|  |  |  |  |
| --- | --- | --- | --- |
| Logotyp  C:\Users\Jerry\Documents\doseLOGO.png | Nazwa organizacji  Politechnika Gdańska | | |
| Symbol projektu  DOSE | Nazwa projektu  Document Oriented System Engineering | | |
| Symbol zadania  IML | Nazwa zadania  Model obiektowy semantyki języka IML | | |
| Typ dokumentu  Analiza istniejących rozwiązań | Tytuł dokumentu  Definiowanie modeli za pomocą MOF | | |
| Data utworzenia dokumentu  2014-07-02 | Numer wersji  1.0.1 | Edycja  1 | Data ostatniej aktualizacji  2014-07-02 |
| Osoba odpowiedzialna  Jarosław Kuchta | Autor tej wersji dokumentu  Jarosław Kuchta | | Status  Szkic |

Spis treści

[1. Wprowadzenie 2](#_Toc391753935)

# Wprowadzenie

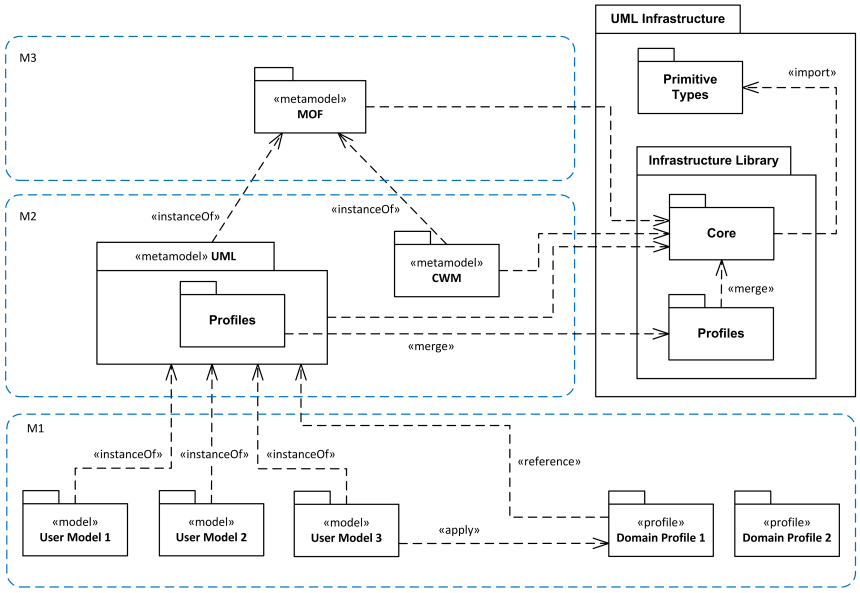
MOF (ang. Meta-Object Facility) jest standardem OMG do definiowania języków modelowania [1]. Względem języków modelowania MOF spełnia taką samą rolę jak notacja EBNF [2] względem języków programowania.

MOF jest opublikowany przez OMG w postaci szeregu specyfikacji:

* MOF 2 Core,
* MOF 2 IDL Mapping,
* MOF 2 XMI Mapping,
* MOF 2 Facility and Object Lifecycle,
* MOF 2 Versioning and Development Lifecycle,
* MOF 2 Query/View/Transformations,
* MOF Model to Text.

Każda z tych specyfikacji może być używana niezależnie. W tym dokumencie w skrócie przedstawiono specyfikację MOF 2 Core.

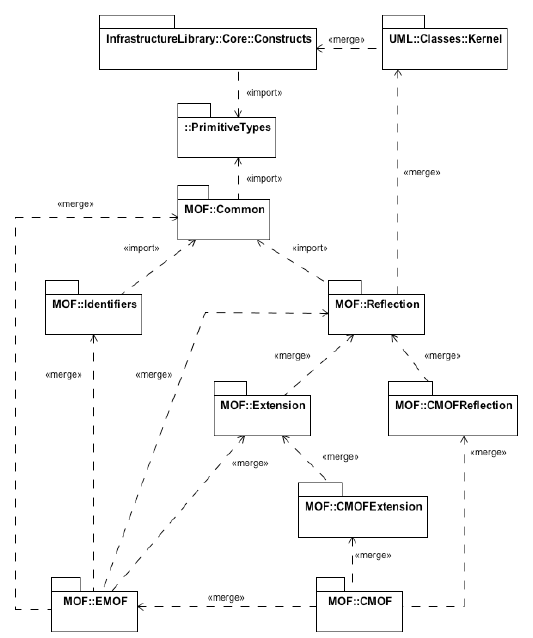
MOF jest zdefiniowany na poziomie M3 meta-modelowania (rys. 1). Poziom M3 (czyli MOF) stanowi język definiowania języków modelowania. Na poziomie M2 w OMG są zdefiniowane dwa języki modelowania: UML oraz CWM. Poziom M1 zawiera już konkretne modele zbudowane w UML. Znaczna część MOF została wykorzystana do opisu infrastruktury UML [3].



Rys. 1. Infrastruktura MOF a UML [4]

# Architektura MOF

Specyfikacja MOF 2 powstawała w tym samym czasie i była opracowywana przez te same firmy i tych samych projektantów co specyfikacja infrastruktury UML 2. Dlatego część pojęć jest wykorzystywana w obu specyfikacjach. W rezultacie MOF 2 importuje część specyfikacji infrastruktury UML 2, a specyfikacja infrastruktury UML 2 włącza do siebie część specyfikacji MOF 2 (rys. 2).



Rys. 2. Architektura MOF 2 w związku z infrastrukturą UML 2

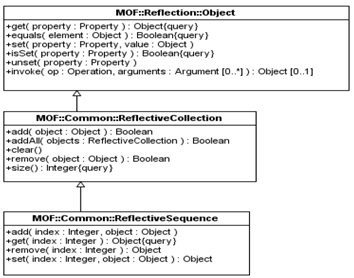
Wszystkie pojęcia MOF 2 są podzielone na szereg pakietów:

* Common – wspólne pojęcia wykorzystywane w całym MOF,
* Reflection – rozszerzenie modelu o możliwości samo-opisywania,
* Identifiers – rozszerzenie modelu o unikatowe identyfikowanie obiektów bez polegania na danych modelu, które mogą się zmieniać,
* Extension – prosta możliwość rozszerzania elementów modelu o pary nazwa-wartość,
* EMOF (Essential MOF) – podzbiór MOF blisko związany z możliwościami obiektowych języków programowania i XML,
* CMOF Reflection – rozszerzone możliwości pakietu EMOF Reflection,
* CMOF (Complete MOF) – określa inne metamodele jak UML 2, łączy EMOF i pakiet Core::Constructs z UML 2,
* CMOF Extension – rozszerzenie możliwości MOF przez CMOF.

## Pakiet MOF::Common

Pakiet MOF::Common zawiera dwa wspólne elementy MOF:

* ReflectiveCollection – kolekcja elementów typu MOF::Reflection::Object,
* ReflectiveSequence – sekwencja elementów typu MOF::Reflection::Object,

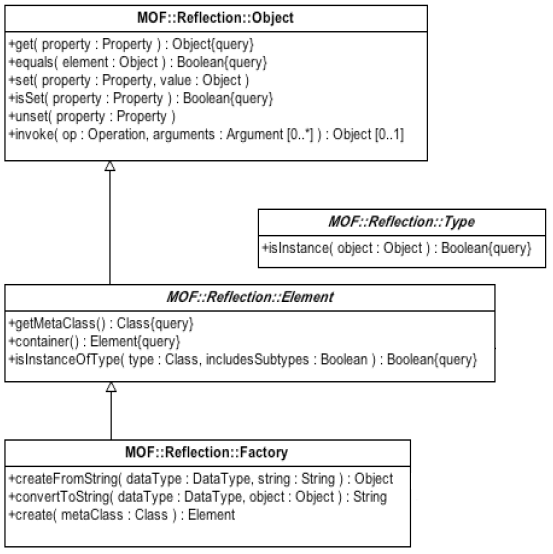


Rys. 3. Pakiet MOF::Common

## Pakiet MOF::Reflection

Pakiet MOF::Reflection definiuje cztery klasy:

* Object – dowolny obiekt,
* Type – typ obiektu,
* Element – obiekt, który ma klasę udostępniającą informację przez refleksję typów,
* Factory – obiekt tworzący obiekty podanej klasy

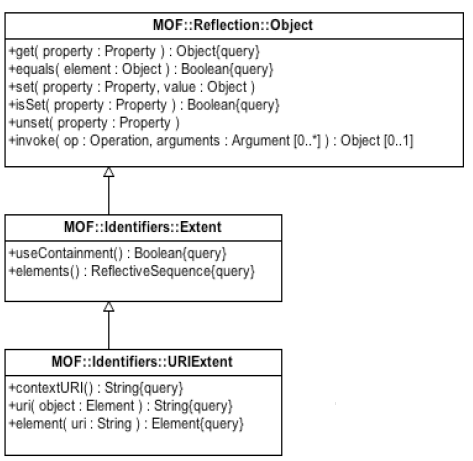


Rys. 4. Pakiet MOF::Reflection

## Pakiet MOF::Identifiers

Pakiet MOF::Identifiers definiuje elementy, które umożliwiają identyfikację obiektów niezależnie od danych, które mogą podlegać zmianie:

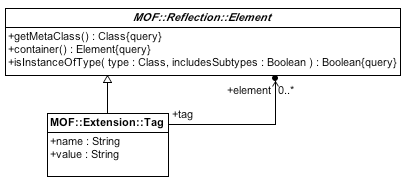
* Extent – kontekst, w którym element jest identyfikowany w zbiorze elementów
* URIExtent – kontekst identyfikacji określany przez URI



Rys. 5. Pakiet MOF::Identifiers

## Pakiet MOF::Extension

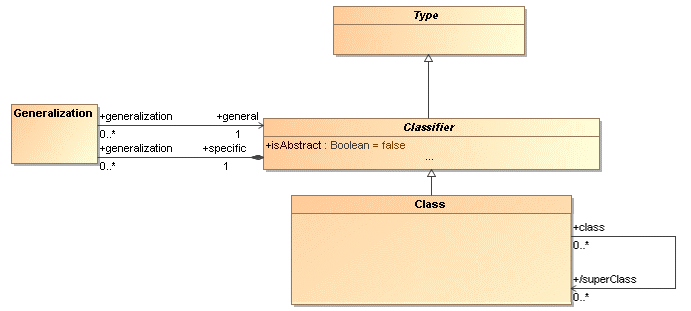
Modele MOF umożliwiają definiowanie takich elementów metamodelu jak klasy, które mają swoje właściwości. Czasami jednak elementy modelu muszą być dynamicznie wzbogacane o dodatkową informację, nie występującą w metamodelu. Umożliwia to element Tag z pakietu MOF::Extension.



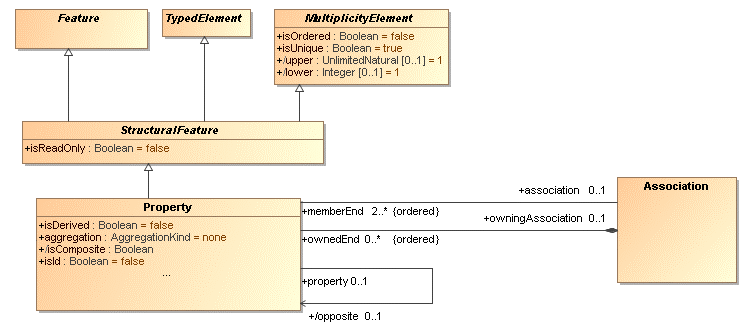
Rys. 6. Pakiet MOF::Extension

## Pakiet MOF::EMOF

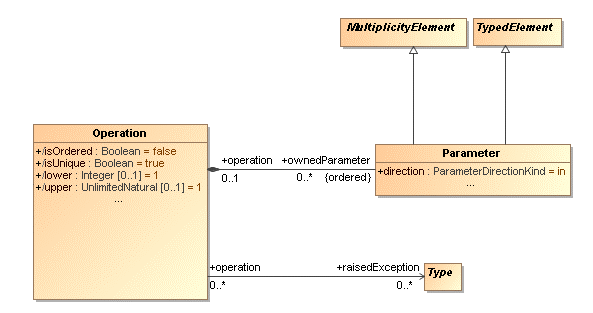
Pakiet MOF::EMOF określa esencję MOF, czyli taki podzbiór obiektów MOF, który ściśle odpowiada pojęciom znajdowanym w obiektowych językach programowania i w XML. Pakiet ten definiuje takie elementy, jak Class, Property, Association, Operation.



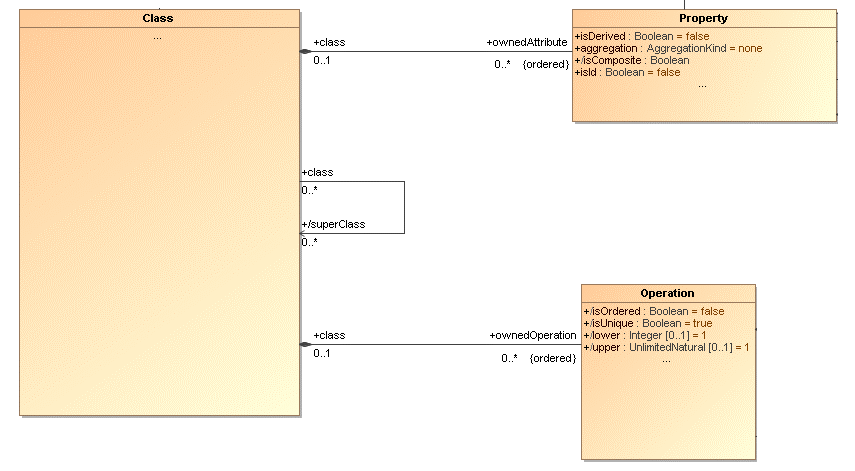
Rys. 7. Definicja klasy w MOF::EMOF



Rys. 8. Definicja właściwości i asocjacji w MOF::EMOF



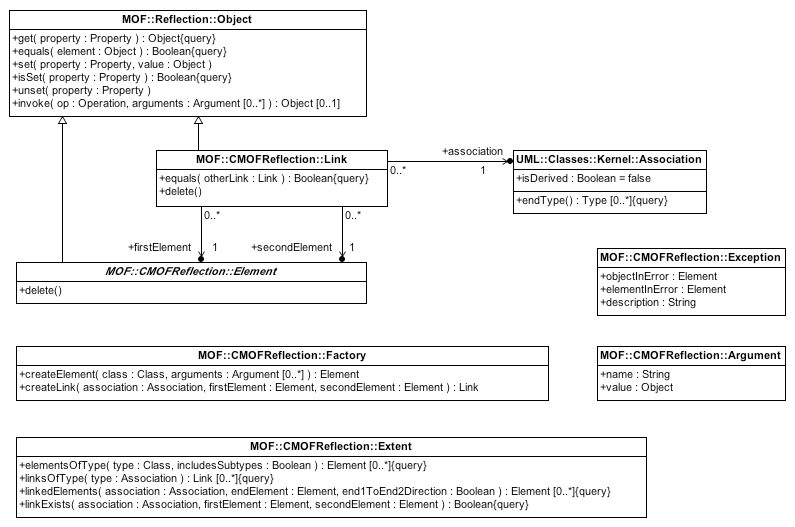
Rys. 9. Definicja operacji w MOF:EMOF



Rys. 10. Powiązanie właściwości i operacji z klasą

## Pakiet MOF::CMOFReflection

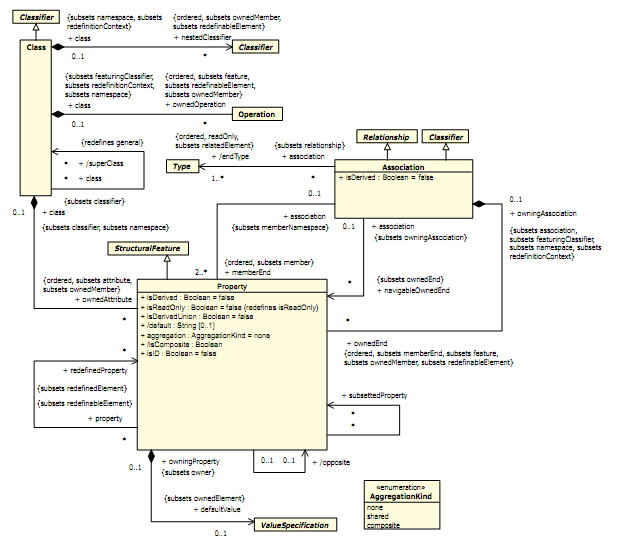
Pakiet MOF::CMOFReflection definiuje elementy rozszerzające możliwości MOF::Reflection dla potrzeb MOF:CMOF.



Rys. 11. Pakiet MOF::CMOFReflection

## Pakiet MOF::CMOF

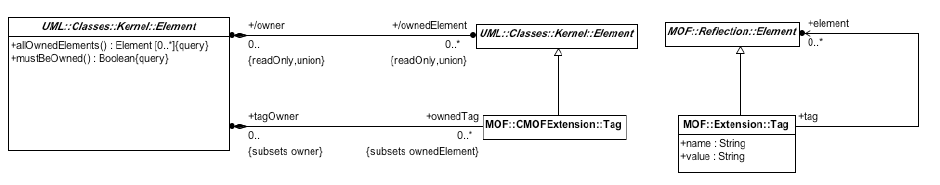
Pakiet ten nie deklaruje własnych elementów, ale łączy inne pakiety, m.in. infrastrukturę UML 2.



Rys. 12. Import elementów z UML do MOF::CMOF

## Pakiet MOF::CMOFExtension

Pakiet MOF::CMOFExtension rozszerza stosowanie Element i Tag do pakietu MOF::CMOF.



Rys. 13. Pakiet MOF::CMOF