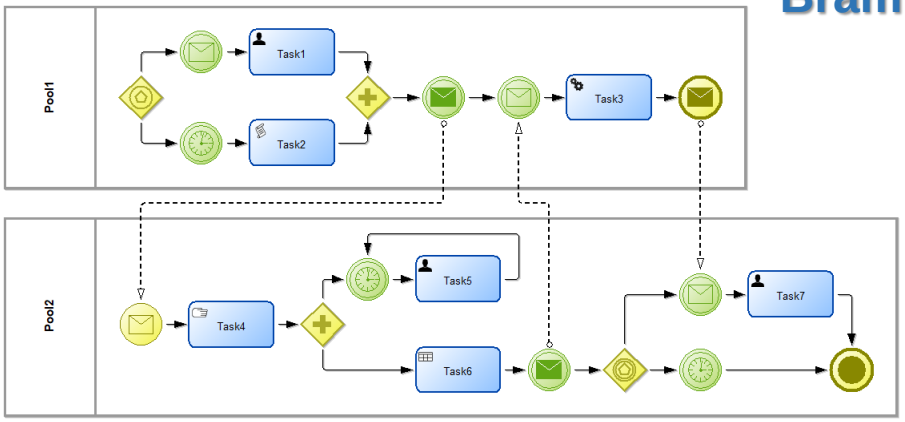
Modelowanie i symulacja procesów biznesowych

Typy aktywności

Zdarzenia początkowe, końcowe i pośrednie

Bramki oparte na danych i zdarzeniach



Aktywności

Aktywności (*activities*) są wykonywalnymi elementami procesu i reprezentują punkty, w których wykonywana jest pewna praca. BPMN 2.0 definiuje trzy typy aktywności:

Zadanie (*task*) jest aktywnością atomową (niepodzielną), której nie można przedstawić jako ciągu elementów przepływu przy założonym stopniu szczegółowości modelu. Określa elementarną czynność realizowaną przez użytkownika lub aplikację podczas realizacji procesu. Sposób wykonania zadania określa jego typ.

Podproces (*sub-proces*) jest szczególnym przypadkiem aktywności, która zawiera szczegółowo opisany sposób realizacji. Jak proces główny ma cechy kontenera obiektów, więc może zawierać elementy przepływu. W modelu podprocesy mogą ukrywać swoją wewnętrzną strukturę (zwinięte) – w takim przypadku są reprezentowane przez pojedynczy symbol graficzny, lub pokazywać strukturę wewnętrzną (rozwinięte) – w takim przypadku elementy podprocesu są otoczone ramką, która oddziela je od procesu głównego lub innych podprocesów.

Typy zadań w BPMN 2.0

Abstrakcyjne* (*abstract task*) – zadanie abstrakcyjne bez określonego typu.

Wysyłka (*send task*) – wysyła komunikat do innego uczestnika, odbiorca może być identyfikowany przez przepływ komunikatu.

Odbiór (*receive task*) – oczekuje i odbiera komunikat od innego uczestnika, proces jest kontynuowany po odbiorze komunikatu, nadawca może być identyfikowany przez przepływu komunikatu.

Użytkownik* (*user task*) – operacja wykonywana przez człowieka przy wsparciu oprogramowania zgodnie z harmonogramem, przebieg zadania nadzorowany przez składnik silnika procesów nazywany menadżerem zadań (*task manager*).

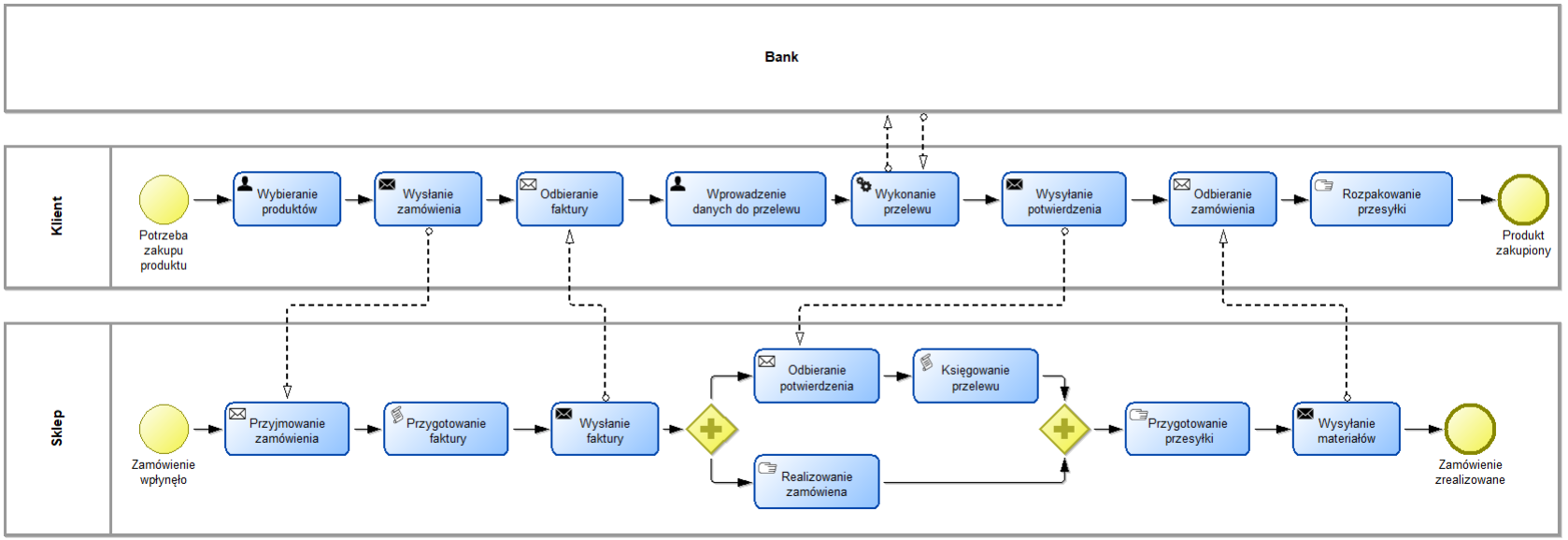
Manualne (*manual task*) – operacja niezautomatyzowana wykonywana przez człowieka bez wsparcia/nadzoru oprogramowania.

Skrypt (*script task*) – operacja automatyczna, bez udziału człowieka, wykonywana przez skrypt realizowany przez silnik procesów biznesowych.

Reguła biznesowa (*business rule*) – operacja automatyczna, lub pół automatyczna wykonywana przez silnik reguł biznesowych.

Serwis* (*service task*) – operacja automatyczna, bez udziału człowieka, zadanie realizowane przez zewnętrzną usługę (np. serwis WWW), dostawca usługi może być identyfikowany przez przepływ komunikatu.

Typy zadań – przykład



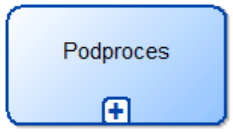
1. *Klient* przygotowuje zamówienie (*user*), wysyła (*send*) i oczekuje (*receive*) na fakturę.
2. Odbiór zamówienia (*receive*) uruchamia proces *Sklepu*, faktura zostaje automatyczne wygenerowana (*script*) i przesyłana (*send*) do *Klienta*.
3. *Sklep* realizuje zamówienie (*manual*) i oczekuje na potwierdzenie przelewu (*recieve*).
4. *Klient* przygotowuje (*user*) i wysyła przelew korzystając z serwisu banku (*service*), wysyła potwierdzenie (*send*) i oczekuje zamówione produkty (*recieve*).
5. Po automatycznym zaksięgowaniu wpłaty (*script*) pracownicy *Sklepu* przygotowują przesyłkę (*manual*) wysyłają zamówione produkty (*send*) do *Klienta*.

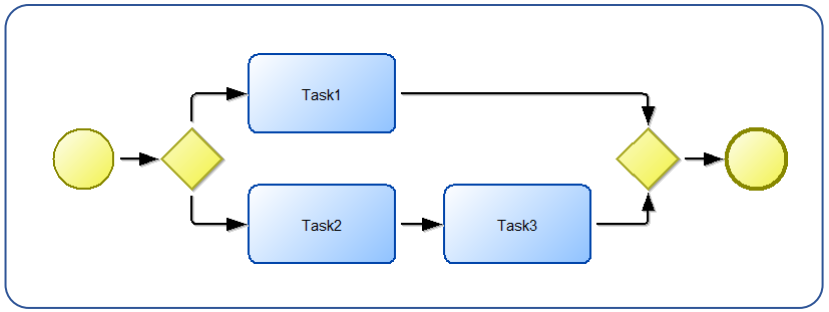
Podprocesy osadzone

Podproces osadzony (krótko: podproces) występuje w sekwencji przepływu procesu nadrzędnego. Nie ma własnej instancji, jest uruchamiany przez proces nadrzędny, gdy token dociera do symbolu podprocesu. Podproces osadzony pozwala na zdefiniowanie złożonej aktywności, która może być wykorzystana w modelu kilkukrotnie, dodatkowo jej elementy składowe mogą być pokazane lub schowane zależnie od pożądanego poziomu szczegółowości.

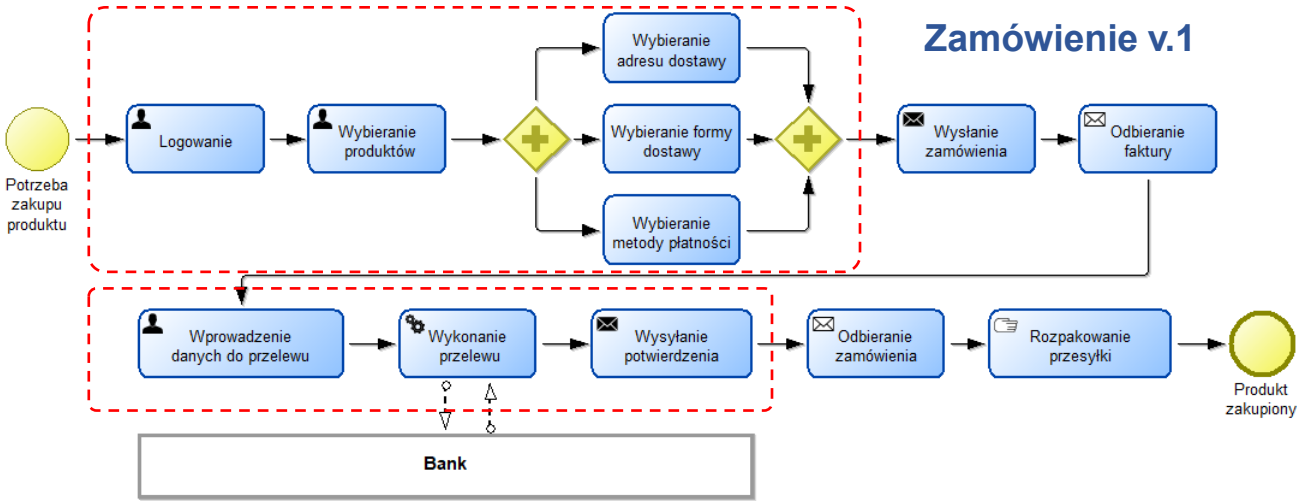
Cechy podprocesów osadzonych

- W wersji zwiniętej: symbol aktywności (*activity*) z dodatkowym znakiem "plus" w kwadratowej ramce, umieszczonym przy dolnej krawędzi;
- W wersji rozwiniętej: pojedyncza ramka bez znaku "plus";
- Rozpoczyna się od pojedynczego zdarzenia początkowego bez wyzwalacza;
- Dozwolony podproces bez zdarzenia początkowego – w takim przypadku wszystkie sekwencje podprocesu otrzymują token równocześnie i są wykonywane równolegle.

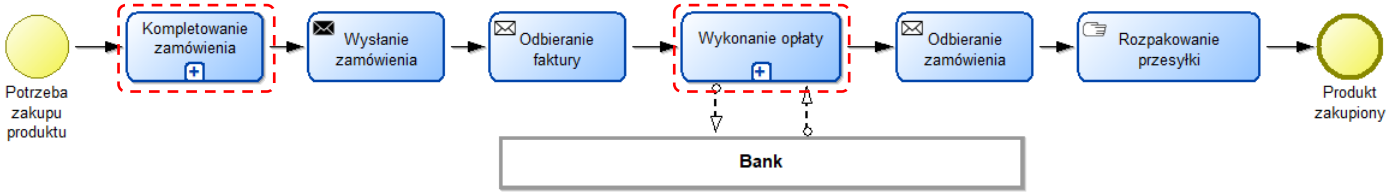




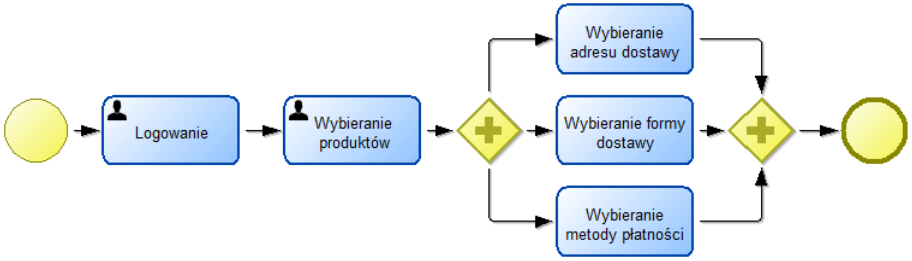
Podprocesy osadzone – przykład 1.



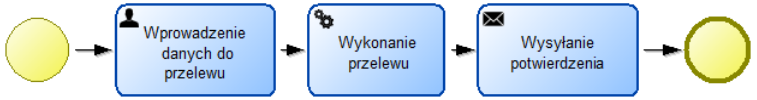
Zamówienie v.2 – podprocesy osadzone



Podproces: Kompletowanie zamówienia



Podproces: Wykonanie opłaty



Bramki (*gateways*)

Bramki – podstawowe elementy określające logikę przepływu sekwencyjnego w procesie. Definiują zachowanie procesu w punktach, w których przepływ rozgałęzia się i łączy. Symbolem bramki jest diament z opcjonalnym symbolem graficznym wewnątrz.

Cechy bramek

- Symbolizują rozgałęzienia przepływu, ich wyjścia mogą być opisane przez wyrażenia warunkowe;
- Nie reprezentują czynności realizowanych w procesie, ich "wykonanie" nie wpływa koszty realizacji procesu (czas, zasoby, itp.);
- Mogą mieć dowolną liczbę przepływów wejściowych i wyjściowych, jedna z tych liczb musi być większa od jeden.

Typy bramek

Bramka wykluczająca (*exclusive OR*, XOR)

Bramka równoległa (*parallel*, AND)

Bramka niewykluczająca (*inclusive*, OR) – nieomawiana

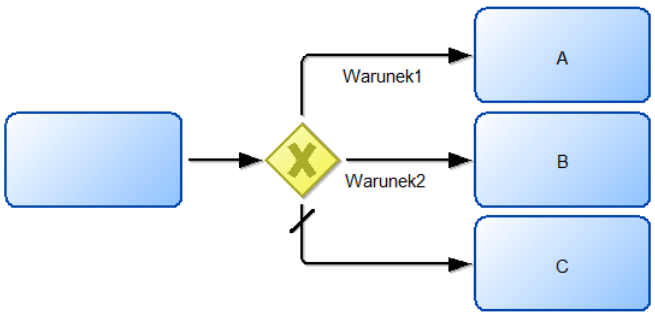
Bramka złożona (*complex*) – nieomawiana

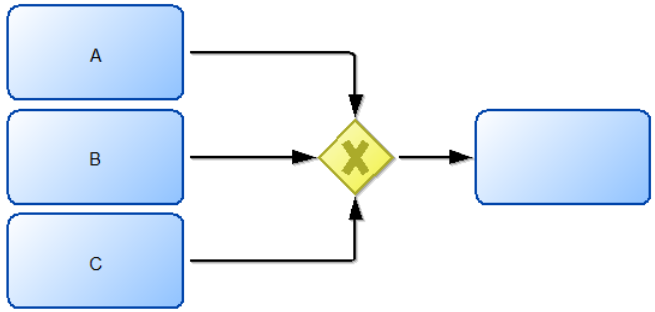
Bramka wykluczająca (*exclusive OR*, XOR)

Rozdzielająca bramka wykluczająca jest domyślnym typem bramki BPMN, tworzy kilka alternatywnych ścieżek przepływu, tylko jedna może być wybrana. Bramka oparta na danych, z każdym przepływem związany jest warunek wykorzystujący dane dostępne w ramach procesu. Warunki powinny mieć charakter wykluczający. Token procesu wybiera ścieżkę, której warunek będzie spełniony jako pierwszy. Dopuszczalny przepływ domyślny wybierany gdy żaden z warunków nie jest spełniony.

Łącząca bramka wykluczająca łączy kilka przepływów sekwencyjnych. Każdy token z wejścia bramki jest natychmiast kierowany na jej wyjście bez synchronizacji z potencjalnymi tokenami pochodzącymi z innych wejść.

BPMN v.2.0.2, 10.6.2.Exclusive Gateway, s.289

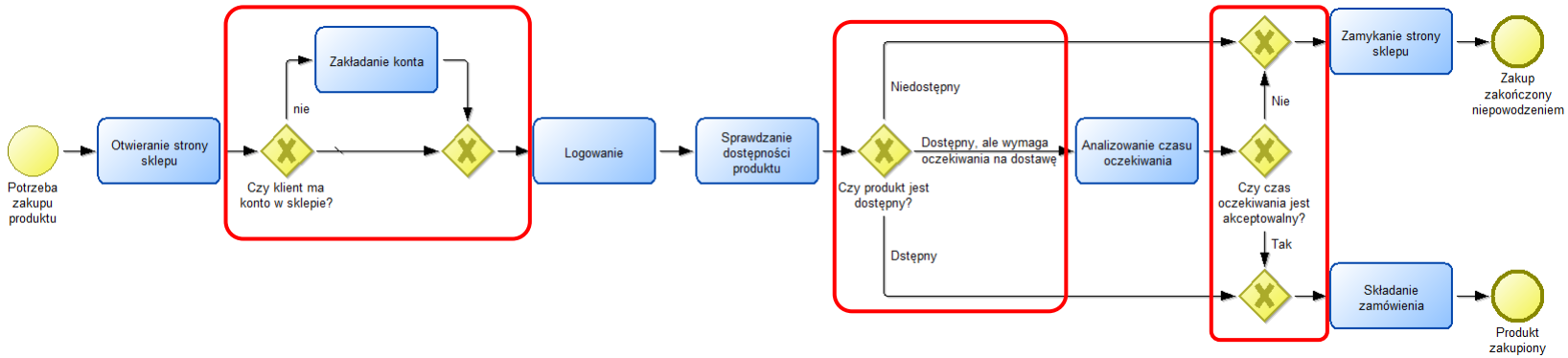




Przepływ domyślny – przepływ bez określonego warunku wybierany przez token gdy żaden z warunków wyjściowych nie jest spełniony.

Bramka wykluczająca – przykład

Zamówienie produktu w sklepie internetowym



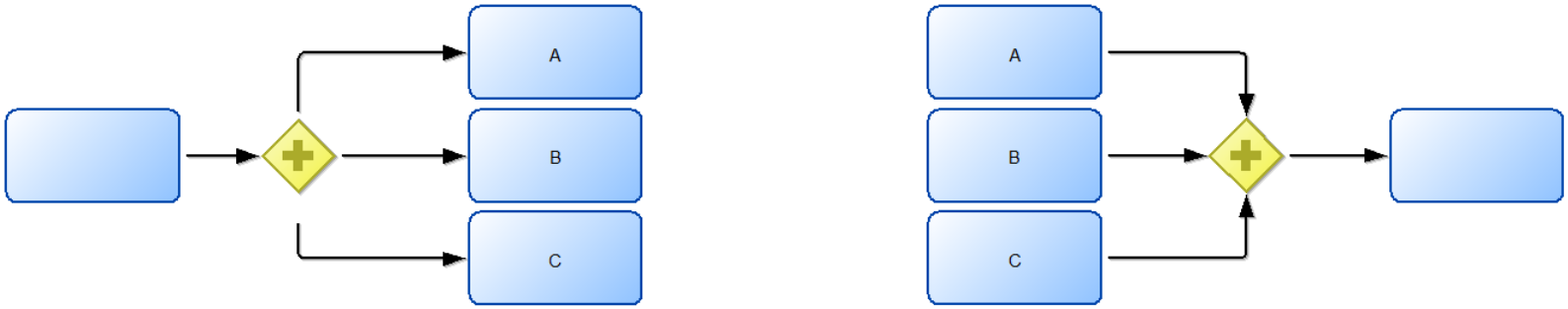
1. Po stwierdzeniu potrzeby zakupu klient otwiera stronę sklepu.
2. Jeżeli ma konto (przepływ domyślny) loguje się, jeżeli nie wcześniej zakłada konto.
3. Sprawdza dostępność produktu. Istnieją trzy możliwości (wykluczające):
   - Produkt dostępny,
   - Produkt niedostępny,
   - Produkt dostępny, ale wymaga zamówienia (przedłużony czas oczekiwania).
4. Jeżeli produkt jest dostępny lub czas oczekiwania jest akceptowalny klient składa zamówienie.
5. Jeżeli produkt jest niedostępny lub czas oczekiwania nie jest akceptowalny opuszcza sklep bez złożenia zamówienia.

Bramka równoległa (*parallel, AND*)

Rozdzielająca bramka równoległa tworzy kilka ścieżek przepływu realizowanych równolegle. Token wchodzący do bramki jest rozdzielany na wszystkie przepływy wychodzące (powstaje rodzina tokenów, każdy z indywidualnym *SubTokenID*). Poszczególne ścieżki są realizowane niezależnie od innych. Przepływ domyślny nie ma w tym przypadku zastosowania.

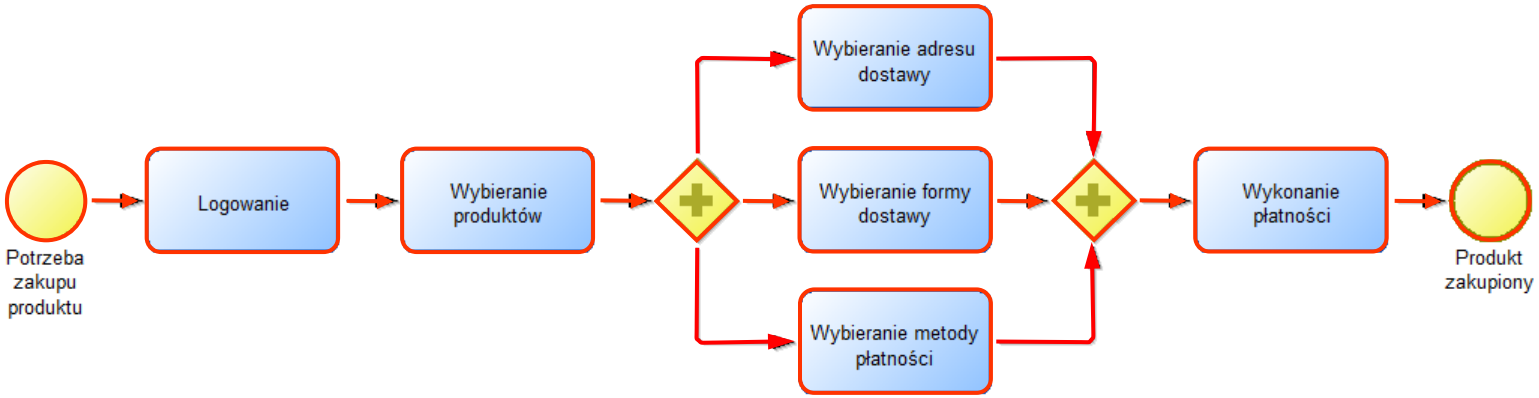
Łącząca bramka równoległa synchronizuje przepływy wejściowe. Oczekuje na zakończenie wszystkich przepływów wchodzących, odcina *SubTokenID* i scala całą rodzinę tokenów w jeden token główny. Wykonanie procesu jest kontynuowane po zakończeniu wszystkich przepływów wchodzących do bramki.

BPMN v.2.0.2, 10.6.3.Parallel Gateway, s.292



Bramka równoległa – przykład

Finalizowanie zamówienia w sklepie internetowym

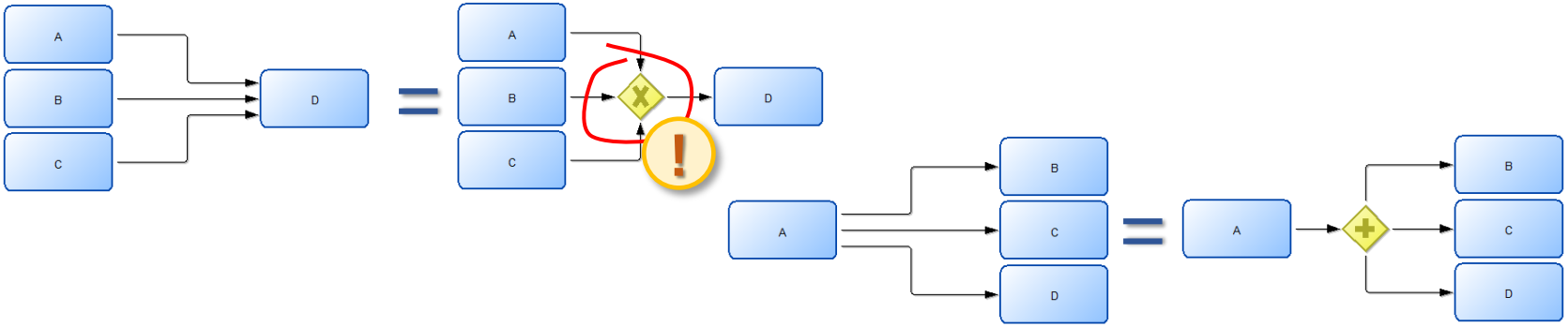


1. Po stwierdzeniu potrzeby zakupu klient loguje się na swoje konto i wybiera produkty.
2. Po skompletowaniu produktów przechodzi do finalizowania zamówienia. W tym celu:
   - wybiera adres dostawy,
   - wybiera formę dostawy,
   - wybiera metodę płatności.
3. Po zakończeniu wszystkich operacji wykonuje płatność, co kończy proces.

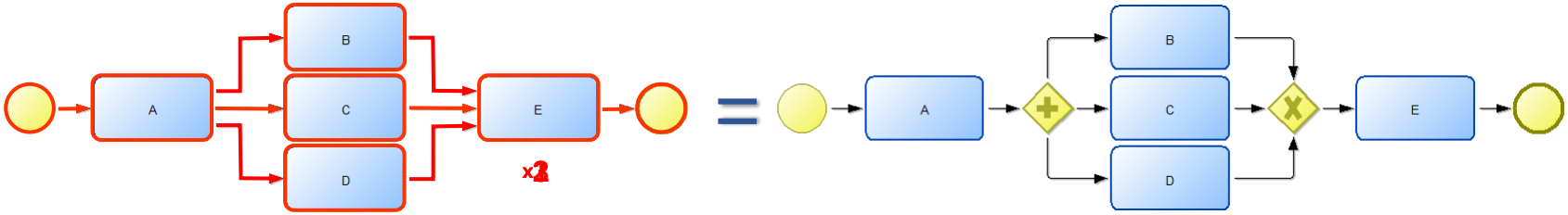
Uwaga: sposób konstrukcji modelu pokazuje, że nie ma znaczenia w jakiej kolejności zostanie wybrany adres, forma dostawy i metoda płatności, ale wszystkie czynności muszą być zakończone przed wykonaniem płatności.

Wybrane cechy aktywności

- Aktywność może mieć kilka przepływów wchodzących i wychodzących.
- Aktywność nie wykonuje synchronizacji tokenów.
- Jeżeli aktywność ma kilka przepływów wychodzących token wybiera wszystkie.



Przykład



Uwaga: Aktywność nie synchronizuje tokenów, więc po dotarciu każdego subtokenu do aktywności "E" będzie ona natychmiast wykonywana i subtoken będzie przepuszczany dalej. W wfekcie aktywność "E" zostanie wykonana trzy razy.

Zdarzenia

Zdarzenie (*event*) reprezentuje oddziaływanie zewnętrzne lub wewnętrzne, które jest istotne dla przebiegu procesu i musi być uwzględnione w modelu w celu zapewnienia jego czytelności. Wystąpienie zdarzenia może być niezwiązane ze zwykłym przebiegiem procesu, a chwila w którym się pojawia może być nieprzewidywalna. Zdarzenia nie generują dodatkowych kosztów wykonania procesu. Typowym zastosowaniem zdarzeń jest zapewnienie komunikacji z innymi procesami lub częściami procesu oraz obsługa sytuacji nadzwyczajnych, które wymagają odejścia od zwykłego trybu działania.

Klasyfikacja zdarzeń ze względu na miejsce wystąpienia

- zdarzenia początkowe – określają sposób inicjalizowania nowego procesu (okrąg z pojedynczą, niepogrubioną krawędzią),
- zdarzenia końcowe – określają miejsce zakończenia procesu (okrąg z pojedynczą pogrubioną krawędzią),
- zdarzenia pośrednie – określają sytuacje występujące w czasie trwania procesu (okrąg z podwójną niepogrubioną krawędzią).

Klasyfikacja zdarzeń ze względu na sposób działania

- rzucające/nadawcze (*throw*) – zdarzenia, które generują informację o wystąpieniu określonego bodźca (generują pewien rezultat),
- chwytające/odbiorcze (*catch*) – zdarzenia, które oczekują na wystąpienie określonego bodźca (wstrzymują przepływ procesu do chwili jego wystąpienia).

Wyzwalacze

Wyzwalacz (*trigger*) określa rodzaj bodźca, który wywołał zdarzenie oraz jego kategorię.

Kategorie zdarzeń w notacji BPMN

- nieokreślone* (*none*) – nie ma określonego wyzwalacza,
- anulowanie (*cancel*) – przerwanie transakcji,
- błąd (*error*) – błąd wykonania czynności lub procesu,
- czas* (*timer*) – określona chwila czasowa,
- eskalacja (*escalation*) – wykonanie dodatkowych operacji wykraczających poza typowy przebieg procesu,
- kompensacja (*compensation*) – wycofanie skutków zakończonych operacji,
- komunikat* (*message*) – wymiana informacji z innym uczestnikiem,
- łącze* (*link*) – przerwanie przepływu sekwencyjnego i wznowienie w innym miejscu,
- sygnał (*signal*) – rozgłoszenie zawiadomienia w ramach całego modelu,
- warunek (*conditional*) – spełnienie wyrażenia warunkowego,
- zakończenie* (*terminate*) – zakończenie wszystkich wątków procesu,
- wielozdarzenie (*multiple*) – wystąpienie kilku bodźców,
- wielozdarzenie równoległe (*parallel mult.*) – jednoczesne wystąpienie kilku bodźców.

* *Wyzwalacze dopuszczalne w modelu poglądowym (s.01-11)*

Zdarzenia początkowe

Zdarzenie początkowe (*start event*) określa sposób inicjowania procesu. Może przechwytywać wyzwalacze, nigdy ich nie generuje. Każde wystąpienie zdarzenia początkowego powoduje powstanie nowej instancji (egzemplarza) procesu i pociąga za sobą wygenerowanie nowego tokenu z kolejnym *TokenID*.

Cechy zdarzeń początkowych

- okrąg z niepogrubioną krawędzią, ikona wyzwalacza biała (niewypełniona);
- brak wejściowych przepływów sekwencyjnych (dozwolony przepływ komunikatu);
- opcjonalne jeżeli warunki uruchomienia procesu są oczywiste;
- obowiązkowe jeżeli w modelu jawnie występuje zdarzenie końcowe;
- dozwolone kilka zdarzeń początkowych.

Kategorie zdarzeń początkowych


nieokreślone* – nie ma określonego wyzwalacza uruchamiającego proces;


czas* – proces uruchamiany w określonej chwili lub cyklicznie;


komunikat* – proces uruchamiany po przechwyceniu komunikatu (*message*);


sygnał – proces uruchamiany po przechwyceniu sygnału (*signal*);


warunek – proces uruchamiany po spełnieniu określonego warunku;


wielozdarzenie – proces uruchamiany przez jedno ze zbioru zdarzeń;


wielozdarzenie równoległe – proces uruchamiany po przechwyceniu wszystkich zdarzeń z określonego zbioru.

Zdarzenia końcowe

Zdarzenie końcowe (*end event*) określa sposób zakończenia procesu. Może generować wyzwalacze (rezultaty), nigdy ich nie przechwytuje.

Cechy zdarzeń końcowych

- okrąg z pogrubioną krawędzią, ikona wyzwalacza czarna (wypełniona);
- brak wyjściowych przepływów sekwencyjnych (dozwolony przepływ komunikatu);
- opcjonalne jeżeli proces nie generuje wyzwalacza;
- obowiązkowe gdy występuje zdarzenie początkowe lub jest generowany wyzwalacz;
- dozwolone kilka zdarzeń końcowych.

Kategorie zdarzeń końcowych


nieokreślone* – proces nie generuje wyzwalacza;


anulowanie – przerwanie transakcji z wycofaniem wykonanych operacji;


błąd – proces zakończony z błędem, wymaga obsługi;


eskalacja – wymagane wykonanie dodatkowych operacji po zakończeniu;


kompensacja – usunięcie skutków czynności, wymaga dodatkowych operacji;


komunikat* – proces wysyła komunikat (*message*) do określonego uczestnika;

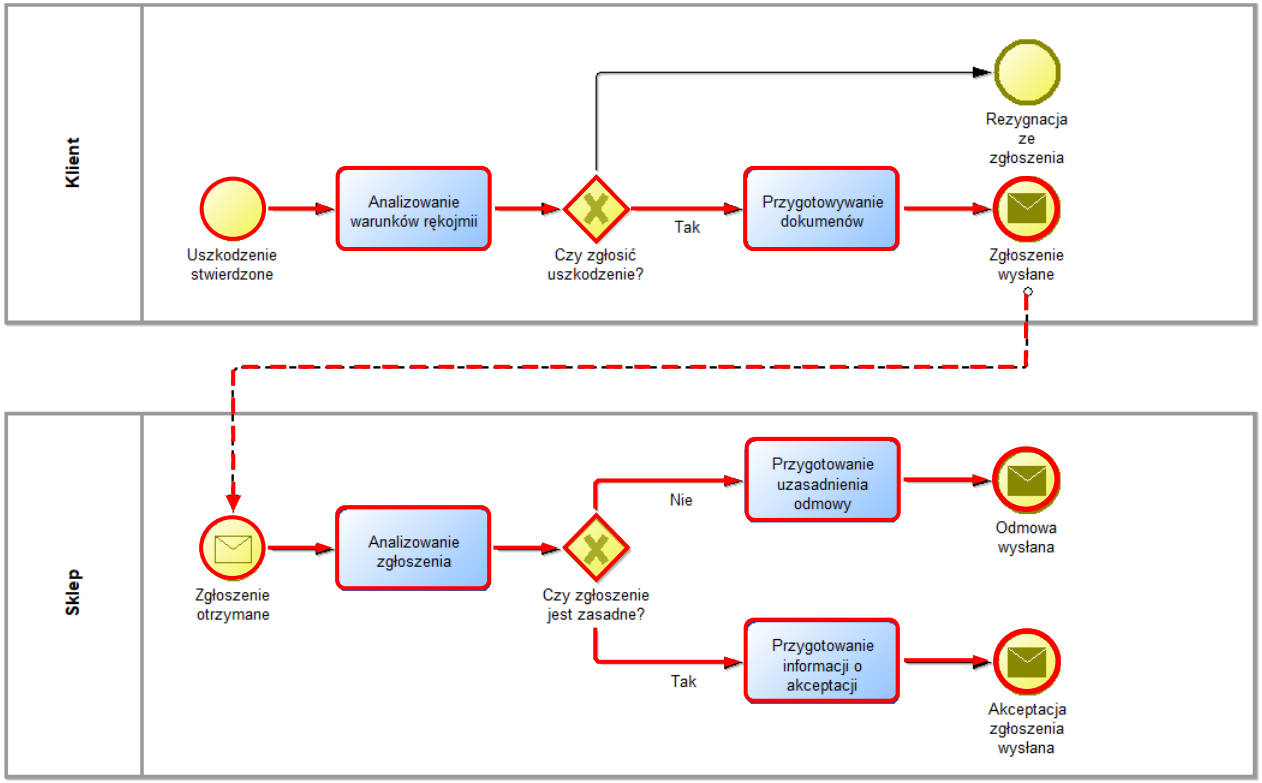

sygnał – rozgłoszenie sygnału (*signal*) do wszystkich uczestników;


zakończenie* – natychmiastowe przerwanie wszystkich wątków procesu;


wielozdarzenie – proces generuje kilka wyzwalaczy.

Zdarzenia końcowe – przykład

Zgłoszenie reklamacji z tytułu rękojmi



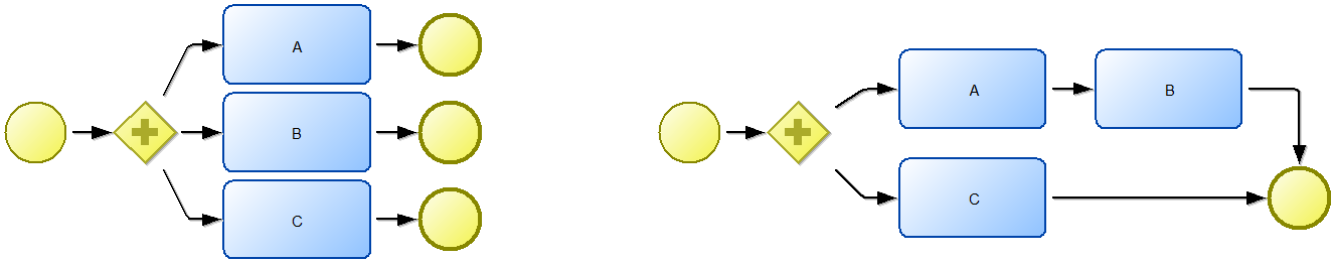
- *Klient* stwierdza uszkodzenie zakupionego produktu,
- Analizuje warunki rękojmi i podejmuje decyzję i wysyła zgłoszenie,
- Odebranie zgłoszenia uruchamia proces w *Sklepie*,
- Pracownik analizuje informacje i akceptuje lub odmawia przyjęcia zgłoszenia.

Zdarzenia końcowe – zakończenie procesu

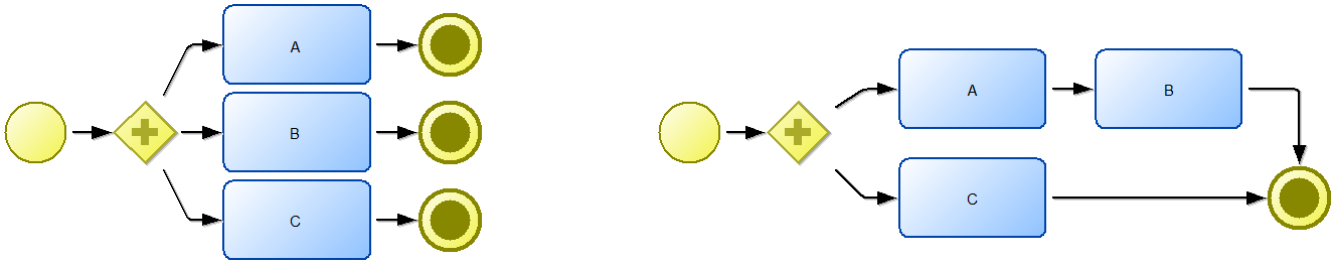
Kategorie zdarzeń końcowych w BPMN 2.0

- zwykłe (nieokreślone) zdarzenie końcowe – kończy pojedynczy wątek procesu;
- przerywające zdarzenie końcowe – kończy wszystkie wątki procesu.

Przykład 1. Procesy zakończą się, gdy tokeny wszystkich wątków dotrą do zdarzenia końcowego, stąd czynności A, B, C zawsze zostaną wykonane.



Przykład 2. Procesy zakończą się, gdy pierwszy subtoken dotrze do zdarzenia końcowego (zostanie zakończony jeden wątek), pozostałe czynności nie będą wykonane, aktualnie realizowane operacje zostaną przerwane.



Zdarzenia pośrednie

Zdarzenie pośrednie (*intermediate event*) występuje pomiędzy zdarzeniem początkowym i końcowym. Może chwytać lub rzucać (generować) wyzwalacze. Wpływa na przebieg procesu, ale nie może uruchamiać lub kończyć instancji procesu.

Typy zdarzeń pośrednich

- zdarzenia pośrednie rzucające (*throwing*) – generują wyzwalacz (*throw trigger*) gdy token procesu dociera do zdarzenia.
- zdarzenia pośrednie chwytające (*catching*) – wstrzymują proces gdy token dociera do zdarzenia, wykonanie jest wznawiane po przechwyceniu wyzwalacza (*catch trigger*). *Uwaga:* Wyzwalacze wygenerowane zanim token osiągnie zdarzenie są ignorowane.

Cechy zdarzeń pośrednich

- okrąg z podwójną niepogrubioną krawędzią;
- w przypadku zdarzeń rzucających ikona wyzwalacza czarna (wypełniona);
- w przypadku zdarzeń chwytających ikona wyzwalacza biała (niewypełniona);
- wymagany przepływ wejściowy i wyjściowy, dozwolony przepływ komunikatu.

Zastosowanie

- prezentacja komunikatów i sygnałów, które muszą być odebrane/wysłane;
- określenie opóźnień czasowych procesu;
- obsługa sytuacji wyjątkowych, kompensacji i eskalacji podczas wykonania procesu.

Kategorie zdarzeń pośrednich

nieokreślone* – nie ma zdefiniowanego wyzwalacza, występuje tylko jako zdarzenie rzucające, używane do monitorowania procesu (kamienie milowe);

czas* – wstrzymuje proces na określony czas lub do pewnej chwili;

eskalacja – generuje wyzwalacz uruchamiający dodatkowe operacje;

komunikat* – generuje komunikat skierowany do określonego użytkownika (z.rzucające) lub oczekuje na nadejście komunikatu (z.chwytające);

kompensacja – generuje wyzwalacz uruchamiający operacje, których zadaniem jest usunięcie skutków zakończonej czynności;

łącze* – połączenie dwóch części procesu w jeden ciąg, eliminuje długie linie przepływów sekwencyjnych, ułatwia realizację pętli i skoków;

sygnał – generuje i rozgłasza sygnał (z.rzucające) lub oczekuje na pojawienie się sygnału (z.chwytające);

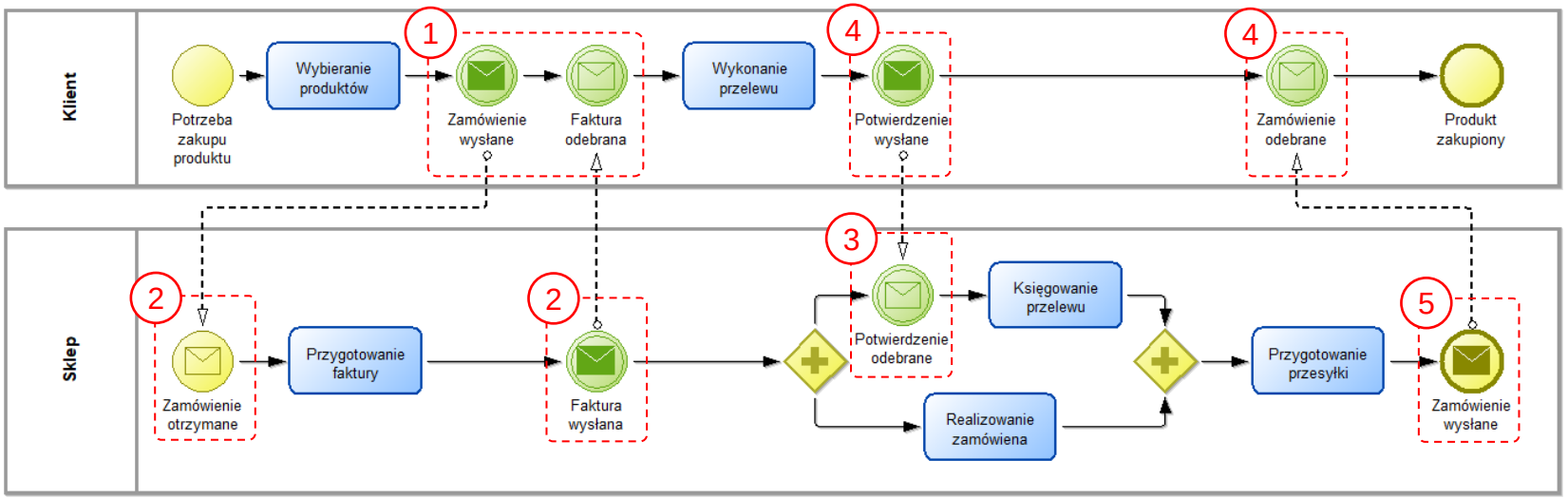
warunek – wstrzymuje proces do chwili spełnienia warunku (proces nie zatrzyma się jeżeli warunek będzie prawdziwy gdy token dotrze do zdarzenia);

wielozdarzenie – generuje kilka wyzwalaczy (z.rzucające) lub oczekuje na wystąpienie jednego z kilku wyzwalaczy (z.chwytające);

wielozdarzenie równoległe – oczekuje na wystąpienie wszystkich zdefiniowanych wyzwalaczy.

Zdarzenia pośrednie – komunikat

Realizacja zamówienia (modyfikacja modelu s.10 – wykorzystanie zdarzeń)



1. *Klient* przygotowuje zamówienie, wysyła je do *Sklepu* i oczekuje na fakturę.
2. Przepływ komunikatu uruchamia proces *Sklepu*, który przyjmuje zamówienie, przygotowuje fakturę i przesyła ją do *Klienta*.
3. Po przesłaniu faktury *Sklep* rozpoczyna realizację zamówienia i oczekuje na wpłatę.
4. Po odebraniu faktury *Klient* wykonuje przelew i wysyła potwierdzenie, a następnie oczekuje na zamówione produkty.
5. Po zakończeniu realizacji, odebraniu przelewu i zaksięgowaniu wpłaty *Sklep* wysyła zamówione produkty, co kończy jego proces
6. Odbiór zamówienia kończy proces *Klienta*.

Zdarzenie komunikat vs. zadania wysyłki i odbioru

Zadanie wysyłki (*send task*) wysyła komunikat do innego uczestnika, jest zakończone po wysłaniu komunikatu.

Zadanie odbioru (*receive task*) oczekuje i odbiera komunikat; jest zakończone po odebraniu komunikatu.

Rzucające zdarzenie komunikat (*throwing*) generuje komunikat skierowany do określonego użytkownika.

Chwytające zdarzenie komunikat (*catching*) oczekuje na nadejście komunikatu, proces jest kontynuowany po odebraniu komunikatu.

Norma BPMN 2.0 nie określa wprost różnic pomiędzy zadaniami wysyłki/odbioru a zdarzeniem komunikat. Zadanie wysyłki jest odpowiednikiem zdarzenia rzucającego, a zdanie odbioru zdarzenia chwytającego wyzwalacz. Różnica wynika z charakteru obiektu: zadania są szczególnym przypadkiem aktywności (reprezentują pracę) i ich wykonanie generuje określony koszt, wykonanie zdarzenia nie generuje dodatkowych kosztów realizacji danego etapu procesu. Dodatkowo: aktywności pozwalają na obsługę sytuacji wyjątkowych (błędów), które mogą wystąpić podczas ich wykonania (*wymaga wykorzystania zdarzeń krawędziowych, element nieomawiany w ramach wykładu*).

Bramki oparte na zdarzeniach

Bramki oparte na zdarzeniach reprezentują punkt rozgałęzienia procesu, w którym wykonanie alternatywnej ścieżki przepływu zależy od wystąpienia pewnego zdarzenia, a nie warunków związanych z danymi procesowymi. Wybór ścieżki może zależeć np. od komunikatu odebranego od innego uczestnika, więc decyzja może być podejmowana w oparciu o dane, które są dla modelowanego procesu niewidoczne.

Przepływy wyjściowe bramek opartych na zdarzeniach nie mogą mieć związanych warunków, muszą prowadzić do zdarzeń pośrednich lub zadania odbioru. Element diagramu umieszczony na przepływie wyjściowym jest częścią konfiguracji bramki i stanowi wyzwalacz, który uruchamia dany przepływ.

Typy bramek opartych na zdarzeniach

- oparte na zdarzeniach początkowych – rozpoczynają proces (bramka nie ma przepływu wejściowego) który zależy od kilku zdarzeń początkowych (*nieomawiana*);
- oparte na zdarzeniach pośrednich – wstrzymują wykonanie procesu do chwili wystąpienia jednego lub kilku zdarzeń.

Elementy dozwolone na przepływach wyjściowych

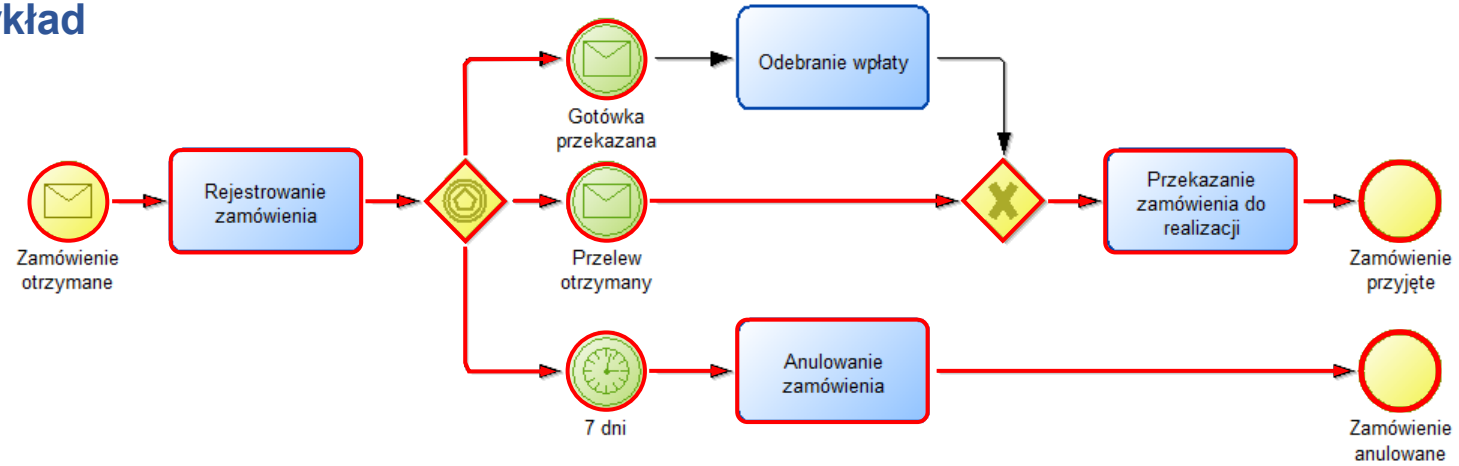
- zdarzenia pośrednie: komunikat, sygnał, timer, zdarzenie warunkowe, wielozdarzenie złożone z wymienionych zdarzeń;
- zadanie odbioru.

Bramka oparta na zdarzeniach pośrednich

Wykluczająca bramka oparta na zdarzeniach pośrednich występuje zawsze przed zdarzeniami, których dotyczy. Gdy token dociera do takiej bramki aktywowane są wszystkie przepływy wychodzące i proces zostaje wstrzymany w oczekiwaniu na wystąpienie zdarzenia związanego z jednym z przepływów. Token wybiera ścieżkę prowadzącą do zdarzenia, które wystąpiło jako pierwsze, pozostałe są dezaktywowane.

Jest to jedyny typ bramki opartej na zdarzeniach pośrednich w BPMN 2.0.

Przykład



Po otrzymaniu zamówienia sklep oczekuje na wpłatę zanim rozpocznie jego realizację (możliwe dwie formy płatności: gotówka lub przelew). Jeżeli wpłata nastąpi przed upływem 7 dni jest księgowania i zamówienie przekazywane jest do realizacji. Jeżeli zamówienie nie będzie opłacone w terminie 7 dni zostaje anulowane.

Dodatkowe materiały

Strona przedmiotu

Zestawienie elementów BPMN, najczęstsze błędy, wzorce procesowe

Modelowanie procesów biznesowych
materiały dla studentów niestacjonarnych kierunku Biznes elektroniczny

https://staff.uz.zgora.pl/gpajak/index.php?doc=mpb-np

Modelowanie procesów biznesowych
materiały dla studentów stacjonarnych kierunku Biznes elektroniczny

https://staff.uz.zgora.pl/gpajak/index.php?doc=mpb-sp