

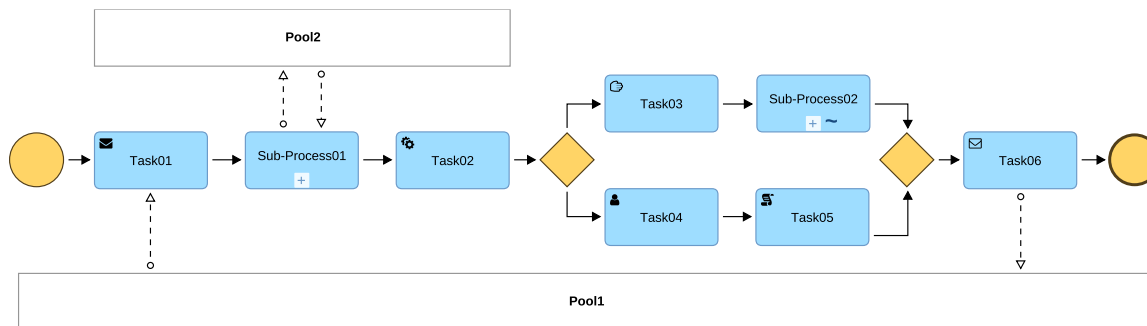
Modelowanie procesów biznesowych

Aktywności

Typy zadań

Podprocesy wbudowane i ad-hoc

Elementy wywoływane





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

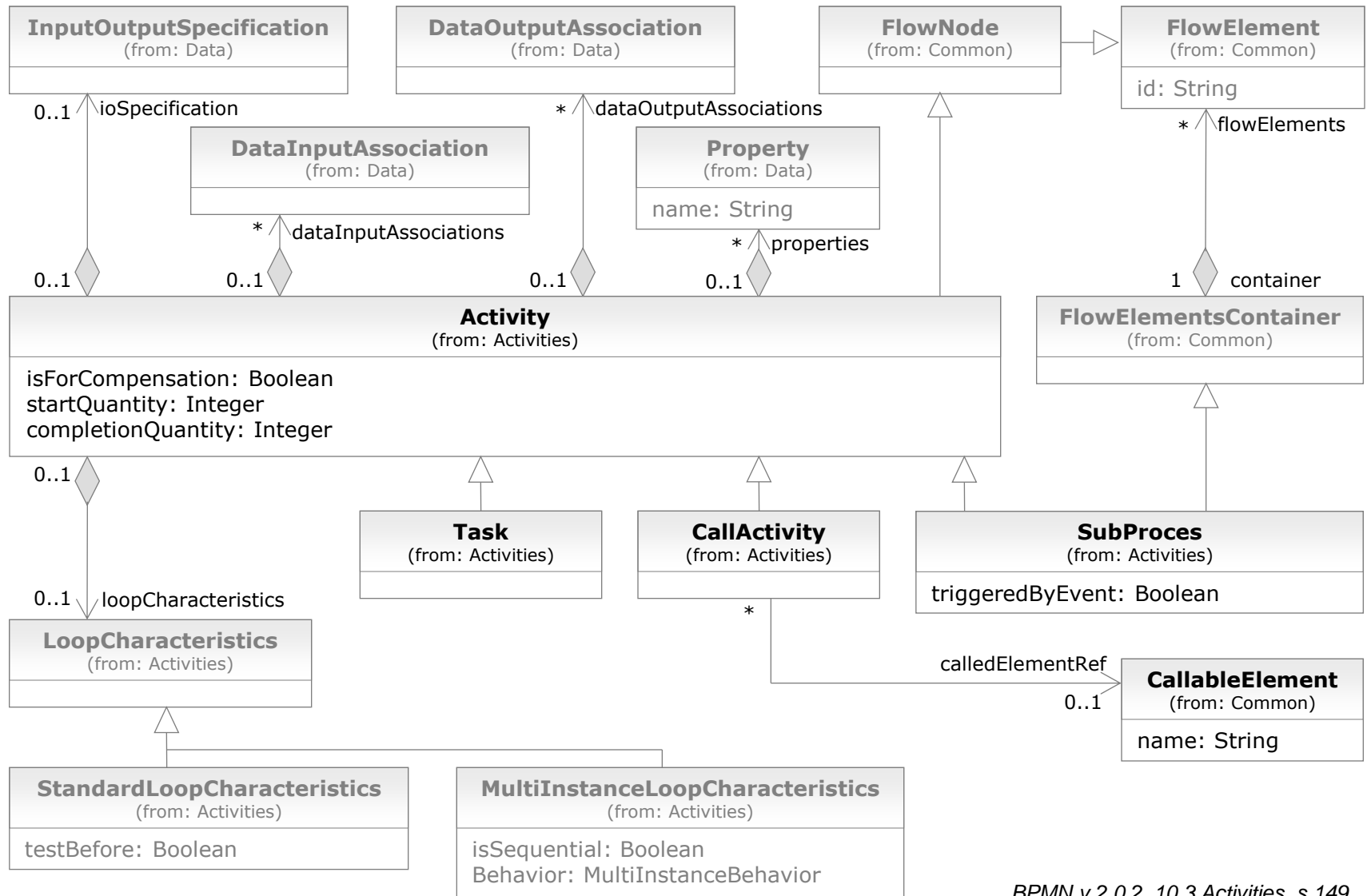
- **Zadanie** (*ang. task*) – aktywność atomowa, nie będzie rozbijana na operacje składowe;
- **Podproces** (*ang. sub-process*) – aktywność złożona, opisana ciągiem elementów przepływu;
- **Aktywność wywoływana** (*ang. call activity*) – określa punkt procesu, w którym uruchamiany jest proces globalny lub zadanie globalne (używane przy automatyzacji).

Opcjonalne cechy aktywności w BPMN 2.0*

- **Pętla standardowa** (*ang. standard loop*) – wykonanie aktywności jest powtarzane wielokrotnie do chwili spełnienia określonego warunku lub określoną liczbę razy;
- **Pętla wieloinstancyjna** (*ang. multi-instance loop*) – aktywność występuje w wielu instancjach (egzemplarzach), które są wykonywane równolegle lub sekwencyjnie;
- **Kompensacja** (*ang. compensation*) – realizuje zadanie kompensacji (wycofania skutków) zakończonych wcześniej czynności.

* zaawansowane elementy BPMN, będą omawiana na osobnym wykładzie

Aktywności – diagram klas



BPMN v.2.0.2, 10.3.Activities, s.149

Atrybuty klasy Activity

- **isForCompensation**: boolean – określa czy aktywność realizuje kompensację,
- **startQuantity**: integer – liczba tokenów, która musi dotrzeć przed wykonaniem aktywności,
- **completionQuantity**: integer – liczba tokenów, która musi być wygenerowana.

Powiązana klasy Activity

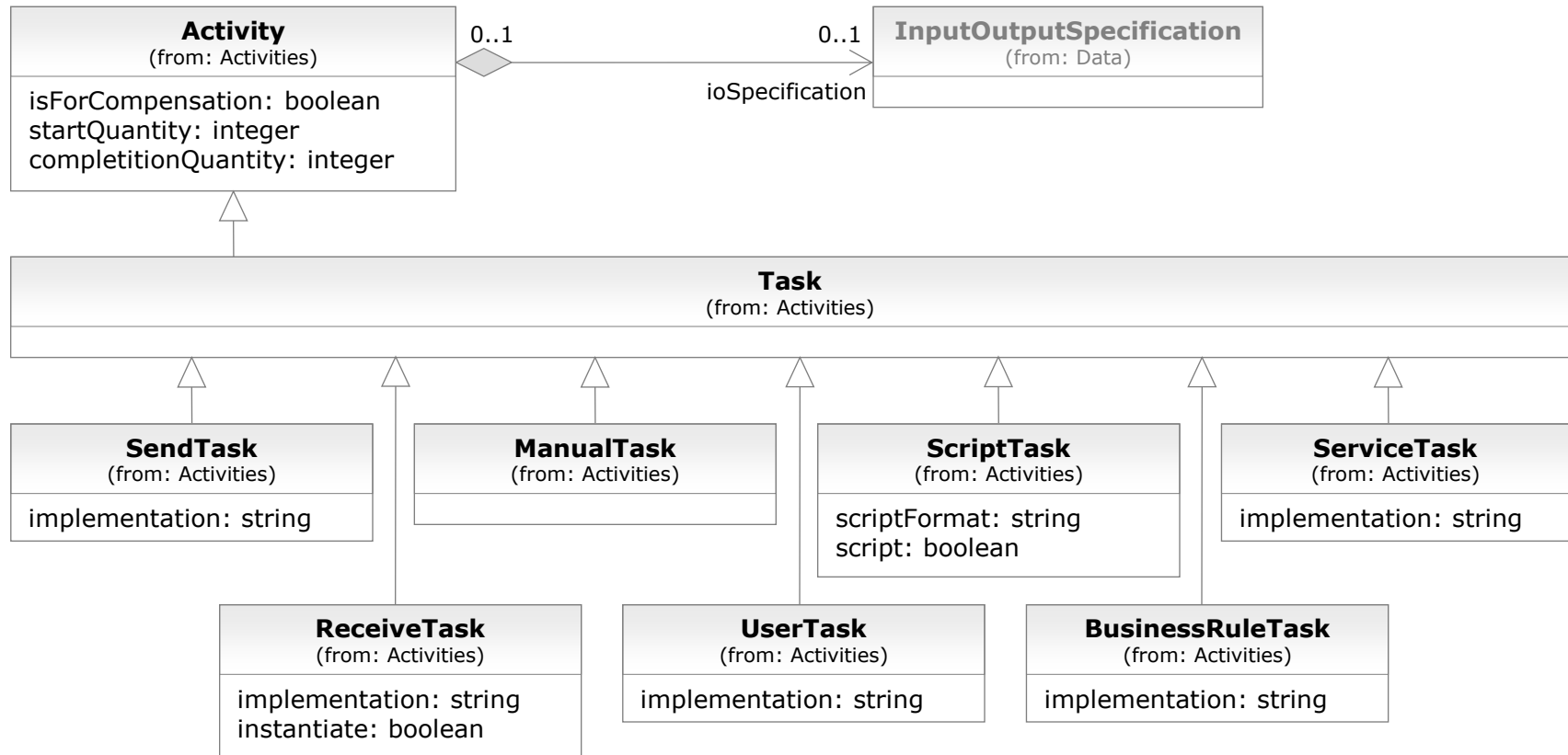
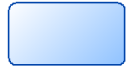
- **loopCharacteristics**: LoopCharacteristics [0..1] – określa czy wykonanie aktywności jest powtarzane i sposób powtórzenia (pętla zwykła lub wieloinstancyjna),
- **ioSpecification**: InputOutputSpecification [0..1] – agreguje zestaw danych przekazywanych jako argumenty wejścia/wyjścia aktywności,
- **dataInputAssociations**: DataInputAssociation [0..*] – określa dane wejściowe aktywności,
- **dataOutputAssociations**: DataOutputAssociation [0..*] – określa dane wyjściowe aktywności,
- **properties**: Property [0..*] – lista własności definiowanych przez model.

Zadanie (ang. *task*) jest aktywnością atomową (niepodzielną), która reprezentuje podstawowe działanie dla założonego stopnia szczegółowości modelu. Sposób wykonania określa typ.

Typy zadań w BPMN 2.0

- **Abstrakcyjne*** (ang. *abstract task*) – zadanie bez określonego typu;
- **Wysyłka** (ang. *send task*) – wysyłka komunikatu do innego uczestnika;
- **Odbiór** (ang. *receive task*) – odbiera komunikat przesłany przez innego uczestnika;
- **Użytkownik*** (ang. *user task*) – operacja wykonywana przez człowieka przy wsparciu oprogramowania zgodnie z przygotowanym harmonogramem;
- **Manualne** (ang. *manual task*) – operacja niezautomatyzowana wykonywana przez człowieka bez wsparcia/nadzoru oprogramowania;
- **Skrypt** (ang. *script task*) – operacja automatyczna, bez udziału człowieka, wykonywana przez skrypt realizowany przez silnik procesów biznesowych;
- **Reguła biznesowa** (ang. *business rule*) – operacja automatyczna, lub pół automatyczna wykonywana przez silnik reguł biznesowych;
- **Serwis*** (ang. *service task*) – zadanie realizowane przez zewnętrzną usługę (np. serwis WWW) lub aplikację.

* Typy zadań dopuszczalne w modelu poglądowym (W01, s.14)





Zadanie wysyłki (ang. send task)

- jest identyfikowane ikoną z czarną (wypełnioną) kopertą;
- wysyła komunikat do innego uczestnika gdy token osiąga zadanie;
- jest zakończone po wysłaniu komunikatu;
- jest wykonywane w pełni automatycznie (bez udziału człowieka);
- odbiorca komunikatu może być identyfikowany przez połączenie zadania z uczestnikiem przepływem komunikatu.

Zadanie odbioru (ang. receive task)

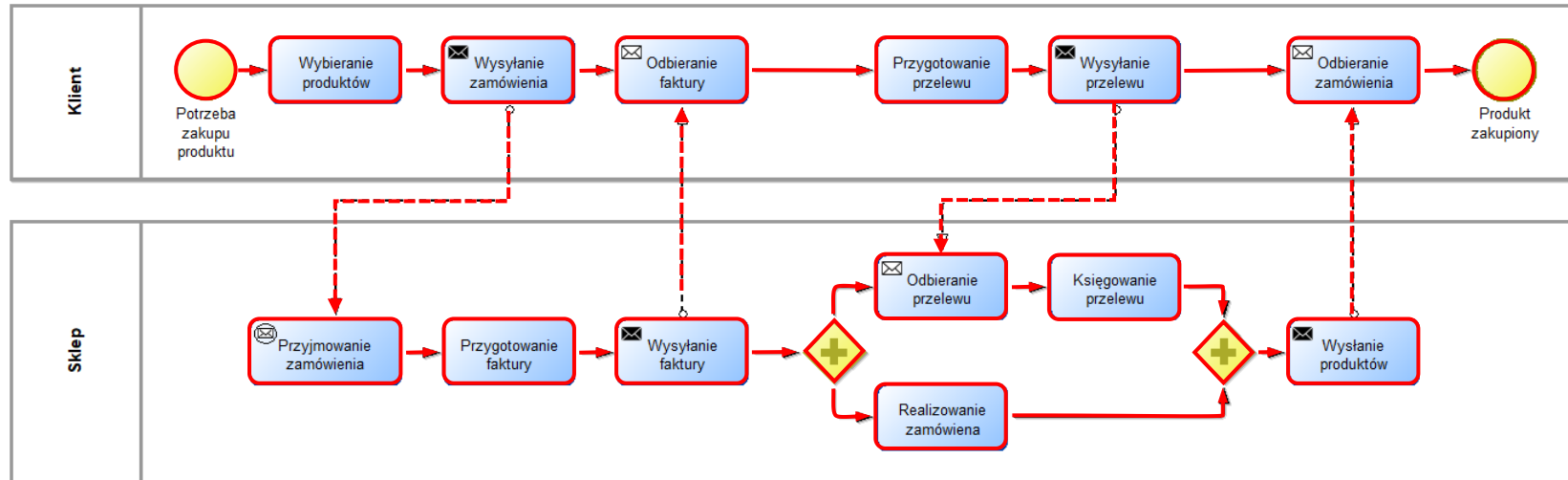
- jest identyfikowane ikoną z białą (niewypełnioną) kopertą;
- jest wykonywane gdy token osiąga zadanie, oczekuje na przepływ komunikatu i odbiera komunikat;
- proces jest kontynuowany po odbiorze komunikatu;
- nadawca może być identyfikowany przez połączenie zadania z uczestnikiem przy pomocy przepływu komunikatu.

Uwaga: Obydwa zadania zakładają wysoki stopień automatyzacji, nie powinny obejmować działań związanych z przygotowaniem lub przetwarzaniem komunikatu. Tego typu operacje powinny być przedstawione jako aktywności poprzedzające lub następujące po zdarzeniu.

Zadania wysyłki i odbioru – przykład



Realizacja zamówienia (procesy prywatne)



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* przygotowuje i wysyła przelew, 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*.



Zadania manualne oraz użytkownika reprezentują czynności, które nie mogą być zrealizowane w pełni automatycznie i wymagają udziału człowieka.

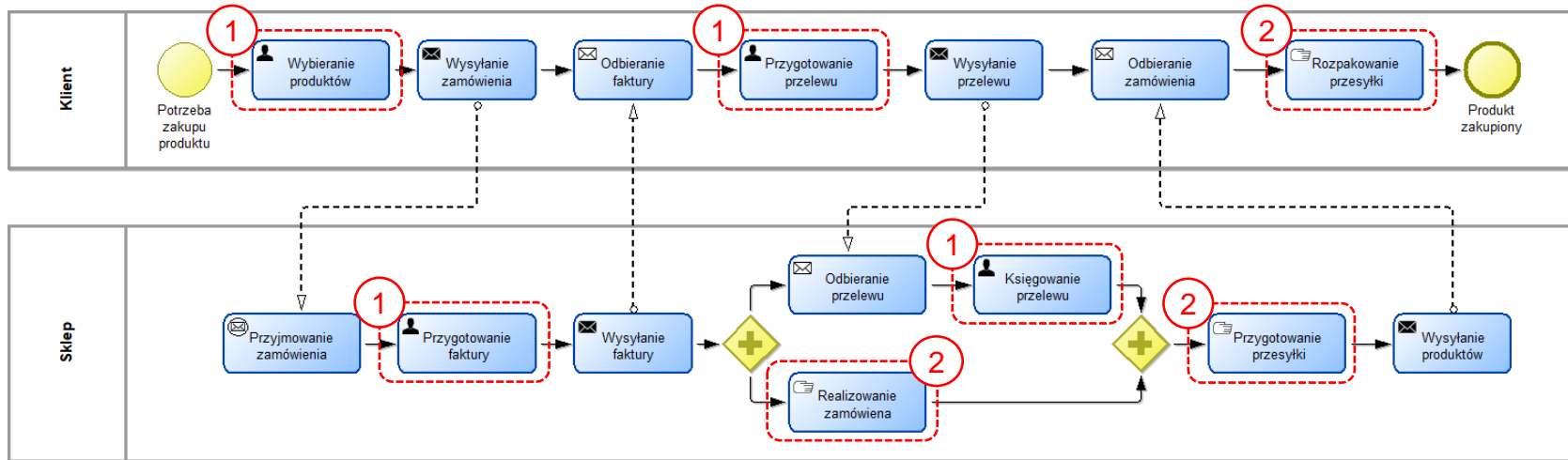
Zadanie manualne (ang. manual task)

- jest identyfikowane przez ikonę z otwartą dłonią;
- jest operacją realizowaną przez człowieka bez wsparcia i nadzoru ze strony silnika procesów biznesowych (silnik procesów nie może śledzić przebiegu tego typu zadań).

Zadanie użytkownika (ang. user task)

- jest identyfikowane przez ikonę z sylwetką człowieka;
- jest realizowane przez człowieka przy wsparciu aplikacji, przebieg jest nadzorowany przez składnik silnika procesów nazywany menadżerem zadań (*task manager*);
- opcjonalnie może być związane z pewnym interfejsem, który umożliwia realizację reprezentowanej czynności, domyślnie technologia interfejsu nie jest określona;
- do implementacji interfejsu standard BPMN 2.0 rekomenduje technologię zgodną ze standardem WS-HumanTask (rozszerzenie standardu BPEL – *Business Process Execution Language*).

Realizacja zamówienia (modyfikacja modelu ze s.8)



1. Zadania użytkownika (*user tasks*):

- Wybieranie produktów, Przygotowanie przelewu;
- Przygotowanie faktury, Księgowanie przelewu.

2. Zadanie manualne (*manual task*):

- Rozpakowanie przesyłki;
- Realizacja zamówienia, Przygotowanie przesyłki.



Skrypt i reguła biznesowa reprezentują czynności, które są realizowane automatycznie bez udziału człowieka.

Zadanie skryptowe (ang. script task)

- jest identyfikowane przez ikonę z zapisaną kartką;
- reprezentuje czynność realizowaną przez zestaw poleceń zapisany w języku programowania (skrypt) dostarczony przez silnik procesów biznesowych;
- jest używane są do automatyzacji często wykonywanych, powtarzalnych operacji, które nie wymagają rozbudowanej analizy danych procesowych.

Reguła biznesowa (ang. business rule task)

- jest identyfikowana przez ikonę z arkuszem/tabelą;
- reprezentuje czynność wykonywaną przez silnik reguł biznesowych (business rules engine) – narzędzie, które pozwala na wykonanie złożonych kalkulacji i analiz (narzędzie zewnętrzne lub element systemu);
- dostarcza mechanizmy pozwalające na uruchomienie reguły biznesowej, przekazanie danych wejściowych i odbiór uzyskanych wyników.

Uwaga: obydwa zadania są uruchamiane gdy token osiąga zadanie, proces jest kontynuowany (token przechodzi do kolejnego elementu) gdy wykonanie skryptu lub reguły zostanie zakończone.

DMN – Decision Model and Notation

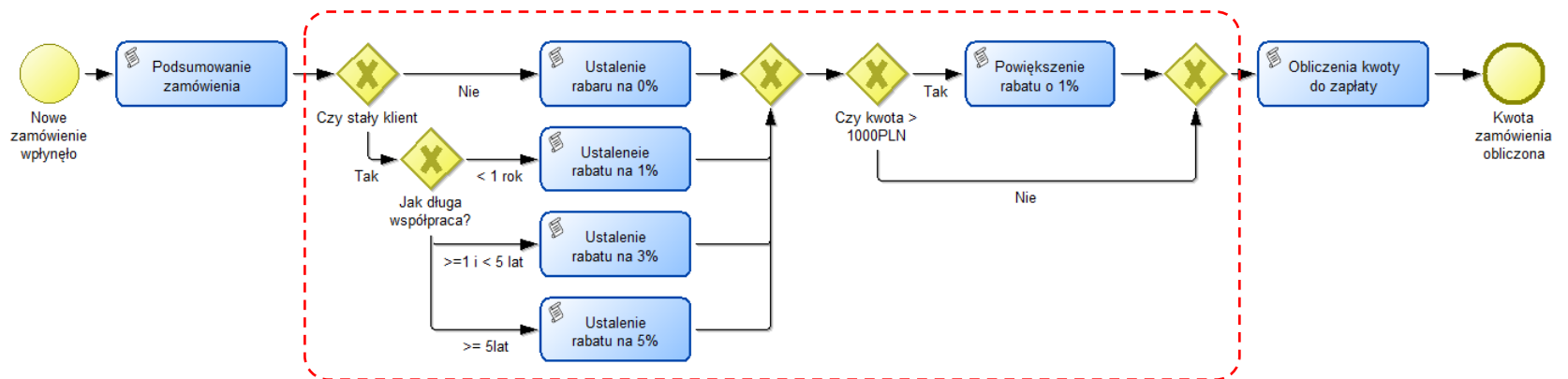
DMN – notacja opracowana i rozwijana przez OMG, przeznaczona do tworzenia modeli decyzyjnych w sposób umożliwiający częściowe lub w pełni automatyczne podejmowanie decyzji biznesowych. Nie jest elementem standardu BPMN, ale jest rekomendowana jako podstawowy sposób opisu reguł biznesowych w modelach BPMN.

Podstawowe elementy standardu DMN

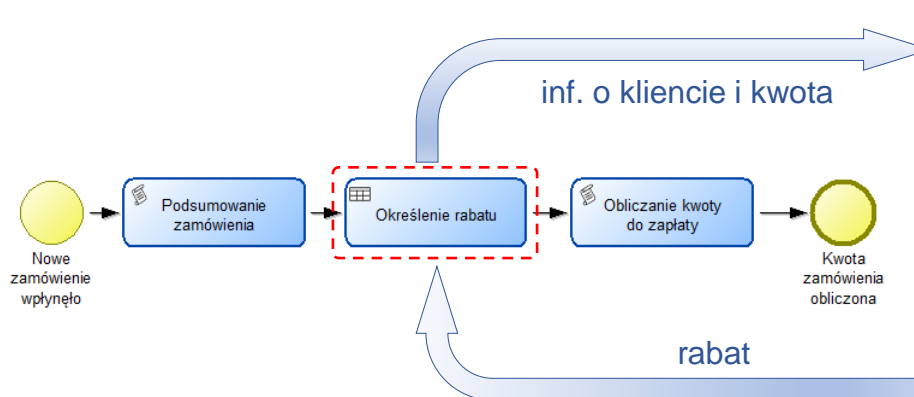
- **Diagramy DRD** (ang. *Decision Requirements Diagrams*) – pokazują sieć zależności łączącą kolejne etapy podejmowania decyzji (decyzje podrzędne);
- **Tablice decyzyjne** – opisują reguły na bazie których podejmowane są decyzje;
- **Język wyrażeń FEEL** (ang. *A Friendly Enough Expression Language*) – notacja umożliwiająca formułowanie reguł i warunków w sposób umożliwiający ich automatyczne przetwarzanie.

Oficjalna specyfikacja DMN: <https://www.omg.org/dmn/>

Obliczenie kwoty zamówienia w.1



Obliczenie kwoty zamówienia w.2 – reguły biznesowe



R	Wejście			Wyjście
	Stały klient	Współpraca	Kwota > 1000	Rabat
1	False	-	False	0%
2	False	-	True	1%
3	True	< 1	False	1%
4	True	< 1	True	2%
5	True	>= 1 and < 5	False	3%
6	True	>= 1 and < 5	True	4%
7	True	>= 5	False	5%
8	True	>= 5	True	6%



Zadanie serwisowe (ang. *service task*) reprezentuje czynność, która jest realizowana w pełni automatycznie, bez udziału człowieka, poza procesem biznesowym.

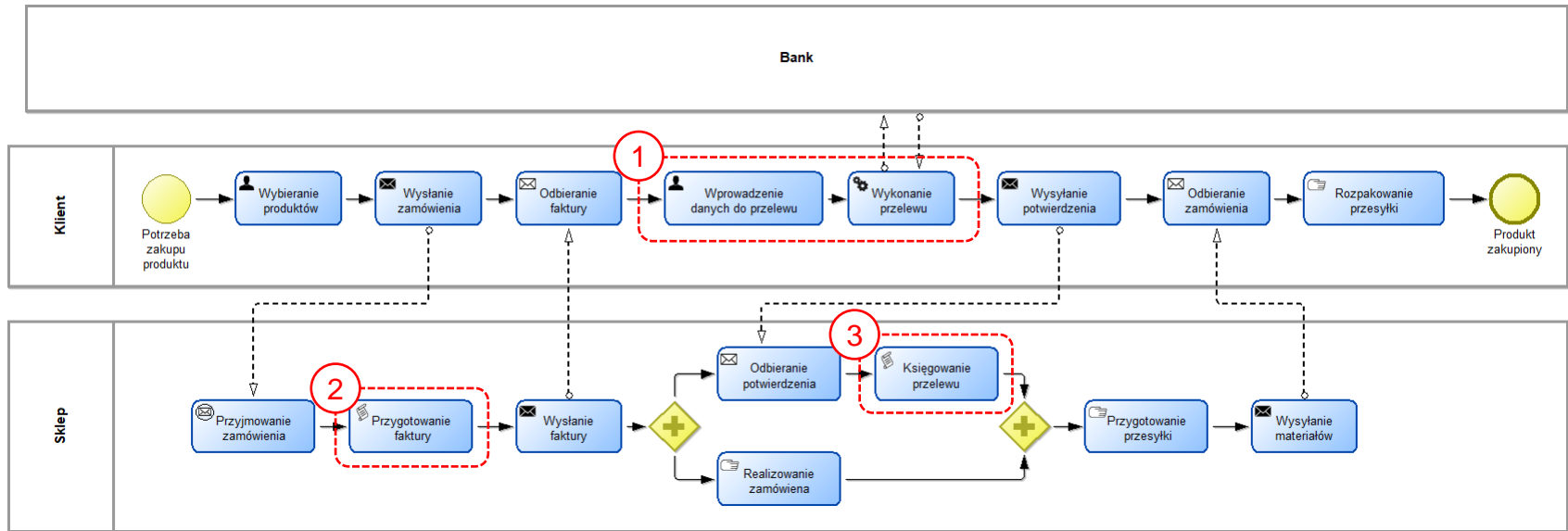
Własności zadania serwisowego

- jest identyfikowane przez ikonę z symbolem przekładni;
- w odróżnieniu od skryptu i reguły biznesowej jest wykonywane przez niezależne narzędzie, poza silnikiem procesów i reguł biznesowych (np. serwis WWW dostarczony przez zewnętrznego uczestnika, lub zautomatyzowana aplikacja uruchamiana niezależnie od procesu);
- jeżeli zadanie wykorzystuje serwis dostarczony przez innego uczestnika, może on być identyfikowany przez połączenie zadania serwisowego z basenem uczestnika przy pomocy przepływu komunikatu.

Uwaga: podobnie jak zadanie skryptowe i reguła biznesowa, serwis jest uruchamiany gdy token procesu dociera do zadania, proces jest kontynuowany (token przechodzi do następnego węzła przepływu) gdy zadanie serwisowe zostanie ukończone.



Realizacja zamówienia (modyfikacja modelu ze s.10)



1. Zadanie serwisowe: Wykonanie przelewu.

2. Zadania skryptowe: Przygotowanie faktury, Księgowanie przelewu.

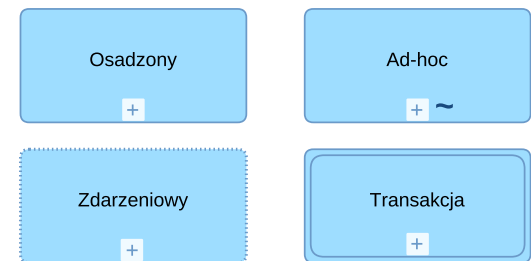
Uwaga: serwis jest zadaniem realizowanym w pełni automatycznie (bez udziału człowieka), więc przed wykonaniem tego zadania Klient musi wprowadzić odpowiednie dane (zadanie użytkownika).

Podproces jest szczególnym przypadkiem aktywności, która zawiera szczegółowo opisany sposób realizacji. Własności podprocesu:

- jest identyfikowany ikoną z "plusem" umieszczoną przy dolnej krawędzi symbolu;
- jak jak proces główny (*top-level process*) ma cechy kontenera obiektów (*flow elements container*) i może zawierać elementy przepływu (*flow elements*).
- w modelu może występować w dwóch stanach:
 - zwinięty – reprezentowany przez symbol, struktura wewnętrzna ukryta
 - rozwinięty – struktura widoczna, odseparowana od procesu głównego ramką.

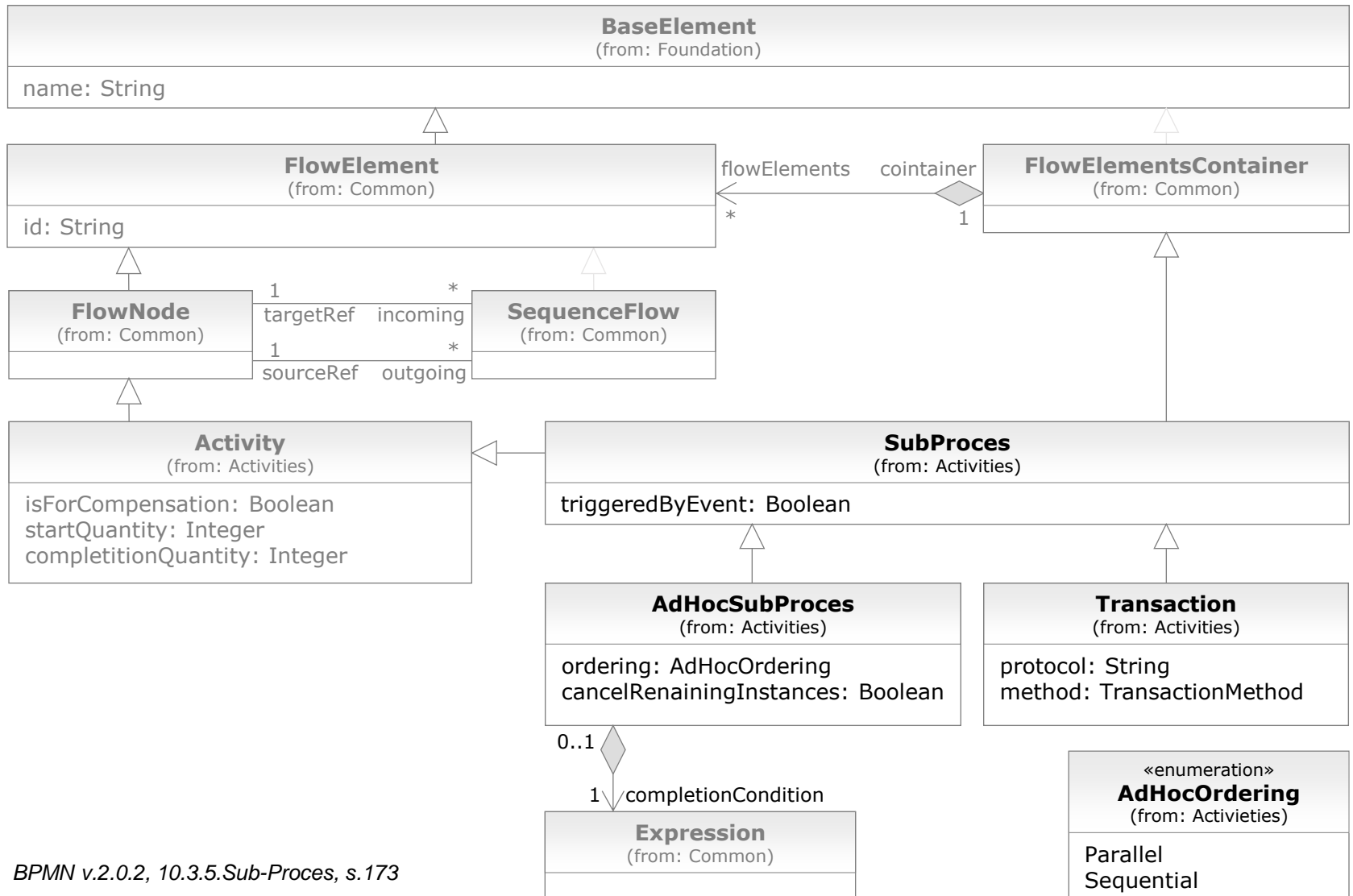
Typy podprocesów w BPMN 2.0

- Osadzony/wbudowany (*ang. embedded sub-process*);
- Ad-hoc (*ang. ad-hoc process*);
- Zdarzeniowy (*ang. event sub-process*);
- Transakcja (*ang. transaction*).



Podprocesy **osadzone**, **ad-hoc** oraz **transakcje** występują w sekwencji przepływu procesu nadrzędnego i są uruchamiane przez token procesu, podprocesy **zdarzeniowe** występują poza sekwencją procesu nadrzędnego i są uruchamiane przez wyzwalacz zdarzenia.

Podprocesy – diagram klas



BPMN v.2.0.2, 10.3.5.Sub-Proces, s.173

Atrybuty i powiązania podprocesów

Klasa **SubProces**: dziedziczy atrybuty **FlowElementsContainer** (jest kontenerem, może zawierać elementy przepływu) oraz **Activity** (jest węzłem przepływu, może mieć wejściowy i wyjściowy przepływ sekwencyjny), dodatkowo:

- **triggeredByEvent**: Boolean – określa czy podproces jest podprocesem zdarzeniowym.

Klasa **AdHocSubProces**

- **completionCondition**: Expression – określa warunek zakończenia procesu,
- **ordering**: AdHocOrdering – określa czy proces może być wykonywany równoległe (Parallel), czy musi być wykonywany sekwencyjnie (Sequential),
- **cancelRemainingInstances**: Boolean – ma zastosowanie gdy ordering = Paralel, określa czy uruchomione instancje są przerywane gdy completionCondition ma wartość True.

Klasa **Transaction**

- **protocol**: String – określa nazwę protokołu, który kontroluje przebieg transakcji,
- **method**: TransactionMethod – odwołanie URI do opisu protokołu używanego przez transakcję (BPMN 2.0 definiuje dopuszczalne protokoły dla transakcji – s.178).

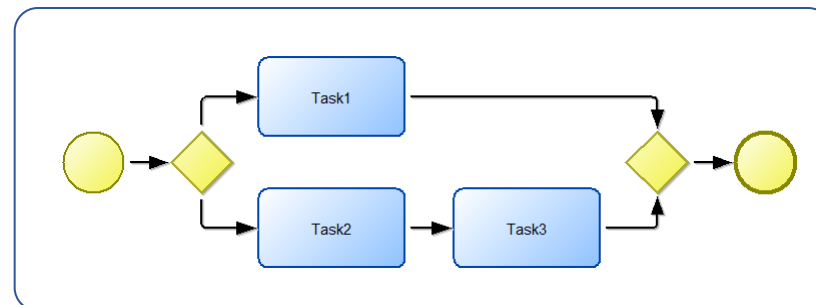


Podproces osadzony (krótko: podproces)

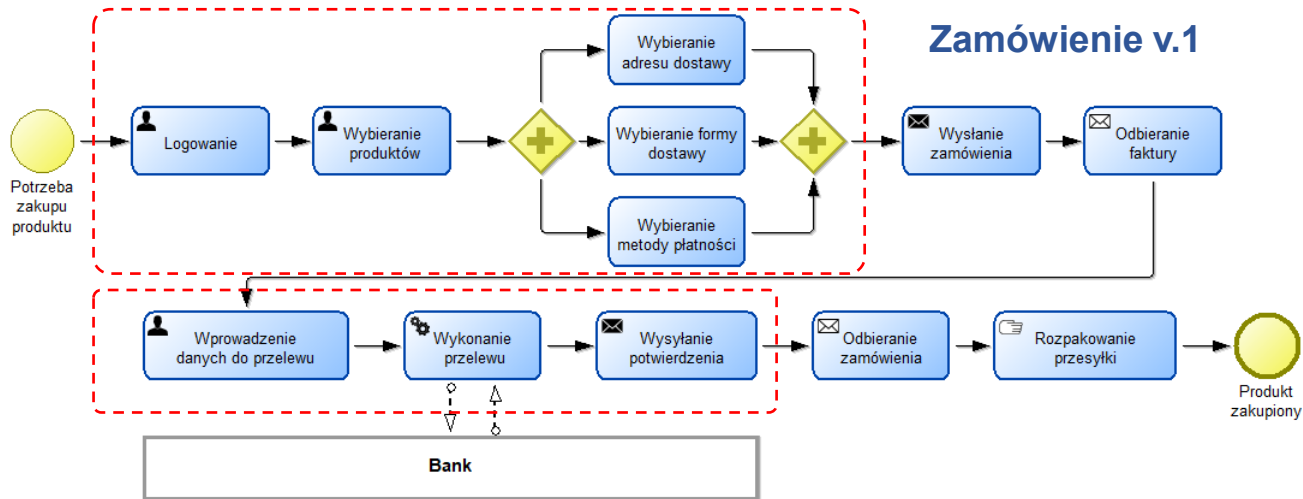
- pozwala na zdefiniowanie złożonej aktywności, która może być użyta wielokrotnie;
- upraszcza strukturę i poprawia czytelność procesu głównego;
- występuje w sekwencji przepływu procesu nadrzędnego;
- nie ma własnej instancji, jest uruchamiany przez token procesu nadrzędnego;
- rozpoczyna się od pojedynczego zdarzenia początkowego bez wyzwalacza;
- jeżeli zdarzenie początkowe nie występuje wszystkie sekwencje podprocesu otrzymują token równocześnie i są wykonywane równolegle.

Symbol podprocesów osadzonych

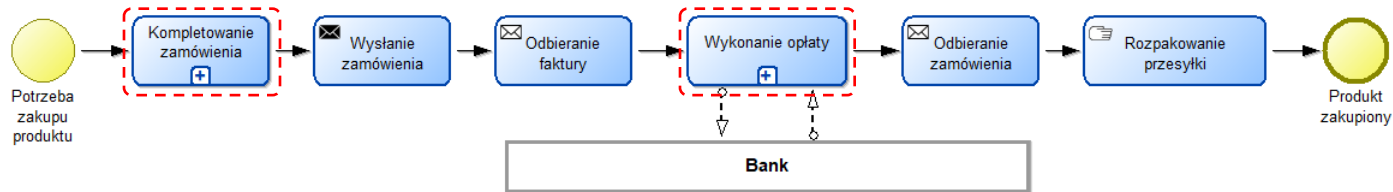
- zwinięty – symbol aktywności z dodatkowym znakiem "plus";
- rozwinięty – prostokąt z pojedynczą, cienką ramką bez znaku "plus".



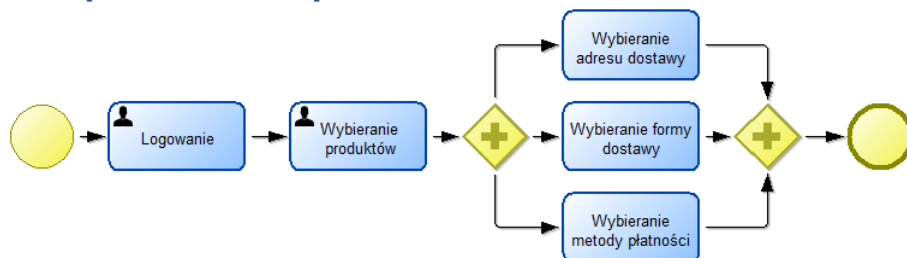
Podprocesy osadzone – przykład 1.



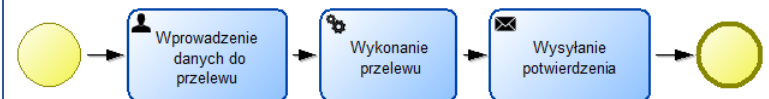
Zamówienie v.2 – podprocesy osadzone



Podproces: Kompletowanie zamówienia

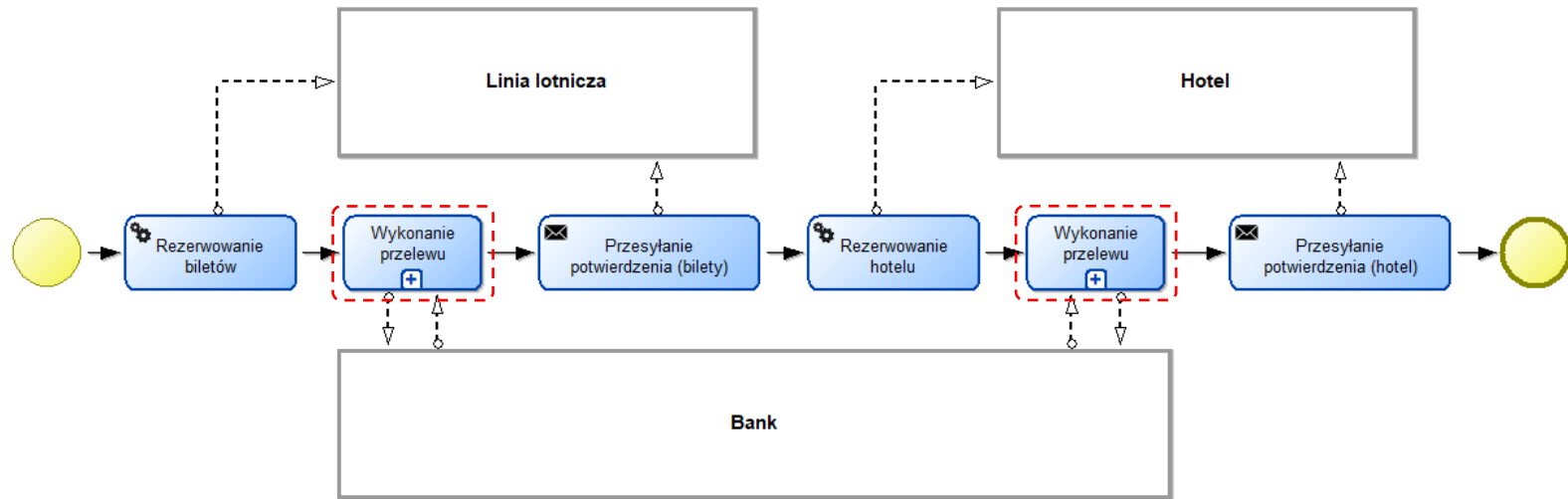


Podproces: Wykonanie opłaty





Planowanie wyjazdu



Podproces: Wykonanie przelewu

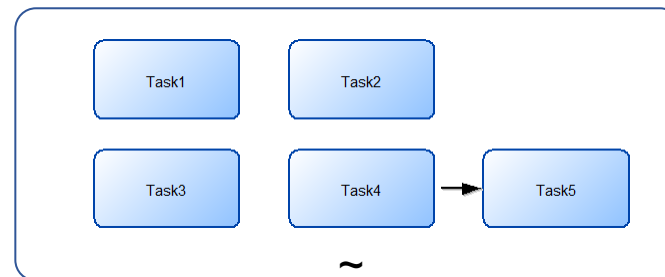
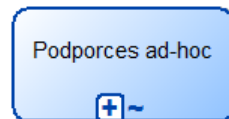


Po rezerwacji biletów lotniczych oraz hotelu należy wykonać przelew (dwie różne kwoty dla dwóch różnych odbiorców). Podproces osadzony "Przelew" modeluje operację, która powtarza się dwukrotnie.



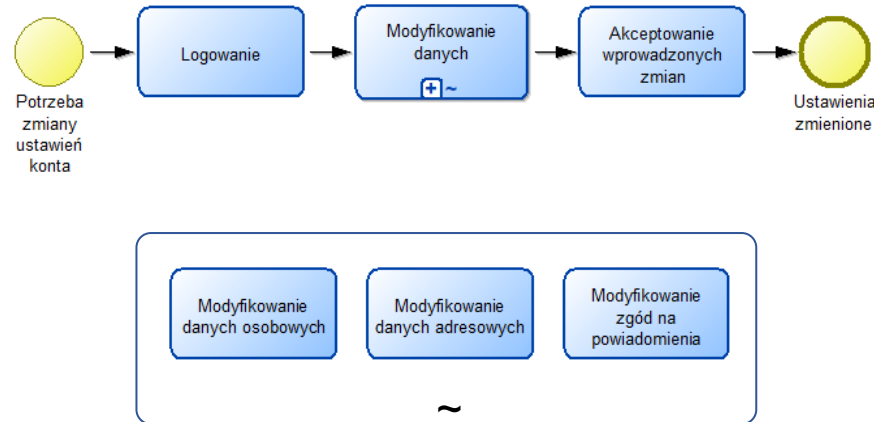
Podproces ad-hoc

- jest identyfikowany symbolem z "plusem" z dodatkowym znakiem "tylda" (~);
- zawiera zestaw aktywności, które mogą być wykonane w dowolnej kolejności, lub pominięte, wybór jest dokonywany podczas aktywowania podprocesu (gdy token osiąga podproces).
- musi zawierać aktywności, przy każdym wykonaniu podprocesu każda z nich może być wykonana dowolną liczbą razy (dopuszczalne nie wykonanie żadnej);
- dopuszczalne jest użycie obiektów danych, bramek, zdarzeń pośrednich, przepływu komunikatów i przepływów sekwencyjnych;
- elementy połączone przepływem sekwencyjnym muszą być wykonane w ustalonej kolejności;
- niedopuszczalne są zdarzenia początkowe i końcowe.





Zmiana ustawień konta w sklepie internetowym



Użytkownik chce zmienić ustawienia swojego konta. Zakres oraz kolejność wykonywanych operacji zależą od potrzeb użytkownika i mogą być wybierane dowolnie (w każdej instancji procesu inaczej).



Aktywność wywołania (ang call activity)

- reprezentuje punkt, w którym proces uruchamia zadanie lub proces globalny;
- jest identyfikowana przez prostokąt z pogrubioną krawędzią;
- w przypadku wywołania podprocesu globalnego ma dodatkową ikonę "plus";
- pozwała na przekazywanie parametrów wejściowych i wyjściowych;
- umożliwia modularyzację modelu poprzez korzystanie z elementów wielokrotnego użytku o charakterze uniwersalnym.

Operacje wykonywane po osiągnięciu aktywności wywołania przez token:

- utworzenie nowej instancji uruchamianego procesu globalnego;
- przekazanie danych wejściowych do uruchamianego procesu globalnego (proces wywoływany nie ma dostępu do danych procesu wywołującego);
- wykonanie sekwencji przepływu wywoływanego procesu globalnego;
- przekazanie danych wyjściowych do procesu wywołującego.

Uwaga: przepływ sekwencyjny procesu wywołującego jest kontynuowany (token opuszcza aktywność wywołania) po zakończeniu zadania/procesu wywoływanego.



Element wywoływany (ang. callable element) to składnik modelu, który może być używany wielokrotnie przez różne procesy poprzez aktywności wywołania.

Zadanie globalne (ang. global task) jest aktywnością atomową wielokrotnego użytku. Dopuszczalne typy zadań:

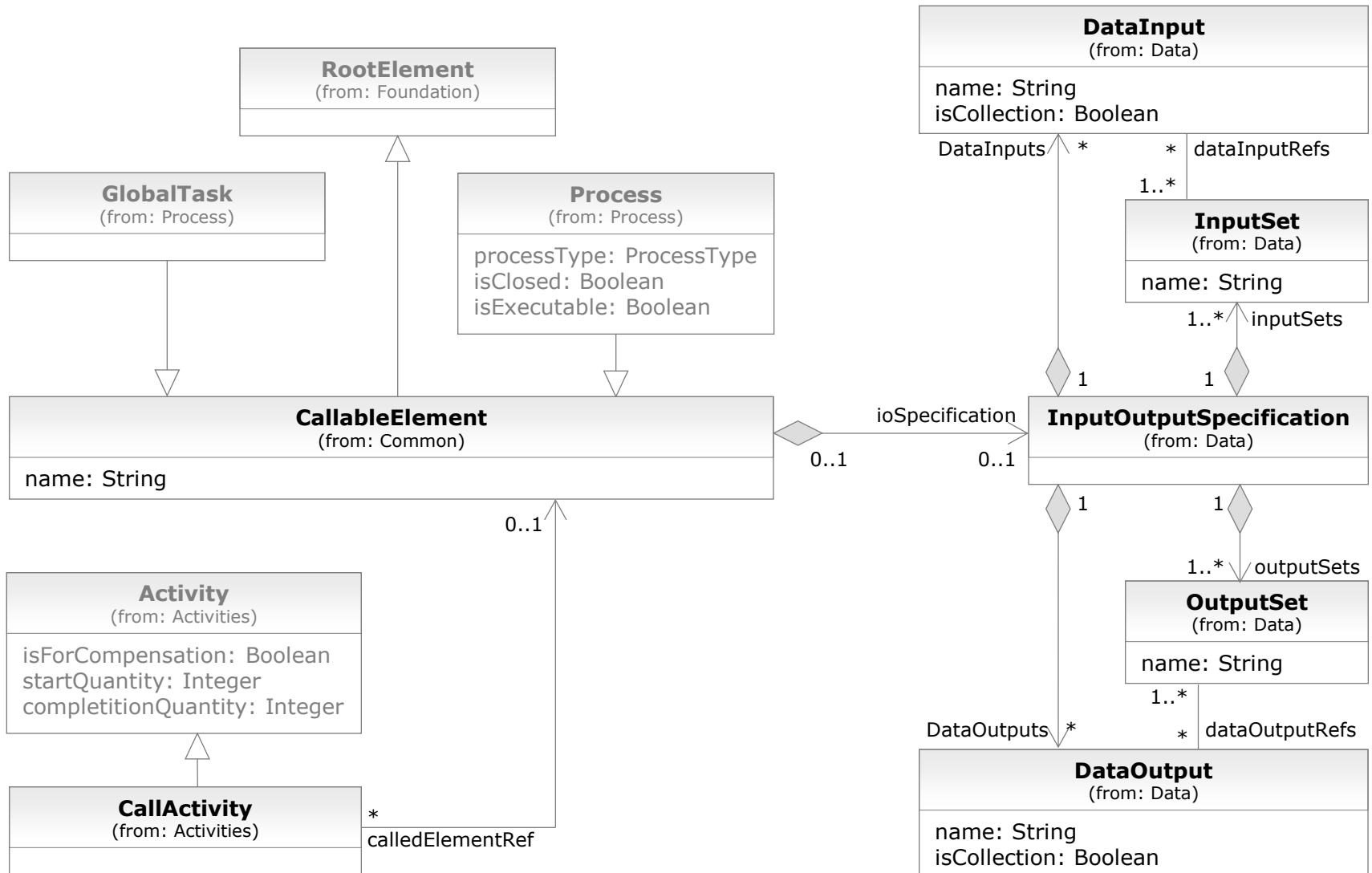
- globalne zadanie użytkownika (ang. global user task);
- globalne zadanie manualne (ang. global manual task);
- globalny skrypt (ang. global script task);
- globalna reguła biznesowa (ang. global business rule task).

Proces globalny (ang. global process) jest kompletnym procesem, który może być uruchamiany niezależnie lub wywołany przez inny proces. Cechy procesów globalnych:

- muszą mieć zdarzenie początkowe bez wyzwalacza (dopuszczalne dodatkowe zdarzenia początkowe);
- nie mają dostępu do danych procesu nadrzędnego;
- komunikują się z procesem nadrzędnym przez parametry wejścia/wyjścia;
- mogą mieć basen, podział na tory i realizować komunikację z innymi uczestnikami.

Uwaga: W odróżnieniu od podprocesu osadzonego proces globalny nie stanowi części procesu nadrzędnego (wywołującego), nie ma dostępu do jego danych i może być uruchamiany samodzielnie.

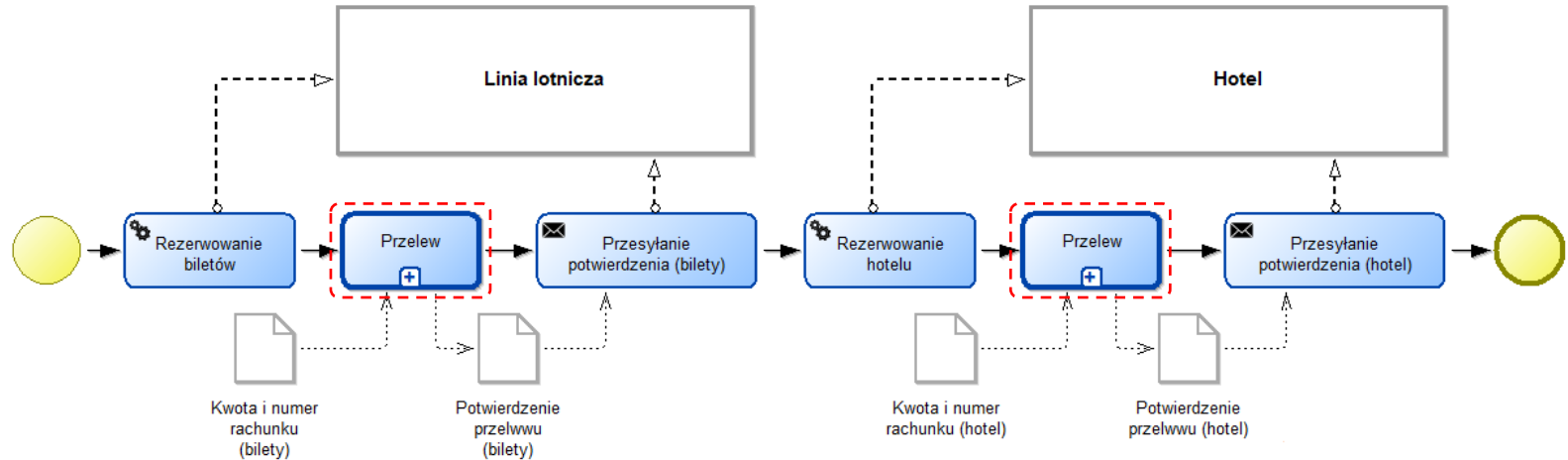
CallableElement i CallActivity – diagram klas



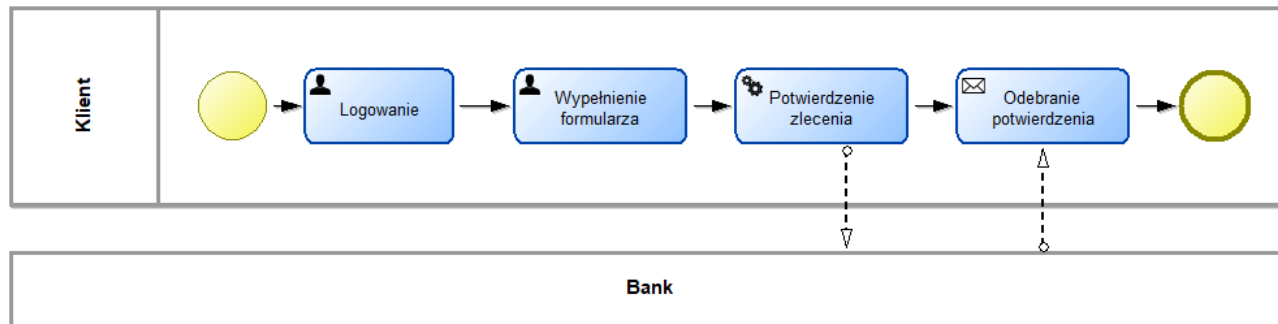
BPMN v.2.0.2, 10.3.6.Call Activity, s.182



Planowanie wyjazdu wersja 2. (s.21)



Proces globalny Przelew



Proces „Planowanie wyjazdu” **wywołuje** proces globalny „Przelew” za każdym razem **przekazując dane odbiorcy i kwotę**, jako wynik otrzymuje potwierdzenie przelewu.