

Szymon  
Drejewicz

# Zrozumieć **BPMN**

## MODELOWANIE PROCESÓW BIZNESOWYCH

ZOBACZ NA PRZYKŁADACH

- AKTYWNOŚCI
- ZDARZENIA
- PRZEPŁYWY
- BRAMKI
- OBIEKTY
- UCZESTNIKÓW
- PODPROCESY, TRANSAKcje I KOMPENSACJE
- KOLABORACJE
- CHOREOGRAFIE
- KONWERSACJE

Wszelkie prawa zastrzeżone. Nieautoryzowane rozpowszechnianie całości lub fragmentu niniejszej publikacji w jakiegokolwiek postaci jest zabronione. Wykonywanie kopii metodą kserograficzną, fotograficzną, a także kopiowanie książki na nośniku filmowym, magnetycznym lub innym powoduje naruszenie praw autorskich niniejszej publikacji.

Wszystkie znaki występujące w tekście są zastrzeżonymi znakami firmowymi bądź towarowymi ich właścicieli.

Autor oraz Wydawnictwo HELION dołożyli wszelkich starań, by zawarte w tej książce informacje były kompletne i rzetelne. Nie biorą jednak żadnej odpowiedzialności ani za ich wykorzystanie, ani za związane z tym ewentualne naruszenie praw patentowych lub autorskich. Autor oraz Wydawnictwo HELION nie ponoszą również żadnej odpowiedzialności za ewentualne szkody wynikłe z wykorzystania informacji zawartych w książce.

Redaktor prowadzący: Barbara Gancarz-Wójcicka  
Projekt okładki: Jan Paluch

Fotografia na okładce została wykorzystana za zgodą Shutterstock.

Wydawnictwo HELION  
ul. Kościuszki 1c, 44-100 GLIWICE  
tel. 32 231 22 19, 32 230 98 63  
e-mail: [onepress@onepress.pl](mailto:onepress@onepress.pl)  
WWW: <http://onepress.pl> (księgarnia internetowa, katalog książek)

Drogi Czytelniku!  
Jeżeli chcesz ocenić tę książkę, zajrzyj pod adres  
[http://onepress.pl/user/opinie/zrobpm\\_p](http://onepress.pl/user/opinie/zrobpm_p)  
Możesz tam wpisać swoje uwagi, spostrzeżenia, recenzję.

ISBN: 978-83-246-4242-7  
Numer katalogowy: 7807

Copyright © Helion 2012

- [Poleć książkę na Facebook.com](#)
- [Kup w wersji papierowej](#)
- [Oceń książkę](#)

- [Księgarnia internetowa](#)
- [Lubię to! » Nasza społeczność](#)

---

# Spis treści

---

<b>Wstęp</b>	<b>5</b>
<b>1. Wprowadzenie do BPMN</b>	
1.1. Proces i model procesu	9
1.2. Dlaczego notacja BPMN?	10
1.3. Poziomy modelowania	12
<b>2. Szybki start — elementarz</b>	<b>15</b>
2.1. Pierwszy krok	15
2.2. Pule i tory	16
<b>3. Bramki — dzielenie i łączenie procesów</b>	<b>19</b>
3.1. Bramki w BPMN	20
3.2. Bramka wykluczająca	23
3.3. Bramka niewykluczająca	24
3.4. Bramka równoległa	27
3.5. Bramka złożona	29
3.6. Bramka oparta na zdarzeniach	32
<b>4. Dzielenie i łączenie bez bramek</b>	<b>35</b>
4.1. Rozgałęzienia bez bramek	35
4.2. Złączenia bez bramek	37
4.3. Kiedy pomijać bramki?	38
<b>5. Przepływy</b>	<b>41</b>
5.1. Przepływ sekwencji	43
5.2. Przepływ komunikatu	50
5.3. Asocjacje	51
<b>6. Obiekty danych</b>	<b>55</b>
6.1. Modelowanie obiektów i magazynów danych	56
6.2. Przepływ obiektów w procesach	57
6.3. Wejścia i wyjścia danych	59
6.4. Kolekcje danych	61

<b>7. Kolaboracje i konwersacje</b>	<b>65</b>
7.1. Modelowanie kolaboracji	65
7.2. Procesy prywatne i publiczne	68
7.3. Wieloinstancyjni uczestnicy procesów biznesowych	71
7.4. Konwersacje	73
<b>8. Zdarzenia</b>	<b>77</b>
8.1. Typy zdarzeń	78
8.2. Jak to wszystko opanować?	96
<b>9. Czynności i podprocesy</b>	<b>99</b>
9.1. Zadania	101
9.2. Podprocesy	110
9.3. Pętle, czynności wieloinstancyjne i kompensacje	119
9.4. Czynność wywołania (call activity)	121
<b>10. Choreografie</b>	<b>123</b>
10.1. Zadanie choreografii	124
10.2. Choreografia złożona	127
10.3. Kolaboracje, procesy i choreografie	131
10.4. Kiedy używać diagramów choreografii?	131
<b>Inne źródła</b>	<b>133</b>

---

# Wstęp

---

Standard BPMN (*Business Process Model and Notation*), nazywany często „notacją”, jest obecnie najpopularniejszym narzędziem do specyfikowania procesów biznesowych. Obserwując trendy w systemach klasy BPMS (*Business Process Management System*), można przypuszczać, że BPMN całkowicie zdominuje rynek i ostatecznie wyprze inne mniej lub bardziej udane standardy podobnego typu (np. EPC – *Event-driven Process Chain*). Jest to standard, który z jednej strony umożliwia opisywanie procesów w sposób zrozumiały dla zwykłych śmiertelników, a z drugiej pozwala na bardzo dokładne opisywanie ich na poziomie technicznym. Ta cecha wydaje się bardzo pożądana, gdyż coraz większa liczba organizacji na świecie staje przed podstawowym problemem: jak wykorzystać potencjał posiadanych systemów informatycznych, zasobów organizacji i ludzi, którzy ją tworzą? Istotne jest więc, jak powiązać te wszystkie elementy, aby zdobyć przewagę konkurencyjną na rynku. Jeszcze 10 lat temu problemem było to, czy i jakie systemy informatyczne należy wdrożyć, żeby nie zostać w tyle. Dzisiaj już nikt nie wątpi, że bez rozwiązań informatycznych trudno świadczyć usługi wysokiej jakości. Wszyscy wiemy, że w wielu aspektach (komunikacja, medycyna, energetyka, zarządzanie, nauka, przemysł, telekomunikacja itd.) komputery ze swoim oprogramowaniem są niezastąpione i doskonale uzupełniają niedoskonałości ludzi. Wygrywa więc nie ten, kto ma szybkie komputery i odpowiednie oprogramowanie, ale ten, kto potrafi optymalnie wykorzystać ten potencjał, a także idealnie dopasować posiadane dobra do charakteru organizacji i ludzi, którzy ją tworzą. Co więcej, już dziś wiemy, że wcale nie chodzi o to, żeby dojść do jakiegoś „stanu perfekcji”, ale o to, żeby stale dostosowywać się do niezwykle dynamicznego otoczenia. Potrzebujemy więc narzędzia, które pozwoli nam na ciągłe analizowanie, monitorowanie i optymalizowanie naszych procesów biznesowych. Standard BPMN wydaje się zatem rozsądnym wyborem, bo z założenia opracowano go tak, aby był czytelny dla wszystkich uczestników tej „zabawy”.

## Do kogo adresowana jest książka?

Niniejszą książkę polecam tym, którzy chcą lub muszą szybko poznać i zrozumieć standard BPMN, a jednocześnie nie są w stanie „strawić” specyfikacji opracowanej przez OMG. Polecam ją także analitykom biznesowym i systemowym, którzy chcą poszerzyć swoją wiedzę, a jeszcze nie poznali żadnego standardu do specyfikowania procesów biznesowych. Polecam ją również studentom kierunków zarządczych i technicznych — Wasze drogi, kochani, zejdą się szybciej, niż myślicie, i to właśnie za sprawą procesów biznesowych...

Nie polecam wykorzystywania niniejszej książki jako materiału referencyjnego, choć maksimum uwagi poświęciłem zachowaniu zgodności z oryginalną specyfikacją OMG BPMN 2.0. W wielu miejscach celowo pomijam niuanse i detale techniczne, aby zachować przejrzystość. Tak więc Czytelnik zapewne nie znajdzie w książce zbyt wielu błędów, ale nie może też oczekiwać, że wyczerpuje ona temat BPMN w pełni. Jako materiał referencyjny zdecydowanie polecam tylko i wyłącznie oryginalną specyfikację, dostępną na stronie *www.omg.org*.

## Układ książki

Treść każdego rozdziału, poza rozdziałami wprowadzającymi, koncentruje się na wybranej grupie elementów notacji BPMN 2.0. Czytelnik znajdzie tu zarówno wyjaśnienie przeznaczenia danej grupy elementów, ich graficzną reprezentację w modelach BPMN 2.0, jak i przykładowe modele, w których zastosowano omawiane elementy. Przykłady zostały przygotowane tak, aby eksponowały istotne aspekty omawianego zagadnienia, a jednocześnie były czytelne i zrozumiałe dla każdego.

Rozdział 1. zawiera krótkie i bardzo ogólne wprowadzenie do standardu BPMN 2.0, a także do pojęć procesu biznesowego i modelu procesu biznesowego. Założeniem tego rozdziału nie było jednak kopiowanie dostępnego w wielu miejscach elementarza w rodzaju „Co to jest proces biznesowy?”, a jedynie wyjaśnienie tego, co jest szczególnie istotne w kontekście omawiania procesów biznesowych w standardzie BPMN 2.0.

Rozdział 2. zawiera trzy proste modele w notacji BPMN 2.0 i jest swego rodzaju krótką prezentacją tego, na czym polega modelowanie procesów biznesowych. Jeśli stoisz teraz w księgarni i zastanawiasz się, czy ta książka jest dla Ciebie, a jednocześnie nie masz pewności, co to jest BPMN, zajrzyj do tego rozdziału.

Rozdział 3. wyjaśnia jeden z najistotniejszych elementów w notacji BPMN, czyli bramki. Każdy rodzaj bramki jest wyjaśniony i opatrzony przykładem. Naukę BPMN warto rozpoczynać od bramek, ponieważ to one sprawiają najwięcej kłopotów, a jednocześnie występują praktycznie w każdym modelu procesu biznesowego.

Rozdział 4. jest kontynuacją poprzedniego rozdziału, gdyż wyjaśnia, jak należy interpretować modele procesów biznesowych bez bramek. Głównym celem tego rozdziału jest zwrócenie uwagi Czytelnika na fakt, że bramki w notacji BPMN mogą czasem być pomijane i dlatego warto mieć pełną świadomość, jak interpretować takie modele.

Rozdział 5. omawia przepływy, czyli, mówiąc prościej, krawędzie czy też potocznie „strzałki”, które łączą pozostałe elementy w notacji BPMN 2.0. Przepływy znalazły się na drugim miejscu, po bramkach, z tego względu, że są także bardzo istotne, a jednocześnie bardzo często umykają uwadze adeptów BPMN — do tego stopnia, że często nie zauważają oni, iż występuje kilka różniących się graficznie i znaczeniowo przepływów.

Rozdział 6. wyjaśnia, jak modelować obiekty przetwarzane w procesach biznesowych oraz jakie rodzaje owych obiektów mogą pojawiać się na modelach i jakie jest ich znaczenie w praktyce.

Rozdział 7. zawiera szereg przykładów niezbędnych do zrozumienia, czym są kolaboracje i konwersacje w standardzie BPMN. Dzięki temu rozdziałowi Czytelnik może się przekonać, że te elementy, choć mają dziwne nazwy, są bardzo praktyczne i przydatne, ponieważ pozwalają na upraszczanie modeli.

Rozdział 8. wyjaśnia zdarzenia BPMN, czyli najtrudniejszy składnik tego standardu. Trudność w tym przypadku polega na tym, że rodzajów zdarzeń jest bardzo wiele (kilkadziesiąt!), a jednocześnie ich zastosowanie często jest skomplikowane i wymaga zrozumienia pozostałych elementów.

Rozdział 9. omawia najbardziej rozbudowaną i kluczową grupę elementów standardu BPMN, czyli czynności. Z tego rozdziału Czytelnik dowie się, jakie są rodzaje zadań i jakie jest ich znaczenie w rzeczywistych procesach biznesowych, co to są podprocesy (w tym transakcje i kompensacje), a także jak modelować pętle i czynności wielokrotne w BPMN 2.0.

Rozdział 10. przybliży tzw. choreografie, czyli modele, których celem jest ekspozycja interakcji między uczestnikami procesów biznesowych. Choreografie są ciekawym rozwiązaniem, aczkolwiek są mało popularne.

## Narzędzie do modelowania w BPMN

Do opracowania diagramów w tej książce użyto dodatku BPMN 2.0 Modeler for Visio firmy Trisotech, który wymaga wcześniejszego zainstalowania aplikacji Microsoft Visio 2007/2010. Dostępne są także samodzielne aplikacje, które nie wymagają niczego poza systemem operacyjnym. W chwili przygotowywania tej książki na rynku dostępnych było kilkanaście darmowych i w pełni funkcjonalnych narzędzi do modelowania w notacji BPMN 2.0.

## Podziękowania

Wielu osobom podziękowania w książce mogą wydawać się mało istotne, lecz tak naprawdę jest to jedyne miejsce, gdzie można pokazać i docenić wkład innych osób, które pracowały nad książką, często nawet nie mając świadomości, że bez nich w ogóle nie mogłaby ona powstać. No więc do rzeczy. Dziękuję:

Mojej żonie za to, że przez wiele miesięcy powstrzymywała dwie małe i ukochane istoty, czyli nasze córki, przed wtargnięciem do pokoju, w którym pracowałem nad książką. Dzieci są bezlitosne, klawisz *Delete* jest po prostu jednym z wielu, które się fajnie naciska, notatki i rysunki są dobre, bo można je kolorować lub ciąć na kawałki, a tata piszący coś na komputerze wygląda tak, jakby nic nie robił, i trzeba go zaangażować w różne zabawy ;-)

Firmie Infovide-Matrix SA za możliwość uczestniczenia w wielu projektach mniej lub bardziej związanych z BPM. Szczególne podziękowania dla Michała Pałuśkiewicza, Krzysztofa Dymickiego i Sebastiana Liska.

Profesorowi Michałowi Śmiałkowi z Wydziału Elektrycznego na Politechnice Warszawskiej, gdzie pracuję jako jego asystent, za to, że przez kolejny rok wykazał się cierpliwością w obliczu mojej opieszałości w pracy naukowej. Pisanie książki jednak pochłonęło dużo mojego czasu. Mam nadzieję, że uda mi się jeszcze opublikować wiele artykułów naukowych.

Uczestnikom dyskusji na temat niniejszej książki na forum dyskusyjnym [goldenline.pl](http://goldenline.pl). Wasze opinie były bardzo cenne. Choć nie zawsze się z nimi zgadzałem, to dzięki nim nabrałem przekonania do niektórych zagadnień. Szczególne podziękowania dla Jarosława Żelińskiego, Piotra Tadeusza Biernackiego i Zbigniewa Misiaka — za Wasze uwagi na forum i bezpośrednio na moją skrzynkę pocztową. Mam nadzieję, że następne książki będziemy pisali razem. Co Wy na to?

Wydawnictwu Helion SA, a w szczególności pani redaktor Barbarze Gancarz-Wójcickiej za pomoc w przygotowywaniu książki.



BPMN to standard opracowany przez organizację Object Management Group (OMG). Jego pełna nazwa to *Business Process Model and Notation*. Podstawowym celem tego standardu jest dostarczenie notacji do opisywania procesów biznesowych, która jest czytelna i zrozumiała zarówno dla biznesowych „użytkowników”, monitorujących procesy i zarządzających nimi, dla analityków, którzy przeprowadzają biznesową analizę procesów, jak i programistów, odpowiedzialnych za ich techniczną implementację. Finalna i stabilna (nie „beta”) specyfikacja standardu BPMN 2.0 została opublikowana przez OMG w styczniu 2011 na stronie [www.omg.org](http://www.omg.org). Specyfikacja ta jest otwarta i dostępna nieodpłatnie dla wszystkich zainteresowanych. Obecnie jest to powszechnie stosowany międzynarodowy standard opisywania procesów biznesowych, uznany przez wiele komercyjnych i niekomercyjnych organizacji na całym świecie, także w Polsce.

## 1.1. Proces i model procesu

Jest kilka powodów, dla których warto i trzeba modelować procesy. Po pierwsze, chcemy je rozumieć. Często jest tak, że dopóki we wspólnym gronie nie ustalą się dokładnie, jak „wygląda” dany proces biznesowy, trudno np. dojść do porozumienia, kto i co powinien zmienić, żeby wdrożyć jakąś zmianę biznesową. Po drugie, procesy biznesowe współcześnie definiuje się w wielu przypadkach po to, aby je zautomatyzować, a dokładniej: wdrożyć na silniku procesów biznesowych. Jednak techniczne wdrażanie i uruchamianie procesu biznesowego nie jest czynnością łatwą i często pochłania wiele dni pracy całych zespołów analityków i programistów. Można więc powiedzieć, że wdrażanie procesów biznesowych jest kosztowne. Jednocześnie trudno jest akceptować te koszty bez jakiegokolwiek metody ich szacowania a priori. A więc trzecim powodem, dla którego warto modelować procesy, jest możliwość estymacji pracochłonności związanej czy to ze zmianami w nich, czy też z ich automatyzacją.

Warto też wyjaśnić, czym jest model procesu oraz czym jest sam proces biznesowy. Powszechnie mówi się, że proces biznesowy to sekwencja uporządkowanych czynności, których celem jest wytworzenie jakiegoś dobra (informacji, towaru, usługi itd.). Zgodnie z definicją specyfikacji OMG BPMN 2.0 w wersji ze stycznia 2011 *proces biznesowy* to sekwencja lub przepływ czynności w jakiejś organizacji, których celem jest wykonanie jakiejś pracy. Tak więc definicja OMG jest nieco szersza, bo procesem jest nie tylko sekwencja czynności, ale także przepływ czynności. O co chodzi? Bez zbędnego rozwlekania tematu — chodzi o to, że procesem biznesowym może być kilka czynności, które są wykonywane, ale ich porządek wykonania nie musi być zawsze taki sam. Dla wielu osób może być dość zaskakujące, że nie tylko uporządkowana sekwencja czynności jest procesem biznesowym, jednak po chwili zastanowienia można się przekonać, że to dobra definicja. Jeśli ktoś nie czuje się przekonany, może na chwilę przeskoczyć do fragmentu o podprocesach w rozdziale 9.2.3 i zobaczyć, że faktycznie takie nieuporządkowane procesy także istnieją i wcale nie są rzadkością. Dodatkowo definicja OMG nie odnosi się do „produktu” procesu, a jedynie stwierdza, że jest to jakaś „praca”. Nie ma więc ograniczenia, że proces musi coś wytwarzać. W tym przypadku wystarczy zauważyć, że we wszystkich znanych organizacjach realizuje się szereg prac, które nie zawsze przynoszą jakiegokolwiek efekty. A więc czy tego chcemy, czy nie, nie wszystkie zadania, które realizujemy, mają racjonalne czy też biznesowe uzasadnienie, co nie zmienia faktu, że wykonujemy te prace.

## 1.2. Dlaczego notacja BPMN?

Notacji do modelowania procesów biznesowych jest kilka. Najważniejsze i najpopularniejsze z nich to: BPMN, EPC i BPEL. Porównując te notacje, można zauważyć kilka istotnych faktów, które mogą pomóc w podjęciu decyzji, jaką notację najlepiej zastosować w danym kontekście. EPC (ang. *Event-driven Process Chain*) jest notacją dużo starszą od BPMN i stanowi część tzw. metody ARIS (dzisiaj własność Software AG). W gruncie rzeczy porównanie BPMN i EPC można sprowadzić do stwierdzenia, że wszystko, co można opisać w notacji EPC, można także opisać w notacji BPMN. To, co odróżnia BPMN od EPC, to fakt, że BPMN posiada zdefiniowany formalnie metamodel. Posiadając więc narzędzia do transformacji modeli, takie jak np. MOLA<sup>1</sup>, można próbować transformować modele BPMN na inne modele (notacje) lub np. na gotowe fragmenty oprogramowania. W przypadku notacji EPC konieczne byłoby, poza opracowaniem samej transformacji, uprzednie zdefiniowanie jej metamodelu. Nie jest to niemożliwe (istnieją już takie nieoficjalne metamodely), ale, jak widać, akurat twórcy EPC do dziś nie zdecydowali się opisać

---

<sup>1</sup> MOLA — <http://mola.mii.lu.lv>

tej notacji w formie metamodelu. Drugim wyróżnikiem BPMN w stosunku do EPC jest fakt, że BPMN 2.0 z założenia jest notacją, która ma służyć do specyfikowania procesów wykonywalnych. Innymi słowy, prawidłowo zdefiniowany model procesu biznesowego w standardzie BPMN 2.0 jest jak kod w języku programowania. Można go uruchomić — przynajmniej w teorii. Cel ten także przyświecał od zawsze notacji BPEL (ang. *Business Process Execution Language*), która w zasadzie została stworzona tylko i wyłącznie do opisywania wykonywalnych procesów biznesowych. Problem BPEL jest jednak taki, że jest to notacja mało wygodna dla ludzi. Standard BPMN teoretycznie ma być w tym kontekście lepszy. W praktyce każdy z dostawców narzędzi klasy BPMS wymusza uzupełnianie modeli BPMN o szczegóły specyficzne dla ich platformy procesowej. Praktycznie więc żaden obecny system BPMS nie obsługuje „czystej” postaci notacji BPMN 2.0. Sprawę porównania EPC i BPMN dodatkowo komplikuje jeden fakt. Choć notacja EPC z założenia nie jest tworzona do specyfikowania procesów wykonywalnych, to na podobnej zasadzie jak twórcy systemów BPMS bazujących na modelach BPMN wymuszają uzupełnianie modeli o szczegóły specyficzne dla ich platformy, tak samo robi główny zwolennik EPC i dostawca systemu BPMS wykorzystującego modele w tej notacji, czyli firma Software AG. Innymi słowy, choć w założeniach BPMN 2.0 jest możliwość definiowania procesów wykonywalnych, a nie tylko analitycznych czy poglądowych, to w praktyce nie jest to takie łatwe. Idea ta okazała się wręcz nie do końca sensowna. Czym innym jest bowiem modelowanie analityczne, a czym innym specyfikowanie procesów wykonywalnych. Stąd między innymi doskonale przyjęta na rynku książka Bruce’a Silvera *BPMN. Method and Style*, która niejako „naprawia” błędne założenie BPMN 2.0, że jedna wspólna notacja może być wygodna i dla analityków, i dla programistów. To trochę tak, jakby drwal i stolarz mogli używać tego samego narzędzia do obróbki drewna... BPMN ma więc dobre założenia, ale ewidentnie będzie to standard, który jeszcze nieraz zmieni się w najbliższych latach. Jak zatem widać, porównanie EPC i BPMN nie jest trywialne. W obu przypadkach można z powodzeniem analizować i wdrażać procesy. Dodatkowo zarówno notacja EPC, jak i BPMN nie powinny być rozpatrywane w próżni. EPC jest bowiem częścią kompleksowej metody analizy ARIS<sup>2</sup>, która pozwala na dużo szerszą analizę biznesową wielu zagadnień istotnych w kontekście procesów biznesowych. Z drugiej strony, BPMN wraz z UML-em doskonale się uzupełniają, jeśli chodzi o specyfikowanie systemów informatycznych i procesów biznesowych, a ponieważ dziś systemy i procesy biznesowe są praktycznie nierozzerwalne, trudno sobie wyobrazić, że poważny dostawca oprogramowania może nie używać co najmniej jednej z tych notacji. Biorąc pod uwagę fakt, że dziś oprogramowanie działa

---

<sup>2</sup> ARIS method — <http://www.softwareag.com>

wszędzie, wszędzie też pojawiają się specyfikacje oprogramowania i specyfikacje procesów biznesowych. Można więc powiedzieć, że EPC jest trochę bardziej „dla biznesu”, a BPMN trochę bardziej „dla IT”.

Jedną szczególnie istotną cechą wyróżniającą BPMN jest jej powszechność. Ze względu na dobrze opisany metamodel BPMN notacja ta jest dostępna w większości narzędzi do modelowania i praktycznie we wszystkich systemach BPMS na rynku. Warto też wziąć pod uwagę fakt, że notacja w pewnym sensie jest wtórna. Ostatecznie będziemy musieli stosować taką, która jest obsługiwana w narzędziach, jakie kupimy. Siłą rzeczy z BPMN będzie nam wygodniej: większa popularność, większa dostępność, mniejsze ryzyko problemów. Dokładnie taki sam kłopot można mieć z językiem, którego zdecydujemy się używać do komunikacji biznesowej. Powszechnie wybrano angielski, choć nie do wszystkiego jest równie dobry.

### 1.3. Poziomy modelowania

Choć sam standard OMG BPMN 2.0 nie definiuje czegoś takiego jak poziom modelowania, praktyka pokazuje, że nie wszystkie składniki tego standardu są równie przydatne na każdym etapie projektu. Do podstawowych zastosowań wystarczy tylko pewien ograniczony zbiór elementów BPMN. Z drugiej strony, do opracowania wykonywalnej wersji procesu konieczne jest świadome stosowanie całej gamy dostępnych elementów tej notacji. Pomiędzy tym ogólnym (poglądowym) i szczegółowym modelowaniem może występować stan pośredni. Tak się bowiem składa, że analitycy biznesowi na poglądowych modelach poprzestać nie mogą, ale także nie mogą zejść do najniższego poziomu modelowania, czyli poziomu modeli wykonywalnych. Wynika to z tego, że zanim podejmie się decyzję o uruchomieniu prac projektowo-programistycznych nad procesami, trzeba najpierw przeprowadzić analizę biznesową tych procesów. Możemy więc wyróżnić co najmniej trzy poziomy<sup>3</sup> szczegółowości modelu w notacji BPMN 2.0:

- **Model poglądowy** — służy jedynie przedstawieniu ogólnego przebiegu procesu biznesowego, bez wnikania w jakiegokolwiek zagadnienia techniczne, takie jak typy zadań, parametry bramek i przepływów warunkowych, typy zdarzeń, nierozwinięte podprocesy, nieprezentowane obiekty danych.
- **Model analityczny** — służy do oceny rozmiaru prac niezbędnych do opracowania i wdrożenia procesu w postaci wykonywalnej; określone są typy zadań i zdarzeń, podprocesy są rozwinięte, obiekty danych są prezentowane na modelach, ale nie mają definicji.

---

<sup>3</sup> Bruce Silver, *BPMN. Method and Style*, Cody-Cassidy Press 2009.

- **Model wykonywalny** — służy do precyzyjnego opisania procesu wykonywalnego, a więc zdefiniowane są wszystkie jego czynności składowe, typy zadań i zdarzeń, parametry bramek w procesie, obiekty wraz z ich definicjami itd.

Poziomy te dotyczą modeli procesów biznesowych. W rzeczywistości jednak tym, co znajduje się w centrum naszego zainteresowania, są procesy biznesowe, a nie tylko ich modele. W związku z tym warto zauważyć, że poziomy przedstawione powyżej nie do końca odzwierciedlają realny cykl pracy z procesami biznesowymi w projektach BPM. W rzeczywistości bowiem występują jeszcze co najmniej dwa poziomy. Poziom wstępny, w którym nie istnieją żadne graficzne modele, a jedynie zdawkowe, lepsze lub gorsze opisy tekstowe procesów — możemy nazwać go poziomem „bez BPMN”. Z drugiej strony, sam model wykonywalny procesu biznesowego także nie wyczerpuje tematu wdrożenia tego procesu, często bowiem poza modelem wykonywalnym w notacji BPMN 2.0 konieczne jest uzupełnienie specyfikacji tego modelu o szczegóły specyficzne dla dostawcy systemu BPMS, w którym te procesy mają działać. Czasem polega to po prostu na konfiguracji szeregu parametrów, czasem są to dodatkowe modele graficzne w jakiejś innej („uzupełniającej”) notacji.

Szczegółowe omawianie poziomów modeli BPMN 2.0 w tym miejscu książki nie ma sensu. Bez znajomości całego standardu trudno mówić o tym, co dokładnie powinno się stosować i na którym poziomie. Dodatkowo standard OMG BPMN 2.0 nie definiuje takiego pojęcia jak „poziom modelu” ani tym bardziej nie określa, jakie elementy można stosować na poszczególnych poziomach. Trudno też narzucać jakieś uniwersalne poziomy, jeśli do pewnego stopnia sposób pracy z modelami procesów biznesowych zależy także od tego, jak prace te są zorganizowane w konkretnym zespole. W niektórych organizacjach prace są bardziej sformalizowane i konieczne jest dostarczanie produktów (specyfikacji) pośrednich, w innych nie ma takiego obowiązku. Warto więc poznać kompletny standard BPMN 2.0 przedstawiony w niniejszej książce, a następnie wraz ze swoim zespołem wypracować metodą kilku prób optymalny sposób modelowania procesów.

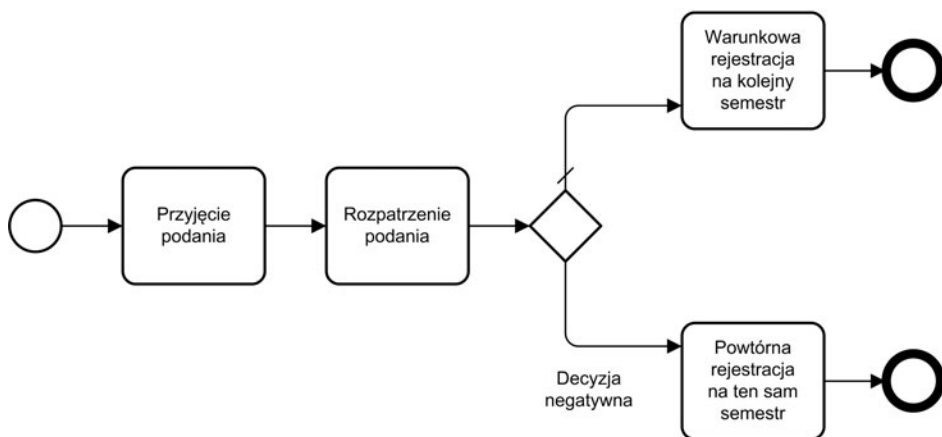


W tym rozdziale zaprezentowane zostały podstawowe elementy notacji BPMN 2.0, których zrozumienie jest niezbędne do lektury dalszych rozdziałów. Poszczególne elementy są przedstawione bardzo ogólnie i skrótowo. Dokładne i wyczerpujące wyjaśnienie znajduje się w kolejnych rozdziałach.

## 2.1. Pierwszy krok

Rysunek 2.1 przedstawia prosty model procesu w notacji BPMN 2.0. Jest to fragment procesu rejestracji studenta na kolejny semestr studiów, przy czym jest to student, który musi złożyć podanie o rejestrację warunkową, ponieważ nie zaliczył jakiegoś przedmiotu. Proces ten składa się z kilku kroków (zaokrąglone prostokąty na rysunku): przyjęcie podania, rozpatrzenie podania i rejestracja warunkowa na kolejny semestr lub powtórna rejestracja na ten sam semestr. Kroki te odzwierciedlają jakąś pracę realizowaną w procesie biznesowym. Występuje tu także jedna bramka (romb na rysunku), która oznacza, że w procesie następuje rozwidlenie, czyli możliwe są różne ścieżki.

Dodatkowo w procesie tym występują zdarzenia (kółka na rysunku): jedno początkowe (krawędź niepogrubiona) i dwa końcowe (krawędź pogrubiona). Zdarzenia w procesach symbolizują istotne sytuacje, które z różnych powodów warto lub trzeba zaznaczyć w modelu procesu. Ostatnim elementem, który występuje najbardziej licznie, są przepływy sekwencji (linie z grotami na rysunku), łączące zdarzenia, zadania i bramki. Dzięki nim wiemy, jaka jest kolejność (sekwencja) działań w procesie. W jednym przypadku przy bramce występuje także tzw. domyślny przepływ sekwencji (linia z grotom na końcu i ukośnikiem na początku, przy bramce). Są to podstawowe elementy notacji BPMN 2.0, których znajomość jest konieczna do rozumienia wszystkich modeli w tej notacji. Problem polega na tym, że w BPMN 2.0 poza przepływami sekwencji wszystkie te elementy mają od kilku do kilkudziesięciu (!) odmian. W związku z tym zapamiętanie całego standardu jest początkowo bardzo trudne, szczególnie że wielu elementów BPMN 2.0 można



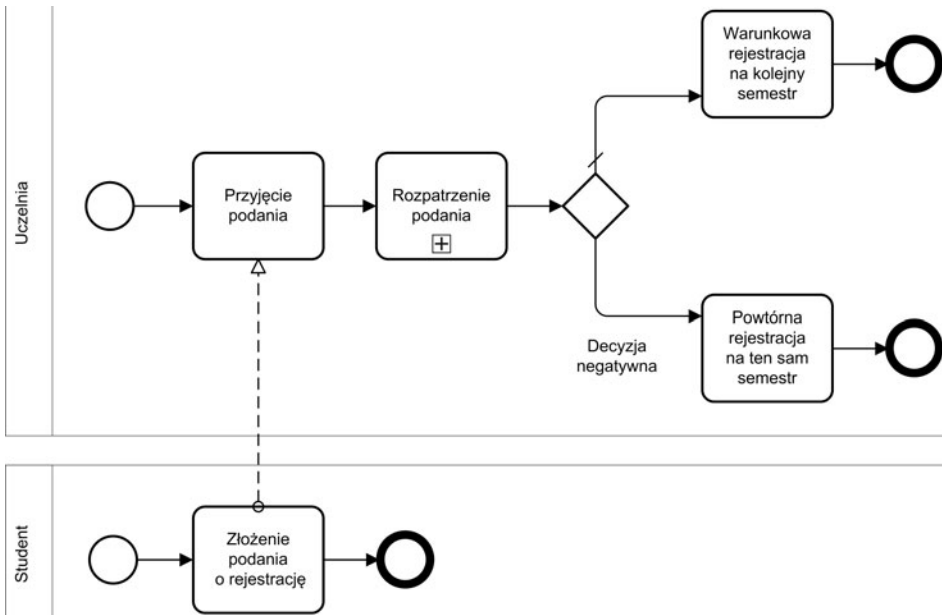
**Rysunek 2.1.** Model prostego procesu w notacji BPMN 2.0

poprawnie użyć tylko w ściśle określonym kontekście. Jest to więc notacja, która jest bardzo rozbudowana zarówno pod względem semantyki (liczby symboli), jak i syntaktyki (liczby reguł „gramatycznych”). Sytuacja na szczęście nie jest aż tak dramatyczna. Przede wszystkim dlatego, że modele w notacji BPMN z założenia tworzy się przy użyciu narzędzi do modelowania. Obecnie w internecie można znaleźć bardzo wiele darmowych narzędzi do modelowania w notacji BPMN 2.0. Do opracowania diagramów w tej książce użyto dodatku BPMN 2.0 Modeler for Visio firmy Trisotech, który wymaga wcześniejszego zainstalowania aplikacji Microsoft Visio 2007/2010. Ale dostępne są także samodzielne aplikacje, które nie wymagają niczego poza systemem operacyjnym. Rozwiązanie firmy Trisotech zostało wybrane z dwóch powodów. Po pierwsze, jest wręcz perfekcyjnie zgodne ze specyfikacją OMG BPMN 2.0. Po drugie, tworzenie diagramów BPMN 2.0 w tym narzędziu jest bardzo wygodne. No i oczywiście trzeci powód: firma Trisotech wyraziła zgodę na nieodpłatne wykorzystanie jej oprogramowania do przygotowania diagramów do niniejszej książki. Czytelnik powinien jednak samodzielnie przetestować i wybrać najwłaściwsze dla siebie rozwiązanie, które umożliwi łatwe tworzenie modeli w notacji BPMN 2.0.

## 2.2. Pule i tory

Przedstawiony poprzednio model prezentował fragment procesu rejestracji studenta z uwzględnieniem tylko tych czynności, które wykonuje uczelnia. Ktoś jednak mógłby się zastanawiać, jak ów proces wygląda w szerszej perspektywie, jaka jest np. rola studenta? Żeby rozwinąć ten model, trzeba wprowadzić uczestników procesu (rysunek 2.2).





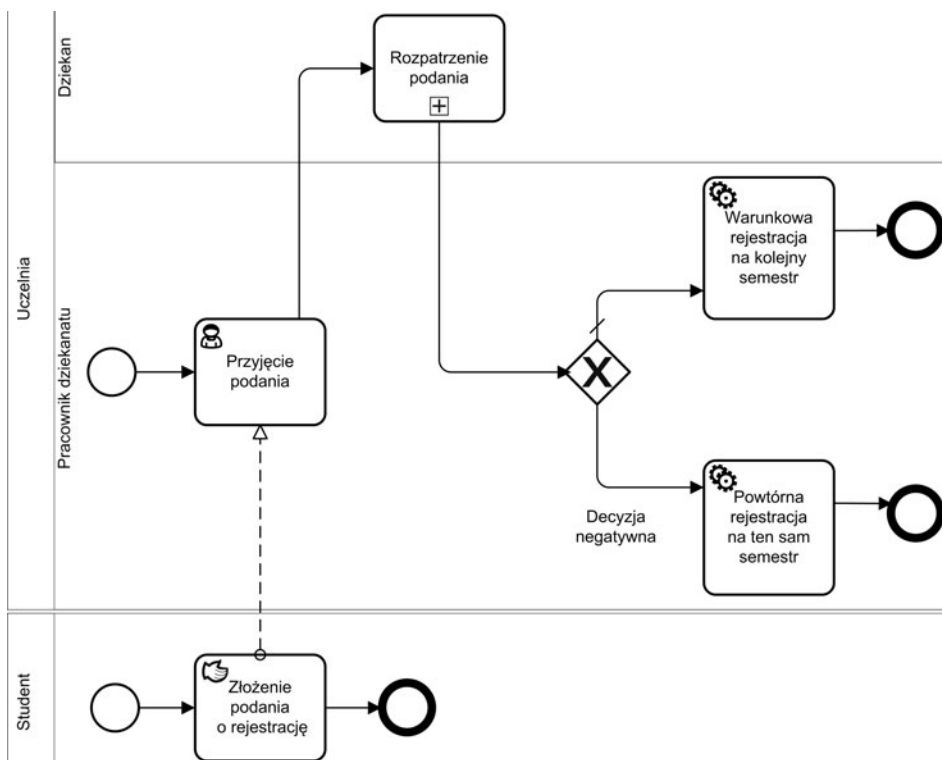
**Rysunek 2.2.** Model procesu z dwoma uczestnikami

W BPMN 2.0 uczestnicy procesu są prezentowani za pomocą pul (duże prostokąty otaczające pozostałe elementy na rysunku). W tej rozwiniętej formie poza czynnościami realizowanymi przez uczelnię widzimy także czynność realizowaną przez studenta. Warto zauważyć, że między uczestnikami nigdy nie ma przepływów sekwencji — mogą występować przepływy komunikatów (na rysunku: przerywana linia z kółkiem na początku i niewypełnionym grotem na końcu).

Jeśli chcielibyśmy dodatkowo jednoznacznie określić, kto na uczelni realizuje poszczególne zadania w tym procesie, możemy podzielić pulę uczelni na tory, czyli dodatkowe podziały wewnątrz puli (rysunek 2.3). Przy okazji, skoro już uszczegóławiamy nasz model, określmy także typy czynności (ikony w narożnikach czynności), które są realizowane w ramach tego procesu. Jeśli tego nie zrobimy, pozostawimy tzw. zadania abstrakcyjne. Nie jest to błąd, a raczej niepełna informacja.

Warto zwrócić uwagę na czynność *Rozpatrzenie podania*, która w tej wersji modelu ma dodatkowy znaczek plusa (+) pod nazwą. Oznacza on, że jest to czynność złożona — podproces. W tym przypadku jest to podproces zwinięty. Równie dobrze moglibyśmy rozwinąć tę czynność i w jej wnętrzu umieścić czynności składowe, ale nie będziemy tego teraz robić. Ważne, żeby mieć świadomość, że taki zapis oznacza, iż dana czynność jest złożona.

Ostatnią rzeczą, na którą warto zwrócić uwagę, jest bramka. Tym razem w jej wnętrzu pojawił się symbol, który oznacza jej typ. Z bramkami jest nieco inaczej



**Rysunek 2.3.** Szczegółowy model procesu

niż z zadaniami. W zadaniach niepokazanie symbolu wewnątrz oznacza „zadanie abstrakcyjne”, natomiast to samo w przypadku bramek oznacza bramkę wykluczającą (XOR), którą oznacza się także znakiem „X”. Tak więc o ile wszystkie pozostałe dodatkowe ikony i elementy są faktycznie dodatkowymi informacjami, o tyle znaczek „X” wewnątrz bramki w nowej wersji modelu nic nie dodaje. Ewentualnie bramka ta jest bardziej zauważalna — nic więcej.

Wszystkie trzy przedstawione wersje modelu procesu rejestracji studenta są poprawne, jednak każda z nich prezentuje inny poziom szczegółowości. Warto mieć świadomość, że jest to jak najbardziej poprawne, choć nie zawsze mamy dowolność w tym, jak szczegółowo rozpiszemy dany proces.

Bramki (ang. *gateways*) w notacji BPMN są podstawowym mechanizmem prezentowania logiki przepływów w procesach biznesowych. Bardzo często bowiem w procesach biznesowych mamy do czynienia nie z sekwencjami kilku aktywności następujących po sobie, ale z różnymi kombinacjami tych aktywności. Przykładowo gdy modelujemy proces rozpatrywania wniosku o dofinansowanie, ścieżka aktywności (kroków) w przypadku pozytywnego rozpatrzenia wniosku będzie przebiegała inaczej niż ścieżka po odrzuceniu wniosku. Zanim jednak zabierzemy się za modelowanie tych dwóch ścieżek, musimy w jakiś sposób określić owo rozwidlenie: kiedy następuje, jakie warunki muszą być spełnione, czy dalsze kroki są realizowane równoległe i niezależnie, czy może mamy do czynienia z wyborem „albo-albo”. Za każdym razem gdy w procesie mamy kilka alternatywnych ścieżek działania, mamy na pewno do czynienia z miejscem procesu, które na modelu może być zaprezentowane właśnie z użyciem jakiejś bramki. W tym rozdziale omówione są poszczególne rodzaje bramek. Należy jednak pamiętać, że bramki w notacji BPMN 2.0 nie zawsze są obowiązkowe, tzn. można poprawnie zamodelować proces bez użycia niektórych bramek — więcej na ten temat w rozdziale 4., „Dzielenie i łączenie bez bramek”.

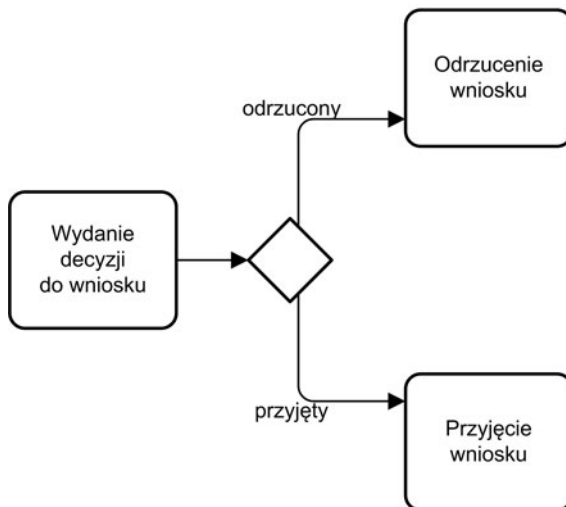
W językach programowania mamy do czynienia z wyrażeniami sterującymi, np. w języku C++: `if`, `switch`, `do while`, `while`, `for`, dzięki którym możemy układać następujące po sobie instrukcje w algorytmy. W procesach biznesowych układamy aktywności (kroki), ale w nieco inny sposób. Dla osób, które mają doświadczenie w pisaniu kodu, może to być początkowo zaskakujące, ale w notacji BPMN nie ma takiego bogactwa wyrażań sterujących, za to odpowiednik instrukcji `if` ma wiele wariantów, choć tylko jeden z nich może być powiązany z wyrażeniem warunkowym.

Zanim przejdziemy do omawiania konkretnych rodzajów bramek w standardzie BPMN 2.0, warto, żeby Czytelnik zapamiętał jedną regułę, której przyjęcie spowoduje lepsze rozumienie każdego z omawianych typów:

*Żadna z bramek w notacji BPMN 2.0 nie ma ograniczenia co do liczby przepływów z niej wychodzących. Nie jest więc prawdą, że zwykła bramka wykluczająca OR to wybór między dwiema ścieżkami. Typowy błąd większości osób modelujących w notacji BPMN polega na tym, że używają bramki wykluczającej wtedy, gdy są dwa możliwe warianty realizacji czegoś w procesie, oraz bramki złożonej (complex gateway), z symbolem gwiazdki wewnątrz, gdy możliwych wariantów realizacji jest więcej niż dwa. Przykładowe modele prezentowane w dalszej części tego rozdziału są przeważnie ograniczone do rozwidlenia procesu na dwie ścieżki, ale wynika to jedynie z „lenistwa” autora, a nie z tego, że nie powinno być ich więcej.*

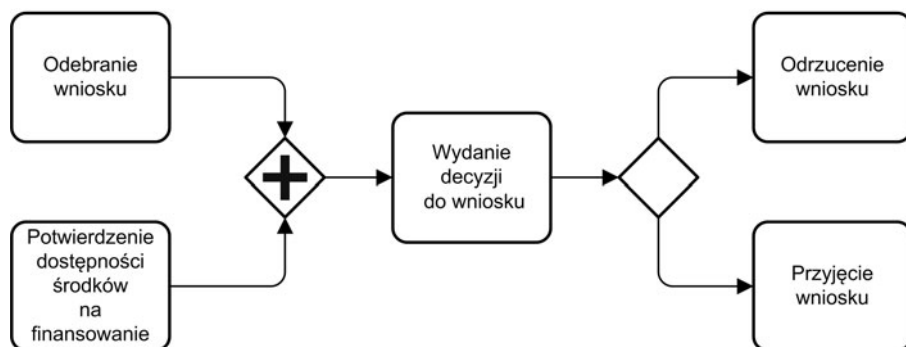
### 3.1. Bramki w BPMN

Wszystkie bramki w notacji BPMN 2.0 są reprezentowane przez znak diamentu, natomiast typ bramki jest określony za pomocą symbolu umieszczonego wewnątrz tego znaku. Warto jednak pamiętać, że brak jakiegokolwiek symbolu wewnątrz bramki oznacza bramkę wykluczającą (rysunek 3.1).



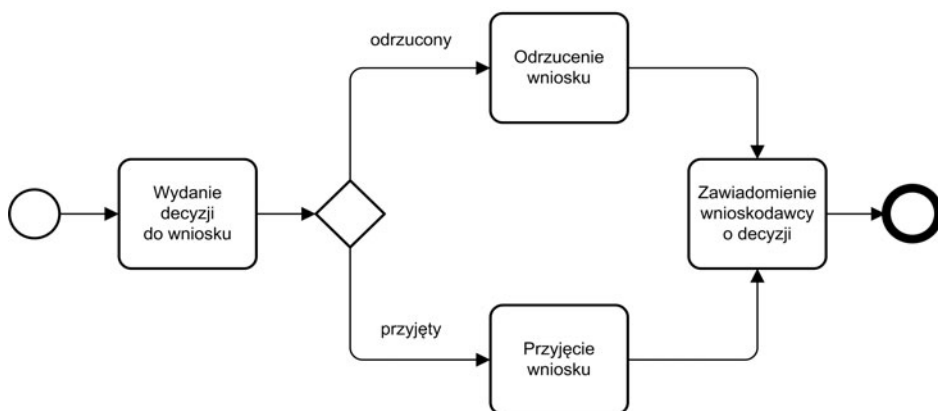
**Rysunek 3.1.** Fragment procesu z bramką (wykluczającą XOR)

W notacji BPMN 2.0 mogą występować bramki łączące kilka alternatywnych ścieżek, nawet jeśli nigdzie wcześniej w modelu nie ma rozgałęzienia procesu. Kontynuując wcześniejszy model rozpatrywania wniosku, można powiedzieć, że proces rozpatrywania rozpoczyna się wtedy, gdy Biuro Rozpatrywania Wniosków odbierze wniosek oraz gdy są jeszcze dostępne środki na dofinansowania (rysunek 3.2).

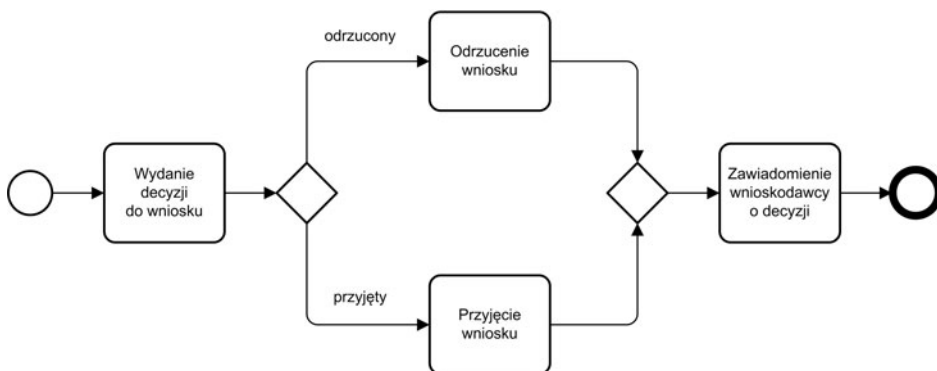


**Rysunek 3.2.** Bramka równoległa łącząca (po lewej) bez wcześniejszej bramki rozdzielającej

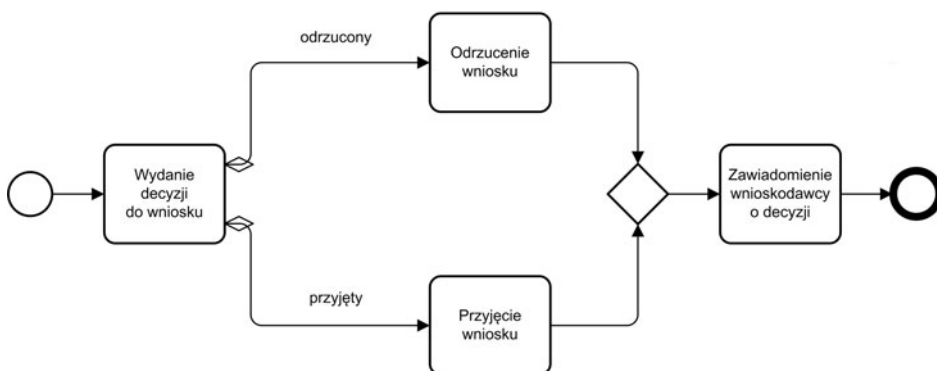
Zanim przejdziemy do omawiania poszczególnych rodzajów bramek, warto zwrócić uwagę na jeszcze jedną rzecz. Jak już wcześniej wspomniano, w notacji BPMN 2.0 można pomijać bramki i modelować fragmenty procesu bez bramek (rysunek 3.6). Możemy także używać warunkowych przepływów sekwencji (rozdział 6., „Obiekty danych”). Całe zadanie komplikuje się, gdy w modelowanym procesie dochodzimy do części, w której z powrotem chcemy złączyć kilka rozłącznych ścieżek i jednocześnie stosujemy kilka sposobów modelowania (z bramkami, bez bramek i z warunkowymi przepływami sekwencji, jak na rysunkach 3.3 i 3.5). Wtedy łatwo popełnić błąd. Dlatego na początku zdecydowanie zalecane jest stosowanie do każdej bramki otwierającej — do pary — bramki zamykającej (rysunek 3.4). W ten sposób unika się późniejszych problemów z interpretacją modeli. Poniżej znajduje się kilka równoważnych modeli całkowicie poprawnych w notacji BPMN 2.0. Jeśli teraz wyobrazimy sobie nieco bardziej skomplikowany proces biznesowy niż ten na przykładzie, dość szybko dojdziemy do wniosku, że w tym przypadku elastyczność notacji BPMN 2.0 może być zgubna...



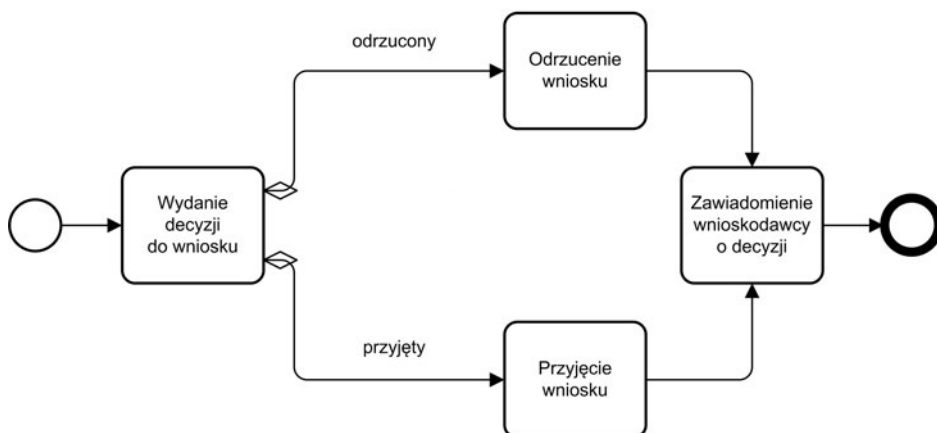
**Rysunek 3.3.** Wariant 1: bramka dzieląca, brak bramki łączącej



**Rysunek 3.4.** *Wariant 2: bramka dzieląca i łącząca do pary (zalecany sposób dla początkujących)*



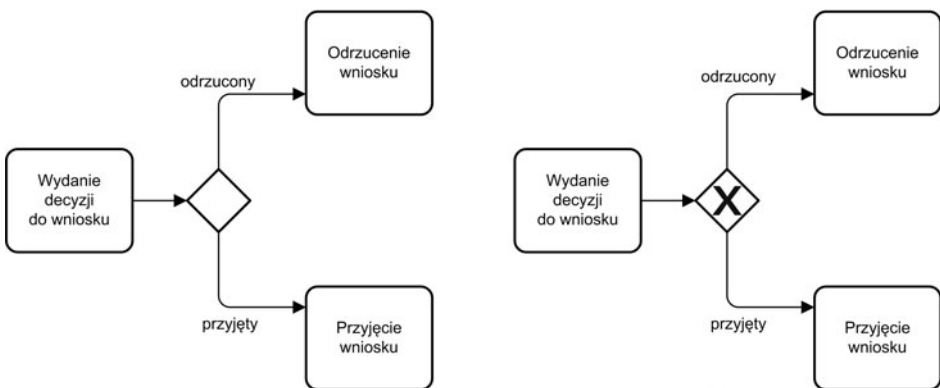
**Rysunek 3.5.** *Wariant 3: brak bramki dzielącej, bramka łącząca*



**Rysunek 3.6.** *Wariant 4: brak bramek, tylko przepływy warunkowe*

### 3.2. Bramka wykluczająca

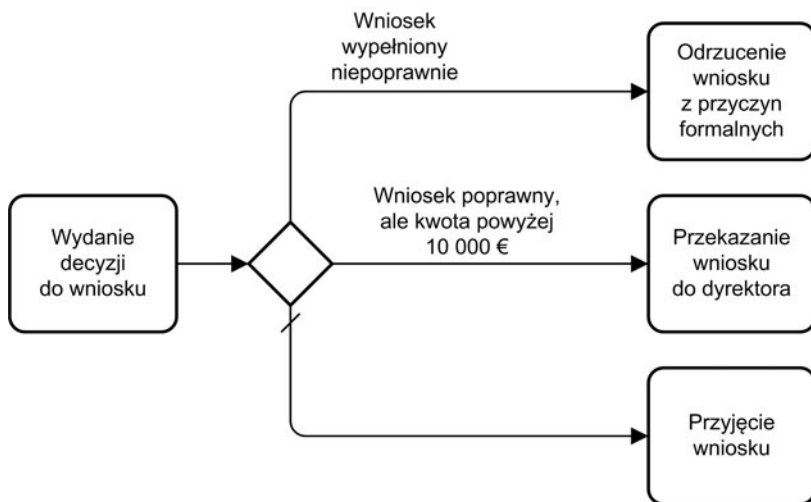
W przypadku gdy w procesie następuje warunkowe rozgałęzienie, tzn. istnieje kilka alternatywnych ścieżek, ale tylko jedna z nich może zostać wybrana w danym przebiegu, mamy do czynienia z **bramką wykluczającą**. Bramka wykluczająca musi być oznaczona poprzez użycie pustego znaku diamentu lub znaku diamentu z symbolem „X” wewnątrz niego (rysunek 3.7). Symbol „X” pochodzi od operatora XOR, który w języku angielskim oznacza *exclusive or*. Jest to operator umożliwiający wybór tylko jednej z wielu ścieżek — w odróżnieniu od zwykłego operatora OR, czyli „lub”, który umożliwia przejście do dowolnej liczby ścieżek alternatywnych. Innymi słowy, zawsze gdy chcemy powiedzieć, że może wystąpić tylko jedna ścieżka alternatywna, musimy zastosować bramkę wykluczającą. W praktyce jest to najczęściej stosowana bramka w modelach procesów biznesowych i potocznie nazywa się ją punktem decyzyjnym. Stąd też dla wygody może być oznaczana po prostu jako diament, bez symbolu wewnątrz. Warto oczywiście konsekwentnie stosować jeden z dwóch zapisów, mieszanie notacji z symbolem „X” i bez może utrudniać interpretację modelu, choć jest poprawne wg standardu BPMN 2.0.



**Rysunek 3.7.** Dwa równoważne sposoby prezentowania bramki wykluczającej w BPMN 2.0

Początkowo zrozumienie bramek w notacji BPMN może być trudne. W szczególności jeśli samą bramkę utożsamiamy ze sprawdzeniem jakiegoś warunku. Jednak jeśli przyjmiemy (zgodnie ze standardem BPMN 2.0), że bramka tylko symbolizuje rozgałęzienie, a dopiero wyjścia z niej odpowiadają za konkretne wyrażenia warunkowe, wtedy zrozumienie i rozróżnianie bramek wykluczających i niewykluczających stanie się prostsze. Tak więc w przypadku bramki wykluczającej każdy przepływ wychodzący odpowiada nie za wartość jednego warunku sprawdzanego na bramce, ale za jeden warunek sprawdzający i jego prawdziwość. Jeśli warunek związany z daną ścieżką wychodzącą z bramki nie jest prawdziwy, na pewno ta

ścieżka nie będzie wybrana — niezależnie od rodzaju bramki, z której wychodzi. Rysunek 3.8 zawiera ilustrację tego zagadnienia. W ten sposób na bramce możemy mieć tyle wyrażeń warunkowych, ile jest ścieżek. Z jednym wyjątkiem: może istnieć ścieżka, która jest domyślna, bez wyrażenia warunkowego, tzn. musi być ona wybrana wtedy, gdy nie została wybrana żadna inna. W przypadku każdej bramki taka domyślna ścieżka jest czymś w rodzaju „ostatniej deski ratunku”; na modelu jest ona oznaczana znakiem ukośnika (/) na krawędzi (rysunek 3.8).



**Rysunek 3.8.** Bramka wykluczająca z dwiema ścieżkami warunkowymi i jedną domyślną

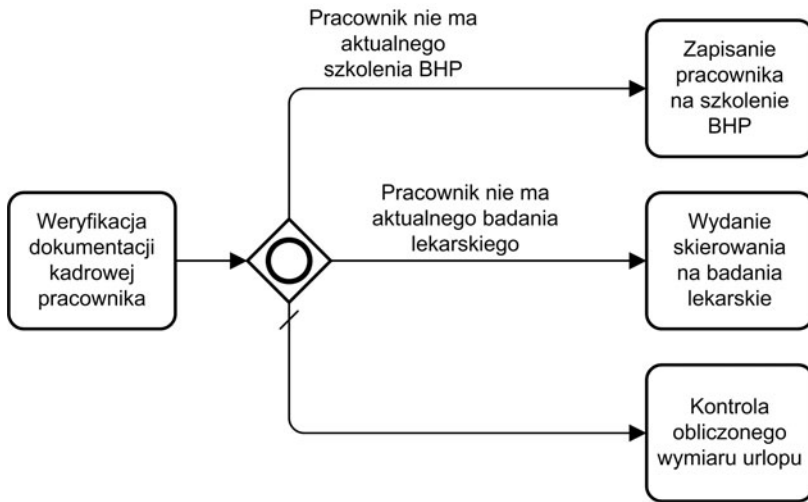
Wyrażenia opisujące poszczególne wyjścia z bramki zgodnie ze standardem BPMN 2.0 mogą być pisane językiem naturalnym. Tak więc może to być swobodny tekst, np.: „Czy osoba jest pełnoletnia?”, lub tzw. pseudokod, np.: `Osoba.wiek >= 18`. Choć w specyfikacji OMG BPMN 2.0 nie jest to jasno określone, można zakładać, że opisywanie warunków w języku naturalnym jest dopuszczalne tylko dla procesów niewykonywalnych (ang. *non-executable*). W przypadku procesów wykonywalnych (ang. *executable*) raczej powinniśmy stosować **wyrażenia formalne**, czyli wyrażenia logiczne w określonym języku. W przeciwnym razie nie moglibyśmy mówić o modelu procesu wykonywalnego. Więcej na temat procesów wykonywalnych można przeczytać w podrozdziale 1.3.

### 3.3. Bramka niewykluczająca

W przypadku gdy w procesie występuje rozgałęzienie, tzn. istnieje kilka alternatywnych ścieżek i możliwe jest wybranie kilku z nich, mamy do czynienia z **bramką niewykluczającą**. Bramka niewykluczająca musi być oznaczona poprzez znak



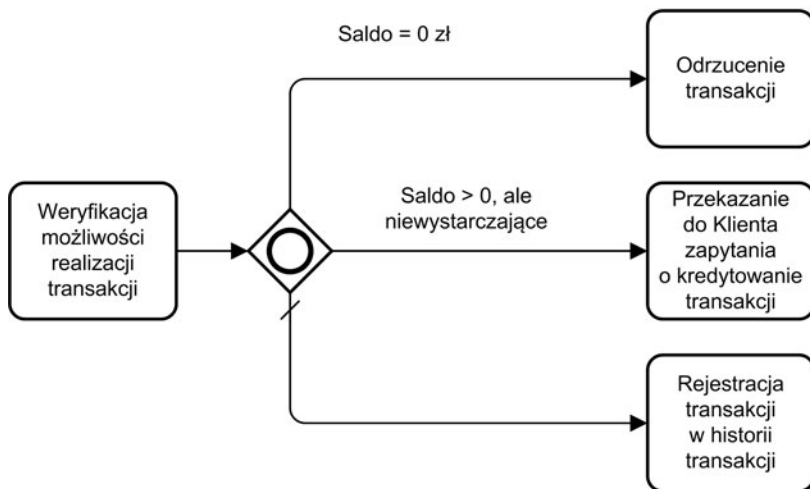
diamentu z symbolem „O” wewnątrz niego (rysunek 3.9). Symbol „O” pochodzi od operatora OR, który oznacza angielskie *or*. W standardzie BPMN 2.0 bramka niewykluczająca oznacza, że w procesie sprawdzane są warunki na wszystkich ścieżkach wychodzących z danej bramki i uruchamiana jest każda ścieżka, której warunek jest prawdziwy. Dla przypomnienia: w bramce wykluczającej możliwa jest tylko jedna ścieżka. Natomiast domyślna ścieżka jest oznaczona symbolem ukośnika (/) i jest — tak samo jak w przypadku bramki wykluczającej — „ostatnią deską ratunku”, czyli ścieżką, która zostanie wybrana, jeśli okaże się, że nie została wybrana żadna inna. W przypadku gdy została wybrana choćby jedna z pozostałych ścieżek, ścieżka domyślna jest pomijana.



**Rysunek 3.9.** Bramka niewykluczająca z dwiema ścieżkami warunkowymi i jedną domyślną

Bramka niewykluczająca może funkcjonować jako punkt decyzyjny, ale także jako punkt zrównoleglenia kilku ścieżek. Zależy to tylko od tego, jakie wyrażenia warunkowe określono dla krawędzi z niej wychodzących. Jeśli są to wyrażenia wzajemnie się wykluczające, wtedy będziemy mieli do czynienia z alternatywnymi przebiegami procesu. Rysunek 3.10 przedstawia fragment procesu realizacji transakcji finansowej. Ponieważ zastosowano bramkę niewykluczającą, ale warunki wzajemnie się wykluczają, możliwa jest tylko jedna ścieżka w procesie. Taki sam efekt osiągnęlibyśmy, stosując w tym przypadku bramkę wykluczającą. Po co więc stosować bramkę niewykluczającą i wykluczające się wyrażenia warunkowe? Bramka niewykluczająca jest elastyczna, tzn. jeśli w trakcie modelowania pojawi się kolejna ścieżka, która będzie opatrzona warunkiem niezależnym od pozostałych, wystarczy po prostu, że ją dodamy. Nie będziemy musieli zmieniać rodzaju bramki i analizować ponownie wyrażen na wszystkich ścieżkach. Może się też okazać, że wyrażenia

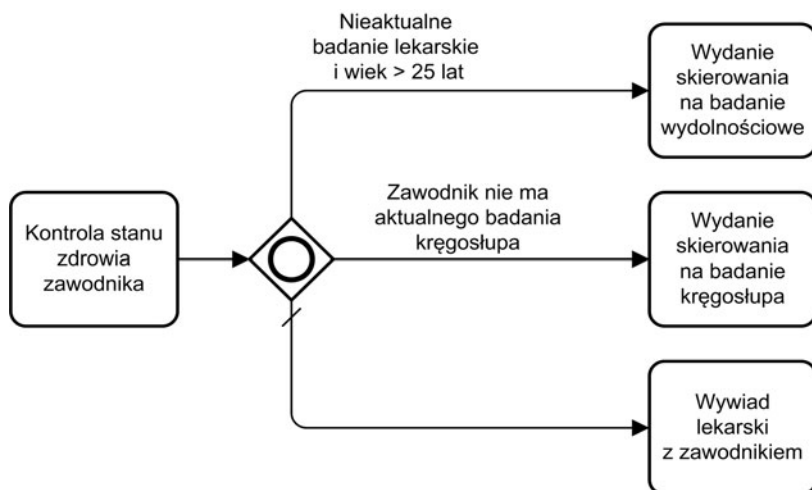
warunkowe mogą zależeć od parametrów zmieniających się w trakcie procesu, i wtedy z góry wiemy, że sama bramka musi być niewykluczająca, a warianty wyrażeń na ścieżkach wychodzących z bramki są trudno przewidywalne (czasem mogą się pokrywać, czasem mogą być zupełnie niezależne). Modelowanie procesów biznesowych — przynajmniej w fazie wstępnej — bardziej przypomina zbieranie różnych założeń co do funkcjonowania firmy czy organizacji niż bardzo jednoznaczne i precyzyjne opisywanie procesu w formie algorytmu. Dlatego taka sytuacja nie jest błędem formalnym, dopóki jesteśmy w trakcie „odkrywania” procesu. W chwili, gdy z ogólnych (poglądowych) modeli procesów biznesowych przechodzimy do modelowania procesów wykonywalnych, pozostawianie bramek niewykluczających z wyrażeniami wykluczającymi się powinno być czymś uzasadnione lub zostać poprawione.



**Rysunek 3.10.** *Bramka alternatywna ze ścieżkami warunkowymi (wykluczającymi się)*

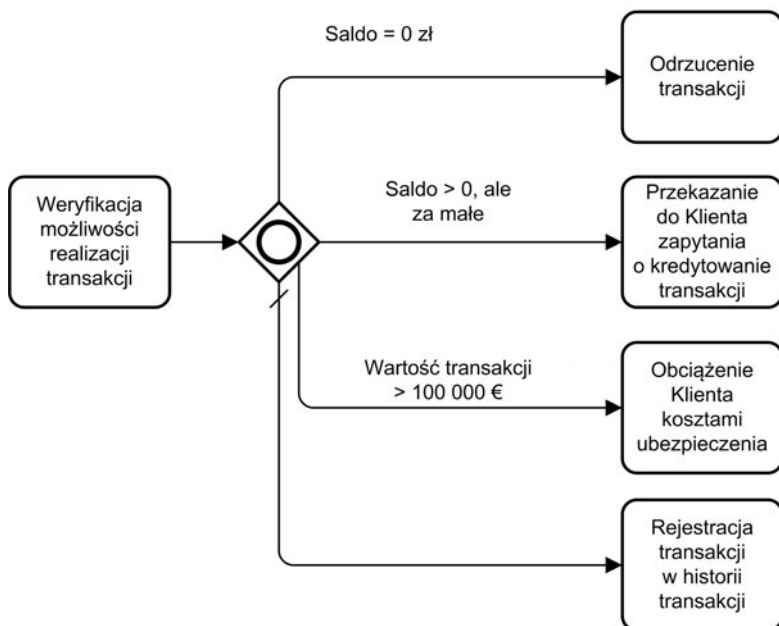
Jeśli wyrażenia warunkowe będą wzajemnie niezależne, możliwy będzie jednocześnie wybór kilku różnych ścieżek w jednym przebiegu procesu. Rysunek 3.11 przedstawia fragment procesu kontroli zdrowia zawodnika sportowego. Ponieważ zastosowano bramkę niewykluczającą, a warunki na wyjściach z niej są niezależne, możliwe są następujące sytuacje:

- zawodnik w trakcie badania otrzyma skierowanie na badanie wydolności serca i jednocześnie skierowanie na badanie kręgosłupa,
- zawodnik otrzyma tylko skierowanie na badanie wydolności serca,
- zawodnik otrzyma tylko skierowanie na badanie kręgosłupa,
- lekarz przeprowadzi jedynie wywiad z zawodnikiem.



**Rysunek 3.11.** Bramka niewykluczająca ze ścieżkami warunkowymi (niezależnymi)

Rysunek 3.12 przedstawia model procesu, w którym występuje jedna bramka niewykluczająca i kombinacja wyrażeń wzajemnie się wykluczających oraz wyrażeń niezależnych. W przedstawionym procesie zawsze, gdy transakcja dotyczy kwoty większej niż równowartość 100 000 €, klient jest obciążany kosztami ubezpieczenia, nawet jeśli transakcja ta miałaby być odrzucona ze względu na zerowe saldo.



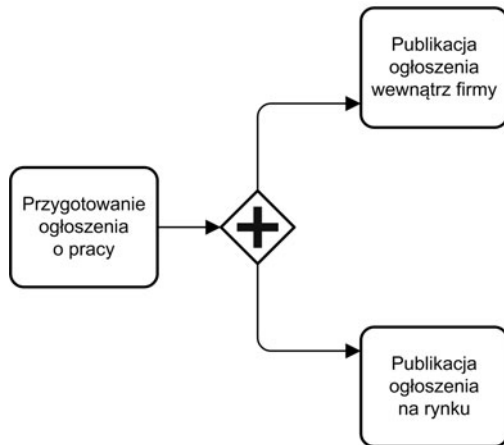
**Rysunek 3.12.** Bramka niewykluczająca z wykluczającymi się i niezależnymi wyrażeniami

Widać więc, że bramka niewykluczająca w standardzie BPMN 2.0 z jednej strony jest wygodna do „skracać” modelu poprzez odpowiednie kombinacje wyrażeń, z drugiej jednak, praktycznie zawsze wymaga dłuższego zastanowienia się i początkowo sprawia problemy w interpretacji modeli procesów biznesowych.

### 3.4. Bramka równoległa

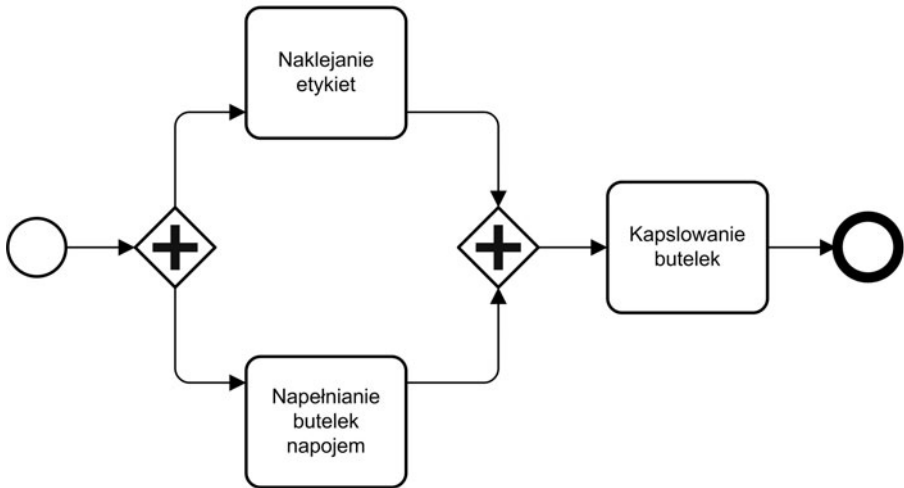
Jeśli mamy do czynienia z procesem, w którym jakieś czynności są realizowane niezależnie, powinniśmy zastosować **bramkę równoległą**. Jest ona reprezentowana przez znak diamentu ze znakiem plusa (+) wewnątrz. Rysunek 3.13 przedstawia przykładowy fragment procesu, w którym następuje rozwidlenie na dwie niezależne ścieżki poprzez zastosowanie bramki równoległej.

Warto zwrócić uwagę na to, że pojęcie równoległości w standardzie BPMN 2.0 (i UML także) jest bliższe temu, co potocznie nazwalibyśmy raczej niezależnością. „Równoległość” intuicyjnie kojarzy nam się z faktem, że jakieś czynności realizowane są jednocześnie (w tym samym czasie). Jednak w tym przypadku nasza intuicja się myli. W standardzie BPMN 2.0 równoległość to niezależność, a nie jednoczesność. Innymi słowy, dwa równoległe procesy nie muszą być realizowane w tej samej chwili, natomiast na pewno muszą być wzajemnie niezależne.



**Rysunek 3.13.** Przykładowy proces z bramką równoległą

Rysunek 3.14 przedstawia przykładowy model procesu produkcji sześciopaka z napojami. Jeśli posiadamy odpowiednią linię produkcyjną (zestaw urządzeń), to możemy zaprojektować proces tak, że napełnianie butelek napojami i naklejanie etykiet może być realizowane niezależnie, tzn. nie musi przebiegać sekwencyjnie i może być jednoczesne.



**Rysunek 3.14.** Bramka równoległa i bramka synchronizująca

Zrównoleglenie ścieżek w procesach biznesowych możemy oczywiście osiągnąć także poprzez zastosowanie bramki niewykluczającej i odpowiednich wyrażeń warunkowych. Jednak w przypadku bramki równoległej mamy zawsze pewność, że wszystkie ścieżki z niej wychodzące będą realizowane. W przypadku bramki niewykluczającej wszystko zależy od prawdziwości wyrażeń warunkowych na przepływach wychodzących z tej bramki.

Bramkę równoległą — zgodnie ze standardem BPMN 2.0 — możemy także stosować do synchronizacji niezależnych fragmentów procesu. W ten sposób możemy zagwarantować to, że jakieś czynności w procesie będą realizowane tylko wtedy, gdy wcześniejsze równoległe ścieżki dobiegną końca. Mechanizm ten został zaprezentowany na modelu procesu rozpatrywania wniosku o dofinansowanie (rysunek 3.16) oraz na modelu procesu produkcji sześciopaka (rysunek 3.14).

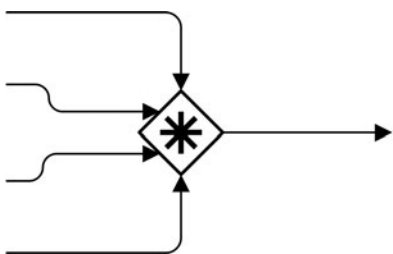
### 3.5. Bramka złożona

**Bramka złożona** jest szczególnym rodzajem bramki w standardzie BPMN 2.0, ponieważ jako jedyna umożliwia określenie jej *warunku aktywacji* oraz *stanu*. Co więcej, stan bramki może (choć nie musi) być wykorzystany w wyrażeniach warunkowych przepływów z niej wychodzących. Dzięki temu mechanizmowi możliwe jest konstruowanie skomplikowanych rozwidleń w procesach, np. wg wzorca projektowego „blocking discriminator” czy „structured partial join”. Pełne zrozumienie (i wyjaśnienie istoty) bramki złożonej jest o tyle kłopotliwe, że wymaga odwołania się do metamodelu BPMN 2.0, niemniej w dalszej części rozdziału znajduje się

przykład mogący ułatwić to tym Czytelnikom, którzy do metamodelu zaglądać nie chcą. Przed prezentacją przykładu warto najpierw wyjaśnić znaczenie pojęć „warunek aktywacji” i „stan”, które w odniesieniu do bramki złożonej są kluczowe.

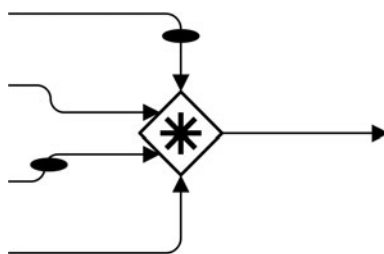
**Warunkiem aktywacji** bramki jest wyrażenie logiczne (tego samego typu co zwykle wyrażenie na przepływie warunkowym, patrz rozdział 6.), którego spełnienie umożliwia przejście przez bramę, czyli kontynuację procesu. Przykładowym warunkiem aktywacji bramki mogłoby być wyrażenie:  $w_{e1} + w_{e2} + \dots + w_{eN} \geq 3$ , oznaczające, że na  $N$  możliwych przepływach wejściowych bramki co najmniej 3 są aktywne. Jeśli jest ich mniej, bramka nadal czeka. Natomiast **stanem** bramki jest jedna z dwóch sytuacji: albo oczekuje ona na uruchomienie, albo na zresetowanie. W metamodelu BPMN 2.0 warunek aktywacji i stan to atrybuty bramki złożonej, odpowiednio `activationCondition:Expression` i `waitingForStart:boolean`. Dodatkowym atrybutem, który możemy wykorzystywać w bramce złożonej, jest licznik aktywacji każdego przepływu wchodzącego do bramki (`activationCount:integer`).

1) Start



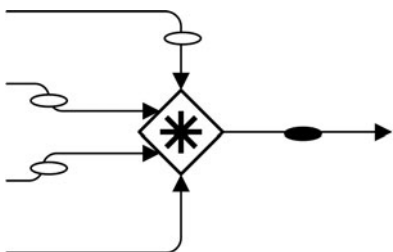
`waitingForStart = true`  
`activationCondition = false`

2) Jedno wejście aktywne



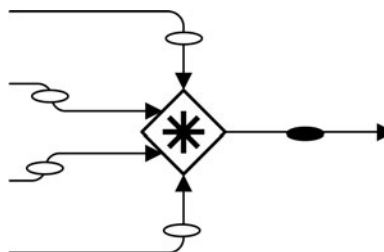
`waitingForStart = true`  
`activationCondition = false`

3) Dwa wejścia aktywne



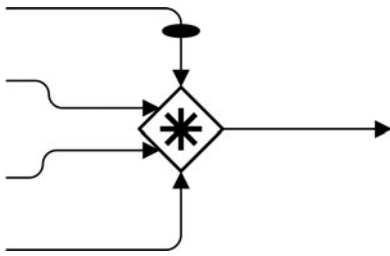
`waitingForStart = true`  
`activationCondition = false`

4) Trzy wejścia aktywne!



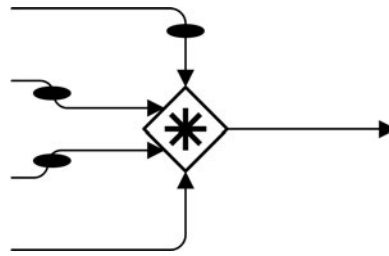
`waitingForStart = true`  
**`activationCondition = true`**

5) Przechodzimy przez bramkę



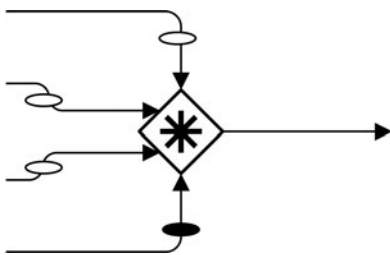
**waitingForStart = false**  
**activationCondition = true**

6) Kolejne wejście aktywne



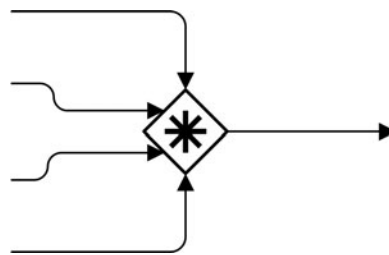
**waitingForStart = false**  
**activationCondition = true**

7) Przechodzimy i resetujemy bramkę



**waitingForStart = false**  
**activationCondition = true**

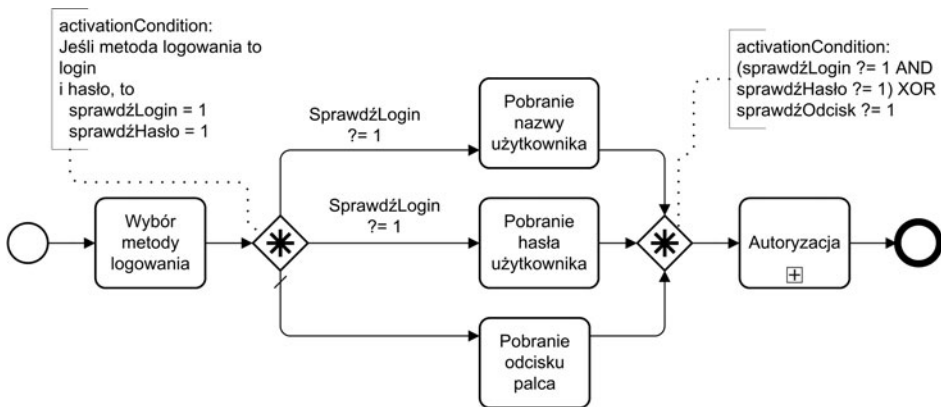
8) I od nowa...



**waitingForStart = true**  
**activationCondition = false**

Powyżej rozpisano krok po kroku działanie procesu z bramką złożoną. Do zaprezentowania aktywnego przepływu użyto wypełnionej elipsy, do zaprezentowania przepływu już raz aktywowanego — pustej elipsy. W tym przykładzie warunek aktywacji (**activationCondition**) jest spełniony, gdy co najmniej 3 wejściowe przepływy zostały aktywowane. Natomiast zerowanie bramki następuje w chwili, gdy wszystkie wejściowe przepływy zostały aktywowane, czyli dla każdego przepływu wejściowego atrybut **activationCount** jest równy 1.

Przepływy wychodzące z bramki złożonej należy traktować dokładnie tak samo jak w przypadku bramki niewykluczającej, tzn. dany przepływ jest aktywowany tylko wtedy, gdy jego wyrażenie warunkowe jest spełnione. Jeśli żaden z przepływów wychodzących nie ma spełnionego wyrażenia warunkowego, wybierana jest domyślna ścieżka. Jeśli domyślna ścieżka nie jest określona, występuje wyjątek. Rysunek 3.15 przedstawia proces autoryzowania się w jakimś systemie. Użytkownik najpierw musi wybrać metodę logowania. Następnie poprzez zastosowanie bramki złożonej z odpowiednio skonstruowanym warunkiem aktywacji wymuszane jest podanie nazwy użytkownika i hasła albo odcisku palca. Druga bramka złożona jest bramką zamykającą, która ponownie weryfikuje warunki w celu zapewnienia kompletności procesu logowania poprzez login i hasło lub poprzez odcisk palca.



**Rysunek 3.15.** Bramka złożona otwierająca i zamykająca

Podsumowując, bramka złożona jest szczególnym przypadkiem bramki w notacji BPMN 2.0 i umożliwia tworzenie modeli procesów, w których obowiązują szczególne reguły przebiegu. W praktyce jednak wielu producentów aplikacji do modelowania i implementacji procesów biznesowych wprost deklaruje, że ich oprogramowanie nie obsługuje tego typu bramki, ponieważ można ją z powodzeniem zastąpić np. bramką niewykluczającą i odpowiednio rozbudowanymi warunkami na przepływach wychodzących.

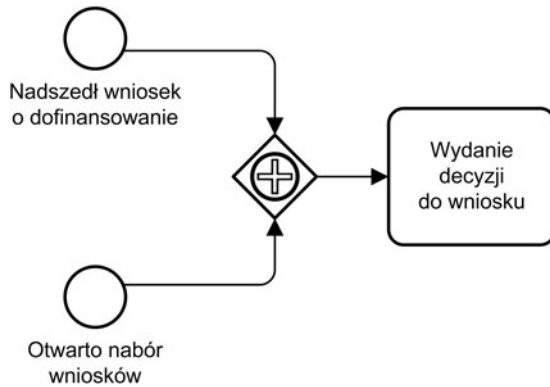
## 3.6. Bramka oparta na zdarzeniach

Na początku rozdziału, podczas dyskusji na temat procesu rozpatrywania wniosków, pojawił się model rozpoczynający się od bramki łączącej dwie ścieżki na początku procesu (rysunek 3.2). Jeśli dodatkowo w modelowanym procesie uwzględnimy *zdarzenia* i przyjmijemy, że rozpatrywanie wniosków rozpoczyna się, gdy nastąpią konkretne zdarzenia: nadejście wniosku i otwarcie okresu rozpatrywania wniosków, powinniśmy zastosować **bramkę opartą na zdarzeniach**.

### 3.6.1. Bramka oparta na zdarzeniach początkowych

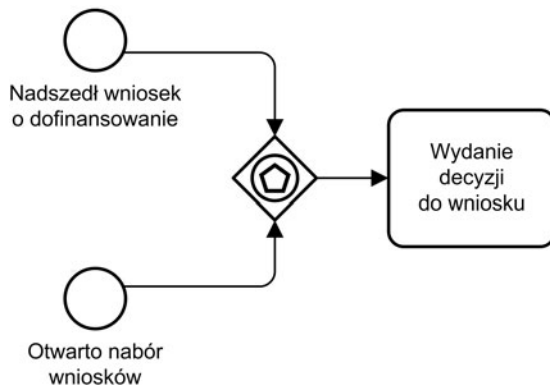
Rysunek 3.16 przedstawia fragment procesu rozpoczynającego się dopiero, gdy nastąpią dwa różne zdarzenia. Ponieważ chcemy wyrazić fakt, że proces rozpoczyna się pod wpływem dwóch zdarzeń, zarówno zdarzenia, które modelujemy, jak i symbol wewnątrz bramki — zgodnie z notacją BPMN 2.0 — są narysowane przy użyciu okręgu z niepogrubioną jedną krawędzią. Więcej informacji na temat zdarzeń, ich typów i rodzajów znajduje się w rozdziale 8., „Zdarzenia”. Mamy tu do czynienia z bramką opartą na zdarzeniach początkowych.





**Rysunek 3.16.** Równoległa bramka rozpoczynająca proces, oparta na zdarzeniach

Model tego procesu rozpoczyna się więc od bramki wyrażającej konieczność zajścia dwóch zdarzeń początkowych. Gdybyśmy chcieli zamodelować nieco inną sytuację, a mianowicie konieczność zajścia jednego z wielu zdarzeń, musimy użyć wykluczającej bramki rozpoczynającej proces, opartej na zdarzeniach (rysunek 3.17).

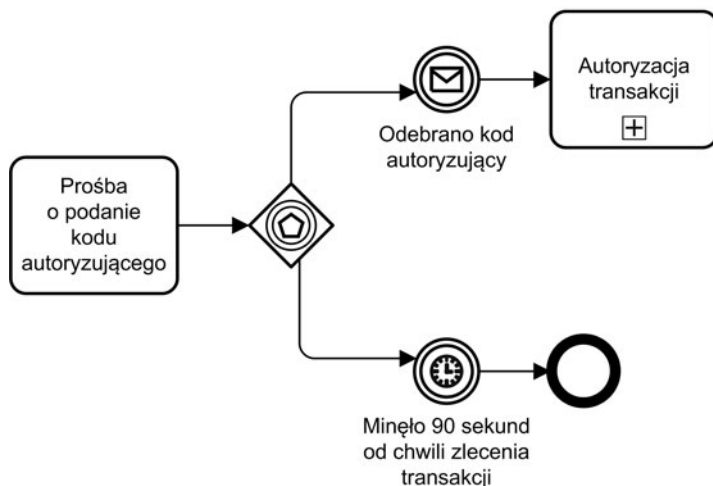


**Rysunek 3.17.** Wykluczająca bramka rozpoczynająca proces, oparta na zdarzeniach

### 3.6.2. Bramka oparta na zdarzeniach pośrednich

Bramka oparta na zdarzeniach pośrednich (rysunek 3.18) występuje zawsze przed zdarzeniami, których dotyczy. Kiedy proces dotrze do takiej bramki, aktywowane są wszystkie przepływy z niej wychodzące. Ponieważ przepływy te muszą być połączone ze zdarzeniami (ewentualnie z *zadaniami odbioru*, patrz podrozdział 9.1.3), w istocie proces zatrzymuje się, oczekując na wystąpienie któregośkolwiek z nich. W zależności od tego, które z tych zdarzeń nastąpi pierwsze, taka ścieżka będzie kontynuowana. Pozostałe ścieżki są pomijane.

Warto pamiętać, że bramka oparta na zdarzeniach pośrednich musi mieć co najmniej dwa przepływy wychodzące, żaden z nich nie może mieć wyrażenia warunkowego, a dozwolone zdarzenia pośrednie następujące po tej bramce to: *Komunikat*, *Sygnal*, *Timer*, *Zdarzenie-warunek*, *Wielozdarzenie* (które może składać się tylko z wcześniej wymienionych). Niedozwolone zdarzenia wraz z bramką opartą na zdarzeniach pośrednich to: *Błąd*, *Anulowanie*, *Kompensacja* i *Link*. Poszczególne rodzaje zdarzeń są omówione w rozdziale 8.



**Rysunek 3.18.** Bramka oparta na zdarzeniach pośrednich

Rysunek 3.18 przedstawia fragment procesu, w którym występuje bramka oparta na zdarzeniach pośrednich. Jest to fragment działań związanych z autoryzacją transakcji w systemie bankowości elektronicznej. Zgodnie z tym modelem użytkownik jest proszony o podanie kodu autoryzującego transakcję. Następnie system czeka 90 sekund. Jeśli w tym czasie odbierze kod autoryzujący, rozpoczyna podproces autoryzacji; jeśli w tym czasie nie zostanie tego kodu, proces się kończy i transakcja nie jest autoryzowana.

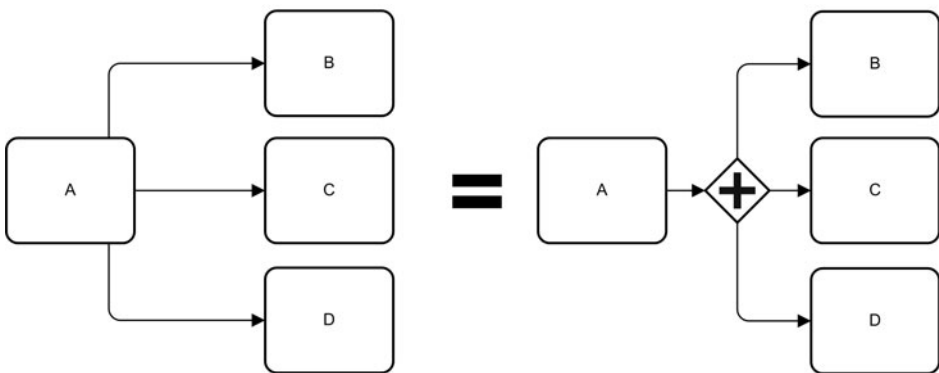
Bramka oparta na zdarzeniach końcowych w notacji BPMN 2.0 nie istnieje — nie miałyby sensu, ponieważ zdarzenia końcowe kończą proces i nie ma potrzeby robienia w nim czegoś jeszcze.

W poprzednim rozdziale omówione zostały wszystkie rodzaje bramek dostępne w standardzie BPMN 2.0. Wspomniano także o tym, że bramki można pomijać, ale należy mieć pełną świadomość, jakie ma to konsekwencje. W szczególności należy uważnie podchodzić do modeli, w których brakuje bramek zamykających. Bardzo łatwo w takich przypadkach popełnić błąd logiczny i w efekcie stworzyć model, który ma zupełnie inne znaczenie niż zamierzone. Dlatego zalecane jest najpierw nabranie biegłości w używaniu bramek, a potem stopniowe przechodzenie na modele bez bramek.

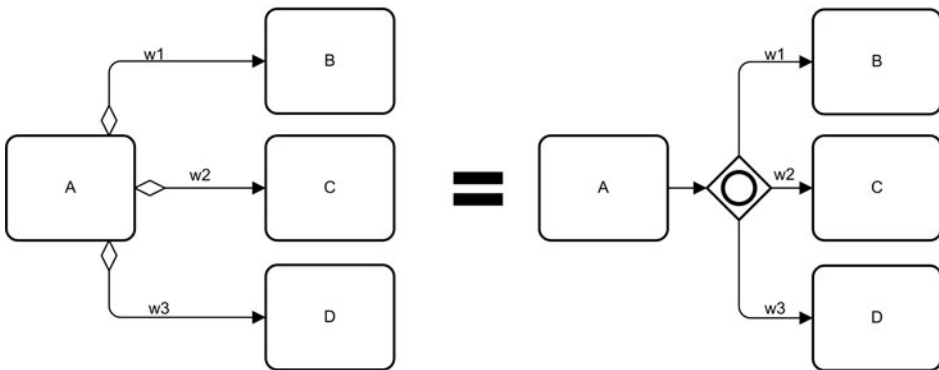
Wystąpienie bramki w modelu procesu biznesowego w notacji BPMN 2.0 oznacza, że istnieje mechanizm, który kontroluje przepływ czynności, a rodzaj bramki mówi o tym, w jaki sposób ta kontrola przebiega. Jeśli natomiast nie ma mechanizmu kontrolującego przepływ czynności, bramkę można pominąć. Z drugiej strony, bramka nie reprezentuje żadnej czynności związanej z wykonywaniem jakiejś „pracy” w procesie. Jest to więc pewien „nadrzędny” mechanizm, który pozwala na sterowanie procesem biznesowym, ale nie wpływa na czas, koszt lub inne wskaźniki operacyjne procesu. W praktyce ewaluowanie bramek w silniku procesów biznesowych może trwać, niemniej optymalizacja tego czasu (kosztu) jest domeną optymalizacji infrastruktury systemów informatycznych, a nie samego procesu biznesowego.

### 4.1. Rozgałęzienia bez bramek

Każda bramka (z wyjątkiem bramki złożonej) określa jedynie sposób, w jaki są obsługiwane krawędzie do niej wchodzące lub z niej wychodzące. Natomiast każda krawędź może, ale nie musi być związana z jakimś konkretnym warunkiem. Jeśli z danej czynności wychodzą przepływy sekwencji bez warunków, mamy do czynienia ze zrównolegleniem (rysunek 4.1) — w specyfikacji BPMN 2.0 określanym mianem *fork* lub *AND-split*. Jeśli są to przepływy sekwencji z warunkami, należy przyjąć, że występuje bramka niewykluczająca (rysunek 4.2).

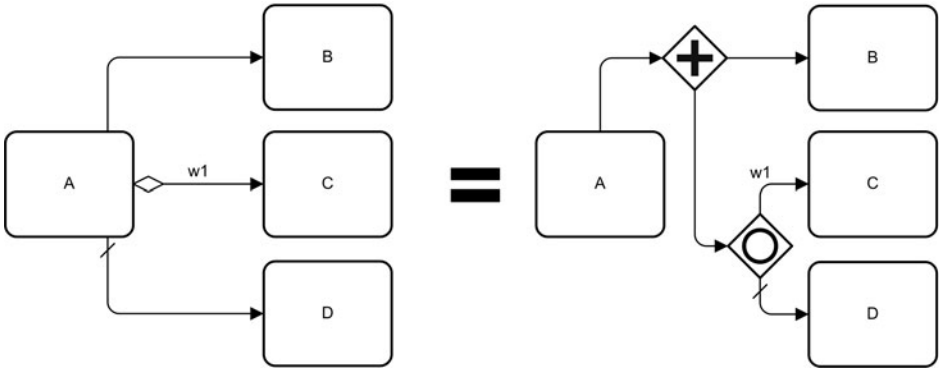


**Rysunek 4.1.** Rozgałęzienie bez bramki i odpowiadająca mu wersja z bramką (znak równości nie należy do notacji BPMN)

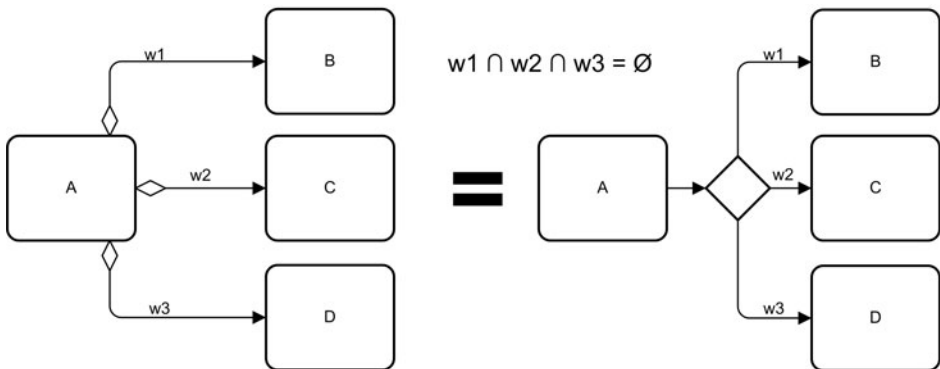


**Rysunek 4.2.** Rozwidlenie przepływów warunkowych bez bramki i z bramką (znak równości nie należy do notacji BPMN)

Jeśli więc nie użyjemy bramki i pomieszamy warunkowe i zwykłe przepływy sekwencji, musimy się nieco nagimnastykować, żeby uzyskać to samo, stosując bramki (rysunek 4.3). Na szczęście takie kombinacje są rzadkością. Najczęściej mamy do czynienia z kompletem przepływów warunkowych wykluczających się albo ze zwykłym zrównolegleniem. Warto jednak zauważyć, że o ile w przypadku zrównoleglenia sprawa jest prosta: zawsze oznacza ono bramkę równoległą, o tyle w przypadku przepływów warunkowych z wyrażeniami wykluczającymi się wzajemnie (np.  $a < 0$ ,  $a = 0$  i  $a > 0$ ) „logicznie” poprawna będzie również bramka wykluczająca (rysunek 4.4).



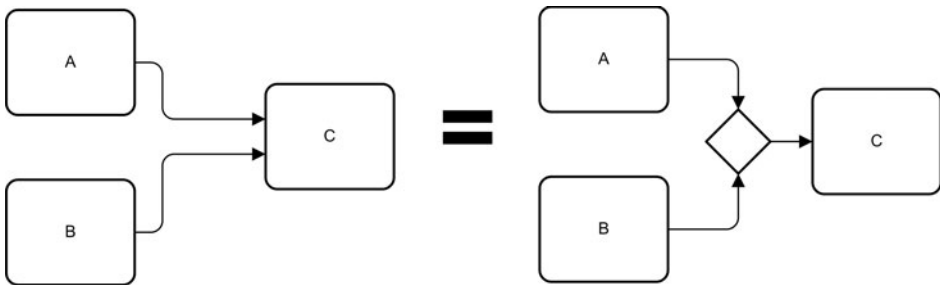
**Rysunek 4.3.** Od lewej: rozgałęzienie bez bramek i odpowiadający mu model z bramkami (znak równości nie należy do notacji BPMN)



**Rysunek 4.4.** Rozwidlenie wykluczających się przepływów warunkowych bez bramki i z bramką (znak równości nie należy do notacji BPMN)

## 4.2. Złączenia bez bramek

Jeśli pominiemy bramkę i połączymy kilka rozłącznych ścieżek procesu, mamy wtedy do czynienia z takim samym złączeniem jak w przypadku bramy wykluczającej. W specyfikacji BPMN 2.0 określa się to mianem *merge* lub *OR-join* (rysunek 4.5). Czynność C rozpocznie się, jeśli tylko zakończy się A lub B. Bardzo często złączenie bez bramy jest mylnie interpretowane jako złączenie z bramą zrównoleglającą (ze znakiem plusa (+) wewnątrz).

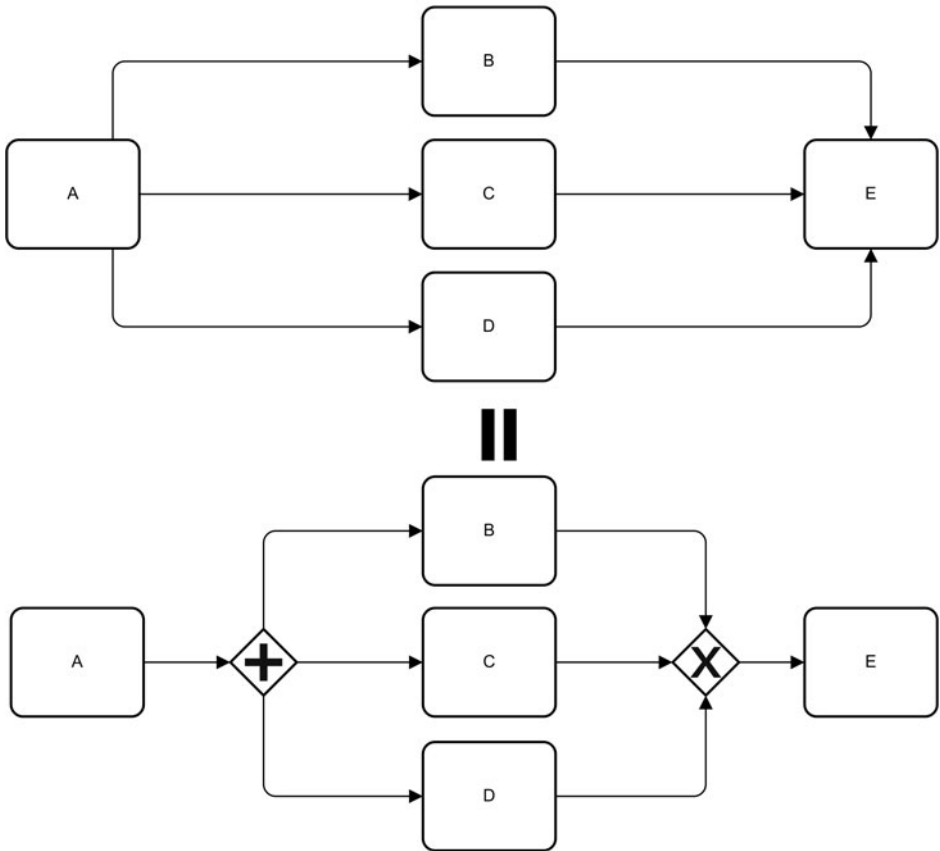


**Rysunek 4.5.** *Od lewej: złączenie bez bramki i z bramką wykluczającą (znak równości nie należy do notacji BPMN)*

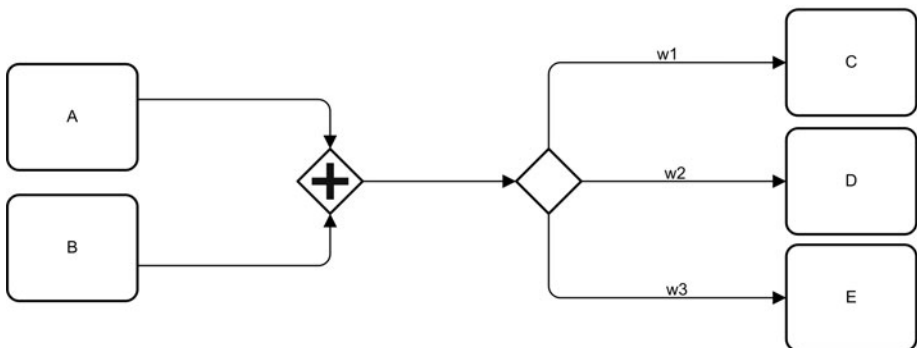
### 4.3. Kiedy pomijać bramki?

Całkowite wyeliminowanie bramek z modeli procesów biznesowych nie jest możliwe. Nie jest też sensowne. Warto jednak zapamiętać jeden model (rysunek 4.6), który można sobie wydrukować i przykleić koło biurka, aby utrwalić sobie, co oznacza pominięcie bramy dzielącej i łączącej, jeśli nie mamy przepływów warunkowych. W tym przypadku dla pewności bramka wykluczająca jest przedstawiona z symbolem wewnątrz.

Zdarza się, że w modelowanym procesie biznesowym występuje złożenie bramek, np. bramki równoległej (synchronizującej) i wykluczającej (rysunek 4.7). W takich przypadkach pozbycie się bramek jest praktycznie niemożliwe.



Rysunek 4.6. „Wzorzec” do zapamiętania



Rysunek 4.7. Złożenie bramek

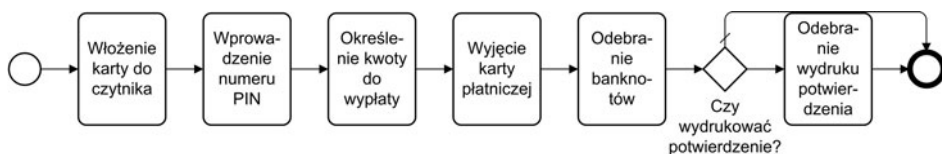




Pozornie modelowanie procesu biznesowego w notacji BPMN polega jedynie na graficznej prezentacji przebiegu tego procesu poprzez odpowiednie rozmieszczenie aktywności, zdarzeń i bram. Jednak równie istotnym elementem każdego modelu procesu biznesowego w notacji BPMN są przepływy (czyli linie łączące), które łączą wszystkie składniki modelu. To dzięki przepływowi możemy określić kolejność kroków w procesie i charakter relacji, które chcemy modelować. Możemy bowiem chcieć rozróżniać przepływ sekwencji od przepływu komunikatów albo sekwencję działań od przepływu danych. W tym rozdziale wyjaśniono, jak modelować poszczególne rodzaje przepływów oraz kiedy stosować każdy z nich.

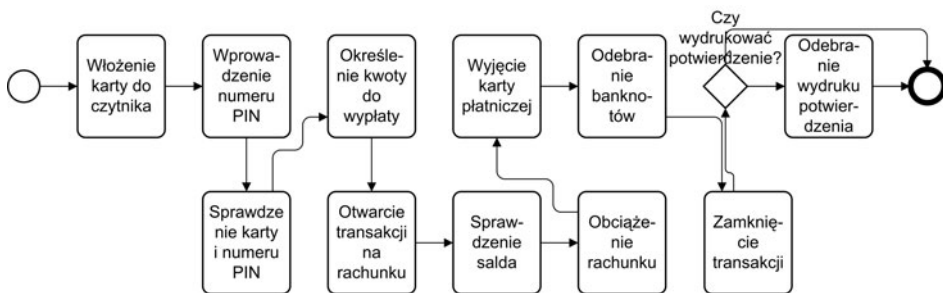
Rozróżnianie rodzajów przepływów jest jednym z pierwszych kroków dekompozycji procesu biznesowego. Bardzo często dopiero w chwili, gdy analityk próbuje jednoznacznie określić rodzaj przepływu, z jakim ma do czynienia, odkrywa, że początkowo prosta sekwencja działań jest bardziej złożona. Wynika to z tego, że przepływ między kolejnymi krokami procesu najczęściej wiąże się z przekazywaniem konkretnych danych między nimi, a nierzadko angażuje dodatkowych uczestników procesu, którzy realizują swoje zadania zupełnie niezależnie od modelowanego procesu.

Przykładowo jeśli rozpatrujemy proces obsługi bankomatu, to można powiedzieć, że składa się on z kilku prostych kroków: włożenie karty do czytnika, wprowadzenie PIN-u, wprowadzenie kwoty, opcjonalne zażądanie wydruku potwierdzenia, odebranie banknotów, wyjęcie karty (rysunek 5.1).



**Rysunek 5.1.** Model obsługi bankomatu — wersja wstępna

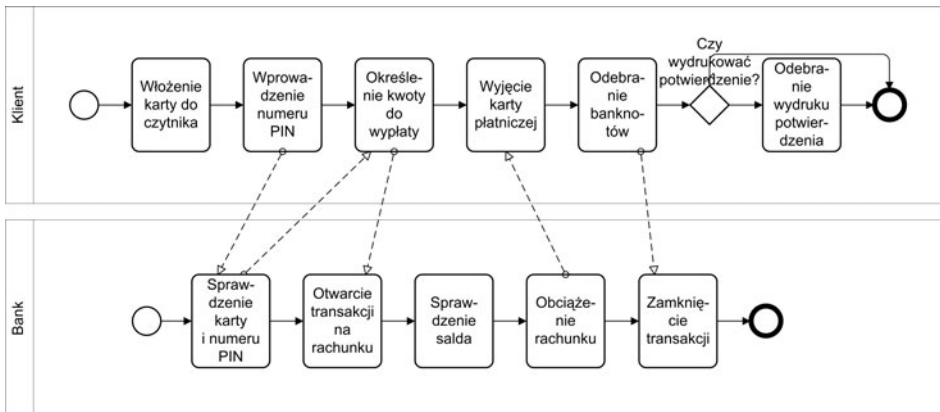
Jeśli jednak spróbujemy się dokładniej zastanowić, jak działa ten proces, możemy zauważyć, że jest to sekwencja interakcji pomiędzy klientem, bankomatem i jakimś systemem bankowym, który musi sprawdzić tożsamość klienta, środki dostępne na koncie, poprawność wprowadzonej kwoty itd. Co więcej, część tych kroków jest sekwencją działań (wprowadzenie PIN-u, wprowadzenie kwoty itd.), a część wiąże się z przesyłaniem jakichś danych między systemami (weryfikacja tożsamości na podstawie karty i PIN-u, sprawdzanie salda itd.). Można więc nieco rozwinąć wcześniej stworzony model (rysunek 5.2) i dodać kilka kroków, które są realizowane w procesie.



**Rysunek 5.2.** Model obsługi bankomatu — wersja rozwinięta

Pierwsza rzecz, którą można zauważyć w nowym modelu, to podział na czynności realizowane przez klienta banku (aktywności umieszczone na górze modelu) oraz czynności realizowane przez system transakcyjny banku (aktywności ułożone w dolnej części modelu). Taki model, choć jest poprawny formalnie w notacji BPMN, nie jest zbyt precyzyjny, ponieważ o ile czynności realizowane przez klienta banku faktycznie można ułożyć w sekwencję, czyli połączyć je przepływem sekwencji, o tyle punkty styku z systemem transakcyjnym mają nieco innych charakter. Są to bowiem czynności związane z komunikacją między systemami (tutaj między systemem bankomatu a systemem transakcyjnym). Dodatkowo między niektórymi aktywnościami możemy wprost określić przepływające dane oraz miejsca powstawania i przetwarzania tych danych. Tak więc choć jest to model nieco dokładniejszy niż poprzedni, nadal nie jest zbyt precyzyjny.

Rysunek 5.3 przedstawia fragment modelu procesu w notacji BPMN, w którym rozróżniono uczestników procesu. Jest to model tego samego procesu biznesowego, który przedstawiono wcześniej. Tym razem jednak przepływy są wyspecyfikowane dokładniej, ponieważ jawnie rozróżniono uczestników. Przykładowo między aktywnością wprowadzania numeru PIN oraz sprawdzania karty i numeru PIN zaprezentowano przepływ komunikatu.



**Rysunek 5.3.** Fragment modelu obsługi bankomatu — wersja szczegółowa

Jak już wcześniej wspomniano, wszystkie prezentowane w tym rozdziale wersje modeli są poprawne i zgodne ze standardem BPMN, jednak każdy z nich prezentuje inny poziom szczegółowości. To, który model jest w danej chwili najbardziej odpowiedni, zależy od tego, jakie jest jego przeznaczenie: poglądowy model raczej musi być uproszczony, żeby pełnił swoją funkcję, model implementacyjny z kolei musi zawierać maksymalnie dużo szczegółów. Więcej informacji na temat przeznaczenia modeli i poziomów dokładności zawarto w rozdziale 3., „Bramki — dzielenie i łączenie procesów”. W dalszej części tego rozdziału zaprezentowano możliwe rodzaje przepływów w notacji BPMN oraz wyjaśniono ich znaczenie. Dla każdego przepływu określono także reguły poprawnego łączenia, aby Czytelnik mógł zweryfikować poprawność modeli, które posiada lub sam tworzy.

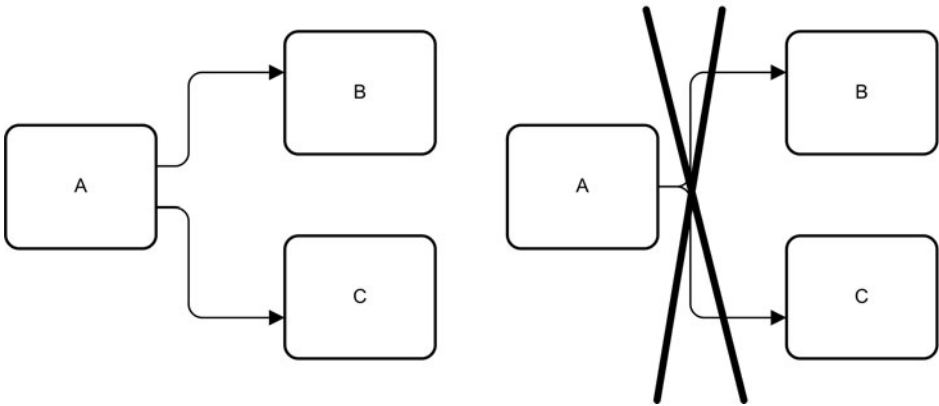
## 5.1. Przepływ sekwencji

**Przepływ sekwencji** w notacji BPMN jest reprezentowany przez linię ciągłą z górnym zamkniętym, wypełnionym (rysunek 5.4). Połączenie dwóch elementów diagramu w ten sposób określa ich następstwo w procesie. Modelując przepływ sekwencji, należy także mieć na uwadze kilka reguł opisanych poniżej.



**Rysunek 5.4.** Modelowanie przepływu sekwencji

Jeden przepływ sekwencji może łączyć tylko dwa elementy, a więc nie może mieć kilku wejść lub kilku wyjść, czyli zakończeń z grotem. Nie wyklucza to jednak sytuacji, w której z jednego elementu wychodzi więcej niż jeden przepływ sekwencji (rysunek 5.5), ani sytuacji, w której do jednego elementu diagramu wchodzi więcej niż jeden przepływ sekwencji.



**Rysunek 5.5.** Jeden przepływ sekwencji może łączyć tylko dwa elementy













Przepływ sekwencji może występować nie tylko między dwiema aktywnościami, ale także między innymi elementami notacji BPMN. W tabeli 5.1 określono, jakie elementy notacji BPMN można łączyć ze sobą poprzez przepływ sekwencji. Jeżeli jakiś element nie występuje w tabeli, oznacza to, że nie można go w żadnym przypadku łączyć z innymi elementami modelu za pośrednictwem przepływu sekwencji.

Przepływ sekwencji można wykorzystywać do łączenia wszystkich podstawowych elementów modelu BPMN, o ile połączenie to odbywa się w obszarze jednego *uczestnika*, czyli w jednej puli. Natomiast między różnymi uczestnikami procesu nie można stosować przepływów sekwencji, tylko przepływy komunikatu (rysunek 5.6).

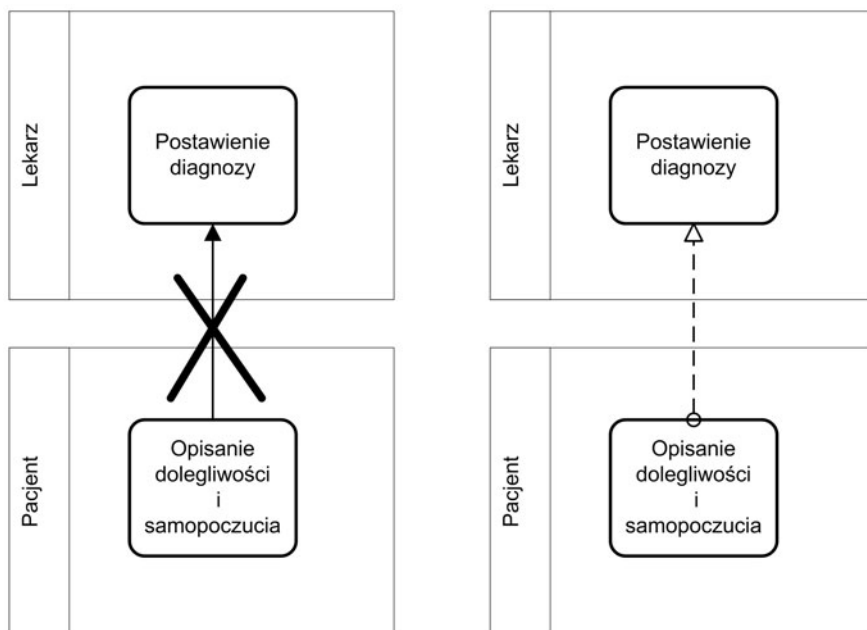
### 5.1.1. Warunkowy przepływ (conditional flow)

W przypadkach, gdy w modelowanym procesie przepływ sekwencji nie następuje bezwarunkowo, tzn. przejście w procesie z jednego kroku do kolejnego jest uzależnione od jakiegoś warunku, zastosowanie ma **warunkowy przepływ sekwencji**. Jest on reprezentowany przez linię ciągłą z grotem zamkniętym, wypełnionym po stronie elementu docelowego oraz tzw. diament, niewypełniony po stronie elementu źródłowego. Rysunek 5.7 przedstawia fragment modelu procesu przygotowywania delegacji, w którym przepływ (wybór ścieżki) zależy od dystansu podróży, a więc mamy do czynienia z dwoma warunkowymi przepływami sekwencji.

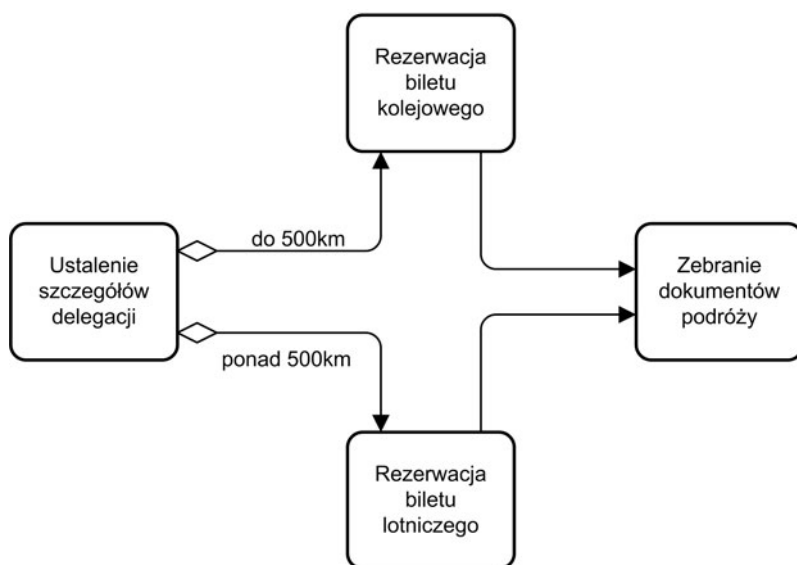
**Tabela 5.1.** Dozwolone łączenia przepływem sekwencji w notacji BPMN 2.0

	Zdarzenie początkowe	Zdarzenie pośrednie	Zdarzenie końcowe	Aktywność	Podproces	Brama
						
Zdarzenie początkowe 	Nie	Tak	Tak	Tak	Tak	Tak
Zdarzenie pośrednie 	Nie	Tak	Tak	Tak	Tak	Tak
Zdarzenie końcowe 	Nie	Nie	Nie	Nie	Nie	Nie
Aktywność 	Nie	Tak	Tak	Tak	Tak	Tak
Podproces 	Nie	Tak	Tak	Tak	Tak	Tak
Brama 	Nie	Tak	Tak	Tak	Tak	Tak

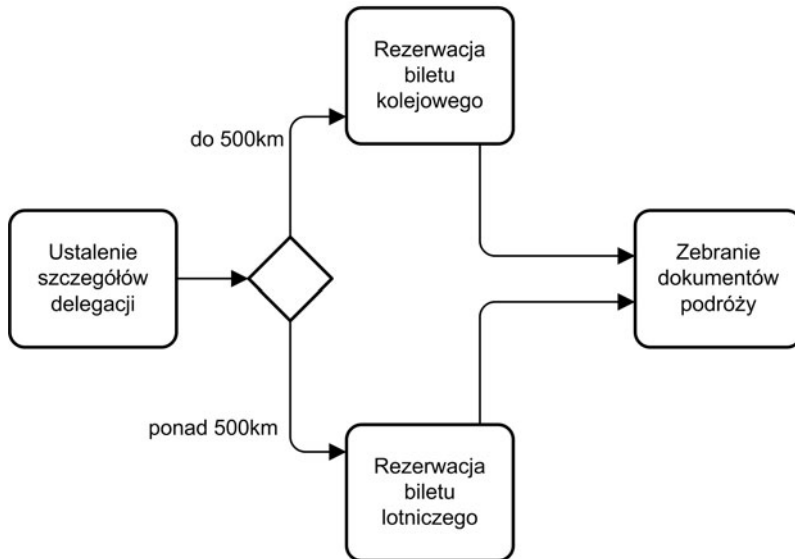
Rysunek 5.8 przedstawia model tego samego procesu, ale z użyciem bramy — w takim przypadku nie ma zastosowania diament po stronie elementu źródłowego, ponieważ nośnikiem tej informacji jest sama brama. Oba sposoby modelowania są zgodne ze standardem BPMN 2.0, niemniej należy pamiętać o tym, aby w pojedynczym modelu konsekwentnie stosować jeden ze sposobów zapisu. Nieuzasadnione mieszanie sposobów prezentacji warunkowych przepływów może utrudniać zrozumienie modeli. Więcej informacji na temat rozwidleń bez bram znajduje się w rozdziale 4., „Dzielenie i łączenie bez bramek”.



**Rysunek 5.6.** Między różnymi uczestnikami procesu stosuje się przepływ komunikatu



**Rysunek 5.7.** Warunkowy przepływ sekwencji bez bramy

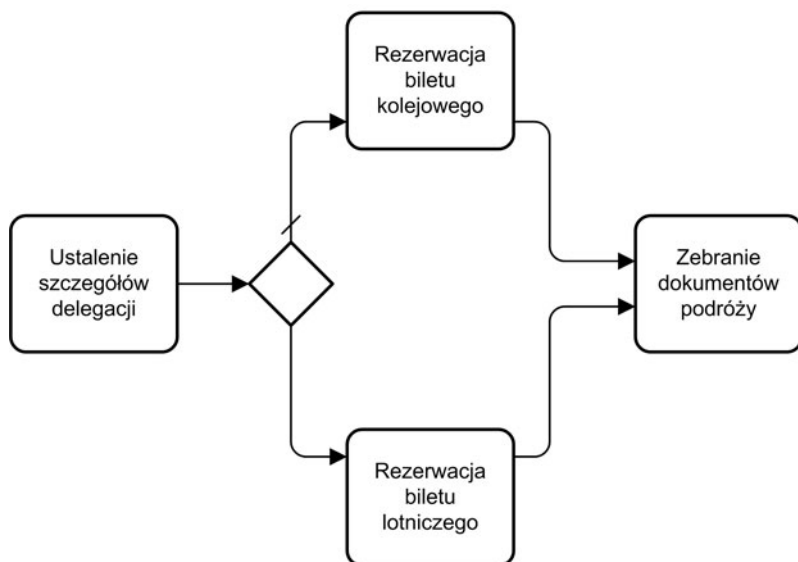


**Rysunek 5.8.** Warunkowy przepływ sekwencji z bramą

### 5.1.2. Domyślny przepływ (default flow)

W przypadku warunkowych przepływów sekwencji możliwe jest także oznaczanie **domyślnych przepływów**. Do tego celu wykorzystuje się domyślny przepływ sekwencji, który jest reprezentowany przez linię ciągłą z grotem zamkniętym, wypełnionym po stronie elementu docelowego oraz ukośnik po stronie elementu źródłowego. Przykładowo jeśli we wcześniej modelowanym procesie domyślnie rezerwowany jest bilet kolejowy, można użyć domyślnego przepływu sekwencji (rysunek 5.9).

Jaka jest różnica i kiedy stosować poszczególne rodzaje warunkowych przepływów sekwencji? Standard BPMN nie narzuca tutaj żadnej reguły. Natomiast warto zwrócić uwagę na praktyczne uzasadnienie każdego zapisu. Przepływy warunkowe bez bramy (czyli z własnymi „diamantami”) są wygodne do stosowania wtedy, gdy już w chwili modelowania wiemy, jakie warunki logiczne muszą być spełnione, żeby dana ścieżka w procesie była możliwa. Często jednak na początku analizy wiemy tylko, że jest rozwidlenie, ale niekoniecznie dokładnie znamy wartości warunków logicznych. Czasami wiemy jedynie, która ścieżka jest domyślna (na ogół jest to rozumiane jako „najczęściej” wybierana ścieżka w procesie) — wtedy możemy zaprezentować domyślny przepływ sekwencji. Oczywiście jeśli dążymy do wdrożenia danego procesu w silniku procesów biznesowych, musimy znać konkretne wartości warunków dla każdej ścieżki; nie wystarczy tylko stwierdzenie, że któraś ścieżka jest domyślna, a inne nie. Ponownie ważny jest cel, dla którego



**Rysunek 5.9.** Domyślny przepływ sekwencji

opracowywany jest dany model procesu biznesowego. Warto też zwrócić uwagę, że poszczególne sposoby opisywania procesów biznesowych (z bramami lub bez) nie zawsze są równoważne. Więcej informacji na ten temat znajduje się w rozdziale 4.

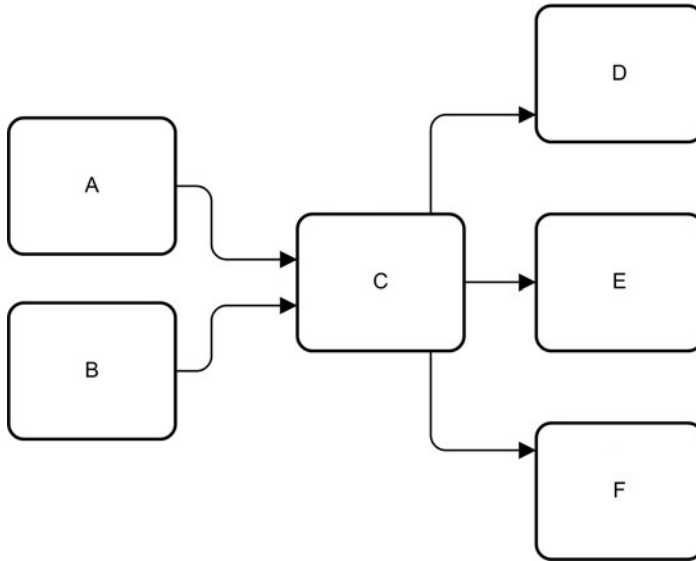
### 5.1.3. Swobodny przepływ (uncontrolled flow)

Swobodny przepływ sekwencji to przepływ nieobarczony żadnym warunkiem ani niepołączony z jakąkolwiek bramą (rysunek 5.10). Taki przepływ występuje zawsze pomiędzy dwiema czynnościami połączonymi w sekwencję. Ciekawszy jest jednak przypadek, gdy swobodne przepływy sekwencji są zastosowane tak, jak na rysunku poniżej, czyli gdy do jednej czynności wchodzi więcej niż jeden swobodny przepływ i z jednej czynności wychodzi więcej niż jeden swobodny przepływ sekwencji. Wtedy czynność C rozpoczyna się po zakończeniu czynności A lub B. Jeśli więc najpierw zakończy się czynność A, a potem B, to czynność C zostanie wykonana dwa razy: po zakończeniu A, a także po zakończeniu B. Natomiast D, E i F zostaną wywołane razem po zakończeniu C.

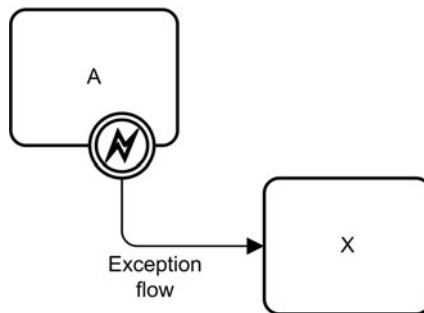
### 5.1.4. Przepływ wyjątku (exception flow)

Przepływ wyjątku — lub też przepływ wyjątkowy — (rysunek 5.11) następuje poza normalnym przebiegiem procesu i zależy od tego, czy nastąpiło pośrednie zdarzenie krawędziowe. Sformułowanie „następuje poza normalnym przebiegiem”





**Rysunek 5.10.** *Przepływy swobodne*



**Rysunek 5.11.** *Przepływ wyjątku*

oznacza, że nie jest to przepływ aktywny w normalnych warunkach, a jedynie w wyjątkowych sytuacjach, identyfikowanych przez odpowiednie zdarzenie krawędziowe. Więcej o zdarzeniach krawędziowych można przeczytać w rozdziale 9., „Czynności i podprocesy”.

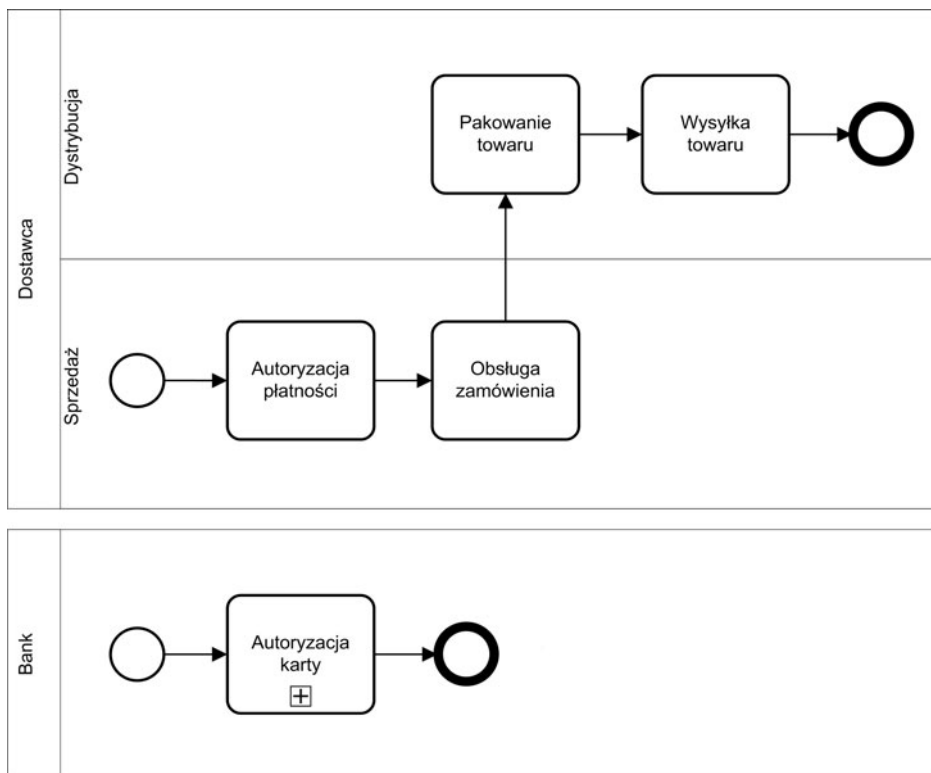
### 5.1.5. Przepływ normalny (normal flow)

Przepływ normalny to każdy przepływ sekwencji, który nie jest przepływem wyjątku, czyli — według standardu BPMN 2.0 — nie wychodzi z krawędziowego zdarzenia pośredniego. Choć ta definicja może wydawać się nieco dziwna i mało sensowna, w dalszej części książki w wielu miejscach będzie używane pojęcie przepływu normalnego w celu wyraźnego podkreślenia, że jakiś fragment procesu

biznesowego jest lub nie jest realizowany w normalnym przebiegu procesu. Może się bowiem okazać, że jakiś podproces jest realizowany poza normalnym przepływem i jednocześnie wcale nie jest to dla nas sytuacja „wyjątkowa”.

## 5.2. Przepływ komunikatu

W przypadku modelowania przepływów między różnymi uczestnikami procesu biznesowego, czyli — w notacji BPMN — między pulami, zastosowanie ma **przepływ komunikatu**. Jest on reprezentowany przez linię przerywaną (kreskowaną) z dwoma grotami: małym okręgiem po stronie elementu nadającego komunikat oraz zamkniętym, niewypełnionym trójkątem po stronie elementu odbierającego komunikat. Rysunek 5.12 przedstawia proces sprzedaży w podziale na dostawcę oraz bank. Rola banku według tego modelu ogranicza się jedynie do autoryzowania karty kredytowej w celu autoryzowania płatności. Między krokiem *Autoryzacja płatności* realizowanym przez dział sprzedaży po stronie dostawcy a krokiem *Autoryzacja karty kredytowej* po stronie banku następuje wymiana komunikatów.



**Rysunek 5.12.** Modelowanie przepływu komunikatów

Należy pamiętać o tym, że przepływ komunikatów nie może być stosowany do łączenia obiektów w obrębie tej samej puli BPMN. Jest on zarezerwowany dla komunikacji między odrębnymi uczestnikami procesu. Jeśli natomiast modelujemy przepływy w obrębie jednego uczestnika, nie możemy używać przepływu komunikatów.

W tabeli 5.2 określono, jakie elementy notacji BPMN można łączyć ze sobą poprzez przepływ komunikatów. Jeżeli jakiś element nie występuje w tabeli, oznacza to, że nie można go w żadnym przypadku łączyć z innymi elementami modelu za pośrednictwem przepływu komunikatów. Uwaga: zdarzenie pośrednie komunikatu wysłanego (1. kolumna, 3. wiersz tabeli) to nie to samo co zdarzenie pośrednie komunikatu odebranego (3. kolumna, 1. wiersz tabeli)! Więcej informacji o zdarzeniach znajduje się w rozdziale 8., „Zdarzenia”.

Dodatkowo przepływy komunikatów w modelach BPMN 2.0 mogą być opisywane przez nazwę komunikatu. Rysunek 5.13 przedstawia fragment modelu z opisanym przepływem komunikatu. Choć opis przepływu komunikatu nie jest obowiązkowy, ułatwia zrozumienie modelu, dlatego warto zawsze określić nazwę przepływającego komunikatu.

Choć początkowo może się wydawać, że modelowanie przepływów komunikatów jest mniej istotne od modelowania przepływów sekwencji, w praktyce jest dokładnie odwrotnie. Najczęściej bowiem problemy analityczne i projektowe dotyczą komunikacji między różnymi uczestnikami procesu biznesowego. Są to sytuacje, w których zespół projektujący proces musi dokładnie wyspecyfikować komunikaty przekazywane pomiędzy różnymi systemami. Także ze względu na naturalną tendencję do automatyzacji procesów biznesowych kluczowe fragmenty modeli procesów biznesowych dotyczą przepływów komunikatów. Im większy stopień automatyzacji, tym większa część przepływów sekwencji przestaje być już domeną procesów biznesowych, a zaczyna być domeną systemów oprogramowania, czyli algorytmów.

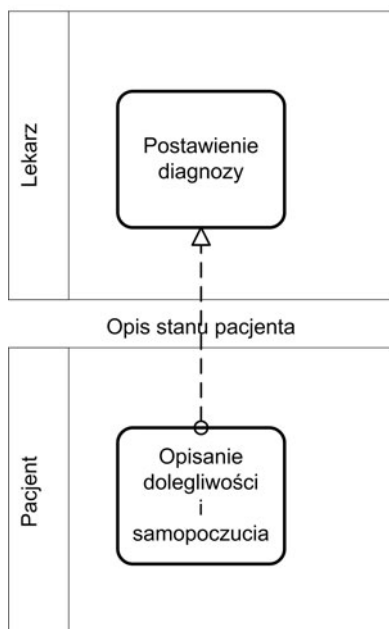
### 5.3. Asocjacje

Kolejny rodzaj relacji między elementami modelu BPMN to **asocjacja**. Asocjacja może być stosowana do prezentowania: notatek tekstowych na diagramie, przepływów obiektów danych w procesie lub powiązania elementów modelu z tzw. artefaktami. Asocjacja jest reprezentowana przez linię kropkowaną. Może ona być zakończona grotem (**asocjacja skierowana**), jeśli zasadne jest specyfikowanie kierunku asocjacji, a w przypadku modelowania przepływu **obiektu danych** (reprezentowanego przez prostokąt z „zagiętym” prawym górnym rogiem) grot jest

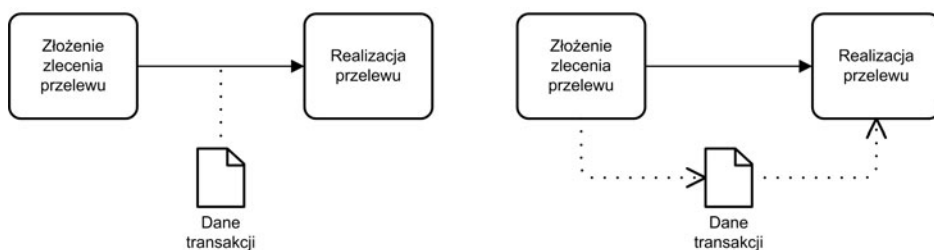
**Tabela 5.2.** *Dozwolone łączenia przepływem komunikatów w notacji BPMN 2.0*

	Zdarzenie początkowe komunikatu	Zdarzenie pośrednie komunikatu odebranego	Zdarzenie końcowe komunikatu	Pula (Uczestnik procesu)	Aktywność	Podproces
Zdarzenie początkowe komunikatu	Nie	Nie	Nie	Nie	Nie	Nie
Zdarzenie pośrednie komunikatu wysłanego	Nie	Tak	Nie	Tak	Tak	Tak
Zdarzenie końcowe komunikatu	Nie	Tak	Nie	Tak	Tak	Tak
Pula (Uczestnik procesu)	Nie	Tak	Nie	Tak	Tak	Tak
Aktywność	Nie	Tak	Nie	Tak	Tak	Tak
Podproces	Nie	Tak	Nie	Tak	Tak	Tak

obowiązkowy (rysunek 5.14, przykład po prawej stronie). Ewentualnie możliwe jest także łączenie przepływu sekwencji z obiektem danych za pośrednictwem asocjacji (rysunek 5.14, przykład po lewej stronie). Więcej na temat obiektów danych można przeczytać w rozdziale 6., „Obiekty danych”.

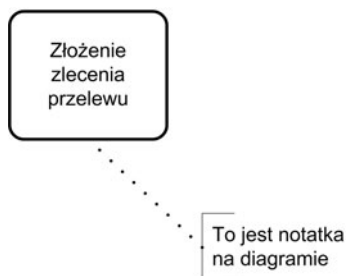


**Rysunek 5.13.** *Przeptyw komunikatu z opisem*



**Rysunek 5.14.** *Możliwe sposoby używania asocjacji do modelowania przepływu danych*

Asocjacje można także wykorzystywać do zaznaczania elementów modelu, których dotyczą notatki umieszczone na diagramie (rysunek 5.15).



**Rysunek 5.15.** *Notatka na diagramie BPMN 2.0*



Bardzo często gdy modelujemy proces biznesowy, poza samym jego przebiegiem zależy nam na tym, żeby zaprezentować także obiekty lub dane, które są przetwarzane w tym procesie. Może to wynikać z faktu, że chcemy np. ustalić, czy jakaś czynność wymaga pewnych informacji wejściowych lub czy jej wykonanie powoduje, że jakiś obiekt się zmienia lub właśnie powstaje. Przy czym poprzez „obiekt” można rozumieć nie tylko dokument czy zbiór informacji, ale także coś, co istnieje fizycznie. Np. jeśli rozważamy proces montowania samochodów, to samochód będzie obiektem wyjściowym; jeśli rozważamy proces zakładania rachunku bankowego, to obiektem przetwarzanym w trakcie tego procesu będzie plastikowa karta płatnicza itd. Można więc powiedzieć, że modelowanie procesów biznesowych bez modelowania przetwarzanych obiektów byłoby bardzo ubogie, a w niektórych przypadkach wręcz pozbawione sensu. Na szczęście standard BPMN 2.0 umożliwia modelowanie obiektów przetwarzanych w procesach biznesowych i ogólnie określa je terminem *Item-Aware Elements*<sup>1</sup>. Zaliczają się do nich: *obiekty danych*, *odwołania do obiektów danych*, *magazyny danych*, *parametry* oraz *dane wejściowe* i *dane wyjściowe*.

Warto jednak zauważyć, że modelowanie obiektów przetwarzanych w procesach biznesowych w standardzie BPMN 2.0 ogranicza się jedynie do ich wskazywania poprzez nazwę lub ewentualnego wskazywania ich typu albo roli w procesie biznesowym. Standard nie przewiduje natomiast opisywania struktury tych obiektów; ewentualnie sugeruje pewne proste schematy XSD do poszczególnych typów obiektów i użycie języka XPath do wyciągania informacji o tych typach. Pozostawia jednak w tym zakresie wolną rękę dostawcom narzędzi. Innymi słowy, jeśli nabędziemy pakiet narzędzi klasy BPMS od jakiegoś producenta, może się okazać, że obiekty przetwarzane w modelowanych procesach biznesowych będziemy

---

<sup>1</sup> Przetłumaczenie tego na język polski jest trudne. W bezpośrednim tłumaczeniu *Item-Aware Element* to „element świadomy rzeczy (szczegółu)”, w praktyce chodzi po prostu o element, który może (ale nie musi) mieć opisaną strukturę. Innymi słowy, może to być cokolwiek.

musieli opisywać w sposób specyficzny dla tego producenta. Zarówno schematy XSD, jak i wyrażenia XPath do wyciągania z nich potrzebnych informacji są szczegółowo opisane w specyfikacji BPMN 2.0 dostępnej na stronach *www.omg.org*.

## 6.1. Modelowanie obiektów i magazynów danych

W standardzie BPMN 2.0 obiekty danych są reprezentowane przez ikonę kartki z zagiętym prawym górnym narożnikiem oraz nazwą obiektu pod spodem (rysunek 6.1).



**Rysunek 6.1.** *Obiekt danych*

Każdy obiekt może mieć tzw. stan (*data state*) określony nazwą. Przykładowo jeśli obiektem jest *Faktura*, to może ona mieć następujące stany: nowa, odrzucona, zaakceptowana, opłacona. Przy czym w danej chwili pojedynczy obiekt może znajdować się tylko w jednym stanie, a więc nasza *Faktura* nie może być jednocześnie np. „opłacona” i „nowa”. Dodatkowo poza obiektami na diagramach mogą pojawić się odwołania do nich (*object data reference*). Posługując się odwołaniami do obiektów oraz nazwami stanów, możemy w modelach procesów biznesowych zawierać nieco więcej informacji niż tylko fakt, że jakiś obiekt jest angażowany w dany proces. Aby to zrobić — zgodnie ze specyfikacją BPMN 2.0 — musimy zastosować zapis, w którym stan jest wskazywany w nawiasach prostokątnych (podobnie jak w UML-u przy diagramach aktywności z obiektami): `<Nazwa obiektu danych> [<Nazwa stanu>]`. W tym przypadku prezentacja graficzna odwołania do obiektu jest identyczna, różna jest jedynie nazwa pod ikoną (rysunek 6.2).



**Rysunek 6.2.** *Odwolanie do obiektu ze wskazanym stanem*

W przypadku gdy dane przetwarzane w procesie biznesowym są przechowywane w określonym miejscu, niezależnie od tego, czy proces właśnie działa, czy nie, należy zastosować magazyn danych (*data store*). W notacji BPMN 2.0 magazyn danych jest reprezentowany przez piktogram przedstawiony na rysunku 6.3.





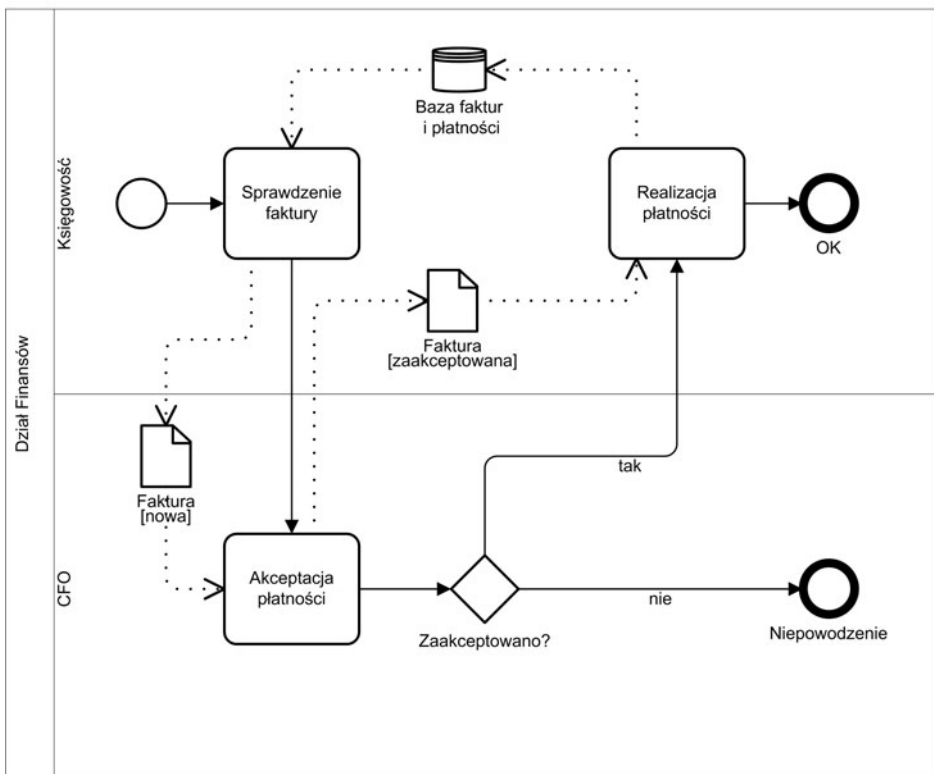
**Rysunek 6.3.** *Magazyn danych*

Czytelnik może się zastanawiać, jaka jest w takim razie różnica między obiektem danych a magazynem danych oraz o co chodzi z tym, że dane mogą zależeć od tego, czy proces działa, czy nie. Otóż niektóre obiekty istnieją tylko w trakcie działania procesu, a inne niezależnie od tego, czy dany proces działa, czy nie. Przykładem obiektu, który istnieje tylko w kontekście określonego procesu, może być „bilet lotniczy”. Jest to obiekt, który powstaje w chwili, gdy pasażer kupuje bilet. W trakcie całego procesu „transportu pasażera” istnieje on realnie i ma duże znaczenie dla tego procesu. Niemniej obiekt ten nie istnieje, zanim pasażer go nie kupi, i nie ma też znaczenia w chwili, gdy pasażer go wykorzysta. Oczywiście można przypuszczać, że linie lotnicze mają jakiś rejestr biletów, ale bilet naszego pasażera nie jest tym samym obiektem co zapis w rejestrze sprzedanych biletów. Z drugiej strony, właśnie ów rejestr biletów jest dobrym przykładem magazynu danych. Nawet jeśli lot się nie odbędzie lub jeśli pasażer zgubi bilet i proces „transportu pasażera” nie nastąpi, w owym rejestrze będzie nadal informacja o tym, że taki bilet został sprzedany.

Innym dobrym przykładem ilustrującym różnicę między obiektami danych i magazynem danych jest proces akceptacji faktur (rysunek 6.4). W procesie tym dział księgowości sprawdza faktury i odpowiada za realizację zaakceptowanych płatności. Natomiast szef działu finansowego akceptuje bądź odrzuca faktury, a dokładniej: płatności wynikające z tych faktur. Faktury istnieją w magazynie danych niezależnie od tego, czy ten proces w ogóle został wykonany, czy nie. Niemniej w trakcie jego realizacji między czynnościami występują przepływy obiektów. Obiektami tymi są faktury, których stan zmienia się z czynności na czynność. Ostatecznie jeśli płatność zostanie zaakceptowana, do magazynu danych „Baza faktur i płatności” trafia informacja o nowej płatności skojarzonej z odpowiednią fakturą.

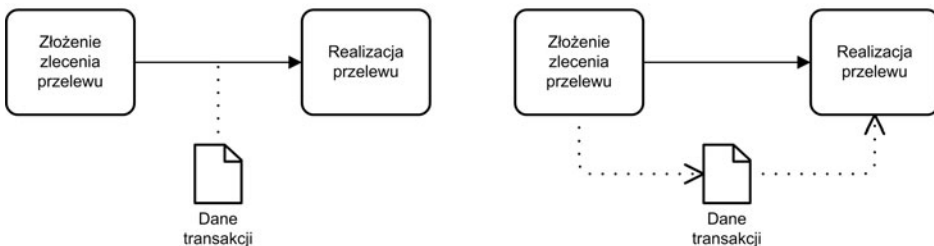
## 6.2. Przepływ obiektów w procesach

Wiedząc, jak modeluje się w notacji BPMN 2.0 obiekty danych, możemy przyjąć się dokładniej temu, jak prezentować przepływ obiektów w procesach. Nie wystarczy bowiem, że na modelu umieścimy kilka obiektów, konieczne jest także wskazanie, które czynności potrzebują tych obiektów, a które je modyfikują lub



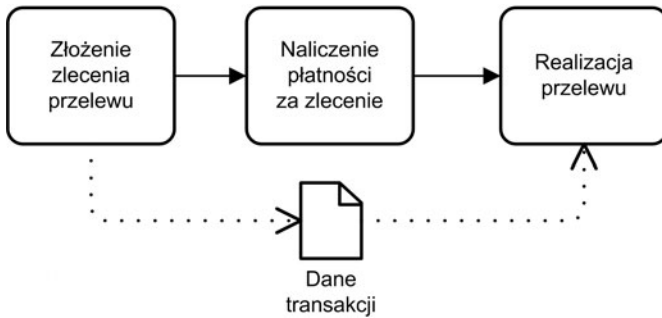
**Rysunek 6.4.** Proces akceptacji faktur

tworzą. W tym celu musimy zastosować asocjacje i możemy to zrobić na dwa sposoby (rysunek 6.5): albo łącząc asocjacją bez grota obiekt z przepływem sekwencji, albo używając asocjacji skierowanych.



**Rysunek 6.5.** Przepływy obiektów (równoznaczne zapisy)

Oczywiście wygodniejsze i krótsze jest używanie pierwszej wersji przepływu obiektów, niestety nie zawsze jest to możliwe (rysunek 6.6). Używanie asocjacji skierowanych jest rozwiązaniem w szczególności, jeśli mamy do czynienia z podpro-



**Rysunek 6.6.** *Przepływ obiektu*

cesem *ad hoc*, w którym w ogóle może nie być przepływów sekwencji. Przykładowy proces *ad hoc* z przepływami obiektów znajduje się w rozdziale 9., „Czynności i podprocesy”.

### 6.3. Wejścia i wyjścia danych

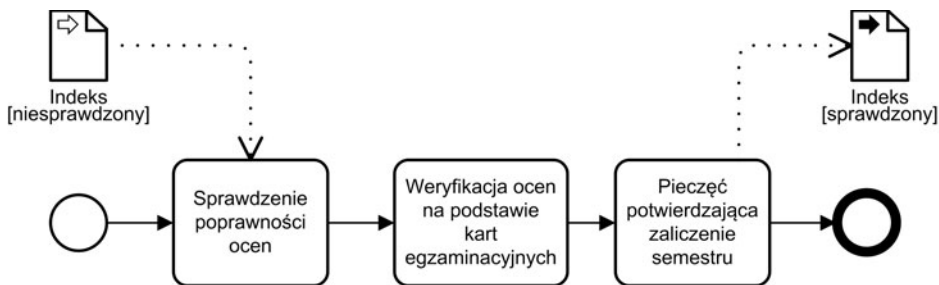
Realizacja praktycznie każdego procesu biznesowego wymaga jakichś danych wejściowych, z drugiej strony, praktycznie każdy proces biznesowy daje jakiś wynik. A więc niektóre obiekty mają szczególne znaczenie, ponieważ są kluczowe dla uruchomienia danego procesu albo są jego istotnym „produktem”. W związku z tym obiekty przetwarzane w procesach biznesowych można podzielić na dwie grupy. Pierwszą grupę stanowią obiekty, które są „wewnętrznie” wykorzystywane w trakcie realizacji procesu. Będą to omówione już wcześniej: obiekty, odwołania do obiektów i magazyny danych. Drugą grupę stanowią obiekty „zewnętrzne” wejściowe i wyjściowe, a dokładniej: wejścia i wyjścia danych (*data input*, *data output*). Są one reprezentowane tak samo jak obiekty danych, z tym że muszą zawierać dodatkowo ikonę sygnalizującą, że nie są to zwykłe obiekty danych, ale wejścia lub wyjścia (rysunek 6.7).



**Rysunek 6.7.** *Od lewej: wejście danych, wyjście danych*

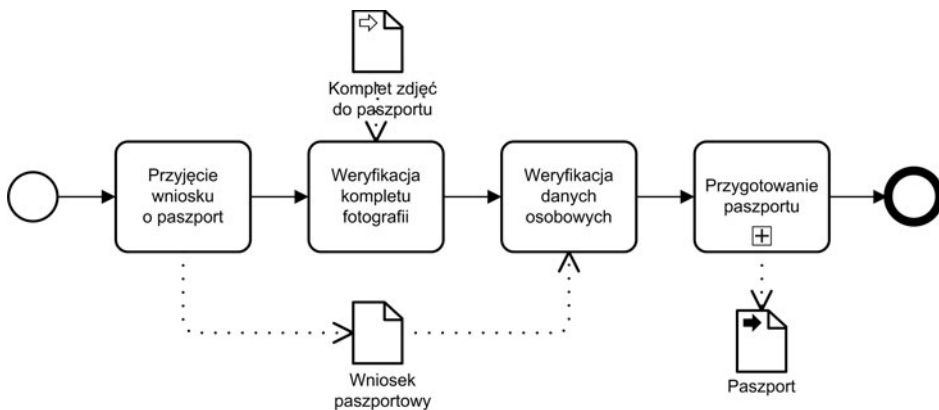
Może się też zdarzyć, że ten sam obiekt występuje jednocześnie w obu rolach, tzn. jest obiektem wejściowym lub wyjściowym i jednocześnie jest przetwarzany w trakcie procesu. Przykładowo w procesie rejestrowania studentów na kolejny

semestr obiektem niezbędnym do realizacji jakichkolwiek działań jest indeks studenta. Oceny wpisane do indeksu muszą być sprawdzone i potwierdzone przez dziekanat. A więc jest to obiekt wejściowy. Z drugiej strony, po pozytywnej weryfikacji studenta i jego ocen dziekanat wbija do indeksu pieczęć potwierdzającą zaliczenie poprzedniego semestru. Zatem jest to także obiekt wyjściowy, ponieważ proces nie tylko „czerpie” z niego informacje, lecz także je w nim „umieszcza”. Na diagramie BPMN 2.0 będą to dwa obiekty o tej samej nazwie, jednak z różnymi ikonami i ewentualnie dodatkowo rozdzielone nazwą stanu w nawiasach prostokątnych (rysunek 6.8).



**Rysunek 6.8.** *Proces z wejściem i wyjściem danych*

Podsumowując, wejścia i wyjścia danych to także obiekty danych, tyle że w odróżnieniu od zwykłych obiektów muszą one istnieć, zanim rozpocznie się proces (wejścia) lub po tym, jak się zakończy (wyjścia). Warto zauważyć, że nie muszą to być obiekty, które są wykorzystywane na początku danego procesu lub wytwarzane pod koniec. To nie ma znaczenia. Istotne jest to, że muszą istnieć przed procesem lub po nim, dlatego równie poprawny będzie model procesu wydawania paszportu z rysunku 6.9.



**Rysunek 6.9.** *Proces wydawania paszportu*

## 6.4. Kolekcje danych

Czasami mamy do czynienia nie tyle z pojedynczym obiektem, co z całą kolekcją obiektów tego samego typu. W takim przypadku możemy zastosować znacznik kolekcji (rysunek 6.10).

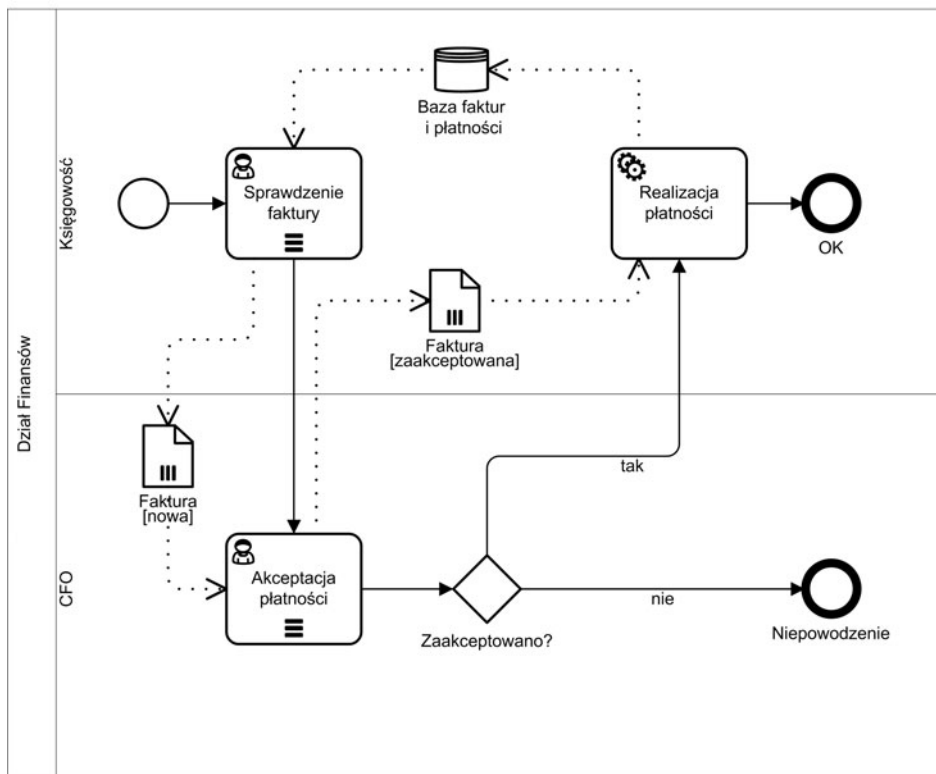


**Rysunek 6.10.** Odwołanie do obiektu ze znacznikiem kolekcji

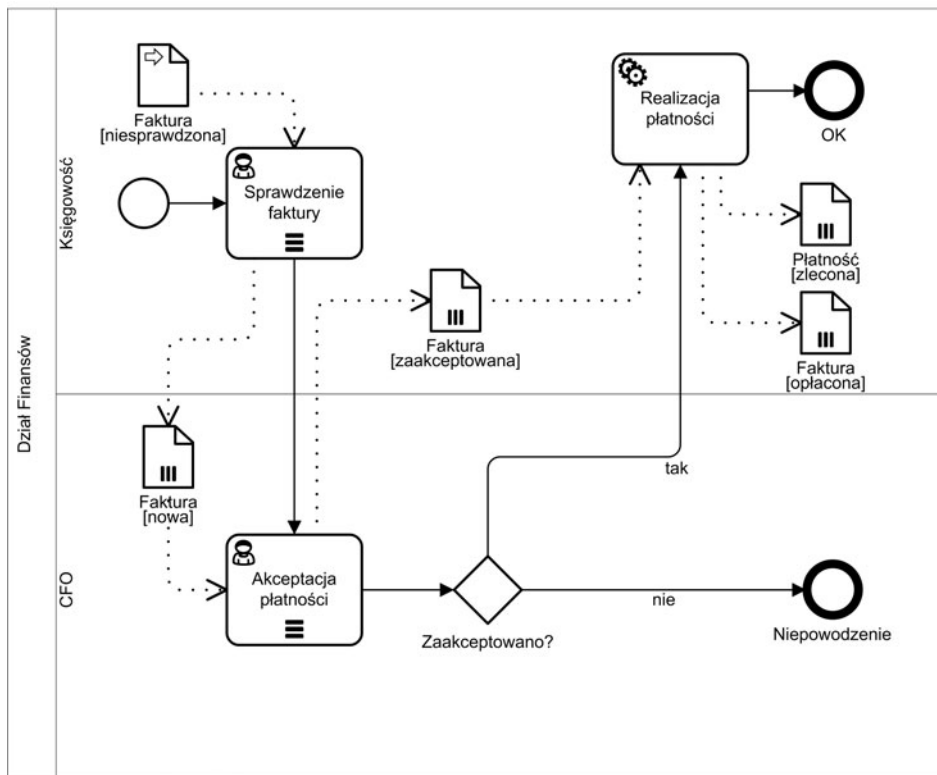
Warto zauważyć, że znacznika kolekcji możemy używać w odniesieniu do: obiektów danych, odnośników do obiektów danych, wejść i wyjść — przy czym nie jest to jednoznacznie określone w specyfikacji BPMN 2.0. Znacznik ten występuje jednak w przykładowych modelach opublikowanych przez OMG wraz ze specyfikacją BPMN 2.0 w dokumencie o nazwie *BPMN 2.0 by Example*. Możemy więc nieco zmodyfikować wcześniej przedstawiony przykładowy model procesu akceptacji faktur (płatności), przyjmując, że cały proces jest realizowany nie dla pojedynczych faktur, ale dla całych kolekcji (rysunek 6.11).

Idąc dalej, zamiast prezentować zewnętrzny magazyn danych *Baza faktur i płatności*, możemy zaprezentować konkretne wejścia i wyjścia tego procesu. A więc na wejściu będziemy mieli kolekcję faktur do sprawdzenia, a na wyjściu kolekcję faktur i zlecenia płatności. Rysunek 6.12 przedstawia zmodyfikowaną wersję naszego modelu.

Na marginesie: na przykładzie tego modelu widać, z czego może wynikać sytuacja, w której księgowość mówi, że faktura jest opłacona, a nasz partner nadal narzeka, że nie dostał jeszcze pieniędzy za swoją usługę. Czym innym bowiem jest płatność, a czym innym faktura. Jeśli nasza firma zleca płatność za fakturę, to czas realizacji tej płatności może zależeć np. od procedur banku, który obsługuje finanse danej firmy. Przykład ten pokazuje także, że pewne istotne niuanse można odkryć dopiero, gdy zejdzie się na poziom obiektów przetwarzanych w procesach biznesowych.



**Rysunek 6.11.** Proces akceptacji kolekcji faktur (wersja 1)



**Rysunek 6.12.** Proces akceptacji kolekcji faktur (wersja 2)





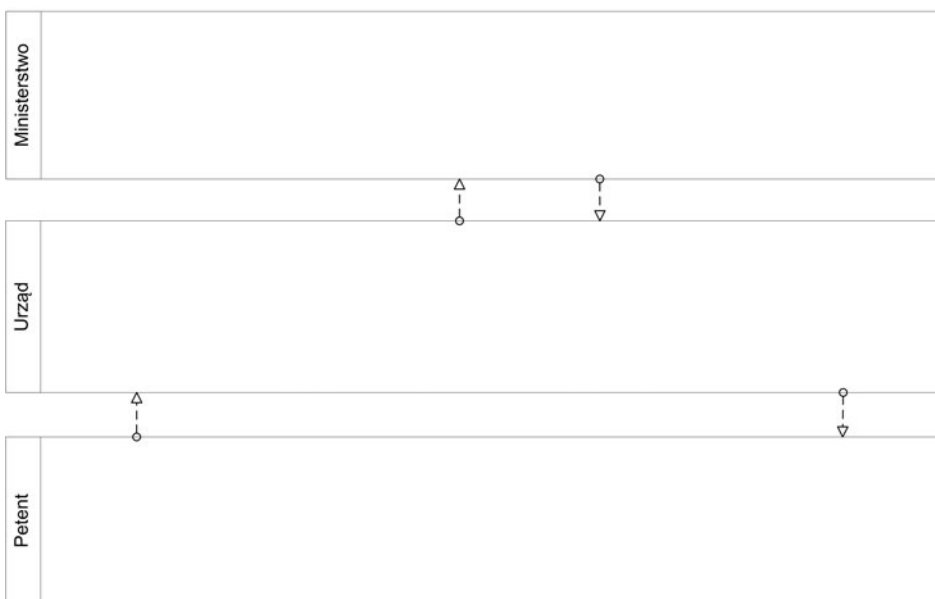
Modelowanie procesów biznesowych nie zawsze musi polegać na tym, że rozpisuje się szczegółowy przebieg procesu na czynności. Czasami okazuje się, że zanim przystąpimy do szczegółowego modelowania, chcemy najpierw uzyskać „ogólny” obraz procesu biznesowego. Przy czym poprzez ogólny obraz procesu należy rozumieć to, kto w ogóle ten proces realizuje oraz jakie zachodzą w nim interakcje. Bardzo często bowiem okazuje się, że sam proces, a dokładniej to, z jakich czynności się składa, nie jest dla nas tak istotne jak to, jakie jednostki organizacyjne są zaangażowane w jego realizację i jakie przyjmują w nim role. Jeśli chcemy np. zrozumieć proces obsługi zapytań o dofinansowanie w Krajowym Urzędzie Pracy, to zanim rozpoczniemy ustalanie sekwencji czynności w tym procesie, wypadłoby najpierw się zorientować, kto go w ogóle realizuje i jakie informacje w nim przepływają. W tym celu właśnie w standardzie BPMN 2.0 zdefiniowano kolaboracje oraz wprowadzono diagramy konwersacji.

## 7.1. Modelowanie kolaboracji

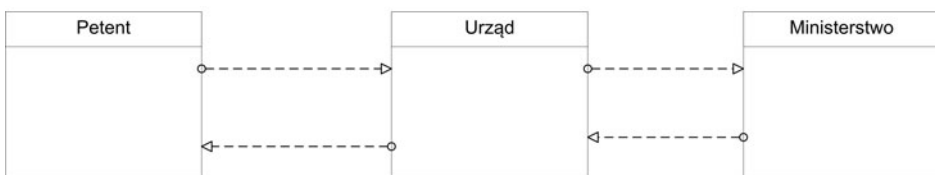
Kolaboracją w notacji BPMN 2.0 jest zbiór kilku (co najmniej dwóch) pul reprezentujących uczestników danego procesu biznesowego. Komunikacja między uczestnikami jest opisywana za pomocą przepływów komunikatów (patrz rozdział 5., „Przepływy”), które mogą, ale nie muszą być nazwane (rysunek 7.1).

Taki model pozwala na bardzo schematyczne zaprezentowanie tego, kto uczestniczy w procesie i ewentualnie jak komunikuje się z innymi uczestnikami. Warto wiedzieć, że pule reprezentujące uczestników procesu biznesowego mogą być ustawione pionowo (rysunek 7.2). Tak rozmieszczone pule do złudzenia przypominają klasy z UML-a.

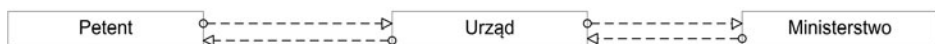
Dodatkowo standard BPMN 2.0 dopuszcza uproszczoną prezentację pul, jeśli są puste, tzn. nie zawierają wewnątrz żadnego procesu. Uproszczenie polega wtedy na tym, że nazwa uczestnika może być umieszczona w dowolnym miejscu puli i nie trzeba stosować linii oddzielającej (rysunek 7.3).



**Rysunek 7.1.** Kolaboracja BPMN

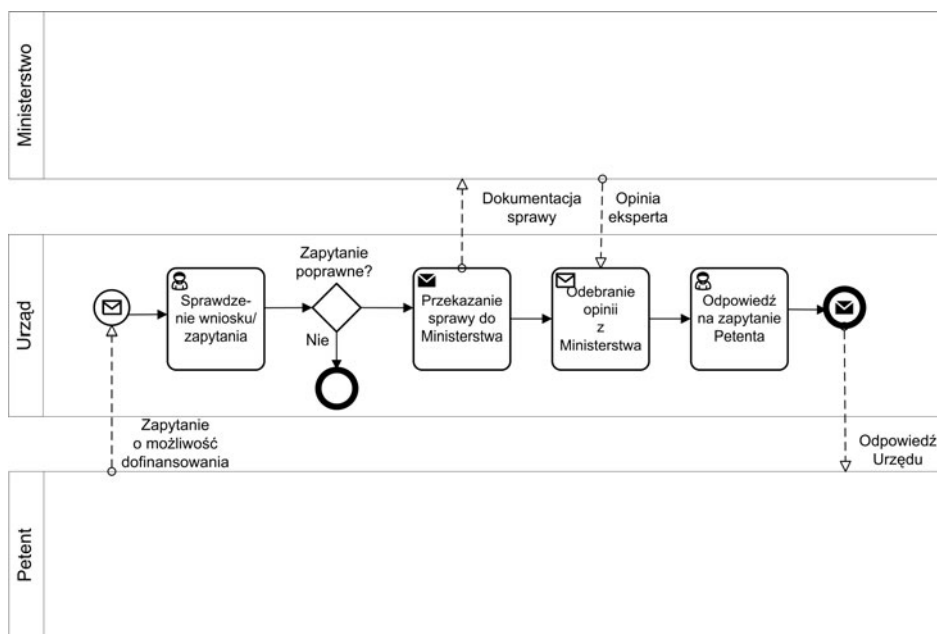


**Rysunek 7.2.** Kolaboracja BPMN z pulami w pionie



**Rysunek 7.3.** Uproszczone prezentowanie pustych pul

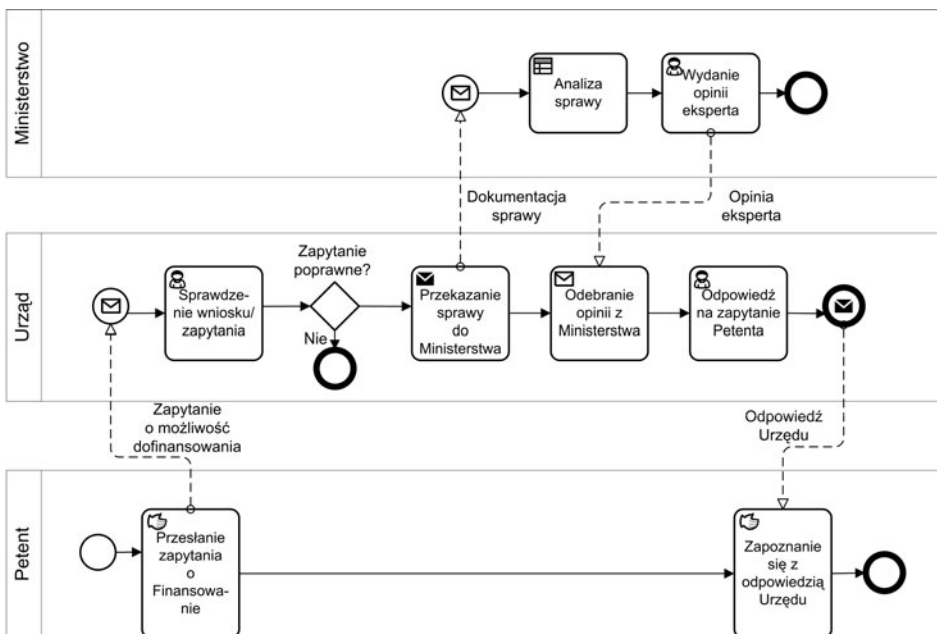
Wróćmy jednak do kolaboracji w BPMN 2.0. Jeśli chcielibyśmy pokazać jakiś proces biznesowy wraz z jego kontekstem, kolaboracje razem ze zwykłymi diagramami procesów są doskonałym rozwiązaniem. W ten sposób ukrywamy rzeczy w danej chwili nieistotne, a jednocześnie sygnalizujemy, że analizowany przez nas proces w jakiś sposób od nich zależy. Rysunek 7.4 przedstawia model, w którym prezentowane są tylko czynności realizowane przez *Urząd*, pominięte są natomiast te czynności, które realizuje *Ministerstwo* lub *Petent*. Dodatkowo komunikaty wymieniane między uczestnikami tym razem są nazwane. Dzięki temu, choć nadal nie wiemy, co dokładnie robią w tym procesie *Ministerstwo* oraz *Petent*, mamy już dużo więcej informacji niż w pierwotnej wersji tej kolaboracji (rysunek 7.1).



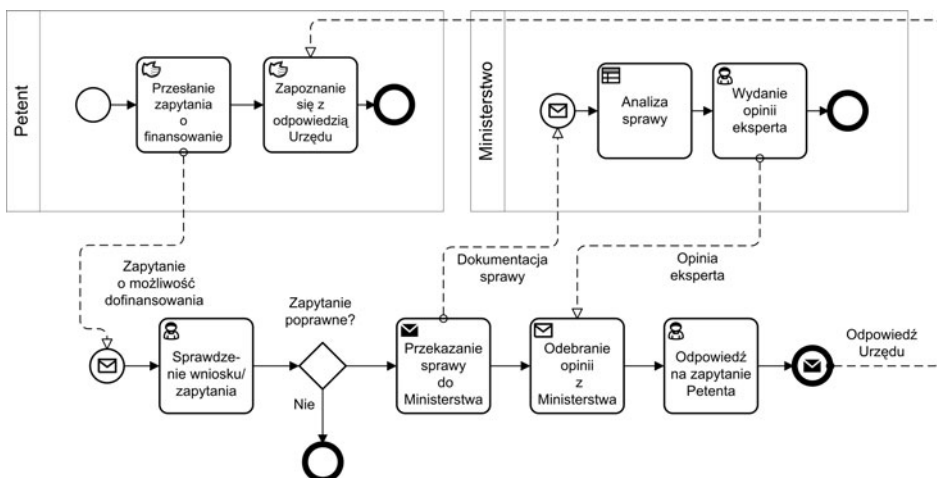
**Rysunek 7.4.** Kombinacja kolaboracji i diagramu procesu BPMN

Idąc dalej, możemy stworzyć kolaborację i pokazać wszystkie czynności. W ten sposób dojdziemy do pełnego obrazu analizowanego przez nas procesu i uzyskamy kolaborację z modelami wszystkich procesów składowych. Możemy przy okazji zrobić dwie rzeczy, żeby poprawić czytelność naszego modelu. Po pierwsze, możemy pominąć obramowanie puli *Urząd* (rysunek 7.5), ponieważ zgodnie z notacją BPMN 2.0 jedna i co najwyżej jedna pula w kolaboracji może być pominięta; przy czym pomijać możemy tylko obramowanie tej puli, która dotyczy uczestnika będącego w centrum naszego zainteresowania, a nie dowolnie wybranej. Skoro od początku koncentrujemy się na tym, co robi w procesie *Urząd*, to zgodnie z tą zasadą możemy pominąć obramowanie puli tego uczestnika.

Po drugie, możemy zmienić rozpiętość pozostałych pul (rysunek 7.6). Standard BPMN 2.0 zezwala na dowolne rozmieszczanie pul na diagramach, przy czym sugeruje, aby osoba tworząca model kierowała się użytecznością tego układu. Jeśli natomiast nie ma potrzeby swobodnego rozmieszczania pul, sugeruje, aby pule były równej rozpiętości w pionie lub w poziomie — zależnie od tego, jak zostały ułożone. To, która wersja jest bardziej czytelna, zależy od indywidualnych preferencji każdego. Rozciągnięte pule wprowadzają wizualny porządek na diagramie, jednak z drugiej strony, czasem lepiej oszczędzić trochę papieru i spróbować ścisnąć diagram poprzez sprytne rozmieszczenie pul poszczególnych uczestników procesu.



Rysunek 7.5. Obsługa zapytania przez Urząd

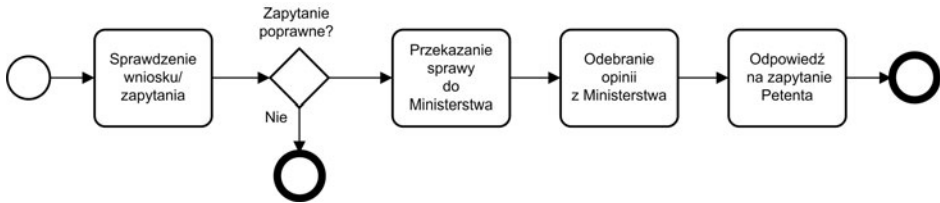


Rysunek 7.6. Obsługa zapytania przez Urząd (pule związane)

## 7.2. Procesy prywatne i publiczne

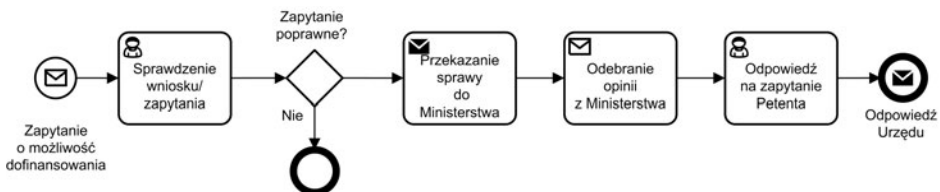
Przy okazji omawiania kolaboracji warto także wspomnieć o podziale procesów biznesowych na prywatne i publiczne. Prywatny proces biznesowy to taki, który jest realizowany przez daną organizację. Przy czym może, ale nie musi to być proces

wykonywalny. Jego model może więc być modelem implementacyjnym (szczegółowym) albo modelem opisowym (ogólnym), stworzonym jedynie na potrzeby dokumentacji. Rysunek 7.7 przedstawia prywatny proces *Urzędu*, czyli ogranicza się jedynie do czynności realizowanych przez *Urząd*. W tej postaci jest to prywatny proces niewykonywalny, czyli możemy go np. zaprezentować komuś z owego urzędu, dla kogo przeprowadzamy analizę, ponieważ jest to model uproszczony. Taki model pozwala na szybkie ustalenie tego, jakie przygotowania organizacyjne są konieczne do wdrożenia takiego procesu.



**Rysunek 7.7.** Proces prywatny (model poglądowy)

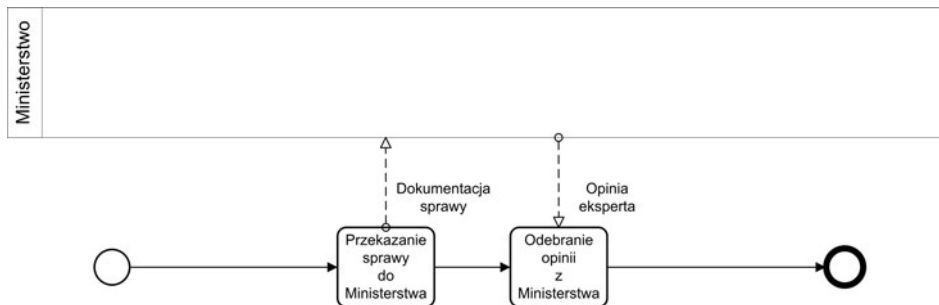
Jeśli jednak chcielibyśmy przedyskutować nasz proces z pracownikiem, który będzie odpowiedzialny np. za wdrożenie go na silniku procesów biznesowych w urzędzie, powinniśmy raczej przedstawić mu prywatny model procesu wykonywalnego (rysunek 7.8). Dla niego interesujące będzie bowiem nie tylko to, z jakich czynności składa się dany proces, ale także które czynności są systemowe, które wywołują jakieś usługi i jakie ewentualnie komunikaty występują w tym procesie. Te informacje dają bowiem jakiejkolwiek pojęcie o zakresie integracji technicznej i ilości pracy, którą trzeba będzie wykonać w celu technicznego wdrożenia tego procesu.



**Rysunek 7.8.** Proces prywatny (model wykonywalny)

Jeśli natomiast mielibyśmy przedyskutować proces z kimś, kto jest odpowiedzialny tylko i wyłącznie za integrację systemów informatycznych z systemami zewnętrznymi, to powinniśmy mu zaprezentować ten sam publiczny proces. Publiczny proces biznesowy to interakcja pomiędzy prywatnym procesem a innym procesem lub uczestnikiem. I, co najważniejsze, jest to uporządkowany zbiór **jedynie** tych czynności

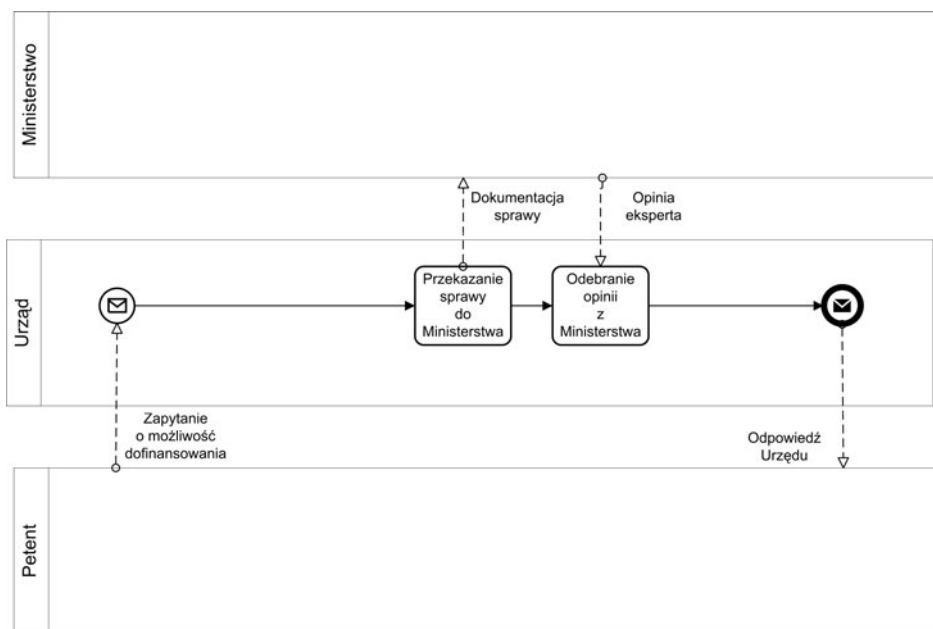
analizowanego procesu, które służą komunikacji z innymi uczestnikami. A więc proces publiczny to inna „wersja” procesu prywatnego. Rysunek 7.9 przedstawia ten sam proces realizowany przez *Urząd*, ale jako proces publiczny.



**Rysunek 7.9.** *Proces publiczny*

Warto zauważyć, że w specyfikacji OMG BPMN 2.0 proces publiczny jest definiowany jako zbiór tylko tych czynności, które służą komunikacji z innymi uczestnikami. Jak jednak widać na powyższym rysunku, praktyczna wartość takiego modelu jest dyskusyjna. Trudno powiedzieć, jaka dokładnie była intencja członków OMG odpowiedzialnych za specyfikację BPMN w wersji 2.0. Jeśli chodziło faktycznie o prezentację tylko czynności i to jedynie tych, które są związane z wymianą komunikatów z innymi uczestnikami, to powyższy model jest poprawny. Brakuje w nim jednak tych elementów, które nie są czynnościami, a mimo to ich istnienie jest uzasadnione jedynie wymianą komunikatów. Chodzi mianowicie o zdarzenia odebrania i wysłania komunikatów. W analizowanym procesie ma to o tyle istotne zdarzenie, że jeśli uwzględnimy nie tylko czynności, ale także zdarzenia, to na modelu pojawi się jeszcze jeden uczestnik — *Petent* (rysunek 7.10). Choć taki model nie jest całkowicie zgodny ze standardem BPMN 2.0, wydaje się bardziej sensowny i zgodny z intencją procesu publicznego.

Czytelnik może się zastanawiać, po co tworzyć kilka wersji tego samego procesu — w końcu jest to dodatkowa praca. Czy nie wystarczyłby jeden szczegółowy model? Nie. Po pierwsze, bardzo często nadmiar szczegółów technicznych utrudnia rozumienie modelu osobom „nietechnicznym”. Po drugie, narzędzia do modelowania w BPMN dostępne na rynku umożliwiają automatyczne ukrywanie fragmentów modelu oraz szybkie kopiowanie modelu pierwotnego z różnymi opcjami modyfikacji. W związku z tym w praktyce można stworzyć dokładny model „dla siebie” i kilka jego kopii dla różnych osób, którym ów model ma być prezentowany lub udostępniany.

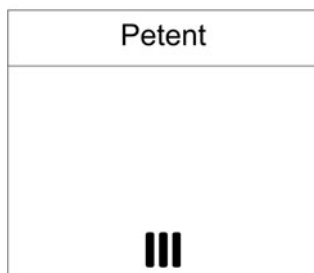


Rysunek 7.10. Proces publiczny ze zdarzeniami wysłania i odebrania

### 7.3. Wieloinstancyjni uczestnicy procesów biznesowych

Jeśli dokładniej przeanalizujemy model procesu obsługi sprawy w Krajowym Urzędzie Pracy, który omawialiśmy do tej pory, być może zauważymy jedną rzecz. Ów proces jest realizowany najpewniej przez jakiś jeden, określony urząd, który prawdopodobnie komunikuje się z jednym, określonym ministerstwem, a nie z różnymi ministerstwami. Z drugiej strony, możemy przypuszczać, że taki sam proces jest realizowany w przypadku każdego petenta, który zwraca się do urzędu z zapytaniem. Jeśli jednak chcielibyśmy wyraźnie zaznaczyć, że mamy do czynienia z procesem realizowanym przez jeden konkretny urząd i jedno ministerstwo, ale dla wszystkich petentów, to powinniśmy jawnie to wskazać poprzez użycie znacznika wieloinstancyjności. Znacznik ten jest reprezentowany przez trzy pionowe kreski (|||) i musi być umieszczony nad dolną krawędzią puli (rysunek 7.11).

Dodatkowo jeśli faktycznie jest to proces realizowany tylko przez jeden, określony urząd i żaden inny urząd tak go nie realizuje oraz ów urząd komunikuje się z jednym, określonym ministerstwem, to możemy zamiast ogólnikowych nazw typu „Urząd” i „Ministerstwo” zastosować nazwy konkretne (rysunek 7.12). Standard BPMN 2.0 pozostawia w tym względzie dowolność: możemy nazywać uczestników ich nazwami własnymi (konkretnymi) albo używać określeń ról, które przyjmują w danym procesie.

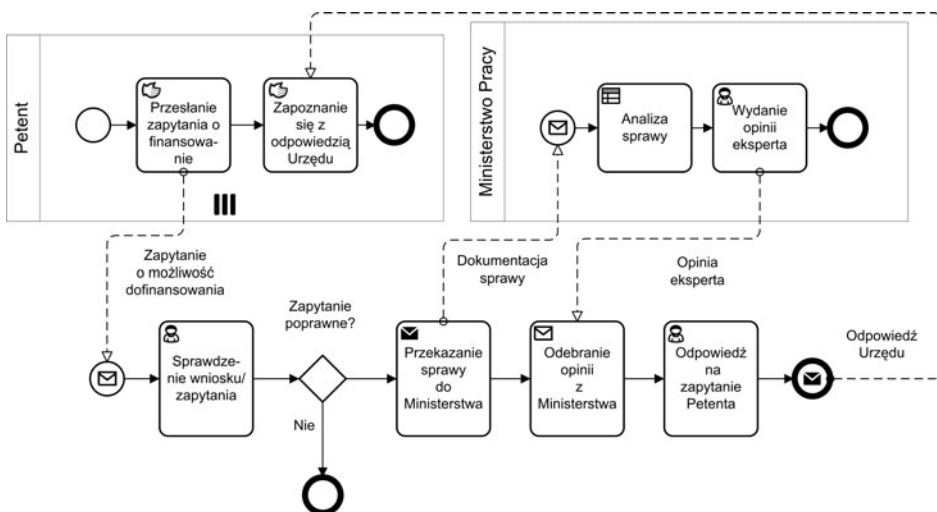


**Rysunek 7.11.** Wieloinstancyjny uczestnik procesu



**Rysunek 7.12.** Kolaboracja z wieloinstancyjnym uczestnikiem (tutaj: Petentem)

Znacznik wieloinstancyjności może być stosowany zawsze, gdy mamy do czynienia z wieloinstancyjnym uczestnikiem procesu biznesowego. Jego użycie nie jest ograniczone jedynie do „czystych kolaboracji”. W związku z tym na wszystkich diagramach w tym rozdziale (i w innych też), na których występuje nasz *Petent* i mowa jest o procesie obsługi sprawy w Krajowym Urzędzie Pracy, możemy umieścić znacznik wieloinstancyjności. Przykładowo model przedstawiony wcześniej na rysunku 7.6 powinien w wersji doprecyzowanej wyglądać tak, jak na rysunku 7.13.



**Rysunek 7.13.** Model z uczestnikiem wieloinstancyjnym



## 7.4. Konwersacje

Tworzenie poglądowych modeli, które umożliwiają obserwację danego biznesu „z lotu ptaka”, jest dobrą praktyką we wszelkich projektach. W ten sposób koncentrujemy się bowiem na szybkim uchwyceniu przestrzeni biznesowej, w której działamy, i uzyskujemy model, który pozwala nam na oszacowanie, jak wiele jednostek organizacyjnych i procesów będziemy musieli przeanalizować. Jest to szczególnie cenne, gdy rozpoczynamy projekt i np. mamy problem nie tyle z tym, jak coś opisywać i modelować (bo notacje znamy), ale z tym, co powinniśmy modelować. W takich sytuacjach ryzyko polega na tym, że możemy się zagłębić w szczególności tam, gdzie nie jest to potrzebne, i jednocześnie nie zajmujemy się wystarczająco dobrze tym, co jest kluczowe.

W literaturze poświęconej analizie biznesowej diagramy tego typu określa się mianem diagramów kontekstu biznesowego (*business context diagram*) i bardzo często na ich podstawie podejmowane są wstępne decyzje o tym, jak będą przebiegały prace analityczne w danym projekcie.

### 7.4.1. Diagramy konwersacji

W notacji BPMN 2.0 rolę diagramu kontekstu biznesowego mogą pełnić diagramy konwersacji. Rysunek 7.14 przedstawia diagram konwersacji uczestników, którzy występowali we wcześniej omawianych kolaboracjach.



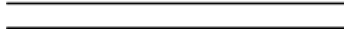
Rysunek 7.14. Diagram konwersacji

Intencją diagramów konwersacji jest ukrycie szczegółów realizacji procesów biznesowych (czyli czynności, bramek, zdarzeń itd.) i wyeksponowanie powiązań między uczestnikami tych procesów. Na diagramach konwersacji uczestnicy są prezentowani za pośrednictwem pul, tak samo jak w przypadku kolaboracji. Natomiast zamiast komunikatów stosuje się konwersacje. Konwersacje to węzły prezentowane w notacji BPMN 2.0 za pośrednictwem symbolu sześciokąta (rysunek 7.15).



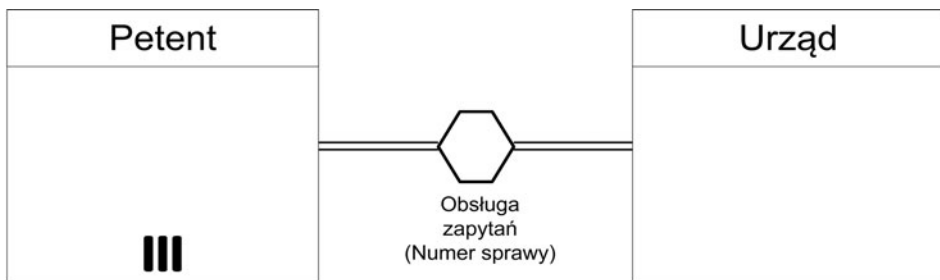
Rysunek 7.15. Węzeł konwersacji

Węzły konwersacji (*conversation node*) są łączone z pulami za pośrednictwem łączy konwersacji (*conversation link*), prezentowanych w notacji BPMN 2.0 za pośrednictwem podwójnej linii (rysunek 7.16). Konwersacja może łączyć się z więcej niż dwoma uczestnikami.



**Rysunek 7.16.** Łączy konwersacji

Węzeł konwersacji agreguje komunikaty wymieniane między dwoma uczestnikami. Na ogół komunikacja ta dotyczy określonych obiektów biznesowych (np. zamówień, faktur, operacji na rachunku itp.). W naszym przykładzie konwersacja między petentem i Krajowym Urzędem Pracy dotyczy „zapytań”. Stąd nazwa tej konwersacji: *Obsługa zapytań*. Można jednak przypuszczać, że każdemu zapytaniu nadawany jest jakiś identyfikator, np. „Numer sprawy”. Jeśli poza nazwą konwersacji chcemy wskazać także ów identyfikator (tzw. klucz korelacji), możemy to zrobić, umieszczając go w nawiasie w nazwie konwersacji (rysunek 7.17). Poza tym, że jest to dodatkowa informacja o samej konwersacji, jest to także mechanizm kierowania wysyłanych i odbieranych komunikatów do odpowiednich instancji procesu. Np. jeśli jednocześnie uruchomione są dwa procesy obsługi zapytania o dofinansowanie, to klucz korelacji *Numer sprawy* zapewnia pośrednio, że odpowiedź Krajowego Urzędu Pracy trafi do właściwego petenta.

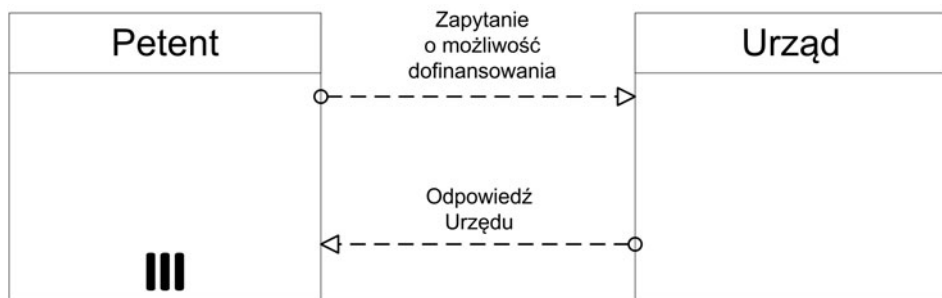


**Rysunek 7.17.** Konwersacja z nazwą i kluczem korelacji

Oczywiście możemy rozłożyć konwersację na konkretne komunikaty wymieniane pomiędzy uczestnikami. W ten sposób otrzymamy zwykły model kolaboracji (rysunek 7.18).

#### 7.4.2. Konwersacje złożone

Bardzo często wymiana komunikatów między uczestnikami jest bardziej skomplikowana. Konwersacja może składać się z wielu konwersacji podrzędnych. W takim przypadku możemy zastosować konwersację złożoną (*sub-conversation*), która



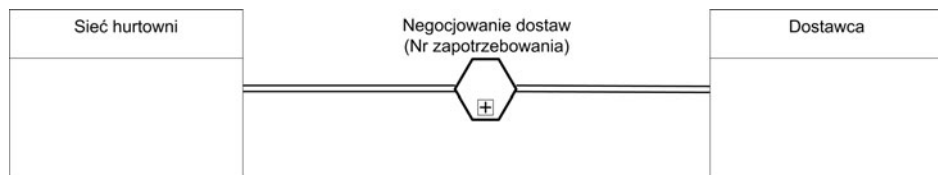
**Rysunek 7.18.** Konwersacja rozłożona na przepływy komunikatów

w notacji BPMN 2.0 od zwykłej konwersacji różni się tym, że ma znacznik plusa (+) nad dolną krawędzią (rysunek 7.19).



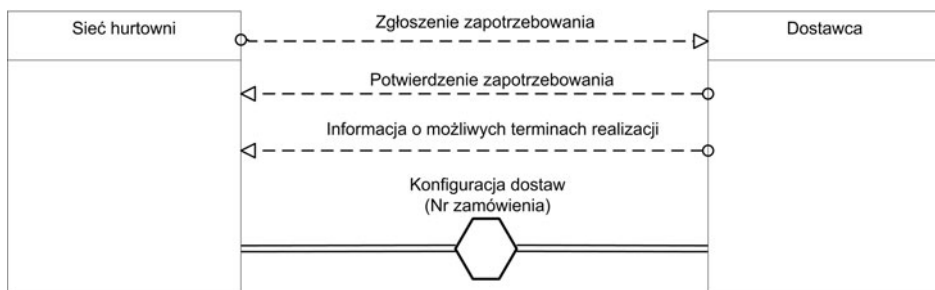
**Rysunek 7.19.** Węzeł konwersacji złożonej

Do zilustrowania konwersacji wykorzystamy proces zaopatrywania sieci kilkudziesięciu hurtowni samochodowych w części różnych producentów. Ten sam proces jest także wykorzystany w rozdziale 9., „Czynności i podprocesy” przy okazji wyjaśniania pętli, czynności wieloinstancyjnych i kompensacji. Założmy, że sieć hurtowni negocjuje z producentami dostawy towaru (rysunek 7.20).



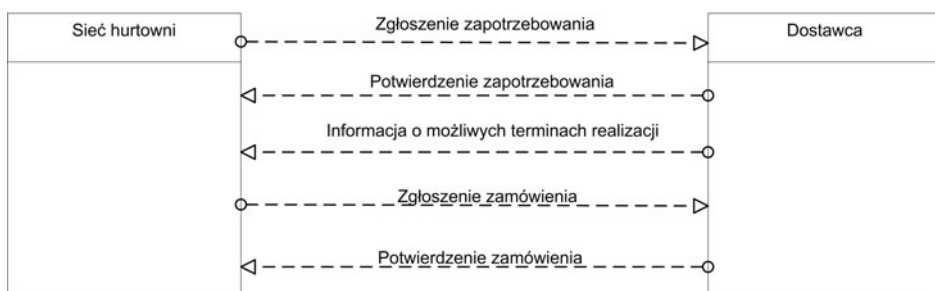
**Rysunek 7.20.** Konwersacja złożona z kluczem Nr zapotrzebowania

W ramach tych negocjacji sieć hurtowni ustala z każdym dostawcą części możliwość zrealizowania zamówienia (rysunek 7.21). Do tego celu wykorzystywany jest obiekt biznesowy *Zapotrzebowanie*, który zawiera takie informacje, jak nazwa części, liczba sztuk i data dostępności towaru. Dostawca potwierdza odebrane zapotrzebowanie i przekazuje informację o możliwych terminach realizacji. Może się bowiem okazać, że nie wszystkie części z zapotrzebowania będą dostępne w terminach oczekiwanych przez sieć hurtowni. Następnie realizowana jest konwersacja *Konfiguracja dostaw* z kluczem *Nr zamówienia*.



**Rysunek 7.21.** Konwersacja złożona rozbita na konwersację i przepływ komunikatów

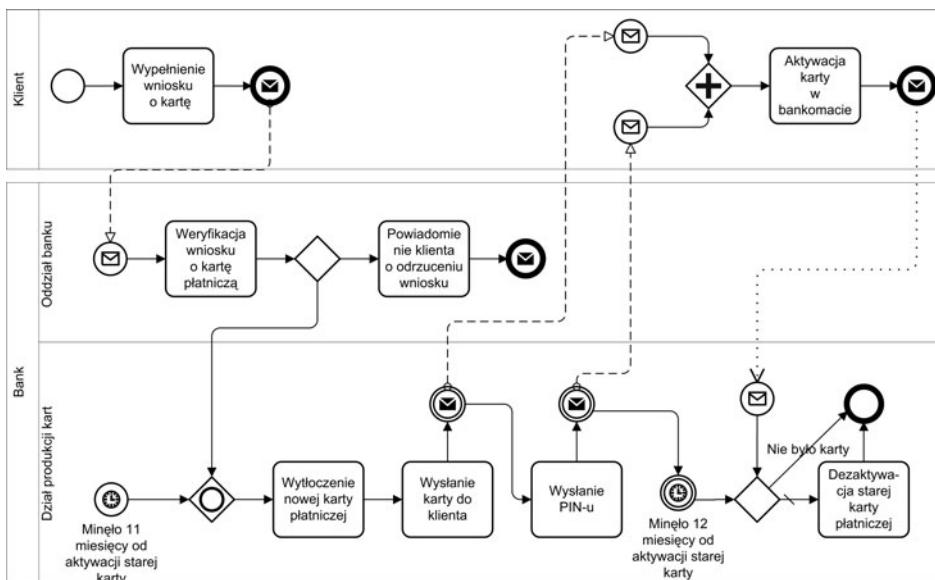
Konwersacja *Konfiguracja dostaw* jest zbiorem komunikatów wymienianych między dostawcą i siecią hurtowni w celu przyporządkowania poszczególnych elementów z zapotrzebowania do zamówienia (rysunek 7.22). Jest to realizowane zgodnie z terminami dostępności towarów podanymi przez dostawcę. *Zapotrzebowanie* służyło do ustalenia tego, co jest potrzebne i na kiedy, a *Zamówienie* jest umową określającą, co będzie dostępne i w jakim terminie. Potwierdzenie zamówienia przez dostawcę oznacza, że zobowiązuje się on do zrealizowania tego zamówienia.



**Rysunek 7.22.** Konwersacja złożona rozbita na komunikaty składowe

Zdarzenie w procesie biznesowym to wystąpienie jakiejś sytuacji, która jest na tyle istotna w przebiegu tego procesu, że warto ją nazwać i zdefiniować. Zdarzeniem może być odebranie lub wysłanie jakiejś wiadomości, nastąpienie jakiegoś momentu, rozpoczęcie lub zakończenie jakiejś czynności itd. Początkowo modelowanie procesów biznesowych z uwzględnieniem zdarzeń może wydawać się trudne, ponieważ w większości przypadków sam proces postrzegamy raczej jako ciąg od razu następujących po sobie czynności. Co więcej, domyślnie — co potwierdzają wieloletnie doświadczenia w projektach analitycznych — większość ludzi zakłada, że proces przebiega bez przerwy i kończy się w kilka sekund, minut, godzin, ewentualnie dni. Jednak procesy biznesowe mają tę specyficzną cechę, że bardzo często ich przebieg zależy od wielu czynników zewnętrznych, tzn. takich, które są poza kontrolą samego procesu. Stąd też przebieg procesu biznesowego nie jest zazwyczaj prostą sekwencją działań, ale mieszaną działań i zdarzeń, i bardzo często proces „zatrzymuje się” (np. na kilkanaście dni) w oczekiwaniu na nastąpienie jakiegoś zdarzenia. Zazwyczaj także, jeśli proces powoduje wytworzenie czegoś, to „coś” nie jest potrzebne temu procesowi, który to wytworzył, ale innym procesom. Dla nich bowiem stanowi „powód” do działania. Zdarzenia można więc też traktować jak spoiwo, które łączy różne sekwencje czynności w złożone i rozbudowane struktury procesów i podprocesów biznesowych. Rysunek 8.1 przedstawia proces aktywacji karty płatniczej, w którym zidentyfikowano więcej zdarzeń niż czynności.

Zdarzenia w BPMN 2.0 są prawdopodobnie najważniejszym i najtrudniejszym elementem tego standardu. Z jednej strony, opanowanie zdarzeń w procesach biznesowych nie jest łatwe, bo jest ich w sumie około kilkudziesięciu (!), a ich zrozumienie wymaga znajomości różnych mechanizmów stosowanych w systemach oprogramowania, jak np. transakcje, obsługa błędów czy kompensacje. Z drugiej strony, żeby sprawnie posługiwać się standardem BPMN 2.0 w „mowie i piśmie”, tzn. żeby umieć tworzyć i czytać modele BPMN, konieczne jest zrozumienie i zapamiętanie wszystkich reguł dotyczących zdarzeń.



Rysunek 8.1. Proces aktywacji karty płatniczej

## 8.1. Typy zdarzeń




W notacji BPMN 2.0 rozróżnia się trzy główne typy zdarzeń (rysunek 8.2): początkowe, pośrednie i końcowe. **Zdarzenia początkowe**, oznaczające początek procesu, na modelach prezentowane są za pośrednictwem okręgu z pojedynczą, niepogrubioną krawędzią. **Zdarzenia pośrednie**, oznaczające sytuacje występujące w trakcie przebiegu procesu, na modelach prezentowane są za pośrednictwem okręgu z podwójną, niepogrubioną krawędzią. **Zdarzenia końcowe**, oznaczające koniec modelowanego procesu biznesowego, na modelach prezentowane są za pośrednictwem okręgu z pojedynczą, pogrubioną krawędzią.



Rysunek 8.2. Główne typy zdarzeń BPMN 2.0

Wszystkie zdarzenia w standardzie BPMN są związane z tzw. wyzwalaczami, przy czym część zdarzeń początkowych i pośrednich jest związana z *przechwytywaniem* wyzwalaczy, a część zdarzeń pośrednich i końcowych jest związana z *aktywowaniem* wyzwalaczy (tabela 8.1).

**Tabela 8.1.** Rodzaje zdarzeń i ich możliwe znaczenie

	 Zdarzenie początkowe	 Zdarzenie pośrednie	 Zdarzenie końcowe
Przechwytywanie wyzwalaczy	Tak	Tak	Nie
Aktywowanie wyzwalaczy	Nie	Tak	Tak

Zdarzenia związane z przechwyceniem wyzwalaczy są istotne, ponieważ na tej podstawie podejmowane są jakieś działania w procesie. Ich wystąpienie powoduje wyzwolenie szeregu działań będących konsekwencją tych zdarzeń — stąd określenie „wyzwalacz”. Natomiast zdarzenia związane z aktywowaniem wyzwalaczy są istotne, ponieważ umożliwiają nam podkreślenie faktu, że jakiś ciąg działań prowadzi do wytworzenia pewnego *rezultatu* (produktu procesu). Jeśli jednak dokładniej zastanowimy się, czym jest wytworzenie *rezultatu*, zauważymy, że równie dobrze możemy je utożsamiać z generowaniem lub, inaczej, *aktywowaniem wyzwalacza*. Na ogół bowiem pojawienie się *rezultatu* jednego procesu oznacza *aktywowanie* kolejnego.

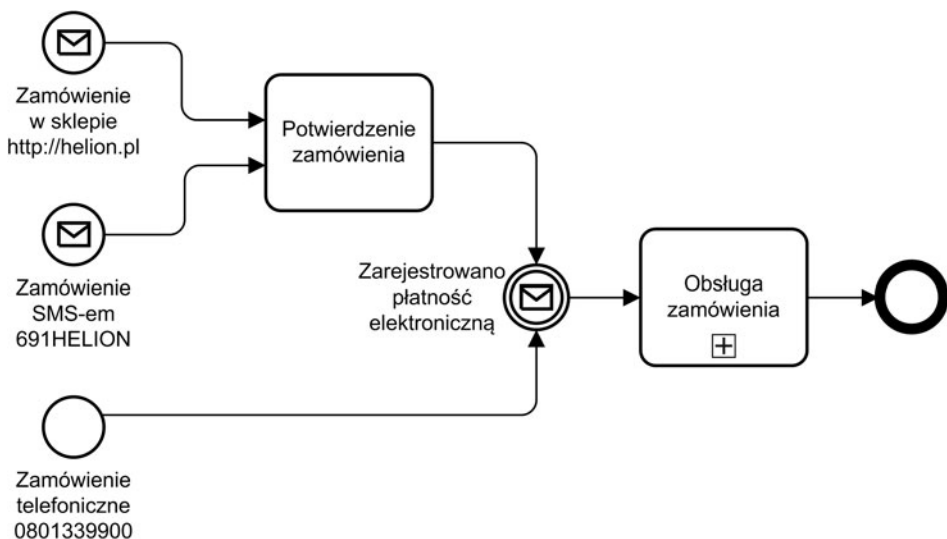
Jeśli Czytelnik nadal ma problem ze zrozumieniem, co znaczy przechwytywanie lub aktywowanie wyzwalacza, może posłużyć się przykładem z żarówką i odkuraczem. Załóżmy, że zapalenie żarówki oznacza, iż możemy rozpocząć odkurzanie, a wyłączenie odkurzacza oznacza, że ktoś inny może przystąpić do mycia i pastowania podłogi. Jeśli teraz z zapalą, zmotywowani jasnymi regułami działania, przystąpimy do realizacji nieskomplikowanego procesu sprzątania, będziemy oczekiwali na zapalenie żarówki i w chwili, gdy się ona zaświeci, rozpoczniemy odkurzanie. W ten sposób przechwyciliśmy wyzwalacz i rozpoczęliśmy realizację czynności odkurzania. Po zakończeniu pracy wyłączamy odkurzacza, tym samym dając komuś innemu znać, że może zaczynać mycie i pastowanie podłogi. W ten sposób aktywowaliśmy inny wyzwalacz.

### 8.1.1. Zdarzenia początkowe

Zdarzenia początkowe, reprezentowane na modelach przez okrąg z pojedynczą, niepogrubioną krawędzią, jak już wcześniej wspomniano, oznaczają początek procesu biznesowego, a więc możemy z nich wyprowadzić przepływy do bram lub aktywności, natomiast nie mogą one mieć żadnych wejściowych *przepływów sekwencji*. Warto jednak pamiętać, że w standardzie BPMN 2.0 zdarzenie początkowe jest opcjonalne, tzn. nie zawsze musimy umieszczać je w modelu. W dalszej części rozdziału przedstawiono reguły, jakie obowiązują podczas modelowania zdarzeń początkowych w notacji BPMN 2.0.

Jeśli proces jest skomplikowany lub warunki jego uruchomienia nie są oczywiste, zalecane jest użycie *zdarzenia początkowego*. Niestety, specyfikacja OMG BPMN 2.0 nie wyjaśnia pojęć „skomplikowany proces” oraz „nieoczywiste warunki uruchomienia procesu”, a więc równie dobrze możemy przyjąć, że jeśli prezentowane zdarzenie początkowe miałoby być tylko symbolem umieszczonym w modelu „na siłę”, to nie musimy go używać. Jeśli natomiast bez tego symbolu zrozumienie modelu byłoby utrudnione, lepiej go nie pomijać. Obowiązkowo natomiast musimy używać zdarzeń początkowych w modelach, w których istnieją zdarzenia końcowe.

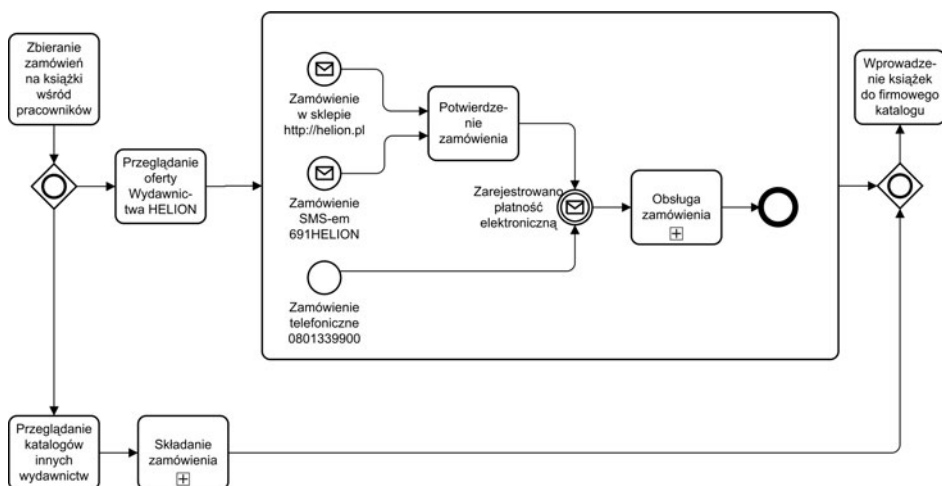
Proces może rozpoczynać się na wiele sposobów, a więc model może zawierać kilka różnych zdarzeń początkowych. Rysunek 8.3 przedstawia przykładowy proces, który może zacząć się albo od złożenia zamówienia drogą elektroniczną, albo od złożenia zamówienia drogą SMS-ową, albo od złożenia zamówienia przez telefon.



**Rysunek 8.3.** *Proces zamawiania książki wydawnictwa Helion*

Proces może być definiowany na kilku poziomach, np. może zawierać podprocesy (patrz podrozdział 9.2) lub czynności wywołania (patrz podrozdział 9.4), które uruchamiają inne procesy. W związku z tym reguły dotyczące zdarzeń należy uwzględniać na każdym poziomie niezależnie. Innymi słowy, jeśli na poziomie podprocesu zastosowaliśmy zdarzenie końcowe, to na tym samym poziomie musimy umieścić zdarzenie początkowe, ale nie jesteśmy do tego zobligowani poziom wyżej (rysunek 8.4).





**Rysunek 8.4.** Proces aktualizacji biblioteki firmowej

Zdarzenia początkowe mogą dotyczyć procesów na następujących poziomach:








- procesów głównego poziomu,
- zagnieżdżonych podprocesów,
- zagnieżdżonych podprocesów zdarzeniowych.

#### 8.1.1.1. Zdarzenia początkowe procesów głównego poziomu

Zdarzenia początkowe procesów głównego poziomu umożliwiają jednoznaczne określenie sposobu, w jaki te procesy są uruchamiane (jak się zaczynają). W notacji BPMN 2.0 jest 7 możliwych zdarzeń początkowych dla procesów głównego poziomu (tabela 8.2): *Nieokreślony*, *Komunikat*, *Czasowe*, *Warunkowe*, *Sygnal*, *Wielozdarzenie*, *Wielozdarzenie równoległe*.

Rysunek 8.5 przedstawia proces otwierania rachunku, który ma cztery zdarzenia początkowe: jedno zdarzenie komunikatu, jedno zdarzenie oparte na sygnale i dwa zdarzenia bez wyzwalacza. Oznacza to, że proces ten może wystartować, jeśli określony uczestnik wyśle odpowiedni komunikat lub wygenerowany zostanie konkretny sygnał albo jeśli klient odwiedzi oddział lub placówkę banku. Na tym przykładzie widać, że pojęcie początku w przypadku procesów biznesowych jest nietrywialne, w szczególności jeśli patrzy się na proces jako współdziałanie wielu uczestników. W procesie przedstawionym na rysunku pozornie wszystko może się rozpocząć tylko od wizyty klienta w oddziale lub w placówce, ponieważ zdarzenia początkowe komunikatu i sygnału są kontynuacją, odpowiednio, otwierania ROR w placówce i otwierania ROR w oddziale przez sekcję obsługi bezpośredniej. Jednak jeśli bank posiada usługę „Szybka pożyczka”, oferującą krótkoterminowy

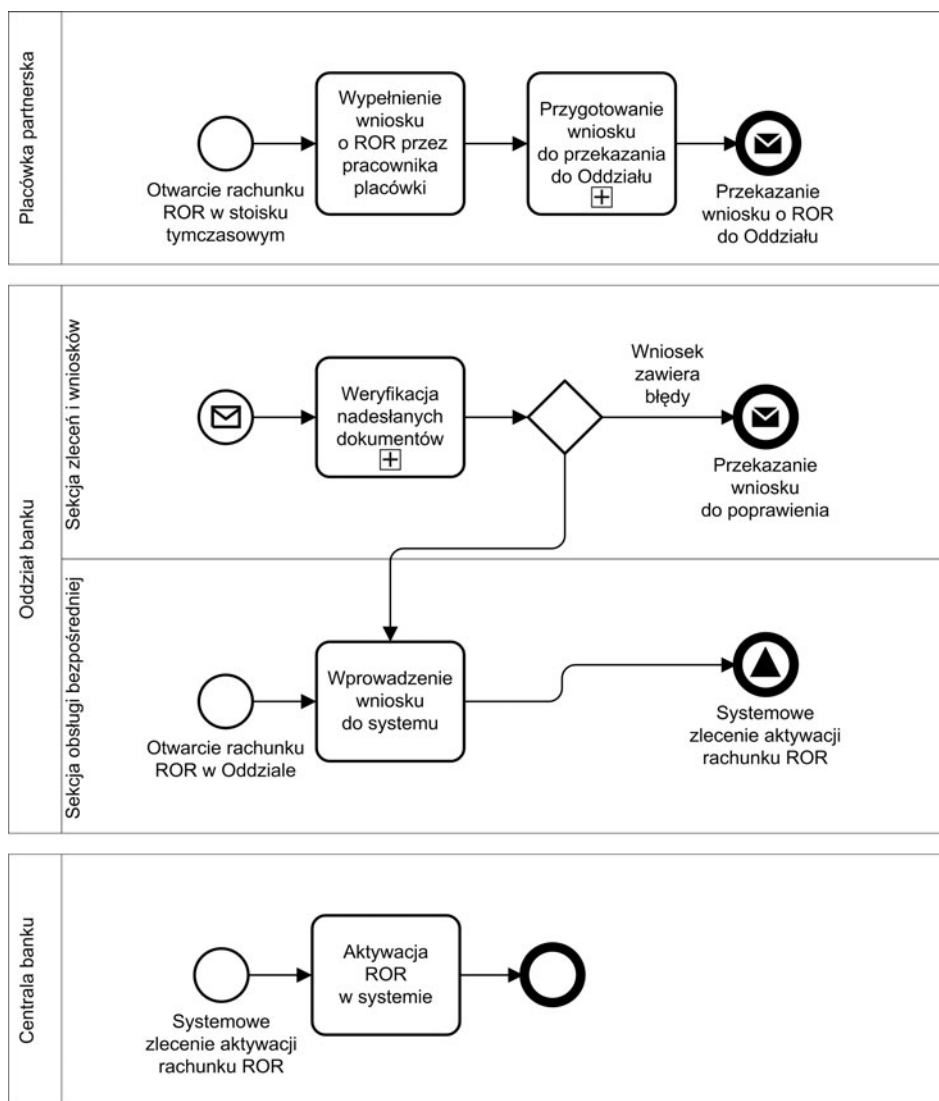
**Tabela 8.2.** Zdarzenia początkowe

Typ	Opis	Znacznik
<i>Nieokreślony</i>	Jeśli nie mamy możliwości wskazania wyzwalacza i jednocześnie musimy lub chcemy zawrzeć w modelu zdarzenie początkowe, wtedy możemy zastosować zdarzenie początkowe bez wyzwalacza.  Jeśli proces głównego poziomu rozpoczyna się od zdarzenia bez wyzwalacza, oznacza to, że może on być uruchamiany przez <i>czynność wywołania</i> (patrz rozdział 9., „Czynności i podprocesy”), czyli może być wywoływany przez inny proces.	
<i>Komunikat</i>	Proces jest uruchamiany na podstawie komunikatu pochodzącego od <i>uczestnika</i> (patrz podrozdział 2.3).	
<i>Czasowe</i>	Proces jest uruchamiany w chwili, gdy następuje określony moment w czasie. Może to być konkretna data i godzina, np. 26 kwietnia 2027 22:40, albo cykl, np. każdy wtorek 17:40.	
<i>Warunkowe</i>	Proces jest uruchamiany w chwili, gdy spełniony jest specyficzny warunek, np. określony wyrażeniem: „Saldo na rachunku poniżej 1000 €” lub „Pacjent jest płci męskiej i ma ponad 50 lat”.	
<i>Sygnal</i>	Proces jest uruchamiany w chwili, gdy nadejdzie sygnał od innego procesu. Należy pamiętać, że sygnał nie jest tym samym co komunikat. Komunikat jest bowiem kierowany do konkretnego odbiory, sygnał natomiast może być wysłany „w eter” i może np. spowodować uruchomienie wielu różnych procesów niezależnie.	
<i>Wielozdarzenie</i>	Proces jest uruchamiany na podstawie jednego z wielu różnych, możliwych wyzwalaczy.	
<i>Wielozdarzenie równoległe</i>	Uruchomienie procesu zależy od wszystkich możliwych wyzwalaczy zdefiniowanych w ramach wielozdarzenia równoległego.	

kredyt do kwoty 50 000 PLN wraz z otwarciem rachunku ROR, aktywacja systemowa tego rachunku ROR nie musi być związana ze zwykłym wnioskiem o otwarcie rachunku ROR, klient bowiem podpisuje umowę związaną: na pożyczkę i ROR.

#### 8.1.1.2. Zdarzenia początkowe podprocesów

Wszystkie podprocesy muszą rozpoczynać się zdarzeniem bez wyzwalacza, ich przebieg zależy bowiem od tego, czy zostaną wywołane przez proces nadrzędny, i w tym kontekście nie ma znaczenia, co jest wyzwalaczem. To proces nadrzędny zawsze kontroluje wystąpienie zdarzenia początkowego podprocesu (tabela 8.3).



Rysunek 8.5. Proces otwierania rachunku ROR w banku

Tabela 8.3. Zdarzenie początkowe podprocesu





Typ	Opis	Znacznik
<i>Nieokreślony</i>	Jeśli chcemy wskazać na modelu, że dane zdarzenie jest początkiem podprocesu, musimy zawsze zastosować zdarzenie bez wyzwalacza. Nie ma znaczenia, czy jest to podproces osadzony (model w modelu), czy wywoływany (ponownie wykorzystywany).	

Należy jednak pamiętać, że ten sam model może przedstawiać proces biznesowy, który może być jednocześnie modelem podprocesu i procesu głównego poziomu. Możemy więc mieć do czynienia z podprocesem, który ma kilka zdarzeń początkowych i część z nich jest związana z konkretnymi *wyzwalaczami*. Te zdarzenia jednak nie mogą być używane w kontekście podprocesu. Innymi słowy, możemy je w ogóle pomijać. Można też zapamiętać taką regułę: jeśli jakiś proces ma kilka zdarzeń początkowych i co najmniej jedno z nich jest zdarzeniem bez wyzwalacza, to może to być proces głównego poziomu i ewentualnie podproces dla innych procesów; jeśli natomiast nie ma zdarzenia początkowego bez wyzwalacza, to na pewno nie mamy do czynienia z procesem, który czasem jest podprocesem.


### 8.1.1.3. Zdarzenia początkowe podprocesów zdarzeniowych

Modelując podprocesy zdarzeniowe, czyli te, których działanie rozpoczyna się na podstawie konkretnego wyzwalacza, musimy użyć jednego z możliwych zdarzeń początkowych (tabela 8.4), takiego jak: *Komunikat*, *Timer*, *Eskalacja*, *Błąd*, *Kompensacja*, *Warunkowe*, *Sygnal*, *Wielozdarzenie* i *Wielozdarzenie równoległe*. W niektórych przypadkach podprocesy zdarzeniowe mogą przerywać działanie procesu nadrzędnego, a o tym, czy ma nastąpić przerwanie, czy nie, decydujemy poprzez dobór właściwego zdarzenia początkowego. Więcej o podprocesach zdarzeniowych można przeczytać w podrozdziale 9.2.2.







**Tabela 8.4.** Zdarzenia początkowe podprocesów zdarzeniowych

Typ	Opis	Znacznik
<b>Komunikat</b>	Podproces jest uruchamiany na podstawie komunikatu pochodzącego i przerywa działanie procesu nadrzędnego — wówczas kontur zdarzenia początkowego jest ciągły. Jeśli podproces nie przerywa procesu nadrzędnego, kontur zdarzenia jest przerywany.	Przerywający 
	Nadawca komunikatu może być powiązany z początkowym zdarzeniem <i>Komunikat</i> za pośrednictwem <i>przepływu komunikatu</i> (patrz rozdział 5., „Przepływy”).	Nieprzerywający 
<b>Timer</b>	Gdy następuje określony moment w czasie, podproces jest uruchamiany i przerywany (kontur ciągły) lub uruchamiany i nieprzerywany (kontur przerywany). Owym „momentem w czasie” może być konkretna data i godzina, np. 26 kwietnia 2027 22:40, albo cykl, np. każdy wtorek 17:40.	Przerywający 
		Nieprzerywający 

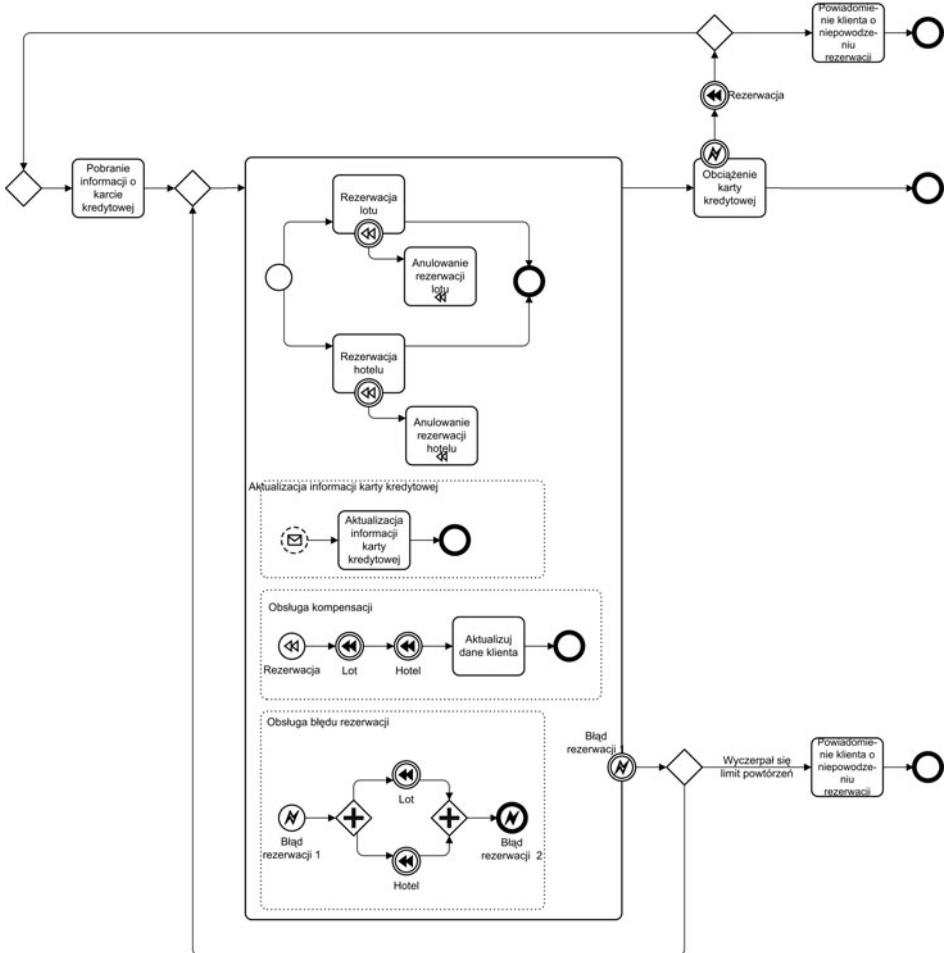
**Tabela 8.4.** Zdarzenia początkowe podprocesów zdarzeniowych — ciąg dalszy

Typ	Opis	Znacznik
<i>Eskalacja</i>	Podproces jest uruchamiany ze względu na konieczność eskalowania jakiejś sytuacji w procesie nadrzędnym (np. realizacja działania przedłuża się i konieczne jest raportowanie problemów do kogoś, kto nadzoruje proces). Jeśli obsługa tej eskalacji przerywa proces nadrzędny, kontur zdarzenia jest ciągły. Jeśli obsługa tej eskalacji nie przerywa procesu nadrzędnego, kontur zdarzenia jest przerywany. Początkowe zdarzenie eskalacji może być wykorzystywane tylko do wywoływania podprocesu osadzonego. Każde zdarzenie eskalacji może mieć określony parametr <code>escalationCode</code> , dzięki czemu możliwe jest eskalowanie w jednym podprocesie wielu różnych sytuacji.	Przerywający 
		Nieprzerywający 
<i>Błąd</i>	Podproces jest uruchamiany, ponieważ wystąpił błąd w procesie nadrzędnym. Wystąpienie błędu zawsze przerywa proces nadrzędny.  Początkowe zdarzenie błędu może być wykorzystywane tylko do wywoływania podprocesu osadzonego.	Przerywający 
<i>Kompensacja</i>	Podproces jest uruchamiany w celu przeprowadzenia kompensacji (wycofania) wcześniejszych czynności. Kompensacja może być zrealizowana dopiero po zakończeniu czynności, które mają być wycofane, więc nigdy nie przerywa procesu nadrzędnego.  Początkowe zdarzenie kompensacji może być wykorzystywane tylko do wywoływania podprocesu osadzonego.	Nieprzerywający 
<i>Warunkowe</i>	Proces jest uruchamiany w chwili, gdy spełniony jest specyficzny warunek, określony np. wyrażeniem: „Saldo na rachunku poniżej 1000 €” lub „Pacjent jest płci męskiej i ma ponad 50 lat”. Jeśli proces nadrzędny jest przerywany, gdy spełnione są warunki, mamy do czynienia ze zdarzeniem początkowym przerywającym — kontur ciągły. W przeciwnym razie kontur zdarzenia jest przerywany.	Przerywający 
		Nieprzerywający 

**Tabela 8.4.** Zdarzenia początkowe podprocesów zdarzeniowych — ciąg dalszy

Typ	Opis	Znacznik
<b>Sygnal</b>	Podproces jest uruchamiany w chwili, gdy nadejdzie sygnał od innego procesu. Należy pamiętać, że sygnał nie jest tym samym co komunikat. Komunikat jest bowiem kierowany do konkretnego odbiory, sygnał natomiast może być wysłany „w eter” i może np. spowodować uruchomienie wielu różnych podprocesów niezależnie. Jeśli proces nadrzędny jest przerywany w celu obsługi sygnału, mamy do czynienia ze zdarzeniem początkowym przerywającym — kontur ciągły. W przeciwnym razie kontur zdarzenia jest przerywany.	Przerywający 
		Nieprzerywający 
<b>Wielozdarzenie</b>	Podproces jest uruchamiany na podstawie jednego z wielu różnych, możliwych wyzwalaczy. Niezależnie od tego, który wyzwalacz zostanie aktywowany, proces nadrzędny może być przerywany; mamy wtedy do czynienia ze zdarzeniem początkowym przerywającym — kontur ciągły. W przeciwnym razie (gdy proces nadrzędny nie jest przerywany) kontur jest przerywany.	Przerywający 
		Nieprzerywający 
<b>Wielozdarzenie równoległe</b>	Uruchomienie podprocesu zależy od wszystkich możliwych wyzwalaczy zdefiniowanych w ramach wielozdarzenia równoległego. W chwili uruchomienia podprocesu proces nadrzędny może być przerywany (kontur ciągły). Jeśli proces nadrzędny nie jest przerywany, zdarzenie początkowe ma kontur przerywany.	Przerywający 
		Nieprzerywający 

Rysunek 8.6 prezentuje fragment modelu procesu zawierającego podprocesy zdarzeniowe. Jest to proces obsługi delegacji, w którym następuje rezerwacja hotelu i przelotu. Dodatkowo w ramach podprocesu rezerwacji mogą zostać uruchomione trzy podprocesy zdarzeniowe: nieprzerywający podproces aktualizacji informacji karty kredytowej, nieprzerywający podproces wycofywania (czyli kompensacji) rezerwacji i przerywający podproces obsługi błędów. Nieprzerywający podproces zdarzeniowy aktualizacji informacji karty kredytowej może być wywołany więcej niż raz w trakcie podprocesu rezerwacji na podstawie komunikatu (gdy np. na daną kartę kredytową w tym samym czasie dokonano innych płatności lub gdy ją zastrzeżono). Podproces zdarzeniowy wycofywania rezerwacji może być wywołany tylko po zakończeniu rezerwacji, ale w chwili realizacji czynności obciążania karty kredytowej. Wynika to po pierwsze z tego, że zdarzenie generujące wyzwalacz kom-



**Rysunek 8.6.** Przykładowy proces z podprocesami zdarzeniowymi

pensacji, nazwane *Rezerwacja*, następuje tylko wtedy, gdy wystąpi błąd związany z obciążaniem karty kredytowej; po drugie, wszystkie kompensacje z definicji wymagają uprzedniego zakończenia procesu nadrzędnego, którego dotyczą. Przerwyjący podproces obsługi błędów może nastąpić w dowolnym momencie, gdy podproces rezerwacji jest aktywny i zidentyfikowany zostanie błąd o identyfikatorze *Błąd rezerwacji 1*.

### 8.1.2. Zdarzenia pośrednie

Zdarzenia pośrednie, reprezentowane na modelach przez okręgi z podwójną, niepogrubioną krawędzią, występują w trakcie procesu i można je wykorzystywać do prezentowania:

- komunikatów, które muszą być odebrane lub wysłane w trakcie procesu,
- opóźnień, które występują w procesie,
- obsługi sytuacji wyjątkowych,
- obsługi kompensacji (wycofania) wcześniejszych czynności.

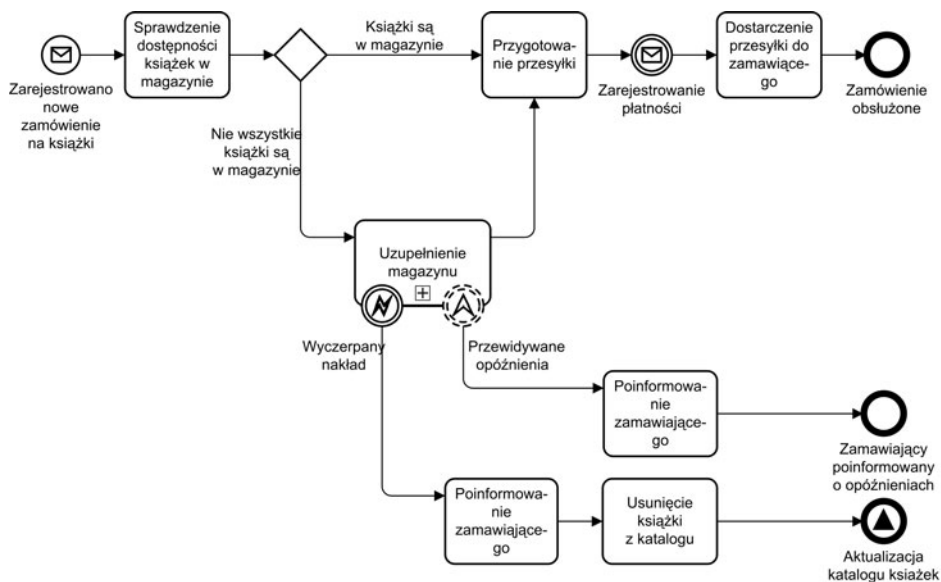
W standardzie BPMN 2.0 wyróżnia się 12 rodzajów zdarzeń pośrednich: *Nieokreślony*, *Komunikat*, *Timer*, *Eskalacja*, *Błąd*, *Anulowanie*, *Kompensacja*, *Warunkowe*, *Łączy*, *Sygnal*, *Wielozdarzenie* oraz *Wielozdarzenie równoległe*. Dodatkowo zdarzenia pośrednie występują w dwóch wariantach:

1. W zwykłym przepływie — jako elementy modelu połączone z innymi, np. czynnościami lub bramkami, za pośrednictwem przepływów sekwencji (patrz rozdział 5.). W tym wariancie mogą służyć do reprezentowania zdarzeń związanych z przechwyceniem wyzwalaczy lub tworzeniem rezultatów (patrz początek rozdziału). Szczegółowe omówienie wszystkich zdarzeń pośrednich w tym wariancie zawarto w podrozdziale 9.1.2.1.
2. Na krawędzi czynności, których dotyczą. W tym wariancie mogą służyć tylko do reprezentowania zdarzeń związanych z przechwyceniem wyzwalacza, ale z rozróżnieniem, czy jest to wychwycenie przerywające (linia konturu ciągła), czy nieprzerywające (linia konturu przerywana). Szczegółowe omówienie wszystkich zdarzeń pośrednich w tym wariancie zawarto w podrozdziale 9.1.2.2.

Rysunek 8.7 przedstawia przykładowy model procesu biznesowego ze zdarzeniami „zwykłymi” i „krawędziowymi”. Zdarzenia pośrednie przypięte do krawędzi czynności *Uzupełnienie magazynu* mają nazwy: *Wyczerpany nakład* i *Przewidywane opóźnienie*.

Kiedy stosować zwykłe zdarzenia pośrednie, a kiedy umieszczać je na krawędziach czynności? Choć specyfikacja BPMN 2.0 nie podaje takiej reguły wprost, analizując definicje poszczególnych zdarzeń pośrednich „zwykłych” i „krawędziowych”, można stwierdzić, że te drugie występują zawsze wtedy, gdy mowa jest o jakiejś wyjątkowej sytuacji, tzn. przebieg procesu nie jest normalny, bo konieczne jest zrealizowanie dodatkowych (wyjątkowych) czynności. Wynika to albo wprost z definicji poszczególnych zdarzeń „krawędziowych” (*Komunikat*, *Timer*, *Warunkowe*, *Sygnal*, *Wielozdarzenie*), albo z sensu tych zdarzeń (*Eskalacja*, *Błąd*, *Anulowanie*, *Kompensacja*). Tylko pośrednie wielozdarzenie równoległe, niezależnie od tego, czy jest umieszczone na krawędzi, czy w zwykłym przepływie, ma takie samo znaczenie i nie musi być związane z przepływem wyjątkowym. Więcej informacji o przepływach normalnych i wyjątkowych można znaleźć w rozdziale 5.





Rysunek 8.7. Model procesu ze zdarzeniami krawędziowymi (czynność Uzupełnienie magazynu)








8.1.2.1. Zdarzenia pośrednie „zwykłe”

W tabeli 8.5 wyjaśnione są wszystkie „zwykłe” zdarzenia pośrednie. „Zwykłe”, czyli takie, które nie są krawędziowymi zdarzeniami pośrednimi.






Tabela 8.5. Zdarzenia pośrednie „zwykłe”

Typ	Opis	Znacznik
Nieokreślony	Służy do oznaczenia jakiegś (niedoprecyzowanej) zmiany stanu w procesie. To zdarzenie może być umieszczone na modelu wtedy, gdy wiemy, że występuje jakieś zdarzenie pośrednie, ale nie wnioskujemy, jaki jest jego rodzaj. Specyfikacja BPMN 2.0 nie definiuje szczegółowo, kiedy można, a kiedy nie można zastosować zdarzenia pośredniego bez wyzwalacza. W praktyce wstępne wersje wszystkich modeli na ogół zawierają właśnie takie zdarzenia pośrednie, a dopiero w późniejszych etapach modelowania są „przeprawiane” na konkretne rodzaje zdarzeń (np. komunikaty).	Tworzenie 

**Tabela 8.5.** Zdarzenia pośrednie „zwykłe” — ciąg dalszy

Typ	Opis	Znacznik
<b>Komunikat</b>	Jeśli w trakcie procesu następuje wysłanie komunikatu, należy zastosować pośrednie zdarzenie tworzenia komunikatu (wypełniona koperta).	Tworzenie 
	Jeśli w trakcie procesu konieczne jest odebranie komunikatu, należy zastosować pośrednie zdarzenie wychwycenia (odebrania) komunikatu (koperta niewypełniona).	Przechwycenie 
<b>Timer</b>	Jeśli kontynuacja procesu zależy od jakiegoś terminu lub cyklu w czasie, wtedy należy zastosować pośrednie zdarzenie wychwycenia czasu.	Przechwycenie 
<b>Eskalacja</b>	Jeśli w trakcie procesu konieczne jest eskalowanie jakiejś sytuacji w procesie nadrzędnym (np. realizacja działania przedłuża się i konieczne jest raportowanie problemów do kogoś, kto nadzoruje proces), wtedy należy użyć zdarzenia pośredniego tworzenia eskalacji.	Tworzenie 
<b>Kompensacja</b>	Jeśli w trakcie procesu zachodzi konieczność wycofania wcześniej zrealizowanych czynności i przeprowadzenia innych (dalszych), należy użyć zdarzenia pośredniego tworzenia kompensacji.	Tworzenie 
<b>Warunkowe</b>	Jeśli kontynuacja procesu zależy od spełnienia jakiegoś warunku logicznego, należy zastosować zdarzenie pośrednie warunkowe. Warunkiem może być wyrażenie w języku naturalnym przypisane do zdarzenia warunkowego.	Przechwycenie 
<b>Łączy</b>	Ten typ zdarzenia pośredniego umożliwia połączenie dwóch części procesu biznesowego w jeden ciąg. Pozornie może się wydawać, że takie zdarzenie nie ma sensu, zawsze bowiem można połączyć elementy modelu przepływem sekwencji. Jednak używając łączy, możemy, po pierwsze, uniknąć zbyt długich linii przepływów sekwencji, a po drugie, odpowiednio operując pośrednimi zdarzeniami łączy tworzonych i wychwytywanych, w łatwy sposób zamodelować pętle i skoki („go to”) w procesie. Warto pamiętać, że pojedyncze zdarzenie pośrednie tworzące łączy może mieć wiele odwołujących się do niego zdarzeń pośrednich wychwytywanych.	Tworzenie 
		Przechwycenie 








**Tabela 8.5.** Zdarzenia pośrednie „zwykłe” — ciąg dalszy

Typ	Opis	Znacznik
<i>Sygnal</i>	Jeśli w trakcie działania procesu następuje wysłanie sygnału lub proces oczekuje na jakiś sygnał, należy zastosować zdarzenie pośrednie sygnału. Zgodnie z tym, jak zdefiniowane jest to w specyfikacji BPMN 2.0, sygnały należy traktować podobnie jak flary świetlne wyrzucane w powietrze w celu przekazania jakiejś informacji wszystkim, którzy mogą być nią zainteresowani.	Tworzenie 
		Przechwycenie 
<i>Wielozdarzenie</i>	Wielozdarzenie pośrednie jest kombinacją wielu różnych zdarzeń pośrednich. Jeśli chcemy w modelu zaprezentować zdarzenie związane z przechwyceniem jednego z wielu zdarzeń składających się na wielozdarzenie, musimy zastosować pośrednie wielozdarzenie wychwytyjące. Jeśli natomiast chcemy zaprezentować zdarzenie związane z wygenerowaniem wielu zdarzeń składających się na wielozdarzenie, to musimy zastosować pośrednie wielozdarzenie tworzące. W przypadku wielozdarzenia przechwytyjącego wystarczy, że zajdzie którekolwiek zdarzenie składowe. W przypadku wielozdarzenia tworzącego zachodzą wszystkie zdarzenia składowe.	Tworzenie 
		Przechwycenie 
<i>Wielozdarzenie równoległe</i>	Pośrednie wielozdarzenie równoległe występuje tylko jako przechwycenie i różni się od zwykłego wielozdarzenia pośredniego tym, że muszą zajść wszystkie zdarzenia składające się na to wielozdarzenie, a nie tylko jedno z nich.	Przechwycenie 





### 8.1.2.2. Zdarzenia pośrednie krawędziowe

Uwaga: wszystkie zdarzenia krawędziowe mogą tylko *przechwytywać* wyzwalacze. Choć więc umieszczamy je na krawędzi jakiejś czynności, są one wywoływane w zupełnie innym miejscu. Notacja „krawędziowa” służy jedynie — jak już wcześniej wspomniano — do prezentowania wyjątkowych sytuacji, gdy następuje odstępstwo od zwykłego przebiegu procesu i rozpoczyna się przepływ wyjątkowy (tabela 8.6).





**Tabela 8.6.** Zdarzenia pośrednie krawędziowe

Typ	Opis	Znacznik
<b>Komunikat</b>	Jeśli w trakcie procesu realizacja jakiejś czynności musi być anulowana, ponieważ pojawia się określony komunikat i następuje realizacja wyjątkowych (nierutynowych) działań, należy zastosować przerywające zdarzenie krawędziowe przechwytyjące komunikatu. Jeśli natomiast trzeba zrealizować te działania, ale bez anulowania czynności, należy zastosować to zdarzenie w wersji nieprzerywającej. W obu przypadkach to, co następuje po krawędziowym zdarzeniu komunikatu, należy traktować jako przepływ wyjątkowy, a nie normalny.	Przerywające  Nieprzerywające 
<b>Timer</b>	Jeśli w trakcie realizacji procesu jakaś czynność musi być anulowana, ponieważ następuje określony moment w czasie (np. 9 maja 1968 roku) i konieczne jest przeprowadzenie wyjątkowych (nierutynowych) działań, należy zastosować przerywające zdarzenie krawędziowe przechwytyjące timera. Jeśli natomiast trzeba zrealizować te działania, ale bez anulowania czynności, należy zastosować to zdarzenie w wersji nieprzerywającej. W obu przypadkach to, co następuje po krawędziowym zdarzeniu timera, należy traktować jako przepływ wyjątkowy, a nie normalny.	Przerywające  Nieprzerywające 
<b>Eskalacja</b>	Jeśli w trakcie realizacji procesu jakaś czynność musi być anulowana, ponieważ następuje zdarzenie eskalacji i konieczne jest w związku z tym przeprowadzenie wyjątkowych działań, należy zastosować przerywające zdarzenie krawędziowe przechwytyjące eskalacji. Jeśli natomiast trzeba zrealizować te działania, ale bez anulowania czynności, należy zastosować to zdarzenie w wersji nieprzerywającej.	Przerywające  Nieprzerywające 
<b>Błąd</b>	Pośrednie zdarzenie przechwycenia błędu występuje jedynie w formie krawędziowej i zawsze przerywa ono czynność, do której jest przypięte. Jeśli zdarzenie to ma nazwany błąd, którego dotyczy, to nastąpi ono jedynie wtedy, gdy wystąpi ów konkretny błąd. Jeśli natomiast nie ma nazwy, to nastąpi zawsze, gdy wystąpi jakikolwiek błąd.	Przerywające 

**Tabela 8.6.** Zdarzenia pośrednie krawędziowe — ciąg dalszy

Typ	Opis	Znacznik
<b>Anulowanie</b>	To zdarzenie może być umieszczone na krawędzi podprocesu tylko wówczas, gdy jest to transakcja. Jeśli zostanie wysłany komunikat anulowania transakcji lub wewnątrz transakcji nastąpi zdarzenie końcowe anulowania, transakcja jest przerywana i następują działania wychodzące z tego zdarzenia.	Przerywające 
<b>Kompensacja</b>	To zdarzenie następuje, gdy uruchomiony zostanie wyzwalacz kompensacji. Kompensacja nie przerywa ani nie anuluje czynności, ponieważ z definicji każda kompensacja może nastąpić dopiero, gdy dany proces się zakończy. Więcej o kompensacjach znajdziesz w podrozdziale 9.3.1.	Przerywające 
<b>Warunkowe</b>	To zdarzenie następuje, gdy spełnione są wszystkie warunki określone wyrażeniem warunkowym przypisanym do tego zdarzenia. Jeśli w przypadku spełnienia tych warunków czynność ma zostać przerwana, należy zastosować wersję przerywającą tego zdarzenia. W przeciwnym razie — wersję nieprzerywającą. W obu przypadkach to, co następuje po krawędziowym zdarzeniu warunkowym, należy traktować jako przepływ wyjątkowy, a nie normalny.	Przerywające  Nieprzerywające 
<b>Sygnal</b>	Jeśli w trakcie realizacji procesu jakaś czynność musi być anulowana, ponieważ pojawia się określony sygnał i konieczne jest przeprowadzenie nietypowych działań, należy zastosować przerywające zdarzenie krawędziowe sygnału. Jeśli natomiast trzeba zrealizować te działania, ale bez anulowania czynności, należy zastosować to zdarzenie w wersji nieprzerywającej. W obu przypadkach to, co następuje po krawędziowym zdarzeniu sygnału, należy traktować jako przepływ wyjątkowy, a nie normalny.  Sygnał jest czymś innym niż komunikat, ponieważ nie ma jednego konkretnego odbiorcy i jest wysyłany „do wszystkich”, ale odbierają go tylko „zainteresowani”.	Przerywające  Nieprzerywające 

**Tabela 8.6.** Zdarzenia pośrednie krawędziowe — ciąg dalszy

Typ	Opis	Znacznik
<b>Wielozdarzenie</b>	Wielozdarzenie pośrednie łączy w sobie wiele zdarzeń pośrednich, przy czym do jego aktywacji wystarczy, że nastąpi jedno ze zdarzeń składowych. Jeśli czynność, do której jest przypięte to zdarzenie, ma zostać przerwana, należy użyć wariantu przerywającego. W przeciwnym razie — wariantu nieprzerywającego. W obu przypadkach to, co następuje po krawędziowym wielozdarzeniu, należy traktować jako przepływ wyjątkowy, a nie normalny.	Przerywające 
		Nieprzerywające 
<b>Wielozdarzenie równoległe</b>	Zdarzenie to ma prawie takie samo znaczenie jak zwykłe wielozdarzenie, z tą różnicą, że muszą zajść wszystkie zdarzenia składowe.	Przerywające 
		Nieprzerywające 

### 8.1.3. Zdarzenia końcowe

Zdarzenie końcowe, reprezentowane na modelach przez okrąg z pojedynczą, pogrubioną krawędzią, oznaczają zakończenie procesu, w związku z tym nie mogą z nich wychodzić przepływy sekwencji. Reguły modelowania zdarzeń końcowych są podobne do reguł obowiązujących w przypadku zdarzeń początkowych.

Każdy model procesu może zawierać zero lub wiele zdarzeń końcowych. Jeśli model zawiera zdarzenie początkowe, to obowiązkowe jest co najmniej jedno zdarzenie końcowe. Innymi słowy, jeśli jest to uzasadnione, możemy pominąć zarówno początkowe, jak i końcowe zdarzenie w modelu. Nigdy jednak nie możemy pominąć zdarzenia końcowego, jeśli proces ma jakiś rezultat (wynik działania) lub gdy model zawiera zdarzenie początkowe.



Rezultatem procesu może być wyzwalacz: *Nieokreślony*, *Komunikat*, *Błąd*, *Eskalacja*, *Anulowanie*, *Kompensacja*, *Sygnal*, *Przerwanie*, *Wielozdarzenie*. W tabeli poniżej wyjaśniono znaczenie każdego z tych rezultatów (tabela 8.7).

Podobnie jak w przypadku zdarzenia początkowego, powyższe reguły odnoszą się do każdego poziomu niezależnie: procesów, podprocesów, podprocesów zdarzeniowych. Jeśli więc na poziomie podprocesu zastosowaliśmy zdarzenie początkowe, to musimy zastosować zdarzenie końcowe, nie musimy jednak z tego powodu stosować zdarzenia końcowego w procesie nadrzędnym.

Tabela 8.7. Zdarzenia końcowe

Typ	Opis	Znacznik
<i>Nieokreślony</i>	Proces kończy się, ale nie jest to związane z wytworzeniem jakiegoś rezultatu lub wytworzony rezultat nie jest <i>Komunikatem</i> , <i>Błędem</i> , <i>Eskalacją</i> , <i>Anulowaniem</i> , <i>Kompensacją</i> , <i>Sygnałem</i> , <i>Przerwaniem</i> albo <i>Wielozdarzeniem</i> .	
<i>Komunikat</i>	Proces kończy się i jego wynikiem jest przekazanie jakiegoś komunikatu do innego <i>uczestnika</i> . Opcjonalnie, jeśli chcemy, możemy połączyć zdarzenie końcowe komunikatu <i>przepływem komunikatu</i> z jego odbiorcą, czyli <i>uczestnikiem</i> .	
<i>Błąd</i>	Takie zdarzenie końcowe oznacza, że w procesie wystąpił błąd. Wszystkie niezależne wątki procesu są przerywane. Jeśli błąd jest nazwany (parametr <i>errorCode</i> ), jest on przechwytywany przez zdarzenie pośrednie typu <i>Błąd</i> o tej samej nazwie (czyli takiej samej wartości parametru <i>errorCode</i> ). Jeśli nie jest nazwany, jest przechwytywany przez każde nienazwane zdarzenie pośrednie typu <i>Błąd</i> .	
<i>Eskalacja</i>	Takie zdarzenie końcowe oznacza, że wraz z zakończeniem procesu powinna nastąpić eskalacja do procesu nadrzędnego (np. realizacja działania przedłuża się i konieczne jest raportowanie problemów do kogoś, kto nadzoruje proces). Jeśli zdarzenie ma określony parametr <i>errorCode</i> , wtedy obsługą eskalacji zajmuje się podproces rozpoczynający się pośrednim zdarzeniem eskalacji o takiej samej wartości <i>errorCode</i> .	
<i>Anulowanie</i>	Ten typ zdarzenia końcowego jest używany w podprocesach transakcji. Oznacza, że transakcja powinna zostać przerwana i powinno zostać wywołane odpowiednie pośrednie zdarzenie anulowania podprocesu transakcji.	
<i>Kompensacja</i>	Ten typ zdarzenia końcowego oznacza, że niezbędna jest kompensacja, czyli wycofanie wcześniejszych działań. Kompensacja jest realizowana przez czynność, która ma przypięte do krawędzi zdarzenie kompensacji lub zawiera zdarzeniowy podproces kompensacji.	
<i>Sygnal</i>	Ten typ zdarzenia końcowego oznacza, że wraz z zakończeniem procesu rozgłaszany jest <i>sygnal</i> . Warto pamiętać, że sygnal jest rozgłaszany do każdego procesu, który może go odebrać. Inaczej jest z komunikatami, które zawsze mają określonego odbiorcę.	

**Tabela 8.7.** *Zdarzenia końcowe* — ciąg dalszy

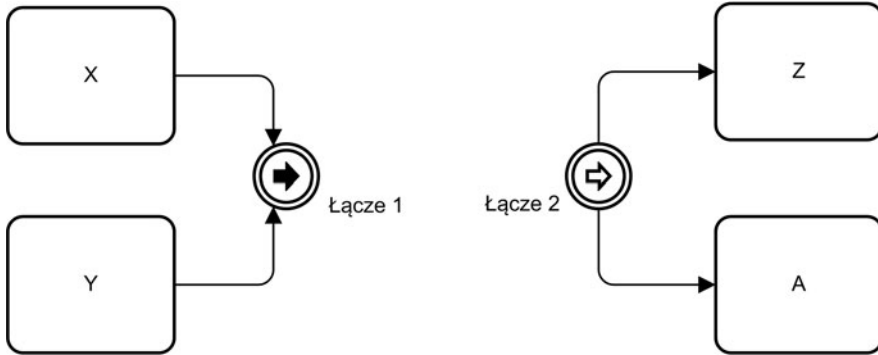
Typ	Opis	Znacznik
<b>Przerwanie</b>	Ten typ zdarzenia końcowego oznacza, że proces powinien niezwłocznie się zakończyć bez kompensacji lub obsługi zdarzeń. Przerwanie dotyczy wszystkich instancji, niezależnie od tego, czy mamy do czynienia z procesem jednoinstancyjnym, czy wieloinstancyjnym (patrz rozdział 7., „Kolaboracje i konwersacje”).	
<b>Wielozdarzenie</b>	Ten typ zdarzenia końcowego oznacza, że wraz z końcem procesu następuje wiele rzeczy, np. jednocześnie jest wysyłanych wiele komunikatów.	

## 8.2. Jak to wszystko opanować?

Do opanowania zdarzeń w standardzie BPMN 2.0 można podejść na dwa sposoby: albo wkuć cały ten rozdział na pamięć, albo zapamiętać ogólne znaczenie poszczególnych wyzwalaczy (*Komunikat*, *Eskalacja*, *Anulowanie*, *Sygnal* itd.) i pięć następujących reguł:

1. Jeśli chcesz narysować zdarzenie, z którego tylko wychodzą przepływy sekwencji, ale nie ma żadnych przepływów wchodzących, to nie może to być zdarzenie pośrednie ani końcowe. Musi to być zdarzenie początkowe, ewentualnie krawędziowe.
2. Jeśli chcesz narysować zdarzenie, które ma zarówno przepływy sekwencji wchodzące, jak i wychodzące, to nie może to być zdarzenie początkowe ani końcowe. Musi to być zdarzenie pośrednie.
3. Jeśli chcesz narysować zdarzenie, do którego tylko wchodzi przepływ sekwencji, ale nie ma żadnych przepływów wychodzących, to nie może to być zdarzenie początkowe ani pośrednie. Musi to być zdarzenie końcowe.
4. Zdarzenie pośrednie łączy źródłowego (piktogram strzałki wypełnionej) może mieć tylko wchodzące przepływy sekwencji. Zdarzenie pośrednie łączy docelowego (piktogram strzałki niewypełnionej) może mieć tylko wychodzące przepływy sekwencji (rysunek 8.8).
5. Zdarzenia przerywające (z kreskowanym konturem) występują tylko w przypadku zdarzeń początkowych podprocesów zdarzeniowych lub zdarzeń pośrednich krawędziowych. I odpowiednio: te pierwsze przerywają proces nadrzędny, te drugie przerywają czynność, do której są przypięte.





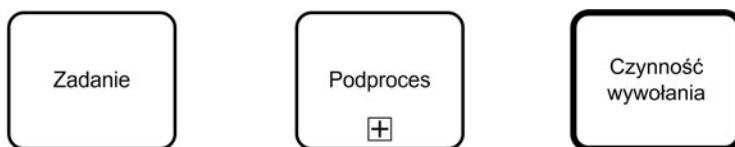
**Rysunek 8.8.** Zdarzenie pośrednie łączy



**Czynność** to pewien fragment pracy realizowanej w procesie biznesowym. W języku potocznym bardzo często zamiast określenia „czynność” lub „aktywność” używa się określenia „krok procesu”. Zgodnie ze standardem BPMN 2.0 czynnościami w procesach biznesowych mogą być:

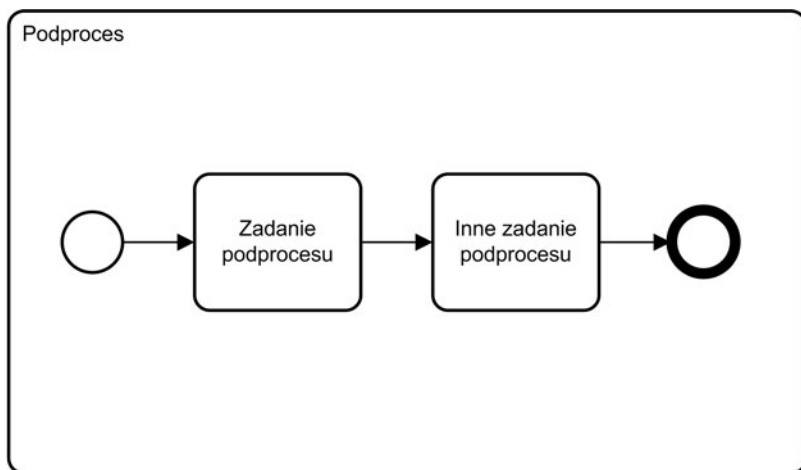
- **Zadania**, czyli czynności atomowe, tj. takie, których nie dekomponujemy na mniejsze, choć możemy określić ich typ: *usługowe, wysłania, odebrania, użytkownika, manualne, skryptowe, reguł biznesowych*.
- **Podprocesy**, czyli czynności nieatomowe, a dokładniej: *podprocesy zdarzeniowe, podprocesy ad hoc* lub *podprocesy transakcji*.
- **Czynności wywołania**, czyli czynności, które funkcjonują jako łącza do innych czynności (zadań i podprocesów).

W notacji BPMN 2.0 zadanie jest reprezentowane przez piktogram prostokąta z zaokrąglonym rogami, podproces ma dodatkowo wewnątrz, przy dolnej krawędzi, znak plusa (+), symbolizujący fakt, że dana czynność jest złożona, natomiast czynność wywołania ma pogrubioną krawędź (rysunek 9.1). Podprocesy mogą także występować na modelach w postaci rozwiniętej — wtedy piktogram czynności nie zawiera wewnątrz znaku plusa, ponieważ w jej wnętrzu jest umieszczony konkretny model tego podprocesu (rysunek 9.2).



**Rysunek 9.1.** Zadanie, podproces zwinięty i czynność wywołania

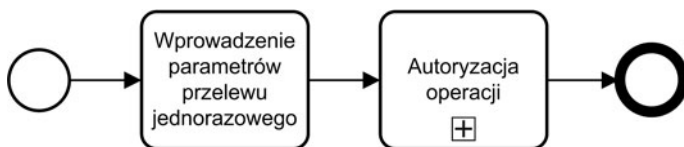
Dodatkowo w odniesieniu do czynności w standardzie BPMN 2.0 możemy stosować tzw. *znaczniki*, które pozwalają na dokładniejsze specyfikowanie tego, w jaki sposób są realizowane poszczególne czynności. Zanim jednak przejdziemy do



**Rysunek 9.2.** Podproces w postaci rozwiniętej

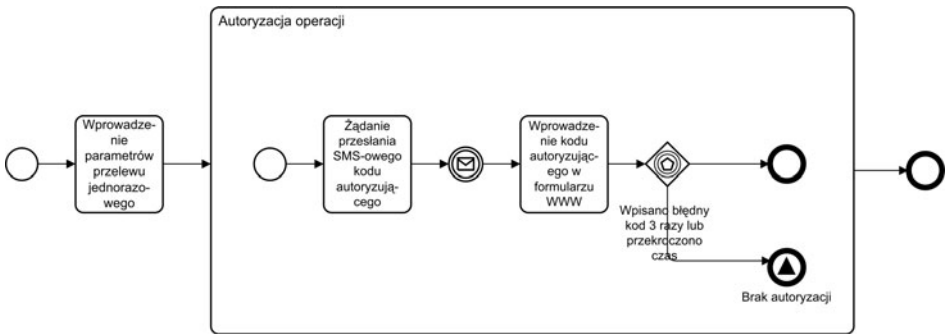
omawiania poszczególnych typów zadań, podprocesów, czynności wywołania i znaczników, wyjaśnijmy na przykładzie procesu tworzenia dyspozycji przelewu bankowego różnicę między zadaniem i podprocesem.

Przyjmijmy, że nasz uproszczony proces tworzenia dyspozycji przelewu jednorazowego (rysunek 9.3) składa się z dwóch czynności: atomowej (*Wprowadzenie parametrów przelewu jednorazowego*) i nieatomowej (*Autoryzacja operacji*). Przyjmijmy także, że poprzez czynność *Wprowadzenie parametrów przelewu jednorazowego* rozumiemy wypełnienie przez klienta banku standardowego formularza WWW (kwota przelewu, numer rachunku odbiorcy, nazwa i adres odbiorcy, data realizacji przelewu). *Autoryzacja operacji* w naszym procesie polega natomiast na wygenerowaniu i odebraniu SMS-owego kodu autoryzującego przez telefon komórkowy, a następnie wprowadzeniu tego kodu do formularza WWW.



**Rysunek 9.3.** Proces składający się z (od lewej) czynności atomowej i podprocesu zwiniętego

Rysunek 9.4 przedstawia ten sam model procesu tworzenia dyspozycji przelewu, ale z rozwiniętym podprocesem *Autoryzacja operacji*. W notacji BPMN 2.0 podproces rozwinięty jest reprezentowany w ten sposób, że całe jego „wnętrze”, czyli składowe czynności, zdarzenia i bramy, jest umieszczone wewnątrz piktogramu czynności bez symbolu plusa — odwrotnie niż w przypadku podprocesu zwiniętego.



**Rysunek 9.4.** Proces składający się z (od lewej) czynności atomowej i podprocesu rozwiniętego

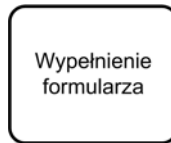
Ktoś mógłby powiedzieć, że wprowadzanie parametrów przelewu jednorazowego też jest czynnością nieatomową i — podobnie jak autoryzacja operacji — podprocesem. Użytkownik przecież musi wprowadzić wiele wartości (kwota, numer rachunku odbiorcy, nazwa odbiorcy itd.), a system może sprawdzić poprawność tych danych i np. w przypadku błędu w numerze rachunku zgłosić konieczność korekty itd. Są to jednak czynności, których dekomponowanie na mniejsze nie ma znaczenia z punktu widzenia analizy procesu biznesowego — prowadzą one bowiem do jednego rezultatu: zestawu konkretnych parametrów niezbędnych do realizacji przelewu. Można powiedzieć, że z perspektywy procesu nie ma znaczenia kolejność wypełniania formularza przelewu oraz to, czy użytkownik kilka razy się pomylił i system skorygował wprowadzone wartości. Są to czynności istotne dla projektanta systemu bankowości elektronicznej, ale nie projektanta procesu realizacji operacji na rachunku bankowym.

Dodatkowo „zapakowanie” jakiejś czynności złożonej do podprocesu ułatwia nam analizowanie i modelowanie procesów w ogóle. W ten sposób możemy bowiem zastosować tzw. myślenie abstrakcyjne, niezwykle przydatne przy analizie biznesowej i inżynierii oprogramowania. Jest to sposób analizowania i rozwiązywania problemów, w którym całe zagadnienie dzielimy na mniejsze, dla nich opracowujemy rozwiązania, a następnie operujemy abstrakcyjnymi nazwami tych rozwiązań. Jest to bardzo wygodny mechanizm, który stosujemy wręcz intuicyjnie, gdy jakiś zestaw zadań jest dla nas oczywisty, zrozumiały i powtarzalny.

## 9.1. Zadania

Zadanie to atomowa (niepodzielna na mniejsze) czynność w procesie biznesowym. W standardzie BPMN 2.0 wyróżnia się 8 typów zadań: *usługowe*, *wysłania*, *odebrania*, *odebrania uruchamiające proces*, *użytkownika*, *ręczne*, *reguły biznesowej*, *skryptowe*. Jeśli chcemy modelować proces bez specyfikowania typu zadań — co

jest dopuszczalne w standardzie BPMN 2.0 — musimy używać **zadań abstrakcyjnych** (*abstract task*), które są reprezentowane przez prostokąty z zaokrąglonymi rogami bez żadnych dodatkowych symboli (rysunek 9.5).



**Rysunek 9.5.** Zadanie abstrakcyjne (tzn. niemające zdefiniowanego typu)

Jeśli natomiast modelujemy szczegółowo proces biznesowy, bo np. ma być uruchomiony na silniku procesów biznesowych, wtedy musimy konkretnie określać typy zadań. Zadania o określonych typach są oznaczone graficznie dodatkowymi ikonami w lewym górnym rogu symbolu zadania. W dalszej części tego rozdziału zaprezentowano i omówiono wszystkie typy zadań.

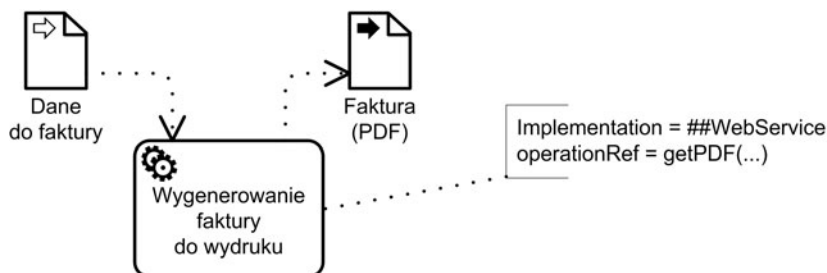
#### 9.1.1. Zadanie usługowe (service task)

Zadanie usługowe to takie, które jest realizowane całkowicie automatycznie, bez udziału człowieka. W notacji BPMN 2.0 zadanie usługowe jest reprezentowane przez prostokąt z zaokrąglonymi rogami oraz ikoną usługi (rysunek 9.6). Domyślnie jest to usługa sieciowa (*Web Service*), ale może to być także jakaś operacja wywoływana przy użyciu innej technologii. O tym, jaka technologia jest stosowana do implementacji zadania usługowego, decyduje parametr `implementation`. Parametr ten może przyjmować następujące wartości: `WebService` (wartość domyślna), `unspecified` (jeśli chcemy pozostawić kwestię implementacji otwartą) lub URI, czyli adres zawierający specyfikację techniczną danej usługi. Sposób specyfikowania parametru `implementation` zależy już od konkretnego narzędzia, które wykorzystujemy do modelowania procesów biznesowych.



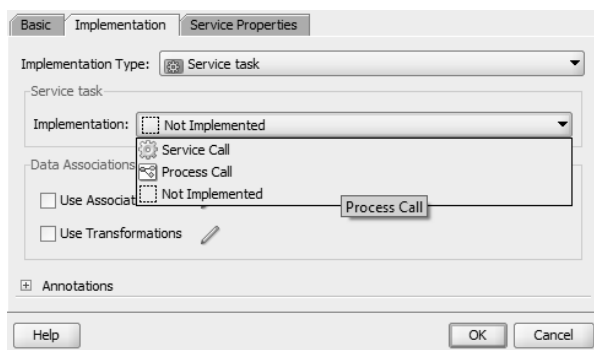
**Rysunek 9.6.** Zadanie usługowe (service task)

Zgodnie ze specyfikacją BPMN 2.0 zadanie usługowe może być związane z konkretną operacją, a ta z kolei zawsze musi być powiązana z dokładnie jednym obiektem wejściowym i co najwyżej jednym obiektem wyjściowym (rysunek 9.7). Więcej informacji na temat obiektów danych znajduje się w rozdziale 6., „Obiekty danych”.



**Rysunek 9.7.** Zadanie usługowe z obiektem wejściowym i wyjściowym

Należy jednak pamiętać, że w praktyce takie niuanse zależą bardziej od dostawcy narzędzia, które wykorzystujemy do implementacji procesu, niż od samego standardu BPMN 2.0. Przykładowo w pakiecie Oracle BPM Studio zadanie usługowe może nie mieć implementacji lub może oznaczać wywołanie usługi sieciowej, którą trzeba zbudować (skonfigurować) przy użyciu edytora Oracle SOA Composite, albo wywołanie innego procesu biznesowego zdefiniowanego w bieżącym projekcie (rysunek 9.8).



**Rysunek 9.8.** Definicja implementacji zadania usługowego w narzędziu Oracle BPM Studio

### 9.1.2. Zadanie wysłania (send task)

Zadanie wysłania to takie, którego jedynym celem jest wysłanie komunikatu do innego uczestnika. W notacji BPMN 2.0 zadanie wysłania jest reprezentowane przez prostokąt z zaokrąglonymi rogami oraz ikoną wypełnionej koperty (rysunek 9.9).



**Rysunek 9.9.** Zadanie wysłania

Jest to zadanie współrealizowane przez silnik procesów biznesowych i jakiś inny system. Nie jest natomiast realizowane przez człowieka czy użytkownika — i należy o tym pamiętać. Bardzo często osoby początkujące modelują, używając zadania wysłania lub odebrania wtedy, gdy ktoś (człowiek) coś wyśle. W takim przypadku należy po prostu zastosować zadanie użytkownika lub zadanie manualne i ewentualnie pokazać, że jest to czynność związana z wysłaniem komunikatu — poprzez zastosowanie przepływu komunikatu.

Ponieważ zadanie wysłania z definicji jest związane z jakimś komunikatem, zgodnie z notacją BPMN 2.0 można je powiązać przepływem komunikatu z innym elementem modelu, np. z zadaniem lub zdarzeniem. Nie jest to jednak obowiązkowe i same zadania wysłania bez przepływów komunikatów są całkowicie poprawne.

### 9.1.3. Zadanie odebrania (receive task)

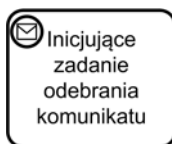
Zadanie odebrania — analogicznie do zadania wysłania — jest zadaniem, którego jedynym celem jest odebranie komunikatu od innego uczestnika. W notacji BPMN 2.0 zadanie odebrania jest reprezentowane przez prostokąt z zaokrąglonymi rogami oraz ikoną niewypełnionej koperty (rysunek 9.10).



**Rysunek 9.10.** *Zadanie odebrania*

Zadanie odebrania odbywa się — tak samo jak zadanie wysłania — bez udziału człowieka. I podobnie jak zadanie wysłania, może być związane przepływem komunikatu z innym elementem modelu.

Zadanie odebrania może dodatkowo występować w wariancie instancjującym, czyli uruchamiającym proces (atrybut `instantiated` ma wtedy wartość `true`) — o ile jest pierwszą czynnością procesu, tzn. nie ma żadnych przepływów sekwencji wchodzących do tego zadania. W tym wariancie zadanie odebrania jest reprezentowane przez prostokąt z zaokrąglonymi rogami oraz ikoną niewypełnionej koperty w kółku (rysunek 9.11).



**Rysunek 9.11.** *Instancjujące zadanie odebrania*



### 9.1.4. Zadanie użytkownika (user task) i zadanie manualne (manual task)

Zadanie użytkownika i zadanie manualne to jedyne zadania, które są realizowane przez człowieka. Różnica między nimi jest taka, że zadanie użytkownika jest realizowane przez człowieka, ale z wykorzystaniem aplikacji komputerowej, natomiast zadanie manualne jest realizowane całkowicie poza systemami oprogramowania. Przykładem zadania manualnego może być stemplowanie dokumentów — w wielu procesach jest to istotna czynność, ale rzadko realizowana inaczej niż ręcznie przez pracownika. W notacji BPMN 2.0 zadania użytkownika i manualne są reprezentowane przez prostokąt z zaokrąglonymi rogami oraz odpowiednimi ikonami (rysunek 9.12).



**Rysunek 9.12.** Zadanie użytkownika (po lewej) i zadanie manualne (po prawej)

W przypadku zadań manualnych warto zauważyć, że choć nie są one w jakikolwiek sposób kontrolowane przez silnik procesów biznesowych, to jednak uwzględniając je w modelu, można coś o nich powiedzieć. Po pierwsze, dzięki temu, że umieszczamy je w modelu, w ogóle wiadomo, że takie zadania są realizowane w procesie. Nie wszystko, co jest oczywiste dla osoby znającej dany proces, jest takie samo dla innych (np. menedżerów wysokiego szczebla). Po drugie, nawet jeśli jakieś zadanie jest realizowane poza systemami oprogramowania i silnikiem procesów biznesowych, zawsze możemy wyświetlić instrukcję wyjaśniającą, jak należy je realizować, i ewentualnie zażądać od pracownika zaznaczenia na formularzu, że wykonał wszystko, jak należy (tzw. lista kontrolna).

### 9.1.5. Zadanie reguły biznesowej (business rule task)

Zadanie reguły biznesowej jest specyficznym typem zadania, ponieważ jest ono realizowane przez dwa systemy: przez silnik procesów biznesowych oraz silnik reguł biznesowych. Rolą silnika procesów biznesowych jest zebranie określonych danych wejściowych i przekazanie ich do silnika reguł biznesowych. Natomiast rolą silnika reguł biznesowych jest obliczenie wartości reguły biznesowej i zwrócenie jej do silnika procesów biznesowych. W notacji BPMN 2.0 zadanie reguły biznesowej jest reprezentowane przez prostokąt z zaokrąglonymi rogami oraz ikoną reguły biznesowej (rysunek 9.13).

Stosowanie zadań reguł biznesowych jest bardzo dobrym rozwiązaniem, umożliwia bowiem odseparowanie logiki biznesowej od logiki procesu biznesowego.



**Rysunek 9.13.** *Zadanie reguły biznesowej*

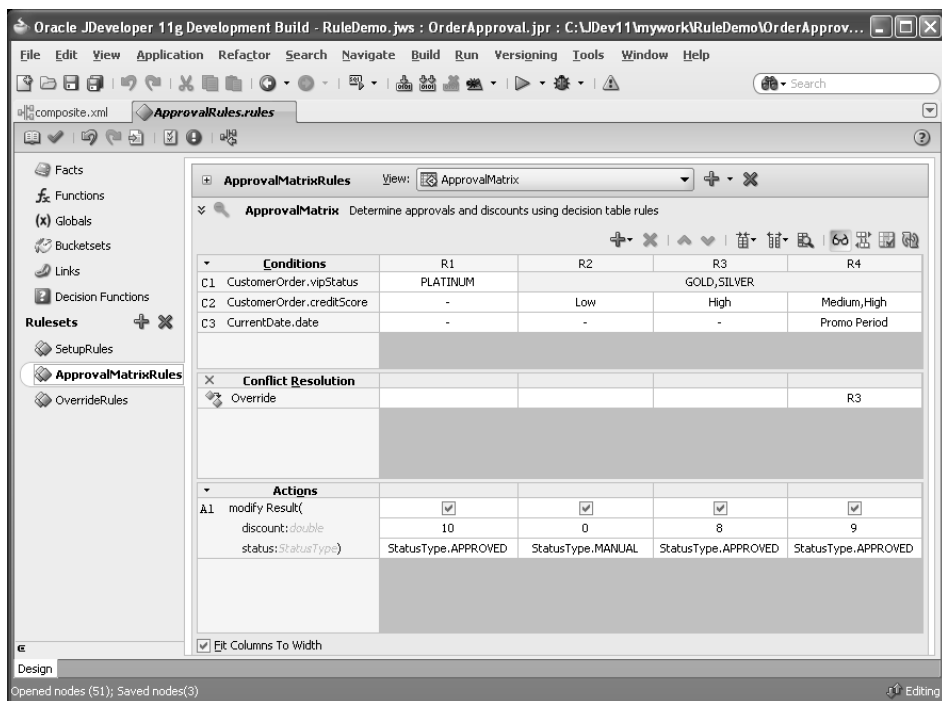
Przykładem może być proces biznesowy przyznawania kredytu hipotecznego. Z jednej strony, każdy, kto chce otrzymać kredyt hipoteczny od banku, musi dostarczyć pewne dokumenty i zrealizować standardową procedurę postępowania, z drugiej, ostateczna decyzja kredytowa jest podejmowana w oparciu o szczegółowe algorytmy zależne od różnych wartości, np. od dopuszczalnego wieku kredytobiorcy ustalonego na 21 lat, minimalnego dochodu miesięcznego ustalonego na 2000 zł itd. Dzięki odseparowaniu tych przestrzeni od siebie, możliwe jest np. modyfikowanie algorytmu podejmowania decyzji kredytowej bez jakiegokolwiek zmiany w logice procesu.

Wadą stosowania zadań reguł biznesowych jest konieczność wdrożenia i utrzymywania silnika reguł biznesowych oraz zarządzania samymi regułami biznesowymi (ktoś musi je zdefiniować, ktoś musi sprawdzić i potwierdzić, ktoś musi je aktualizować itd.). Na ogół jednak dostawcy systemów klasy BPMS oferują pakiety, w których silnik reguł biznesowych jest dodatkiem do silnika procesów biznesowych i jest zintegrowany z interfejsem do wygodnego definiowania reguł biznesowych w postaci tabelarycznej, skryptowej lub wizualnej. Na rysunku 9.14 przedstawiony jest przykładowy sposób definiowania reguły biznesowej w narzędziu Oracle JDeveloper.

Zadanie reguły biznesowej, podobnie jak zadanie usługowe (*service task*), może mieć określony parametr `implementation`, określający technologię, w której jest ono implementowane. Dopuszczalne wartości parametru `implementation` dla zadania reguły biznesowej to: `##unspecified` (domyślna), `##WebService` lub adres URI wskazujący zasób definiujący szczegóły implementacji zadania.

### 9.1.6. Zadanie skryptowe (*script task*)

Zadanie skryptowe to zadanie realizowane przez silnik procesów biznesowych w oparciu o skrypt przygotowany przez analityka lub programistę procesów biznesowych. Język skryptowy wykorzystany do zdefiniowania zadania skryptowego zależy od silnika procesów, z którego korzystamy — standard BPMN 2.0 nie definiuje składni języka skryptowego dla zadań skryptowych. Może to być np. skrypt w języku Groovy. W notacji BPMN 2.0 zadanie skryptowe jest reprezentowane przez prostokąt z zaokrąglonymi rogami i ikoną skryptu (rysunek 9.15).

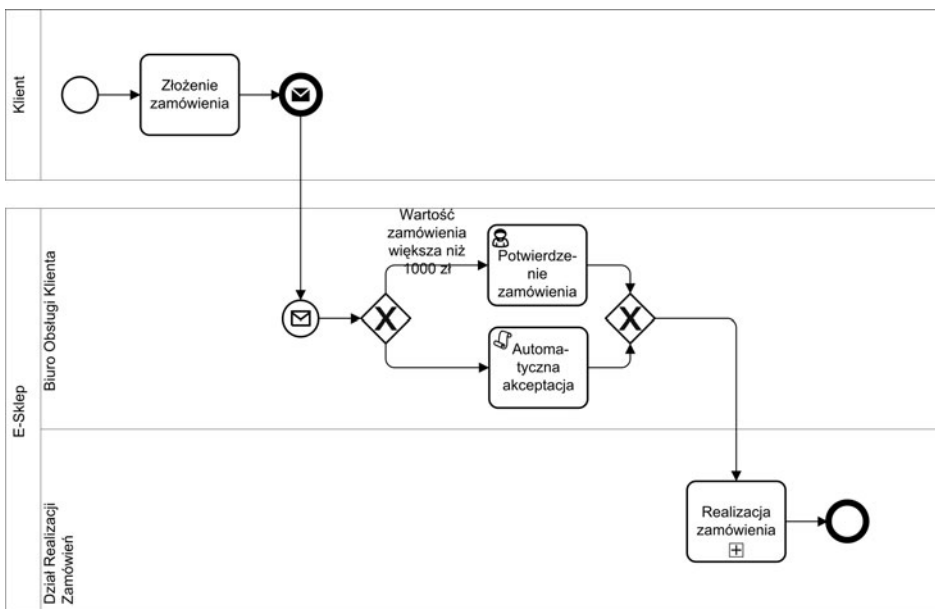


Rysunek 9.14. Przykładowy interfejs edycji reguł biznesowych



Rysunek 9.15. Zadanie skryptowe

Zadania skryptowe pojawiają się w tych miejscach procesu, gdzie konieczne jest przygotowanie obiektów danych (patrz rozdział 6.) lub ich przetworzenie. Dobrym przykładem tego, w jaki sposób można wykorzystywać zadania skryptowe, może być proces obsługi zamówień w sklepie elektronicznym, w którym do pewnej wartości zamówienia nie ma konieczności jego potwierdzania przez pracownika Biura Obsługi Klienta (rysunek 9.16). W procesie tym zamówienia o niedużej wartości są automatycznie (za pomocą zadania skryptowego) akceptowane i przekazywane do realizacji. Tylko zamówienia, których wartość przekracza określoną kwotę, wymagają angażowania człowieka.



**Rysunek 9.16.** Proces obsługi zamówień w e-sklepie

### 9.1.7. Tabela typów zadań

Zrozumienie typów zadań jest o wiele prostsze, jeśli przyporządkuje się je do odpowiednich elementów architektury SOA (*Service-Oriented Architecture*). Załóżmy, że w organizacji, w której implementujemy i wdrażamy procesy biznesowe, funkcjonują:

- szyna usługowa, czyli oprogramowanie wyspecjalizowane w kolejkowaniu i przekazywaniu usług pomiędzy systemami funkcjonującymi w danej architekturze,
- systemy dziedzinowe, czyli systemy udostępniające konkretne usługi, specyficzne dla dziedziny, którą obejmują (np. system wydruku świadczy usługi transformacji obiektów danych na dokumenty PDF w oparciu o zadane szablony, system magazynowy świadczy usługi umożliwiające śledzenie i aktualizowanie stanów magazynowych itd.),
- silnik procesów biznesowych, czyli oprogramowanie, które wykonuje zdefiniowane modele procesów biznesowych,
- silnik reguł biznesowych, czyli oprogramowanie, które oblicza wartości reguł biznesowych oraz umożliwia zarządzanie nimi (przykładowo złożoną regułą biznesową może być algorytm wyliczania zdolności kredytowej osoby fizycznej, w którym uprawniony pracownik ustala wartości pewnych stałych, np.

minimalny dochód roczny, minimalny i maksymalny wiek itd., oraz algorytm „widełkowania” danego klienta na podstawie jego danych),

- system obiegu zadań (*workflow*), czyli interfejs umożliwiający uczestnikom procesu realizację jego zadań poprzez wypełnianie odpowiednich formularzy i przeglądanie listy zadań, spraw itd.

W poniższej tabeli (tabela 9.1) poszczególne typy zadań są przyporządkowane do poszczególnych elementów architektury SOA, a także wskazany jest efekt wykonania danego zadania.

**Tabela 9.1.** Architektura SOA i typy zadań

Typ zadania	Zaangażowane elementy SOA	Rezultat wykonania
Zadanie usługowe	System dziedziny Silnik procesów biznesowych	Wyjściowy obiekt
Zadanie wysłania	System dziedziny Silnik procesów biznesowych	Wysłany komunikat
Zadanie odebrania	System dziedziny Silnik procesów biznesowych	Odebrany komunikat [Uruchomiony proces]
Zadanie użytkownika	Silnik procesów biznesowych System <i>workflow</i>	Obiekty danych
Zadanie manualne	–	(dowolny, ale niekontrolowany przez silnik procesów biznesowych)
Zadanie reguły biznesowej	Silnik procesów biznesowych Silnik reguł biznesowych	Wartość reguły biznesowej
Zadanie skryptowe	Silnik procesów biznesowych	Wartości parametrów LUB Obiekty danych

Zrozumienie poszczególnych typów zadań bez znajomości architektury SOA i przeznaczenia poszczególnych elementów (komponentów) funkcjonujących w takiej architekturze może być niezwykle trudne. Jeśli więc Czytelnik nie posiada takiej wiedzy, warto, aby ją uzupełnił, zapoznając się z innymi publikacjami, które wyjaśniają zagadnienia architektury SOA. Całość komplikuje fakt, że dziś dostawcy systemów klasy BPMS (*Business Process Management System*) nie oferują wyłącznie

silnika procesów biznesowych, ale zintegrowane środowisko składające się z silnika procesów biznesowych, silnika reguł biznesowych oraz systemu *workflow*. W związku z tym często użytkownicy tych systemów nie do końca wiedzą, że mają do czynienia ze zintegrowanym rozwiązaniem, a nie zwykłą aplikacją komputerową, co dodatkowo utrudnia zrozumienie sensu poszczególnych typów zadań.

#### 9.1.8. Zadanie globalne (global task)

Zadanie globalne to takie, które może być wywoływane w innych miejscach procesu poprzez **czynność wywołania**. Zadanie globalne nie ma żadnej specjalnej reprezentacji w notacji BPMN 2.0. Jeśli gdzieś w modelu zastosujemy czynność wywołania, która odnosi się do jakiegoś zadania, zadanie to automatycznie staje się zadaniem globalnym. Zadanie globalne należy raczej traktować jako pojęcie abstrakcyjne aniżeli typ zadania. Zadanie każdego typu może być zadaniem globalnym.

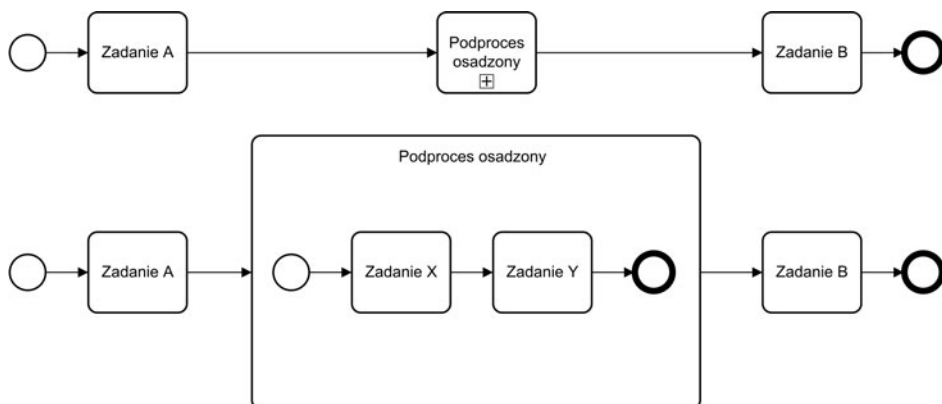
## 9.2. Podprocesy

Na początku rozdziału wyjaśniliśmy na przykładzie, jaka jest różnica między podprocesami i czynnościami atomowymi. Teraz skoncentrujemy się na samych podprocesach. W standardzie BPMN 2.0 zdefiniowano cztery rodzaje podprocesów: osadzone, zdarzeniowe, ad hoc i transakcje. Dodatkowo każdy podproces może być podprocesem globalnym. Podproces globalny to proces, który może być wywoływany jako podproces w wielu miejscach poprzez **czynność wywołania**. W notacji BPMN 2.0 proces globalny nie ma żadnej specjalnej reprezentacji graficznej.

#### 9.2.1. Podproces osadzony (embedded subprocess)

Podproces osadzony — lub też zagnieżdżony — to czynność, która po prostu została zdekomponowana na mniejsze czynności. Bardzo często nawet w samej specyfikacji zamiast określenia „podproces osadzony” używa się po prostu określenia „podproces”. Właściwie wszystkie zasady, które obowiązują w przypadku modelowania zwykłych procesów, obowiązują też w przypadku podprocesów. A więc podproces może składać się ze zdarzeń, przepływów, czynności, bram i obiektów danych. Jedyną charakterystyczną cechą podprocesów osadzonych jest to, że mogą rozpocząć się tylko i wyłącznie zdarzeniami bez typu. Inaczej mówiąc, żaden podproces osadzony nie może mieć zdarzenia początkowego jakiegoś konkretnego typu.

W notacji BPMN 2.0 podproces osadzony jest reprezentowany przez piktogram czynności zwiniętej lub rozwiniętej (rysunek 9.17). W wersji zwiniętej piktogram czynności ma znak plusa przy dolnej krawędzi, w wersji rozwiniętej „wnętrze” czynności zawiera dokładny model podprocesu osadzonego.

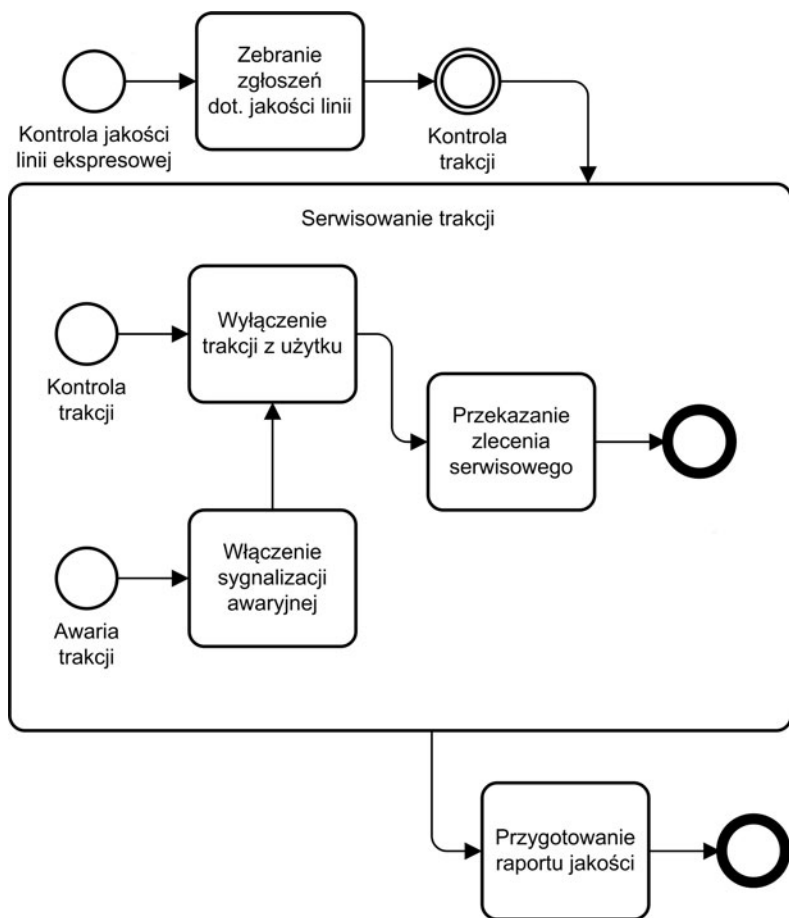


**Rysunek 9.17.** *Proces z podprocesem osadzonym zwiniętym (na górze) i rozwiniętym (na dole)*

Warto jednak zauważyć, że ten sam proces może być jednocześnie podprocesem osadzonym i procesem głównego poziomu. Jeśli bowiem zawiera on więcej niż jedno zdarzenie początkowe, to możemy go wywołać na różne sposoby: poprzez **czynność wywołania** — wtedy będzie to podproces osadzony, albo poprzez inne zdarzenie początkowe — wtedy będzie to proces głównego poziomu. Rysunek 9.18 przedstawia model procesu z podprocesem osadzonym serwisowania trakcji kolejowej, w którym występują dwa zdarzenia początkowe. Jeśli nastąpi awaria trakcji, proces ten zostanie wywołany jako proces głównego poziomu i nastąpi kolejno: wyłączenie sygnalizacji awaryjnej na całym odcinku, wyłączenie trakcji z użytku i — na koniec — przekazanie zlecenia serwisowego. Jeśli natomiast w trakcie realizacji procesu kontroli jakości linii nastąpi zdarzenie pośrednie *Kontrola trakcji*, to proces *Serwisowanie trakcji* zostanie wywołany jako podproces osadzony.

### 9.2.2. Podproces zdarzeniowy (event subprocess)

Podproces zdarzeniowy różni się od zwykłego podprocesu osadzonego tym, że musi rozpoczynać się zdarzeniem konkretnego typu. Podproces taki nie może mieć więcej niż jednego zdarzenia początkowego, jak w przypadku podprocesu osadzonego! Tym samym podproces zdarzeniowy jest specyficznym procesem, który „reaguje” wtedy i tylko wtedy, gdy nastąpi konkretne zdarzenie. Stąd też każdy podproces zdarzeniowy jest realizowany poza **normalnym przepływem** (patrz rozdział 5., „Przepływy”), gdyż nie wynika z sekwencji czynności w procesie nadrzędnym, ale z wystąpienia określonego zdarzenia — a to, jak wiadomo, może zależeć od czynników zewnętrznych, pozostających poza procesem nadrzędnym.

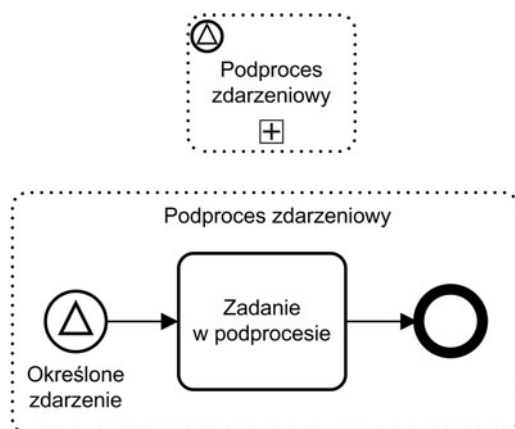


**Rysunek 9.18.** Proces kontroli jakości linii kolejowej

W notacji BPMN 2.0 podproces zdarzeniowy jest reprezentowany przez piktoqram czynności, której krawędź jest przerywana. Podobnie jak w przypadku podprocesu osadzonego, podproces zdarzeniowy może być zwinięty lub rozwinięty (rysunek 9.19).

W nowej wersji (rysunek 9.20) zdarzenie awarii jest „dystrybuowane” poprzez sygnał *Awaria trakcji*. Sygnał ten powoduje uruchomienie podprocesu zdarzeniowego *Awaria trakcji*, zatrzymanie procesu kontroli jakości linii i następuje jedynie włączenie sygnalizacji awaryjnej. Tak więc nowa wersja procesu jest mniej bezpieczna, bo awaria nie powoduje wyłączenia trakcji z użytku. Dodatkowo jeśli w trakcie kontroli jakości nastąpi awaria trakcji, to kontrola jakości zostanie przerwana. Z drugiej strony, jeśli nie nastąpi awaria, a proces kontroli jakości będzie przebiegał bez problemów, nigdy nie zostanie włączona sygnalizacja awaryjna.





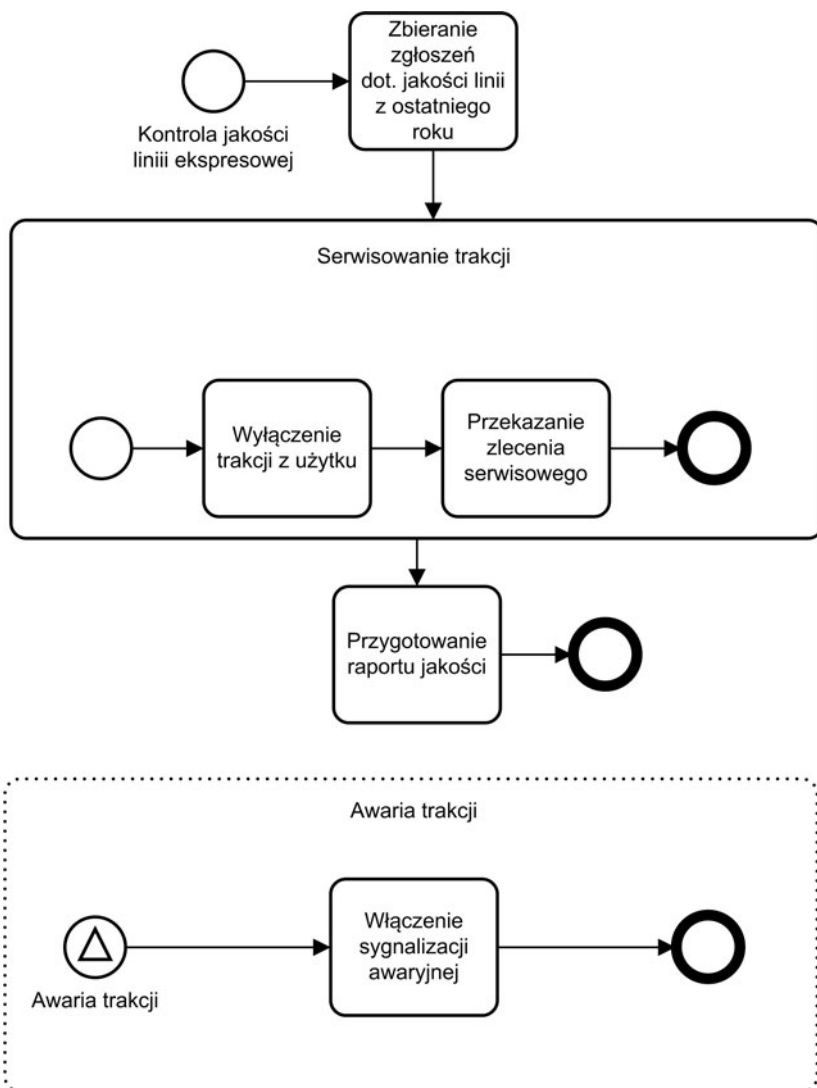
**Rysunek 9.19.** Podproces zdarzeniowy zwinięty (na górze) i rozwinięty (na dole)

O tym, że nastąpienie awarii trakcji spowoduje przerwanie kontroli jakości linii, decyduje przerywające zdarzenie początkowe podprocesu *Awaria trakcji*. Jest to szerzej wyjaśnione przy okazji omawiania zdarzeń początkowych podprocesów zdarzeniowych w rozdziale 8., „Zdarzenia”.

### 9.2.3. Podproces ad hoc (ad-hoc subprocess)

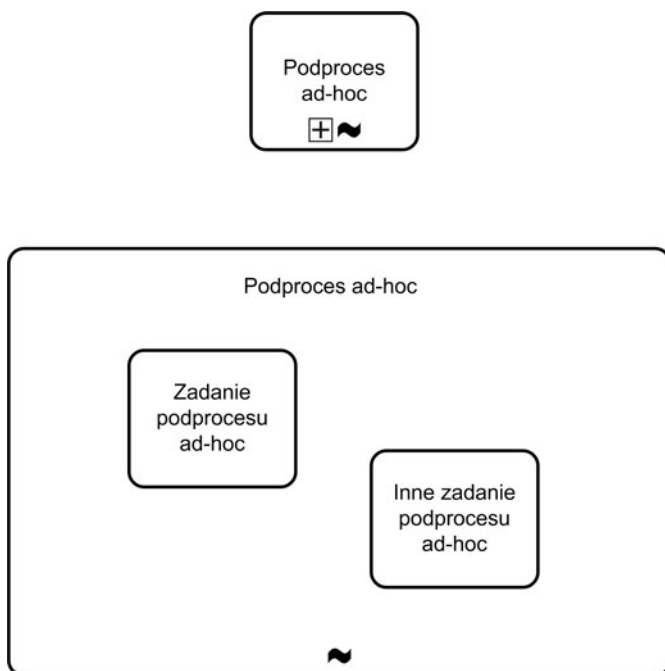
Podproces ad hoc to specjalny typ podprocesu, który jest po prostu grupą czynności realizowanych swobodnie, bez ściśle narzuconej kolejności. W pierwszej chwili może wydawać się to dziwne, ale jak się okazuje, bardzo wiele procesów w rzeczywistości może być realizowanych właśnie w ten sposób. W notacji BPMN 2.0 podprocesy ad hoc różnią się od zwykłych jedynie tym, że mają znacznik tyldy (~) (rysunek 9.21). Czynności w podprocesie ad hoc nie muszą być połączone w sekwencję, jak ma to miejsce w przypadku pozostałych rodzajów podprocesów. Mogą po prostu być swobodnie rozmieszczone, a ich położenie wewnątrz obramowania podprocesu nie ma wpływu na ich kolejność czy sposób realizacji.

Czytelnik może się zastanawiać, jak w takim razie taki podproces się rozpoczyna, jak przebiega i jak się kończy, skoro nie ma ustalonego porządku. Sposób rozpoczynania jest po prostu nieustalony i o tym, która czynność będzie realizowana, decyduje jej wykonawca. Przebieg zależy od dodatkowego parametru *ordering*, który może mieć jedną z dwóch wartości: *Parallel* lub *Sequential*. Wartość *Parallel* oznacza, że czynności w podprocesie są realizowane równolegle. Wartość *Sequential* oznacza, że czynności są realizowane pojedynczo, tzn. że choć nie są uporządkowane w sekwencję o określonej kolejności, w danej chwili realizowana jest tylko jedna z czynności. Warto jednak zauważyć, że każda z czynności w podprocesie ad hoc może być realizowana wielokrotnie.



**Rysunek 9.20.** Proces z podprocesem osadzonym i podprocesem zdarzeniowym

Natomiast zakończenie podprocesu zależy od parametrów `completionCondition` oraz `cancelRemainingInstances`. Parametr `completionCondition` zawiera wyrażenie warunkowe — jeśli jest ono prawdziwe, wtedy proces jest przerywany. Parametr `cancelRemainingInstances` może mieć wartość `true` lub `false`. Jeśli ma wartość `true` i jednocześnie podproces ad hoc jest w trybie `Parallel`, wtedy spełnienie warunku `completionCondition` jest równoważne z anulowaniem wszystkich czynności składowych danego podprocesu ad hoc, które były w trakcie realizacji.



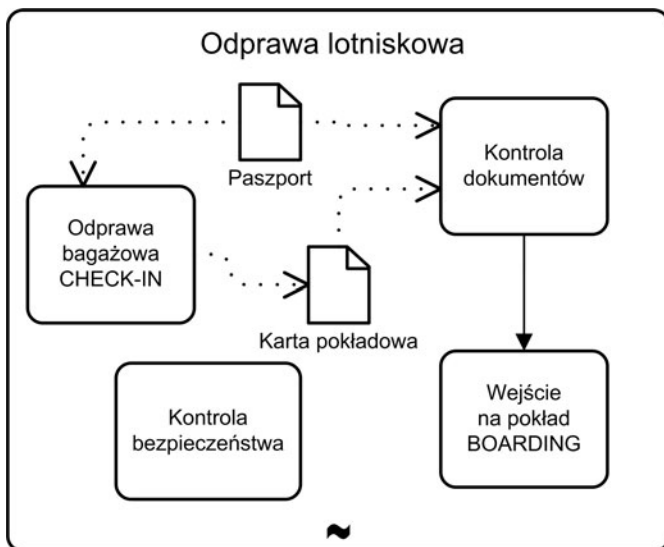
**Rysunek 9.21.** Podproces ad hoc — zwinięty i rozwinięty

Na rysunku 9.22 przedstawiony jest podproces ad hoc *Odprawa lotniskowa*, w którym „swobodnie” występują cztery zadania składowe: *Odprawa bagażowa CHECK-IN*, *Kontrola dokumentów*, *Kontrola bezpieczeństwa* i *Wejście na pokład BOARDING*.



**Rysunek 9.22.** Podproces ad hoc *Odprawa lotniskowa*

Czasami jednak może się okazać, że mamy do czynienia z podprocesem, w którym niektóre czynności składowe są swobodne, a inne połączone w sekwencje. W takim przypadku w podprocesie ad hoc możemy po prostu te czynności połączyć zwykłym przepływem sekwencji. Może się też okazać, że nie jesteśmy w stanie zdefiniować kolejności czynności, ale wiemy, jak przepływają między nimi obiekty danych — wtedy możemy stosować przepływy obiektów danych. Rysunek 9.23 przedstawia model podprocesu ad hoc, w którym niektóre czynności są połączone w sekwencję, a inne jedynie przepływami obiektów.



**Rysunek 9.23.** Podproces ad hoc z przepływami sekwencji i przepływami obiektów

#### 9.2.4. Podproces transakcji (transaction subprocess)

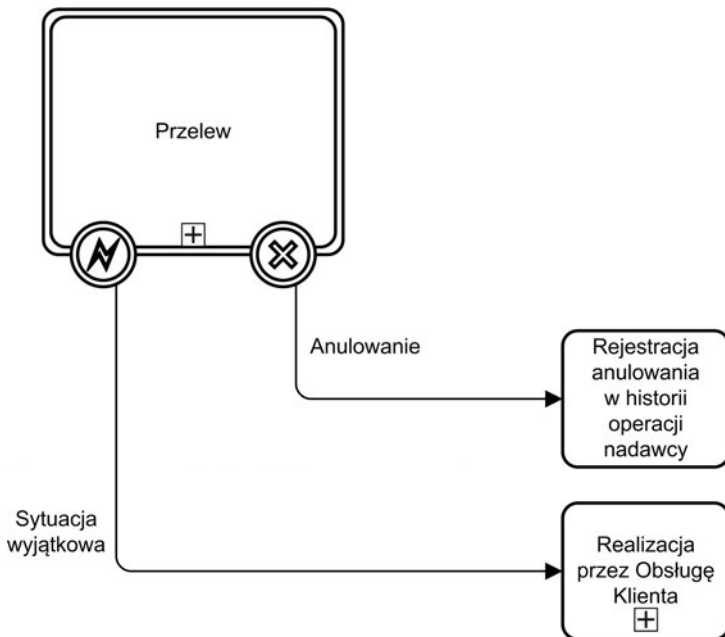
Podproces transakcji — lub potocznie po prostu „transakcja” — to podproces, którego przebieg jest kontrolowany przez tzw. protokół transakcyjny. Protokół taki gwarantuje, że realizacja danego podprocesu jest ściśle kontrolowana. Jeśli taki podproces zakończył się bez żadnego błędu lub nie zdarzyło się nic wyjątkowego, mamy gwarancję, że wszystkie czynności składowe, w tym inne podprocesy, również dobiegły końca w prawidłowy sposób. Jeśli natomiast podproces z jakiegoś powodu został anulowany, dla wszystkich czynności, które zostały już wywołane, uruchamiane są czynności kompensacji i następuje wycofanie (ang. *roll back*) transakcji.

Anulowanie transakcji może nastąpić tylko w dwóch przypadkach: gdy nastąpiło końcowe zdarzenie anulowania (które zresztą może być używane tylko i wyłącznie w transakcjach) lub gdy *komunikat* anulowania został odebrany poprzez protokół

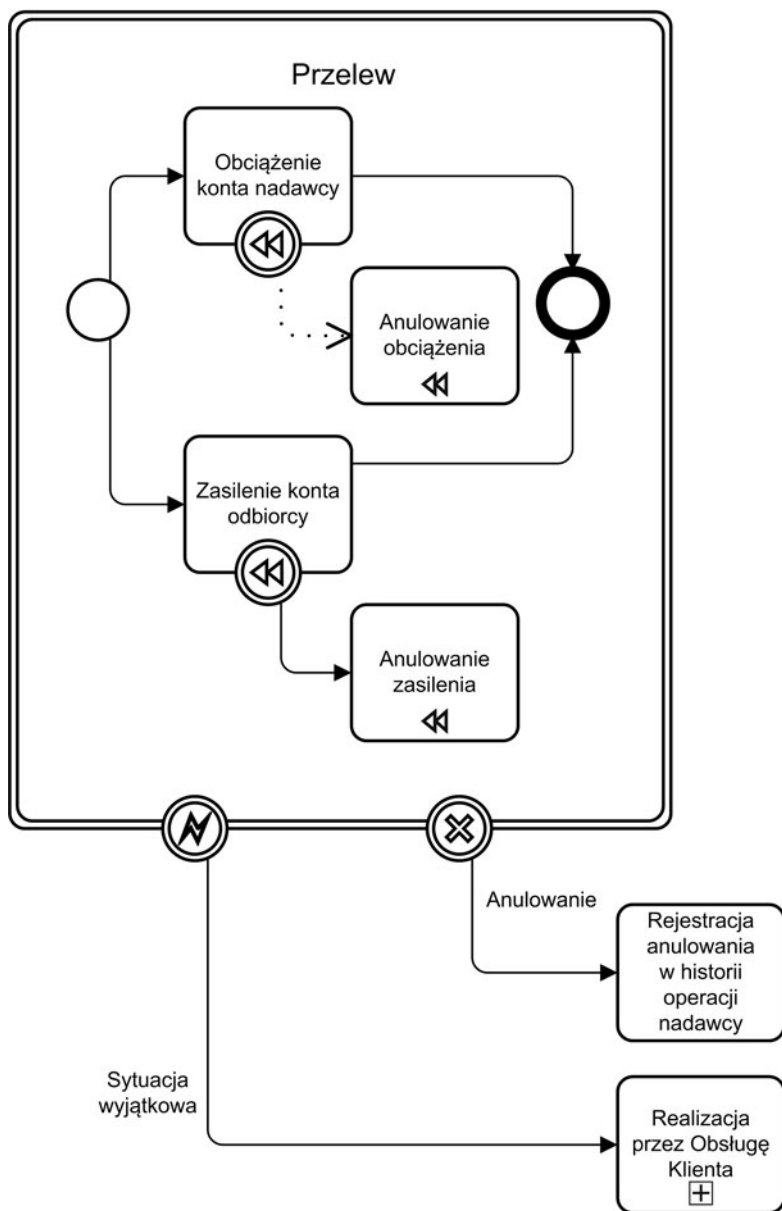
transakcyjny, w którym realizowana jest dana transakcja. O ile pierwszy wariant zależy od tego, czy w odpowiednim miejscu modelu umieścimy końcowe *zdarzenie* anulowania, o tyle *komunikat* anulowania może pojawić się zawsze. Możemy oczywiście opracować model, w którym zaprezentowane zostanie anulowanie modelowanej transakcji w oparciu o *komunikat* anulowania, ale jeśli tego nie zrobimy, a w trakcie uruchamiania podprocesu transakcji pojawi się taki komunikat, to transakcja i tak będzie anulowana.

Może się też zdarzyć, że podproces transakcji jest przerywany w wyniku błędu i nie ma możliwości przeprowadzenia kompensacji, tak jak przy anulowaniu transakcji. W takim przypadku przerywany jest cały podproces i dalszy przebieg jest kontynuowany od krawędziowego, pośredniego zdarzenia błędu. W ten sposób tworzymy podproces, który faktycznie funkcjonuje zgodnie z ogólnie przyjętym w informatyce schematem transakcji, tzn. nigdy nas nie zawiedzie, ponieważ niezależnie od tego, czy zdarzy się coś przewidywalnego (anulowanie), czy nieprzewidywalnego (błąd krytyczny), mamy możliwość „posprzątania bałaganu” w danych i utrzymania kontroli nad działaniem procesów czy systemów informatycznych.

W notacji BPMN 2.0 podproces transakcji jest reprezentowany podobnie jak podproces, z tym że jego krawędź jest zdublowana (rysunki 9.24 i 9.25).



**Rysunek 9.24.** Transakcja zwinięta



Rysunek 9.25. Transakcja rozwinięta

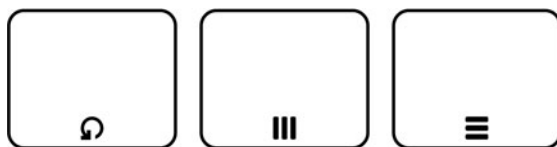
### 9.3. Pętle, czynności wieloinstancyjne i kompensacje

Procesy biznesowe z założenia są zbiorami wielokrotnie powtarzanych czynności, przy czym owo powtarzanie może następować nie tylko w wyniku ponownej realizacji całego procesu, ale także ze względu na konieczność wielokrotnego zrealizowania jakiejś czynności w obrębie jednej instancji danego procesu.

Rozważmy np. proces zaopatrywania sieci kilkudziesięciu hurtowni samochodowych w części od różnych producentów. Jest to proces biznesowy realizowany wielokrotnie w ciągu roku, składający się m.in. z takich czynności, jak zebranie stanów magazynowych z poszczególnych hurtowni czy składanie zamówień u konkretnych dystrybutorów na konkretne części w określonych ilościach. Innymi słowy, nie dość, że cały proces jest realizowany kilkanaście razy w roku, to jeszcze w ramach jego pojedynczej instancji obsługiwane są wszystkie hurtownie w sieci. Do każdej hurtowni dostarczane są części od różnych dostawców. Dodatkowo właściciel sieci hurtowni ma np. do dyspozycji tylko 8 samochodów dostawczych, każdy przyporządkowany do jednego obszaru Polski. Tak więc za każdym razem, gdy trzeba dostarczyć zamówione części do poszczególnych hurtowni, konieczne jest zrealizowanie podprocesu zaopatrzeniowego niezależnie dla każdego z ośmiu obszarów.

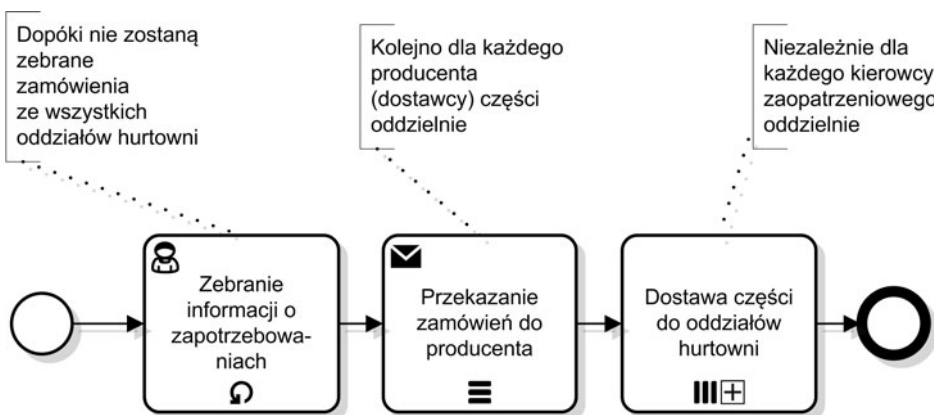
W przedstawionym procesie mamy więc do czynienia z wieloma czynnościami, które powtarzają się na różne sposoby. Czynność zbierania stanów magazynowych jest powtarzana dla każdej hurtowni. Można przypuszczać, że ktoś w centrali firmy zbiera stany magazynowe ze wszystkich hurtowni aż do momentu, gdy nastąpi określony czas, np. 20. dzień miesiąca. A więc mamy tutaj pewną czynność, która jest powtarzana, dopóki nie zostanie spełniony jakiś warunek, czyli mamy do czynienia z pętlą. Przekazywanie zamówień też jest realizowane wielokrotnie: oddzielnie dla każdego dystrybutora. Tutaj można przypuszczać, że zamówienia do poszczególnych dystrybutorów są przekazywane niezależnie, choć najprawdopodobniej realizuje to jedna osoba w firmie, która sprawdza zamówienie i akceptuje jego koszt, a następnie przechodzi do kolejnego zamówienia, sprawdza je i akceptuje koszt itd. A więc mamy tę samą czynność powtarzaną kilka razy. W standardzie BPMN 2.0 jest to **wieloinstancyjna czynność sekwencyjna**. I wreszcie mamy podproces zaopatrzeniowy, który jest powtarzany niezależnie dla każdego obszaru geograficznego, czyli wszystkie 8 obszarów kraju jest obsługiwanych równolegle. W standardzie BPMN 2.0 jest to **wieloinstancyjna czynność równoległa**.

W notacji BPMN 2.0 w celu oznaczenia, że dana czynność jest powtarzana, wprowadzono tzw. **znaczniki**. Są to małe ikony umieszczone wewnątrz czynności, wyśrodkowane przy dolnej krawędzi (rysunek 9.26).



**Rysunek 9.26.** Zadania ze znacznikami (od lewej): pętli, wieloinstancyjności równoległej i sekwencyjnej

Rysunek 9.27 przedstawia opisany wcześniej model procesu zaopatrywania sieci hurtowni. Jak widać, znaczniki mogą występować niezależnie od ikon typów zadań; mogą także występować w przypadku podprocesów. Zarówno dla zadań, jak i dla podprocesów ich znaczenie jest takie samo.



**Rysunek 9.27.** Proces zaopatrzeniowy hurtowni z częściami samochodowymi

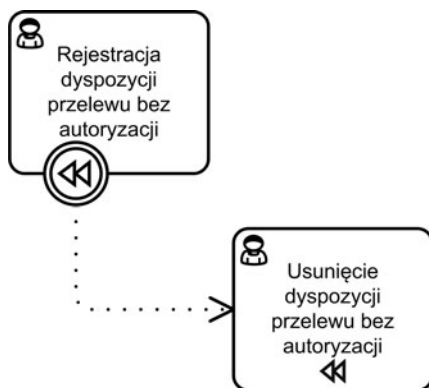
### 9.3.1. Czynność kompensacji (compensation activity)

Czynność kompensacji to czynność w procesie biznesowym, której celem jest wycofanie wcześniej wykonanego działania. Jeśli np. klient banku chce, aby bank każdego piątego dnia miesiąca przelewał określoną kwotę z rachunku podstawowego na rachunek oszczędnościowy, to konieczne jest: a) utworzenie zlecenia stałego w systemie transakcyjnym oraz b) wydanie dyspozycji upoważniającej bank do automatycznego przelewania pieniędzy między rachunkami bez każdorazowego autoryzowania tej operacji przez klienta banku. Jeśli jednak w trakcie procesu konfiguracji zlecenia stałego okaże się, że ostatecznie klient nie potwierdza chęci aktywacji takiej usługi, konieczne jest wycofanie zlecenia stałego i dyspozycji przelewu bez autoryzacji. Oczywiście można zaplanować ten proces tak, aby realizacja pewnych działań następowała dopiero, gdy bank ma pewność, że może utworzyć zlecenie stałe i zarejestrować dyspozycję przelewu bez autoryzacji. Wtedy nie byłoby problemu z kompensowaniem niepotrzebnych czynności. Jednak w prak-



tyce nie zawsze jest możliwe dowolne operowanie procesem i łatwiej jest zagwarantować, że ewentualne niepotrzebne czynności zostaną wycofane poprzez dodatkowe czynności kompensacji, niż że proces zawsze będzie realizowany tak, iż nie wystąpią problemy.

W notacji BPMN 2.0 czynność kompensacji jest reprezentowana przez piktoqram czynności ze znacznikiem kompensacji (rysunek 9.28). Ważniejsze jednak jest to, że taka czynność zawsze musi być powiązana asocjacją ze zdarzeniem krawędziowym kompensacji (rysunek 9.28). W rozdziale 8. Czytelnik znajdzie więcej informacji o zdarzeniach krawędziowych oraz przykładowy model procesu z kompensacjami.



**Rysunek 9.28.** Zadanie użytkownika i zadanie kompensacji

Wycofywanie czynności nie musi odbywać się poprzez pojedyncze czynności czy zadania, ale może być realizowane przez rozbudowane podprocesy — wtedy mamy do czynienia z podprocesami kompensacji.

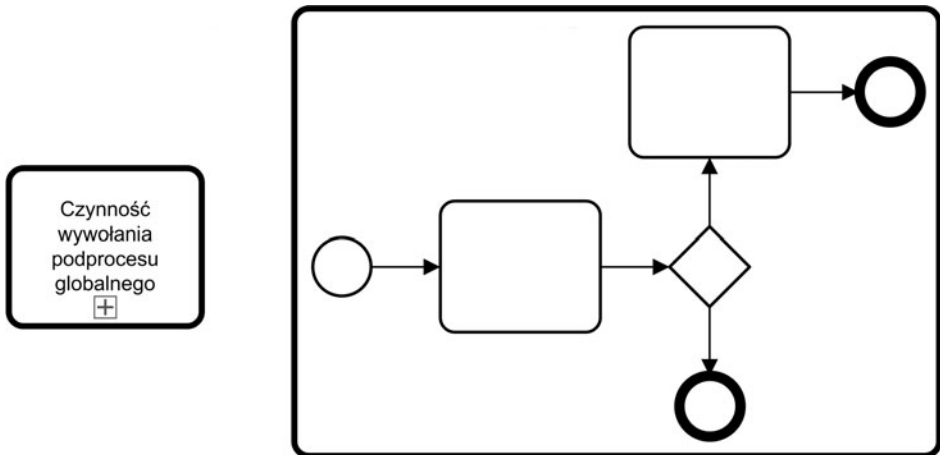
## 9.4. Czynność wywołania (call activity)

Czynność wywołania to czynność, która występuje w modelu w charakterze łączka do innego procesu globalnego lub zadania globalnego. Wystąpienie czynności wywołania oznacza, że wykonywany jest proces globalny lub zadanie globalne, na które wskazuje dana czynność wywołania. W notacji BPMN 2.0 czynność wywołania jest reprezentowana identycznie jak czynność, do której się odwołuje, z tą różnicą, że jej krawędź jest pogrubiona (rysunek 9.29).

Ponieważ, jak już wcześniej wspomniano, czynność wywołania może odwoływać się do całego procesu, możliwe są wywołania podprocesów zwiniętych i rozwiniętych (rysunek 9.30). W przypadku podprocesu rozwiniętego pogrubiona krawędź oznacza, że jest to proces globalny, może więc być ponownie wywoływany w innym miejscu.



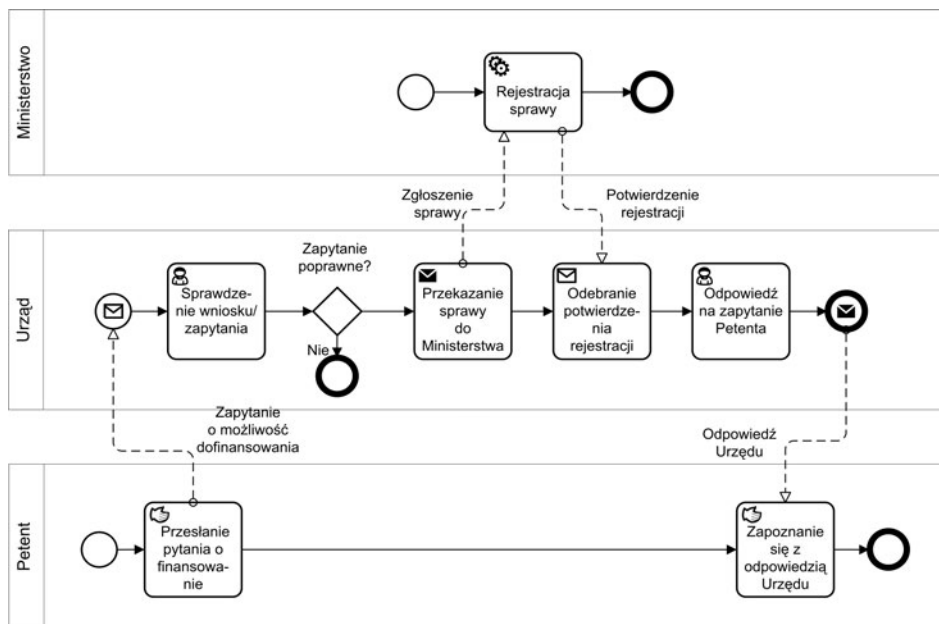
**Rysunek 9.29.** Czynność wywołania zadania globalnego (tutaj: zadania użytkownika)



**Rysunek 9.30.** Czynność wywołania podprocesu globalnego (po lewej: zwiniętego, po prawej: rozwiniętego)

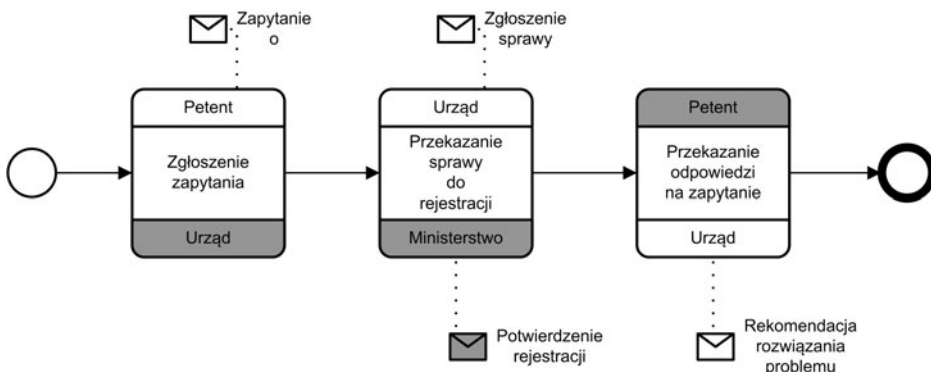
Na początku tego rozdziału omawialiśmy proces tworzenia dyspozycji przelewu jednorazowego. Jeśli zawsze będziemy korzystali z takiej samej formy autoryzacji, niezależnie od tego, czy chcemy zrealizować przelew jednorazowy, otworzyć lokatę, czy utworzyć zlecenie stałe, zawsze posłużymy się odwołaniem do czynności nieatomowej *Autoryzacja operacji*. Wtedy właśnie będziemy mogli skorzystać z czynności wywołania, a ów proces autoryzacji będzie procesem globalnym.

Standardowy proces BPMN opisuje czynności, jakie wykonują uczestnicy procesu biznesowego (rysunek 10.1). Choreografia natomiast opisuje tylko interakcje zachodzące między uczestnikami (rysunek 10.2). W przypadku modelu procesu biznesowego — jeśli chcemy — możemy opisać działania dotyczące jednego, wybranego uczestnika procesu. W choreografii musi być ich zawsze co najmniej dwóch, nie możemy stworzyć choreografii dla jednego uczestnika.



Rysunek 10.1. Model procesu biznesowego

Elementy, które mogą występować w choreografiach, to: zadania choreografii, choreografie złożone, wywołania choreografii, wywołania globalnych zadań choreografii, bramki, zdarzenia i przepływy. W tym rozdziale szczegółowo wyjaśnione są jedynie zadania choreografii oraz choreografie złożone.



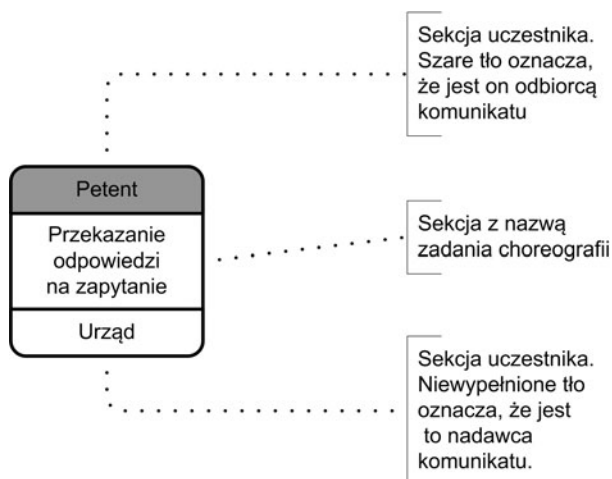
**Rysunek 10.2.** *Choreografia na podstawie procesu powyżej*

Uwaga: choreografie w standardzie BPMN 2.0 wykorzystują różne pojęcia zdefiniowane na potrzeby zwykłych procesów, kolaboracji i konwersacji. W związku z tym do zrozumienia niniejszego rozdziału niezbędne jest posiadanie kompletnej wiedzy na temat modelowania procesów w BPMN 2.0. Jeśli jakiś element w choreografii ma znaczenie odmienne od wynikającego z wcześniejszych rozdziałów, jest to wyraźnie zaznaczone. Jeśli natomiast jakieś pojęcie (np. „wieloinstancyjny uczestnik procesu”) jest użyte bez wyjaśnienia, oznacza to, że jego znaczenie jest takie samo jak podane wcześniej w odpowiednim rozdziale.

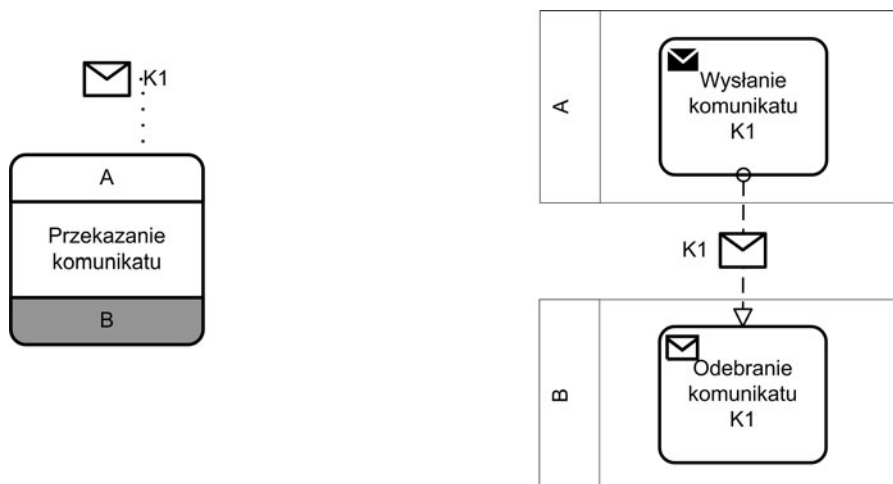
## 10.1. Zadanie choreografii

Każde zadanie choreografii jest reprezentowane przez prostokąt z zaokrąglonymi rogami podzielony na co najmniej trzy sekcje (rysunek 10.3) i opisuje ono wymianę komunikatów pomiędzy uczestnikami procesu. Jedna, centralna sekcja zawiera nazwę zadania, pozostałe zawierają nazwy uczestników, między którymi następuje wymiana komunikatów. Przy czym sekcja każdego uczestnika, który odbiera komunikat, musi mieć wypełnione tło. Nie ma natomiast znaczenia, czy na górze jest nadawca, a na dole odbiorca, czy odwrotnie.

Zadanie choreografii może występować w dwóch postaciach: jedno- i dwukierunkowej. Jednokierunkowe zadanie choreografii odpowiada fragmentowi procesu, w którym jeden z uczestników wysła komunikat (rysunek 10.4). Dwukierunkowe zadanie choreografii odpowiada fragmentowi procesu, w którym poza wysłaniem następuje odpowiedź zwrotna (rysunek 10.5). Różnica w notacji między tymi rodzajami zadań choreografii polega na umieszczeniu dwóch (zadanie dwukierunkowe) lub jednego (zadanie jednokierunkowe) komunikatu przy sekcjach uczestników. Należy pamiętać, że w przypadku zadania dwukierunkowego komunikat zwrotny musi także mieć wypełnione tło, podobnie jak sekcja uczestnika odbierającego ów komunikat.



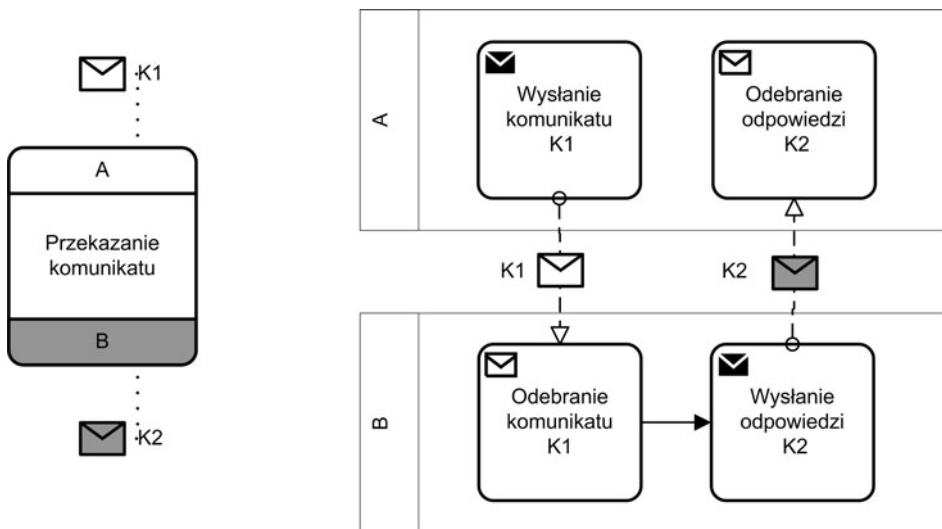
**Rysunek 10.3.** Zadanie choreografii



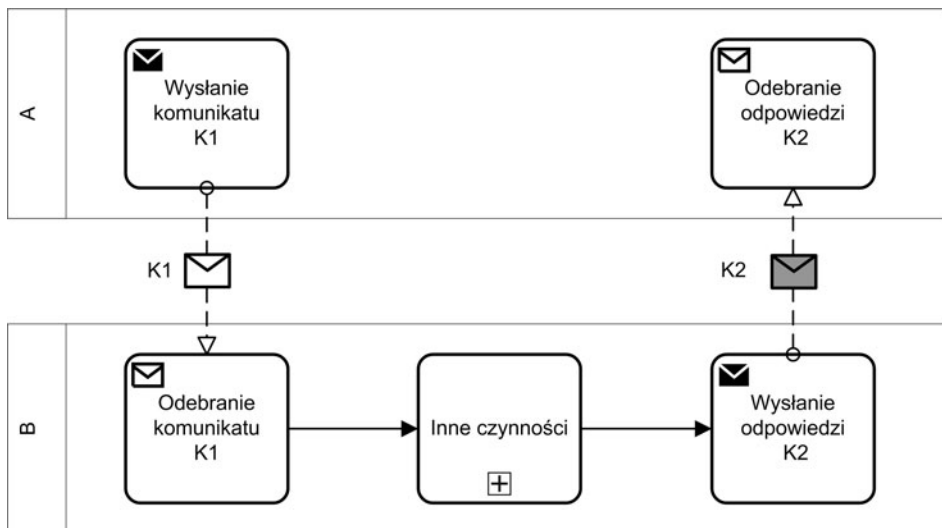
**Rysunek 10.4.** Jednokierunkowe zadanie choreografii i odpowiadający mu fragment procesu

Warto jednak zauważyć, że dwukierunkowemu zadaniu choreografii mogą odpowiadać różne modele procesów. Nie musi być tak, że odpowiedź następuje natychmiast po odebraniu komunikatu (rysunek 10.6). Nie musi też być tak, że wysłanie odpowiedzi jest odrębną czynnością (rysunek 10.7), może to być bowiem zadanie usługowe, które pobiera dane wejściowe i odpowiada pewnym wynikiem.

Zadanie choreografii nie musi być ograniczone do dwóch uczestników. Zawsze jednak musi być jeden nadawca. A więc poprawnym zadaniem choreografii będzie także takie, przedstawione na rysunku 10.8.

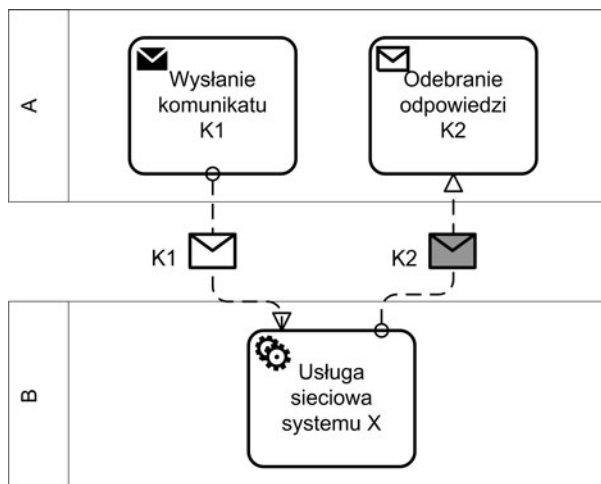


**Rysunek 10.5.** Dwukierunkowe zadanie choreografii i przykładowy odpowiadający mu fragment procesu

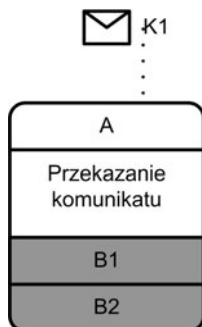


**Rysunek 10.6.** Inny przykładowy odpowiednik dwukierunkowego zadania choreografii

Podobnie jak w przypadku zwykłych czynności w procesach, zadania choreografii mogą mieć charakter powtarzalny. W takiej sytuacji należy zastosować odpowiedni znacznik pętli, wieloinstancyjności równoległej lub wieloinstancyjności sekwencyjnej. Interpretacja tych znaczników jest identyczna jak w przypadku czynności. Zadania choreografii ze znacznikami oraz odpowiadające im fragmenty procesu przedstawiono na rysunkach 10.9, 10.10 i 10.11.



**Rysunek 10.7.** Odpowiednik dwukierunkowego zadania choreografii z zadaniem usługowym



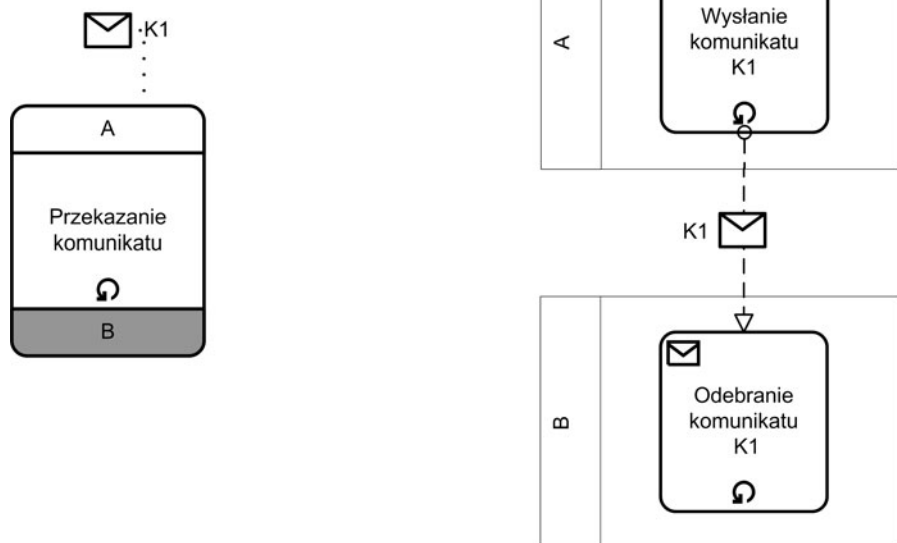
**Rysunek 10.8.** Zadanie choreografii z wieloma odbiorcami

Jeśli uczestnik choreografii jest uczestnikiem wieloinstancyjnym, jego sekcja musi być także opatrzona znacznikiem wieloinstancyjnego uczestnika (rysunek 10.12).

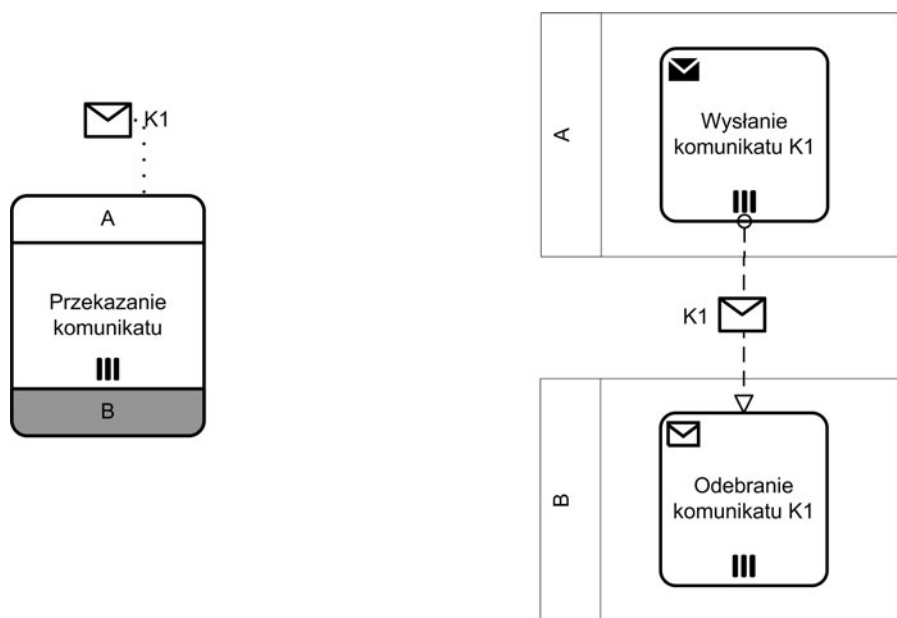
Przedstawione powyżej znaczniki (pętle, wieloinstancyjności równoległej i sekwencyjnej) mogą także być stosowane w przypadku choreografii złożonych, które są omówione w dalszej części rozdziału. W obu przypadkach znaczniki te mają taką samą interpretację.

## 10.2. Choreografia złożona

Choreografia złożona to zadanie choreografii, które składa się z innych zadań choreografii lub jest wielokrotną wymianą komunikatów między uczestnikami. Zwykle zadanie choreografii oznacza wysłanie komunikatu i opcjonalnie odebranie odpowiedzi. Zdarza się jednak, że następuje wymiana większej liczby komunikatów

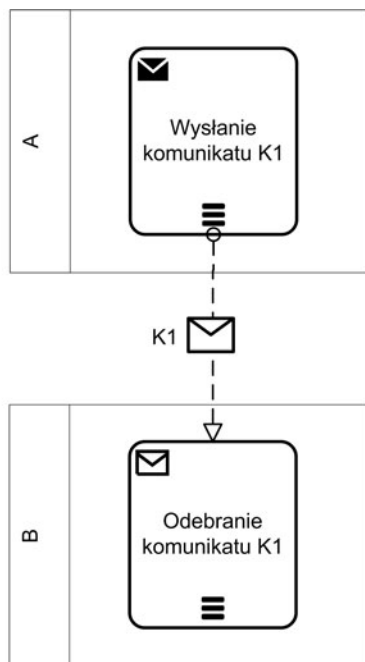


**Rysunek 10.9.** Zadanie choreografii w pętli i odpowiadający mu fragment procesu

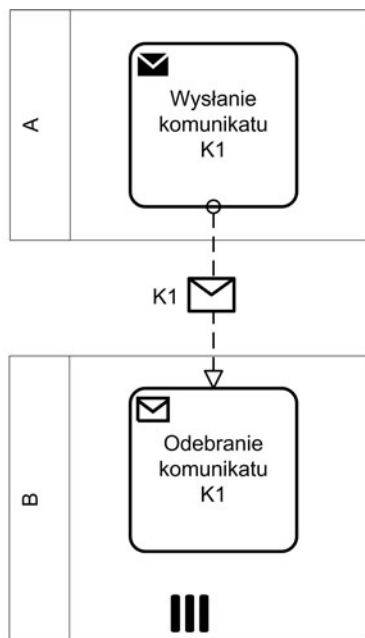


**Rysunek 10.10.** Wieloinstancyjne równoległe zadanie choreografii i odpowiadający mu fragment procesu



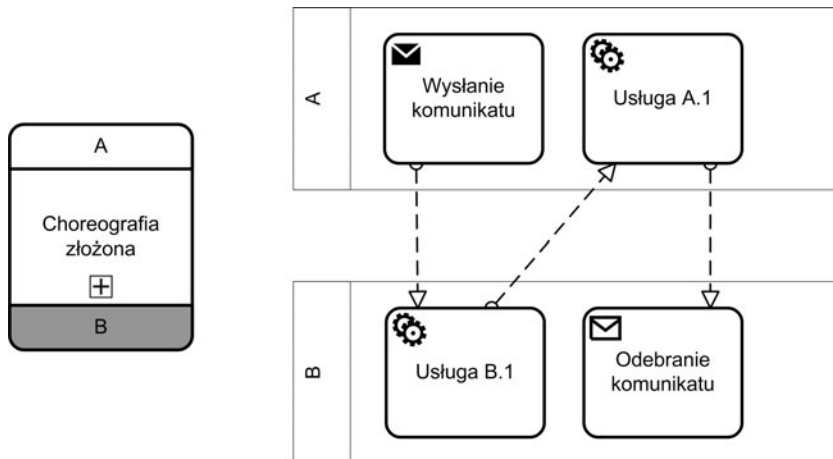


**Rysunek 10.11.** Wieloinstancyjne sekwencyjne zadanie choreografii i odpowiadający mu fragment procesu

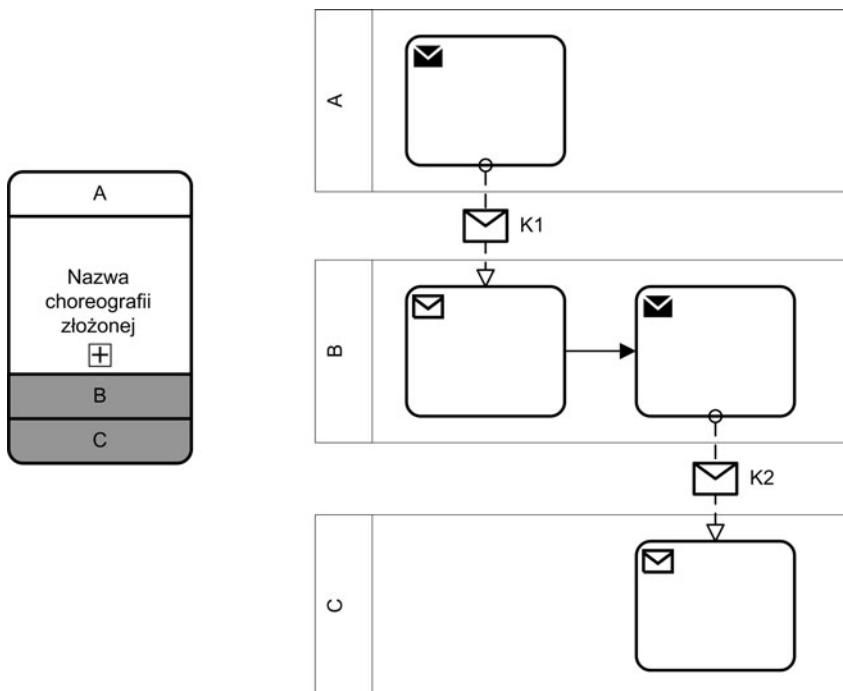


**Rysunek 10.12.** Zadanie choreografii — uczestnik B jest wieloinstancyjny

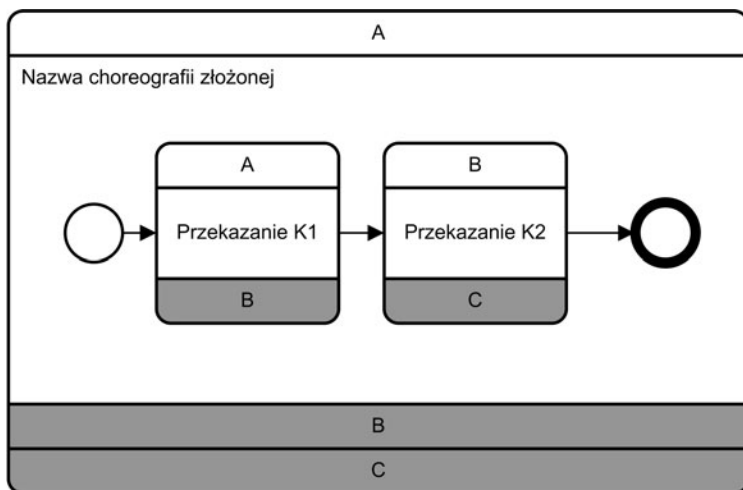
(rysunek 10.13) lub w wymianę komunikatów jest zaangażowanych więcej niż tylko dwóch uczestników (rysunek 10.14 – wersja zwinięta, rysunek 10.15 – wersja rozwinięta choreografii złożonej).



**Rysunek 10.13.** Choreografia złożona i przykładowy odpowiadający jej fragment procesu



**Rysunek 10.14.** Choreografia złożona i inny niż powyżej odpowiadający jej przykładowy fragment procesu



Rysunek 10.15. Choreografia złożona z wcześniejszego przykładu w formie rozwiniętej

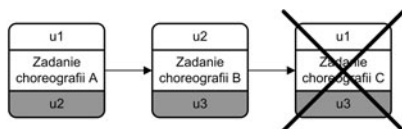
### 10.3. Kolaboracje, procesy i choreografie

W przypadku choreografii obowiązuje jedna, kluczowa zasada: nadawca w zadaniu choreografii musi być zaangażowany we wcześniejszym zadaniu choreografii w roli nadawcy lub odbiorcy. W związku z tym choreografia przedstawiona na rysunku 10.16 jest niepoprawna. Jednoznaczne przełożenie jej na odpowiedni model procesu biznesowego jest niemożliwe.

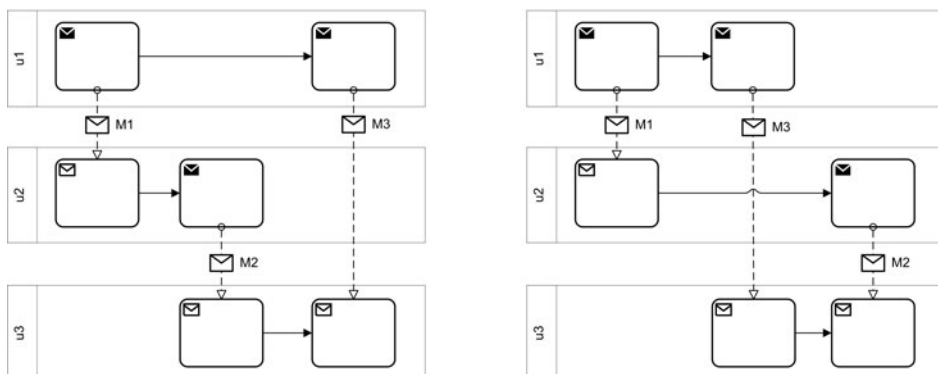
### 10.4. Kiedy używać diagramów choreografii?

W pierwszej chwili choreografie mogą wydawać się dziwne i skomplikowane. Zresztą od chwili wprowadzenia przez OMG nadal nie są szczególnie popularne wśród analityków biznesowych i systemowych. Niemniej są bardzo ciekawym narzędziem do modelowania tego, co w większości notacji jest pominięte, czyli wzorców wymiany komunikatów (ang. *Message Exchange Patterns*, w skrócie: MEPs). Podobnie jak w przypadku modelowania w UML-u możemy ukrywać „wnętrza” komponentów i koncentrować się jedynie na uzgodnieniu interfejsów, które owe komponenty wykorzystują, tak samo tu możemy ukrywać szczegóły procesu biznesowego i eksponować komunikaty wymieniane między jego uczestnikami.

Uważny Czytelnik może stwierdzić, że przecież to samo możemy robić, używając kolaboracji i konwersacji! Częściowo tak, jednak w przypadku choreografii standard BPMN 2.0 narzuca kilka reguł, które wymuszają tworzenie jednoznacznych modeli bez rozpisywania ich szczegółów. Dzięki temu możliwe jest ustalenie, jaki



?



**Rysunek 10.16.** Błąd w choreografii i jego konsekwencja

proces odpowiada danej choreografii. Oczywiście nadal może się okazać, że mamy wiele możliwych rozwiązań, ale muszą one być zgodne z określonym wzorcem MEP. W przypadku kolaboracji i konwersacji mamy po prostu do czynienia z pomijaniem szczegółów.

---

## Inne źródła

---

1. [www.omg.org](http://www.omg.org) — witryna organizacji Object Modeling Group, która jest twórcą m.in. standardu BPMN 2.0. Stamtąd można pobrać najnowszą i aktualną wersję specyfikacji standardu.
2. [www.bpmnstyle.com](http://www.bpmnstyle.com) — witryna książki *BPMN. Method and Style* Bruce'a Silvera, guru standardu BPMN.
3. Thomas Allwayer, *BPMN 2.0* — dobra książka do BPMN, niestety napisana na podstawie wersji beta standardu BPMN 2.0, dlatego zawiera kilka niezgodności z jego końcową (oficjalną) wersją.
4. [www.bpminstitute.org](http://www.bpminstitute.org) — portal ekspercki zawierający szereg materiałów na temat zarządzania procesami biznesowymi.
5. [www.businessprocessincubator.com](http://www.businessprocessincubator.com) — portal społecznościowy umożliwiający pobieranie przykładowych modeli, prowadzenie dyskusji i opiniowanie innych modeli, a także pobieranie narzędzi do modelowania.