliwym uzupełnieniem/rozbudową, która może zostać wykorzystana w dowolnym momencie działania procesu głównego,
 - wywołyany zdarzeniem – musi posiadać określoną przyczynę, która doprowadzi do jego uruchomienia,
 - zwykły – jest natomiast jednym z etapów przepływu, uruchamianym zawsze, jeśli znajduje się na aktualnej ścieżce działań bieżącego procesu (konieczny element przepływu).
- Obsługuje sytuacje, w których na proces działają następujące czynniki zewnętrzne:
 - odebranie wiadomości (Message),
 - wystąpienie błędu (Error),
 - konieczność nadzwyczajnej interwencji (Escalation),

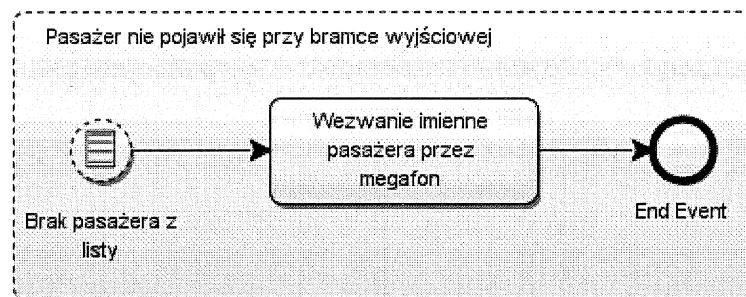
- konieczność kompensacji (Compensation),
- spełnienie określonego warunku (Conditional),
- odebranie sygnału (Signal),
- połączenie kilku zdarzeń (Multiple).
- Musi posiadać jedno i tylko jedno początkowe zdarzenie uruchamiające.
- Może posiadać tylko jedno zdarzenie końcowe.
- Może rozpoczynać się zdarzeniem Compensation Start Event (jeśli dokonuje kompensacji) i/lub kończyć się zdarzeniem Compensation End Event (jeśli samo kompensacji wymaga).
- Zdarzenie uruchamiające może, ale nie musi przerywać procesu nadzędny.

Symbol	Wyzwala / przyczyna	Opis
	Event Sub-Process (podproces)	Oznaczenie podprocesu (tu: zwinięte – bez pokazywania szczegółów). Symbol zdarzenia w lewym górnym rogu odpowiada zdarzeniu uruchamiającemu umieszczonemu wewnątrz tego podprocesu. Krawędź rysowana linią kreskową.
	Message – Interrupting (wiadomość – przerywające)	Odebrana wiadomość wymaga przerwania procesu nadzędnego i przejścia do podprocesu wywołanego tym zdarzeniem. Np. klient powiadamia firmę o rezygnacji ze złożonego wcześniej zamówienia (czyli realizacja zamówienia zostaje przerwana), natomiast firma rozpakowuje towary przygotowane do wysyłki i ew. obciąża klienta przewidzianymi w regulaminie kosztami manipulacyjnymi.
	Message – Non-Interrupting (wiadomość – nieprerywające)	Zostanie uruchomiona czynność wywołana odebraniem wiadomości. Np. wiadomość o zmianie listy gości sympozjum powoduje wycofanie identyfikatorów dla osób, które się nie pojawią i/lub przygotowanie identyfikatorów dla osób zgłoszonych w zamian.
	Escalation – Non-Interrupting (nadzwyczajna interwencja – nieprerywające)	Obsługa eskalacji – podjęcie nadzwyczajnej interwencji. Np. kierownik kwiatarni wysyłkowej podejmuje decyzję o dostarczeniu klientowi gratis dodatkowego kosza z różami (jako rekompensatę za opóźnienie w dostawie zamówionego przez tego klienta bukietu).

Symbol	Wyzwalač / przyczyna	Opis
	Escalation – Interrupting (nadzwyczajna interwencja – przerywające)	Obsługa eskalacji – nadzwyczajna interwencja przerywa wykonanie procesu nadzędnego podlegającego eskalacji. Np. główny inżynier decyduje o wstrzymaniu linii produkcyjnej (ze względu na wadliwą partię części dostarczonych przez kontrahenta), natomiast uruchomiony podproces obejmuje działania związane z jak najszybszym dostarczeniem na linię produkcyjną odpowiednich części (np. przez odesłanie do dostawcy wadliwych części z prośbą o natychmiastową ich wymianę).
	Error (błąd)	Obsługa błędu określonego w nazwie zdarzenia lub dowolnego błędu (jeśli błąd nie został określony). Np. niekompletne dane dostarczone przez klienta przerywają proces nadzędny wprowadzania danych do systemu, natomiast podproces będzie obejmował uzyskanie brakujących danych (kontakt telefoniczny z klientem lub e-mail z prośbą o uzupełnienie danych).
	Compensation (kompensacja)	Uruchomienie czynności obsługi zdarzenia kompensacji (typu throw) z głównego przebiegu procesu nadzędnego. Np. klient rezygnuje przy kasie z zakupu towarów, ponieważ zapomniał/zgubił portfel (zatem następuje przerwanie procesu nadzędnego zakupu towarów), zaś uruchomiona kompensacja będzie polegała na odłożeniu towarów przez obsługę sklepu z powrotem na półki.
	Conditional – Interrupting (warunkowe – przerywające)	Spełnienie warunku związanego z tym zdarzeniem przerywa proces nadzędny. Np. trzecie niepoprawne logowanie do systemu powoduje przerwanie procesu uzyskiwania dostępu do systemu (przez zablokowanie sesji użytkownika).
	Conditional – Non-Interrupting (warunkowe – nieprerywające)	Spełnienie warunku związanego z tym zdarzeniem uruchamia podproces. Np. został osiągnięty minimalny niezbędny poziom części w magazynie, zatem zakład zamawia u dostawcy kolejną partię części do montażu (ale proces produkcji nie zostaje przerwany – taśma montażowa nadal pracuje).
	Signal – Interrupting (sygnał – przerywające)	Odebranie sygnału przerywa proces nadzędny i uruchamia podproces. Np. sygnał o awarii hydrauliki podnośnika w wywrotce powoduje przerwanie kursu ciężarówki do miejsca docelowego i zjazd do bazy (podproces).

Symbol	Wyzwalać / przyczyna	Opis
	Signal - Non- Interrupting (sygnał - nieprzerwywające)	Po odebraniu sygnału zostaje uruchomiony podproces bez przerywania procesu nadziednego. Np. sygnał o korkach na pierwotnej trasie autokaru powoduje ustalenie przez pilota wycieczki nowej trasy i skierowanie na nią kierowcy (podproces).
	Multiple - Interrupting (wielokrotne – przerywające)	Po wystąpieniu jednej z określonych przyczyn proces nadziedny zostaje przerwany i uruchomiony zostaje podproces. Np. dokonywana przez serwis sklepu komputerowego „kontrola zerowa” fabrycznie zapakowanego zestawu komputerowego wykazuje uszkodzenie maszyny w czasie transportu (w wyniku zalania wodą), zatem przerwane zostaje przygotowanie komputera dla klienta (proces nadziedny), zaś rozpoczęta zostaje procedura reklamacyjna u kuriera lub dostawcy (podproces). Inne możliwe przyczyny uruchomienia tego podprocesu to stwierdzenie braku jednego z zamówionych elementów zestawu czy teżomyłkowe dostarczenie innego zestawu niż zamówiony.
	Multiple - Non- Interrupting (wielokrotne – nieprzerwywające)	Po wystąpieniu jednej z określonych przyczyn zostaje uruchomiony podproces (bez przerywania procesu nadziednego). Np. dokonywana przez serwis sklepu komputerowego „zerowa kontrola” fabrycznie zapakowanego zestawu komputerowego wykazuje obluzowanie obudowy, zatem przygotowanie sprzętu dla klienta jest kontynuowane (proces nadziedny), ale usterka zostaje natychmiast usunięta przez dokręcenie obluzowanych śrubek (podproces).

- Przykład podprocesu uruchamianego zdarzeniem:

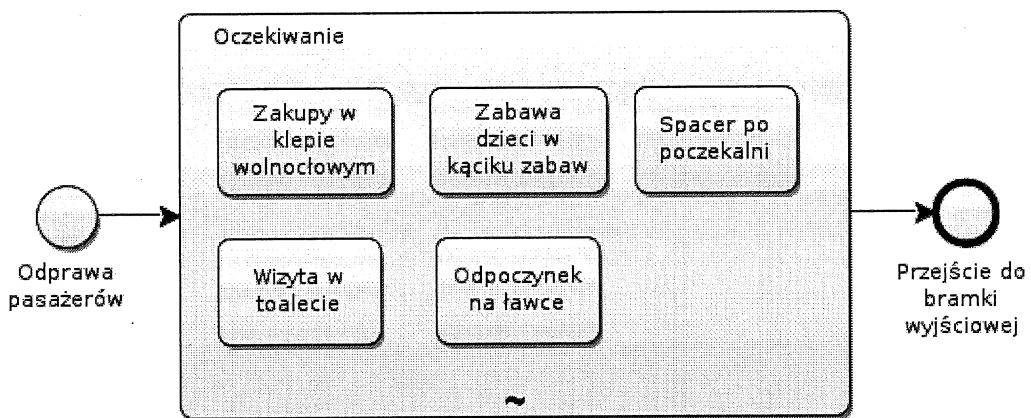


Podprocesy zależne od wykonawcy (Ad-Hoc Sub-Process)

- Ad-Hoc Sub-Process – wydzielona w ramach procesu grupa czynności, które nie muszą być ze sobą bezpośrednio powiązane (każda z nich może, ale nie musi być

wykonywana), natomiast o ich wykonaniu, częstotliwości czy ewentualnej kolejności decydują wykonawcy.

- Zwykle czynności wewnętrz takiego podprocesu nie są ze sobą powiązane przepływanymi sterowaniem, ale w razie potrzeby można takie połączenia ustalić.
- Np. przygotowanie rozdziału książki lub raportu dla zarządu, gdzie trzeba zebrać potrzebne informacje, przygotować tekst, być może trzeba będzie przygotować i dołączyć do niego jedną lub więcej grafikę (wykresy, ilustracje itp.), zapewne trzeba będzie ten tekst ostatecznie zredagować itd.
- Ograniczenia – w tego typu podprocesach:
 - muszą pojawić się czynności,
 - mogą wystąpić dane (Data Objects i Data Associations), zdarzenia (Intermediate Events), bramki (Gateways), przepływy sterowania (Sequence Flows), grupy (Groups) asocjacje (Associations) oraz wychodzące lub przychodzące przepływy wiadomości (Message Flows),
 - nie mogą wystąpić: zdarzenie początkowe (Start Event), zdarzenie końcowe (End Event), oznaczenia konwersacji i ich połączeń (Conversations i Conversation Links) oraz oznaczenia choreografii (Choreographies) – o 3 ostatnich elementach będzie jeszcze mowa w dalszej części szkolenia.
- Jeśli w podprocesie pojawią się symbole danych wejściowych związane z zadaniami, znaczy to, że wykonawca zadania nie może go rozpoczęć bez odpowiednich danych (np. notatki potrzebne do redagowania tekstu).
- Jeśli zadania zostaną połączone przepływaniami sterowania, znaczy to, że muszą one być wykonywane w podanej kolejności, choć zadanie kolejne nie musi być wykonywane natychmiast po zakończeniu poprzedniego.
- Przykład podprocesu uzależnionego od wykonawcy:



Błędy i przerwania procesu

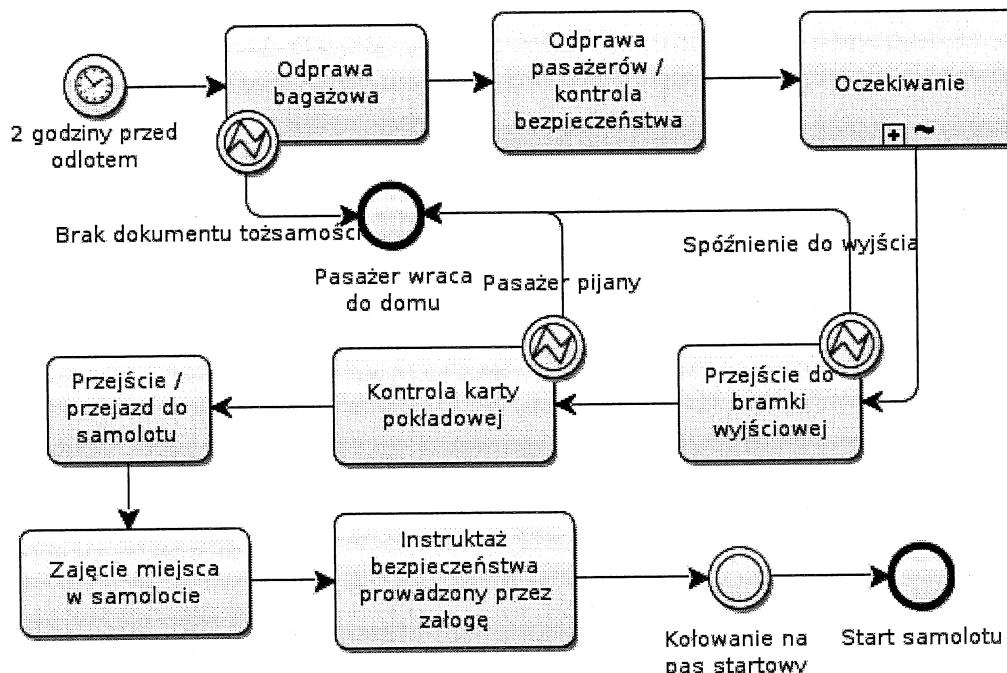
Obsługa błędów procesu

- Błąd (zdarzenie Error) oznacza, że:
 - pojawił się poważny problem podczas wykonywania czynności lub jej wykonanie się nie powiodło,
 - czynność/proces zostaje w tym momencie przerwana lub zawieszona.
- Błąd wymaga zawsze obsługi.
- Błąd nieobsłużony powoduje, że:
 - zachowanie procesu jest nieokreślone – zaprojektowany proces będzie niestabilny,
 - pojawią bardzo duże problemy w praktyce (np. szkodliwy chaos w środowisku pracy czy też zawieszanie się programów komputerowych).
- Błąd może być oznaczany jako:
 - Zdarzenie końcowe (End Event) typu throw – jeśli jest ostatecznym wynikiem działania procesu (najczęściej obocznym w stosunku do głównego celu procesu).
 - Zdarzenie pośrednie (Intermediate Event) typu throw w głównym przebiegu procesu – jeśli dotyczy to zakłócenia tego przebiegu (np. sytuacji, gdy do procesu trafiają błędne dane wejściowe).

Uwaga: sprzeczność w specyfikacji – powyższe sformułowanie dot. Error Intermediate Event znajduje się w tabeli 8.41 - *Error attributes and model associations* (na s. 110). Natomiast w tabeli 10.90 - *Intermediate Event Types Attached to an Activity Boundary* (na s. 284) oraz w sekcji *Sequence Flow Connections* (na s. 288) jest wyraźnie zapisane, że to zdarzenie pośrednie może być dołączane tylko do czynności i nie może być używane w normalnym przebiegu procesu.

- Zdarzenie pośrednie typu catch dołączone do czynności (Boundary Intermediate Event).
- Zdarzenie początkowe (Start Event) typu catch, podprocesu uruchamianego zdarzeniem (Event Sub-Process) – jeśli podproces ten służy do obsługi błędu.

- Przykład procesu z uwzględnieniem błędów:



Przerwania procesu

- Każda czynność może zostać przerwana.
- Przerwanie może dotyczyć następujących sytuacji:
 - wystąpienie błędu (zob. *Obsługa błędów procesu*, s. 64),
 - wystąpienie innego przerywającego (Interrupting) zdarzenia Boundary Intermediate Event (np. odebranie wiadomości czy też sygnału), zob. też *Zdarzenia włączane do czynności*, s. 30,
 - przerwanie zwykłego procesu/podprocesu przez uczestnika – zdarzenie Terminate End Event (zob. też *Rodzaje efektów działania procesu*, s. 12),
 - anulowanie podprocesu transakcyjnego – zdarzenie Cancel Boundary Event (zob. też *Podprocesy transakcyjne (Transactions)*, s. 68).
- We wszystkich tych przypadkach (z wyjątkiem Terminate End Event) potrzebne lub nawet konieczne jest wykonanie dodatkowych czynności związanych z przerwaniem, np. poinformowanie uczestników o zaistniałej sytuacji lub też wycofanie czynności (zob. też *Wycofanie kroków zakończonej czynności (Compensation)*, s. 65).

Wycofanie kroków zakończonej czynności (Compensation)

- Kompensacja – wycofanie kroków zakończonej wcześniej czynności, jeśli jej wynik lub możliwe efekty nie są już potrzebne i powinny zostać wycofane.
- Np. odwołanie rezerwacji lotu oraz noclegu w hotelu w przypadku rezygnacji z uczestnictwa w targach.

- Np. uprątnięcie narzędzi po wykonaniu naprawy lub zakończeniu pracy.
- Ograniczenia:
 - nie można kompensować czynności, która trwa,
 - jeśli dotyczy podprocesu, obejmuje te jego czynności składowe, które zostały wykonane (i wymagają kompensacji).
- Obsługa kompensacji (compensation handler):
 - obejmuje wszystkie kroki niezbędne do odwrócenia efektów czynności i ma dostęp do danych wycofywanego podprocesu,
 - jest realizowana za pomocą czynności (Task lub Sub-Process).

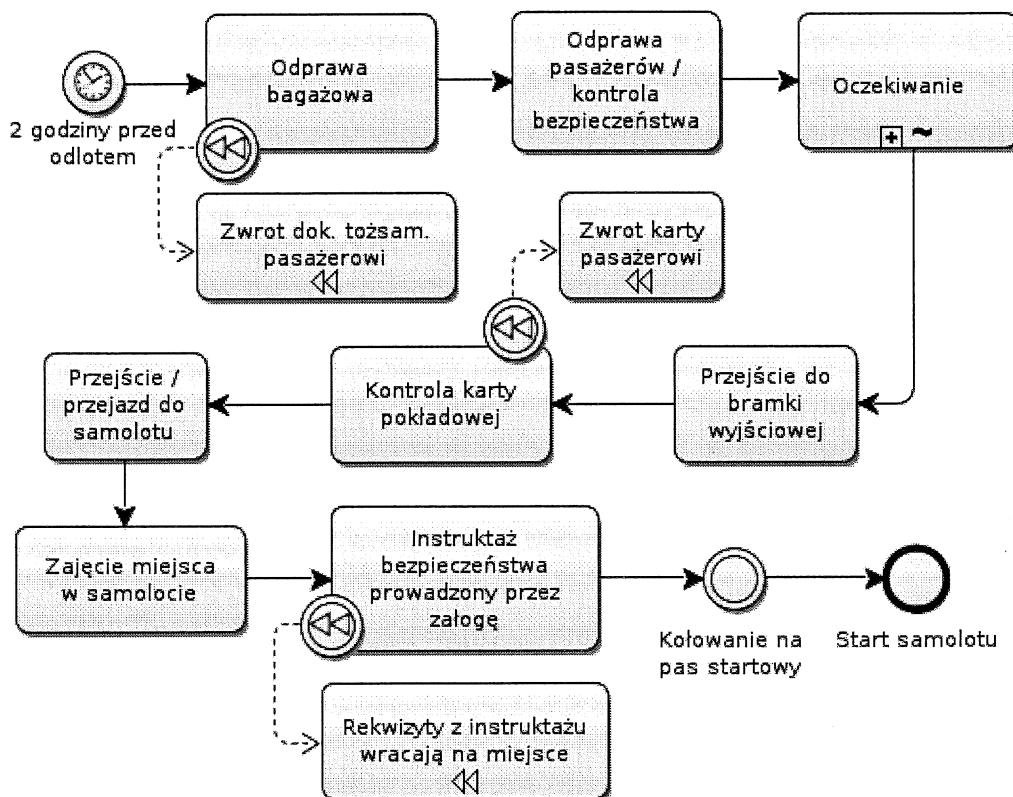
Uruchamianie kompensacji

- Kompensacja jest wywoływana przez zdarzenie kompensacji (pośrednie lub końcowe), zazwyczaj uaktywniane przez obsługę błędów, jako część anulowania proceesu lub przez inną kompensację.
- Zdarzenie to uściśla zarazem, które czynności podlegają kompensacji.
- Kompensacja nie jest częścią normalnego przepływu w procesie, zatem jest ona łączona asocjacją, a nie przepływem sterowania.
- W przypadku pętli lub podprocesów typu Multi-Instance, każde z nich ma własny podproces kompensujący (Compensation Event Sub-Process), który ma dostęp do odpowiednich porcji danych pochodzących z przetwarzania tymi elementami.

Konwencje dla kompensacji

Symbol	Kategoria	Opis
	Compensation Task (zadanie kompensacyjne)	Wykonywane, gdy konieczna jest kompensacja czynności. Znacznik kompensacji może być używany na symbolu zadania razem ze znacznikiem iteracji lub zwielokrotnienia.
	Compensation Sub-Process (podproces kompensacyjny)	Wycofywanie efektów czynności wykonanych wcześniej. Znacznik kompensacji może być używany razem z innymi symbolami uściślającymi typ podprocesu.
	Ad-Hoc Compensation (doraźny podproces kompensacyjny)	Przykład połączenia oznaczeń podprocesu (tu: doraźna kompensacja, np. zwrot pieniędzy za niedostarczony do klienta towar).

- Przykład procesu z uwzględnieniem kompensacji:



Związek między obsługą błędów a kompensacją

- Nie można wykonywać kompensacji czynności, które nie zakończyły się poprawnie.
- Jeśli czynność nie zakończyła się poprawnie, wówczas obsługa błędu odpowiada zarazem za zapewnienie takiego efektu, by kompensacja nie była konieczna po zakończeniu obsługi błędu.
- Jeśli dla konkretnego podprocesu nie ma wyspecjalionego Error Event Sub-Process, w przypadku wystąpienia błędu domyślnie wywoływana jest kompensacja dla wszystkich czynności tego podprocesu.

Podprocesy transakcyjne (Transactions)

- Transakcja (Transaction Sub-Process) – zestaw skoordynowanych czynności realizowanych przez niezależny system wykorzystywany w środowisku biznesowym.
- Koordynacja ta ma na celu zapewnienie kompletnego i spójnego wyniku działań dla wszystkich uczestników podprocesu.
- Działanie/przebieg podprocesu transakcyjnego podlega kontroli tzw. protokołu transakcyjnego (transaction protocol).
- Podprocesy transakcyjne działają na zasadzie „wszystko albo nic” – wszystkie czynności składowe muszą zostać wykonane, aby transakcja została zakończona sukcesem.
- Jeśli tak się nie stanie, może być potrzebna kompensacja (np. rollback w bazie danych – czyli wycofanie zmian wprowadzonych w rekordach operacją commit).
- Możliwe są trzy zakończenia transakcji:
 - poprawne,
 - anulowanie,
 - błąd.

Symbol	Nazwa	Opis
	Transaction Sub-Process (podproces transakcyjny)	Oznaczenie podprocesu transakcyjnego (tu: zwinięte, bez widocznych czynności składowych podprocesu). Krawędź rysowana podwójną linią ciągłą.
	End Event (zdarzenie kończące podproces)	Oznacza naturalne i bezbłędne zakończenie podprocesu transakcyjnego. Może mieć swoje uściślenie (np. wysłanie wiadomości).
	Cancel Event (anulowanie podprocesu)	Oznacza anulowanie podprocesu przez co najmniej jednego uczestnika.
	Error Event (błąd podprocesu)	Oznacza wystąpienie błędu przerywającego podproces.

Poprawne zakończenie transakcji

- Następuje, jeśli ścieżka podprocesu osiągnie jakikolwiek naturalny koniec (wyjście z podprocesu), który nie jest związany z anulowaniem (Cancel End Event).
- Jest sygnalizowane zwykłym przepływem sterowania (Sequence Flow) wychodzącym z podprocesu.
- Nieco różni się od normalnego zakończenia innych rodzajów podprocesów:
 - Sterowanie nie powraca natychmiast na poziom procesu nadziednego.

- Najpierw protokół transakcyjny sprawdza, że wszyscy uczestnicy poprawnie zakończyli realizację transakcji.
- Jeśli u choć jednego uczestnika transakcji doszło do pojawienia się błędu lub anulowania, sterowanie trafia do odpowiedniego w takiej sytuacji zdarzenia (Cancel lub Error), nawet jeśli z sytuacji u pozostałych uczestników wynikłoby inne zakończenie podprocesu.

Anulowanie transakcji (Cancel Event)

- Oznacza wymuszenie przerwania transakcji przez co najmniej jednego z uczestników.
- W takiej sytuacji czynności wewnętrz podprocesu są przeznaczone do anulowania – będzie ono obejmować:
 - wycofanie efektów procesu (rolling back),
 - kompensację niektórych czynności.
- Do podprocesu zostanie wówczas włączone zdarzenie anulowania (Boundary Cancel Intermediate Event), z którego wyrowadzony jest przepływ sterowania.
- Wyjście tym przepływem oznacza ogólnie, że:
 - efekty transakcji zostały wycofane (rolled back),
 - zostały zakończone wszystkie niezbędne kompensacje.
- Tylko transakcje mogą podlegać anulowaniu:
 - Cancel Event może być dodawany tylko i wyłącznie do podprocesów transakcyjnych,
 - niedopuszczalne jest jego używanie w szczególności na poziomie normalnego przebiegu procesu oraz włączanie do procesów innych typów.
- Wiadomość anulująca proces może być odebrana tylko za pośrednictwem protokołu transakcyjnego, kontrolującego przebieg podprocesu.

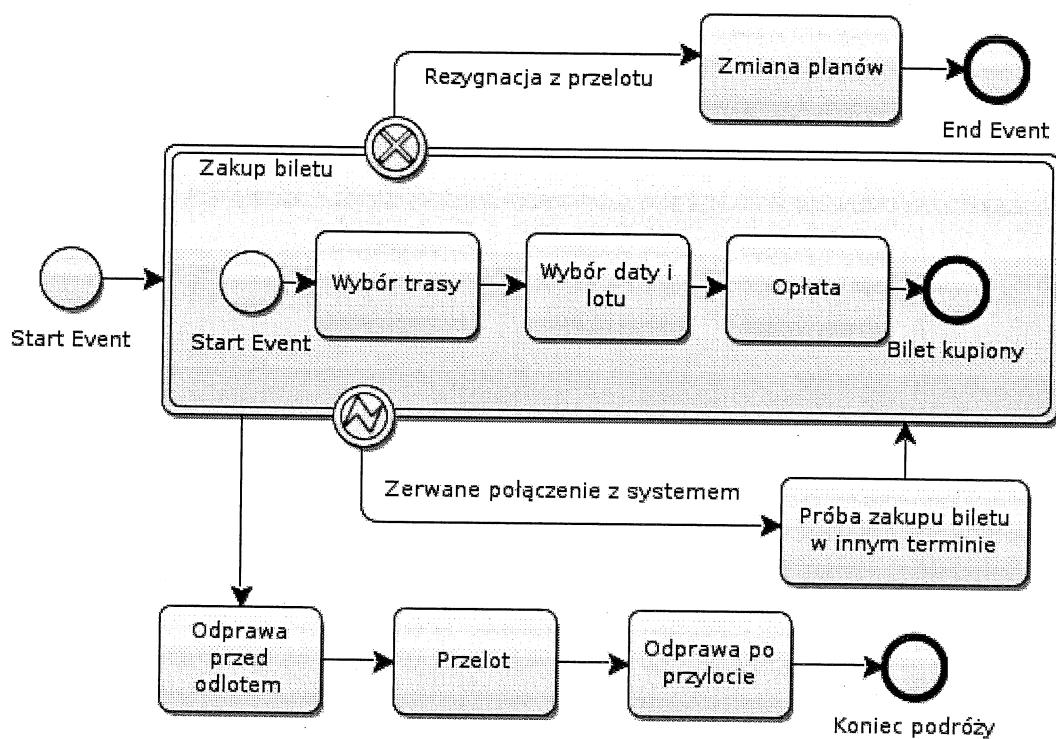
Uwaga

- Żaden inny mechanizm przerywający działanie transakcji (np. Error Event czy Timer Event) nie wywołuje kompensacji.

Błąd transakcji (Error Event)

- Oznacza, że coś poszło na tyle źle, iż:
 - niemożliwe jest skuteczne zakończenie transakcji czy też jej anulowanie,
 - podproces zostaje przerwany bez kompensacji.
- Takie sytuacje są oznaczane za pomocą Error Intermediate Events włączanych do podprocesu oraz wychodzących z nich przepływów sterowania.

- Przykład procesu z uwzględnieniem podprocesu transakcyjnego:



Interakcje między uczestnikami procesu

- Podstawowym poziomem opisu procesów w BPMN są diagramy partytury (Orchestration), które pokazują szczegóły procesów (opisane w poprzednich tematach szkolenia).
- Czasem potrzeba jednak pokazać proces z innych punktów widzenia:
 - współpracy między uczestnikami procesu (Collaboration),
 - uproszczonej współpracy uczestników procesu (Participants), niejako ich „rozmowę” (Conversation).

Schemat współpracy w procesie (Collaboration)

- Czynności w procesach są realizowane przez uczestników (Participants).
- Participant – jednostka biznesowa (np. firma, oddział firmy czy klient) lub rola biznesowa (np. sprzedawca lub nabywca), która realizuje lub odpowiada za proces biznesowy.
- Jeśli zachodzi potrzeba pokazania ich współpracy, korzysta się z diagramu Collaboration, na którym zobrazowany jest podział odpowiedzialności uczestników procesu za poszczególne czynności.
- Do tego celu służą przede wszystkim baseny (Pools) oraz tory (Lanes), określane zbiorczo jako Swimlanes.

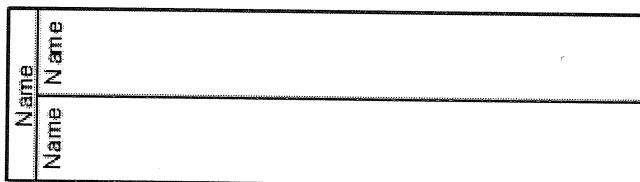
Baseny (Pools)

- Basen (Pool):
 - graficzna reprezentacja uczestnika procesu (Participant),
 - wydzielony poziomy lub pionowy obszar diagramu,
 - może (ale nie musi) zawierać czynności wykonywane przez uczestnika w ramach procesu oraz zdarzenia związane z jego działaniami (wynikające z tych działań lub na te działania wpływające).
- Basen może obrazować np.:
 - klienta,
 - pracownika (kierownik, księgowy, monter),
 - rolę (lider, koordynator, sprzedawca, nabywca),
 - dział/wydział/oddział firmy (zarząd, księgowość, produkcja, marketing, sprzedaż),
 - system informatyczny (finansowo-księgowy, logistyczny),
 - maszynę/urządzenie (obrabiarka, robot przemysłowy),
 - całą firmę lub instytucję (dostawca, producent, biblioteka),
 - czasem grupę czynności (montaż, obsługa klienta, analiza itp.), jednakże do tego celu lepiej używać grup (Groups).

- Nazwa basenu musi być oddzielona od jego zawartości kreską/krawędzią.
- Jeśli proces obejmuje działania tylko jednego uczestnika, basen nie musi być pokazywany.
- Baseny mogą być rozmieszczane na diagramie w dowolnej ilości i dowolny sposób, który odpowiada potrzebom osób tworzących i korzystających z diagramu.

Tory (Lanes)

- Tor (Lane) – to część składowa basenu.
- Tory są używane, gdy trzeba zwiększyć szczegółowość podziału organizacyjnego w realizacji procesu.
- Np. dział księgowości (basen), w którym wyszczególnione są tory odpowiadające głównemu księgowemu i kasjerowi, albo dział obsługi klienta z torami kierownika, sprzedawcy i serwisanta.
- Ilość torów również zależy od specyfiki opisywanej sytuacji oraz potrzeb twórców i odbiorców diagramu.
- Nazwa toru może znajdować się w dowolnym miejscu toru, ale nie może być oddzielona krawędzią/kreską od zawartości (czyli czynności umieszczonej w obrębie toru).
- Przykład basenu z torami (bez czynności) pokazuje poniższa ilustracja:



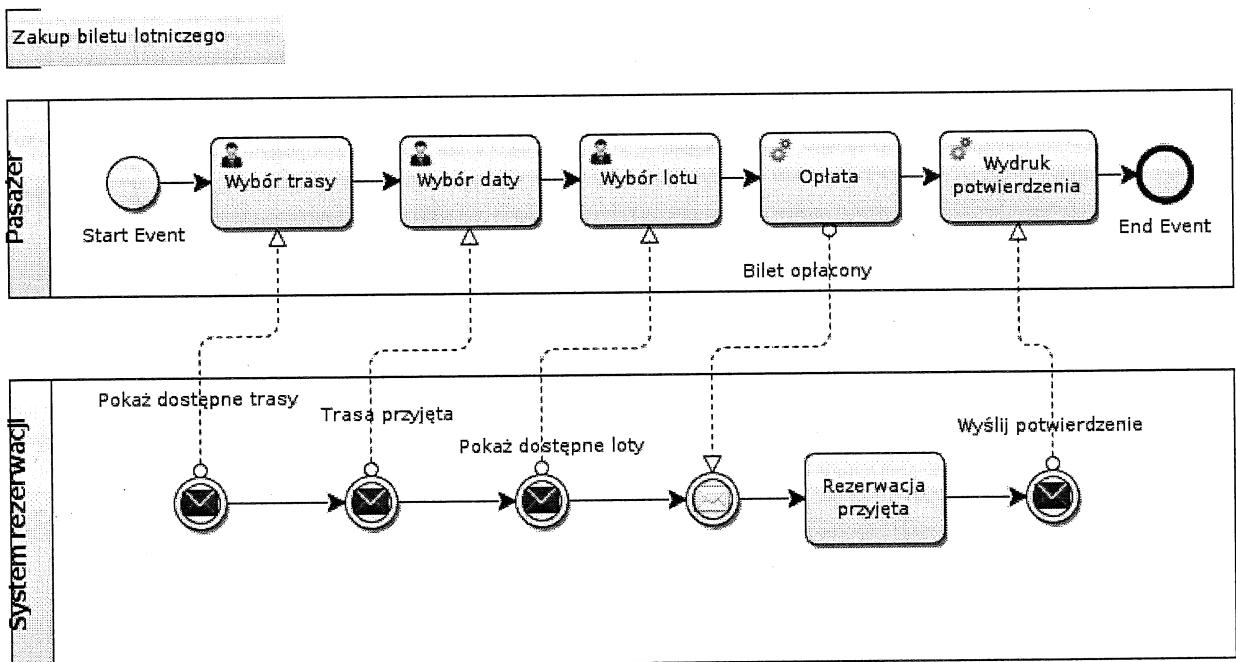
Komunikacja między uczestnikami procesu

- Na diagramie współpracy (Collaboration) obrazowana jest komunikacja między obiektami (czyli uczestnikami procesu).
- Używane są tutaj przepływy wiadomości (Message Flows), które mogą łączyć ze sobą wyłącznie dwa różne baseny (Pools) lub też wybrane elementy składowe basenów. Nie mogą natomiast łączyć elementów składowych w ramach tego samego basenu (do tego celu służą przepływy sterowania – Sequence Flows).
- Podstawowe oznaczenie przepływu wiadomości pokazuje poniższa ilustracja:



- Przepływy wiadomości są dopuszczalne między następującymi elementami diagramu:
 - basenami (Pools),
 - zadaniami (Tasks),
 - podprocesami (Sub-Processes),

- zdarzeniami pośrednimi odebrania i wysłania wiadomości (Message Intermediate Events – catch i throw).
- Każdy z wymienionych wyżej elementów może zarówno wysyłać (być początkiem), jak i odbierać (być końcem przepływu) wiadomości, z wyjątkiem zdarzenia pośredniego typu throw, które może być jedynie początkiem, a nigdy końcem przepływu wiadomości.
- Przykład diagramu współpracy z uwzględnieniem komunikacji między uczestnikami:



Schemat koordynacji działań uczestników procesu (Choreography)

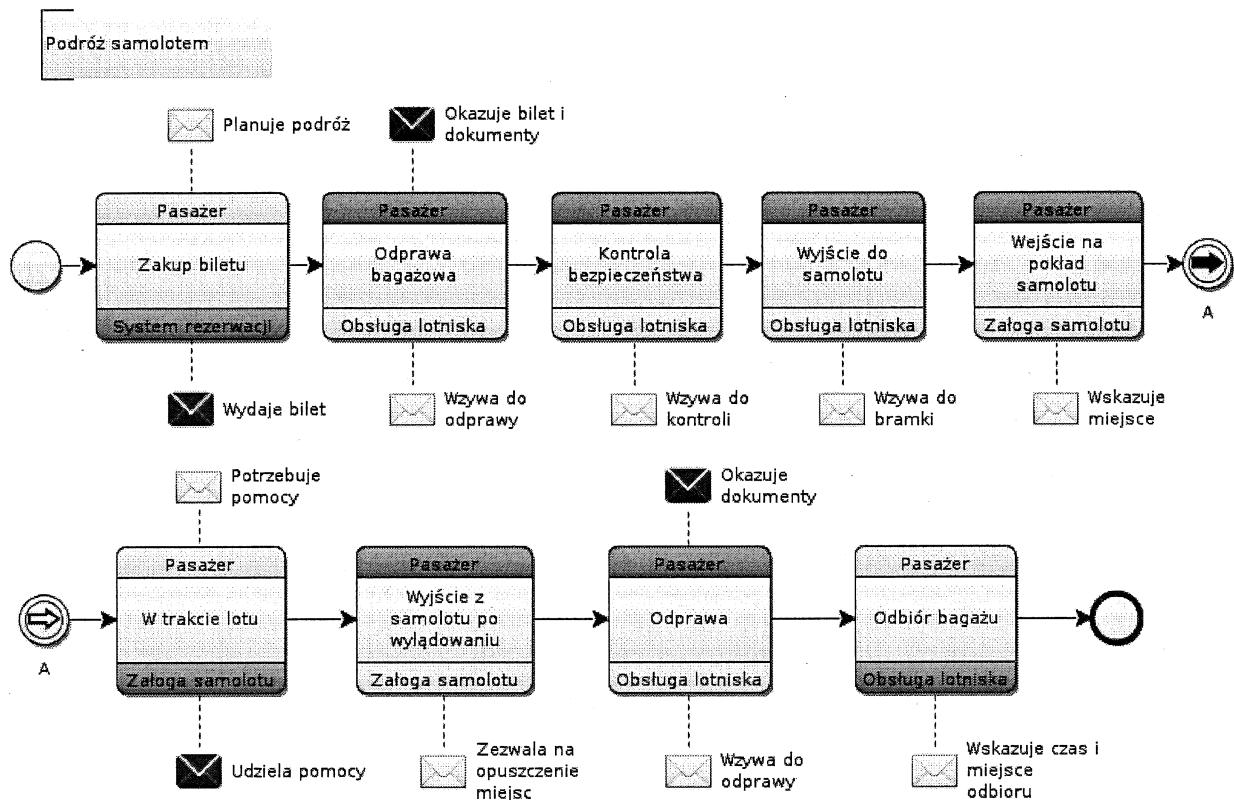
- Choreografia – inny (niż partytura) sposób spojrzenia na proces:
 - Ukazuje koordynację działań uczestników procesu (Participants).
 - Koncentruje się raczej na wymianie informacji (Messages) między uczestnikami, niż na konkretnych działań poszczególnych uczestników procesu.
- Choreografie można też potraktować jako rodzaj (formę opisu) umów biznesowych między co najmniej dwiema firmami lub organizacjami.
- Choreografia jest zatem opisem oczekiwanych zachowań, przede wszystkim w postaci proceduralnych umów biznesowych, pomiędzy wchodzący w interakcję uczestnikami.
- Daje możliwość pokazania:
 - wymiany informacji między uczestnikami procesu,
 - logicznych związków między uczestnikami procesu (przedstawianych kiedy indziej jako konwersacje).
- Choreografia umożliwia współpracującym stronom zaplanowanie ich procesów biznesowych do bezkonfliktowego współdziałania – sekwencja wymienianych

wiadomości potrzebuje pogłębionego spojrzenia w postaci partytur (Orchestrations) procesów biznesowych poszczególnych uczestników procesu/współdziałania.

- Model wyrażony w kontekście choreografii umożliwia też wyszczególnienie interfejsów dla procesów poszczególnych partnerów.
- Podstawową jednostką choreografii jest zadanie (Choreography Task), czyli MEP (Message exchange pattern = 'wzorzec wymiany wiadomości') – pełni podobną rolę, jak czynność w partyturach.
- Choreography Task składa się z jednej lub więcej interakcji między współpracującymi ze sobą uczestnikami procesu.
- Choreography Tasks są związane ze sobą za pomocą przepływów sterowania (Sequence Flows) i w ten sposób układają się w uporządkowane sekwencje (czyli choreografie).
- W ramach procesu opisywanego w postaci choreografii można też uściślić rodzaje zdarzeń początkowych i końcowych oraz stosować bramki (Gateways) – szczegóły zob. specyfikacja BPMN, rozdział 11 *Choreography*.

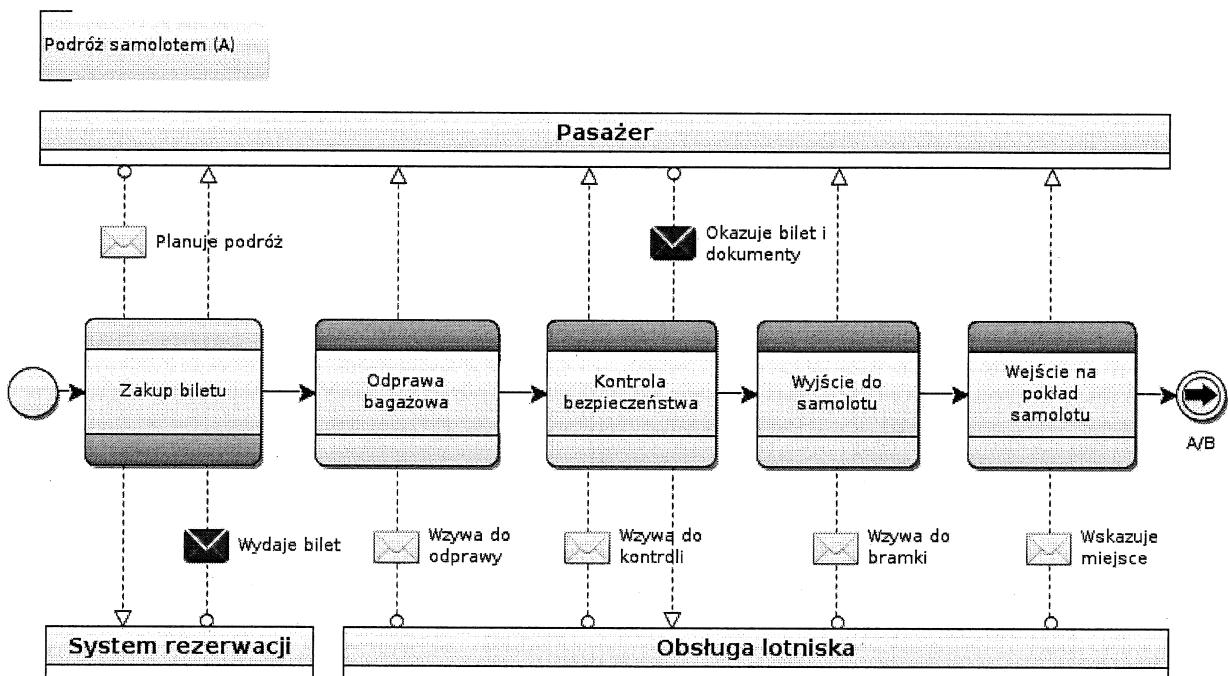
Symbol	Nazwa	Opis
	Start Event (zdarzenie początkowe)	Oznaczenie początku choreografii.
	End Event (zdarzenie końcowe)	Oznaczenie końca choreografii.
	Choreography Task (zadanie)	Podstawowy element składowy choreografii – pojedynczy zestaw interakcji. Górny element Participant Band – oznacza jednego z uczestników procesu biorącego udział w interakcji (może być ich więcej). Jaśniejsze tło oznacza uczestnika inicjującego interakcję. Dolny element Participant Band oznacza drugiego z uczestników (również może ich być więcej). Ciemniejsze tło oznacza pozostałych, nie-inicjujących, uczestników interakcji.
	Message – initiation (wiadomość inicjująca)	Wiadomość, która rozpoczyna interakcję w ramach Choreography Task. Oznaczona jako kontur bez wypełnienia (biały/jasny).
	Message – response (wiadomość nieinicjująca)	Wiadomość, która stanowi odpowiedź w ramach konkretnej interakcji. Oznaczona ciemnym wypełnieniem. Nie musi wystąpić (nie każda wiadomość inicjująca musi mieć oznaczaną odpowiedź).

- Przykład procesu ukazanego za pośrednictwem choreografii:



- Choreografie mogą być włączane do diagramu współpracy (Collaboration).
- W takiej sytuacji:
 - Nie wymagają podawania nazw uczestników na wstęgach nazw (Participant Bands) – nazwy uczestników są bowiem widoczne na basenach (Pools).
 - Basen (jeden lub więcej) może występować w roli czarnej skrzynki (black box), czyli nie zawiera szczegółowego rozpisu czynności/procesów przebiegających wewnątrz basenu – będzie natomiast wysyłać i odbierać wiadomości (Message Flows).

- Przykład (fragmentu początkowego) procesu z choreografią umieszczoną na diagramie współpracy (Collaboration):



Poglądowy schemat komunikacji w procesie (Conversation)

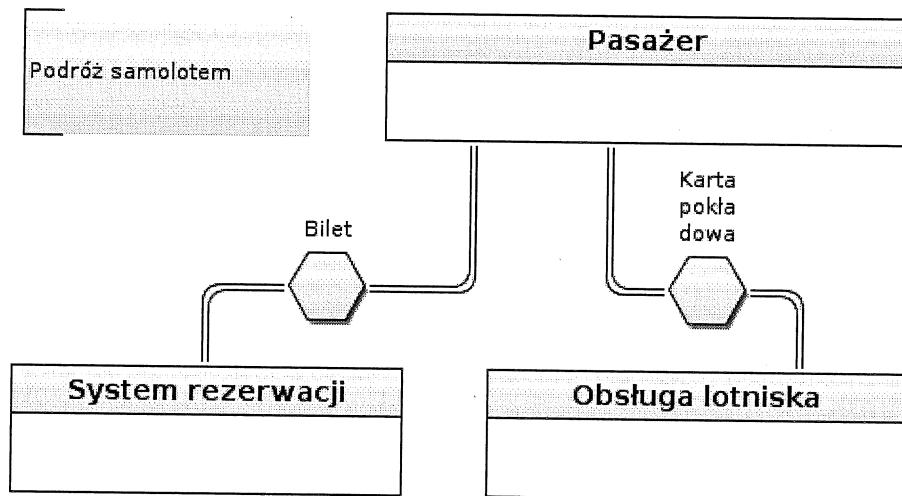
- Aspekt współpracy można pokazać też pod nieco innym kątem – uproszczony do schematu komunikacji między uczestnikami procesu opartego o konwersacje.
- Służy do tego diagram konwersacji (Conversations).
- Konwersacja to logiczne grupowanie wymiany wiadomości (Message Flows).
- W praktyce często odnosi się do obiektów biznesowych takich jak zamówienie, dostawa czy zlecenie.
- Wymiana wiadomości odzwierciedla realizację różnych scenariuszy biznesowych. Może być:
 - prosta – obejmuje tylko wiadomość i reakcję na nią,
 - złożona – nie tylko dwustronna, ale nawet wielostronna.
- Np. w logistyce uzupełnienie stanów magazynowych wymaga realizacji następujących scenariuszy:
 - przygotowania zamówienia,
 - zorganizowania środków transportu z uwzględnieniem różnych zamówień (aby jak najlepiej wykorzystać dostępne środki),
 - przebycia odpraw celnych,
 - uregulowania należności,

- kontroli jakości dostawy.
- Konwersacja jest bardzo istotna z punktu widzenia choreografii, ale tam nie jest widoczna.
- Różnica polega na tym, że:
 - choreografia dostarcza wielokierunkowego spojrzenia na komunikację – modelowana w choreografii wymiana wiadomości dotyczy wielu uczestników procesu,
 - partytura (Orchestration) procesu dotyczy pojedynczego uczestnika.
- Niezależnie od różnicy ujęcia, koncepcja konwersacji pozostaje identyczna w choreografii i partyturze, zaś wymiana wiadomości pokazana w konwersacji będzie ostatecznie zrealizowana w partyturze procesu.
- Różnica między diagramem współpracy a konwersacją:
 - diagram Collaboration – spojrzenie dedukcyjne (top-down, od ogółu do szczegółu), z uwzględnieniem wymiany wiadomości i konwersacji z perspektywy czasu,
 - diagram Conversation – pełny obraz wszystkich konwersacji odnoszących się do modelowanej domeny jest dostępny właśnie na diagramie konwersacji.
- Inaczej mówiąc – diagram konwersacji daje obraz „z lotu ptaka” dla różnych konwersacji w odniesieniu do domeny.
- Konwersacje można też prezentować bardziej szczegółowo – przede wszystkim w mniejszych układach (obejmujących pary komunikujących się obiektów) z ich rozbiciem na osobne diagramy i z dodatkowym wyszczególnieniem wiadomości przepływających w ramach pojedynczej konwersacji.

Symbol	Nazwa	Opis
	Conversation (konwersacja)	Obrazuje zestaw przepływów wiadomości zgrupowanych wokół wspólnego pojęcia i/lub identyfikatora (np. „Zamówienie” i/lub „Nr Zamówienia”). Może wiązać ze sobą dwóch lub więcej uczestników procesu.
	Conversation Link (połączenie konwersacji)	Połączenie konwersacji z uczestnikami procesu, którzy w danej konwersacji biorą udział (tu połączenie pokazane razem z konwersacją, która ma swoją nazwę i/lub identyfikator).
	Sub-Conversation (konwersacja uszczegółowiona)	Konwersacja, która ma swoje uszczegółowienie – po rozwinięciu można zobaczyć przede wszystkim rozpis takiej konwersacji na poszczególne wiadomości.

Symbol	Nazwa	Opis
	Call Global Conversation (przywołanie konwersacji globalnej)	Przywołanie konwersacji globalnej, czyli szczegółowej definicji komunikacji przygotowanej na poziomie ogólnym modelu i przeznaczonej do jej wykorzystania w różnych miejscach diagramu. Konwersacja globalna nie może zawierać konwersacji zwykłych i uszczegółowionych ani dalszych wywołań konwersacji globalnych.
	Call Collaboration (przywołanie schematu współpracy)	Przywołanie schematu współpracy.

- Przykład diagramu konwersacji:



Bibliografia

- Specyfikacja BPMN – Business Process Model and Notation (BPMN), Version 2.0.1, OMG, September 2013.

