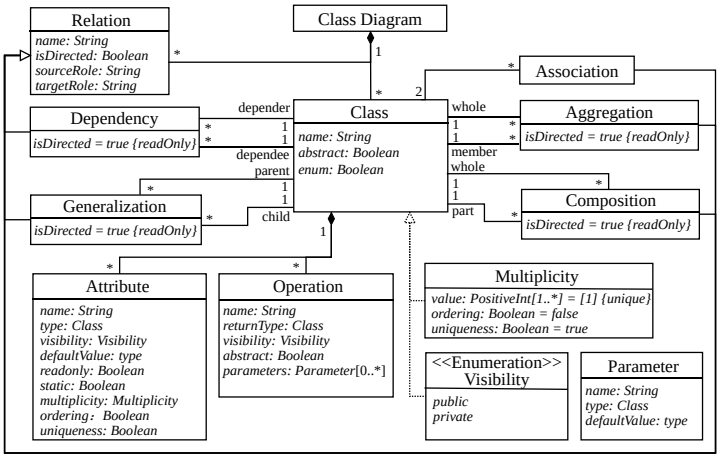

OCL constraint:

context a: Aggregation
inv: a.sourceRole = whole **and** a.targetRole = member
context d: Dependency
inv: d.sourceRole = depender **and** d.targetRole = dependee

context c: Composition
inv: c.sourceRole = whole **and** c.targetRole = part
context g: Generalization
inv: g.sourceRole = parent **and** g.targetRole = child

Fig.3. Meta-model of class diagrams

Table I. Properties of attributes

Propert	Description
name	A string of characters that provides an meaningful identifier to the attribute.
type	The type of the values that held by the instances of the attribute.
visibility	The value of this property can be public or private (see class Visibility in Fig.3), indicating the instances of the attribute can or cannot be directly requested by other objects, respectively.
defaultValue	The value assigned to the instances of the attribute when the attribute is instantiated.
readOnly	Whether the value of an instance of the attribute can be modified (false) or not (true) after the instance is initially assigned with a value.
static	Whether the attribute is static (true) or not (false). If the attribute is static, the attribute can only have one instance, and the instance is shared by all of the objects instantiated from the class which the attribute belongs to; otherwise each object has its own instance of the attribute.
multiplicity	The number of the attribute's instances referenced by the name of the attribute. Only positive integers can be used as the values of this property (see class Multiplicity in Fig.3).
ordering	This property only become meaningful when the multiplicity of the attribute is more than 1. This property indicates whether the instances of the attribute needs ordering (true) according to the value of instances or not (false).
uniqueness	This property only become meaningful when the multiplicity of the attribute is more than 1. This property indicates whether different instances of the attribute can have the same value (false) or not (true).

Table II. Properties of operations

Property	Description
name	A string of characters that provides an meaningful identifier to the operation.
returnType	The type of the values returned by the operation.
visibility	The visibility of the operation. The value of this property can be *public* or *private* (see class *Visibility* in Fig.3), indicating the operation can or cannot be directly requested by other objects, respectively.
abstract	Whether the operation is abstract (true) or not (false). Abstract operations cannot have implementations.
parameters	The parameters of the operation. Each parameter has three properties: *name, type and defaultValue* (see class *Parameter* in Fig.3).