Molecule
name: String
weight: int

compound   1..*

Contains

particle   3..*

Atom
name: String
weight: int
nobleGas: boolean

context Atom **inv** NonNegativeAtomicWeight:

     $self.weight > 0$

context Atom **inv** UniqueAtomicWeight:

     Atom::allInstances()−>isUnique(x | x.weight)

context Molecule **inv** InertNobleGases:

     self.particle−>(x | x.nobleGas)−>isEmpty()

context Atom **inv** AtLeastOneNobleGas:

     Atom::allInstances()−>exists(x | x.nobleGas)

Figure 1: Running example: a UML class diagram with OCL constraints

This example is unsatisfiable:

- There must be at least one noble gas (invariant *AtLeastOneNobleGas*).

- Noble gases do not participate in chemical reactions, so they do not appear in any molecule (invariant *InertNobleGases*).

- Due to the multiplicity of association Contains, each atom belongs to at least one molecule (multiplicity 1..*).

It is possible to turn this model into a satisfiable one by removing the invariant *AtLesatOneNobleGas*. In this way, it will be possible to populate the classes with objects. Notice, however, that it will be impossible to create an atom which is a noble gas.