Let :

* attributes = set of the attributes (belonging to entity types or relationship types)
* entities = set of the entity types

= {},

such as:  
 - that describes the composition of   
 --,   
 with and is an attribute of

* relations = set of the relationship types

= {},

such as:   
 :: ,   
- x- ,  
- ,

E entities : descr-E (), avec n

1. Generate class pojo.E

* E, E’ : private List<E’> E’ ;
* E, E’ : private E’ E’ ;
* E, : private List<R> RList ;
* E, : private R R ;
* E, E’, : private List<R> RList ;
* E, E’, : private R R;

1. Generate class EService

INSERT

* insertEandLinkedItems(E)
* if //s’il n’existe pas de rôle [1- ?] défini sur E

then insertE(E)

else

//s’il n’existe pas de rôles opposés obligatoires

if

then insertE(E, {}, {}), //Ei étant un objet persistant

with

with :

SELECT

* E[] getEList(ECondition)
* E[] getEListBy
* E, ,
  + - then
    - else

Max :

* + //role obligatoire
    - Then

UPDATE

* updateEList(ECondition, ESetClause)
* update(E) // utilise le E.id en guise de ECondition
* E, ,
  + - then
    - else
    - then
    - else

DELETE

* deleteEList(ECondition)
* delete(E) //utilize le E.id en guise de ECondition
* E, ,
  + - then
    - else

R(), avec n

1. Generate class pojo.R
2. Generate class RService

INSERT

* insertRandLinkedItems(R)
* if ,

then attachPersistentItemsByR()

SELECT

* if m = 0
  + then getRList()
  + else getRList()
  + then // utilise l’id en guise de condition
  + else
* If m > 0 then getRListByRCondition(RCondition)

UPDATE

* if m = 0
  + then updateRList()
  + else updateRList(, RSetClause)
  + Then // utilise l’id en guise de condition
  + Else // utilise l’id en guise de condition
* If m > 0 then updateRListByRCondition(RCondition, RSetClause)

DELETE

* if m = 0
  + then deleteRList()
  + else deleteRList()
  + then // utilise l’id en guise de condition
  + else
* If m > 0 then deleteRListByRCondition(RCondition)