

## Complex conditions

In real contexts it is often needed to have more complex statements about deontic clauses, which combine atomic statements in structured logical constructs.

To fulfil this requirement, the abstract class named `mco-core:FactComposition` is declared, which extends `mvco:Fact` and three subclasses of this new class modeling logical intersection ("AND" operator) and logical union ("OR" operator), and logical negation ("NOT" operator), respectively named `mvcocel:FactIntersection`, `mco-core:FactUnion` and `mco-core:FactNegation`. This approach on one hand leaves unvaried the construct and the semantics defined in the MVCO between `mvco:Permission` and `mvco:Fact`, and on the other hand it extends the expressiveness of the model.

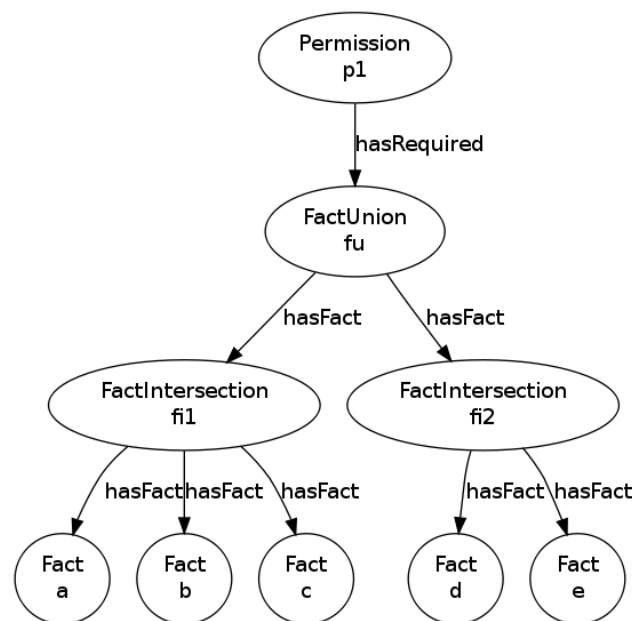
The semantics of the abstract class `mco-core:FactComposition` is that of collecting a set of Facts taking part in the specification of a complex `Permission`. For this purpose, an object property `hasFact`, whose range is `mvco:Fact`, is given. This, together with the stated hierarchy between `mvco:Fact` and `FactComposition`, implies that also concrete instances of subclasses of `FactComposition` can be included as elements in a recursive way, thus allowing for (potentially) infinite-nesting of logical clauses.

### Example

The semantics of subclasses of `FactComposition`, namely `FactIntersection` and `FactUnion`, is that of exactly specifying the logical form of these clauses. The following logical statement:

( a AND b AND c ) OR ( d AND e )

can be expressed as shown in Figure 14 — Diagram and RDF/XML excerpt of a fact composition example, in which the diagram of the left provides the graphical representation of the RDF/XML excerpt on the right.



```
<mvco:Permission rdf:about="p1">
  <mvco:hasRequired
  rdf:resource="fu"/>
</mvco:Permission>
<mco-core:FactUnion rdf:about="fu">
  <mco-core:hasFact
  rdf:resource="fi1"/>
  <mco-core:hasFact
  rdf:resource="fi2"/>
</mco-core:FactUnion>
<mco-core:FactIntersection
  rdf:about="fi1">
  <mco-core:hasFact
  rdf:resource="a"/>
  <mco-core:hasFact
  rdf:resource="b"/>
  <mco-core:hasFact
  rdf:resource="c"/>
</mco-core:FactIntersection>
<mco-core:FactIntersection
  rdf:about="fi2">
  <mco-core:hasFact
  rdf:resource="d"/>
  <mco-core:hasFact
  rdf:resource="e"/>
</mco-core:FactIntersection>
```

```
<mvco:Fact rdf:about="a"/>  
<mvco:Fact rdf:about="b"/>  
<mvco:Fact rdf:about="c"/>  
<mvco:Fact rdf:about="d"/>  
<mvco:Fact rdf:about="e"/>
```

**Figure 1 — Diagram and RDF/XML excerpt of a fact composition example**