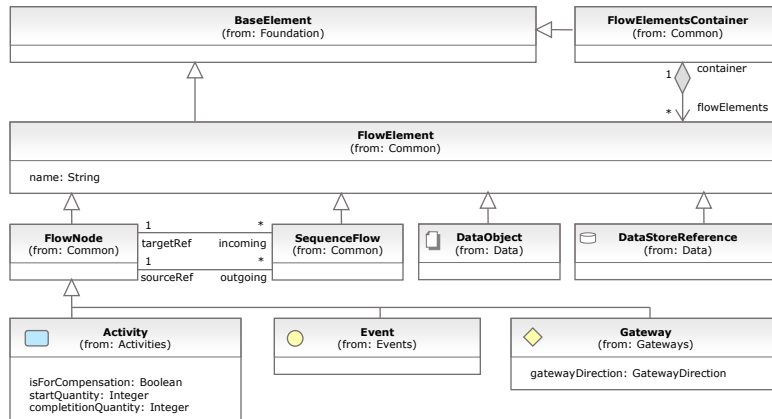


Modelowanie procesów biznesowych



Metamodel BPMN

elementy notacji UML

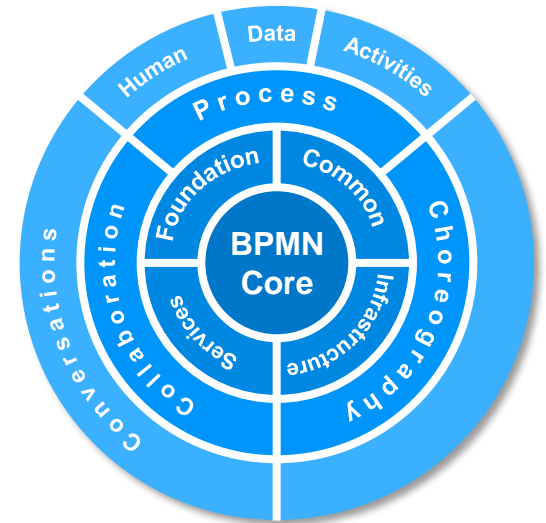
typy procesów

wybrane składniki pakietów:

Common, Process i Collaboration

Metamodel – zbiór koncepcji opisujących elementy notacji i zasady ich użycia (semantykę i syntaktykę).

Specyfikacja BPMN opiera się na koncepcji pakietów zorganizowanych w warstwy. Kolejna warstwa wprowadza nowe elementy używając pojęć zdefiniowanych w pakietach należących do warstw leżących głębiej. Wszystkie elementy notacji bazują na składnikach zdefiniowanych w rdzeniu (BPMN Core).



Pakiety BPMN Core

- **Foundation** – zbiór elementów używanych przez pozostałe pakiety rdzenia;
- **Common Elements** – elementy notacji, które mogą być używane różnych typach diagramów (proces, choreografia, kooperacja);
- **Services** – elementy niezbędne do modelowania serwisów, interfejsów i operacji (składniki używane w modelach wykonywalnych);
- **Infrastructure** – definicja składni, zakresu widoczności i przestrzeni nazw wszystkich elementów notacji, zasady odwołania do elementów zewnętrznych (spoza BPMN).

Metamodel opisuje elementy BPMN używając notacji UML.

Elementy notacji UML – diagram klas

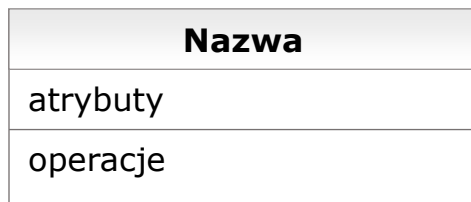
Obiekt – abstrakcyjny byt reprezentujący pewną rzecz lub pojęcie obserwowane w modelowanym systemie. Cechy obiektu:

- tożsamość – cecha umożliwiająca identyfikację,
- stan – aktualna wartość danych składowych (atrybutów),
- zachowanie – zestaw operacji możliwych do wykonania na danych składowych.

Klasa – wzorzec definiujący strukturę pewnej grupy obiektów o takiej samej budowie (atrybutach) i zachowaniu (dozwolonych operacjach).

Diagram klas – podstawowy diagram notacji UML opisujący strukturę modelowanego systemu w postaci hierarchii klas i zależności między nimi.

Wybrane elementy diagramu klas



klasa



*uogólnienie
(dziedziczenie)*



*powiązanie
(asocjacja)*



kompozycja

Elementy notacji UML – klasa

Nazwa klasy
atrybuty
operacje

Nazwa klasy określa nazwę opisywanej grupy obiektów,
atrybuty charakteryzują dane składowe obiektów klasy,
operacje opisują zachowanie obiektów.

Uwaga: obiektami opisywanymi przez metamodel BPMN są symbole notacji (aktywności, bramki, zdarzenia, itp.), z punktu widzenia semantyki istotny jest jedynie stan obiektów (atrybuty) stąd ich zachowanie (metody) nie będzie analizowane.

Przykłady

Prostokąt
wierzchołek: Punkt kolor: Liczba
Rysuj() Pole(): Liczba Obwód(): Liczba

Punkt
x: Liczba y: Liczba
Rysuj()

Osoba
nazwisko: Tekst imię: Tekst dataUrodzenia: Data adresZamieszkania: Adres

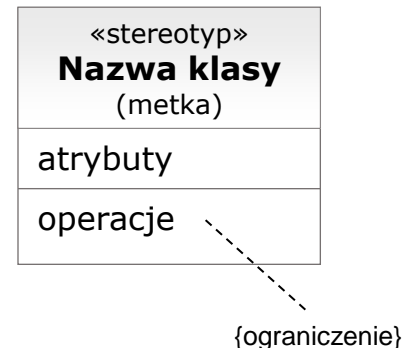
Adres
kod: Tekst miasto: Tekst ulica: Tekst nrDomu: Tekst

Data
dzień: Liczba miesiąc: Liczba rok: Liczba

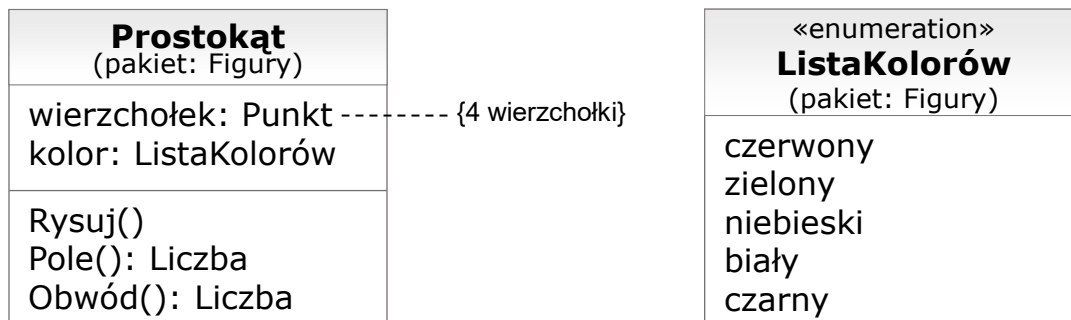
Elementy notacji UML – mechanizmy rozszerzania

UML jest notacją otwartą, udostępnia mechanizmy **rozszerzania**:

- **stereotypy** – rozszerzają słownictwo, pozwalają tworzyć nowe bloki konstrukcyjne specyficzne dla danego modelu;
- **metki** – rozszerzają listę właściwości bloku konstrukcyjnego, uzupełniają go o nowe informacje;
- **ograniczenia** – rozszerzają znaczenie bloku konstrukcyjnego, ustalają reguły określające zasady funkcjonowania bloku.



Przykłady



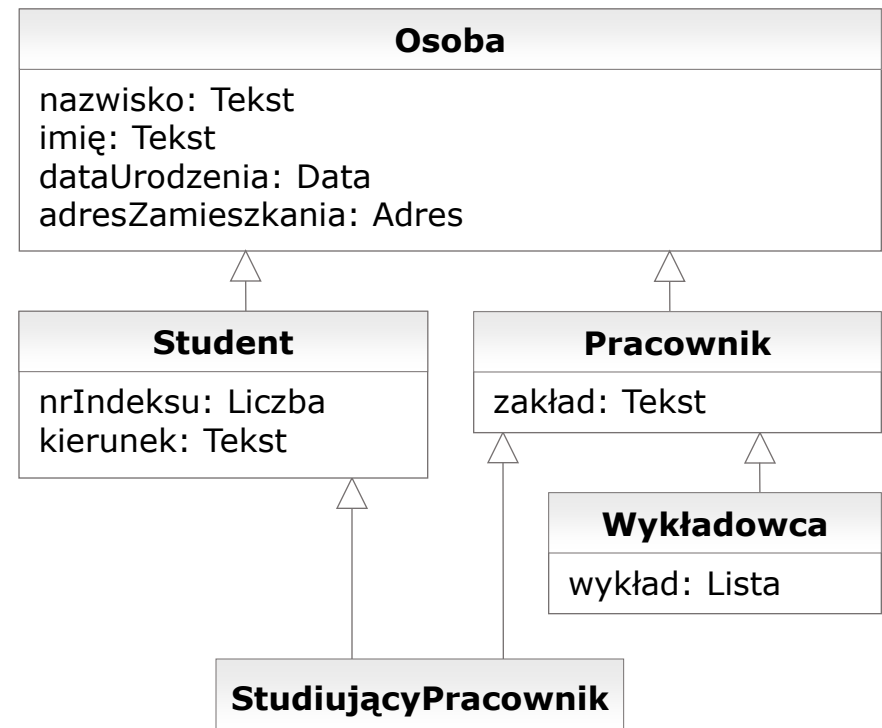
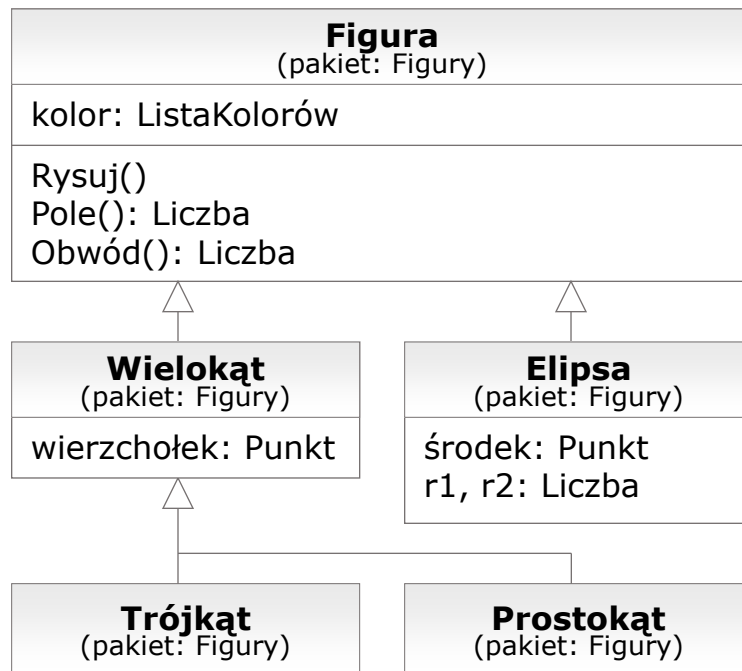
Klasa **ListaKolorów** wprowadza zestaw identyfikatorów określających kolory, ograniczenie ustala dozwoloną liczbę wierzchołków, metka określa pakiet, w którym zdefiniowano klasy.

Uwaga: «enumeration» jest standardowym stereotypem UML określającym typ wyliczeniowy


Elementy notacji UML – uogólnienie

↑ **Uogólnienie** (dziedziczenie) opisuje związek pomiędzy klasą ogólną (przodkiem) a jej szczególnym przypadkiem (potomkiem). Potomek dziedziczy (przejmuje) wszystkie cechy przodka, a w szczególności jego atrybuty i operacje, dodatkowo może wprowadzić własne. Jeżeli klasa ma jednego przodka występuje dziedziczenie pojedyncze, jeżeli wielu dziedziczenie wielokrotne.

Przykłady



Elementy notacji UML – asocjacja

 **Powiązanie** (asocjacja) to związek strukturalny, który opisuje połączenie pomiędzy obiektami. Asocjacja zwykła umożliwia przechodzenie od obiektów jednej klasy do drugiej i odwrotnie (jest dwukierunkowa). Asocjacja skierowana pozwala przejść tylko w jedną stronę (jest jednokierunkowa).

Przykłady



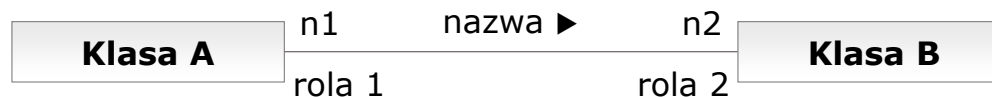
Każdy Student studiuje na pewnym Wydziale. Znajac Studenta można określić jego Wydział, znając Wydział można określić listę Studentów.



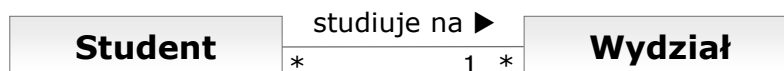
Każdy Użytkownik ma swoje Hasło. Znajac obiekt Użytkownik można określić Hasło, ale na podstawie Hasła nie można znaleźć Użytkownika.

Atrybuty powiązań

- Nazwa – określa istotę związku, może posiadać kierunek odczytu;
- Role – opisują funkcje, które pełnią klasy w konkretnym powiązaniu;
- Liczebność – określa liczbę obiektów (min-max), które mogą wystąpić w powiązaniu.



Przykład



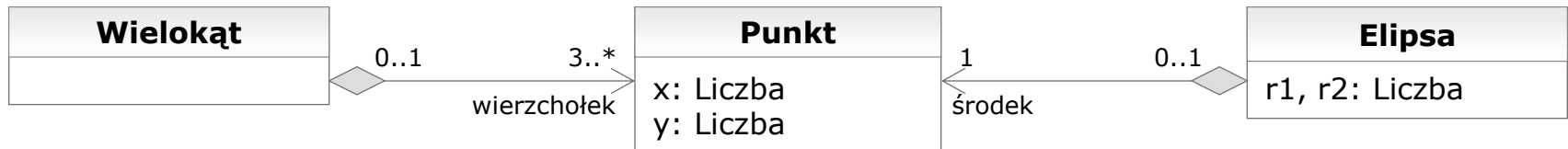
Student studiuje na co najmniej jednym Wydziale. Wydział ma dowolną liczbę studentów, może nie mieć żadnego (* jest skrótem dla 0..*).

Elementy notacji UML – kompozycja



Kompozycja jest szczególnym przypadkiem powiązania opisującym zależność całość-część. Część może należeć tylko do jednej całości, jest jej nieodłącznym składnikiem, a jej istnienie zależy od istnienia całości (usunięcie całości pociąga za sobą usunięcie wszystkich części).

Przykłady

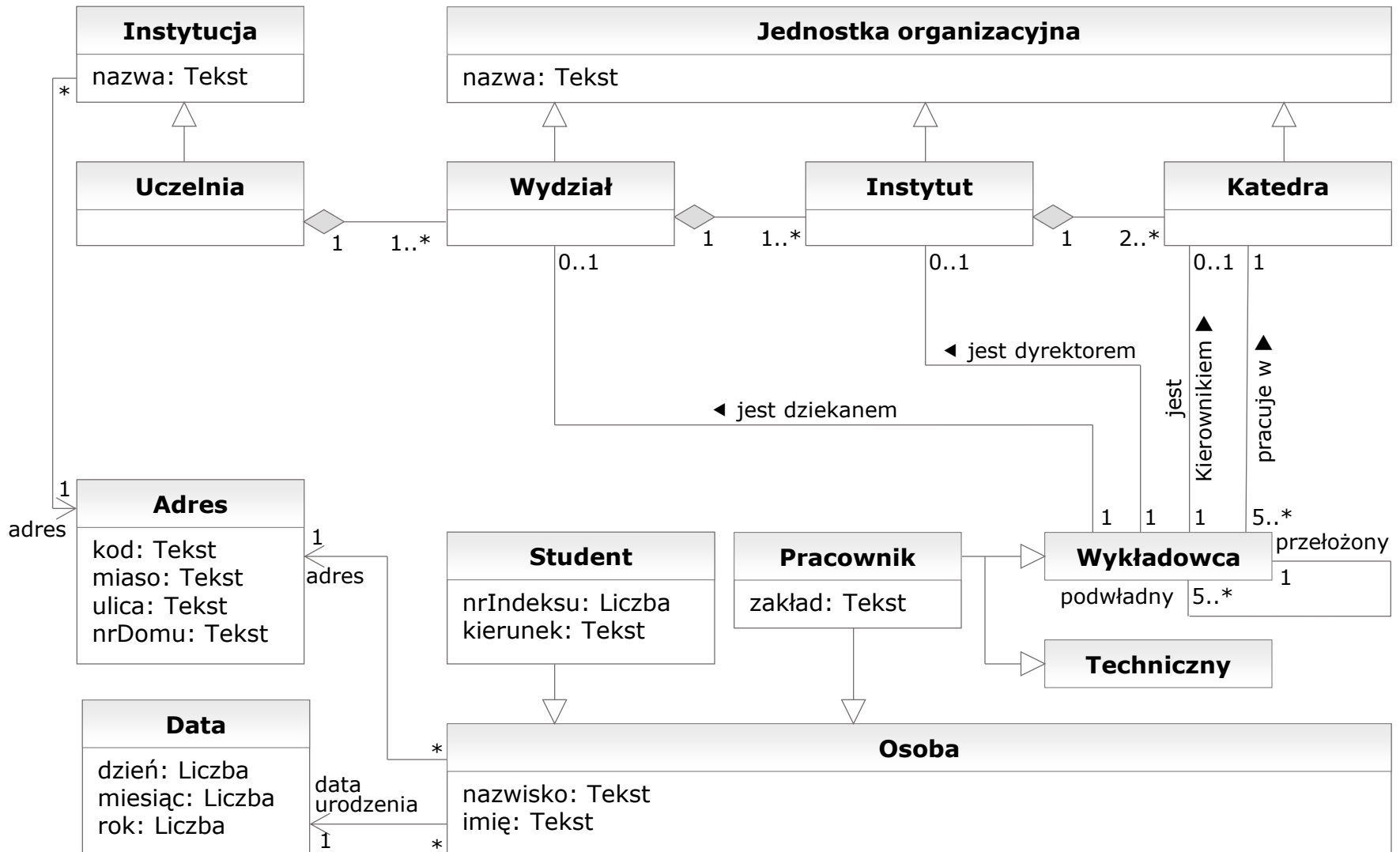


- Wielokąt zawiera co najmniej trzy Punkty, które są jego wierzchołkami (te Punkty są częścią Wielokąta).
- Elipsa zawiera dokładnie jeden Punkt, który jest jej środkiem (ten Punkt jest częścią Elipsy).
- Punkt może być częścią co najwyżej jednego Wielokąta lub Elipsy (istnieją Punkty, które nie są częściami tych figur).

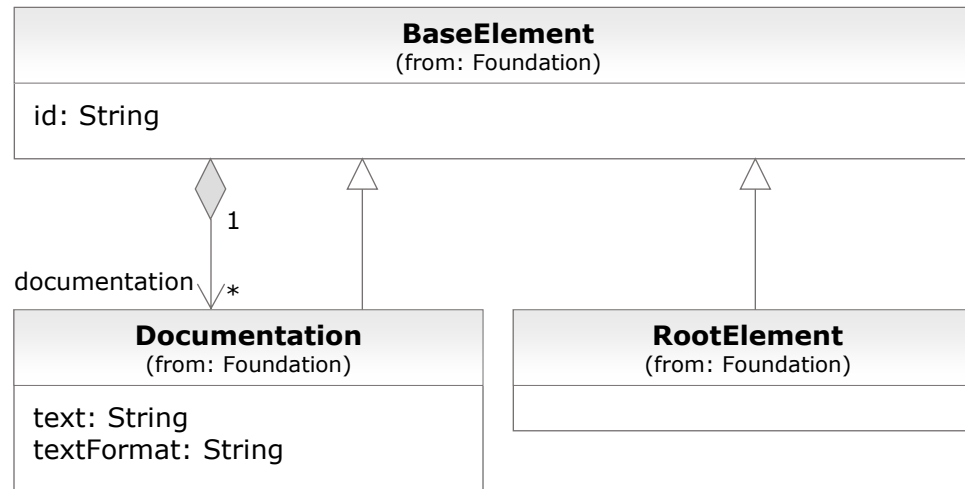


- Zakład jest częścią jednego Instytutu, Instytut składa się z co najmniej dwóch Zakładów.
- Instytut jest częścią jednego Wydziału, Wydział składa się z co najmniej jednego Instytutu.
- Wydział częścią jednej Uczelni, Uczelnia składa się z co najmniej jednego Wydziału.

Elementy notacji UML – przykład diagramu klas



BPMN Core – pakiet Foundation

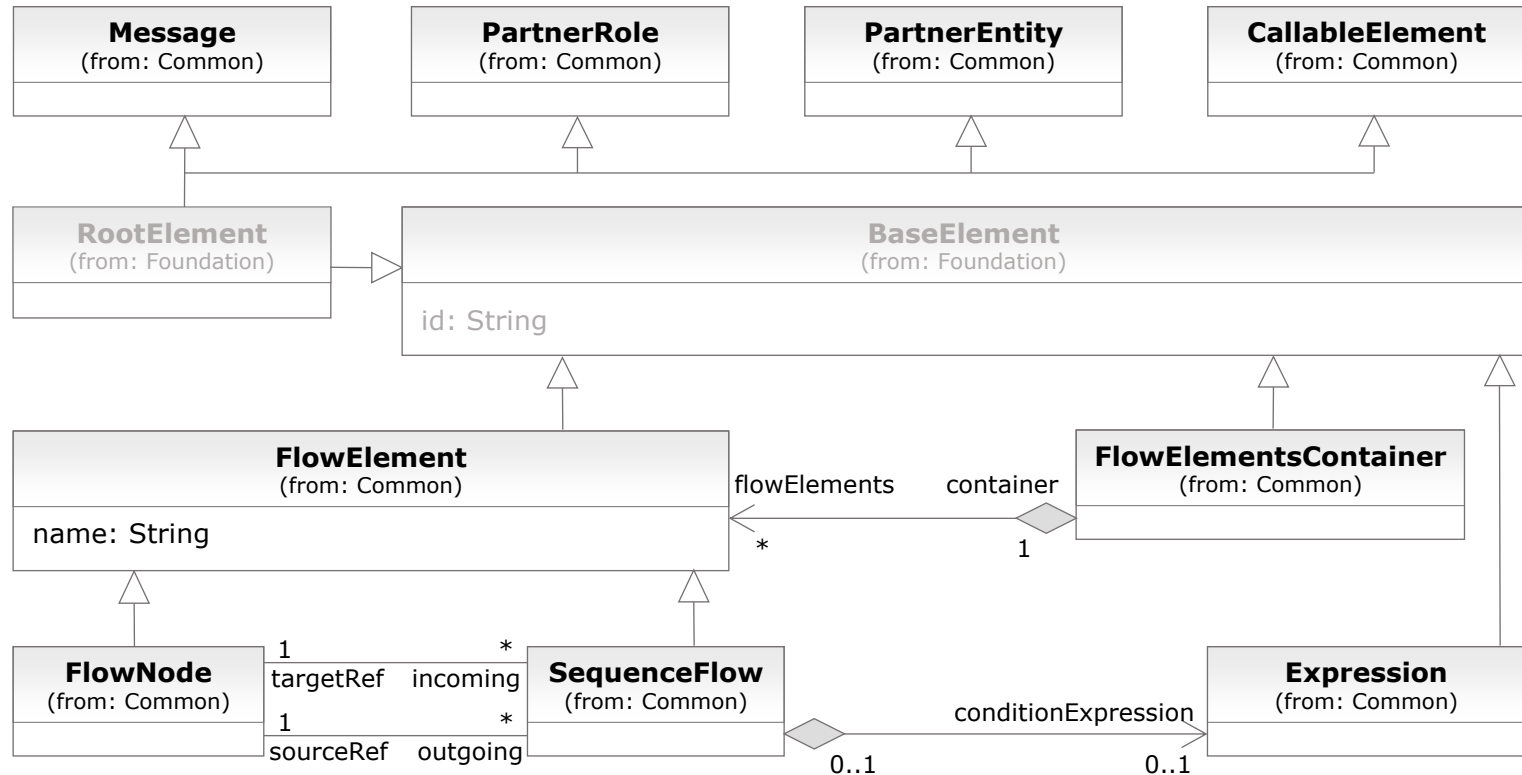


BPMN v.2.0.2 8.3.Foundation, s.53

Klasy pakietu Foundation

- **BaseElement** jest przodkiem większości elementów notacji, wprowadza atrybuty, które dziedziczą elementy potomne (każda klasa potomna posiada te atrybuty):
 - **id**: String – unikalny identyfikator elementu (String – typ tekstowy);
 - **documentation**: Documentation [0..*] – opis elementu.
- **Documentation** jest klasą reprezentującą dokumentację każdego elementu notacji.
- **RootElement** jest przodkiem elementów o specyficznym cyklu życia, które nie są usuwane wraz z elementami w nich zawartymi (np. Proces, Kooperacja).

BPMN Core – pakiet Common Elements



BPMN v.2.0.2, 8.4.Common Elements, s.64

Klasy pakietu Common Elements

- **FlowElement** – przodek wszystkich elementów przepływu (węzły, przepływy sekwencyjne, obiekty danych, s.5-19),
- **FlowNode** – szczególny przypadek **FlowElement**, przodek wszystkich węzłów w diagramach (aktywności, bramki, zdarzenia),
- **SequenceFlow** – szczególny przypadek, reprezentuje przepływy sekwencyjne,
- **FlowElementContainer** – przodek klas reprezentujących diagramy BPMN,
- **Expression** – warunek wyrażony w języku naturalnym (nie jest wykonywalny),
- **Message** – zawartość komunikatów przesyłanych pomiędzy uczestnikami (s.5-27),
- **PartnerRole** – rola uczestnika procesu (s.5-23),
- **PartnerEntity** – typ uczestnika procesu (s.5-23),
- **CollableElement** – przodek elementów BPMN, które są współużywalne w różnych procesach (mogą być współużywalne przez różne procesy).

Podstawowe elementy BPMN

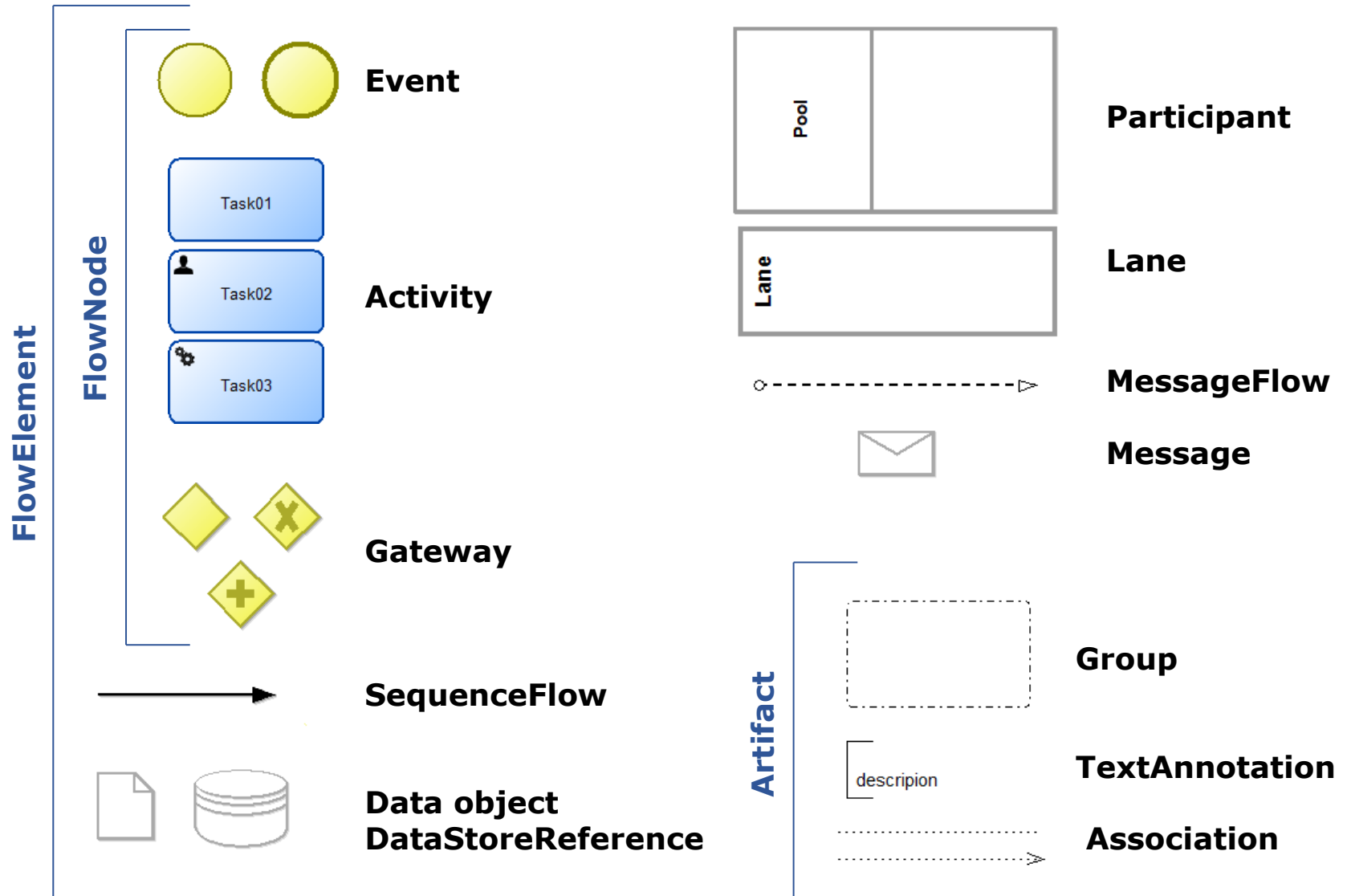
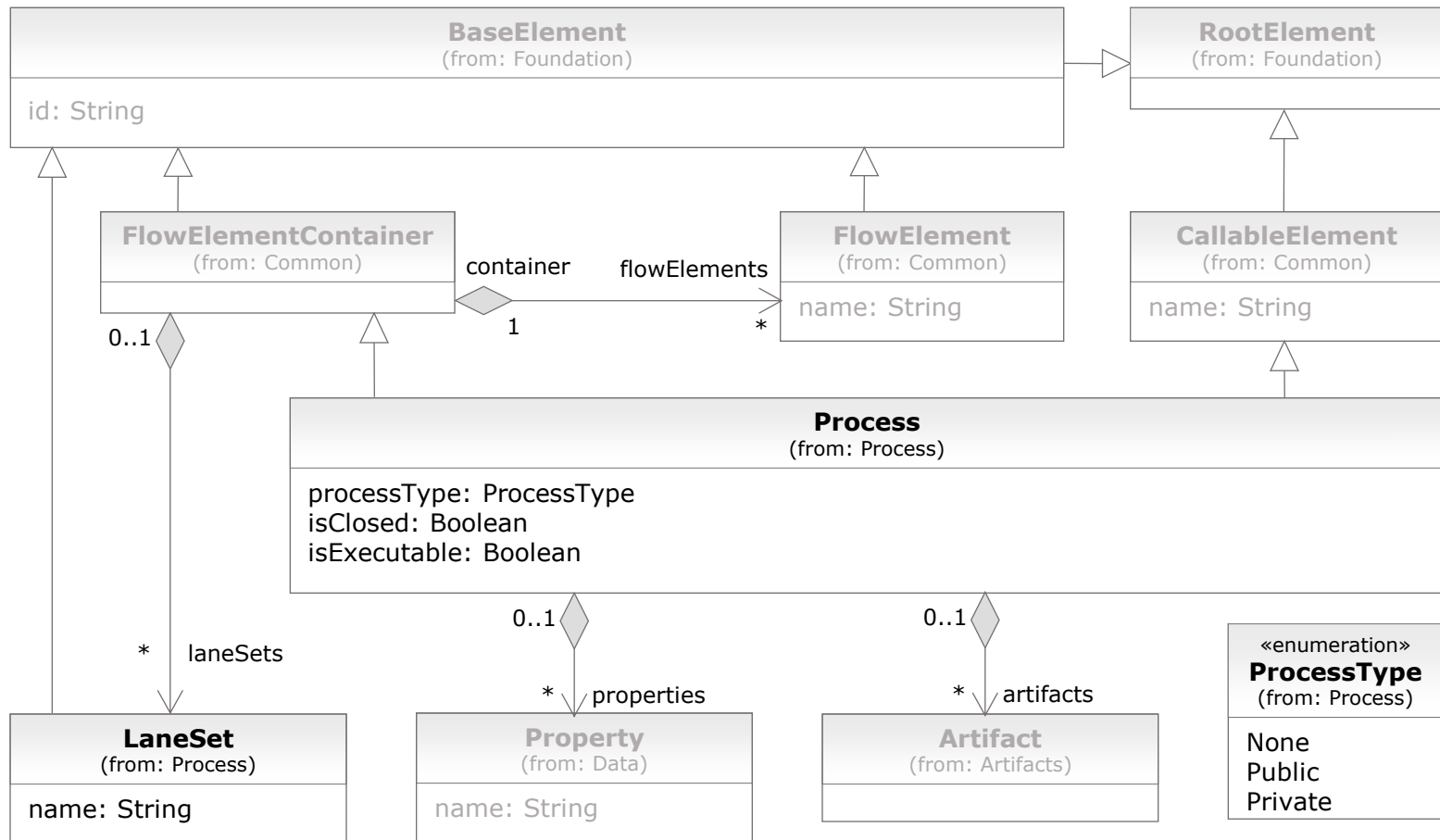


Diagram procesu – klasa Process



BPMN v.2.0.2, 10.1.Process-General, s.144-145

Atrybuty i powiązania klasy Process

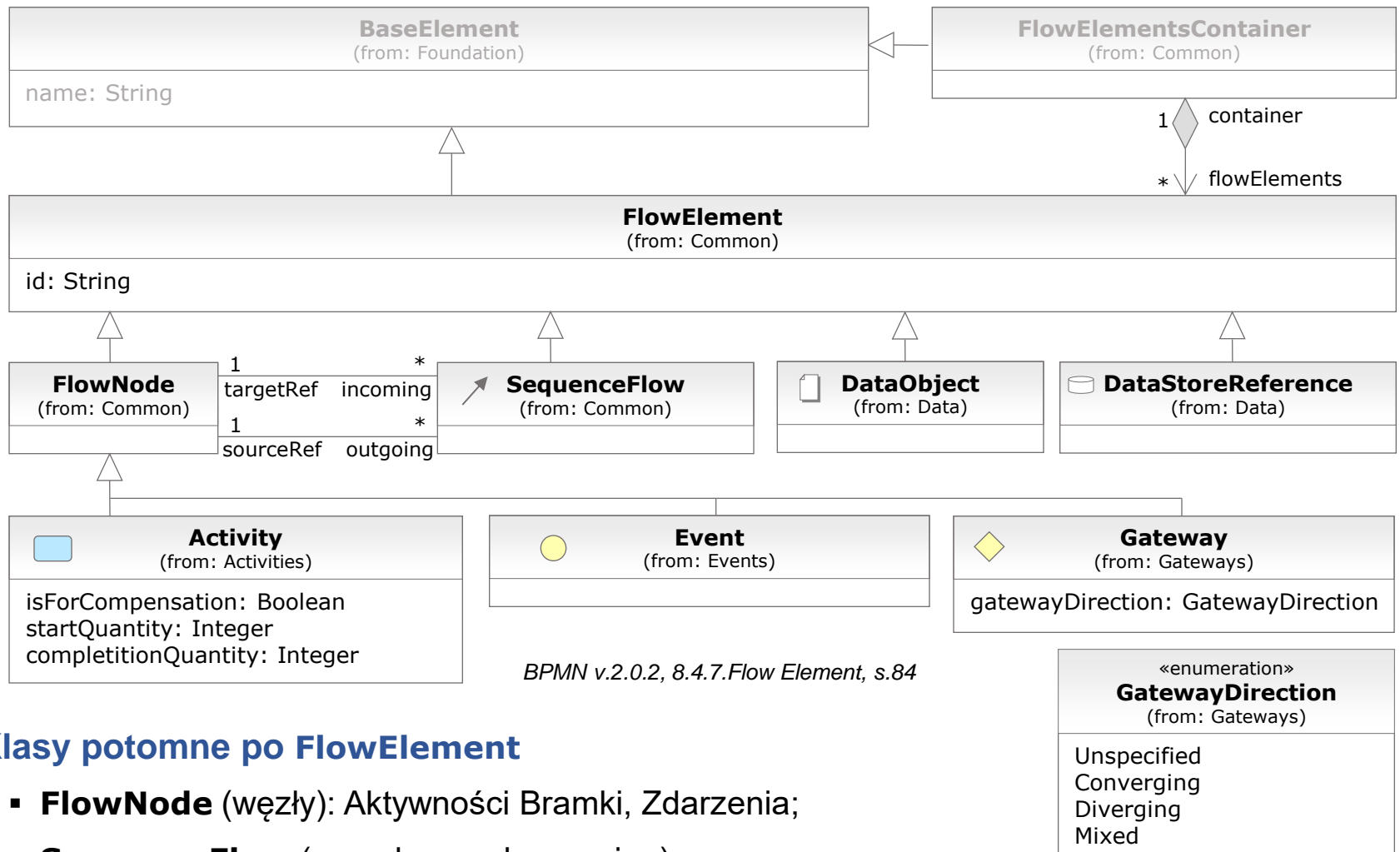
Klasy bazowe Process

- **FlowElementContainer** – przodek klas reprezentujących diagramy BPMN, zawiera dowolną liczbę elementów przepływu (**FlowElement**) oraz grupy Torów (**LaneSet**, s.5-25),
- **CallableElement** – przodek elementów BPMN, które są współużywalne w różnych procesach (mogą być współużywane przez różne procesy).

Atrybuty klasy Process

- **processType**: ProcessType – typ procesu (s.5-16),
- **isExecutable**: Boolean – określa czy proces jest wykonywalny (s.5-16),
- **isClosed**: Boolean – określa możliwość wystąpienia interakcji z innymi procesami,
- **artifacts**: Artifact [0..*] – lista artefaktów powiązanych z procesem (s.5-20),
- **properties**: Property [0..*] – lista własności procesu (BPMN dopuszcza możliwość definiowania dowolnych własności widocznych dla elementów procesu),
- atrybuty odziedziczone z **FlowElementContainer**
- **flowElements**: FlowElement [0..*] – lista elementów przepływu związanych z procesem (s.5-19)
- **laneSets**: LaneSet [0..*] – grupy Torów (**Lane**) związanych z procesem (s.5-25).

Elementy przepływu – klasa FlowElements



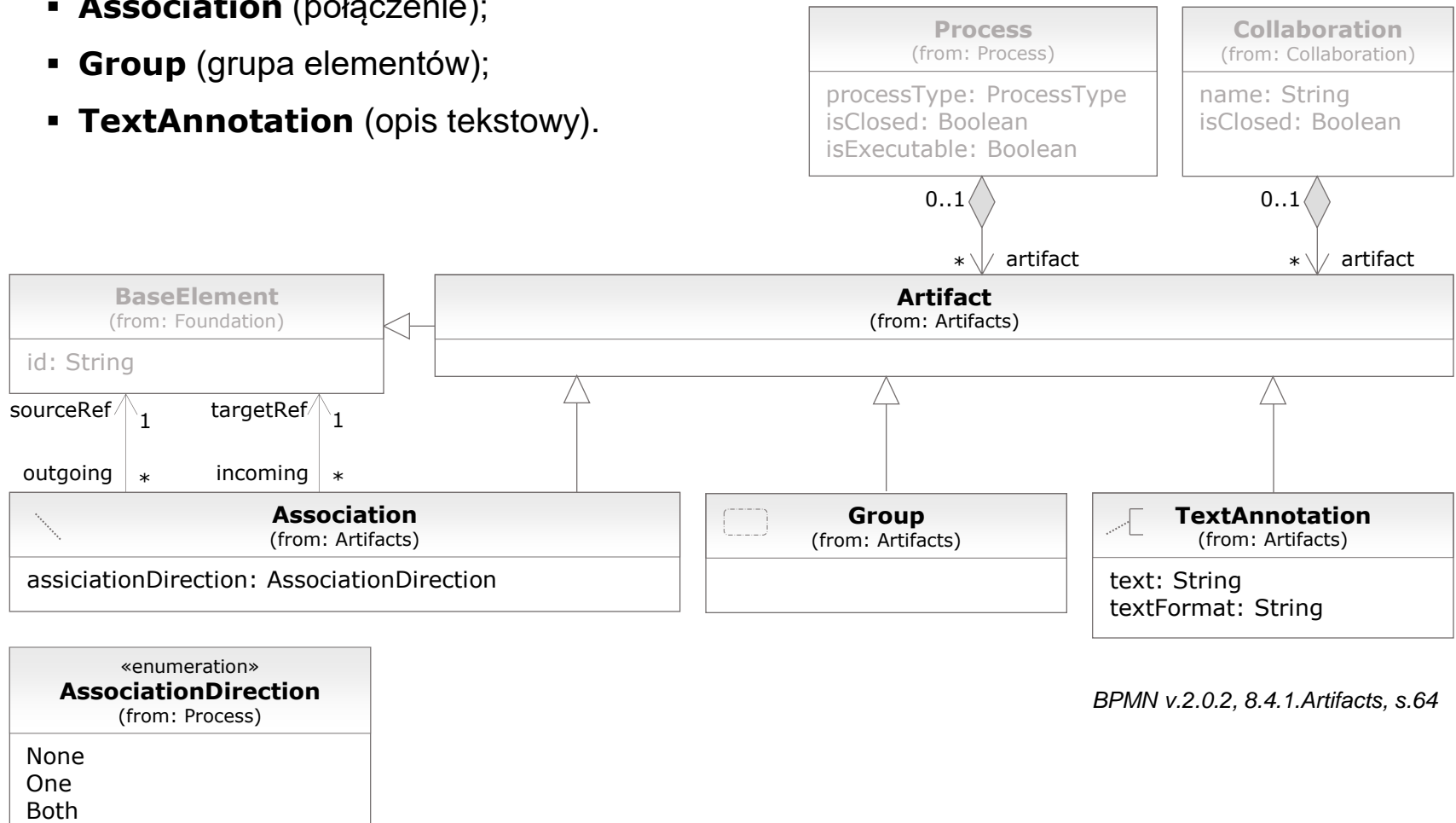
Klasy potomne po FlowElement

- **FlowNode** (węzły): Aktywności Bramki, Zdarzenia;
- **SequenceFlow** (przebiegi sekwencyjne);
- **DataObject** (obiekty danych) oraz **DataStoreReference** (magazyny danych).

Artefakty – klasa Artifact

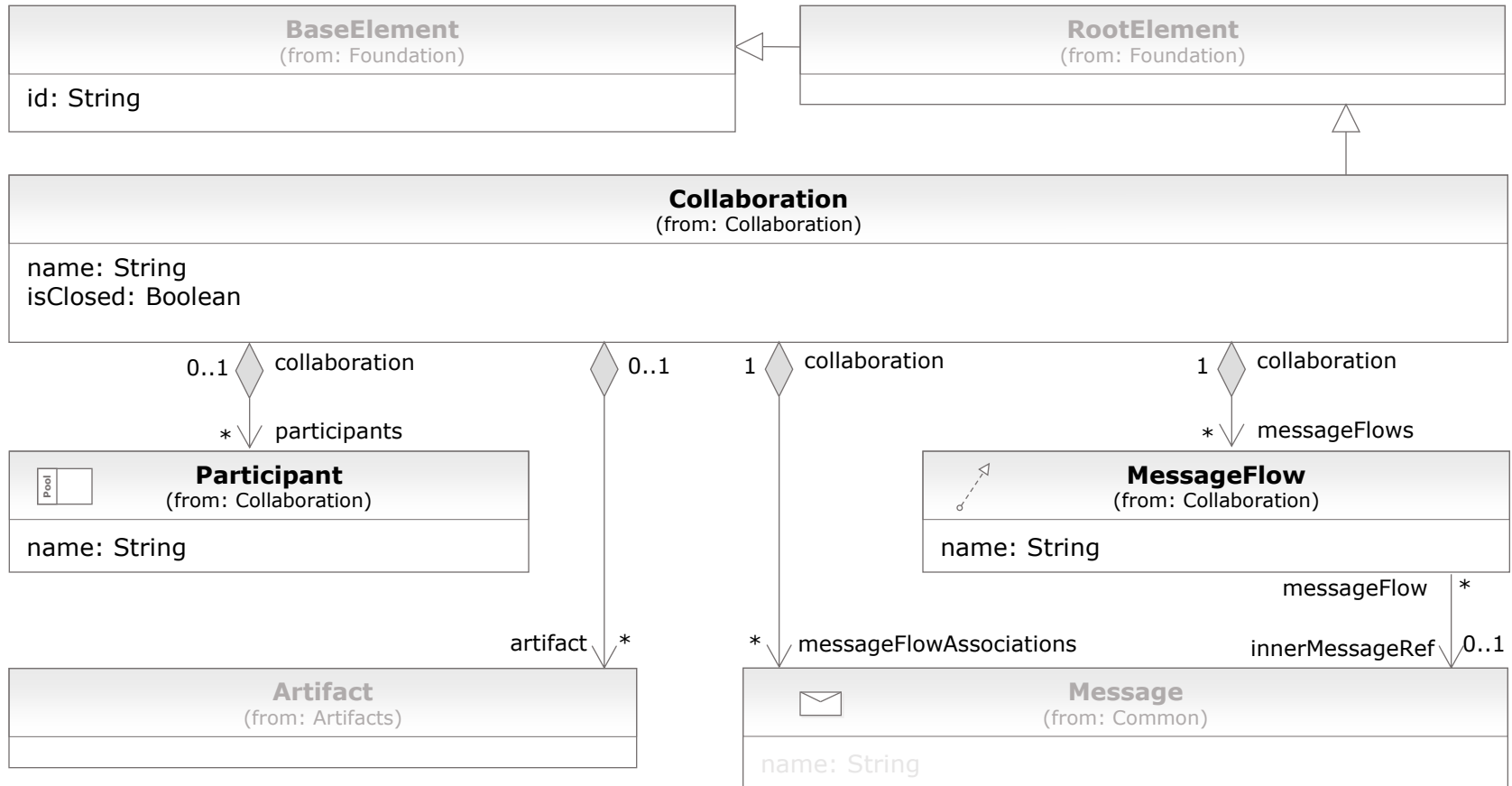
Klasy potomne po Artifact

- **Association** (połączenie);
- **Group** (grupa elementów);
- **TextAnnotation** (opis tekstowy).



BPMN v.2.0.2, 8.4.1.Artifacts, s.64

Diagram kooperacji – klasa Collaboration



BPMN v.2.0.2, 9.1.Collaboration-Genaral, s.108

Atrybuty i powiązania klasy Collaboration

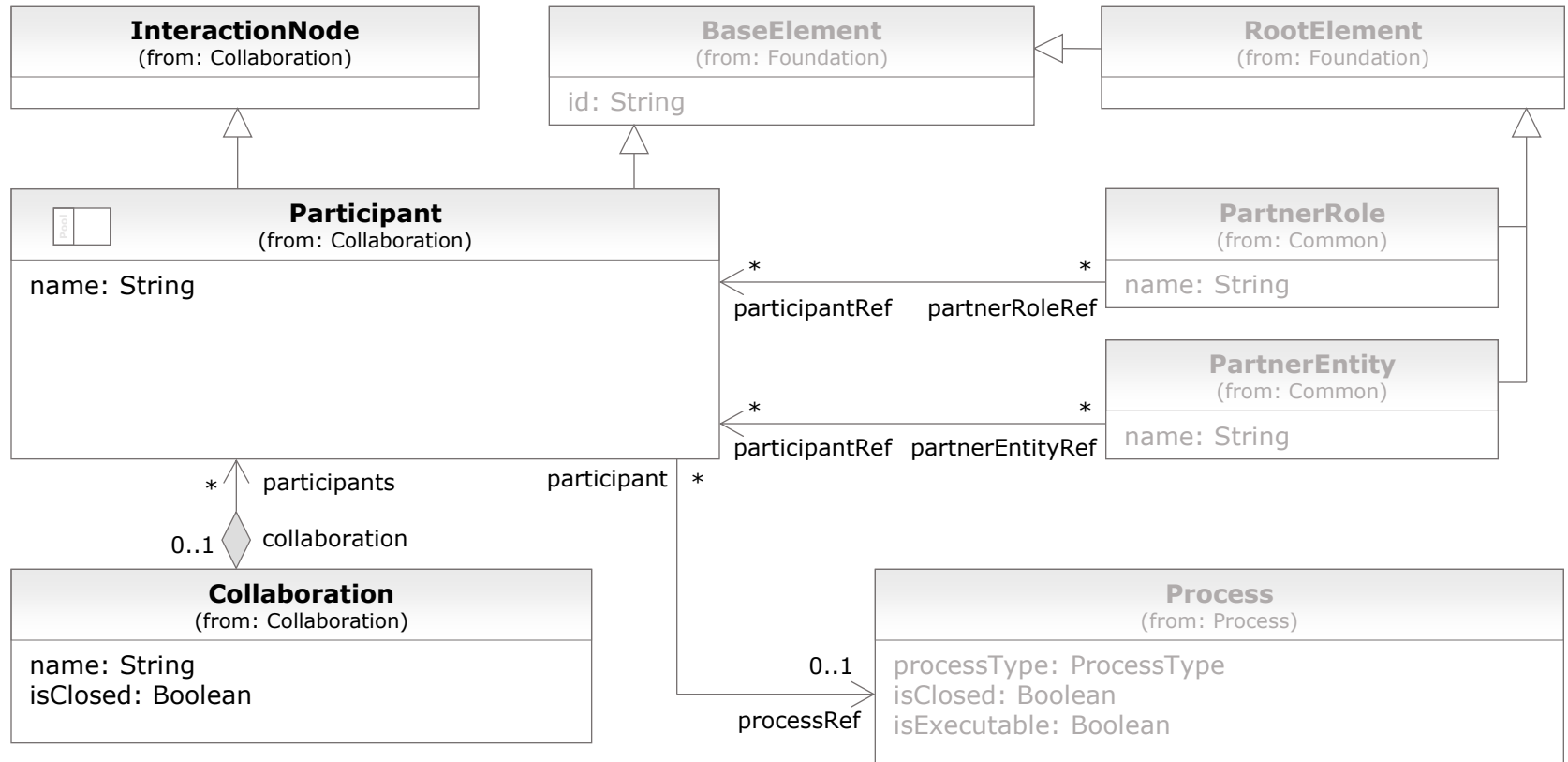
Klasy pakietu Cooperations

- **Collaborations** – reprezentuje Diagram Kooperacji,
- **Participant** – reprezentuje Uczestnika Kooperacji (s.5-23),
- **MessageFlow** – reprezentuje przepływ Komunikatu (s.5-27).

Atrybuty i powiązania klasy Collaboration

- **name**: String – opis tekstowy Kooperacji,
- **isClosed**: Boolean=true – określa możliwość wystąpienia dodatkowych Przepływów Komunikatów, które nie są uwzględnione na diagramie,
- **artifacts**: Artifact [0..*] – lista Artefaktów (s.5-20) powiązanych z Kooperacją,
- **participants**: Participant [0..*] – lista Uczestników (s.5-23) biorących udział w modelowanej Kooperacji,
- **messageFlow**: MessageFlow [0..*] – lista Przepływów Komunikatów występujących w modelowanej Kooperacji (s.5-27).

Uczestnik – klasa Participant



BPMN v.2.0.2, 9.3.1.Participants, s.113

Atrybuty i powiązania klasy Participant

Klasy bazowe Participant

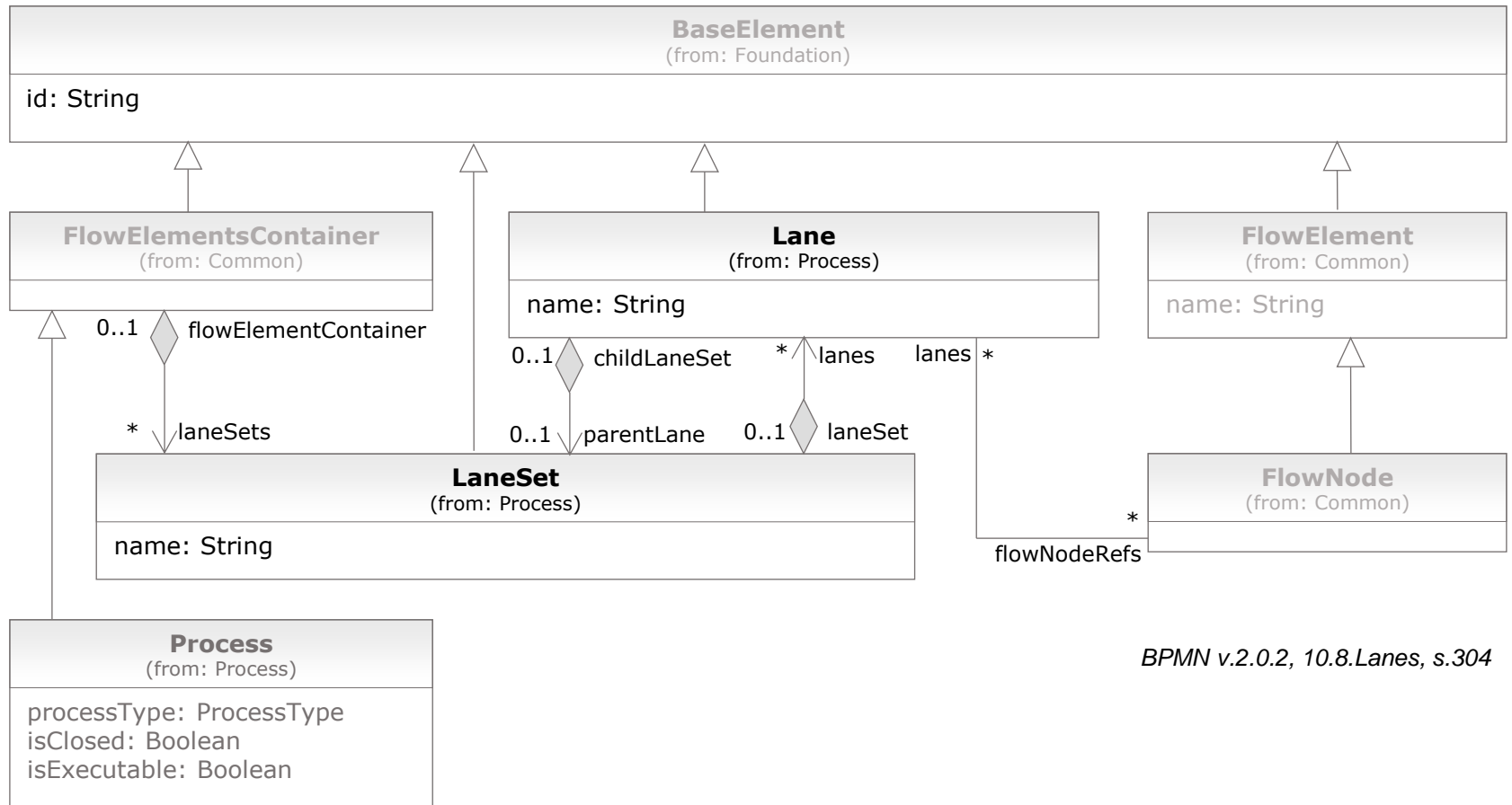
- **BaseElement** – przodek większości elementów BPMN zdefiniowany w BPMN Core,
- **InteractionNode** – klasa elementów, które mogą być źródłem lub celem przepływu komunikatów (**MessageFlow**, s.5-27).

Atrybuty i powiązania klasy Participant

- **name**: string – opis tekstowy Uczestnika,
- **processRef**: Process [0..1] – identyfikuje proces wykonywany przez Uczestnika w Kooperacji,
- **partnerRoleRef**: PartnerRole [0..*] – (*ang. role*: rola) określa rolę, którą Uczestnik pełni w kooperacji (np. sprzedawca, nabywca, pośrednik, itp.),
- **partnerEntityRef**: PartnerEntity [0..*] – (*ang. entity*: jednostka, podmiot) określa rodzaj/typ podmiotu, który reprezentuje Uczestnik (np. przedsiębiorstwo, spółka, osoba fizyczna, itp.).

Uwaga: na diagramie kooperacji uczestnik jest reprezentowany przez Basen (**Pool**), który może być opisany atrybutami **name**, **partnerRole** i **partnerEntity**.

Tory – klasy Lane i LaneSet



BPMN v.2.0.2, 10.8.Lanes, s.304

Atrybuty i powiązanie klas Lane i LaneSet

Tor (**Lane**) dzieli proces na części przypisane do poszczególnych jednostek lub ról realizowanych w obrębie uczestnika.

Atrybuty klasy Lane

- **name**: string – nazwa Toru,
- **childLaneSet**: LaneSet [0..1] – kontener zawierający Tory osadzone,
- **flowNodeRefs**: FlowNode [0..*] – lista elementów umieszczonych w Torze.

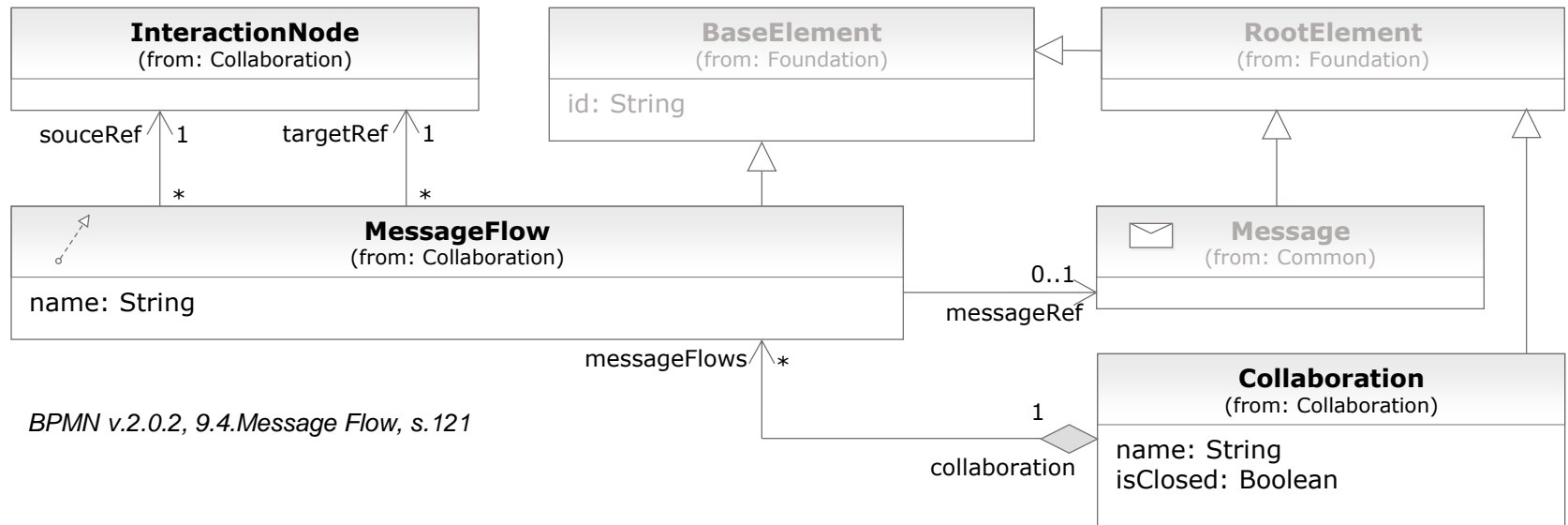
LaneSet jest kontenerem dla jednego lub kilku Torów.

Atrybuty klasy LaneSet:

- **name**: sting – nazwa kontenera (niewidoczna na modelu),
- **process**: Process – proces będący właścicielem kontenera (s.5-14),
- **lanes**: Lane [0..*] – Tory umieszczone w kontenerze,
- **parentLane**: Lane [0..1] – Tor, który jest właścicielem kontenera.

Uwaga: z powiązań pomiędzy klasami **Lane** i **LaneSet** wynika, że: **LaneSet** składa się (powiązanie całość-część) z dowolnej liczby elementów klasy **Lane**, **Lane** może zawierać jeden element klasy **LaneSet**, stąd elementy **Lane** mogą być zagnieżdżone (Tor może być podzielony na kolejne Tory).

Przeływ komunikatu – klasa MessageFlow



Atrybuty i powiązania klasy MessageFlow

- **name**: String – opis tekstowy Przeływu Komunikatu,
- **sourceRef**: InteractionNode – węzeł, który jest początkiem przepływu komunikatu,
- **targetRef**: InteractionNode – węzeł, do którego komunikat jest skierowany,
- **messageRef**: Message [0..1] – komunikat powiązany z przepływem.