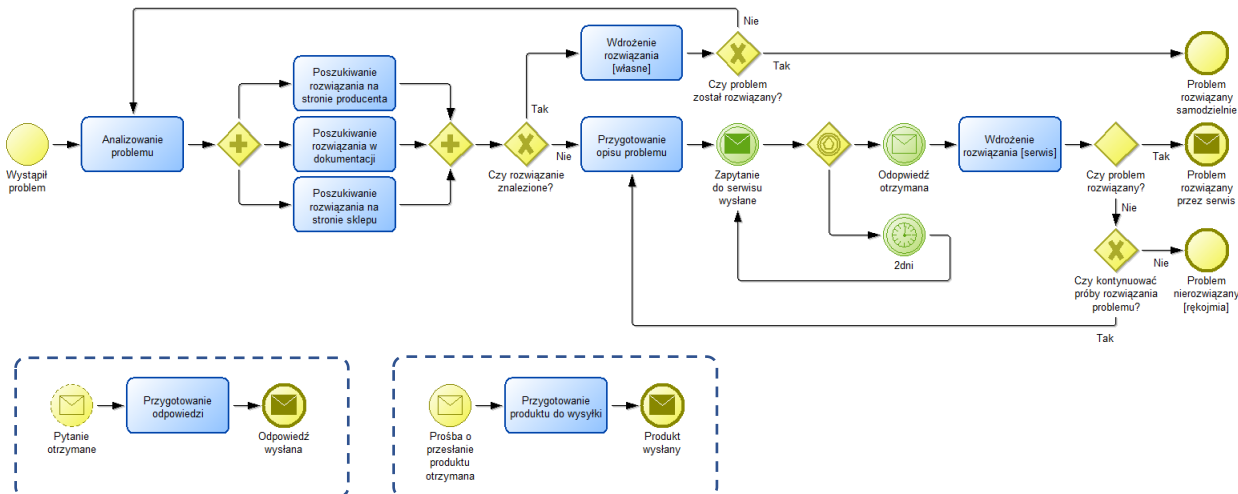


Modelowanie procesów biznesowych

Powtórzenie wiadomości Analiza przykładowego procesu



Charakterystyka analizowanego procesu

Klient zakupił pewien produkt, po pewnym czasie użytkowania pojawił się problem techniczny, który utrudnia lub uniemożliwia jego użytkowanie. Przed skorzystaniem z praw przysługujących z tytułu rękojmi Klient podejmuje próbę rozwiązania problemu samodzielnie lub przy wsparciu serwisu sprzedawcy.

Podstawowy przebieg procesu

1. *Klient* analizuje problem i podejmuje próbę samodzielnego rozwiązania (dokumentacja, strony www sklepu i producenta). Po znalezieniu możliwego rozwiązania podejmuje próbę jego wdrożenia.
2. Jeżeli problem został usunięty proces zostaje zakończony, jeżeli nie *Klient* powraca do analizy i poszukiwania rozwiązania. Operacje mogą być powtarzane wielokrotnie.
3. Jeżeli samodzielna próba rozwiązania nie powiodła się Klient przygotowuje opis problemu, przesyła go do *Serwisu* i oczekuje na odpowiedź. Po uzyskaniu odpowiedzi próbuje wdrożyć zaproponowane rozwiązanie.
4. Jeżeli problem został usunięty *Klient* przesyła informację do *Serwisu* i proces zostaje zakończony, jeżeli nie *Klient* powraca do opisu problemu i ponownie przesyła go do *Serwisu*. Operacje mogą być powtarzane wielokrotnie.
5. Po każdym niepowodzeniu *Klient* podejmuje decyzję czy kontynuować próbę usunięcia problemu przy wsparciu serwisu, czy skorzystać z rękojmi.

Możliwe zakończenia pozytywne

1. Klient sam znalazł rozwiązanie problemu.
2. Klient rozwiązał problem przy wsparciu serwisu.

Przebieg procesu

1. Utworzenie instancji procesu gdy wystąpił problem – zdarzenie początkowe.
2. Analizowanie problemu – zadanie.
3. Samodzielne poszukiwanie informacji (dokumentacja, strony www) – zadania.
4. Rozgałęzienie przepływu na dwie ścieżki alternatywne – bramka wykluczająca, wzorzec WP04
5. Wdrożenie własnego rozwiązania jeżeli zostało znalezione i zakończenie procesu – zadanie i zdarzenie końcowe
6. Przygotowanie opisu problemu, komunikacja z serwisem, wdrożenie zaproponowanego rozwiązania, przesłanie informacji o powodzeniu i zakończenie procesu – zadanie, zdarzenia pośrednie lub zadania wysyłki/odbioru, zdarzenie końcowe wysyłające komunikat.

Przepływ sekwencyjny, aktywności, bramki

Przepływ sekwencyjny określa kolejność elementów przepływu (aktywności, bramki, zdarzenia) w ramach pojedynczego diagramu procesu.

Aktywności (*activities*) są wykonywalnymi elementami procesu i reprezentują punkty, w których wykonywana jest pewna praca.

Zadanie (*task*) jest aktywnością atomową (niepodzielną), której nie można przedstawić jako ciągu elementów przepływu przy założonym stopniu szczegółowości modelu. Określa elementarną czynność realizowaną podczas wykonywania procesu.

Bramki są elementami określającymi logikę przepływu sekwencyjnego w procesie. Definiują zachowanie procesu w punktach, w których przepływ rozgałęzia się i łączy.

Rozdzielająca bramka wykluczająca tworzy kilka alternatywnych ścieżek przepływu, tylko jedna z nich 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 (bez określonego warunku) wybierany gdy żaden z warunków nie jest spełniony.

Zdarzenie (*event*) reprezentuje oddziaływanie zewnętrzne lub wewnętrzne, które jest istotne dla przebiegu procesu i musi być uwzględnione w modelu w celu zapewnienia jego czytelności. Zdarzenia nie generują dodatkowych kosztów wykonania procesu.

Klasyfikacja zdarzeń ze względu na miejsce wystąpienia

- zdarzenia początkowe – określają sposób inicjalizowania nowego procesu,
- zdarzenia końcowe – określają miejsce i sposób zakończenia procesu,
- zdarzenia pośrednie – określają sytuacje występujące w czasie trwania procesu.

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).

Wyzwalacz (*trigger*) określa rodzaj bodźca, który wywołał zdarzenie oraz jego kategorię. Komunikat generuje komunikat skierowany do określonego użytkownika (zdarzenie rzucające) lub oczekuje na nadejście komunikatu (zdarzenie chwytające).

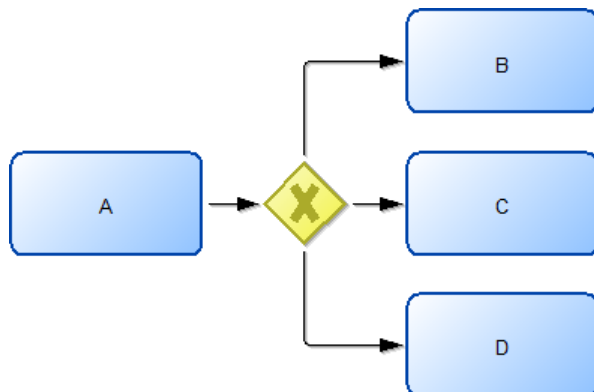
WP04. Wyłączny wybór (*exclusive choice*)

Opis: Podział pojedynczego przepływu na kilka alternatywnych, z których tylko jeden jest realizowany.

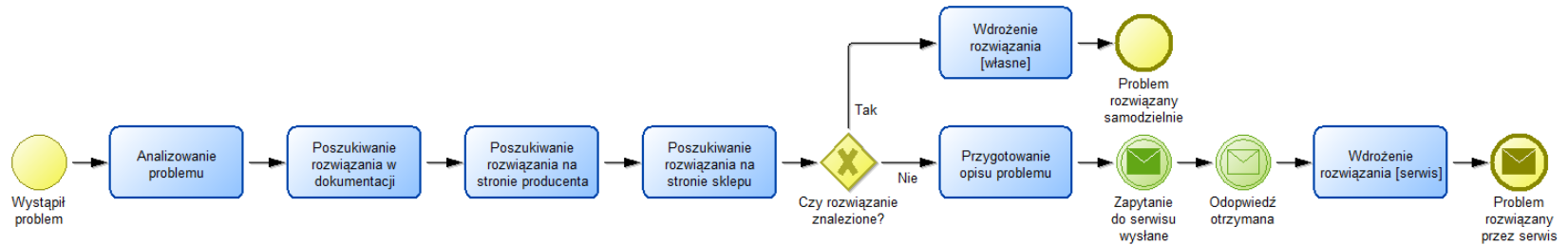
Komentarz: Wzorzec wprowadza warunek, którego ocena pozwala na wybór jednego zadania z kilku kolejnych, warunek wykorzystuje dane dostępne w ramach procesu.

Przykłady: w zależności od stanu magazynu, produkt jest albo pobierany z magazynu albo zamawiany w hurtowni, w zależności od decyzji dziekana podanie studenta jest rozpatrzone albo pozytywnie albo negatywnie.

Element BPMN: rozdzielająca bramka wykluczająca.



Przebieg optymistyczny



1. Po wystąpieniu problemu *Klient* przeprowadza jego analizę.
2. Poszukuje rozwiązania: dokumentacja, strona producenta, strona sklepu.
3. Jeżeli rozwiązanie zostało znalezione wdraża je, co kończy proces.
4. Jeżeli rozwiązanie nie zostało znalezione:
 - przygotowuje opis problemu,
 - wysyła zapytanie do *Serwisu*,
 - oczekuje na odpowiedź,
 - wdraża rozwiązanie,
 - wysyła potwierdzenie rozwiązania problemu co kończy proces.

Przepływy alternatywne i równoległe, powtarzanie operacji

Uzupełnienie procesu

1. Operacje niezależne/równoległe: samodzielne poszukiwanie rozwiązania w różnych źródłach nie jest sekwencyjne, Klient może przeszukiwać różne źródła równoległe (równoległe wątki, bramka niewykluczająca AND, wzorce WP02 i WP03).
2. Wielokrotne powtarzanie operacji: *Klient* może powtarzać proces poszukiwania rozwiązania, może wielokrotnie wysyłać zapytanie do *Serwisu* (pętla lub cykl zrealizowany przy pomocy bramki wykluczającej – wzorce WP10 i WP21)
3. Alternatywne zakończenie procesu: Klient może zrezygnować z prób rozwiązania i skorzystać z praw przysługujących z tytułu rękojmi (bramka wykluczająca).

Rozdzielająca bramka równoległa tworzy kilka ścieżek przepływu realizowanych równoległe. 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 (bez określonego warunku) 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.

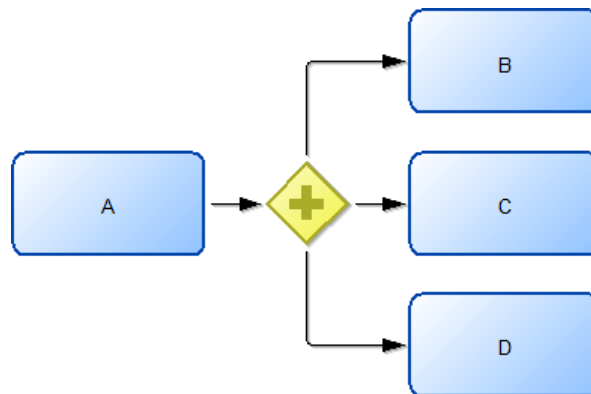
WP02. Podział równoległy (*parallel split*)

Opis: Pojedynczy przepływ jest dzielony na kilka przepływów realizowanych równoległe.

Komentarz: Wzorzec umożliwia podział pojedynczego przepływu na kilka przepływów umożliwiając jednoczesną realizację zadań. Przepływy mogą, ale nie muszą, zostać ponownie zsynchronizowane w przyszłości.

Przykłady: po wpłynięciu zamówienia rozpoczyna się wykonanie wszystkich podzespołów zamówionego wyrobu, po wpłynięciu przelewu za zamówione produkty są one pakowane i wystawiana jest faktura.

Element BPMN: rozdzielająca bramka równoległa.



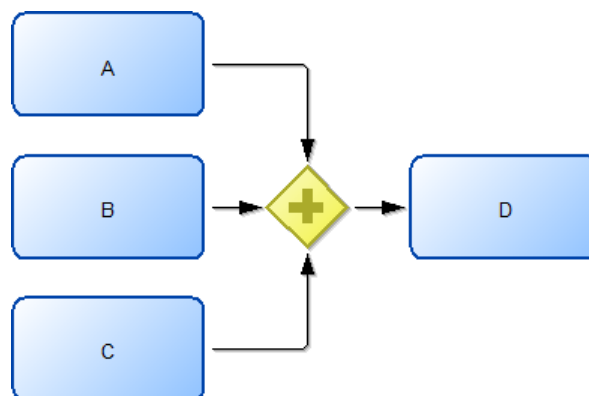
WP03. Synchronizacja (*Synchronization, AND-join*)

Opis: Połączenie kilku przepływów sekwencyjnych w jeden z oczekiwaniem na zakończenie wszystkich przepływów wchodzących (patrz [WP02](#)).

Komentarz: Zapewnia połączenie kilku wątków realizowanych równolegle. Sterowanie jest przekazane do kolejnego węzła po węźle synchronizującym po zakończeniu wątków na wszystkich przepływach równoległych. Nie jest możliwe kilkukrotne wykonanie wątku związanego z daną gałęzią przed zakończeniem synchronizacji.

Przykłady: wysyłka zamówienia może nastąpić gdy zakończono pakowanie produktów i zarejestrowano wpłatę, montaż wyrobu można rozpocząć gdy wykonano wszystkie podzespoły.

Element BPMN: łącząca bramka równoległa.

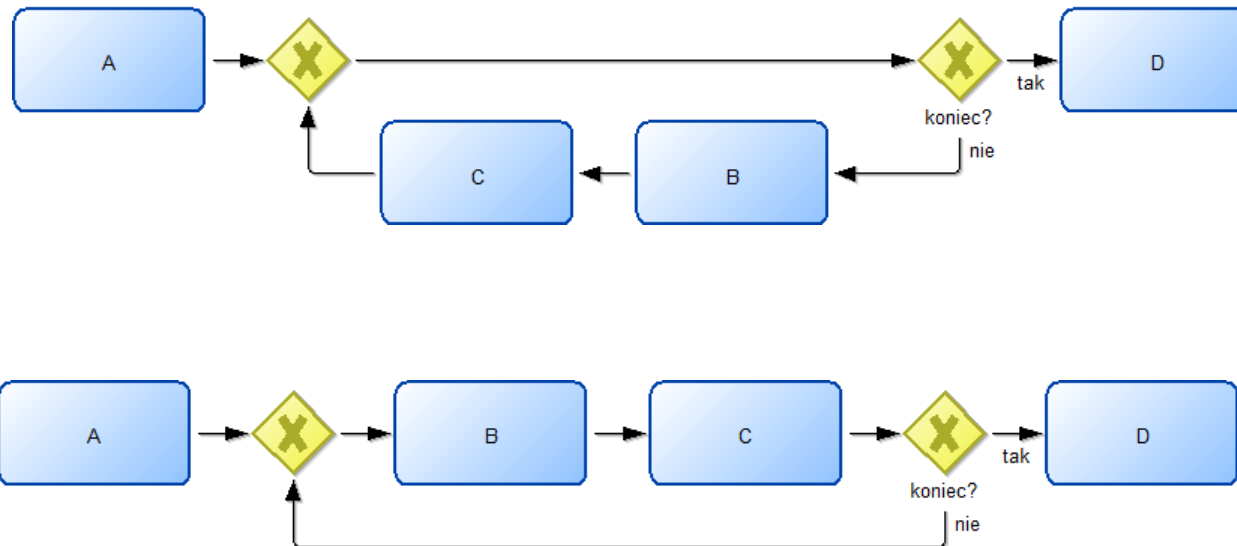


WP21. Pętla strukturalna (*structured loop*)

Opis: Wielokrotne wykonanie zadania lub podprocesu.

Komentarz: Wzorzec wprowadza warunek, którego ocena wpływa na przerwanie lub kontynuowanie pętli. Warunek może być sprawdzany na początku pętli (wariant *while*: zadania mogą nie wykonać się ani razu) lub na końcu pętli (wariant *repeat*: zadania wykonują się co najmniej jeden raz). Pętla strukturalna ma jeden punkt wejścia i jeden punkt wyjścia.

Przykłady: jeżeli są wolne zaproszenia kontynuuj ich rozsyłanie, powtarzaj pobieranie produktów z magazynu aż całe zamówienie zostanie skompletowane.

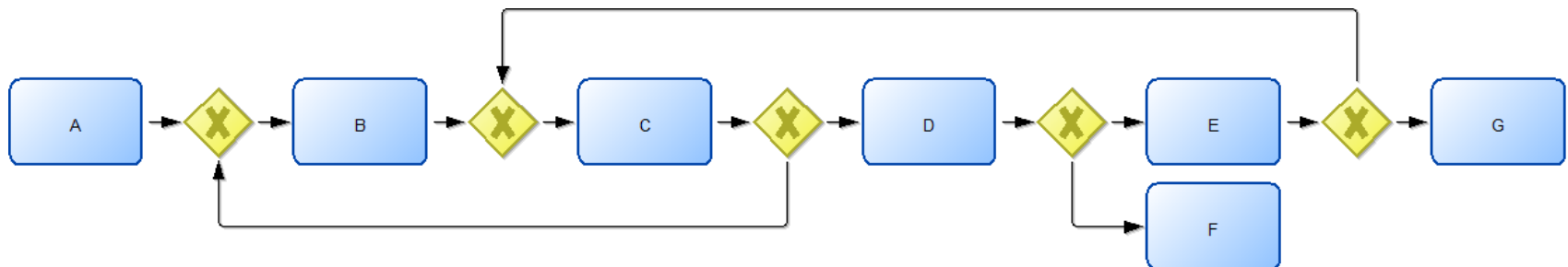


WP10. Dowolny cykl (*arbitrary cycle*)

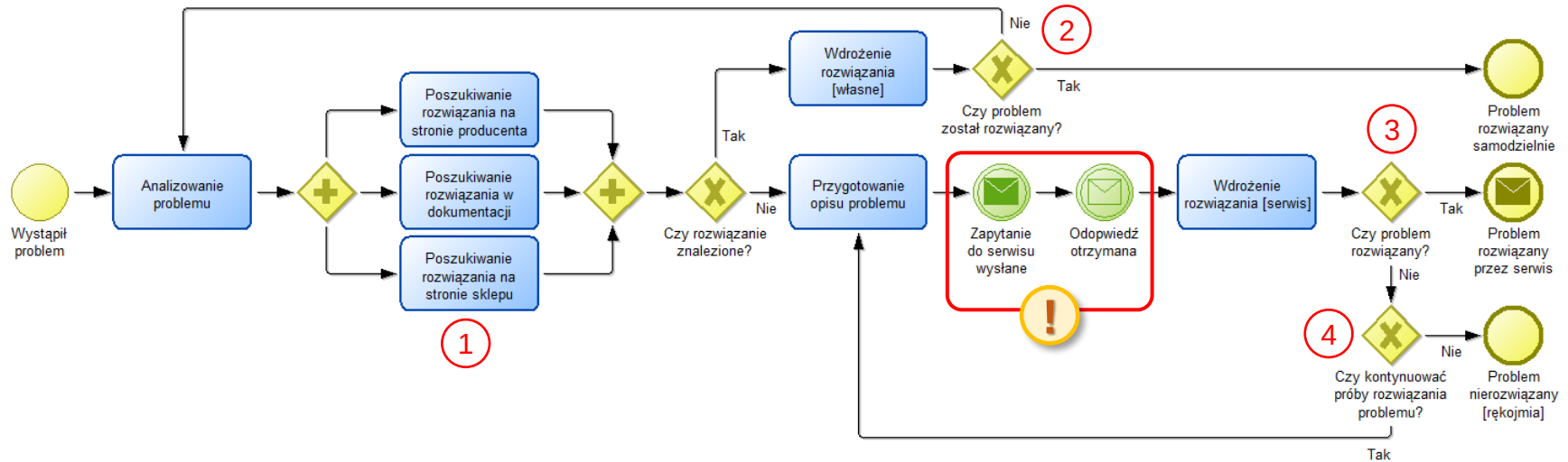
Opis: Cykl niestukturalny o większej niż jeden liczbie punktów wejścia lub wyjścia.

Komentarz: Wzorzec pozwala na powtarzanie zadań w sposób nieustrukturyzowany, bez ograniczeń dotyczących formatu. Umożliwia odwzorowanie złożonych cykli może jednak prowadzić do niejednoznaczności.

Przykłady: w trakcie kompletowania zamówienia klient wielokrotnie szuka produktu i dodaje go do koszyka, po przejściu do realizacji zamówienia może zapłacić albo powrócić do koszyka i kontynuować kompletowanie zamówienia, albo wycofać się z zakupu.



Podstawowy przebieg procesu



1. Operacje niezależne.
2. Wielokrotne powtarzanie procesu poszukiwania rozwiązania (cykl).
3. Wielokrotne przesyłanie zapytania do Serwisu (cykl).
4. Decyzja o skorzystaniu z rękojmi (alternatywne wyjście z cyklu i zakończenie).

Problem: Serwis nie reaguje na otrzymane zapytanie, proces utknie.

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.

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.

Elementy dozwolone na przepływach wyjściowych

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

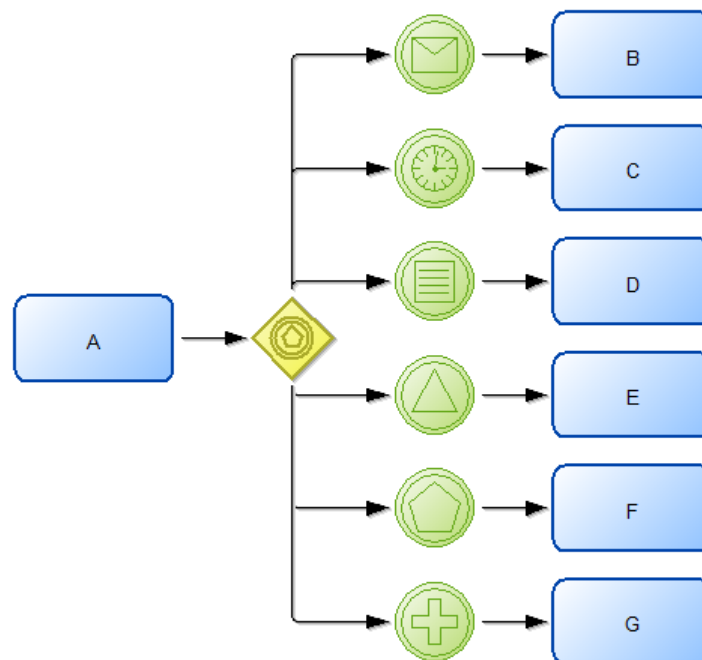
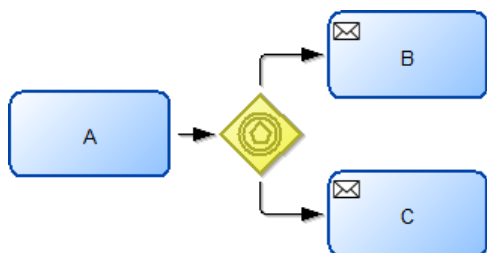
WP16. Odroczony wybór (*differed choice*)

Opis: Wybór jednego z przepływów alternatywnych na podstawie interakcji ze środowiskiem operacyjnym.

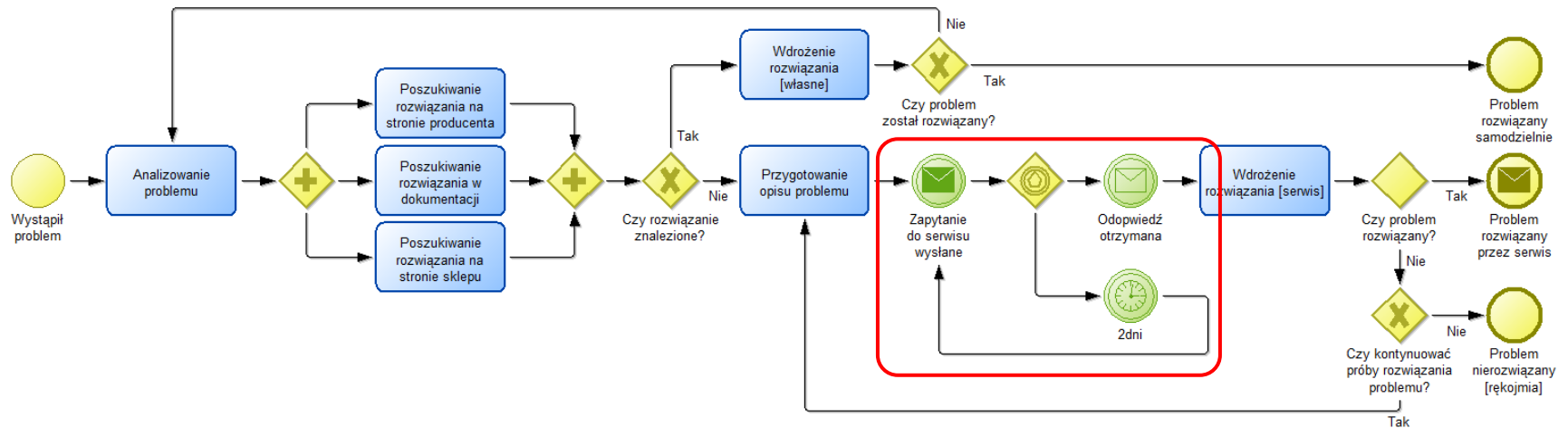
Komentarz: Wybór przepływu odbywa się na podstawie stanu procesu, a nie danych, które są przetwarzane. Aktywowany jest przepływ związany z wyzwalaczem, który pojawił się jako pierwszy, pozostałe przepływy są wycofane.

Przykłady: sklep oczekuje na potwierdzenie wpłaty, jeżeli informacja dotrze w ciągu pięciu dni zamówienie jest wysyłane, gdy minie ustalony czas zostanie wycofane.

Element BPMN: bramka oparta na zdarzeniach pośrednich.



Uzupełniony proces



Klient oczekuje na reakcję Serwisu przez dwa dni, jeżeli nie otrzymuje odpowiedzi ponownie wysyła zapytanie.

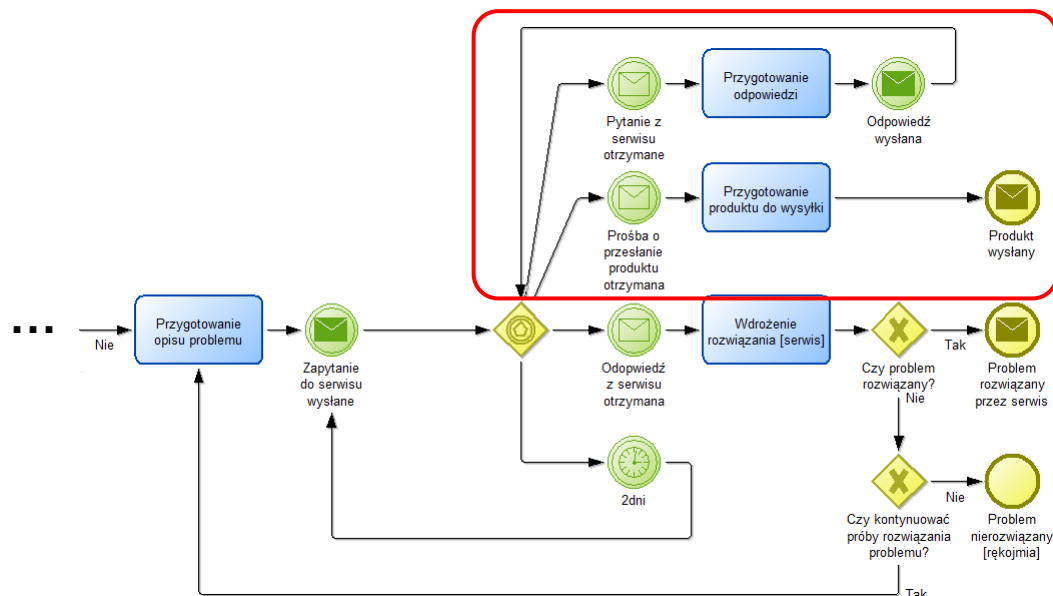
Możliwe uzupełnienie (nieuwzględnione): jeżeli Serwis nie reaguje pomimo wielokrotnego przesłania zapytania można przerwać cykl (bramka wykluczająca).

Rozszerzenie: uwzględnienie reakcji Serwisu

Dodatkowe elementy:

- rozwiązanie problemu przez *Serwis* wymaga uzupełnienia informacji (*Serwis* przesyła zapytanie do *Klienta*),
- zdalne rozwiązanie problemu nie jest możliwe, *Serwis* prosi o przesłanie produktu.

Rozwiązanie: dodatkowe przepływy alternatywne wychodzące z bramki opartej na zdarzeniach uwzględniające dodatkowe komunikaty wysłane przez *Serwis*.



Problem: istotna komplikacja i utrudniona czytelność modelu.

Podproces zdarzeniowy jest szczególnym przypadkiem podprocesu, który nie jest uruchamiany przez proces nadrzędny tylko przez określony wyzwalacz zdarzenia. Jest używany wewnątrz pewnego procesu nadrzędnego (procesu głównego lub innego podprocesu), jednak nie występuje w jego sekwencji przepływu.

Cechy podprocesów zdarzeniowych

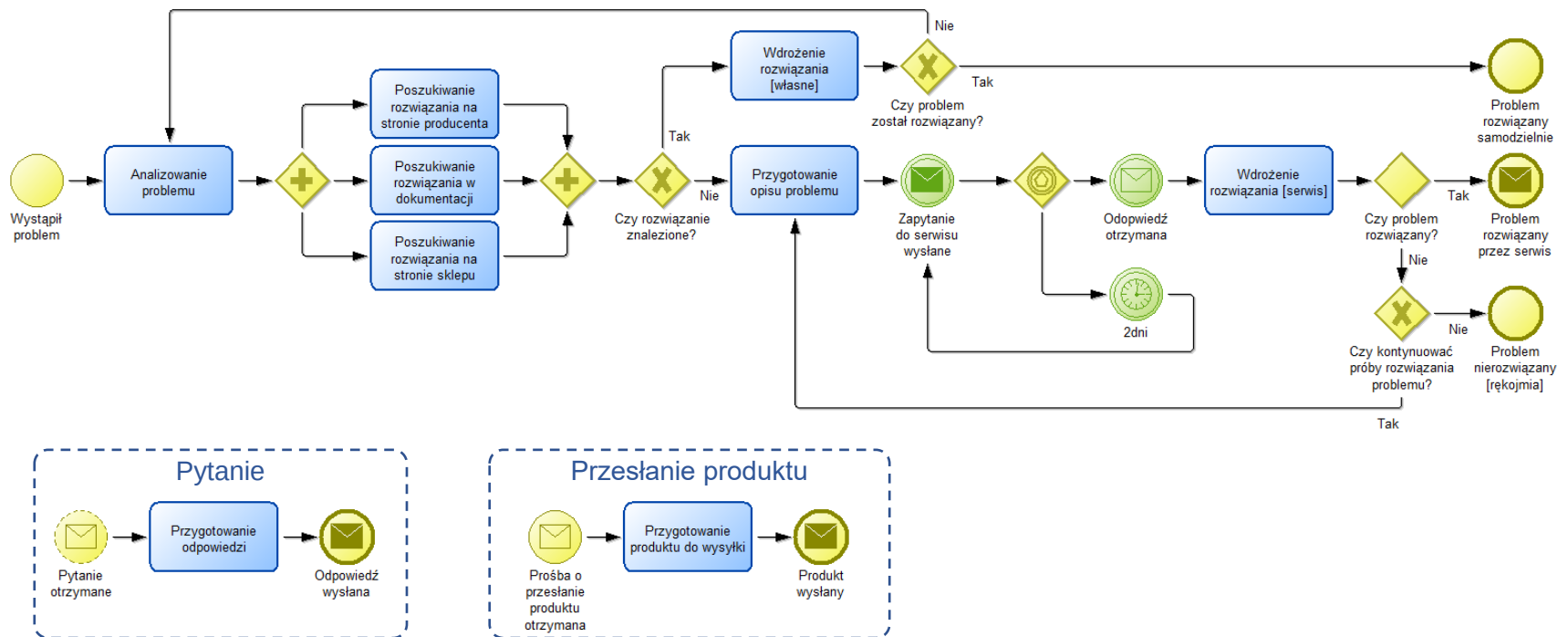
- ma dokładnie jedno zdarzenie początkowe z określonym wyzwalaczem, dozwolone: błąd, czas, eskalacja, kompensacja, komunikat, sygnał, warunek, wielozdarzenie, wielozdarzenie równoległe;
- nie ma przepływu wejściowego i wyjściowego;
- może być uruchomiony tylko gdy proces nadrzędny jest aktywny, może nie być wykonany ani razu lub wielokrotnie (po każdym przechwyceniu wyzwalacza).

Typy podprocesów zdarzeniowych w BPMN 2.0

- Przerywające (*interrupting*) – po przechwyceniu wyzwalacza przerywają proces nadrzędny i wykonują własny przepływ sekwencyjny;
- Nieprzerywające (*non-interrupting*) – po przechwyceniu wyzwalacza wykonują własny przepływ sekwencyjny równoległe z procesem nadrzędnym, ich zakończenie nie kończy procesu nadrzędnego.

Typ procesu określa zdarzenie początkowe: okrąg z ciągłą krawędzią dla procesów przerywających lub okrąg z przerywaną krawędzią dla procesów nieprzerywających.

Wykorzystanie podprocesów zdarzeniowych



1. Nieprzerywający podproces zdarzeniowy "Pytanie" uruchamiany komunikatem przesyłanym przez Serwis, gdy występuje konieczność uzupełnienia informacji (nie przerywa procesu głównego, zdalne rozwiązanie problemu jest kontynuowane).
2. Przerywający podproces zdarzeniowy "Przesłanie produktu" uruchamiany komunikatem przesyłanym przez Serwis, gdy zdalne rozwiązanie problemu nie jest możliwe (przerywa proces główny, próba zdalnego rozwiązania problemu została zakończona).